The Nottingham Trent University Library & Information Services SHORT LOAN COLLECTION

Pate	Time	Date	Time
2 - 548	AND A Am		
		The real of	
o the state		THE SHE WAS	
	Million Party of the lat		

Please return this item to the Issuing Library. Fines are payable for late return.

THIS ITEM MAY NOT BE RENEWED

Short Loan Coll May 1996

ProQuest Number: 10183555

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10183555

Published by ProQuest LLC (2017). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code Microform Edition © ProQuest LLC.

ProQuest LLC. 789 East Eisenhower Parkway P.O. Box 1346 Ann Arbor, MI 48106 – 1346

MMPHIL 6 .

A Gaming Simulation of Manufacturing Organisations

by

Juan Ignacio Igartua

A thesis submitted in partial fulfilment of the requirements of the Council for National Academic Awards for the degree of Master of Philosophy.

October 1991

Nottingham Polytechnic



ACKNOWLEDGEMENTS

Firstly, I would like to express my gratitude to my supervisor Dr Graham Tranfield for his advices, constant support, and guidance throughout the fulfilment of this project and specially in the writing process.

Secondly, I would like to thank Justino Unamuno, Iñaki Elguezua, Jesus Santa Cristina, and Juan Galardi for their help and friendship since all of us met in England to extend our knowledge in the production area.

I would also like to thank Phil Moore for his technical help during the completion of this work.

Special thanks to the people in the office Farhad Fassihi, Steve Wood, and Alex Seiler, who have all become very good friends.

Most of all, I would like to thank my parents and two sisters for their continual support, patience and the sacrifices they have made throughout my education. At this stage, I would also like to remember my grandfather, who unexpectedly died during the study of this work.

ABSTRACT

Manufacturing organisations are inevitably divided into individual departments at some level. Each of these departments will have their own objectives and the success of the organisation depends on the extent to which these departmental objectives are in line with the overall objectives of the organisation. They must coordinate their efforts towards the objectives of the whole organisation, forgetting their departmental rivalries. Sadly this is not the case in many western manufacturing companies.

Many companies would therefore like to realign their departmental objectives and an important step in this process is to educate their staff. They must understand how their decisions affect other parts of the company and through this the overall company performance. This is inevitably a dynamic and cross functional process, as decisions are being made throughout an organisation in response to a continually changing environment.

Therefore, education can no longer be seen as a static and independent process, where people only understand how they work, and how others work in a stand-alone basis. What is needed is a new and more integrative approach, where people understand not only how they work and why, but also how they affect other areas of their manufacturing organisation, in other words how their policies and performances fit within the policies and performances of the whole organisation.

Computer games appear to be appropriate tools for that integrated educational approach. Players are not only able to experiment with technical variables, but also they are able to experience a fundamental aspect of organisational life, that of bargaining and negotiation in the decision-making process. They are faced with the inevitably interdepartmental rivalry so common in organisations, and with the conflict between this rivalry and the need to cooperate in a company basis.

Work therefore took place at Nottingham Polytechnic to develop a dynamic, multi-user, computer based gaming-simulation which was designed to highlight the interactions between departments.

TABLE OF CONTENTS

110.8

1.2 1 24

Ya . messa

Page **1 INTRODUCTION** 1

A fance water a sub to be

2 MANUFACTURING ORGANISATIONS: COORDINATION AND EDUCATION	5
2.1 INTRODUCTION	5
2.2 THE STRUCTURE OF MANUFACTURING ORGANISATIONS	6
2.2.1 INTRODUCTION	6
2.2.2 FUNCTIONAL STRUCTURE	9
2.3 COORDINATION IN MANUFACTURING ORGANISATIONS	11
2.3.1 TECHNOLOGICAL IMPLEMENTATION	15
2.4 DEALING WITH COORDINATION	19
2.4.1 INTRODUCTION	19
2.4.2 THE ARMY SOLUTION	20
2.4.3 THE RELIGION SOLUTION	21
2.4.4 THE GOAL DICTATES THE SOLUTION	22
2.4.5 EXAMPLES OF APPLIED COORDINATION	23
2.4.5.1 WESTERN COUNTRIES' APPROACH	23
2.4.5.2 JAPANESE'S APPROACH	25
2.4.6 TOWARDS THE "GOAL DICTATED" COORDINATION APPROACH	26
2.4.6.1 INTRODUCTION	26
2.4.6.2 DEFINE OBJECTIVES	27
2.5 THE ROLE OF EDUCATION	31
2.5.1 INTRODUCTION	31
2.5.2 EDUCATING FOR CHANGE	32
2.6 CONCLUSIONS	34
3 METHODOLOGIES FOR EDUCATION AND TRAINING	36
3.1 INTRODUCTION	36

3.2 EDUCATION AND TRAINING THROUGH COMPUTERS 37 3.3 BUSINESS GAMING-SIMULATIONS, WHAT ARE THEY ? 39 3.3.1 INTRODUCTION 39 40

3.3.1.1 THE DEVELOPMENT OF THE THEORY OF GAMES

S. 82. 25

Stort also as

18. 14.24

3.3.1.2 THE ADVENT OF DIGITAL COMPUTERS	40
3.3.2 BUSINESS GAMES	41
3.3.3 GAMING-SIMULATIONS	43
3.3.4 BUSINESS GAMES AND GAMING-SIMULATIONS AS EDUCATIONAL TOOLS	44
3.4 PAST AND FUTURE TRENDS IN BUSINESS GAMING-SIMULATIONS	48
3.4.1 INTRODUCTION	48
3.4.2 TRADITIONAL MODELS	49
3.4.3 MODERN MODELS	52
2. E. CONGLINSTON	55

14.14

12 .00

1.2. 19

5

. 7.

4 NOTTINGHAM'S POLYTECHNIC GAMING-SIMULATION	57
4.1 INTRODUCTION	57
4.2 TOTAL ENTERPRISE BUSINESS GAMES	58
4.3 THE DESIGN PROCESS	60
4.3.1 INTRODUCTION	60
4.3.2 SETTING OBJECTIVES AND PARAMETERS	64
4.3.2.1 THE SUBJECT MATTER	65
4.3.2.2 PURPOSE	66
4.3.2.3 PLAYERS' CHARACTERISTICS	67
4.3.2.4 OPERATOR CHARACTERISTICS	69
4.3.2.5 CONTEXT OF USE	69
4.3.2.6 RESOURCES	70
4.3.3 MODEL DEVELOPMENT	71
4.3.3.1 BUILD OF THEORETICAL COMPLEX MODEL	72
• Functions	73
Elements of the model	74
4.3.3.2 SIMPLIFY MODEL	77
4.3.3.3 ORDERS' APPROVAL	80
4.3.3.4 FEEDBACK	82
4.3.3.5 THE TIME ELEMENT	85
4.3.4 DECISIONS ABOUT REPRESENTATION	87
4.3.4.1 STYLE	88
4.3.4.1.1 LEVEL OF ABSTRACTION	88
4.3.4.1.2 TIME FRAME	88

taiss the wat

4.3.4.1.3 INTERACTION	90
4.3.4.2 FORM	90
4.3.4.2.1 SCENARIO	90
4.3.4.2.2 ROLES	91
4.3.4.2.3 PROCEDURES AND RULES	92
4.3.4.2.4 EXTERNAL FACTORS	93
4.3.4.2.5 VISUAL IMAGERY AND SYMBOLS	93
4.3.4.2.6 ACCOUNTING SYSTEM	94
4.3.5 CONSTRUCTION AND MODIFICATION OF THE GAMING-SIMULATION	98
4.3.6 PREPARATION FOR USE BY OTHERS	99

3

5 SOFTWARE DESIGN	101
5.1 GENERAL ARCHITECTURE	101
5.2 USERS' VIEW OF THE GAME	102
5.2.1 MENU SYSTEM	103
5.2.2 BROWSING SYSTEM	106
5.2.3 REPORT SYSTEM	109
5.2.4 MESSAGES	112
5.3 DEALING WITH TECHNICAL PROBLEMS	116
5.3.1 FILE CONFLICTS	117
5.3.2 COORDINATION	117
5.3.3 TIME SIMULATION SYSTEM (RUN FACTORY)	120
5.3.4 PRODUCTION SIMULATION SYSTEM (RUN FACTORY)	121
5.3.5 QUOTATION APPROVAL SIMULATION SYSTEM (FINISH OF A DAY)	123
5.3.6 SHIPPING SIMULATION SYSTEM (FINISH OF A DAY)	123
5.3.7 FEEDBACK SIMULATION SYSTEM (FINISH OF A DAY)	124
5.3.8 ENQUIRIES SIMULATION SYSTEM (FINISH OF A DAY)	125

6 CONCLUSION

126

REFERENCES

Appendix A. DEMING'S THOUGHTS ABOUT CO-OPERATION.	
Manufacturing Conflicts.	138
Appendix B. WORKSHOP OF MANAGEMENT SIMULATIONS.	150
Appendix C. GAMES'S SOFTWARE.	204
Appendix D. EVALUATION OF THE GAME.	321
Appendix E. SEVENTH NATIONAL CONFERENCE ON PRODUCTION RESEARCH.	325

State State State

5, nr 1

1.18. 1. 1. 1.

un en Kal Silvanan

in the second second

5

en entrette de la grande d'anne de la grande d La grande de la grand 1. 2. 5 24

an a dalar an shina d

1 INTRODUCTION

To compete in the world markets of the 1990's it is necessary to do more than to simply seek cost efficiency as manufacturing companies have done in the past.

It is evident that market conditions have become more dynamic, more global, and more customer driven. Product price is also no longer the main factor affecting business performance. Other non-price competitive factors such as quality, design, delivery, and customer service have become equally, if not more important.

Moreover, all these market factors are dynamically changing at a rate that forces manufacturing organisations to change themselves into more flexible and dynamic organisations. They are compelling manufacturing organisations into a continuous changing process. Manufacturing managers have realised there needs to be continuous improvements in quality, reduced cost, reduced inventory, shortened flow times, and improved customer relations. They believe these changes must be implemented to maintain or improve their competitive position.

As it is clearly stated by Hayes, Wheelwright, and Clark [1], in the book Dynamic manufacturing, creating the learning organisation, "It is manufacturing management's responsibility to change." Indeed as an educational video [2] states in the opening minutes:

"Change - timeless as life itself, certain as tomorrow's sunrise - change and our response to it are the driving force in today's manufacturing environment. Sometimes simple, often complex, change is inevitable..."

The implementation of new technologies and new management techniques has been companies' first response to this complex changing environment in order to regain or obtain the successful competitiveness needed in this customer driven market.

But what companies will have to learn, if they have not already done so by experience, is that there is no such thing as a quick technological fix. New technology is important but is not the answer, it must not come first [3]. In the majority of cases, it is people who transform raw materials into profitable high quality products. New technology and techniques, as well as organisational change must be shaped by this fact, and by the organisational requirements and needs to achieve continuing improvement. People and the question of how best to help, coordinate, and support them, will be some of the central manufacturing issues of the 1990s. Technology will still be important, but companies can no longer afford to use technology to minimise the role of people in the organisation because they are, in many cases, the key to business success. Companies will have to make and implement a strategic decision to put people first, to build organisations that support these people and their functions, and to use technology as a means of making more effective use of peoples' skills and abilities. The keywords are empowering, teamworking, reduced hierarchies, decentralised control and decision making, and coordinated organisational structures.

However, before any of the above can be achieved, companies will have to "fight" against organisational problems, i.e. outdated attitudes, resistance to change, over specialised narrowly based roles, vested interests, power relationships, conflicts, company politics and cultures. But what is more important, they will have to change the outlook of company employees from that of departmental performance defenders to that of coordinated individuals and functions engaged in a common goal of ongoing improvement in company competitiveness, profitability and business performance. To carry out this first task manufacturing organisations will have to understand how each of the functions' decisions affect other functions of the manufacturing organisation and through that, the overall company performance. This will inevitably be a dynamic educational process, as in real life, that could be carried out with the aid of computer based educational tools, more specifically computer based games and simulations.

This thesis is therefore an evaluation of computer based gaming simulations as tools for education in this area.

It starts with a review of the importance of coordination among the areas of a manufacturing organisation working in a changing environment.

Chapter 3 then reviews how computer games are used as educational tools in a variety of dynamic environments and in particular looks at the different approaches to education in relation to manufacturing management.

The main part of the work for this thesis was then the construction of a computer based gaming-simulation specifically designed to address the problems of coordination in manufacturing organisations. A paper reporting the progress of this work was presented at the seventh National Conference on Production Research [97]. A complete copy of this paper can be found in Appendix E.

The methodology for developing the game is described in chapter 4 and its software design is described in chapter 5.

The conclusion then discusses the effectiveness of this game and the future of gaming-simulations in the manufacturing area in general.

2 MANUFACTURING ORGANISATIONS: COORDINATION AND EDUCATION

2.1 INTRODUCTION

Managing organisations is of particular interest in this century. In the past large organisations were restricted to basically three types of organisations governments, armies and religions. But this is not the case any more. From the beginning of the nineteenth century new types of large organisations have arisen, the industrial and services companies being the most clear examples [4]. Nowadays, the number of industrial companies that exceed ten thousand employees is probably at least two orders of magnitude greater than the handful of such large companies in existence during the previous century.

In the coming sections of this chapter the existence of two very important issues manufacturing organisations will have to address in order to succeed in today's dynamic environment will be shown: coordination and education.

In order to have a better picture when explaining the importance of these two elements, it will first be explained how manufacturing organisations are structured and the different existing alternatives.

In a second step, the importance of coordination in a changing environment, and the different ways it can be achieved will be explained. The relation between the way manufacturing organisations are coordinated and their

associated success is also identified.

In a third step, the links between **coordination** and **education** will be established, and hence the importance of **education** in today's manufacturing organisations will be highlighted.

2.2 THE STRUCTURE OF MANUFACTURING ORGANISATIONS

2.2.1 INTRODUCTION

Manufacturing organisations have, since their development, been structured in a **hierarchical pyramid of command** as shown in figure [2.1]. The reasons for the utilisation of this kind of structure must be found in human's span of control, individuals are only able to directly handle relatively few people compared to the number of people in the organisation. This way of organising leads to problems related to the fact that the transactions needed to provide products and customer services are not hierarchical, they are horizontal and not vertical as it is shown in figure [2.2].



Figure 2.1 Hierarchical Structure



Figure 2.2 A fundamental problem. Taken from Plowan [93]

Another element inherent in all kind of organisations, including manufacturing ones, and somehow related to the hierarchical pyramid of command is the specialisation of labour.

In his 1776 book, **The Wealth of Nations**, Adam Smith explained how he was able to increase the productivity of a group of pin makers by more than a thousand-fold through specialisation (division) of labour. **Specialisation of labour** means the division of a complex job into simpler tasks so that one person or group may carry out only identical or related activities [5]. The purpose of organising is, mainly, to improve productivity. Through specialisation of labour, it is possible for members of the manufacturing organisation to concentrate in a single area, resulting in increasing output. In nearly all manufacturing organisations, most of the work activities are of a specialised nature.

Another element that appears in conjunction with the idea of **specialisation of labour** is the one of **size**. Manufacturing organisations were at the start small ones and easy to manage. In a fledgling company all staff pitch in willingly and do each each other's jobs if need be. Need creates a vital team spirit. But

as the company grows it becomes more complex to manage, and due to the specialisation of labour it becomes functionally differentiated.

These two reasons, specialisation of labour and size has caused manufacturing organisations to structure their organisations in departments, each of them involved with a specialised issue.

The process of dividing organisations into departments is defined as a way of grouping related work activities into manageable units in order to contribute to a more effective and efficient use of the manufacturing resources available.

The most common departments to be found in a manufacturing company are: Product Design, Production Planning, Production Control, Purchasing, Marketing, Finance, and Personnel. Based on the same elements, manufacturing organisations have different ways of creating their departmental division, each of which is based on a particular specialisation element (function, product,..). These different ways of breaking down into departments are [5]:

- By function.
- By product.
- By customer.
- **By geographic territory.**
- By project.
- By a combination approach.

2.2.2 FUNCTIONAL STRUCTURE

Although the existence of all previously mentioned possible alternatives, most of manufacturing organisations are divided, at some stage, in a functional way, where different activities are performed by different functions.

A management function is defined as sets of closely related management tasks, which require similar skill for their efficient performance.

These functions form a basic classification of the science of Production Management (figure [2.3]). Each function contains sets of related tasks, each of them requiring planning, direction and control. Planning and control tend to be special to each function. The idea of direction is more general by nature.

FUNCTION	TYPE OF TASK	CONTROLS	INPUTS	OUTPUTS
PRODUCT DESIGN	Plans final form of product	Quality Control	ldeas, Market research, R&D.	Parts Hist, Drawings.
PRODUCTION PLANNING	Plans how product is to be made	Process Control, Maintenance.	Parts Hist, Drawings, Sales forecasts	Make or Buy, Plant list, Layout, Routes, Operation times, Tooling.
PRODUCTION CONTROL	Plans material supply and processing activities	Progressing, Loading, Inventory Control	Sales prog., Parts list, Plan list, Routes, Op. times. Make or Buy.	Programmes, Orders, Purchase delivery, Schedules.
PURCHASING	Finds sources Supply contracts	Purchase, progressing.	Purchase delivery. schedules, Make or Buy.	Purchase Orders.
MARKETING	Finds/develops markets, Sales and distribution	Sales control.	Sales orders	Sales records, Sales programmes.
FINANCE	Plans investment, profit and cash flow.	Budgetary Control, Standard costing.	Annual prod. progs., involces in/out, Bank payments in/out.	Budgets, Accounts, Balance sheet, Proflit and Loss Acc
PERSONNEL	Plans employment, conditions welfare, training, promotion.	Merit rating, Attendance.	Hired, Fired, Promoted, Retired.	Employees list, Conditions of employment.
SECRETARIAL	Plans communications and data processing.	Data control.	Production system design.	Software.

Figure 2.3 Management functions. Taken from Burbidge [34]

This division of manufacturing organisations into separate functions was developed during the industrial revolution and based on the "scientific management" philosophy, advocated by Frederick Taylor, whereby functional experts each qualified in a different area reported to a general manager [6].

By grouping related functions, manufacturing organisations form their departments on the basis of specialised activities such as finance, marketing, production, engineering, personnel, and so on, as seen in figure [2.4].



Figure 2.4 Functional departments.

This division of manufacturing organisations by function, although useful in environments where technical efficiency and quality are important, creates certain problems. Employees in specialised departments may become more concern about the functioning and performance of their own department rather than about the overall company's.

Responsibility for a particular functional or other organisational unit tends to narrow the focus of those who manage them. The resulting parochial behaviour can lead to unfortunate outcomes, affect a firm's competitiveness, unless the integration across those separate units is well managed. To paraphrase Winston Churchill, "*Each person has only to do his duty to wreck*

the world."

An example of this assertion is the conclusion reached by a survey conducted in the UK in May 1990, called Managing change in the 90's [7]:

"Compartmentalisation in an organisation creates diverseness and breeds conflict which is a significant obstacle for change. Some managers' clear awareness of this does not seem to have had a major impact on the types of initiatives planned for the future."

Because of all this and taking into account the usually conflicting purposes of various departments in a functionally based structure, managers must ensure that an effective means of coordination exists among all the departments taking part in the organisation's functioning.

Note: Some more explicit examples of the existing functional conflicts and their financial influences can be seen in Appendix A.

2.3 COORDINATION IN MANUFACTURING ORGANISATIONS

With manufacturing organisations structured in a functional way, coordination of those different functions is of vital importance for the success of the company. An important contribution on the matter of coordination and its importance is addressed by Galbraith is his theory of co-ordination and control [8], where he talks about the strategies organisations can use to coordinate and control the people and processes who comprise them. His starting point is the assumption that for an organisation to perform satisfactorily it must be able to coordinate the activities of its various elements or subunits, be they individuals, groups, or departments, in an effective way. The ability to bring different specialists together is and will become a critical differentiator, determining success or failure.

Another example corroborating the importance of coordination, in general terms, is the one expressed by Kiyoshi Suzaki [9], where he compares a well-performing orchestra to a well-performing factory. Each musician has skills that are well developed through extensive training. Each instrument has its one tone and is tuned and well cared for by each player. But there is also a conductor who interprets the music score and coordinates the efforts of all players into a polished performance. Only when these elements fit together does the whole orchestra produce an outstanding performance with beautiful harmony, tone and rhythm. No single element should be missing. No single player or instrument should be out of tune. To emphasise this latter idea a diagram showing the main organisational characteristics of conventional and progressive companies is shown in figure [2.5].

The "activity chain" concept is another useful approach for achieving coordination [10]. This concept is based on the importance of the integration

	Conventional Company	Progressive Company
Operational Character	istic	
Setup time	Long	Short
ot size	Large	Sma I I
Inventory	Large	Sma 1-1
loor space	Large	Small
Fransportation	Long	Short
ead time	Long	Short
Defect rate	High	Low
Machine trouble	High	Low
Organisational Charact	teristic	
Structure	Rigid	Flexible
Prientation	Local optimization	Total optimization
Communication	Long chain of command	Open communication
Agreement	Contract-based	Trust-based
Jnion focus	Skill-based	Company-based
KIII base	Narrow	Broad (flexible)
Suppliers	Many	Selected few

Figure 2.5 Characteristics of Manufacturing in a Conventional and a Progressive Company. Taken from Suzaki [9]

between functions and departments. The concept of "activity chain" denotes a continuous chain of activities associated with the dealing of an essential task of an industrial enterprise, such as product development, production flow, and customer orders. An activity chain will cut across functions and departments, as is illustrated in figure [2.6], as opposed to traditional vertical communication. This "activity chain" concept also fosters a Production Management Concept that will give a coherent picture of the way in which production is to be managed. This latest concept includes mutual agreements between parties involved in the manufacturing process, such as sales, engineering design, production engineering, and the various production units.



Figure 2.6 Activity Chain. Taken from Frick [10]

In a broader way, Schonberger in his book "Building a Chain of Customers" [11], emphasises that the principles of world-class performance must now be applied to the companies as a whole, replacing stand-alone departmental strategies. Quoting Schonberger: "*The whole organisation, every single part of it, has to be part of the improvements*".

In a more specific context many other people, and leading them Burbidge [12,13,14], have addressed the importance of coordination in manufacturing organisations, by continually pointing out the benefits from breaking down shop floor activities into product based rather than functional elements. A thought that could be applied to all areas of a company. This way of dividing into departments will help to re-integrate the organisation in a new way where the importance of the coordination factor will decrease, due to the inherent coordination factor attached to this structure's nature. This product-based structure will, in his opinion, be needed if manufacturing companies want to confront the market of the 1990s where customer service and everything it implies are the competitive elements.

A related matter is illustrated in the paper written by C.A. Voss et al. [15], where they clearly identify the nature of the integration and coordination provided by the programme management organisation as vital to the achievement of the fast and effective product development needed to compete in the actual competitive market.

Another study complementing this view was by Voss [16] himself. In his study of success and failure in the implementation of advanced manufacturing technology, he found that organisational integration, coordination, was likely to be associated with successful implementation.

Therefore, the importance of coordination can, undoubtedly, be seen in the way implementation of new technology in manufacturing organisations has been carried out.

2.3.1 TECHNOLOGICAL IMPLEMENTATION

Many specialists think that the implementation of new technology, and change (in a broader context), fail to succeed because the human element is ignored. Market demands for shorter delivery times and improved quality have created a need for better integration of the many activities involve in modern manufacturing. At the same time new technology provides potentially powerful means. Nevertheless, experience indicates that a much broader approach is needed, including many aspects and many parts of the industrial organisation, in order to fully utilise modern technology to meet the competitive challenges.

As it is clearly stated in the article about MRPII "The price of ignorance" [17], the success of the companies implementing new technologies does not lie on the technological tools themselves, but in a philosophy that recognises the interdependence between activities and the importance of integration throughout the organisation. That philosophy, must not only involve new manufacturing techniques, but new techniques of management and of performance measurement.

Using "revolutionary" technologies requires a similarly radical degree of organisational change along a number of dimensions, including the skills profile, the functional and hierarchical structure, the philosophy of management and control, and the underlying culture of the organisation [18]. It is the lack of that parallel organisational change, that has caused the failure of implementations, through a persistent resistance to change as is shown in figure [2.7].

Furthermore, those organisational changes that should have been implemented along with the technological ones might fail or have already failed, (if tried to implement them), not because a lack of structural merit but because either they have not been understood or they have not been adequately coordinated. Communication, coordination and integration are the three key elements in any change, and the more manufacturing organisations understand this fact, the more effective and successful those changes will be [19].

······································	
Pressures to change	Resistances to change
	4
Market environment	Lack of awareness and
	understanding
Technology push is	Lack of skills
catalysing change - firms have to adapt to get best out of their investments	Organisational inertia - adapt the technology to the existing organisation
	"Sedimented traditions"
	Lack of alternative models
	Supplier pressure to shape
	organisation to fit their technology package
	Politics - defending positions, preserving power
	Culture - "the way we do things around here" - made up of individual attitudes, beliefs, group norms, established traditions

Figure 2.7 'Force-field' analysis of technology change. Taken from Levy [18]

For example, A.t. Kearney [20] report in the conclusions of their survey of CIM:

...to be effective as a means of achieving competitive advantage, it (CIM) must be introduced as part of a new strategic thrust in the organisation.

...it is easier to install and make operative a new software application than to introduce a cultural change which accepts shared data, multi-functional decision-making and new organisation structures. Similarly Barrar et al. [21] in their study of MRP conclude:

Success in implementation (of MRPII) therefore, must recognise...changes in ideology, organisation and management process.

Also Buckingham and Bessant [22] :

.. only by moving beyond seeing technology as a substitution device can organisations develop the flexibility necessary to fully benefit from integrated technologies.

The successful implementation of any kind of change in manufacturing companies is not based on the existence of technologically advance islands, totally independent and scattered in a sea of confusion, but on the context of a total business enterprise. A business enterprise where there are not separate elements but links in a chain, a business enterprise where all the different functions and departments are coordinated and integrated towards an overall, multi-disciplinary and ever improving compromise ("Concert building" [23]), in which neither technology nor organisational structure or process has been static, but has instead been in a state of constant evolution.

Innovation must be focused on collaborative efforts within and outside manufacturing organisations, identifying interdependent activities across the value chain that could be used to drive the organisation to a better competitive spectrum, where the customer needs are really fulfilled. Yet another important illustration of the importance of cooperation in the implementation of Total Quality Management (TQM), an example of organisational change, is compactly summarised in one of the vertices of the Joiner Triangle: "All One Team" [24,25]. This coordination, cooperation, is a foundation stone of the Deming philosophy.

Concluding, it can be said that whichever the organisational approach taken (e.g., "activity chain", product based break down,...), manufacturing organisations will have to continue facing their challenges, improvements, and changes, in a departmental framework. The recognition of this fact, implies the acceptance for the need of coordination of all the different areas, and departments involved in the functioning of the company.

2.4 DEALING WITH COORDINATION

2.4.1 INTRODUCTION

Every manufacturing organisation struggles, intentionally or intuitively, to find a way of coordinating its functions. Therefore, in the following sub-sections some stereotype solutions used in different organisations will be reviewed. It should be stressed that no organisation uses just one solution. Actually, in Goldratt's opinion, everyone uses a blend where one solution is dominant but where at least traces of other solutions exist [4].

2.4.2 THE ARMY SOLUTION

Probably the most known and ancient solution is the army one - discipline. Discipline is a way of coordination that attempts to minimise the distortions introduced when interpreting guidelines and detailed instructions. For this sort of coordination to work, it is necessary for the lower ranks to understand exactly what is required by each order that is issued. For example, the meaning of the order "march" is not simply to walk forward, but to hold yourself upright, swing arms, hold the rifle in a specific way, etc..

In this context, the task of the lower management ranks is basically to carry out the current orders according to the respective predetermined process. This idea is summed up by the army expression "*do it by the book*".

This methodology is an attempt to neutralise the lower management levels from making any significant decision on their own, leaving the decisions to the very top of the pyramid. This discipline method tries to avoid the need to rely on local optimums. This model is the one advocated by Frederick Taylor in his "scientific management" philosophy [26].

Although this approach has a positive short term influence in manufacturing organisations (i.e. things are done) due to the need for synchronisation, it has an undesirable long term effect on the organisation (i.e. people are not involved). That lack of involvement will create conflict and unnecessary bureaucracy, that will make the manufacturing organisation unable to cope

with the dynamic environment faced by manufacturing companies.

2.4.3 THE RELIGION SOLUTION

An almost antithesis of the army method is the approach used by religionbased organisations. Organisations based on the religious model do not use a predetermined, strict translation between a guideline and instructions. Rather they have developed a very detailed code of what is right and what is wrong. This code is the connection between a situation and the appropriate guideline rather than between a guideline and a detailed instruction, lower levels of the organisation can decide what to do without waiting for a decision to come from the top. This method tries to give maximum autonomy and decision power to the lowest possible rank in an organisation, it minimises the problems for coordination because every element knows how to perform to achieve the purpose of the whole organisation.

Although this seems to be a good approach, it has some problems when dealing with organisations broken down into departments that have to synchronised mutual efforts in order to carry out their activities.

2.4.4 THE GOAL DICTATES THE SOLUTION

The basic reason for the previous two different approaches is probably based on the means that have to be used in order to achieve each organisation's goal. The goal of a religious based organisation is achieved when each individual behaves separately in a certain way. The goal of an army based organisation, however, can only be accomplished through a synchronised effort of many individuals.

Manufacturing organisations suffer from the army's basic problem even though usually not to the same degree. Based on their departmental divisions, the only way for manufacturing companies to achieve their goals is by the coordination and synchronisation of the different elements inside them. Nevertheless, manufacturing organisations have always been aware of the advantages of the religious based approach and have tried to use this methodology by trying to define rules, **measurements**, that are supposed to define what is "right" and what is "wrong". Manufacturing organisations have continuously tried to find the balance between the army based approach and the religion based one.

The definition of general goals, and of measurements that will be used to assess how the different elements of the organisation influence that goal, will be the way ahead for companies seeking for coordination.

2.4.5 EXAMPLES OF APPLIED COORDINATION

Many Western manufacturing organisations have balanced their approach towards the army method, while Japanese companies have tend more towards the religion method.

Therefore, in the next two sections two limited examples of these two approaches will be shown, looking at the way coordination is respectively understood in Western countries and Japan.

2.4.5.1 WESTERN COUNTRIES' APPROACH

The Western countries' approach towards coordination is very much based on the army discipline and in the use of techniques as an strategy to gain effective coordination. Western countries have been emphasising the technological aspects of coordination as a way for its achievement rather than other existing aspects (e.g. organisational,..). They have tried to achieve coordination by implementing techniques from top to bottom. They have trained people to carry out tasks rather than help them understand what is happening and why is needed to happen. They have tried to implement techniques (e.g. OPT, KANBAN, SPC,..), especially the ones used in Japanese manufacturing, but looking only to the tools provided by them rather than to the general existing

philosophy and inherent overall understanding behind them.

This approach has, however, turned out to be a failure in many cases. There are many written examples that refer to failures when trying this approach in Western companies. A clear example of this fact is the study conducted by A.T.Kearney [20]. This study of manufacturing companies in Great Britain, Japan, the United States and Western Europe found that when it comes to gaining benefits from new technology, Britain is at the bottom of the league table. That study also claims that of the £ 1.9 billion spent on factory automation each year in Britain, some £ 600 million is wasted.

As it is concluded by D.M. Lascelles et al. [27] :

"Managers tend to seek instant solutions; techniques are often seen as an end in themselves, giving little proactive thought to the introduction and use of such techniques within the overall framework of managing improvement". This frustrating experience is not a new one, having been perceived all through the 1980s. Some of them are also refer in an article by Paul Levy et al. [18]. There are some other interesting comments from Andrew Owen et al. [28] and Brenton R. Groves [29].

2.4.5.2 JAPANESE'S APPROACH

Japanese companies have approached coordination from a more general point of view. First, they have reduced their hierarchies, flattening the hierarchy as shown in figure [2.8]. This supposes that the decisions are not taken by single people but by groups, multi-disciplinary teams. In this team system, communication and decision-making occur bottom-up from a web of fellow workers, instead of top-down as in an army hierarchy. In this integrated environment, reward comes from empowering others, not by climbing over them.



Figure 2.8 Flat hierarchy

This former idea is the one that has made Japanese organisations so successful. They see their success as the successful management of the web of highly dependent relationships existing in manufacturing organisations [30]. They emphasise harmony and cooperation among the different elements rather than individual functions and responsibilities. Each member or function is expected to do whatever is believed to be most important for the fulfilment of the goals of the collectivity at any given moment in time [31].

A second element is their view of techniques as a tool, but not as a solution to coordination. Japanese manufacturing organisations base their coordination on organisational as well as technological aspects [32]. They argue that successful coordination is achieved when the technical and human sides of the change process are jointly optimised. It is this general vision of coordination and the techniques used for it that have helped them to successfully implement philosophies like JIT, TQM,...

If there is a "secret" to Japan's success, that one lies in the continuous improvement process generated by a whole system, and not, as some Western counterparts have assumed, in specific and independent parts of that system.

2.4.6 TOWARDS THE "GOAL DICTATED" COORDINATION APPROACH

2.4.6.1 INTRODUCTION

Among the powerful forces influencing the coordination performance of manufacturing organisations, three emerge as outstanding. First is organisational hierarchy, second is specialised work and third is the reward
system [33].

Based on the fact that the improvements in manufacturing organisations must be carried out in a hierarchy and specialised predetermined structure, there is one only way ahead towards successful coordination. This breakthrough implies the definition of a reward system, in other words, the definition of global objectives towards which the local ones should be focused. Inherent to this concept is the need to understand the dynamics and interactions of these objectives throughout the organisation. That process of understanding should involve the people that define the objectives as well as the people that try to achieve them.

2.4.6.2 DEFINE OBJECTIVES

The importance of the definition of objectives for manufacturing organisations is very much supported by people like Burbridge and Goldratt. They think that for coordination to work in a manufacturing organisation, a common goal and a way to measure its achievement has to be found. This means, that there is an implicit need to clarify how and to what extent areas, functions or departments contribute positively to the global success of the organisation. Burbidge et al. [34,35], for example, thoroughly discuss the integration process and its importance within or between management functions in industrial enterprises. One of the conclusions that is derived from one of the

studies [34] about the integration across functions is that : "*The most important need for interfunctional integration is the integration of goals*". This assertion is based on the fact that one function inevitably affects the conditions for the operations in other functions, and therefore stresses the need for focusing on the motivation to perform activities which will be beneficial for the whole organisation.

Also Goldratt [36] presents the concept of the inherent conflict. The inherent conflict states that in many cases "the objectives of local areas are in conflict with the objective of the global organisation", and that the way out for it, is to look for a situation where "the objective of a local area is to contribute positively to the objective of the entire organisation".

He also mentions the functional problems encountered when trying to apply his own "Theory of Constraints" philosophy to one function, stating : "The lesson today is clear. Before any function can go on an ego trip, demonstrating and waving results (and by that digging its own grave) - before any function can start individual improvements, all functions should decide together in a common way."[37]

To summarise:

The most important element for the accomplishment of a successful coordination, integration, is the definition of a common global goal towards which all the different existing elements of the factory ,local areas, should

be directed to.

Another element related to the definition of objectives is the control system being used to measure them. This control system (rules, measurement) will have to be established to be good not only at one level or for one function, it will have to be appropriate for all management levels and for all the various tasks that managers carry out.

Deming, for example, in his study of the theme of cooperation [24,25] becomes aware of the barrier which the merit rating forms to real change and to coordination, as an example of change. The worst case is that of the effect of the performance appraisal constrained by a fixed distribution where it becomes necessary to put somebody else under in order to get a higher rating.

Some other people have also demonstrated, that the importance different functions or departments of a manufacturing company give to a certain performance measure (cost/price structure), a rating form in itself, varies greatly as it can clearly be seen in figure [2.9].



Figure 2.9 Departmental misunderstanding. Taken from Vollman [38]

There are also other references of traditional methods of accounting and finance, being seen nowadays as archaic [38,39]. The writers state on those articles that the existing methods can no longer be used in an industrial environment where so many things have and are changing as a necessity, mainly because those methods are in themselves one of the reasons for making that change fail.

Therefore, there is no point in trying to achieve common goals in an organisation that rewards its functions based on local optimums. Functional performance measures can no longer be seen as local measures in isolation, because they are not and they have never been. For example, customer service and delivery performance measures are frequently used. A manager can perform extremely well on these measures if he is not also responsible for inventory investment. Purchasing agents can score well on the cost dimension if not held accountable for inventory carrying costs, vendor lead times, or quality. Marketing personnel can perform well on total sales if not responsible for inventory investment or shop disruptions caused by continual expediting and overtime.

Throughout these previous ideas the elimination of local performance measures are not being advocated. But what is encouraged, however, is the analysis of the impact of the local measures on the more important global measures of long-term profit, market share, etc. Also it is seen as a need to study the impact that these measures have in other functional areas and production departments. So, the point is clear, the system of measurement and reward has to change. It can no longer be based in local measures, individual results, confronted against each other and established in isolation, but in a global system of performance measurement were it can be established how functions, and different areas, influence the whole performance, objective, of the manufacturing organisation.

Attached to this idea is the need for understanding the whole of the organisation, the objectives (global,local) and the dynamics and interactions of manufacturing organisations.

2.5 THE ROLE OF EDUCATION

2.5.1 INTRODUCTION

The importance of the educational process in the success of manufacturing organisations, was addressed throughout the 1980s as it is explained by Marilyn M. Helmes [40], who also concludes that behaviourial and human relations concerns will continue being of primary importance for the successful implementation of change and that commitment to substantial training and education will be needed.

2.5.2 EDUCATING FOR CHANGE

These introductory comments are not isolated, there are many people who emphasise the huge importance of education and training as a way to introduce effective change at all levels and areas of the manufacturing organisation. When they contemplate the introduction of change, they also see the need to understand and learn, the nature of the change and, in particular, its implications on the corporate culture and values.

The reason for this consensus on the role of education is based on the fact that change can not be achieved without people. And people need to learn and understand, they have to believe that they are in fact the initiators of the change, and this can only be secured by **communication** and **training**. In fact one leads to the other. In other words change is synonymous with learning. A company in which planned proactive change is the norm is, by definition, a company in which training and learning is inherent and continuous. As it is stated by Alan Marsden [41] the first step then is to communicate. Make everyone in the organisation aware of the impact of the change process upon culture. People should be educated to understand culture. They need to understand that it is a system of shared values (what is important) and beliefs (how things work) that interact with the company's people, organisation and controls. Other similar comments could be found in the article "Education: The key to JIT success" [42]. But this process of education, encounters an important barrier in the organisation itself, in the way it is managed and structured, as it has been stated throughout the previous sections. The so emphasised problem of functional competition and conflict avoids the completion of a cooperative management.

As it is explained in the article "Change is as good as rest... provided you train for it" [43] and based on a study conducted by Ashridge Management College, organisations are attempting to reverse or at least mitigate some of the more stultifying effects of Taylor's legacy. Manufacturing organisations are becoming flatter, more market-driven, more customer oriented, increasingly decentralised and generally more fluid.

This trend and the perception of its need, has major implications on how manufacturing organisations are managed. The ability to manage *horizontal* issues (such as quality and service) across the company, and therefore across departmental boundaries, is becoming more and more important compared to the traditional vertical management.

So, how should education be focused in this coming pro-cooperative environment? How could education emphasise the need for cross-functional involvement inside an ever changing and conflicting organisation?

2.6 CONCLUSIONS

Once having seen the importance of coordination and how organisational boundaries become less of a concern as codestiny, a word for common goal, is developed among people and functions in maintaining, looking for, a competitive position and achieving the common goals of the organisation. If people focus only on internal disputes, the organisation will most likely become extinct. And the way to the definition and understanding of that codestiny, as well as the problems associated in its search is **AWARENESS**.

As more people understand the vision of the future and the direction of the company, as well as how each individual fits into the picture, the whole system will be effectively integrated, coordinated, and become capable of responding to necessary changes. Therefore, the first task of manufacturing organisations is to fully understand their problems [44].

Organisations have to be framed in a process of learning, the "Learning Organisation" has to be created, an atmosphere of recognition to the role of education and training in the awareness process. That "Learning Organisation" has to be established through a cross-functional learning process where all will comprehend the dynamics of the process they are immersed in.

If that is the case, manufacturing organisations will have to educate their people in a cross functional process which also will have to take into account the dynamics of the environment companies are working in. In this context,

the role of education in the development of change can not be over-stated. Indeed, the success of the different changing processes manufacturing organisations are involved in, depends upon the ownership, commitment and awareness of the people affected throughout all levels and functions of the organisation, being every of these prerequisites achievable through an effective education and training program.

Therefore, education can no longer be seen as a static and independent process, where people only understand how they work, and how others work on a stand-alone basis. What is needed is a new and more **integrative** approach, where people understand not only how they work and why, but also how they affect other areas of their manufacturing organisation, in other words how their policies and performances fit inside the policies and performances of the whole organisation. This is certainly a **dynamic** processs as decisions are being made throughout the organisation in response to a continually demanding environment. Thus, in the next chapter it will be explained the different existing educational methodologies and in particular it will be emphasised the role of computer based educational programs (e.g. computer based gaming-simulations) as tools for a more **dynamic**, **interfunctional**, and **goal integrated** approach to the learning process.

3 METHODOLOGIES FOR EDUCATION AND TRAINING

3.1 INTRODUCTION

As it has been concluded in the previous chapter, education can no longer be seen as a static and independent process but as an integrative and dynamic one.

Now that the main objectives for the educational process have been defined, there is a need to define the tools which will help carry out that educational process inside its philosophical framework.

The importance of the tools chosen for educational process is clearly stated by Robert I. Millard [45]. He concluded in his research about MRP implementation and its failures and successes related to the educational aspects that: "Success did seem to be related to the amount of outside assistance and to the breadth of educational effort applied during the system implementation. While these results suggest that education and training are important ingredients in the implementation process, they suggest that the type and quality of materials used may be more important than the time spent with them."

This assertion can be numerically seen in table [3.1], which has been taken from the previously mentioned study.

Correlat with MRF	ion of Implement Success	ation Variables	
	Outside Assistance	Education Days per Employee	Breadth of Educ. Matls,
Increase in turnover Decrease in lead time	+ 0.50	+ 0.04	+ 0.33
Perceived project success	+ 0.25	+ 0.12	+ 0.13
MRP usage class	+ 0.12	+ 0.11	+ 0.24

Figure 3.1 Correlation of Implementation Variables with MRP success. Taken from Millard [45]

Nowadays, there are many well known training companies that are using different techniques and tools for educational purposes. These educational aids range from textbooks, seminars and short courses to videos; from manual games and simulations to computer based games and gaming-simulations.

Within this diversity of possible choices, this study will focus on the analysis of the computer based educational aids, and will try to establish whether these computer based aids can address the problems of **dynamics** and **goal integration** (cross-functional coordination) in the educational process.

3.2 EDUCATION AND TRAINING THROUGH COMPUTERS

Traditionally the process of education has involved lectures, discussions, and "programmed learning" approaches requiring the student to sit and read the text, answering questions at the end of each section as a way of feedback and interchange. More recently there has been a considerable number of educational videos produced on manufacturing subjects by such companies as the Oliver Wight organisation. These are useful ways to provide information but are rather limited in the way they help people understand a dynamic and real situation of the sort that exists within a manufacturing organisation. Additionally any activity that requires the student to just listen and not participate tends to get very boring. Nowadays it is the computer and its flexible structure that comes out to face the educational challenge.

The advent of the personal computer and its never ending technological improvement (e.g. speed, memory, flexibility, increase in the number of software tools,..) [46] has encouraged a explosion in the amount of educational software available. There are many written examples backing up this and explaining the use of computers for teaching purposes [47,48,49]. In one of these, for example, fourteen packages in the field of P.O.M. (Production and Operations Management) are reviewed.

The reason for the appearance and increase in the use of this sort of educational tool must be seen in the opportunity that computers provide towards the generation of an environment where theory and practice can be demonstrated in the context of real and immediate operating situations [50]. This possibility of having real experience in dealing with a "real" environment, undoubtedly, results in more effective learning. It helps to create discussion about a real question and reinforce in that way the learning process.

The topic of business games and simulation is therefore discussed in the following section. These, seem to offer an understanding of the concepts of interaction and dynamics which have already been identified as being so important in manufacturing.

3.3 BUSINESS GAMING-SIMULATIONS, WHAT ARE THEY ?

3.3.1 INTRODUCTION

Since 1960, Management Games also called Business Games have become a more widely known - but rather less widely used - technique of management education and training. They are not yet used, of course, with the same frequency and intensity as case studies but games have and will undoubtedly become another tool of the management teacher, to be selected and used whenever appropriate. Historically, management games are directly traceable to war games. These developed with chess, continuing in 1798 with the German "war" game called "*Neues Kriegspiel*" that was a "TEWST" - Tactical Exercise Without Troops and many more broadly used during the 19th century. It was not until many years later that as a result of a visit by a member of the A.M.A. (American Management Association) to the U.S. Naval Academy, it was decided by the A.M.A. that it might be worthwhile applying the techniques of war games to

management education.

Although the existence of war games was one of the stimuli to the development of management games, there were also two other factors that assisted in the development of games. These were the Development of the Theory of Games and The Advent of Digital Computers [51].

3.3.1.1 THE DEVELOPMENT OF THE THEORY OF GAMES

The publication of "The Theory of Games and Economic Behaviour" [52] by Von Neumann and Morgenstern in 1946 led economists, operational research workers and management teachers to consider, in new ways, the analysis of action in competitive situations. Although the Theory of Games offered little in the way of specific applicable analytic tools, it did illuminate the analogy between games and industrial situations. It also made people familiar with game concepts and greatly stimulated interest in problems connected with games.

3.3.1.2 THE ADVENT OF DIGITAL COMPUTERS

The advent of computers and the continuous technological advances they have been experiencing, contributed and are still contributing to the development of business games. The reasons for this were and still, partially, are based on three advantageous main elements:

1. The use of computers in a management game adds considerably to the "drama" of the game.

2. Computer manufacturers seized upon games as a good sales opportunity for their product in so far as they wished to sell computers for business, rather than scientific, purposes.

3. The existence of computers, with their speed and flexibility provided designers of games with the opportunity to develop complex mathematical models and interactions to underlie games whilst still keeping the administration of the game relatively simple.

All these factors, occurring at about the same time, accelerated interest in the transference of the gaming technique from a purely military training to that of management education.

3.3.2 BUSINESS GAMES

Management games as such started in the late 1950s, with the development of computerised management games dating from not long after this period [53].

Historically speaking, management games originate from three major design foundations: (1) role-play, (2) gaming, and (3) computer simulation.

1.Role-Play: In role-play, the player of the game assumes a prescribed role in a particular situation. He is given the opportunity to feel what is of interest in that position. It is hoped that players gain a greater understanding of the roles and relationships as well as a better awareness of their own activities. Moreno's [54] work with "psychodrama", a form of therapy for mental illness, is most often cited as the beginning of use of role-play as a vehicle for extending research into human behaviour in varied learning environments.

2.Gaming: Traditionally, games consist of interactions among group of players (decision makers) placed in a prescribed setting and constrained by a set of rules and procedures. The behaviour and the interaction can possibly involve competition, cooperation, conflict or even collaboration.

3.Computer Simulations: Computer simulations has roots in mathematics, probability theory, and other associated mathematical techniques. In the early days the purpose of computers simulations was to find answers quickly and accurately by following a predicted algorithm. In the corporate area, computers have been used extensively for analytical simulations to support the evaluation of implementation alternatives for new facilities and products prior to commitment to important decisions.

3.3.3 GAMING-SIMULATIONS

These three previously explained streams of development (role-play, gaming, and computer simulations) serve as a general background for **gamingsimulations**, and **computer based gaming-simulations**, words that from now on will be used indistinctively. The distinction between these three types of activities is not clear-cut; there is a great deal of overlap. Nurtured in this context **gaming-simulation** appears as a new intermediate element. **Gamingsimulation**, is an hybrid form involving the performance of game activities in simulated contexts [55].

In these **gaming-simulations**, the environment and activities of participants have the characteristics of games: players have roles to play, goals they seek to achieve, activities to perform, constraints on what can be done, and payoffs (positive and negative) as a result of their actions and the actions of other elements in the system (including chance). But also, in a **gaming-simulation**, the game roles, goals, activities, constraints and consequences, and the linkages between them are patterned from real life, they are a simulation of the real-world systems. Gaming-simulations differ from role-play management games in several important ways. Role-playing is an instructional technique, an element of gaming-simulations, but the latter also include other components. In most roleplaying exercises the participant is assigned a role and given a general outline of a situation; from there the action is freewheeling. In gaming-simulations,

on the other hand, roles are defined in interactive systems. In other words, the emphasis is on the role as it interacts with other roles; the model creates the basis for the dynamic interaction, including also all the constraints, rewards and punishments that characterise games.

Gaming-simulations thus differ from role-playing exercises in the degree of structure and formalisation they entail and in their emphasis on interaction processes rather than on the playing of individuals roles. Furthermore, in many instances of classroom role-playing, several students participate while the remainder of the class watches, or the class is broken into small groups that engage in parallel role-plays. On the other hand, when gaming-simulations are used, all students are participants; none are passive observers.

3.3.4 BUSINESS GAMES AND GAMING-SIMULATIONS AS EDUCATIONAL TOOLS

The standard mode of teaching and training has historically been the lecture or the lecture and discussion method. But these methods have some limitations, weaknesses that Cathy Stein Greenblat [55] think could be overcome with the use of gaming-simulations.

She thinks that the most critical problem traditional teaching methods have to face, is that learners are passive recipients of information. They rarely have to put their ideas forward, and their attention frequently wanders to other goals or goals of greater interest.

Secondly, in her opinion, material is presented in a sequential way. This linear approach makes it difficult for the learner to grasp the nature of the whole. This is particularly so in instances in which one is trying to teach the appreciation of the simultaneity of events and actions.

A third shortcoming of the lecturing method is that it is a difficult task to present systems characteristics clearly. The multiple connections and consequences of a given element or action are difficult to convey verbally. While it is often impossible to understand a social system by simply immersing oneself in it, because the complexity is too confusing, the lecturing process all too often suffers from oversimplification. In opinion of Greenblat, Gaming-simulations can address all these shortcomings in various ways. First, in a gaming-simulation, in contrast to a lecture situation, everybody is an active learner. They must make decisions, pay the consequences, articulate positions, and make the system work. After play, post-game focuses on what happened and why, how that relates to the counterpart, and the limitations of the gaming-simulation. Thus the learning experience builds upon the philosophy of the old Chinese proverb:

I hear and I forget I see and I remember I do and I understand

There are also other positive aspects of gaming-simulation as a learning tool. The active involvement of gaming-simulations in the learning process implies a experimental learning that permits discovery, experimentation, and reflection.

Second, rather than being lectured to and then engaging in discussion, students experience the topic as a whole, since many components of the system are presented simultaneously.

Third, and finally, gaming-simulations are particularly useful for conveying systems characteristics. It has long been recognised that systems are more clearly described using graphical models, such as diagrams or pictures, or using physical representations, than through using verbal models. In Loveluck's opinion [51], gaming-simulations force participants to realise the nature and importance of effective organisation and intelligent cooperation, in relation to a "business-like" situation and problem. He also argues that participants in gaming-simulations must discover for themselves where the problems in this "business-like" situation lie, giving the player a superior understanding of the situation in comparison to merely being told what these problems are. Gaming-simulations also provide a focal point for thought and discussion and establishes a common basis for communication inside a feedback process. Loveluck also explains that although sometimes gaming-simulations could be seen as unrealistic, this fact is a strength rather than a weakness. The specific assumptions upon which the game is based, is an important starting point for consideration of the way in which the "real world" managerial problems evolve and impinge upon one another.

Another contribution to this matter is the one from C. Brand et al. [56] who precisely defined which is, in their opinion, the role of games which simulate real-life organisations. It is their opinion that games applied to organisations will, among other advantages, "*enable participants to understand the wider aspects of their organisation*".

With all these explanations in mind, it is concluded that while gamingsimulations will not solve all educational problems, they offer, exciting opportunities for teaching and learning about many social systems and social problems. They are an outstanding educational element when dealing with the understanding of dynamic social systems (e.g. manufacturing organisations) where the interaction, dynamics and overall perspective of the system, rather than a local one, are of a great importance. They are extremely useful when the response of the players to certain problems and dynamic situations in the context of continuous interactions, is a part of the environment to be studied. In respect to manufacturing organisations, these are considered to be systems that interface with others in complex and uncertain ways, and also tend to become more complex as time goes on. This idea, as it is already been discussed, requires ever more sophisticated ways of managing both their internal workings and their interactions with the environment. In this complex and dynamic environment is where gaming-simulations can provide companies with an effective educational tool. They provide a relatively safe environment for the learning of a whole range of management concepts, practices and skills in the context of a world where many themes are interdependent.

Based on the fact that gaming-simulations can have a great role in the educational process of understanding and awareness of the different existing functions and interactions inside a manufacturing company, there have been and there are many different approaches to their development. The next section will look at them, seeking for a suitable approach to the already stated ideas about the role of education in a manufacturing organisation.

3.4 PAST AND FUTURE TRENDS IN BUSINESS GAMING-SIMULATIONS

3.4.1 INTRODUCTION

The use of computerised gaming-simulations for business and management education as a tool that encourages the development of an individual's wider understanding of the integrated nature of business organisations, appears to be well established. With this notion in mind, it will then be discussed the way designers have tackled the development of these computerised gaming-simulations. Discussing, as well, the suggestions done by Tom Burgess [57] about the design of gaming-simulations.

He maintains, that the design of most, if not all, of the existing computerised gaming-simulations have hardly changed from the original implementations. It would seem that very few take advantage of the capabilities which are evident to other types of modern software (e.g. graphic user interfaces, context sensitive help, networking and so on).

In the next two subsections, the traditional and modern, future, design approaches to the development of computerised gaming-simulations will be compared, trying to establish which should, in the context of management education, be the most appropriate approach for the current manufacturing environment.

3.4.2 TRADITIONAL MODELS

The traditional computer based gaming-simulation models were and are basically based on a single factor: the engine that drives them. In other words, the fact that they are based on mathematical models, models that try to imitate a business or part of it, and its relationship within an environment. The role of the computer in this sort of gaming-simulation is mainly a computational one. In this context computers are seen as "number crunching" calculators that will avoid the burden task of doing all the calculations involved manually. Decisions are fed into the machine which then computes the results for later distribution. In essence, the computer remains a calculating device used because it is fast or simply because it is there, and can handle more complex equations. Examples of these kind of games are the STORM [58], COMPUTER MODELS for OPERATIONS MANAGEMENT [59], TEAMSKILL [60], TOPAZ [61], EXECUTIVE [62], BISSIM [63], all of which have been thoroughly analysed as part of the research project.

Their "improvement" has also gone in that direction, trying to enrich their performance by increasing their numerical complexity (i.e. increasing the number of parameters in the mathematical model used, expanding their relationships, and making these models more robust). Quoting Ahituv and Newmann [64] :

"The model attempts to duplicate the real world under varying states of nature for a given set of parameters."

This computational approach tempts designers to construct ever more complex models which can become dangerous. The reason for that comes from the fact that they are not sufficiently complex to become a valid econometric model of a business but, on the other hand, they are too sophisticated for effective education. The added complexity does not necessarily add clarity and comprehension to the learning process.

These models assume that the real-life equivalent is able to be modelled, moreover they presume that an optimal mathematical solution (input) is available, desirable, and a reality, and that individuals and organisations work in a rational way.

But this whole idea lacks of an essential element when dealing with gamingsimulations as an educational tool: **people**. In this highly technical approach the social context of the individual is ignored; remaining unexplored aspects of group membership and social interaction (e.g. potential conflict of functional and organisational goals). In these more traditional group of games the tasks involve, consist of simulated business decision-making of various degrees of complexity in which the main focus of interest lies in the business performance of the simulated company rather than in the organisation of the group and its **interactions**.

This computational approach does not properly address "dynamism". Decisions, in this kind of games, are made regularly and simultaneously and in most of the cases at the start, removing from that sort of games an essential variable in managerial decision-making: **timing**.

Concluding, this traditional approach lacks two most important elements when trying to address the educational process of manufacturing organisations: internal interactions and dynamism. Elements that are extremely important in today's changing and cross-functional manufacturing environment.

3.4.3 MODERN MODELS

The modern computer based gaming-simulation models were and still are being developed as an answer to the weaknesses associated with the approach used by traditional models.

This approach has involved the creation of more realistic gaming-simulations rather than creating more equational complex ones. These gaming-simulations are not exercises in "beating the machine". Their strategy is to increase the realism of such games by replacing *hard* complexity (i.e. equations) with *soft* complexity. In opinion of Robert Johnston [65], *soft* complexity is obtained by providing a situation where goals may be unclear, ambiguous and not always quantifiable, where relations are not always deterministic, where all the possible decisions choices are neither known nor limited, where the information required to choose between them may be neither complete nor available and where the consequences of the choices are not set, clear or totally consistent. Furthermore he states that to do that, the computer model should be replaced by a human process relying on a broad and not totally rational or complete understanding of the simulation situation, where choices, information, and decisions are based on human

interactions and relationships that are at the heart of every business process.

This same idea of the *soft* complexity is also suggested in a study conducted by Walter E. Rohn [66]. In his study he states that there is an increasing customer demand for business games on behavioral and management-attitude problems. He also thinks that what is needed are appropriate models for the realistic simulation of all human factors influencing the effectiveness of an organisation and the quality of managerial decision making. He states that every step forward in this field allows more reality in simulating real life in gaming-simulations. Clear examples of this kind of games are the one developed by the University of Bradford called BUSGAM [67], and MOSES (Manufacturing Organization Simulation and Evaluation System) [68].

But for this to be achievable, there is an important need for advanced computer technology. As it is concluded by Nigel Bryant et al. [69] in his article "Changing technology and management games":

"Our final argument is that technology influences the type of games we have, and the types of games influence the type of management games we have. Furthermore, because there is a time lag between the development of new technology and the game developers taking advantage of this new technology, then we should be able to look at the present day technology and predict the way in which management games are going to develop." The comparison between the traditional and modern approaches when designing gaming-simulations is summarised by Coote [94] in table [3.2].

SUMMARY OF MAIN DIFFERENCES BETWEEN COMPUTER SIMULATIONS AND COMPUTER ASSISTED SIMULATIONS			
Criteria	Computer Simulations	Computer Assisted Simulations	
role of computer	central: to run simulation on; for precision	peripheral; as springboard to social activity; to enhance simulation procedures	
main interactions	between computer and participants; Participants face screen and respond to computer prompts	between participants who face and respond to each other	
aim of simulation	"winning" against computer	exploring social situations	
decision process	essentially rational (manipulation of mathematical variables)	socio-political or power bargaining (negotiation, coalition forming, bargaining)	
participant roles	puzzle solvers and keyboard operators	social actors in a variety of roles	

Figure 3.2 Summary of differences between computer simulations and computer assisted simulations. Taken from Coote [94]

Four of the most successful business games in Great Britain are evaluated in Appendix B. These were presented in a workshop held the 6th of February 1991 at the Management Centre, University of Bradford. The evaluation is a personal opinion about the four games, where the existence of the two design approaches formerly mentioned can also be seen.

3.5 CONCLUSION

Computers add an important element to the educational process, that is the possibility to experience "real-life" situations in a dynamic environment. Moreover, that main advantage, in respect to more traditional methods of teaching, could be used for the development of successful computer gaming-simulations that could be applied to manufacturing organisations.

But what kind of gaming-simulations? The main approaches have been shown and based on experts' opinions the relative advantages have been discussed.

In our opinion and having looked to the two approaches, the way ahead should be a compromise between the two, where computer power should be used to address specific elements of manufacturing organisations but within an integrated, interactive and dynamic framework, similar to that manufacturing organisations have to face. Those elements will certainly improve the rate of realism attached to gaming-simulations without making them more complex and rigid.

The learning opportunities inherent to this approach are undoubtedly greater. This means not only that participants are able to experiment with technical variables (product levels, marketing strategy, and so on), but also that they are able to experience a fundamental aspect of organisational life, that of bargaining and negotiation in the decision-making process. They are faced with the inevitably interdepartmental rivalry so common in organisations, and with the conflict between this rivalry and the need to cooperate on a company basis.

Inevitably, that approach to the design of games is related to the existence and effective use of the technical computer tools available. Loveluck [70] and Bryant et al. [69], for example, refer to that relation stating that business gaming-simulations will be more interactive and dynamic, focusing on the "organisational behaviour" of participants through the use of elements such as interrupt functions, graphical displays, flexible designing, and networking.

This research is directed to look at the feasibility of such games. Thus, the possibility of creating such games, based on the actual existing hardware and software, will be evaluated.

Therefore, the current work will try to develop a computer gaming-simulation based on those formerly mentioned innovative elements. A gaming-simulation will be used to understand the dynamic and interfunctional aspects of manufacturing organisations. In particular, to make people aware of the implications of their behaviour on others as well as on the whole organisation, as an essential step towards the understanding of the importance of coordination.

For this reason, in the next chapter it will be shown what has been our approach in designing Nottingham Polytechnic's gaming-simulation and which have been the problems encountered on its development.

4 NOTTINGHAM'S POLYTECHNIC GAMING-SIMULATION

4.1 INTRODUCTION

As it has been stated in the previous section, computers offer an exceptional number of educational advantages compared with other more traditional methods of teaching. It was also discussed how those educational methodologies could successfully be applied to manufacturing organisations as computer models.

It has also been suggested that for those computer models to be successful, when applied to manufacturing organisations, they should address specific elements of manufacturing companies inside an integrated, interactive and dynamic framework. It is in this context where this chapter of the thesis will try to address the process of development of Nottingham Polytechnic's gaming-simulation, looking at the different steps of the development as well as the problems encountered in the process.

It is also the aim of this chapter to look at the feasibility of these kind of gaming-simulations in the actual technological (software and hardware) context.

4.2 TOTAL ENTERPRISE BUSINESS GAMES

Before explaining the design process of Nottingham Polytechnic's gamingsimulation, the concepts of total enterprise business game and the production management game will firstly be introduced.

These two terms although closely related diverge in the degree in which they concentrate on the production area of manufacturing organisations.

Total enterprise business game is a term used to refer to games that include all the main functions of a business enterprise as its decision inputs: marketing, production, and finance. Due to their flexibility, total enterprise business games can fulfil a great variety of purposes. They are mostly used as integrative teaching techniques to bring together all the functions of business and allow participants to view them as a whole. In management development and executive programs, games often serve to break down the conceptual walls of narrow specialised thinking and cause businessmen to better appreciate the difficulties of their counterparts employed in other business functions [71,72]. Nevertheless, most of the existing total enterprise business games are more marketing, finance, tactical, and strategy based rather than production based. Most of them can also be categorised in the **traditional model** group of computer games ,(refer to previous chapter), taking little account of the dynamics and real interactions of the situation they are resembling; although there is a view that these trends are changing [72].

At the other end of the spectrum is the **production management game** concept. Although there are many games in which the term "production" is simply used to describe a game of a total enterprise nature, **production management games** refer to those that concentrate on the organisation and control of the production function or, more specifically, they are concerned with the decision-making necessary to ensure that goods are made in accordance with the requisite quality standards, in the requisite quantities, at the requisite times, and at a minimum cost.

Based on this idea, most production management games tend to concentrate on the specific production function of an organisation in isolation, rather than integrating the production function with the financial, marketing and other elements of the organisation [73]. Although these games offer a high degree of manufacturing detail, they tend to produce suboptimum decision-making when examining the organisation as a whole. The advantages that a half way house between the above two concepts could offer are obvious. A hybrid concept of a production function could be used to link the business-making policy of the organisation as a whole. That hybrid concept will have to simplify the production subsystem in order to show the interactions with the other subsystems of the organisation (particularly marketing, personnel, and product research and development), and also emphasise the importance of the production function as the key element of a manufacturing enterprise.

Nottingham Polytechnic's gaming-simulation is designed to be such a hybrid game.

4.3 THE DESIGN PROCESS

4.3.1 INTRODUCTION

The designing process of gaming-simulations is generally defined as being a creative one and it is therefore impossible to find precise rules about how they should be designed. However, notes and guidelines about the design and construction of gaming-simulations can be found.

Many experts associated with business games, gaming-simulations, and games in general have contributed to the game design process by publicly stating their ideas about the guidelines to be used.

Loveluck [51], for example, considers the properties of a game from the constructor, designer, point of view as follows:

- Situation being simulated.
- Model underlying the game
- Information made available

- Stress of the situation
- Decision-making required
- Reports required
- Special Assignments
- Composition of teams

Another similar approach is the one recommended by G.I. Gibbs [74] where he defines fourteen design stages:

- Aims
- Context
- Target population
- Models
- Scenario
- Information Flow
- Cast List
- Role elaboration
- Player objectives
- Statement of rules
- Description of constraints
- Materials and equipment
- Evaluation
- Reprography

The model that represents the design process defined by G.I. Gibbs is shown in figure [4.1].



Figure 4.1 Game's design process. Taken from Gibbs [74]

Yet another analogous contribution is the one by C. Brand and T. Walker, who in their article called: "The simulation of 'real-life' organisations within a game" [56] define a logical approach to games design. In their opinion a successful game simulating a real-life organisation must adopt a logical problem-solving approach throughout the design process. This means that the game's development can be subdivided into the following distinct stages:
- Developing the game brief
- Detailed research programme
- Production of a model of the organisation
- Formulation of the outline game
- Client review
- Detailed game design
- Play-Testing
- Production of the final game

A more recent and also more comprehensive approach to the design process is the one described by C.S. Greenblat. In her book called "Designing Games and Simulations - An illustrated handbook" [55] she states that the process of design of a gaming-simulation can be thought of as consisting of five stages.

- Stage I: Setting objectives and parameters
- Stage II: Model development
- Stage III: Decisions about representation
- Stage IV: Construction and modification of the gaming-simulation
- Stage V: Preparation for use by others

This methodology appears to include all of the steps proposed by the other authors in a coherent framework and consequently has been used in this development. Another important element attached to this design process guideline is the fact that, although the defined sketch presents the process as unidirectional (the designer moves from Stage I to Stage II, and sequentially until Stage V), the actual process in itself is much more about moving back and forth among these different stages. In general, however, the "push" is in the forward direction - that is, the process is basically sequential, though there is considerable iteration.

With all this in mind, the coming sections will be concerned with the way Nottingham Polytechnic's gaming-simulation has been designed and constructed based on Greenblat's design process.

4.3.2 SETTING OBJECTIVES AND PARAMETERS

At the outset of the design process it is crucial that the gaming-simulation designer considers what he or she is trying to do and to what ends. This means delimiting as clearly as possible the subject matter, purpose, intended operators and participants, and the context of use for the proposed gamingsimulation.

In Greenblat's [55] opinion, this stage is too often skipped or undertaken casually, and the would-be designer's initial enthusiasm is translated into only a vague formulation, with no thought given to what the gaming-simulation is supposed to convey, to whom, and under what conditions of play. The result of this neglect is often failure or a product that is totally inappropriate for the intended audience.

Therefore, it is critical to make these decisions before anything further is done, and the good designer will spend considerable time formulating such a statement with the firm conviction it will guide him or her later in the process. Wise decisions in the coming stages will only be possible by reference to criteria that are implicit or explicit to the specifications in this stage.

One also should be aware that the **gaming-simulation's** design process has to manage two factors simultaneously, game and simulation. In this context it has to be remembered that the game roles, goals, activities, constraints and consequences, and the linkages between them are patterned from real life, in other words, they simulate these elements of the real-world systems.

4.3.2.1 THE SUBJECT MATTER

The first and more obvious consideration is that of the **subject matter**. At this early point although the designer should not be concerned with a detailed statement, he should, however, restrict the extent of the educational topic he is going to present to the players.

In Nottingham Polytechnic's game the subject matter is very much related to what has been stated in the previous two chapters of the thesis. Consequently the subject of the matter will have very much to do with manufacturing organisations, and the theme of coordination among them. That idea can be summarised in the next paragraph:

"The gaming-simulation to be created is to simulate the way manufacturing organisations work, looking particularly at the way the different functions involved in the company process interact with each other in a dynamic, customer driven environment. Particular attention will be paid to the conflicts that arise from those interactions and the manner in which functional actions are influenced by local objectives instead of global ones."

4.3.2.2 PURPOSE

The question of purpose is closely connected with that of subject matter. Indeed, in some cases, the question of purpose may be more important than that of subject matter and can involve the definition of more than one purpose. If that is the case those purposes will have to be evaluated, and the most important ones will be considered. The question of purpose consists of broad statements of educational intent. These statements express goals which outline the concepts, structures, processes, facts and attitudes, which are intended to give the participants experience. The aims that Nottingham Polytechnic's gaming-simulation pursues are related to the importance of coordination and interfunctional harmony within manufacturing companies. This idea is fully summarised in the next paragraph:

"The gaming-simulation to be created is designed to serve as a catalyst to present issues and problems concerning the existing interactions between different functions of a manufacturing company within a dynamic, customer driven environment. In this context, an important aim of the gaming-simulation is to promote the understanding and necessary awareness of COORDINATION AND ITS IMPORTANCE IN MANUFACTURING ORGANISATIONS."

4.3.2.3 PLAYERS' CHARACTERISTICS

Another matter to be defined at this stage of the design process is the target population of the game. It has to be stated to whom the gaming-simulation is aimed, and which are the characteristics of those playing it. This topic is so important because games have to be designed to meet the needs of those being educated, and through this ensure their motivation and understanding of the process being undertaken.

As an example, the characteristics of a game about some aspects of business management should be different if it is to be played by audiences of community members to show them the pressures on business leaders than it would be if it is to be played by corporate executives to sensitise them to the problems of coordination of departments in the firm. It is important then, to state the target group of players and think carefully about their characteristics. These have implications for the degree of complexity it can be presented, the degree of formality that must be included, and the need for background information.

Nottingham Polytechnic's gaming-simulation is intended to two major potential groups of players:

(a) Manufacturing engineering students who desire to understand how the different functions of the manufacturing process fit together and what are the conflicts associated with their coordination in a dynamic customer driven environment. (b) People in middle management within different departmental functions of a company, that although having experience of the elements and decision-making procedures used in their respective functions, have no understanding of how other functions work and the reasons why. In other words, people wanting to understand how their functional behaviour affects other departments and consequently the company as a whole.

4.3.2.4 OPERATOR CHARACTERISTICS

A question directly related to the characteristics of the players is that of the characteristics of the likely operators. It must be decided who these people are and what aims, interests, and abilities they have.

In Nottingham Polytechnic's case, although the game could be used by a teacher, group leader, or conference organiser; it is basically intended to be used by lecturers at the manufacturing department of the mentioned polytechnic.

4.3.2.5 CONTEXT OF USE

Specification of the context of use will give the designer the ability to determine how many players must be accommodated in play sessions. If the gaming-simulation is composed of a certain number of players, the gamingsimulation must be designed with that number of roles, or in a way players could be divided into smaller groups for simultaneous play. The answer to this point will also determine the time length of the game. This time factor is very important for future decisions, and is closely related to the complexity of the model. As indicated by Duane Dillman [75] :

"The length of a simulation exercise should probably be governed by the amount of time necessary for the participants to uncover the governing relationships of the exercise and then develop the appropriate strategies and ideas and observe the results of their actions."

In other words, the defined time frame should be such that will allow the players to evaluate their approach to the problem being faced by a provided feedback. This common sense matter is considered by experts [71,72,73] as one of the major weaknesses of total enterprise games. Instead, these have overemphasised the decision input and model complexity factors, making the time span for game play too short to complete the learning cycle.

In Nottingham Polytechnic's model, the time frame thought to be adequate to start with, is at least two hours. That time frame could be changed based on the experience gain every time a game is played. Those two hours could be break down into some runs of above thirty minutes each, separated one from another by a analytical discussion about the first run. During that discussion the subject matter should be addressed.

4.3.2.6 RESOURCES

Another element of this stage about objectives and parameters, has to do with the resources available for the development of the gaming-simulation and the ones needed for its future use.

70

In the game covered in this thesis, the resources needed and available are very much related to what is needed for researching about the feasibility of multiuser, real-time gaming-simulations of manufacturing organisations. That is, the existence of a network of PCs (NOVELL [96]) and software capable of working in that environment, issues that will be discussed in the stages concerned with the representation, construction and modification of the gaming-simulation, and software design.

Coming back to the five main stages of the design process, and once all the steps regarding the setting of objectives and parameters have been clarified, it is time to move to the second stage of the design process: Developing the conceptual model.

4.3.3 MODEL DEVELOPMENT

The development of the gaming-simulation's conceptual model is extremely important as a system can not be simulated unless is understood. The task at this stage of the design process is to describe as clearly as possible the most salient aspects of the system to be simulated, that is, indicate the substantive content of the real-world system.

This description will present a simplification of the real system that is likely, however, to be far too complex to fully translate it into a gaming-simulation, particularly if play is to be relatively short (i.e., 1-2 hours). Then, the designer will have to review the model, identifying those elements and relationships that are believed to be of greatest importance for inclusion. This process involves not only the definition of concrete elements and specific linkages, but also the interconnection of their roles, goals, resources, and rules. The information processing that constitutes part of the model also has to be defined. A flow network has to be described, specifying the amount and kind of information to be handled and the timing of its presentation.

It is very much a two stage process. First a complex model is defined for the most important elements and interactions of real life systems. In a second stage, the model must then be simplified in order to convert the model into a game.

4.3.3.1 BUILD OF THEORETICAL COMPLEX MODEL

The general game model needs to be as extensive as possible, where all the different elements and interactions affecting the production process of manufacturing organisations could be seen. Its structure is based on the information flow associated to the production process, which is not relevant to this thesis and of which knowledge is assumed.

In Nottingham Polytechnic's gaming-simulation, the elements to be analysed have much to do with the functions within manufacturing organisations, their internal elements, and the existing interactions among them. All these factors were mentioned in chapter two and Appendix A, and will be the core playing ingredients of the gaming simulation.

Functions

The first step in creating this model was to establish the different core functions that the factory being simulated will be comprised of. Based on what was expressed about the functional structure in the second chapter of the thesis, it can be stated that there are seven functions to be considered :

- Product design
- Production Planning
- Production Control
- Purchasing
- ♦ Marketing
- Finance
- Personnel

From these seven functions, there are only five that could be consider fully related to the production process (Product design, Production Planning, Production Control, Purchasing, and Marketing). Additionally, Production Planning and Production Control may be considered as just one function from a game's point of view.

Elements of the model

Once the kind of manufacturing organisation and the functions that will be included in the model are established, it is time to define the core elements that, in our opinion, form part of the real interactions and conflicts within manufacturing organisations. These elements are the factors that influence and distort decision making in the functional areas of manufacturing organisations.

A summary list of the elements, divided by function, that can be found in a manufacturing organisation are :

Product design

- New design suggestions
- Modifications
- Design modification policy.
- Design's ease of manufacture
- Modular design

Marketing

- Customer orders
- Forecasting
- - Lead time
 - Customer service policy (price and delivery)

Production Planning & Control

- Routing
- Production lead times
- Stock policies
- Made it / Bought it
- Quality policy
- Scheduling policy

Purchasing

- Purchasing batch policy
- Quality policy
- Delivery policy

Based on those functions and elements the general model defined is a highly complex one, as seen in figure [4.2] and figure [4.3].



Figure 4.2 Nottingham Polytechnic's general game model

As it can be seen from the previous list and figures, there are a lot of



Figure 4.3 Nottingham Polytechnic's general game model.

elements that influence real life manufacturing organisations and that, due to their huge complexity, can not be addressed within a simulation game. Any attempt to do so, would imply the creation of a real simulation system capable of coping with all the internal and external elements of manufacturing organisations, something hardly possible. A model based on all those factors would not be a successful framework for a gaming-simulation's final purpose: **Learning and Experiencing.** Hence, this general model had to be simplified in order to continue addressing the interactions and conflicts of manufacturing organisations, but within the existing technological and design constraints.

4.3.3.2 SIMPLIFY MODEL

The main problem when designing a game related to manufacturing organisations, is to simplify the real organisation sufficiently for players to handle it in gaming terms while retaining crucial relationships and patterns of interaction.

Additionally it is necessary to simplify the game from the developers point of view so that he can quickly create a prototype that will give insight into the game's behaviour, but again there must not be so much simplification that core problems are not addressed at an early stage.

In order to create the initial version of the gaming-simulation, the model was simplified in two different ways. Firstly, the number of functions was reduced to two (Production Planning and Control, and Marketing). The reason for that was based on the fact that the interaction between these two functions is one of the most important and more conflicting matters of manufacturing organisations, as was seen in Appendix A and is stated by Goldratt [37].

Secondly, the number and extent of the functional elements to be included in the gaming-simulation was reduced. That simplification was once more done with the idea of keeping the sense of the interaction between the two functions while reducing the complexity of the game in order to make it playable. The resulting list of elements to be integrated into the model was :

• Inputs :

- Marketing / Sales :
 - * Customer enquiries
- Production Planning / Control :
 - * Orders to deliver

• Decisions to make :

- Marketing / Sales :
 - * Price
 - * Delivery Date
- Production Planning / Control :
 - * Scheduling (Quantity + Sequence)

• Outputs :

- Marketing / Sales :
 - * Orders to deliver
- Production Planning / Control :

* Stock to deliver the orders

- Company :
 - * Benefits or loses
- Feedback :
 - Customer enquiries



Consequently, the model itself will have the structure shown in figure [4.4].

Figure 4.4 Nottingham Polytechnic's actual game model.

In this latter model the information flows in a sequential loop, from the customers that make enquiries to the people in the Marketing/Sales department who will have to quote those enquiries. Afterwards, the orders will go through a procedure which will simulate the customers' acceptance process (i.e. some quotations are accepted and some are not). From there the information about the accepted enquiries flows to the people responsible of the Production

Planning/Control department that will decide about the scheduling needs required for manufacture of the products ordered by the customers. With the defined scheduling the game will simulate the production process, which will update the information about the stock situation. Finally, the loop is closed by a preestablish relation (feedback) between the customer service performance (price + due date performance) and the amount of customer enquiries to enter the system the next period to be simulated.

Therefore, the gaming-simulation will consist of two basic, very simple, mathematical models. The one that regulates the customer quotations' approval (i.e. firm customers' orders) and the one that regulates the feedback (i.e. customers' enquiries).

4.3.3.3 ORDERS' APPROVAL

The orders' approval function is the one that simulates the acceptance of the sales department's quotation by the customer. If the quotation is accepted, it becomes a firm order that will have to be delivered to the customer and therefore, the planning department should have into account in its scheduling.

This approval function is based on two elements:

- Quotation Price
- Quoted Delivery Date.

These two elements depend on two factors : the closeness to the customer's price and delivery date expectations. The spectrum for these two factors is from 0% to 100%, depending on whether the quotation is far from what the customer expects, or equals, even improves, his expectations. The models that regulate this two factors are shown in figure [4.5].



Figure 4.5 Orders' approval functions.

Hence, the ultimate model that simulates the approval process is function of a combination (multiplication) of the previous two factors and a calculated pseudo-random number (percentage). If the multiplication of the two factors associated with the quotation is greater than or equal to the generated pseudorandom number, the quotation is converted into a firm order. Otherwise the customer's quotation is rejected. Approval = Random Number (%) <= [Price factor (%) * Delivery factor (%)]

4.3.3.4 FEEDBACK

In all gaming-simulations there must be a feedback component capable of showing the dynamics of the model being played and measuring the convenience of the decisions taken. In Nottingham Polytechnic's gaming-simulation, the feedback component chosen is the *number of customer enquiries* for each gaming period.

That number of customer enquiries is function of two elements :

- ♦ Average Quoted Price
- Due Date performance.

These two elements are, themselves, associated with two factors, each of them giving a numerical estimation (percentage) of the closeness to the market's price constraints and factory's due date performance. The spectrum for these two factors is from 0% to 100%, depending on whether these two factors are far from what the market and customer expects, or equals, even improves, its expectations. The models that regulate this two factors are shown in figure [4.6].



Figure 4.6 Feedback approval functions.

Hence, the ultimate model that simulates the feedback process is function of a combination of the previous two factors (minimum of the two factors) and the average number of enquiries so far. If the minimum of the two factors is low, the number of customer enquiries for the next simulation period will be reduced. On the other, hand if the two factors are more positive then the number of enquiries for the next simulation period will be increased.

The use of the average number of enquiries as an element of the feedback system is due to the fact that a malfunction in the factory performance will not affect the factory's competitiveness in a drastic and immediate way, rather in a smooth way. In other words, if the factory does not perform well today, it does not mean that this afternoon will be out of business. To implement this idea, a numerical function (depending on the number of days already simulated had to be added to the feedback system (figure [4.7]). The reason for that was that the effect of the average number of enquiries in the feedback system could not properly be seen in the defined gaming time and therefore, it had to be featured.



Figure 4.7 Smoothing function.

Consequently, the feedback function is :

Number of enquiries = 1 + [((6n-1)/5n)*average number of enquiries*min(pe,de)]

where "n" is the number of periods already gamed, and "min(pe,de)" is the minimum factor (percentage) associated to the market price and due date performance. It has to be said that the number of enquiries can never be zero, it will be at least one. That feedback element is calculated for each one of the final products in the factory being simulated.

There is also another important element in the model which has not been mentioned yet, but is primarily present in all aspects and stages of the model. That important element is **Time**.

4.3.3.5 THE TIME ELEMENT

Its relevance is based on the dynamic nature of games, and on the use of time as a factor in the decision-making processes embedded in a management game. As Kibbee et al. [76] expressed long ago :

"The second major contribution of management games from the learning standpoint is the time dimension...the game has advantages over a case study in the training of a manager. It is alive. Its state is constantly changing in response to previous actions. Planning must consider the present and future simultaneously. Emergencies must be anticipated. Crisis must be met. With no other teaching technique has it been possible to demonstrate so vividly the effects of sequential decision-making in a business environment.

Time is an important factor in games in another way. Participants are often placed under conditions of severe time limitations to simulate the stress encountered in a real management situation". Whilst the claims for the "dynamic" aspect of gaming-simulations is an aspect to be confirmed by this thesis's work, a major characteristic which detracts from this "dynamism" is that decision-makers in most of the games are forced into making decisions on a regular cyclical basis. This characteristic forces games into an analytic framework and into an unrealistic environment in which all decisions are made regularly and simultaneously. For a game regarded as truly "dynamic" it is vitally important, to integrate the time dimension into the structure of the game.

Based on the former assertion, Nottingham Polytechnic's gaming-simulation has tried to address the time element in a positive way. The approach taken for that, is based on the inclusion of **time** as an "endogenous" factor [77] within the game. That method, opposed to the inclusion of **time** as an "exogenous" factor [77], will permit the player effectively use **time** as part of his strategy. That approach is highly encouraged by Loveluck [77], who thinks that to reflect contemporary management practice, time must be introduced as an endogenous variable. In his opinion this must be done by developing "realtime" games which will involve the formulation of multi-user computer models, into which time compression and consumption is made a specific decision alternative. Thus, one of the objectives of this work is to study the feasibility of the aforementioned "real-time" games. There are several possible ways in which this "real-time" games may be created. Nottingham Polytechnic's gaming-simulation has done it in two different ways.

First, time has been introduced as an "endogenous" factor by simulating the passing of each day (at a rate preestablished by the game manager; e.g. 3 minutes per day) regardless of the decisions made. The point in time at which decisions are made becomes a decision in its own right; timing of decisions becomes part of team strategy. The difficulty of this approach lies in the

technology used (hardware and software), a limitation experienced throughout this thesis's work.

Second, the model has been developed in a way that will allow the simulation to be stopped (frozen by an interrupt function); avoiding, at the same time, the input of data from any other of the functions involved in the game. A dialogue can then be set up which allows the participants to discuss their behaviour and address their worries.

It is only under these conditions that the timing of decisions becomes a genuine option in the decision-makers' range of alternatives, and that games become models of dynamics rather than comparative statics.

4.3.4 DECISIONS ABOUT REPRESENTATION

The decisions to be made at this stage of the gaming-simulation design process are of two different natures. They either are related to the style of the game or to its form.

4.3.4.1 STYLE

. 33

The dimensions of the gaming-simulation related to the style can be divided into the following three components :

4.3.4.1.1 LEVEL OF ABSTRACTION

In Nottingham Polytechnic's gaming-simulation the discussion of the level of abstraction is very much related to the use of a computer as the mechanism for playing the game.

The game is shown to the players in a numerical way. Players are dealing with a set of data, that give them information about the status of the different areas involve in the simulation of the manufacturing organisation. Moreover, players response to the game is also a numerical one. They have to input numerical data in order to answer to the situation they are presented with.

4.3.4.1.2 TIME FRAME

Another aspect that must be considered when deciding about the game's representation is **time**. The more "realistic" the gaming-simulation is, the more important this factor is.

There are two elements related to the time aspect. One is the real time that players spend playing the game and another one is the time that is going to be represented in the game. The former has already been defined in the "Context of Use" section in the stage about **SETTING OBJECTIVES AND PARAMETERS**, now the latter element has to be defined. The period to be represented is related to the subject matter and purpose specified in the stage about objectives and parameters.

There are actually two aspects of the time to be represented that must be considered: the length of total time to be represented by a game session, and the length of time to be represented by a cycle. The former must be long enough to encompass the full expression of the behaviour and to show the response of the system to the players' principal decision options. The latter must be fine enough to show the true shape of the dominant dynamics.

In Nottingham Polytechnic's gaming-simulation, the decided starting time frame is based on having a cycle representing at least 10 days (8 working hours per day) played twice, and separated by a discussion about the first cycle. Anyway, the time of representation should be continually reviewed and adjusted based on the experience gained during the played gaming sessions. For that to be possible, the gaming-simulation had been constructed in a flexible way, allowing the game manager to adjust that period time to whatever he/she thinks is appropriate.

4.3.4.1.3 INTERACTION

The decision about the kind of interactional structure to be used in the gaming-simulation is also an important feature, and is inherent in the nature of the game being explained in this thesis.

Nottingham Polytechnic's game is considered to be a highly interactive one, due to the nature of the manufacturing relations that are being presented. Different players are linked to a network which manages the game's common files. This high interaction implies that the game is a more looser, more informal, and less predictable than tightly-structured games.

4.3.4.2 FORM

Another matter when making decisions about representation is the question of form and of how to incorporate each element of the conceptual model into the gaming-simulation. This question can be divided into six different elements (scenario, roles, procedures and rules, external factors, visual imagery and symbols, and accounting system).

4.3.4.2.1 SCENARIO

The players in Nottingham Polytechnic's game are situated in a manufacturing organisation structured in a functional way. The two functions the company consists off are : Marketing/Sales and Production Planning/Control.

S. 4.

In this scenario, it was decided that the manufacturing company to be simulated had to be quite simple to start with and within a game structure capable of handling more complex future models.

The initial limitations were :

34 6

- Five final products.
- No parts. Products are manufacture with no material.
- One operation manufacturing process.
- Two resources
 - One produces three final products.
 - The other produces the other two.
- Make to stock

Then, the problem that players have to face in this "simpler", but flexible manufacturing context, is the one of coordination between the mentioned functions in order to make the whole company to success.

4.3.4.2.2 ROLES

The kind of roles associated with the gaming-simulation are related to the two existing functions within it. The players that undertake the tasks connected to the Marketing/Sales department, will have to answer the incoming enquiries, by quoting their prices and delivery dates.

On the other hand, the players concerned with the Production Planning/Control department will have to define the adequate schedule (quantity and sequence) required to produce enough stock for the customer orders to be delivered on time. Their decisions will be based on the existing information about the firm customers' orders.

In both simulated departments the decision-making process is subject to a situational stress. That stress pressure is due to two factors : the time pressure element (explained in the model development stage) and the inadequacy of the information they have to base their decisions on. That latter factor is a vital educational element for addressing the importance of coordination.

4.3.4.2.3 PROCEDURES AND RULES

The decisions about procedures and rules of a gaming-simulation are very important when dealing with non computer based games. But when dealing with computer games, the rules are determined by the designer's decisions about the structure and functioning of the program/s, and by the set of game parameters decided by the game manager.

4.3.4.2.4 EXTERNAL FACTORS

The Nottingham Polytechnic's gaming-simulation model has only two external factors : the customer enquiries entering the system and the firm customer orders to be delivered.

Both of them are regulated by functions that use pseudo-random numbers, as was explained in the previous stage of the design process. The information attached to those factors, concerns the quantity of the enquiry/order and its expected delivery date. Those values are calculated with pseudo-random numbers and will vary between preestablished values, that are parameters of the game that may be modified by the game manager.

4.3.4.2.5 VISUAL IMAGERY AND SYMBOLS

The definition of the visual representation to be used in the gaming-simulation is very much related to the materials and equipment used for its construction.

The gaming-simulation concerning this thesis is a computer based one, and therefore its equipment is based on two elements : Hardware and Software.

As it was explained in the design stage related to the setting of objectives and parameters, the hardware use is an existing network of PCs (NOVELL [96]). The software used to create the program is a database management system

capable of handling great amounts of information in a multi-user environment. The software chosen was FOXBASE+ [78], and the reasons for that were based on the fact that the system provided a very convenient means of creating certain features of the game (screens, menus, etc) and at the same time it was one of the fastest DBMS available. The database approach also provided the means for concurrent information availability, a element of recognised importance [79].

In this context, the visual representation chosen was a "browsing system", that can be defined as a full screen display system with input and data selection capabilities. That "browsing system" can be seen in figure [4.8], and will also be referred to in the chapter about the software design.

RE: Fra Fra Fra	KOUNCE M K2 K3	PESCA: PINAL PINAL PINAL	ASSEM. Assem. Assem.	01 02 03	489 188 188	SETUP1 1.99 1.50 2.99	3ETUP2 1.10 1.60 2.10	1.29 1.78 2.39	SETUP4 1.30 1.80 2.49	SETUP5 1.49 1.30 2.59	
DUSE :		t (A		(Del)		(Ret	(FR)				

Figure 4.8 Browsing system.

4.3.4.2.6 ACCOUNTING SYSTEM

When defining the elements of Nottingham Polytechnic's gaming-simulation, the performance measure system (accounting system) was defined as one of the important elements to enter the model. As it was explained in the second chapter of this thesis, the "reward" system is one of the elements behind the conflicts between local and global objectives, in other words between the existing functions or areas of a manufacturing organisation and the organisation itself. A conflict that is the core subject of the educational objectives to be addressed in this thesis's game.

With that in mind, the game is based on the "competition" of different functional elements within an organisation, rather than the competition for an existing market between different vendors. In that context, it can not be clearly defined who is doing well or winning. However, what can be fostered is a discussion about how the different functions and the whole company are performing. To "measure" that performance, two kind of performance measures were defined : local and global ones. While the local performance measures were defined to be associated with the performance of the different functional departments being simulated, the global ones were defined to be related to the performance of the company, as a whole. That arrangement was thought to make players aware of the fact that although the performance of each one of the departmental functions could be very positive, the performance of the company as a whole might not. The system was also thought to make people aware of the interrelations and conflicts between the performance in the different departments.

95

The selected local performance measures were :

Marketing/Sales Department

- Number of enquiries received
- Number/Percentage of enquiries quoted
- Number/Percentage of firm orders
- Number of enquiries to be delivered on the current period

- Due date performance
- Average delivery delay
- Volume of sales orders

Production Planning/Control

- Number/Percentage of incomplete orders
- Number/Percentage of started orders
- Number/Percentage of finished orders
- Each product's production
- Number of set-ups scheduled
- Time/Percentage spent in set-ups
- Level of stock

On the other hand the selected global performance measures were :

- ♦ Throughput
- Inventory
- Operating Expenses

These are measures that could easily be related to standard financial measures as :

- ♦ NET PROFIT (NP) = THROUGHPUT OPERATING EXPENSES
- ♦ RETURN ON INVESTMENT = NET PROFIT (NP) / INVENTORY

These are some of the performance measures as proposed by people like Goldratt [80], William J. Bruns et al. [81], and S. Kaplan [82].

All of the former mentioned performance measure elements, are calculated for each one of the simulated periods, also keeping a record of the average performance so far.

Coming back to the stages concerned with the design of gaming-simulations, and once the decisions about representation have been made, it is time to start constructing the game.

4.3.5 CONSTRUCTION AND MODIFICATION OF THE GAMING-SIMULATION

The construction of gaming-simulation was based on the prototyping methodology [83]. The game construction is based on building and using a model of a system for designing, implementing, testing, and installing the final game. When all game parts are assembled into a prototype, the gaming-simulation must be field tested, debugged, retested, and so on until it operates successfully. Considerable trial and retrial is usually necessary before the game generates the behaviours and outcomes characteristics of the referent system. Each error or disappointment, leads the designer to a reexamination of the conceptual model and the design decisions made, as well as to check for possible failures in the construction process; all that in search of a missing element or elements inaccurately linked to others. This whole process could be quite disappointing, for the point at which you are "almost there" but not "there yet". Greenblat [55] suggests that when the game has run smoothly 10 times, it is time to proceed to the next and final stage of the design process.

In theory, at this stage of the design process, it has to be decided whether to use a computer for game operations, and if so construct the needed programs. Due to the nature of the objectives of this research the decision about the use of computers for game operations was a component inherent to the research in itself. Nottingham Polytechnic's gaming-simulation was, by definition, to be developed on computers linked to a network.
With this in mind, and due to the complex nature of the technical matters involved in the construction and development of the programs discussed in this thesis's computer game, there is a whole section named "SOFTWARE DESIGN" dedicated to this aspect.

4.3.6 PREPARATION FOR USE BY OTHERS

After looking at the four previous stages of the gaming-simulation's design process, it is time to look at the last of the stages. The stage related to the preparation of the gaming-simulation for use by others.

This stage is an important and often complicated one. This stage of the design process is carried out when the designer is satisfied with the way the gamingsimulation works (i.e. technically) and corresponds to the "real-life" situations. The stage about preparation of the game for use by others, involves the creation of the operator's manual and the dissemination of the gaming-simulation. The operator's manual should have a technical overview, a explanation of the game's conceptual model, a description of the game in operation, how the game should be run, and a guide to post-play discussion. Additionally, the dissemination involves the preparation for the distribution of the gaming-simulation and its review, both informally and formally.

This preparation stage has not been accomplished during the design of Nottingham's Polytechnic gaming-simulation. The reason for that is based on the fact that the result achieved is not completely satisfactory. Although it is thought that the educational purposes could be accomplished by a basically similar model (user friendlier), the existing technical limitations make the game non-playable at an advance stage of the game (e.g. too much information to handle, the game slows down at a great pace, etc).

5 SOFTWARE DESIGN

5.1 GENERAL ARCHITECTURE

The architecture of the system is based on a network of PCs (NOVELL [96]) and database management system (FOXBASE+ [78]). The system uses three personal computers and the network's file server. While two of the personal computers will respectively simulate input/output tasks related to the Marketing/Sales and Production Planning/Control departments, the third personal computer is used to control the game (simulate the time system) as well as simulate the run of the factory (generate enquiries, simulate the approval of quotations, simulate the production process, and simulate the delivery of customer orders). On the other hand, the file server will store the data files being used in the gaming-simulation, allowing their access from any of the personal computers. The actual hardware architecture of the system is shown in figure [5.1].



Figure 5.1 Game's computer architecture.

The flow-chart of the whole game can be seen in figure [5.2]. That figure shows the different data files used in the game as well as the relations among them.



Figure 5.2 Game's flow-chart.

Each computer runs its own specific program [Appendix C], the important elements of which are described in the section about "technical problems".

5.2 USERS' VIEW OF THE GAME

Nottingham Polytechnic's gaming-simulation is based on four screen handling facilities : menus, browsing system, report system and messages.

The menu systems gives the users the opportunity to decide which kind of activity they are going to perform. The browsing system on the other hand, is related to the more specific data view and edition activity.

The report system will show the information concerning each department's daily and overall performance, as well as the global company's one. Finally, the messages are used to inform the players about the status of the game.

5.2.1 MENU SYSTEM

The menu system is used in different computers to perform different activities. In the computer simulating the factory run there are different menus that will allow the game manager to :

• Configure the resources of the factory (figure [5.3], figure [5.4])



Figure 5.3 Game manager's menu option.



Figure 5.4 Game manager's menu option.

• Start the game (figure [5.5]).

and the second second second second second





• Save a day's final data, run another day, or exit the game (figure

[5.6])



Figure 5.6 Game manager's menu option.

• To continue or quit a "frozen" game (figure [5.7])



Figure 5.7 Game manager's menu option.

The program simulating the Planning area works with only one menu system. That one, allows the user to select the printer as the reports' output device and display the reports (figure [5.8]).



Figure 5.8 Planning department's menu option.

The program simulating the Marketing/Sales area uses the previous report menu system, as well as another one that allows the player in this area to input the standard price for each one of the final products, or select the display/input of quotations (figure [5.9]).



Figure 5.9 Marketing/Sales department's menu option.

5.2.2 BROWSING SYSTEM

The "browsing" system, as it has already been defined in the design stage about representation, is a full screen display system with input and data selection capabilities.

The information displayed on the browsing system depends on the subprogram that the browsing system is affecting to. The browsing system for the Marketing/Sales department, for example, displays information about the coming enquiries (figure [5.10], [5.11]), giving the players the possibility to :

866	MOTOR	MUDEL 5	12.99	12	04/07/91	RNGO
1994 1992	MOTOR	MODEL 4	20.00	11	01/07/91	ENGO

CODE 11991 11992 11993	QUANTIT 12 20 19	STUCK O O O	DATE 91/07/91 02/07/91 91/07/91	RESOURCE FAG1 FAG1 FAG1 FAG1	CAPACITY 480 180 180	UNTINE 12.34 13.24 16.23	32TUP 1.99 1.18 1.29	TOTALTINE 2h28m 6h24m 2h42m
								•
TATUS	: (P)1a	nn (C)o	de (Arrow	z>				

- Figure 5.11 Enquiries' browsing.
- Quote them as enquired » (A)ccept
- Quote something different » < Enter >
- Select the enquiries by code » (C)ode
- Select the enquiries by resource code » (R)esource
- Select the enquiries to quote » < Arrows, PgUp, PgDn >

The one associated with the Planning Department, on the other hand, displays information about the customers' firm orders and about the scheduled orders (figure [5.12], [5.13]). The key choices for these two displays are :

Customers firm orders' display :

CODE MBB1	QUANTIT 12	STOCK	DATE 01/07/91	RESOURCE FAO1	CAPACITY 480	UNTINE 12.34	SETUP 1.00	TUTNLTIME 2h28m
1993	19		82/87/91 91/97/91	Fa91 Fa91	489 489	13.24 16.23	1.10	6h24m Zh42m
TATUS	S: (P)lann	(C)ode	CAPPON	CORF.		NIRCE-FA	A1	



- Display the scheduled orders » (P)lan
- Select the firm orders by code » (C)ode

- Select the firm orders by resource code » (R)esource
- Select the customers' firms orders » < Arrows, PgUp, PgDn >

a state and a second a second the mark with a second second second second second second second second second s

Scheduled orders' display :

HOGZ HOTUR HODEL 2 20.90 264.80 FA01 9.60 STARTED	NOWSE	0	1)eu	CURRENT	(R)esource SELECTIO	(S)tatus 1: CODE-AL	APPOME> RESOURCE	(Bel)	(Return)
1992 HOTUA HODEL 2 28.99 264,89 7491 9.99 STAKITE									
1992 MOTOR MODEL 2 20.99 264,89 7491 9.99 STARTED									
1993 MUTUR MUDEL 3 19.69 162.39 7461 19.69 FINISHED	1993 1992	MOTOR MOTOR	MODEL	3	18.89 29.99	162.30 264,89	7691 7691	10.08 9,00	p in ished Started

Figure 5.13 Scheduled orders' browsing.

- Display the customers' firm orders » (S)tatus
- Create a new scheduled order » (N)ew
- Select the scheduled orders by code » (C)ode
- Select the scheduled orders by resource code » (R)esource
- Select the scheduled orders to be re-scheduled » < Return >
- Delete the selected scheduled order » < Del >
- Move through the scheduled orders » < Arrows, PgUp, PgDn >

There is also another browsing system associated to the computer simulating the run of the factory. That one is used to input information about the resources within the company to be simulated. That information is related to set-ups and capacities (figure [5.14]). The browsing system gives the game managers the possibility to :

RESC FAGE FAGE	UNCE DE I FI I FI	SCRIPTI NAL ASS NAL ASS NAL ASS	DN EM, 01 EM, 02 EM, 03	CAPAC ITY 480 480 480	SETUP1 1.00 1.50 2.00	SETUP2 1.19 1.60 2.19	32TUP3 1.20 1.70 2.30	1.30 1.00 2.49	SETUP5 1.40 1.90 2.50
ROUSE :	(E)xit	Arrou	aa> ∢Du	al) (N)e	w (Rets	4PH>			

Figure 5.14 Resources' browsing.

- Exit this display » (E)xit
- Move through the existing list of resources » < Arrows >
- Delete an existing resource from the list » < Del >
- Create a new resource » (N)ew
- Edit an existing resource » < Return >

Because the game is played interactively, the browsing screens are refreshed as enquiries are received and quoted, and orders are produced. After a day has finished they will also be refreshed.

5.2.3 REPORT SYSTEM

The report system used in all program areas (Factory, Planning, and Marketing/Sales) is a full screen display of the performance measures associated to each one of the areas. Those reports can also be diverted to the printer.

The factory reports, for example, show the global performance measures associated to the whole of the factory (figure [5.15]).



Figure 5.15 Factory's performance.

On the other hand, the reports associated to the Planning area display information about the WIP, production, set-ups, stock levels, etc...(figure [5.16],[5.17],[5.18],[5.19]).

		PLANNING BEFARTNE Performance (P	NT'S PAST age 1)	
Haek Code	0 Bat M903 De:	a 91/97/91 script MOTON MODEL 3		
14 mm - (14 mm - 140 mm -	of incomplete ord of started orders of finished order:	DAY QUENALL IFX & O A A I 1 I 1	x over total orders x over total orders x over total orders	DAY OVERALL 0.00 0.00 0.00 0.09 100.00 100.00

Figure 5.16 Planning reports.



Figure 5.17 Planning reports.



The second stand

and a second with a later and a second a reading the second

Figure 5.18 Planning reports.



Figure 5.19 Planning reports.

Finally, the reports concerning the Marketing/Sales area will show information about the enquiries received and quotations made (figure [5.20],[5.21]).





		PER	S DEPARTMENT FORMANCE (Pa	s PAST (* 2)	1		
Acek		Date	61/67/91				
Code	1001	Descript	MOTOR MODEL	1			
Junted Eng	uirss' Ju	CCess X :		Achieves	a orders	SUCCOST	х :
		DAY	OVERALL			DAY	OVERALL
Delivery	Date	0.00	9,66	Pelivery Delivery	Dato	0.00	6,66

Figure 5.21 Marketing/Sales reports.

5.2.4 MESSAGES

The programs in the game-use two different kind of messages. Some of them they are simple information messages, which give information about the status of the system or different kind of warnings (figure [5.22], [5.23], [5.24], [5.25], [5.26], [5.27], [5.28], [5.29], [5.30], [5.31]).







Figure 5.23 Messages.



*...... B...

and the second second to be a second to a second second

はないないい、人を見た、 あっていたい はない いい ひち かっ

Figure 5.24 Messages

		RUMNING PACTORY		
DATE :	0 2/07/91		TIME:	6:34
	PRESS (END)	08 Ctr]~F 70 FM		are.

Figure 5.25 Messages.



Figure 5.26 Messages.



Figure 5.27 Messages.



2.20

Trate . dis the server

あるという

いっていない いたいにない あいないない

ないたいとうないであっていたいで、こうちょう

Status AV. T. Hand

1

and the so toport is the

Figure 5.28 Messages.

1 ...

and a manage

	calculating sales response
1	

Figure 5.29 Messages.



Figure 5.30 Messages.



Figure 5.31 Messages.

There are some other messages that are input screens, which are used to input information regarding different elements of the game as input of quotations, configuration of dates, input of standard price, and so on (figure [5.32], [5.33], [5.34], [5.35], [5.36]).

	dna.	TATIONS IMPUT	
Article Code	19 H 1-1	Description	1 A
Quantity Normal Price Delivery Date	12 (H) 20,00 (H) 97291	Quoted Price Quoted Delivery Date	20.00
Enquire Date Enquire Time	07/91	Quotation Date Quotation Time	

Figure 5.32 Input screen.

	INPUT OF ST	ANDARD PRICE	ב	
Code	1111 Descript	MOTER MODEL	1]
Stando	8st 242,61	Standyrice	10.00	

Figure 5.33 Input screen

	CONFIGURE BATE TO RUN.
ENTE	THE GAME'S STARTING DATE : 01/07/31

Figure 5.34 Input screen.



Figure 5.35 Input screen.



Figure 5.36 Input screen

5.3 DEALING WITH TECHNICAL PROBLEMS

The source code and structure of each of the sub-programs, as well as the data files names, structures, and associated index files can be found in [Appendix C].

Although FOXBASE+ [78] offers many good programming features, it is not a system specifically designed to cope with simulations. Moreover, things get complicated when trying to simulate "real-time" simulations. Therefore, there are some technical problems that need to be solved by the creation of suitable programs and procedures.

5.3.1 FILE CONFLICTS

Nottingham Polytechnic's gaming-simulation is based on a multi-user environment. In this context it is necessary to have a procedure to deal with the data access conflicts, i.e. one user attempts to access a record or file which is currently being accessed by another user.

FOXBASE+ [78] offers the existence of two locking functions, RLOCK() and FLOCK(), that deny the use by others of a record or a file respectively. Those two functions together with the "ON ERROR" command line offer the possibility to make the program behave in one way or another when a data conflict arises. Therefore, and based on those elements it was created a procedure called "err_fix" [Appendix C] for handling the data conflicts in each one of the sub-programs.

いろう ちんのない いちんない いちんしん ちんしん

i white white is

5.3.2 COORDINATION

When trying to simulate a "real-time" gaming-simulation in a personal computer under the DOS operating system [84], there is one major limitation to be aware of. That limitation is the impossibility of multitasking (performance of different instructions at the same time).

That restriction restrains the creation of gaming-simulations based on a continuous algorithm. In other words, the computer simulation will perform

tasks on a sequential loop, in contrast to real-life situations, where multiple events happen at the same time. That implies that the information concerning the simulation will not be continually updated (i.e. real-time system). The information will only be updated every time the loop enters a new cycle in which all the different calculations will be sequentially calculated. The cycle time is variable and dependent on the time needed to carry out each of the main program tasks.

in the same

あたい こう ちょう たちち いたち ない いのいかない とう

A the standard and the standard and a standard and a

and all all and and the state of a look all

That loop will continually run until the game is stopped by the game's manager or a simulated period has come to its end. The loop will always start checking for the former situation to happen. In second place the program will calculate the absolute simulation time (i.e. simulation time elapsed since the start of the game). If that absolute simulation time has changed since the last loop, then a new calculation cycle will be started. That calculation cycle will involve the transformation of the simulation absolute time into simulation time (i.e. simulation's running day, month, year, hour, and minutes), and the simulation of the production system. Finally the program will check for the gaming period to finish (i.e. a days's simulation has finished). If that is the case, the program will force the run of tasks associated to the finish of a period : the quotation approval, shipping, feedback, and enquiries simulation systems. The cyclical algorithm of the simulation is shown in figure [5.37], and the procedure that performs that algorithm is called "RUNFACT" in the factory program [Appendix C].



and the stand of the stand of the stand of the

Figure 5.37 Nottingham Polytechnic's cyclical algorithm.

where the second of the second have reader and second

The information calculated in each one of the algorithm's cycles is sent to the different data files for updating. That information will also be used to coordinate the two departmental programs with respect to the running of the simulation.

William Star Participation

The only way information can migrate from one computer to another, within a multi-user database management system, is by data files. Therefore, the program created to operate Nottingham Polytechnic's gaming-simulation has been constructed with a control data file called "SYSTEM.DBF" [Appendix C]. That data file is a one record file and contains all the relevant variables needed to control the simultaneous run of the three departmental programs (e.g. running date and time, game's status [frozen, period conclusion, game finished,..], game's global settings, etc.). This enables the game manager to control the other players. For example, if the game is stopped (frozen or a day has ended) the departmental players will not be able to input data. If the day has finished, reports will appear in each one of the departmental computers, etc..

5.3.3 TIME SIMULATION SYSTEM (RUN FACTORY)

The time simulation procedure is based on a function of the FOXBASE+ [78] database management system. That function, called "sys(2)", returns the number of seconds elapsed since midnight.

The first value returned by that function will defined the simulation's absolute starting time. Based on that starting absolute time, the time simulation procedure will calculate the consequent relative times. These relative times will later be affected by an amplification factor that will finally return the actual simulation time. That simulation time will again be transformed to a suitable form for displaying and processing (i.e. "DAY/MONTH/YEAR" «» "HOURS:MINUTES").

The procedure used to carry out all this calculations is called "TIMECAL" in the factory sub-program (SYS11.PRG), and can be seen in [Appendix C].

5.3.4 PRODUCTION SIMULATION SYSTEM (RUN FACTORY)

One assumption of the production simulation system is that a manufacturing resource can not cope with more than one production order at a given time. It also takes into account the influence of the set-up times for each product.

「「「日本」」、「「日本」」」、「「日本」」

Another important element of this calculation is the aforementioned sequential execution of the programs under the DOS operating system [84]. The sequential execution implies, that the production simulation will be carried out by calculating the situation at the end of each simulation cycle period. The stock and production situation will only be known at the end of each cycle, but not during it.

If that is the case, there are three kind of orders and two set-up situations that had to be taken into account, when making the calculations :

- » Orders :
- Order starting and finishing in the current cycle period.
- Order not starting but finishing in the current cycle period.
- Order starting but not finishing in the current cycle period.

» Set-up situation :

- Order with set-up equal to the previous order.
- Order with set-up different to the previous order.

Those elements can be combined into six different calculational situations that the production simulation program will have to cope with. These six different situations are shown in figure [5.38].

a to a to and truck of the solution in the second



Figure 5.38 Production calculation.

Another characteristic of the production simulation system is the fact that it is based on "batch" transfer policy, by which the manufactured products will be transferred to stock only when the whole scheduled batch has been processed.

The procedure used to carry out this calculations is called "RUNPROD" in the factory sub-program (SYS11.PRG), and can be seen in [Appendix C].

5.3.5 QUOTATION APPROVAL SIMULATION SYSTEM (FINISH OF A DAY)

This procedure is defined to simulate the customers' acceptance to what the marketing department quoted for the enquiries received. The procedure is called "GOODSERV" [Appendix C] and it operates according to the rules defined in the model of "orders' approval" (see section 4.3.3.3).

5.3.6 SHIPPING SIMULATION SYSTEM (FINISH OF A DAY)

The procedure "SHIPPING" [Appendix C] is used to calculate which orders will be delivered to the customers. The manufacturing company to be modelled is a "make to stock" one.

If sufficient stock has been produced at the end of a period to completely cover an order then, that order is shipped. The order is marked as completed and stock levels adjusted.

If stock levels do not exist to completely cover an order no partial orders are shipped, the order will be delayed, and this will then affect the feedback in the game.

5.3.7 FEEDBACK SIMULATION SYSTEM (FINISH OF A DAY)

The feedback procedure "FREQUEN", in the factory sub-program [Appendix C], is the one that will control the feedback process of the gaming-simulation.

Feedback occurs because at the end of each day a calculation of the number of enquiries to enter the game the next day is made. That number is a function of the average price quoted, and due date performance (see "Feedback" in section 4.3.3.4). The day being simulated is then divided up into the appropriate number of periods (variable called "frequency"). An enquire will enter the game every period. The actual time of arrival of each one of the enquiries within its own period is based on a random number generation.

The feedback effect can be seen in figure [5.39] and figure [5.40].



Figure 5.39 Enquiries' feedback.



Figure 5.40 Schedule orders' situation.

5.3.8 ENQUIRIES SIMULATION SYSTEM (FINISH OF A DAY)

The procedure that controls the generation of those enquiries is called "ENQGEN" which is within the factory sub-program [Appendix C].

In generating the enquiries there are two elements to be calculated : the delivery date and price. These two elements are calculated by a pseudo-random number which fluctuates between a predetermined low and high limit. Separate limits may be specified for each of the final products.

6 CONCLUSION

In researching the background to this thesis it has become clear that there are few games available that deal with the specific problem of interdepartmental coordination in manufacturing companies, despite its central importance to the success of such companies. Even these games tend to ignore the dynamic aspects of this interaction.

The use of computer based gaming-simulations based on networked personal computers appears to be an appropriate tool for use in education in this areas.

A simplified model has been created which provides the basis for developing such a gaming-simulation and a game has been developed using standard database management system : -FOXBASE+ [78]. This system provided a very convenient means of developing many aspects of the game such as the menus, data input screens, and browsing lists. However it was difficult to use in a number of aspects.

Firstly, it was very difficult to control the progression of time. This resulted in considerable programming effort in this area and even then this slowed the game down to a point where it was difficult to play. Detailed timing tests were carried out¹ and these are shown in Appendix D. From these tests it can be

¹ The evaluation of the actual gaming-simulation was based on the run of several games. Those games involved the set-up of the whole system, and the dynamic input of data (i.e. quotation of enquiries and schedule of orders). Decisions had to be made and input in the system while the gaming-simulation was running. The games run consisted of 10 simulations days each.

seen that, using this approach to controlling time in the game, the game slows down as more orders are loaded in. This is because there is more data to process in each cycle of the time control loop.

No break . at 2 a lo

- 34 Sec. 1. 2. 201 1. 1. 62.03

このできないであっている ちんちん ちんちん ちんちん ちんちん ちんちん ちんちん

Secondly, there were considerable problems in dealing with simultaneous updating of data by different users. The FOXBASE+ software [78] did provide both record locking and file locking facilities but there was considerable programming effort needed to give appropriate responses to users where conflicts were identified.

Finally, the means of data representation provided by this software, i.e. menus, input screens, and browsing lists, provides the user with only a limited view of what the simulation is trying to represent, although it could be argued that this is in many ways an accurate simulation of manufacturing organisations where it is often difficult to see what is actually happening.

Although these problems did in fact mean that the game produced was not successful in the sense that it could not be used in the planned role, they are in a sense superficial and the speed of current developments in software and hardware means that it is only a matter of time before they are overcome. In fact the development of a number of windows based database management systems (including FOXPRO 2.0 [95], the current development of the software used in this research) would make much more sophisticated representation of the data possible than with software that was available at the beginning of the research.

As a consequence this research supports the idea that gaming-simulations can be developed in this area and they are likely to be useful educational tools.

and the state of the second

REFERENCES

the second s

[1] Hayes, Robert H., Wheelwright, Steven C., and Clark, Kim B.; "Dynamic manufacturing"; New York; 1988.

[2] Advanced Manufacturing Application Product Suite Modular Series Introductory Video. Taken from Edwards, J.; "Educating for Change"; BPICS Control, October/November 1990. a the state of the

[3] Kidd, P. T.; "Organisation, People and Technology : Towards Continuing Improvement in Manufacturing"; Proceedings of the sixth CIM-Europe Annual Conference; May 1990.

[4] Goldratt, E. M. and Fox, R. E.; "The Theory of Constraints Journal, volume 1-1"; Tonic Enterprises Ltd. London, England; 1987.

[5] Mondy, R. Wayne et al.; "Management and Organizational Behaviour";Allyn and Bacon, 1990.

[6] Twiss, Brian C. et al.; "Managing Industrial Organizations"; Pitman, 1980.

[7] Managing Change in the 90's - People in Business. Survey conducted in the UK in May 1990. Taken from Churchill, L.; "Beyond Functional Excellence"; Total Quality Management, February 1991.

[8] Galbraith, J.; "Organization Design: An Information Processing View";Interfaces, vol 4, no 3, pp 28-36, 1974.

[9] Kiyoshi, Suzaki; " The New Manufacturing Challenge - The Technique for Continuous Improvement"; The Free Press, 1987.

[10] Frick, Jan et al.; "Activity chains: A method for identifying and evaluating key areas of integration in SME's"; Proceedings of the Sixth CIM-Europe Annual Conference, May 1990.

[11] Schonberger, Richard J.; "Building a Chain of Customers"; Free Press, April 1991.

[12] Burbidge, J. L.; "The Introduction of Group Technology"; London: Heinemann, 1975.

[13] Burbidge, J. L.; "Group Technology in the Engineering Industry";Mechanical Engineering Publication, 1979.

[14] Burbidge, J. L.; " Production Control: The future choice"; 5th National Conference of Production Research, 1989.

[15] Voss, C. A. et al.; "Managerial Integration and CIM - a strategic response."; Computer-Integrated Manufacturing Systems, May 1991.

[16] Voss, C. A.; "Organisational change and CAD/CAM effectiveness"; Working Paper MATT-88W-006, Graduate centre for the Management of Technology, University of Cincinatti, USA, June 1988.

[17] Interview to Tony Kelly, Principal Consultant of Hoskyns Group's; "The price of ignorance"; Computerised Manufacturing, June 1989.

[18] Levy, Paul et al.; "Organizational strategy for CIM"; Computer-Integrated Manufacturing Systems, May 1991.

[19] Helms, Marilyn M.; "Communication: The Key to JIT Success"; Production and Inventory Management Journal, 1990.

[20] Kearney, A. T.; "CIM: Competitive Advantage or Technological Dead End?"; Consultant Report, 1989.

[21] Barrar, P. et al.; "Decision Processes in the Design; Implementation and Use of CAPM Systems in Medium-sized Organisations"; Final report to SERC/ACME Directorate Grant GR/E/21278, 1989.

[22] Buckingham, J. et al.; "Beyond Substitution, Organisational Implications for the Successful Use of Integrated Technology"; A Flexible Future? Prospects for Employment and Organisation in the 1990s, Business School Conference, 1989.

[23] Brown, David S.; "Concert Building: Management's New Goal"; Journal of Systems Management, February 1990.

[24] Neave, Henry R.; "Deming's 14 Points for Management"; Salisbury,British Deming Association, 1989.

[25] Neave, Henry R.; "The Deming Dimension"; Tennessee, SPC Press, 1990.

[26] Taylor, Frederick; "Scientific Management: comprising shop management, the principles of scientific management [and] Testimony before the special House committee / With a foreword by Harlow S. Person.", Westport [conn.]: Greenwood Press, 1972.

[27] Lascelles, D. M.; "The key issues of a quality improvement process";Int. Jnl. Prod. Res., 1990.

[28] Owen, Andrew et al.; "Implementing MRP packages - The lessons learned"; Coopers & Lybrand Associates.

[29] Groves, Brenton R.; "The missing element in MRP - People"; Production and Inventory Management Journal, 1990.

[30] Oliver, Nick et al.; "The Japanization of British Industry"; Basil Blackwell Ltd, 1988.

[31] Manden, Yosuhiro et al.; "Innovations in Management - The Japanese Corporation"; Industrial Engineering and Management Press, 1985.

[32] Burnes, Bernard; "Barriers to employee involvement in technical change: 'more than a case of the good guys and the bad guys"; Adv. Manuf. Eng., April 1990.

[33] Pascale, Richard Tanner et al.; "The Art of Japanese Management -Applications for American Executives"; Simon and Schuster, 1981.

[34] Burbidge, John L. et al.; "Integration in Manufacturing"; Computers in Industry, No 9, 1987 p 297-305.

[35] Burbidge, John L.; "Production Flow Analysis, For Planning Group Technology"; Oxford University Press, 1989.

[36] Goldratt, E. M. and Fox, R. E.; "The Theory of Constraints Journal, volume 1-2"; Tonic Enterprises Ltd. London, England, 1987.

[37] Goldratt, E. M. and Fox, R. E.; "What is this thing called The Theory of Constraints"; Tonic Enterprises Ltd. London, England, 1990.

[38] Vollman, Thomas E.; "Cutting the Gordian Knot of Misguided Performance Measurement"; Industrial Management & Data Systems, 1991.
[39] Waldron, David; "Accounting for manufacture - adequate or archaic?"; Logistics, February 1991.

[40] Helms, Marilyn M.; "Meeting the Human Resource Challenges of JIT through Management Development"; Jnl. Management Development 9, 1990.
[41] Marsden, Alan; "Training for Change"; The Plant Engineer, March/April, 1990.

[42] Heard, Julie A.; "Education: The Key to JIT Success"; Proceedings from the 1985 Conference of American Production & Inventory Control Society.
[43] Tiernan, Tony; "Change is as good as rest...Provided you train for it"; Works Management, August 1989.

132

1 . C. L . . .

[44] Wedberg, George H.; "...But First, Understand The Problem"; Journal of Systems Management, June 1990.

[45] Millard, Robert I.; "Is MRP training aimed in the right direction?"; Production and Inventory Management Journal, 1989.

[46] Batley, T. W.; "Microcomputer Simulation for Teaching Operations Management"; Int. Jnl. of Operations and Production Management, 1991.

[47] Chong, Philip S.; "Using Microcomputer-based MRP Software to Teach Materials Management"; Production and Inventory Management Journal, 1989.

[48] De Lurgio, Stephen A. et al.; "Teaching Integrated Production and Information Control System Principles Using a Spreadsheet Simulator"; Production and Inventory Management Journal, 1989.

[49] Wieters, C. David et al.; "Software Supporting Introductory Courses in Production/Operations Management"; Production and Inventory Management Journal, 1989.

[50] Franklin II, Carter L.; "Are Microcomputers Useful in Management Education?"; Journal of Systems Management, 1990.

[51] Loveluck, Clive; "Notes on the construction, operation and evaluation of Management Games"; Management Games Ltd., 1969.

[52] Neumann John Von et al.; "Theory of Games and Economic Behaviour";London: Wiley, 1953 [1967 reprint].

[53] Tansey, P. J. et al.; "Simulation and Gaming in Education"; Methuen: London, 1969.

[54] Moreno, J. L.; "Who Shall Survive?"; Boston: Beacon House, 1953.

[55] Greenblat, Cathy Stein; "Designing Games and Simulations - An Illustrated Handbook"; SAGE Publications, 1988.

[56] Brand, C. et al.; "The Simulation of 'real-life' organizations within a game": Perspectives on Academic Gaming & Simulation 6, 1981.

[57] Burgess, Tom; "Management Simulation-Games and Artificial Intelligence"; Simulation/Games for Learning 20-3, 1990.

[58] Emmons, Hamilton et al.; "STORM: Quantitative Modelling for Decision Support"; Holden-Day, Oakland, CA 1986.

[59] Hall, Owen P.; "Computer Models for Operations Management"; Addison-Wesley, 1989.

[60] "TEAMSKILL - Production Management Exercises"; Hall Marketing, 1978.

[61] "TOPAZ"; Edit 515 Ltd.

[62] "EXECUTIVE"; April Computing Executive Ltd.

[63] Sykes, Philip; "BISSIM - Computerised Business Simulation Exercises";Perspectives on Gaming & Simulation 9, 1984.

[64] Ahituv, N. et al.; "Principles of Information Systems for Management";Dubuque, IA: William C. Brown, 1983.

[65] Johnston, Robert; "Complex Interactive Computer-Based Simulations Without Mathematical Modelling"; Simulation & Games 18-4; 1987.

[66] Rohn, Walter E.; "The Present State and Future Trends in Management Games For Management Development in Germany"; Simulations & Games, 1986.

[67] Gray, Carey et al.; "The Development of a 'HOT' Business Management Game"; Perspectives on Gaming and Simulation 14, 1989.

the second state of the
[68] Pruett, James M. et al; "MOSES: Manufacturing Organization Simulation and Evaluation System"; Simulation, January 1990.

[69] Bryant, Nigel et al.; "Changing Technology and Management Games"; Perspectives on Gaming and Simulation 14, 1989.

[70] Loveluck, Clive; "Some Conceptual Changes in Micro-Computer Based Games"; Simulation/Games for Learning 19-4, 1989.

State Cal + Sugar

[71] Keys, Bernard; "Total Enterprise Business Games - An Evaluation"; The Guide to Simulations/Games for Education and Training, 1980.

[72] Keys, Bernard; "Total Enterprise Business Games"; Simulation and Games 18-2, June 1987.

[73] Partridge, S. E. et al.; "Designing Management Games for Production Management Training"; Simulation and Games 5-3, September 1984.

[74] Gibbs, G. I.; "Handbook of Games and Simulation Exercises"; E. & F.N. SPON Ltd, 1974.

[75] Dillman, Duane; "The Design and Development of Simulation Exercises"; Paper presented at the meetings of the American Educational Research Association, Minneapolis, MN, March 1970.

[76] Kibbee, J. M. et al.; "Management Games"; Reinhold, New York, 1961.[77] Loveluck, Clive; "The Function of TIME in Management Games and Simulations": Simulation/Games for learning 20-1, March 1990.

[78] Goley, George F.; "Dynamic of FOXBASE PLUS Programming"; Dow Jones-Irwin, US, 1988.

[79] Crookall, J. R.; "Computer Integration of Advanced Manufacture"; Proceedings of Institution of Mechanical Engineers, Vol. 200, No. B4, 1986. [80] Goldratt, E. M. and Fox, R. E.; "The Theory of Constraints Journal, volume 1-6"; Tonic Enterprises Ltd. London, England; 1987.
[81] Bruns, William J. et al.; "Accounting & Management, Field Study

Perspectives"; Edited by William J. Bruns, Jr. Robert S. Kaplan, 1987.

[82] Kaplan, Robert S.; "Advanced Management Accounting"; London: Prentice Hall, 1988.

[83] Lantz, Kenneth E.; "The Prototyping Methodology", Prentice Hall,Englewood Cliffs: Prentice-Hall, 1986.

[84] Jamsa, Kris; "DOS: The Complete Reference"; Berkeley, Calif.: Osborne McGraw-Hill, 1987.

[85] Pegels, C. Carl; "Integrating Functional Areas for Improved Productivity and Quality"; Int. Jnl. of Operations & Production Management 11-2, 1991.
[86] Tuttle, H. C.; "Breaking the 'Wall' between Design and Manufacturing"; Production, May 1983, pp. 63-6.

[87] Reitzle, W.; "Confrontation or cooperation?"; The FMS Magazine 7, 1989.

[88] Shapiro, B. P.; "Can Marketing and Manufacturing Co-exist?"; Harvard Business Review, September-October 1977, pp. 104-14.

[89] Powers, Thomas L.et al.; "Marketing and Manufacturing Conflict: Sources and Resolution"; Production and Inventory Management Journal, 1988.

[90] "The Hard Sell"; Manufacturing Engineer, March 1991.

[91] Kotler, Philip; "Marketing Management"; Prentice-Hall, Englewood Cliffs, NJ 1977, p.12.

[92] Bechtold, Ronald K.; "The Available-to-Promise Tool"; APICS Annual Conference Proceedings 1981, pp. 386-389.

[93] Plowman, Brian; "Management Behaviour"; Total Quality Management, August 1990.

[94] Coote, Alan et al.; "Some Human and Machine Aspects of Computerized Simulations"; Perspectives on Gaming & Simulation 10, 1985.

[95] "FOXPRO 2.0"; Fox Software 1991.

[96] "NOVELL", Novell Incorporated, 1986.

[97] Tranfield, G. and Igartua, J. I.;"An Interactive gaming-simulation of manufacturing organisations"; Advances in Manufacturing Technology VI. Proceedings of the seventh National Conference on Production Research, 1991.

APPENDIX A

the second s

efficiente de la construcción de la construcción de la construcción de la

DEMING'S THOUGHTS ABOUT CO-OPERATION.

MANUFACTURING

CONFLICTS

DEMING'S THOUGHTS ABOUT CO-OPERATION

The theme of genuine co-operation is compactly summarised in one of the vertices of the Joiner Triangle : "All One Team". This is a foundation-stone of the Deming philosophy. Yet again he becomes aware of the barrier which the merit rating forms to real progress. The worst case is performance appraisal constrained by a fixed distribution where it becomes necessary to put somebody else under in order to get a higher rating. Surely the better way is to have everybody working for the company rather than against each other or against other departments in the company. Departments in the company are so often managed in a competitive framework. Yet every department is a supplier or a customer, or both, to every other department. In a inner competitive environment, what matters the trouble which you cause to the other departments? Indeed, it can then actually be *advantageous* for you to cause trouble to other departments (or at least as much as you can get away with).

Deming's Example : Huge financial advantages of co-operation

Harm that is caused by internal competition and conflict and the fear which is thereby generated, and good that is brought about by internal co-operation and teamwork, is of massive proportions. A Purchasing Manager, under pressure to reduce his figures, changes to a cheaper source, even if he buys poorer products and service as the result. Engineering design imposes unnecessarily tight tolerances to compensate for the fact that manufacturing never reaches the standards asked of it. Departments performing better than budget start spending near the end of the year because they know that otherwise their next year's budget will be reduced. As the end of the month looms, salesmen start doing everything they can to meet their quotas, which scant regard to the problems caused to manufacturing, administration and delivery, let alone to the customer. Figures get massaged, computations "redefined", so that reports show more of what senior management want to see. The CYA (cover your posterior) syndrome holds sway.

「ないいない」「「ないない」」」「「ないないないない」」」「ないない」」」」「ないない」」

The following simple illustration indicates clearly something of what is won or lost in environments of co-operation and conflict respectively. It is based on a example that Deming used in his 1988's seminars, and that Henry R. Neave [24] modified.

For brevity and ease, the illustration is in a small scale. It is not difficult to imagine how the numbers multiply in more realistic examples.

	EFFECTS						
Areas and their options	on Area A	on Area B	on Area C				
AREA A							
AREA B							
AREA C							
		<u></u>					

1. 1. 1. 2. 2. Marth

a the man is as which a

and the state of the second and a second of the state of

Not Bear & St. B.

A . W. Sak

A State of the state

Figure Al

The blank chart shows that the illustration concerns and organisation comprised of just three areas or departments. In the left-hand column will be listed options available for each area to adopt or not adopt according to their choice. The remaining columns will show the potential effects of adoption of the options.

				<u> </u>
		EFFECTS	1	
Areas and thelr options	on Area A	on Area B	on Area C	
AREA A				
CīD	+			
(11)	+			
(111)	+			
AREA B				
CID		+		
(11)		+		
AREA C				
CID			+	
(11)			+	
(111)			+	
		and the stand state of the stat		

S'Landard

a -- Ben and

The of the second second



When there are barriers between the areas (old style management, MBO, inner competition, etc.), each area naturally adopts options which are beneficial to itself.

		EFFECTS	1	
Areas and their options	on Area A	on Area B	on Area C	on the company
AREA A				
CID	+			
(11)	+			
(111)	+			
AREA B				
CID		+		
(11)		+		
AREA C				
CID			+	
(11)			+	
(111)			+	
Net effect of adopted				

at Par of

1. 2. 5

「おいろろう」

at a second a la

Figure A3

However, let us examine the effect of this primitive decision-making mechanism both on the individual areas and hence (by summation) on the whole company. For ease of illustration, let us assume that each + or - corresponds to gain or loss of the same amount of money throughout.

		EFFECTS		
Areas and their options	on Area A	on Area B	on Area C	on the company
AREA A				
CID	+	-	-	
(11)	+	-	+	+
(111)	+	-	-	-
AREA B				
(1)	2 24	+	-	
(11)	+	+	-	+
AREA C				
(1)	+	+ .	+	+ + +
(11)	-	_	+	_
CITIO	-	-	+	-
Net effect of adopted options	+ +		0	0

a long the week on I are

1. 8. 2. E. Ast.

「ない、ないという」、 いいのののないないない

Figure A4

Options locally beneficial to one area may well be quite the opposite for other areas. Suppose each are's gains or losses are as indicated. The net effect on the company happens to be zero in this case, i.e. equal to the effect of doing nothing at all (which would be a rather easier way of achieving the same result!). Depending on the details, the net effect could have been positive, zero or negative.

		EFFECTS		
Areas and their options	on Area A	on Area B	on Area C	on the company
AREA A				
CID	+	-	-	-
(11)	+	-	+	÷
(111)	+	-		-
AREA B				
(1)	-	+	-	-
(11)	+	+	-	+
AREA C				
CID	+	+	+	+ + +
(11)			+	-
(111)	_	_	+	-
let effect of adopted	+ + +	+	+ + +	++++-

Figure A5

If the management environment improves so that areas become aware of the effect of their actions on other areas (they may have been aware already, of course, but it was not in their interests to pay any attention) and the old inter-departmental rivalries are placed by genuine teamwork for mutual and company benefit, options are now only adopted if they produce net benefit to the company. Consequently, only three of the previous eight options are now adopted. (Entries relevant to adopted options are characterised in the Figures by bold print; the options now rejected have been shaded over.) By this more judicious and restricted choice of action (i.e. wise choices of both actions and inactions), everybody gains.

		EFFECTS		
Areas and their options	on Area A	on Area B	on Area C	on the company
AREA A				
(1)	+	-	-	-
(11)	a fra		+	+
(111)	+	-	_	-
(1/)	-	+	+	+
(۷)	-	+	+	+
(vi)			+	-
AREA B				
CID	-	+		-
(11)	+-	+		+
CIII)	+	-	+	+
CIVD	+	-	+	+
AREA C				
CID	+	+	+	+ + +
(11)	_	-	+	
CIIID	_	-	+	-
(17)	+	+	-	+
(۷)	+		_	-
Net effect of adopted options	+ + + +	+ +	++++	+ + + + + + + + + + + + + + + + + + + +

Figure A6

Further, in the improved environment, options which previously never saw the light of day are now considered. These are options that are locally disadvantageous to the area which can adopt them but which give benefit to other areas. Amongst this greater range of options, again the ones to be adopted or not adopted are now chosen according respectively to wether they are or are not of net benefit to the whole

the state is the second state of the second st

company. The bottom line results speak for themselves.

INTEGRATING FUNCTIONAL AREAS

Based on the article "Integrating Functional Areas for Improved Productivity and Quality" [85].

Figure A7 shows the existing degree of coordination between the different departments of a factory.

The sector and a sector

and the second of the second of the

and a feat the state of the state of



- PP = Production Planning
- PS = Production Styling
- PE = Production Engineering
- ME = Manufacturing Engineering
- MK = Marketing
- SE = Service Engineering
- PD = Product design
- MF = Manufacturing
- SS = Sourcing and Suppliers
- MR = Market Research

DESIGN AND PRODUCTION CONFLICT

In recent years the problem of conflict or separation between the product design and product engineering function, on the one hand, and manufacturing and manufacturing-process engineering, on the other hand, has become an issue of serious concern. Tuttle [86] refers to this conflict as the "wall" between product design engineering and manufacturing engineering.

There is, unfortunately, a fundamental conflict of objectives, which becomes clear when the following is taken into account [87] :

• Every designer would like to achieve a functionally superior and therefore often complex product.

S. Web Strand Star

and a second sec

◆ The production engineer will take fewer risks in mass production runs by using easily controllable materials and processes, so he wants to manufacture products from components which are not too numerous and complicated and which involve the simplest possible assembly processes. At the forefront of the designer's considerations is the comprehensive functional performance of the total system and all its components.

The avoidance or minimisation of the negative aspects of this dualism will lead to new solutions through cooperation alongside the introduction of modern information processing technology (CIM) and other organisational measures such as communications engineers forcing through the new "watchword" of integration.

MARKETING AND MANUFACTURING CONFLICT

Shapiro [88] was one of the early authors to question if marketing and manufacturing could co-exist. He cited situations where lack of co-operation between the two functions could lead to open warfare. It is often seen in the form of incorrect forecasting and inventory planning and a general atmosphere of mistrust that results

in less than optimal performance of the company as a whole. Also other articles refer to the same problem [89,90].

Especially dangerous is the situation where either one or the other becomes dominant. If marketing dominates, the firm can become so sales-minded that manufacturing can not operate effectively. Alternatively, the firm can become so manufacturing-oriented that the needs of the customers are forgotten in the name of smooth operations. Top management must be alert to avoid conflict developing.

In order to understand sources of this conflict, motivations of the marketing and manufacturing functions must be understood. Kotler [91] has described marketing as "the analyzing, organizing, planning and controlling of the firm's customer-impinging resources, policies, and activities with a view to satisfying the needs and wants of chosen customer groups at a profit." Thus marketing is primarily concerned with forces outside of the firm. Manufacturing's domain, however, is concerned with a different set of goals, namely the on-time production of required products, the meeting of cost objectives, and maintenance of expected quality standards and targeted customer services levels [92].

See. S

Figure A8 summarises the sources of conflict between marketing and manufacturing.

Source	Example
Role orientation	Background of individual
	Performance evaluation criteria
Information between	Customer demand as seen by
firm and environment	marketing
Information within firm	Forecasts from marketing to
	production
	Build schedules from
	production to marketing
Product flow to environment	Loading or shortages in
	distribution channel
Product flow within firm	Material shortages
	Expedites or cancellations

Figure A8

149

APPENDIX B

,此"数据的学校"的变形,"这个","关系"的是"公路"的"公路"的"公司","帮助"的问题,"你们"的"资格"的"管路"的"管路"。

MA LOPE IS I MERE TO BE IS IN THE PARTY IS NOT

WORKSHOP OF MANAGEMENT SIMULATIONS

Computers in Teaching Initiative (CTI) The Management Centre, University of Bradford

WORKSHOP OF MANAGEMENT SIMULATIONS

INTRODUCTION

This workshop of Management Simulations was held on Wednesday the 6th of February 1991 at the Management Centre, University of Bradford.

The attendance to this one day workshop was possible due to the invitation received from Gillian Holmes, at the Management Centre, who is the CTI Information Officer.

The Computers into Teaching in Teaching Initiative (CTI) aims to promote a greater awareness of the potential of information technology (IT) to enhance the teaching and learning process, and to facilitate the development of skills in the use of computers in the teaching process.

Twenty one discipline-based centres have been established with the aim of assisting teachers to identify, acquire and use appropriate IT resources, course-ware and software.

The Centre for Accountancy has, in 1990, set up five regional centres, each with specialist interests. The northern regional centre for Business and Management studies is based at the Management Centre, University of Bradford, and will seek to :

• Collect and disseminate information to teachers throughout the northern region, on the use of computers in education, and materials available.

Provide a base for the coordination of activity and information exchange among IT users in the region, including the organisation of demonstrations and workshops.

• Liaise with institutions and IT users in the region to establish a database of available software/course-ware and to publish evaluations of software/course-ware.

WORKSHOP

The workshop was based on a display of four different Management Simulations (Simulation/Games). These simulations were Network Proteus, Topaz, Executive and Bissim, which were considered by the organisers as four of the best simulation/games available now at the market place.

The number of people who attended the workshop was around twenty, their most common position was lecturer or senior lecturer in business, at the Bradford Management Centre, and Polytechnic and Colleges around Bradford.

and the state of the state of the

NETWORK PROTEUS

Forewords

The business simulation "PROTEUS" was developed at Manchester Business School about six years ago (1985). That first release was a mainframe version sponsored by IBM, because its interest in developing new software applications for its existing hardware.

The piece of software written turned to be non transportable. Based on that disadvantage and after six years of running experience at the school, they start a new project called *Project Network Proteus*.

The project Network Proteus (founded by the Training Agency, under the direction of Dr Syel Howell) was started at the very beginning of 1991, and is expected to be finished by January 1992. The objective of the project is to improve and transfer the mainframe version of Proteus to a rewritten network version. That network version is due to be available by January 1992. The network needed is a local area network of PCs with 640K of RAM.

The main architect for this project is Pat White from the Manchester Business School, and formerly IBM employee. He was the one who presented the Network Proteus at the Bradford workshop.

Technicalities

Network Proteus was described as a qualitative simulation rather than a quantitative one. The simulation is not numerically oriented.

That characteristic made the designers decided that the programming language to be used had to be more data oriented, rather than numerical. That fact together with the need to work in a multi-user environment, meant that the language program to be used had to be a Database Management System (DBMS).

Among the different DBMS software, they decided to use PARADOX from Borland. They select this software because it was thought to be most user friendly system.

The hardware requirements were a network of PCs (hard disk and 640K RAM) could be used. The PCs had to be connected to a file server, with at least four Megabytes per playing group (data files). The program was meant to run under PARADOX supported by a spreadsheet for calculations. The expected sale price is $\pounds1,000$.

Context

The simulation is based on a rainwear factory, manufacturing umbrellas. Each team (factory) is formed by five people (positions) that have a different role in the business. These five roles are :

- Marketing
- Production
- Purchasing
- Finance
- Manager Director

Although there can be different teams (factories) working at the same time, it is not suppose to be a competition game, where teams compete against each other.

The simulation was described as a total business game capable of making everybody know everything. It is thought to be a game where people get involved all over the factory, seeking for collaboration inside the team.

Apart from the teams and their members, there is also another key element in

Network Proteus, the Tutor.

The Tutor has two main tasks :

- Generator of the external world.
- Analyzer of the teams' results.

The Tutor generates the external world by sending different messages to the players. Those notes are related to actual running problems of the factory, and concern areas like quality, stock levels, etc. Apart from standard messages (contained in a library), the Tutor can create and add his own ones. The Tutor can also decide who is he sending the notes to, giving him the option to mess up well performing teams if wanted.

Further developments could include a expert system to generate the Tutor's notes.

Simulation Description

The simulation is based on teams of five people performing five different factory's roles. Each of the roles is simulated in one computer of the network and therefore, the roles are interconnected.

The interaction between roles come from two possible sources :

- Access to different roles' data.
- MAIL system.

The MAIL system is based on a directory of names and addresses of the people may be contacted, to ask, complain, inform,.., about a certain issue related to the run of the factory. Thus, the interaction works by sending and answering to each role's mail. Although there are five roles in the game, only two of them were described, Production and Marketing.

Production Management System (Production Director)

The player/s dealing with this role have to make decisions concerning the production area. Their task is to define the production plan and the operations scheduling.

The production plan is based on sales expectations given by the Marketing function. After defining the plan, the parts requirements will be calculated based on the existing BOM (Bill of Materials). This calculation is carried out by a MRP system, not very accurate nor complicated but convenient for the purposes of the simulation. At this stage is possible to use the existing spreadsheet.

This parts requirement will give information about the need of material, as well as resources loads. Information, that may make lead to amend the production plan.

The production director will also be able to introduce new manufacturing products in the manufacturing process. It will take three simulation months for those new products to enter the manufacturing process. In the mean time, he will also have to deal with all the internal problems (MAIL) as well as with the external ones (TUTOR'S notes). the second s

Marketing

Due to the nature of the business being simulated, rainwear company, the role of the Marketing function in this factory is very characteristic. The marketing demand is of a seasonal nature.

The marketing people will have to decide about the expecting sales, forecasts (long

and short range), competition, prices, advertising, discount rates,...To support these decisions the Marketing people can access to information about the weather, list of customers and their acquisitions, etc.

Players on this role will also have to decide about the sales strategy, deciding whether they are selling through agents or their own sale force.

Functioning Characteristics

The old version of PROTEUS was used for MBA, 2nd year Managers and Postgraduates students. The new version (Network Proteus) targets the same population.

Each run cycle is one simulation month (i.e. four week period time). The time needed to run six cycles (six simulation months) is three real full-time working weeks.

PAT WHITE'S EXPLANATIONS

** ***

and the second a second of

5. 2

Acres - Star - 1 the 25 and the

Michigan Bar

UNIVERSITY OF BRADFORD MANAGEMENT CENTRE EMM LANE BRADFORD WEST YORKSHIRE BD9 4JL UK



Director and Professor of Management DAVID T H WEIR MA DipPSA FBIM

L UK 25 (0274) 542299 Telex 51309 UNIBED G Fax (0274) 546866 E-mail ghictiaUK.AC,BRADFORD

COMPUTERS IN TEACHING INITIATIVE CENTRE FOR BUSINESS AND MANAGEMENT STUDIES Coordinator: RICHARD WELFORD BA MA PhD PGCE ABIM

14 February 1991

Mr J I Igartua Nottingham Polytechnic

Dear Mr Igartua

I hope you enjoyed the workshop on management simulations last week, despite the cold weather! I enclose some information on Network Proteus which Pat White has sent for distribution to you. Pat has undertaken to feed information on future developments to me for wider distribution - so watch this space! If you have any further queries on the other simulations we saw, please don't hesitate to get in touch with me, or directly with the suppliers.

Yours sincerely

lif Ettobuls

Gillian Holmes

Sponsored by the Institute of Chartered Accountants in England and Wales and the Computer Board for Universities and Research Councils

The Network Proteus Project

In 1985 the IBM company awarded a very substantial grant (of hardware, software and money for the employment of development staff) to the Manchester Business School. In addition two senior IBM staff were seconded to the School to support the project for its three year duration.

A key element in MBS gaining the award in competition with many other business schools was a proposal to build a management simulation of a new kind for use in the MBA programme. Like most computer projects which break new ground, the simulation project, called Proteus after the god who could assume many forms, identified a number of problems. Even so, it was clear that the novel features of the simulation added considerable value, and it is still in regular use at the IBM project partner sites, the business schools at Aston, Strathclyde and Warwick Universities.

The Proteus project was designed to make full use of the facilities provided by the IBM award, specifically the 4381 mainframe, the VM operating system, and the PROFS, AS and SQL software packages. While this contributed to the richness of the simulation, it has been a major obstacle to its wider use.

In the summer of 1989, MBS responded to an initiative by the Training Agency of the Department of Employment by submitting a proposal for rebuilding and extending the original Proteus simulation in a new, more widely accessible environment based on a network of personal computers. This proposal was accepted and a contract for a two year project, christened Network Proteus to reflect its predecessor and the new environment, was signed in January 1990. The contract is between the Training agency and the University of Manchester, is based at MBS, and has the active support of the original partner sites. The training branch of the National Health Service is also showing an interest which is also likely to lead to collaboration.

Network Proteus, like the mainframe-based development, contains features absent from the vast majority of management simulations currently in use. When playing a typical business game, the participants are presented with a limited selection of quantitative data for a hypothetical company, balance sheet, profit and loss account, sales history, etc., and possibly additional figures about the size of the market and the competition. The players, usually operating in teams, make a standard set of decisions about allocation of resources, which are fed into the model to generate a new set of data for the next operating period, the objective being to finish with a set of figures superior to the other teams. If the teams are competing with each other in the same market, the simulation takes on all the attributes of a zero-sum game. If they are not carefully controlled, these simulations can degenerate into an attempt to beat the algorithm.

Both the earlier and the current Proteus projects provide for larger and more realistic flows of numeric data, but the range and richness of the simulation is also substantially extended by the introduction of qualitative information in the form of files of memos, letters and commercial intelligence clippings which must receive attention. For example, in one of the Proteus models being developed, the simulation of a rainwear manufacturing company, the management team is faced with

a letter complaining about quality from the chief buyer of a major customer

an insurance claim for damage caused by one of the company's delivery vehicles

persistent bad timekeeping by the best worker in the general office

queues around the recently introduced vending machines

an adverse movement in the exchange rate with a country from which several products are imported

" a the last a the start a sta

In addition they have to prepare for a forthcoming ministerial visit. Failure to cope with these issues can have unfortunate consequences. If, for example, the complaint about quality is ignored, the business will be lost leaving the team with a mountain of unsaleable stock. While coping with these qualitative problems, they still have to manage the numbers. In contrast to existing games, no decision need be taken. The team doesn't have to order raw materials, nor change production schedules. The results of not doing so may leave a lot to be desired!

A second key feature of Network Proteus is the access to information. In a conventional game the entire team is usually in possession of all the data available. In a Network Proteus model, each team member can see the data relevant to his or her own role. Thus, the marketing director can check on advertising expenditure but has no access to detailed production schedules. If he needs data from them, he must enlist the aid of his production director who, as in real life, has his own problems to contend with. Also if team members call for the same information they may get what appear to be inconsistent answers. A request for last month's sales figures from the marketing director will probably produce totals of orders taken. The same request from the production director would be satisfied by shipment figures, while the finance director could well get goods invoiced less returns credited.

Reference was made earlier to the volume of numeric data in a Network Proteus model. The project attempts to copy the real world by providing substantial amounts of detail, supported by tools for interrogation and data reduction. As an example, in the rainwear model, the players have access to a file containing all the orders received in the last two years of simulated operations, upwards of 1,000 entries. Information about the level of customer returns is part of this file. Accurate analysis of the data in the file will show that the returns rate for the complaining customer is an order of magnitude greater for the product which is the subject of the complaint than for any other customer/product combination. In this way the project should achieve a secondary objective, introducing students to some of the tools and techniques available through information technology to support management decisions.

The earlier Proteus project did not fully justify its name, since it provided only one scenario, usable in two ways. From the outset Network Proteus has been designed to assume many forms. To achieve this, a simple modelling language has been defined and a shell to interpret it has been written. The language will allow us to simulate a wide variety of business situations. These may be team exercises or case studies for analysis by individuals. The project plan has allocated resources for the production of two major team based simulations. One is the rainwear manufacturing example referred to above. The other, in complete contrast, will simulate the running of an NHS hospital under the recently introduced Resource Management Initiative. At present, the RMI is being piloted in a small number of hospitals, one of which is associated with the project. A number of individual case studies will also be produced. The first of these, already under development, requires the student to analyse the inventory problems of a personal computer dealer.

and a manual is an and the second second and so at a second second and a second second second second second sec

where with suiters .

Network Proteus models will, wherever possible, employ commercially available PC software. The two key areas will be a data base package which provides flexible and easy to use storage, retrieval and manipulation facilities, and a spreadsheet to reduce the effort involved in handling numeric data. Another possibility being examined is the use of an expert system to assist in the control of the more complex models by the tutor.

The project plan calls for the delivered product to include

the model definition language and the shell to interpret it

the two major simulations described above

a number of supporting case studies

PC based tutorials to introduce the Network Proteus environment and the commercial packages used

tools to enable Proteus users to develop their own models and case studies

In addition, although there is no contractual obligation to do so, we hope to include data capture facilities so that Network Proteus can be used as a tool for research into the learning process and the behaviour of teams using computer networks. The anticipated completion data is December 1991.

The project is directed for the University by Dr S D Howell, Lecturer in Management Accounting and Control at MBS. Computer development is managed by Mr P L White, one of the secondees from IBM to the earlier project, since retired from IBM and appointed to an honorary fellowship at MBS.

P L White 5.2.91

TOPAZ

Forewords

Topaz is one of a range of management games produced by EDIT 515 Ltd. Topaz simulates major company functions enabling between 12 and 48 players divided in teams which compete against each other. An elaboration of TOPAZ is the base for "The Scotsman Management Game" run by "The Scotsman" newspaper.

EDIT 515 Ltd. has been developing games for 20 years, being a very experienced company dealing with this sort of games.

Technicalities

TOPAZ is a game developed in FORTRAN, running in a stand alone machine. Its price is £2500 with an academic discount of 15%.

Game's description

TOPAZ is a management game where 3 or 8 teams compete against each other. The teams are formed by 3 to 6 people, each of them concerned with a different role. The game has to be played for a minimum of 5 plays and a maximum of 10.

The areas within the game are : Marketing, Finance, Personnel, Production Planning, R&D, and Purchasing, although the speaker, Bill Robertson, insisted in the importance of the existing conflict between Marketing and Production Planning. He also insisted in the general interaction that is encouraged through the run of the game.

After a discussion of one hour, each team have to take some decisions concerning different areas of the factory. Those decisions are filled in the input form and will

afterwards input in the computer. The computer after computing those inputs (30 to 45 minutes), will display some numeric results.

Game's elements :

Product Description

The factory that is being gamed has three different final products with an unknown bill of materials (BOM). They are three black boxes that could mean anything.

Production Process

The production process of the factory being gamed is based on an assembly shop an a machine shop. The assembly shop is where the product is assembled and the machine shop is where the parts for that assembly process are machined.

As there is no BOM (Bill of Materials), only final products, the times considered by the game for machining and assemble are some fixed predetermined ones. The level of quality of the final products is directly proportional to the assembly time spent.

The production process defined for the game, assumes that there is no WIP (Work in Progress), that means that if there are not any final products to be produced, there will not be any parts being produced either.

About the shifts, the machine shop can be run with 1,2, or 3 shifts. On the other the assembly shop is only run with one shift, with a limited number of men and overtime available.

The game, also, allows the possibility to by machines, that will only be possible to be bought if the financial state of the factory is satisfactory. The machine will start working at a deferred time.

Marketing

The decisions to be taken in the Marketing area are related to sale price, credits given, design (fashion, technical advances,..), quality (warranties), deliveries, competitiveness,..

They also will have to decide about advertising matters, having to choose from three different kinds of advertising.

As it was explained in the introduction, the game is designed to run under the conflict between Production and Marketing, and therefore they will have to reach an agreement by adjusting their different objectives. The other departments role will be to do what comes from the adjustments agreed between Marketing and Production Planning.

Research and Development (R&D)

There is a possibility to introduce product improvements but of coarse at a certain expense. That expense is a cumulative and therefore, does not always mean that because you spend money you are going to get the improvement. The research and development can turn out to be none, minor, or major.

That improvement status is controlled by random numbers :

IF random number > cumulative money spent THEN improvement (minor or major)

Personnel

Their task is to keep people working in the factory happily, trying to avoid negative actions that could influence the work of the factory.

The happiness of the workforce and the factors that influence it are graphically shown

in figure B1.



Finance

The factory being gamed also has a finance department, where decisions about the finance state of the factory are discussed.

a not all all a lot

And the second second

Objective of the game

The objective of running the game could be explained as :

"To get, from among the different teams playing, the highest share price at the end of the game."

That share price value depends basically on the Net Assets per share, although it is also affected by the following long-term parameters :

- Dividend Performance.
- Liquidity.
- Current Market Share.
- Production Resources.
- Research and Development Effort

• Stock and Backlog position.

All these factors imply that the success of the gamed factory will be achieved by balancing the customer service performance and the factory's profitability (Sales Revenue, Cost, Overhead, Miscellaneous).

Game's cycle

The game's cycle is described in figure B2.



The forms used for playing TOPAZ are shown in the coming pages.

TOPAZ FORMS

•										
ſ	GAME HISTORY GAME HISTORY GAME HISTORY GAME HISTORY GAME HISTORY	- 'THE :	SCOTSMAN' MANAGE CREATED AND ADMINISTERED BY	BBIT	- 1989	(Round One)	THI GAN YEI	S IS THE FIR E HISTORY FO R 1988, QTR	ST PART R ROUND . 1	OF THE ONE.
l	-	- GROUP 1	COMPANY 1	IDENTITY No	L Y	EAR 1988	QUARTER 1			
5	PEROPT BASED ON DECISION	B BELOW	PLAN	T/MATERIALS	/PERSONNE	1		RODUCTS/ORDEF	S/STOCKS	S L tridburt 1
TINKE I	CHECK THEM BEFORE GOING	FURTHER.	MACHINES AVAILAB	LE LAST QUARTER		13	BEOUESIED	PRODUCT1	2045	1 1165*
CULTS I		PRODUCT I PRODUCT 2 PRODUCT 3	MACHINES AVALAD	CE NEAT QUARTER		13	PRODUCED	5564	3139	1202
	MAJOR IMPROVEMENTS TAKEN UP	0 0 1	VANS AVAILABLE LA	AST QUARTER		21	SERVICED	1 164	94 50	37
8	PRICES EXPORT AREA	118 195 390	SKILLED MAN HOL	IAS:			DELIVERED TO	PRODUCT 1	PRODUCT	Z PRODUCT
6	HOME AREA	108 183 378	ABSENTEEISM	UUANTER		108	EXPORT	. 4250	2200	800
	PROMOTION IN TRADE FRESS	6 6 10	USED LAST OUAR	TER WEEKS NEXT OU	AR168	26169	SOUTH	300	250	100
	MERCHANDISING	4 4 5					NORTH	700	475	180
	FITTING TIMES (MINS)	115 170 330	AVAILABLE LAST	OVARTER		14196	OPDERS TROM	PRODUCT 1	PRODUCT	2 PRODUCT 3
1		5 2 W 1 N A	I BREAKDOWN TIM	E		39	EXPORT SOUTH	3094	1737	640
		32 11 11	USED LAST QUAR	TCR		12205	WEST	101	94	70
	SALESMAN'S SALARIES 35 SKILLED WAGE RATE 440	SHIFT LEVEL 2	MACHINE EFFICIE	NCY (%)		97.5	NORTH	595	364	145
8	MANAGEMENT BUDGET 75		MATERIAL UNITS			1844	L EXECRI	3530	1881	2 Photoci,
SMIS	MAINTENANCE HOURS 60	MACHINES SOLD	DELIVERED			15000	SOUTH	292	230	100
ABA	DIVIDEND 4	CREDIT PERIOD 30	USED CLOSING STOCK			15448	WEST NORTH	104	101 364	70
NY IS	INFORMATION ON COMPANIES	MARKET SHARES	. ON ORDER FOR N	EXT DUARTER		18000	BACKLOG OF DR	RAS PRODUCT 1	PRODUCT	2 PRODUCT
ALL	DELIVERIES REQUESTED TO	ODUCT 1 PRODUCT 2 PRODUCT 3		SALESME	SKILLED M	L UNSKILLED	EXPORT	0	0	6
ASON ASON	EXPORT 4	250 2200 800 300 250 100	AT START OF OTR	20	46	MEN 98	I WEST	a o	a a	19
S Int	FULL IF STARRED WEST	120 85	RECAUITED	1	0	6	I NORTH	0	0	0
Day	NOATH '	100 475 180	TRAINED	ő	0	U	PRODUCTS \$10C	SIN PRODUCT 1	PRODUCT	2 PRODUCT:
	RESEARCH EXPENDITURE	10 12 10	LEFT AVAIL FOR NEXT (0 018 21	0 46	6 98	EXPORT SOUTH	720	319 20	0
	SALESMAN HIRED 1 DIS SKILLED MEN HIRED 0 DIS	SMISSED 0 TRAINED 0 SMISSED 0 TRAINED 0					WEST NORTH	46 60	19 119	15 34
	MATERIALS ORDERED 1800 FROM SUPPLIER MACHINES ORDERED	DO 2 DELIVEPIES 8 0					FRODUCI INFROVENENTS	MINOR	PRODUCT HINOR	2 PRODUCT: NONE.
CONTAIN MANUAL	COY.INFO. ADVERT. RESEARCI EXPEND. EXPEND. COMPANY 1 66000 32000 COMPANY 3 66000 32000 COMPANY 3 66000 32000 COMPANY 4 66000 32000 COMPANY 5 66000 32000 COMPANY 6 66000 32000 COMPANY 7 66000 32000 COMPANY 8 66000 32000	H CONSUMER ASSESS	IENT RATINGS	PROD EXP. S. 2.7 12.5 2.7 12.5 2.7 12.5 2.7 12.5 2.7 12.5 2.7 12.5 2.7 12.5 2.7 12.5 2.7 12.5 2.7 12.5	UCT 1 W. 12.5 12.5 12.5 12.5 12.5 12.5 12.5 12.	N. EXP 12.5 2.4 12.5 2.4 12.5 2.4 12.5 2.4 12.5 2.4 12.5 2.4 12.5 2.4 12.5 2.4 12.5 2.4 12.5 2.4 12.5 2.4 12.5 2.4 12.5 2.4	PRODUCT 2 S. W. 12.5	N. EXP. 12.5 2.2 12.5 2.2	PRODUC S. 12.5 12.5 12.5 12.5 12.5 12.5 12.5 12.	T 3 W. 12.5 12. 12.5 12. 12. 12.5 12. 12. 12.5 12. 12. 12.5 12. 12. 12.5 12. 12. 12.5 12. 12. 12.5 12. 12. 12.5 12. 12. 12.5 12. 12.
	GROSS NAT. PRODUCT (SEASONA)	LLY ADJ) 718(SM)	UNEMPLOYED I	N S.AREA(S	EASONALL	Y ADJ) 6%	CENTRAL B	ANK RATE FOR	NEXT QI	rR. 84
DUE ST		PRICES PRODUCT 2 PRODUCT 3 E H E H	TOTAL SKIL EMPLOYED WAGE	LED RATE IN	COMPANY	PRODUCT 1 E H	PRICES PRODUCT 2 E H	PRODUCT 3	TOTAL	SKILLED WAGE RAT
50	COMPANY 1 + 118 108 19	95 183 390 378 95 183 390 378	165 4 165 4	40 . 00	MPANY 2 MPANY 4	118 108 118 108	195 183 195 183	390 378 390 378	165 165	440
Soo a	COMPANY 5 118 108 1	95 183 390 378	165 4	40 · CC	MPANY 6	118 108	195 183	390 378	165	440
ALL DATE	COMPANY / 118 108 / 19	92 183 340 348	4 601	40 CC	MPANI 6	118 100	195 165	390 378	105	440 -
Sup Sup	OVERHEADS	PROI	TT & LOSS ACCOUNT			BALANCE SHEE	I	CA	SH FLOW	3
EXAN	PROMOTION & ADVERTISING 66	000 SALES	14	66651	ASSETS		200000	TRADING RECEIPTS		1442862
943	SALES OFFICE 122	998 OPENING STOCK VA	LUE	40665	MACHINES		1247050	INTEREST RECEIVED		0
	GUARANTEE SERVICING 9 TRANSPORT FLEET 121	430 MATERIALS PURCHA 388 SKILLED WAGES	SED 4	74352 36778	PRODUCT S	TOCKS	143302 79675	ADDITIONAL LOANS		0
	HIRED TRANSPORT 21	872 UNSKILLED WAGES	1	98633	MATERIAL	STOCKS	25902	TRADUC DAVIAGUTE		1304005 1
	RECAUITMENT & TRAINING 4	000 LESS CLOSING S100	CK VALUE 1	05577	CASH INVE	STED	0	CAPITAL PAYMENTS		0.2
	MAINTENANCE 31.	200 COST OF SALES 258	8	21476	LABILITIES			INTEREST PAID	т	2630
	BUSINESS INFELLIGENCE	500 GROSS PROFIT/LOS	s 6	45175	TAX ASSES	SED	122116	TAX PAID		1747
	CREDIT CONTROL 6	193 INTEREST RECEIVED		0	BANK OVER	DRAFT	88400	DIVIDENDS PAID		44000
B	OTHER COSTS 7	299 INTEREST PAID OVERHEADS	5	2630 21136	UNSECURE	DLOANS	0	Auroanner 111		
8	TOTAL OVERHEADS 521	136 DEPRECIATION		41529			1401205	FOR NEXT QUARTER		369000-
A 101		TAX ASSESSED		Ŭ	NET ASSETS		1401325			
385		NET PROFIT/LOSS		79880	ORDINARY CA	APITAL	1100000	PRICE OF MATERIAL		
Com	TAXABLE PROFIT/LOSS ACCUMULATED 79	880 DIVIDENDS PAID	FEBVER	44000	TOTAL SHAR	ENOLDERS FUNDS	1401325	ORDERED FOR NEXT QUARTER (1000 UNIT	51	37934
2	******	IRANSTERNED TO F	ESERVES	1000						76
		DANY I CONDANY 9 CO	OUDANY 2 CONDARY	A CONDANS	5 COMDE	NV 6 COMPAN	Y 7 COMPANY F			

...

,

ないのないのないたか
GAME HISTORY GAME HISTORY GAME HISTORY -	GROUP	CREATED AND ADMINISTERED B			1988 AR 1988	GAME YEAR QUARTER 21	HISTORY. 1988, QTR	2	
REPORT BASED ON DECIS	IONS BELOW,		PLANT/MATERIA	LS/PERSONNEL		PR	ODUCTS/ORDE	AS/STOCKS	I Dr
CHECK THEM BEFORE GOI	NG FURTHER.	MACHINES A	VAILABLE LAST OUAR	TER	13	PLOUESTED	1 3975	2300	,
	PRODUCT 1 PRODUCT 2 FROD	ANCHINES A	VALDELE REXT GOAN		21	PRODUCED	4096	2371	1
MAJOR IMPROVEMENTS TAKEN UP	U U		ISLE LAST QUARTER			SERVICED	98	58	
PRICES EXPORT AREA	118 195 40	0 SKILLED MA	N HOURS:		27048	DELIVERED TO	PRODUCT 1	PRODUCT 2	IP
	6 6		ISM		113 20727	EXPORT	275	1700 200	
ADV. SUPPORT	8 8 1	0 NOTICE OF	STRIKE WEEKS NEXT	OUARIER	0	WEST	100	100	
MERCHANDISING	4 4	MACHINE	OURS:			DROFES FORM	PRODUCT	PRODUCT 2	2 P
FITTING TIMES (MINS)	115 170 35	0 AVAILABLE BREAKDON	LAST OUARTER		30	EXPORT	3300	1832	-
SALESMEN ALLOCATED TO EXPORT	.3 s 3 w 1 N	4 MAINTENA	NCE FOUARIER		750 9448	SOUTH	1 278	223	
SALESMAN'S SALARIES	35 COMMISSION	1 MACHINE	EFFICIENCY (%)		97.1	NOHTH	639	393	
SKILLED WAGE RATE MANAGEMENT BUDGET	75	I MATERIAL	INITS.		1306	SOLD 10	PRODUCT 1	FRODUCT2	I P
MAINTENANCE HOURS	60 MACHINES SOLD	0 OPENING S	TOCK		18000	SOUTH	278	220	
DIMDEND	0 CREDIT PERIOD 3 0 VANS SOLD	0 USED 0 CLOSING S	TOCK		7390	WEST	639	393	
NFORMATION ON COMPANIES	0 MARKET SHARES		FOR NEXT QUARTER		16000 23390	BACKLOG OF DEDLES	PRODUCT 1	PRODUCT	2 P
DELIVERIES REQUESTED TO.	PRODUCT 1 PRODUCT 2 PROD	<u>ver 3</u>]	SALES	MEN (SKILLED ME!	N UNSKILLED	EXPORT	0	0	
NOT DELIVERED IN SOUTH	275 200 1	O AT START	OF QTA 2	1 46	98	WEST	0	0	
HOLL D STARRED WEST	600 300 1	0 i RECRUITER	2	0 0	0	NORTH		L BOODUCT	
ESTABLE EXPENDIFIEF	10 12 2	0 LEFT		1 1 0 0	6	1 EXPORT	420	187	
CALIFORNIA (1997)		1 AVAIL FOR	NEXTOTA 2	2 47	98	SOUTH	5 36	0 17	
SKILLED MEN HIRED	DISMISSED 0 TRAINED	1				NORTH	21	26	
MATERIALS ORDERED	16000					PRODUCT	PRODUCTI	PRODUCT	2 P
FROM SUPPLIER MACHINES ORDERED	C DELIVERIES							1 PASOR	
COMPANY PRODUCT (SEAS	PRICES	TOTAL	SKILLED	(SEASONALLY	PRODUCT 1	PRICES	K RATE FOR	NEXT QT	R.
COMPANY 1 118 108	195 183 400 3	R EMPLOYED	440 I	COMPANY 2	E H 118 108	<u>Е Н</u> 195 183 4	E H	167	WA
COMPANY 3 118 108	195 183 400 3	167	44D	COMPANY 4	118 108	195 183 4	00 383	167	1
COMPANY 7 118 108	195 183 400 3	167	440	COMPANY 8	118 108	195 183 4	00 383	167	2
OVERHEADS		PROFIT & LOSS ACCO			BALANCE SHEE	T	CA	SH FLOW	
PROMOTION & ADVERTISING	60000 SALES		1390259	ASSETS:		T	ADING RECEIPTS		1410
SALESMEN'S SALARIES ETC SALES OFFICE	129346 13846 L OPENING STOL	X VALUE	105577	PROPERTY MACHINES		200000 · c. 1215874 ·	APITAL RECEIPTS		
GUARANTEE SERVICING	11300 + MATERIALS PL	RCHASED	581990	VANS	0000	134346 IN	VESTMENTS SOLD		3
HIRED TRANSPORT	0 : I UNSKILLED W	\GES	158634	MATERIAL SI	TOCKS	140166	DOMUNAL LUANS		122
PRODUCT RESEARCH RECRUITMENT & TRAINING	10000 MACHINE RUN	NING STOCK VALUE	187946	CASH INVES	TED	0 0	RADING PAYMENTS APITAL PAYMENTS		1321
MAINTENANCE WAREHOUSING & PURCHASING	31200 COST OF SALE 8608	s	816991	I HARBING		19	TEREST PAID	н	:
BUSINESS INTELLIGENCE	0 GROSS PROF	1/LOSS	573268	TAX ASSESS	ED	688198	UX PAID		123
MANAGEMENT CREDIT CONTROL	5910 INTEREST REC	EIVED	0	BANK OVER	DRAFT	124119 0	IVIDENDS PAID		
OTHER COSTS	7248 INTEREST PAIL OVERHEADS	2	500790	UNSECUTED	LOANS	0	VERDRAFT I IMIT		
TOTAL OVERHEADS	500790 DEPRECIATION	r D	40132	NET ASSETS		1430532	DR NEXT QUARTER		40
		-	29207	0000000000	DITA	1100000			
	NET PROFILE	UBB		UNUINANY CAP	-TI ALL	330532 P	RICE OF MATERIAL		
TAXABLE PROFIT/LOSS	100007		•	RESERVES		1430533 0	ROERED FOR NEXT		2

171

and the second second

GAME HISTORY GAME HISTORY GAME HISTORY	L_	GROUP 1	CREATED AND ADMINISTERED BY COMPANY 1	IDENTITY.	1 9 B Y No 1	YEAR 198	9 88 a		EAR 1988, QT	r. 3	Ur
REPORT BASED ON DEC	ISIONS BELOW	N.		PLANT/MATE	ERIALS/PERSO	ONNEL		FROCUCT O	PRODUCTS/OADI	PRODUCT 2	I P
CHECK THEM BEFORE G	OING PURTHEF	R.	MACHINES AV	AILABLE LAST O	UARTER	1	13	REQUESTED	3960	1 2320	
	PRODUCT 1 PR	ODUCT 2 PRODUCT3						PRODUCED	4078	2392	
MAJOR IMPROVEMENTS TAKEN UP	1 0	0 0	VANS AVAILAE	LE LAST UDAN	ich .	·	21	SERVICED	1 93	1 55	
PRICES EXPORT AREA	115	195 400	SKILLED MAN	AST OUARTER		276	36	DELIVERED	10 PRODUCT 1	PRODUCT 2	1 P
The rest			ABSENTEEIS	м		1	25	EXPORT	3000	1650	
PROMOTION IN TRADE PRESS ADV. SUPPORT	1 10	6 10	NOTICE OF 5	TRIKE WEEKS N	EXT QUARTER	203	0	WEST	100	100	
MERCHANDISING	4	4 4	MACHINE HO	URS				NORTH	1 600	350	
FITTING TIMES (MINS)	1 117 !	170 350	AVAILABLE	LAST QUARTER		141	96	OPDERS FR	PRODUCT 1	PRODUCT	2 [P
SALESMEN ALLOCATED TO EXPORT	13 5 3 1	W 1 N 5	MAINTENAN	CE		7	54	SOUTH	261	210	
PAILERMAN'S SALADIES	35 1.0000	T MORE	USED LAST	DUARTER		92	67 1	NORTH	102	94 369	
SKILLED WAGE RATE	440 SHIFT L	EVEL 2	in the second contract of			57		\$0LD 10	PRODUCT 1	PRODUKT	P
MANAGEMENT BUDGET	80		OPENING ST	OCK		73	90	EXPORT	3285	1649	
MAINTENANCE HOURS	60 MACHIN 5 CREDIT	VES SOLD 0	DEUVERED			160 117	00 66	SOUTH WEST	261 102	211 94	
VANS BOUGHT	0 · VANS S	OLD 0	CLOSING ST	OCK	150	116	24	NORTH	597	369	
INFORMATION ON COMPANIES	PRODUCT I TH	DOUCT 2 PRODUCT 3	AVAILABLE	NEXT QUARTER		236	24	EACODD OF	DROLAS PRODUCT 1	PRODUCT	2 F
DELIVERIES REQUESTED TO EXPO	at 3000 (10	650 600 .	I PERSONNEL:	SA	LESMEN SKILLE	D MENI UNSK	CILLED	SOUTH	0	0	
NOT DELIVERED IN SOUT	H 260 I	220 125	AT START OF	FOTR	22	47 9	8	WEST NORTH	0	0	
NORT	H 600	350 150	DISMISSED		0	0	0	PPODUCIS ST	CAS H PRODUCT I	PRODUCT	2 1 1
RESEARCH EXPENDITURE	10 ;	10 12	LEFT		0	0	6	EXPORT	135	188	
SALESMAN HIRED 0	DISMISSED 0		AVAIL FOR*	NEXTOTR	22	47 9	8 1	I SOUTH	4	9 23	
	• • • • • •										
SKILLED MEN HIRED	DISMISSED 0	TRAINED 0						I NORTH	24	7	
SKILLED MEN HIRED D	12000	TRAINED 0						PRODUCT	24	7	2 7
SKILLED MEN HIRED D MATERIALS ORDERED FROM SUPPLICA MACHINES ORDERED	DISMISSED 0 12000 2 0	DELIVERIES 5						PRODUCT	24 PRODUCT 1 k15 MINOR	7 PRODUCT: NONE	2 4
I SKILLED MEN HIRED	DISMISSED 0 12000 2 0	DELIVERIES 5						PRODUCT MORTH	24 PRODUCT 1 k15 MINOR	7 PRODUCT: NONE	2 4
SKILLED MEN HIRED D MATERIALS ONDERIED FROM SUPPLICA MACHINES ORDERED GROSS NAT. PRODUCT (SEA	DISMISSED 0 12000 2 0 ASONALLY ADJ	TRAINED 0 DELIVENIES 5 728 (\$M)	UNEMPLOYE	D IN S.A	REA (SEASON	NALLY ADJ)	63	CENTRAL	24 PRODUCT 1 MINOR BANK RATE FO	7 PRODUCT: NONE R NEXT QT	2 4 ' 'R.
SKILED MEN HINED D MATERNALS ONDERED FROM SUPPLICA MACHINES ORDERED GROSS NAT. PRODUCT (SEC. COMPANY INFORMATION PRODUCT I	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRODUCT 2 E N	TRAINED O DELIVENES 5 728 (SM) PRODUCT 3 E H	UNEMPLOYE TOTAL EMPLOYED W	SKILLED	REA (SEASON COMPAN INFORMAT	NY PRO	DOUCT 1	PRICES PADDUCT 2 CENTRAL	24 PRODUCT 1 MINOR BANK RATE FO	7 PRODUCT: NONE R NEXT QT TOTAL EMPLOYED	R.
ISKLED MEN HIRED D MATERIALS ORDERED IFROM SUPPLYER MACHWIES ORDERED GROSS NAT. PRODUCT (SE. COMPANY INFORMATION E PRODUCT INFORMATION E 115 108	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRODUCT 2 E M 195 183	TRAINED O DELIVENIES 5 728 (SM) PRODUCT 3 E 400 383	UNEMPLOYE TOTAL EMPLOYED W 167	D IN S.AJ SKILLED VAGE RATE 440	REA (SEASON COMPAN INFORMAT	NY PACE NY PACE (2 115	000CT 1 H 5 108	PADUCT PADUCT CENTRAL PRICES PRODUCT 2 E H 195 183	24 PRODUCT1 MIS MINOR BANK RATE FO PRODUCT3 E NR 400 383	7 PRODUCT: NONE NONE R NEXT QT TOTAL EMPLOYED 167	R.
SKILLED MEN HIRED D MATERALS ONDERED IFAOM SUPPLY IFAOM SUPPLY GROSS NAT. PRODUCT (SEA COMPANY PRODUCT (SEA INFORMATION E COMPANY 115 108 COMPANY 3 115 108 COMPANY 5 115 108	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRODUCT 2 E N 195 103 195 103 195 183	TRAINED O DELIVENIES 5 728 (SM) PRODUCT 3 E 400 363 400 363 400 363	UNEMPLOYE TOTAL EMPLOYED W 167 167 167	SKILLED SKILLED VAGE RATE 440 440	COMPAN INFORMAT COMPANY COMPANY COMPANY	NALLY ADJ) NY PRO (2 115 (4 115 (6 115	0 61 DOUCT 1 H 5 108 5 108 5 108	PROUCE MANUEL CENTRAL PRICES PADUCE 2 E H 195 183 195 183 195 183	24 PRODUCT 1 MIS MINOR EANK RATE FO PRODUCT 3 E ANG 383 400 383 400 383	7 PRODUCT: NONE R NEXT QT TOTAL EMPLOYED 167 167 167	R.
SKILED MEN HIRED D MATERIALS ONDERED I IFAOM SUPPLICA MACHNIES ORDERED GROSS NAT.PRODUCT (SE. I COMPANY PRODUCT (SE. INFORMATION E COMPANY 1 115 108 COMPANY 5 115 108 COMPANY 7 115 108	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRICES PRICES PRICES 195 183 195 183	TRAINED 0 DELIVENES 5) 728 (SM) PRODUCT 3 400 363 400 363 400 383	UNEMPLOYE TOTAL EMPLOYED W 167 167 167 167	SKILLED AGE RATE 440 440 440 440	COMPAN INFORMAT COMPANY COMPANY COMPANY	NY PROV 10 PROV 2 115 2 115 2 4 115 2 4 115 2 6 115 2 8 115	0000CT } H 5 108 5 108 5 108 5 108	PROUEL PROUEL PROUEL PRICES PROUEL 2 E H 195 183 195 183 195 183	24 PRODUCT1 MIS MINOR BANK RATE FO PRODUCT 3 400 383 400 383 400 383	7 PRODUCT: NONE NONE R NEXT QT TOTAL EMPLOYED 167 167 167 167 167	R.
SKILED MEN HIRED D MATERIALS ONDERED FROM SUPPLICH MACHINES ORDERED GROSS NAT. PRODUCT (SE. COMPANY INFORMATION E COMPANY 1 115 108 COMPANY 3 115 108 COMPANY 5 115 108 COMPANY 7 115 108	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRIOCES PRIOCES PRIOCES 105 103 195 183 195 183	TRAINED 0 DELVENES 5) 728 (SM) PRODUCT 3 400 363 400 363 400 363 400 363 900 FIG	UNEMPLOYE TOTAL EMPLOYED V 167 167 167 167 167	D IN S.AI SKILLED VAGE RATE 440 440 440 440	COMPAN INFORMAT COMPANY COMPANY COMPANY	NY PRICE ADJ) NY PRICE 2 115 2 4 115 2 6 115 2 6 115 2 8 115 BALANC	64 0000CT 1 5 108 5 108 5 108 5 108 5 108 CE SHEET	CENTRAL PRODUCT 2 E PRICES PARCES PARCES PARCES PARCES PARCES PRI	24 PRODUCT1 MIS MINOR BANK RATE FO PRODUCT 3 PRODUCT 3 PRODUCT 3 PRODUCT 3 PRODUCT 3 PRODUCT 3 PRODUCT 3 PRODUCT 3 PRODUCT 1 PRODUCT 3 PRODUCT 3 P	7 PRODUCT: NONE NONE R NEXT QT TOTAL EMPLOYED 167 167 167 167 167	rR.
SKILLED MEN HIRED D MATERIALS ONDERED IFAOM SUPPLICH MACHINES ORDERED GROSS NAT. PRODUCT (SEL COMPANY PRODUCT (SEL INFORMATION E COMPANY 115 108 COMPANY 5 115 108 COMPANY 7 125 108 PROMOTION & ADVERTISING OVERHEADS	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRICES PROVICT H 195 103 195 183 195 183 195 183	TRAINED 0 DELVENES 5) 728 (SM) PRODUCT 3 400 363 400 363 400 363 400 363 400 363 500 363 900 FI SALES	UNEMPLOYE TOTAL EMPLOYED V 167 167 167 167 167 T & LOSS ACCOU	50 IN S.A. SKILLED VAGE RATE 440 440 440 440 1307379	COMPAN INFORMAT COMPANY COMPANY COMPANY	NALLY ADJ) NY PRO 2 115 2 4 115 2 6 115 2 6 115 2 8 115 BALANC	64 Douct 1 H 5 108 5 108 5 108 5 108 5 108 5 108	CENTRAL PRODUCT 2 E PRICES PARCES PARCES PARCES PARCES PRI	24 PRODUCT1 MIS MINOR BANK RATE FO PRODUCT 3 400 383 400	7 PPRODUCT: NONE NONE R NEXT QT TOTAL EMPLOYED 167 167 167 167 167	2 4 'R. S W/A 133
SKILED MEN HIRED D MATERIALS ONDERED FROM SUPPLER MACHIES ORDERED GROSS GROSS NAT. PRODUCT (SE. COMPANY PRODUCT (SE. COMPANY PRODUCT (SE. COMPANY 115 108 COMPANY 3 115 108 COMPANY 5 115 108 COMPANY 7 115 108 COMPANY 8 115 108 COMPANY 9 115 108 COMPANS 5 115 108 SALES OFFICE SALES OFFICE	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRODUCT 2 PRICES PRODUCT 2 195 163 195 16	TRAINED D DELVERIES 5 PRODUCT 3 5 PRODUCT 3 6 400 363 400 363 400 383 400 383 PROFI SALES	UNEMPLOYE TOTAL EMPLOYED V 167 167 167 167 167 T & LOSS ACCOU JIE	2D IN S.AI SKILLED JAGE RATE 440 440 440 440 440 1307379 187946	COMPAN COMPANY COMPANY COMPANY COMPANY COMPANY COMPANY COMPANY	RALLY ADJ) NY Partition (2 115 (4 115 (4 115 (5 115) BALANC S S S S S S S S	64 DOULCT 1 H 5 108 5 108 5 108 5 108 5 108 5 108 5 108 108 108 108 108 108 108 108	CENTRAL PRICES P	24 PRODUCT1 MIS MINOR BANK RATE FO PRODUCT3 400 383 400 383 400 383 400 383 CC TRADING RECEIPTS CAPITAL RECEIPTS CAPITAL RECEIPTS	7 PRODUCT: NONE	2 1 'R. S W/ 133
SKILLED MEN HIREO D MATERIALS ONDERED FROM SUPPLER MACHINES GREERED FROM SUPPLER MACHINES GREERED FROMOUT (SEL COMPANY PRODUCT (SEL COMPANY PRODUCT (SEL COMPANY 115 108 COMPANY 1 115 108 COMPANY 5 115 108 COMPANY 7 115 108 SALES OFICE SALES OFICE BALES OFICE COMANTES SERVICING TRANCERS EFCE EFMACHINE	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRODUCT2 E DA 195 163 195 163 19	TRAINED D DELVERIES 5 DELVERIES 5 PRODUCT 3 5 PRODUCT 3 400 383 400 383 400 383 400 383 5 SALES OPENING STOCK VALL MATERIALS PURCHAS SULTED WARES	UNEMPLOYE TOTAL EMPLOYED V 167 167 167 167 T & LOSS ACCOU UE ED	ED IN S.AJ SKILLED JAGE RATE 440 440 440 1307379 187946 510022	COMPANY COMPANY COMPANY COMPANY COMPANY COMPANY COMPANY	NY Pathy ADJ NY Pathy TDON E (2 115 (3 112 BALANC 5 S Entry UCT STOCKS Control	6% Douct ; H 5 108 5 108	CENTRAL PRICES P	24 PRODUCT1 MIS MINOR BANK RATE FO PRODUCT3 400 383 400 400 400 400 400 400 400 400 40	7 PRODUCT: NONE NONE NONE NONE NONE NONE NONE NON	2 4 PR. SS W/P
SKILLED MEN HIREO D MATERIALS ONDERED FROM SUPPLICE FROM SUPPLICE MACHINES ORDERED GROSS NAT. PRODUCT (SE. COMPANY INFORMATION PRODUCT (SE. COMPANY PRODUCT (SE. COMPANY 115 108 COMPANY 3 115 108 COMPANY 7 115 108 GUADANTES SLAINES FIC SALES OFFICE GUADANTES SERVERING IMANTENDAT FLEET HAREU TRANSPORT HEE <td>DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRODUCT2 E V 1 95 103 195 103 195 183 195 183 195 183 195 183 0 12997 1 10870 1 10870 1 10870 1 0</td> <td>TRAINED D DELVERIES 5 DELVERIES 5 PRODUCT 3 1 PRODUCT 3 1 400 383 400 383 400 383 980FI SALES 0 OPERING STOCK VALL MATERIALS PURCHAS SKILLED WAGES SKILLED WAGES</td> <td>UNEMPLOYE TOTAL EMPLOYED W 167 167 167 167 17 & LOSS ACCOU UE ED</td> <td>ED IN S.AJ SKILLED /AGE RATE 440 440 1307379 187946 510022 91660 156534</td> <td>COMPAN COMPAN COMPANY</td> <td>NY PRILY ADJ J NY PRIN (2 115 (4 115 (6 115 (6 115 (8 115) BALANC S ERTY UNES UCT STOCKS RIAL STOCKS</td> <td>6% DOULCT ; H 5 108 5 10</td> <td>CENTRAL PRICES PRICES PRODUCT 2 E 01 195 183 195 183</td> <td>24 PRODUCTI MIS MINOR BANK RATE FO PRODUCT 3 E C PRODUCT 3 E C PRODUCT</td> <td>7 PRODUCT: NONE NONE NONE NONE NONE NONE NONE NON</td> <td>2 4 'R. S W/P 133</td>	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRODUCT2 E V 1 95 103 195 103 195 183 195 183 195 183 195 183 0 12997 1 10870 1 10870 1 10870 1 0	TRAINED D DELVERIES 5 DELVERIES 5 PRODUCT 3 1 PRODUCT 3 1 400 383 400 383 400 383 980FI SALES 0 OPERING STOCK VALL MATERIALS PURCHAS SKILLED WAGES SKILLED WAGES	UNEMPLOYE TOTAL EMPLOYED W 167 167 167 167 17 & LOSS ACCOU UE ED	ED IN S.AJ SKILLED /AGE RATE 440 440 1307379 187946 510022 91660 156534	COMPAN COMPAN COMPANY	NY PRILY ADJ J NY PRIN (2 115 (4 115 (6 115 (6 115 (8 115) BALANC S ERTY UNES UCT STOCKS RIAL STOCKS	6% DOULCT ; H 5 108 5 10	CENTRAL PRICES PRICES PRODUCT 2 E 01 195 183 195 183	24 PRODUCTI MIS MINOR BANK RATE FO PRODUCT 3 E C PRODUCT	7 PRODUCT: NONE NONE NONE NONE NONE NONE NONE NON	2 4 'R. S W/P 133
SKILED MEN HIRED 0 MATERIALS ONDERED FROM SUPPLICA MACHINES ORDERED FROM SUPPLICA MACHINES ORDERED MACHINES ORDERED GROSS NAT. PRODUCT (SEC. MACHINES ORDERED COMPANY PRODUCT (SEC. INFORMATION E COMPANY 1.15 COMPANY 5 1.15 COMPANY 7 1.15 GROSS SATARES SALAHIES ETC SALESMENTS SALAHIES ETC SALSS OFFICE IMAD TRANSPORT FLECT IHARD TRANSPORT INFORMANTONT INFORMANTONT	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRICES PRIOUCES PRIOUCES 195 103 195 103 103 103 103 103 103 103 103 103 103 103 103 103 105 00 105 000 105 000 105 000 105 000 105 000 105 000 105 000 10	TRAINED 0 DELWERES 5 DELWERES 5 DELWERES 5 PRODUCT 3 8 400 363 400 363 400 383 400 383 SALES PROFI SKILED WAGES SKILED WAGES MATERIALS FURCHAS SKILED WAGES MACKINE RUNNING SUNSKILE OWAGES	UNEMPLOYE TOTAL EMPLOYED W 167 167 167 167 167 167 167 167	ED IN S.AI SKILLED /AGE RATE 440 440 440 1307379 187946 510022 91060 158634 63678 250185	REA (SEASON INFORMAT COMPANY C	NY PRILY ADJ J NY PRIN (2 1) 5 (2 1) 5 (3 1) 5 (4 1) 5 (5 1) 5 (6 1) 5 (7 1) 5 PALANC S. RIAL STOCKS RIAL STOCKS ORS	0 63 000UCT 3 H 6 108 5 108 5 108 5 108 5 108 5 108 5 108 1 2 E SHEET	CENTRAL PRICES PAICES P	24 PRODUCTI MIS MINOR BANK RATE FO PRODUCT3 EANK RATE FO PRODUCT3 400 383 400 300 30 400 300 30 400 300 30 400 300 300 400 30	7 PRODUCT: NONE NONE NONE NONE NONE NONE NOTAL EMPLOYED 167 167 167 167 167 20 20 20 20 20 20 20 20 20 20 20 20 20	2 1 7R. 133 122 139
SKILED MEN HIRED D MATERIALS ONDERED FROM SUPPLY FROM SUPPLY FROM SUPPLY MACHINES ORDERED GROSS NAT. PRODUCT (SEX) GROSS NAT. PRODUCT (SEX) FRODUCT (SEX) COMPANY PRODUCT (SEX) INFORMATION FRODUCT (SEX) COMPANY 1 115 COMPANY 5 115 COMPANY 5 115 COMPANY 7 115 GUADANTE SENVCING GUADANTE SENVCING ITANSPORT FLEET HABU TRANSPORT I PRODUCT RESEARCH FROMUCT HEADY I PROPUCT RESEARCH FROMUCT HEADY	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRIOUCS E 0 195 103 195 103 100 0 13000 13000 13000 13000 13000 13000 100000 100000 10000 100000 10000 100000 100000 100000	TRAINED 0 DELWENES 5 DELWENES 5 PRODUCT 3 1 PRODUCT 3 1 400 363 400 383 400 383 5ALES OPENING STOCK VALL MATERIALS PURCHAS SKILED WAGES MATERIALS PURCHAS SKILED WAGES MATERIALS PURCHAS SKILED WAGES MATERIALS PURCHAS COST OF SALES	UNEMPLOYE TOTAL EMPLOYED W 167 167 167 167 T & LOSS ACCOU Jee K VALUE	D IN S.AI SKILLED SKILLED SKILLED 440 440 440 1307379 187946 510022 91060 156534 63678 250165 761155	COMPAN INFORMAI COMPANY COMPAN	NY PROVIDENT OF THE PRO	0 63 00000013 5 108 5 108 5 108 5 108 5 108 1 CE SHEET	CENTRAL PRICES PAICES P	24 PRODUCT 1 PRODUCT 3 PRODUCT	7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	2 1 'R. S W/ 133 12 139
SKILED MEN HIRED D MATERIALS ONDERED IFAON SUPPLICA MACHINES ORDERED IFAON SUPPLICA MACHINES ORDERED IFAON SUPPLICA MACHINES ORDERED IFAON SUPPLICA GROSS NAT. PRODUCT (SEX IFAON SUPPLICA INFORMATION PRODUCT 1 COMPANY 1 INFORMATION FRODUCT 1 COMPANY 1 1.15 COMPANY 5 1.15 COMPANY 7 1.15 RECONTON & ADVERTISING SALESENTS SALAMIES ETC SALES OFFICE GUADANTE SENVCING I TRANSPORT FLET HIREU TRANSPORT PRODUCT RESEARCH RECHINTMENT & TRAINING MARTEHAUGUS A PUTCHASING MARTEHAUGUS MARTEHAUGUS A PUTCHASING MARTEHAUGUS	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRIOUCES PRIOUCES PRIOUCES 195 103 195 103 103 195 103 105 105 105 105 105 105 105 105 105 105	TRAINED D DELIVERIES 5 DELIVERIES 5 PRODUCT 3 PRODUCT 3 PRODUCT 3 400 400 383 400 383 400 383 400 383 400 383 SALES OPENING STOCK VALL WARTERALS PURCHAS SKILLTO WAGES UNSKILLED WAGES UNSKILLED WAGES UNSKILLED WAGES GROSS PROFIT/LOSS	UNEMPLOYE TOTAL EMPLOYED V 167 167 167 167 167 T & LOSS ACCOU JRE ED	ED IN S.AI SKILLED VAGE FATE 440 440 440 440 440 1307379 187946 510022 91060 91060 510625 546224	COMPAN INFORMAI COMPANY COMPAN	NALLY ADJ) NY partition (2 11) (4 11) (4 11) (5 11) (6 11) (7 11)	64 000UCT 3 H 5 108 5 108	CENTRAL PRICES PAICES P	24 PRODUCT 1 MIS MINOR BANK RATE FO PRODUCT 3 E 00 383 400 383 400 383 400 383 100 3	7 PRODUCT: NONE NONE R NEXT QT TOTAL EMPLOYED 167 167 167 167 167 167 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	2 1 'R. S W/P 133 12 139
SKILED MEN HIRED D MATERIALS ONDERED I IFAON SUPPLICA MACHINES ORDERED GROSS NAT. PRODUCT (SEL MACHINES ORDERED GROSS NAT. PRODUCT (SEL I INFORMATION E PRODUCT (SEL COMPANY 1.15 108 COMPANY S 1.15 108 COMPANY S 1.15 108 COMPANY T 1.15 108 COMPANY S 1.15 108 COMPANY S 1.15 108 COMPANY T 1.15 108 COMPANY S 1.15 108 COMPANY S 1.15 108 COMPANY T 1.15 108 GROSS SALAMES ENCING I INDUST TRANSPORT FLET INREUTANSPORT INDUST TRANSPORT FLET INRELIVELISTICE	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRICES PRIODUC2 ENDOUC1 195 103 195 183 195 184 195 184	TRAINED 0 DELVERIES 5 DELVERIES 5 PRODUCT 3 PRODUCT 3 PRODUCT 3 400 400 383 400 383 400 383 SALES OPENING STOCK VALL MATERIALS PURCHAS SKILLED WAGES UNSKILLED WAGES UNSKILLED WAGES UNSKILLED WAGES UNSKILLED WAGES GROSS PROFIT/LOSS INTEREST RECEMPTO	UNEMPLOYE TOTAL EMPLOYED W 167 167 167 17 & LOSS ACCOU JUE ED	ED IN S.A. SKILLED 440 440 440 440 440 440 440 44	COMPAN INFORMAN COMPANY COMPAN	NALLY ADJ) NY PR 100 P 2 115 2 112 4 115 2 112 8 112 BALANC S. ENTY INVESTO INVESTO INES: UCT STOCKS SKETESTD TICSN: OVERDART	0 6% 000007 1 5 108 5 108	CENTRAL PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES CENTRAL 195 183 195 183	24 PRODUCT 1 MIS MINOR BANK RATE FO PRODUCT 3 E ANK RATE FO PRODUCT 3 E ADD 383 400 383 400 383 400 383 CONTRACT AND CAPTAL RECEIPTS INTEREST RECIVERS ADDITIONAL LOANS TRADING PAYMENT CAPTAL	7 PRODUCT: NONE NONE R NEXT QT TOTAL EMPLOYED 167 167 167 167 167 167 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	2 4 9R. 133 12 139
SKILED MEN HIREO D MATERIALS ONDERED IFAOM SUPPLICA MACHINES ORDERED IFAOM SUPPLICA MACHINES ORDERED IFAOM SUPPLICA MACHINES ORDERED IFAOM SUPPLICA COMPANY PRODUCT (SEL INFORMATION E COMPANY 1.15 INFORMATINY 1.15 COMPANY 1.15 BALS COMPANY PROMOTION & ADVERTISING SALES OFICE GUADANTES SERVICING SALES OFICE GUADANTES SERVICING ITANARONT FLET HIREU TRANSPORT PRODUCT RESARCH INCELIVISING A FUNCHASING MAINTERANCE MAINTERANCE MAINTERANCE MAINTERANCE	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRIODUCZ ENCOUNT 195 103 195 183 195 183	TRAINED 0 DELANEMES 5 DELANEMES 5 PRODUCT 3 PRODUCT 3 PRODUCT 3 PRODUCT 3 400 383 400 383 400 383 SALES OPENING STOCK VALL MATERIALS PURCHAS SKILLED WAGES UNSKILLED WAGES UNSKILLED WAGES UNSKILLED WAGES SKILLED WAGES UNSKILLED WAGES SKILLED WAGES UNSKILLED WAGES UNSKILLED WAGES UNSKILLED WAGES SKILLED WAGES SKILLED WAGES SKILLED WAGES GROSS PROFIT/LOSS SKILLED WAGES MITEREST PRID SKILES	UNEMPLOYE TOTAL EMPLOYED W 167 167 167 167 167 167 167 Constant 167 167 167 167 167 167 167 167	ED IN S.A. SKILLED 440 440 440 440 440 440 1307379 187946 510022 91660 156634 635702 91660 156634 63575 546224 0 54672 440 54575 546724 0 54575 546724 0 54575 545724 0 54575 545724 0 54575 545724 0 54575 545724 0 54575 545724 0 54575 545724 0 54575 545724 0 545755 5457575 5457575 5457575 5457575 5457575 5457575	COMPAN INFORMAN COMPANY COMPAN	NALLY ADJ) NY PRITON 2 115 2 4 115 2 6 115 2 6 115 BALANC S. ERTY INVESTO VCT STOCKS RIAL STOCKS ORS SESESSED TOORS TESS. OVERDRAFT CURED LOAKS	0 64 DOUCT 1 5 108 5 108 5 108 5 108 5 108 5 108 5 108 5 108 108 5 108 108 5 108 108 5 108 5 108	CENTRAL PRICES PRICE	24 PRODUCT 1 HITS MINOR BANK RATE FO PRODUCT 3 E ANK RATE FO PRODUCT 3 400 383 400 383 400 383 400 383 100 300 100 30	7 PRODUCT: NONE NONE NONE R NEXT QT TOTAL EMPLOYED 167 167 167 167 167 167 167 167 167 167	2 1 9 R. 9 R. 133 12 139 5
SKILED MEN HIRED D MATERIALS ONDERED I IFAON SUPPLICA MACHINES ORDERED GROSS NAT. PRODUCT (SEL I COMPANY PRODUCT (SEL INFORMATION E COMPANY 1.15 INFORMATION E COMPANY 1.15 COMPANY S 1.15 COMPANY S 1.15 COMPANY T 1.15 COMPANY S 1.15 COMPANY S 1.15 COMPANY T 1.15 GROSS SALINES ENCONG I GROMONDA & ADVERTISING SALINES ENCONG GRAMMES SALANIES ENCONG I GRAMMES ENVERNG I I TANABORIT FLET I NEULIMENT & TANING MAINTEWAXCE MAINTEWAXCE MAINTEWAXCE MAINTEWAXCE MAINTEWAXCE MAINTEWAXCE MAINTEWAXCE MAINTEWAXCE	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRICES PRICUES PRICUES 195 103 195 103 195 183 195 185 185 195 18	TRAINED D DELWENES 5 DELWENES 5 PRODUCT 3 PRODUCT 3 400 400 400 400 383 400 A00 833 400 840	UNEMPLOYE TOTAL EMPLOYED V 167 167 167 167 167 167 167 167	ED IN S.AI SKILLED 440 440 440 440 440 1307379 187346 510022 91060 158634 63570 546224 0 54575 546224 0 54575 546224 0 54575 546224 0 54575 546224 0 54575 546224 0 54575 546224 0 54575 546224 0 54575 546224 0 54575 546224 0 54575 546224 0 54575 546224 0 54575 546224 0 54575 54575 546224 0 54575 546224 0 54575 546224 0 54575 54575 54575 54575 54575 54575 54575 54575 54575 54575 54575 54575 54575 54575 54575 54575 54575 54575 545755 545755 54575 55755 5457575 5457575 545757575 5457575 545757575 545757575 5457575	COMPANY COMPAN	NALLY ADJ) NY PR 100 P 2 115 2 115 2 115 2 115 2 115 2 115 2 115 BALANC S. ENTY BALANC UGT STOCKS RAL STOCKS RAL STOCKS ORS INVESTID INSESSED TOORS UCYMEDIAAFT CURED LOANS SETS	64 0000071 5108	CENTRAL PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES CONTRAL 195 183 195 183 1	24 PRODUCT 1 MINOR BANK RATE FO PRODUCT 3 E OUT 7 E	7 PRODUCT: NONE NONE R NEXT QT TOTAL EMPLOYED 167 167 167 167 167 167 167 167 167 167	2 1 9R. 133 122 135 5 46
SKILLED MEN HIRED D MATERALS ONDERED I IRADES ONDERED I IRADA SUPUCA MACHINES ORDERED GROSS NAT. PRODUCT (SEL I IMACHINES ORDERED I GROSS NAT. PRODUCT (SEL I INFORMATION E PRODUCT (SEL COMPANY 1.15 108 COMPANY 5 1.15 108 COMPANY 7 1.15 108 GROSS SALARIES ETC SALESMERS SALARIES ETC SALESMERS SALARIES ETC SALESMERS SALARIES ETC MARTENACE MANTENACE MANTENACE MANTENACE MANTENACE MANTENACE MANTELILICITICE MANTENACE MANAGEMENTOL OTHER COSTS	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRICES PRIOUCT 2 E 195 103 195 103 195 183 195 185 185 195 185 195 185 195 185 195 185 195	TRAINED D DELWENES 5 DELWENES 5 PRODUCT 3 PRODUCT 3 400 400 400 400 400 400 833 400 840 983 400 840	UNEMPLOYE TOTAL EMPLOYED V 167 167 167 167 167 167 167 167	ED IN S.A.I SKILLED 440 440 440 440 440 440 440 44	COMPANY COMPAN	NALLY ADJ) NY PR (2 115 (2 115 (2 115 (3 112 (4 115 (5 112 (5 112 (6 112 (7 115 (8 112 (9 112 (10 510CXS RAL STOCKS 005 (10 5525520 TOORS 5525520 TOORS 5525520 TOORS 5525520 TOORS 5525520 TOORS 5515 5525520 552520 100 552520 100 552520 100 552520 100 552520 100 552520 100 552520 100 552520 100 552520 100 552520 100 552520 100 <t< td=""><td>6% 0000071 5 108 5 1</td><td>CENTRAL PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES CONTRAL 195 183 195 183</td><td>24 PRODUCT 1 MINOR BANK RATE FO PRODUCT 3 E OUT 7 E OUT 7 E</td><td>7 PRODUCT: NONE NONE R NEXT QT TOTAL EMPLOYED 167 167 167 167 167 167 167 167 167 167</td><td>2 4 2 4 3 3 4 6</td></t<>	6% 0000071 5 108 5 1	CENTRAL PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES CONTRAL 195 183 195 183	24 PRODUCT 1 MINOR BANK RATE FO PRODUCT 3 E OUT 7 E	7 PRODUCT: NONE NONE R NEXT QT TOTAL EMPLOYED 167 167 167 167 167 167 167 167 167 167	2 4 2 4 3 3 4 6
SKILLED MEN HIRED D MATERALS ONDERED ITADA SUPUCA MACHINES ORDERED ITADA SUPUCA MACHINES ORDERED ITADA SUPUCA MACHINES ORDERED ITADA SUPUCA GROSS NAT. PRODUCT (SE, INFORMATION E ON INFORMATION E I	DISMISSED 0 12000 2 0 ASONALLY ADJ PRICES PRIOUCT 2 F000UT 2 195 103 195 183 195 18	TRAINED 0 DELAYENES 5 DELAYENES 5 PRODUCT 3 1 PRODUCT 3 1 400 383 400 383 400 383 400 383 9 9 SALES 0 OPFAING STOCK VALL VAGES UNSKILED WAGES UNSKILED WAGES UNSKILED WAGES<	UNEMPLOYE TOTAL EMPLOYED W 167 167 167 167 167 167 167 KOSS ACCOU UE ED K VALUE	ED IN S.A. SKILLED 440 440 440 440 440 440 440 1307379 187946 510022 91660 156534 63570 546224 0 546224 0 54627 506515 546224 0 54575 0 -4543	COMPANY COMPAN	NALLY ADJ) NY PR (2 115 (2 115 (2 115 (2 115 (3 115 (4 115 (5 115 (5 115 BALANC Service NVESTO INVESTO INSESSED IOANS VORDARAFT CURRARAFT CURED LOANS SETS	6 6 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	CENTRAL PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES PRICES CONTRAL 195 183 195 183	24 PRODUCT 1 MIS MINOR BANK RATE FO PRODUCT 3 EANK RATE FO State PRODUCT 3 EANK RATE FO State COMPACT A PRODUCT 3 EANK RATE FO State CANS RECEIPTS INTEREST PRODUCT 3 EANK RATE FO CANS RECEIPTS INTEREST PRODUCT 3 INTEREST A INTEREST PRODUCT 3 INTEREST A INTEREST PRODUCT 3 INTEREST A INTEREST A INTE	7 1 PRODUCT: 1 NONE 1 N	2 4 2 4 3 3 4 6

172

the Contraction of the second second

and a star

Partie ad a series

1									
GAME HISTORY	THE S	COTSMAN' MAN	AGEMENT GAN	ME ~ 1989	(Round One)	Тні	IS IS THE FO	URTH AND	LAST
GAME HISTORY GAME HISTORY		CREATED AND	PAIP	ERE	ten per	PAR	T OF THE GA	ME HISTOR	<i>ı</i> .
GAME HISTORY		ADMINISTERED BY			16 the	YEA	AR 1988, QT	'R. 4	
	GROUP 1	COMPANY 1	IDENTITY No	. 1 Y	EAR 1988	QUARTER 4			
DEPORT BASED ON DEGLETOUS	PELON	P	LANT/MATERIA	ALS/PERSONNE	iL		PRODUCTS/ORDE	ERS/STOCKS	
CHECK THEM BEFORE GOING PO	RTHER.	MACHINES AVA	LABLE LAST QUAR	TER	13	PRODUCT GLS	PRODUCT 1	PRODUCT 2	1180
LODAL CONTRACTOR	UCT I PRODUCT 2 PRODUCT 3					PRODUCED	4972	3155	, 1216
MAJOR IMPROVEMENTS TAKEN UP	0 1 0	VANS AVAILABL	LE LAST QUARTER			SERVICED	89	53	20
PRICES EXPORT AREA 1 HOME AREA 1	15 195 400 08 183 385	SKILLED MAN AVAILABLE L	HOURS: AST QUARTER		27636	DELIVERED TO	PRODUCT	PRODUCT 2	PRODUCT
PROMOTION IN: TRADE PRESS	6 10 7	USED LAST O	A WARTER		121 25360	EXPORT SOUTH	3700	2250	1 800
ADV. SUPPORT MERCHANDISING	10 15 8 4 4 5	NOTICE OF ST	RIKE WEEKS NEXT	QUARTER	0	WEST NORTH	1 100	110	· 60 179
FITTING TIMES (MINS)	15 ' 170 340	I MACHINE HOU	IRS: AST QUARTER		14196	ORDERS FROM	PRODUCT 1	PRODUCTZ	PRODUCT
SALESMEN ALLOCATED TO EXPORT 13 5	3 W 1 N 5	BREAKDOWN	TIME		31	EXPORT	3897	1976	636
		USED LAST O	UARTER		11686	WEST	122	111	71
SKILLED WAGE RATE 440	SHIFT LEVEL 2	MACHINE EFF	ILIENLY (%)		97.2	5010 10	PRODUCTI	420	PRODUCT
MANAGEMENT BUDGET BO		OPENING STO	TS: CK		11624	EXPORT	3835	1976	636
MAINTENANCE HOURS 60 1 DIVIDEND 0	CREDIT PERIOD 30	USED			12000	WEST	1 304	242	· 131 71
INFORMATION ON COMPANIES	VANS SOLD 0 . MARKET SHARES 0	CLOSING STO	CK IR NEXT QUARTER		8694 10000	NOATH	715	426	158
DELIVERIES REQUESTED TO	T 1 PRODUCT 2 PRODUCT 3	AVAILABLE N	EXT QUARTER		18694	EXPORT	ILIS PRODUCT	PRODUCT 2	PRODUCT
EXPORT 3700	2250 800	PERSONNEL	SALESI	MEN ISKILLED ME	MEN I QR	1 SOUTH	3	Ö	0
Full F STARRED WEST 100	110 60	RECRUITED	UIA 2	0 0	6	NORTH	Ő	ō	Ď
NUAM 725	450 170	TRAINED		0 0		PEODUCTS STOCK	SW PRODUCTI	PRODUCT 2	PRODUCT
RESEARCH EXPENDITURE 13	10 12	LEFT AVAIL FORMS	атотя 2	2 47	98	I EXPORT	0	2/4	189
SKILLED MEN HIRED 0 DISMIS	SED 0 TRAINED 0 SED 0 TRAINED 0					I WEST	12	0 24	5 25
MATERIALS ORDERED 10000						PRODUCT	PRODUCT 1	PRODUCT 2	PRODUCT
FROM SUPPLIER 2 MACHINES ORDERED 0	DELIVERIES 4						MINOR	NONE	MINOR
COY. INFO. ADVERT. RESEARCH	CONSUMER ASSESS	TENT RATINGS							
COMPANY 1 69000 35000	****	****							
COMPANY 3 69000 35000	****	****							
COMPANY 5 69000 35000	****	****							
COMPANY 7 69000 35000	**** ****	****							
COMPANY B 69000 35000									
GROSS NAT. PRODUCT (SEASONALL)	(ADJ) 733(\$M)	UNEMPLOYED	IN S.AREA	(SEASONALL	Y ADJ) 6%	CENTRAL B	ANK RATE FOR	R NEXT QTR	. 81
COMPANY PRODUCT 1 PRO INFORMATION E H F	ICES	TOTAL S		COMPANY	PRODUCT	PRICES	PRODUCT 3		SKILLED
COMPANY 1 115 108 195	183 : 400 385	167	440	COMPANY 2	115 108	195 183	400 385	167	440 2
COMPANY 3 115 108 195 COMPANY 5 115 108 195	183 400 385 183 400 385	167	440	COMPANY 4 COMPANY 6	115 108 115 108	195 183 195 183	400 385 400 385 -	167 167	440
COMPANY 7 115 108 195	183 400 385	167	440 1	COMPANY 8	115 108	195 183	400 385	167	440
OVERHEADS	PROF	IT & LOSS ACCOUN	1		BALANCE SHEE	т <u> </u>	C	ASH FLOW	
PROMOTION & ADVERTISING 69000) SALES		1508647	ASSETS:-		200000	TRADING RECEIPTS	1	446074
SALES OFFICE 14925	OPENING STOCK VAU	UÉ	250185	MACHINES		1155842	INTEREST RECEIVED	0	0
TRANSPORT FLEET 127030	SKILLED WAGES	жD	127626	PRODUCT S	TOCKS	55960	ADDITIONAL LOANS	D 5	0
I HIRED TRANSPORT 4127	MACHINE RUNNING		184802 74276 ,	DEBTORS	STOCKS	157726 538302	TRADING PAYMENT	5 1	422481
I RECRUITMENT & TRAINING 3000 I MAINTENANCE 3120/) LESS CLOSING STOCE) COST OF SALES	K VALUE	213686 794299	CASH INVES	STED	0	CAPITAL PAYMENTS	S	10000
WAREHOUSING & PURCHASING 12453 BUSINESS INTELLIGENCE 3000	GROSS PROFIT/LOSS		714348	LABILITIES:-	SED	70089	INVESTMENTS BOU	GHT	0
1)		0	CREDITORS	DBAST	496396	LOANS REPAID		6441
I MANAGEMENT BOOOL	I INTERECT DECENTO			UNSECURE	DLOANS	0	SAUDENUS FAID		U
I MANAGEMENT B0000 I CREDIT CONTROL 6544 I OTHER COSTS 7346	INTEREST RECEIVED		7152	ongecone					
I MANAGEMENT 80001 I CREDIT CONTROL 6544 OTHER COSTS 7340 ·	INTEREST RECEIVED INTEREST PAID OVERHEADS DEPRECIATION		539976 38134	on de contre		1100000	OVERDRAFT LIMIT	R	529000
малидемент 80001 Сперт сонтар. 654. ОТНЕК СОSTS 734. 1 ТОТАL OVERHEADS 53997.6	interest received interest paid overheads depreciation tax assessed		7152 539976 38134 70089	NET ASSETS		1429986	OVERDRAFT LIMIT	R	529000
Imanuferment 80001 CREDIT CONTROL 654. OTHER COSTS 7340 TOTAL OVERHEADS 539970 Imanual Control 1 Imanua Control	INTEREST RECEIVED INTEREST PAID OVERHEADS DEPRECIATION TAX ASSESSED NET PROFIT/LOSS		7152 539976 38134 70089 58997	NET ASSETS ORDINARY CA RESERVES	PITAL	1429986 1100000 329986	PRICE OF MATERIA	R 	529000
MANUGEMENT B0001 CREDT CONTROL 654. OTHER COSTS 7341 TOTAL OVERHEADS 539971 TAXABLE PROFIT/LOSS 233631	INTEREST RECEIVED INTEREST PAID OVERHEADS DEPRECATION TAX ASSESSED NET PROFIT/LOSS DIVIDENDS PAID TANASFERRED TO RE	SERVES	7152 539976 38134 70089 58997 0 58997	ORDINARY CA RESERVES TOTAL SHARE	PITAL CHOLDERS FUNDS	1429986 1100000 329986 1429986	PRICE OF MATERIAL ORDERED FOR NEX QUARTER (1000 UN	R 	35734

EXECUTIVE

Forewords

The Executive range of simulations model the Motor Industry, using real data. The Executive games can be used to illustrate a number of management training aspects. Their run is based on 9 teams that compete against each other and the existing market. The Executive range is use at more than 100 companies and educational institutions.

Game's purposes

The Executive game has been used by Manufacturing, Finance and Service companies. It has been employed as a run/test of the learning elements taught during a previous training program.

The main learning objectives are :

- Team Building
- ♦ Decision making,...
- Awareness of the existing functions in a factory
- ♦ Strategic Planning,...
- ♦ Finance,...

Executive is considered to be a fully interactive game. Its interaction is based on the fact that the decisions taken by one team will influence the other teams playing the game. The Executive's Master Simulation simulates the influence of non actively playing teams.

Game description

The Executive game is based on the Western car market, which is divided into 16

market sectors. The reason why this market was chosen, is because it is a fairly known business and it is also easy to understand. Game's players what it is like.

The working process is very similar to the one described in TOPAZ. Players are gathered in groups, where each of their members performs a different role. After a discussion period, the players will decide about certain aspects of the factory. Those decisions will be entered into the computer for later processing. Some numerical results will come out after. Those results are the summary reports of the performance of the factory.

あいいまです いい きい うたい しょういう

1. The state of the second state of the state of the second state of the second state of the

The factory to be played starts from scratch, there is only money. The player has to decide about the way the money is going to be spend (workforce, facilities,..). The game could also start from an existing set-up or historic position.

The decisions could be supported by accessing to previous teams' information. The Tutor can also access that information.

Technicalities

Apart from the basic model, there are some other different versions of Executive. These models have some added modules that will allow the more specialized study of factory related issues as Quality, Finance, and Marketing.

The number of teams playing is between one and nine. If the game's purpose is teambuilding, leadership,...it is advice to work with teams of five people; and two, three or even four players if the purpose is strategy, finance,...The number of teams playing will influence the Market Place playing with. The game's cycle is one year.

The reports generated by Executive are shown in the next pages.

EXECUTIVE REPORTS

Executive-300 Business Simulation. Release 4.40. A Product of April Computing Executive Ltd. Purchased by : April Computing - PW

PRIVATE COMPANY REPORT

Tea	m 1 -	CAT Motors			Exercise	: STD	End Of Ye	ar 3
Se	ctor/					Material	s	
Ľ	ode 1	Produced	Sold	STOCKS	Price	Cost	Market Share	*
6	Felix	155806	114230	62007	9500.00	5970.57	3.17	
11	Tom	155806	142475	23241	12000.00	8677.99	4,61	
16	Garfield	10380	5983	8683	36000.00	13585.41	0.60	
Tot	al Workfo	rce	40000					
Pro	ductivity		8.05	cars / m	an / year			
Pro	ductivity	Index	1.04					
Day	s lost to	strikes	10					
Tot	al Market	Sma	11	L. Med	υ.	Med	Luxury	
	Sizes	2275	643	3606546	309	1325	990380	

PROFIT AND LOSS ACCOUNT (£ millions)

Sales	3010,27	
Cost of Sales *		2345.97
Gross Profit	664.31	
Overheads:		
Fixed Overheads		249.01
Cost of Stock Upkeep		62.88
Promotion		68,10
Professional Charges		2,95
Depreciation		48.60
Operating Profit	232.77	
Interest on Current Account	52,78	
Interest on Loans		23.73
Pre-Tax Profit	261.82	
Tax		91.64
	BBMBBBBB	
Post-Tax Profit	170.18	

* Cost of Sales is calculated as

Opening Stock	314.41	
plus Materials Cost	2424.39	
plus Wages	453.08	
less Closing Stock		845.91
Cost of Sales	2345.97	

	CASH FLOW	(£ m111ions)
Opening Bank Balance	586.84	
Receipts:		
Revenues	3010.27	
Net Interest	29.05	
Paymonto		
Materials Cost		2424.39
Wages Cost		453.08
Total Overheads		382.94
Loan Repayments		124.97
Tax Payments		177.26
Balance Before Loans	63,52	
New Normal Loan	250.00	
Closing Bank Balance	313.52	

- ----

. . . .

÷

Sealed and the

the week's the way

and the second second second

BALANCE SHEET	(£ millions)
Fixed Assets: Cost Deprectation	600.00 -162.60
Book Value	437.40
Current Assets: Stock Bank Balance	845.91 313.52
Current Liabilities: • Outstanding Emergency Loan Tax Liabilities	0.00 -91.64
Net Current Assets	1067,80
Total Assets Less Current Liabilities	1505.20
Capital and Reserves: Shareholder's Equity Retained Profit Total Shareholders Funds	500.00 755.20 1255.20
Long Term Liabilities: Outstanding Normal Loan	250,00
Total Capital Employed	1505.20

1

FINANCIAL INDICATORS

- 4 - 12 - 10 - -

.

100 127 12

a the day a

and the state of the state of the state

Outstanding Debt (£m)	250.00	Return On Assets	17.39 %
Current Ratio	12.65	Gross Margin	22.07 %
Quick Ratio	3.42	Post-Tax Profit / Sales	5.65 %
Liquidity Ratio	*******	Profit / Employee (£)	6545.41

There are no R & D projects for this team.

	E	XTRAS	INCLUDED	
		MDL 1 Felix	MDL 2 Tom	MDL 3 Garfield
1.	Larger Engine Option	Yes	Yes	Yes
2.	5-Speed gearbox	Yes	Yes	Yes
з.	Fuel Injection	Yes	Yes	Yes
4.	Alloy Wheels	Yes	Yes	Yes
5.	Sunroof	Yes	Yes	Yes
6.	Headlamp Washwipes			Yes
7.	Spoiler Set	Yes	Yes	Yes
8.	Metallic Paint	Yes	Yes	Yes
9.	Electric Windows	Yes	Yes	Yes
10.	Electric Aerial	Yes	Yes	Yes
11.	Automatic Transmission			Yes
12.	Anti-lock Braking (ABS)	Yes	Yes	Yes
13,	Electric Seat Heating			Yes
14,	Electric Wing Mirrors			Yes
15.	Air Conditioning			Yes
16.	Power Steering		Yes	Yes
17.	Fog Lamps	Yes	Yes	Yes
18.	Cruise Control			Yes
19,	Heated Front Screen	Yes		Yes
20.	Leather Upholstery			Yes
21.	Run Flat Tyres			Yes
22.	Electrical Adjustable Seats			Yes
23.	Adjustable Steering Column			Yes
24.	Central Locking	Yes	Yes	Yes
25.	Self Leveling Suspension			Yes
26.	Towing Package			Yes
27.	Limited Slip Differential			Yes
28.	Sports Steering Wheel	Yes		
29.	Low Profile Tyres		Yes	Yes
30.	CD Player	Yes	Yes	Yes

			data on	COMPET	TITION -	Year 3	E.		
1. SALES I	DISTRIBU	TION							
Team				Sales	By Size	Group			
	Small		L. Me	:d	U. Me	ď	Luxury		
1 1 1			11423	0 *	14247	5 *	5983	*	
2 2 2			16208	14 *	17355	4 *	9610	*	
3 3 3	110245	*	11580	1 *	12291	1 *			
3			12223	85 *	15/15	0			
Foreign S	ales 2165398	\$	309219	96	251522	9	974787		
*.Models	in stock								
2. MARKET	DISTRIE	UTION	Produc	t Pric	es (£) /	Market S	Share (%)		
Team		Sma 11		L. M	ed	U. Me	ed	ເມນ	ry
1 1 1				9500	3.17	12000	4.61	36000	0.60
2 2 2				9250	4.49	11500	5.61	32000	0.97
3 3 3 3	Ę	5000	4.84	9000	3.21	11500 8500	3.98 4.44		
3				8000	3.39				
3. MARKET	SECTORS	5							
Team			Market	Sector	Of Model				
	1	2		3	4	5			
1. 2. 3.	6 6 5	11 11 6		16 16 11	7	10			
4. COSTS			B	asic Ma	terials C	costs (£)		
Team					Hode 1				
	1		2		3		4	5	
1 2 3	5970 5187 3183	.57 .39 .17	8677. 7663. 5187.	99 20 39	13685.41 11200.06 7663.20	76	63,20	5187	39

!′

5. MODEL SPECIFICATIONS

Team 1	EXTRAS S	SELECTED	
	MDL 1 Felix	MDL 2 Tom	MDL 3 Garfield
1. Larger Engine Option	Yes	Yes	Yes
5-Speed gearbox	Yes	Yes	Yes
Fuel Injection	Yes	Yes	Yes
4. Alloy Wheels	Yes	Yes	Yes
5. Sunroof	Yes	Yes	Yes
Headlamp Washwipes			Yes
7. Spoiler Set	Yes	Yes	Yes
8. Metallic Paint	Yes	Yes	Yes
9. Electric Windows	Yes	Yes	Yes
10. Electric Aerial	Yes	Yes	Yes
11. Automatic Transmission			Yes
Anti-lock Braking (ABS)	Yes	Yes	Yes
13. Electric Seat Heating			Yes
14. Electric Wing Mirrors			Yes
15. Air Conditioning			Yes
16. Power Steering		Yes	Yes
17. Fog Lamps	Yes	Yes	Yes
18. Cruise Control			Yes
19. Heated Front Screen	Yes		Yes
20. Leather Upholstery			Yes
21. Run Flat Tyres			Yes
22. Electrical Adjustable Sea	ts		Yes
23. Adjustable Steering Colum	n		Yes
24. Central Locking	Yes	Yes	Yes
25. Self Leveling Suspension			Yes
26. Towing Package			Yes
27. Limited Slip Differential			Yes
28. Sports Steering Wheel	Yes		
29. Low Profile Tyres		Yes	Yes
30, CD Player	Yes	Yes	Yes

....

Team 1 has no R & D projects online.

No extras have been selected by team 2.

Team 2

RESEARCH AND DEVELOPMENT PROJECTS

		MDL 1 Hai 6	MDL 2 Hai 11	MDL 3 Hai 16
1.	Aerodynamic Remodelling	Online	Online	Online
3.	Facelift	Online	Online	Online
4.	Turbo Charging	Online	Online	Online
8.	Four Wheel Drive	Online	Online	
9.	Diesel Engine	Online	Online	Online
10.	Electronic Engine Control	Online	Online	Online
12.	Multi Valve Engine	Online	Online	Online
13.	Improved Build Quality	Online	Online	Online
14.	Automatic Crash Restraint	Online	Online	Online
15.	Stronger Passenger Cell	Online	Online	Online
16.	Catalytic Converter	Online	Online	Online
18.	Anti Engine Noise System	Online	Online	Online
20.	Breathalyser Interlock	Online	Online	Online
21.	Drowsiness Detection	Online	Online	
22.	Automatic Window Wipers		Online	On11ne

No extras have been selected by team 3.

Team 3 RESEAR	CH AND DEVELOPMENT PROJECTS				
	MDL 1 Exp 5	MDL 2 Exp 6	MDL 3 Exp 11	MOL 4 Exp 7	MDL 5 Exp 10
3. Facelift 21. Drowsiness Detection	Online	Online	Online Online	Online Online	Online

6. PR(DUCTION				
Team	Total	Productivity	Productivity	Days Lost	Weekly
	Workforce	Index	(cars/man/year)	To Strikes	Salary (£m)
1	40000	1.04	8.05	10	235
2	20000	2.07	16.05	7	270
3	60000	1.04	10.44	2	220
Team	Sales (£) / Employee	Pre Tax Pre Employee (1	ofit / E)		

1 75256.82 6545.41 2 190133.40 20088.95 3 85843.61 1164.26

7. PROMOTION BUDGET

Team	(£m)		

5

1	68.10
2	53.30
3	126.00
2 3	126.00

いちないないというないないない

ACC. Shitter was been and

Issue 3 of the Car Journal

Profits for team 1 were £170.18m Profits for team 2 were £261.16m Profits for team 3 were £45.41m

Economists predict that next year inflation will be around 4%.

TUTORS REPORT - Year 3

Team	Sales	Gross	Operating	Post Tax	Stock	Shareholders
	(£m)	Profit (£m)	Profit (£m) Profit (£m)	Value (£m)	Funds (£m)
1	3010.27	664.31	232.77	170.18	845.91	1255.20
2	3802.67	1265.99	453.13	261.16	118.06	930.09
3	5150.62	939.66	92.11	45.41	212.89	712.08
Team	Gross Margin	Sales (%) Margi	Cu n (%) Ra	rrent Quick tio Ratio	Return on Assets (%))
1 2 3	22.07 33.29 18.24	7.7.	3 12 2 5	.65 3.42 .17 4.33	17.39 24.14	

and Smell & 5.5

い水

2. 2. 2. 2. W. 2. 5.

41

NODEL DETAILS BY TEAM

Team 1 - CAT Motors			
Model Name	Felix	Tom	Garfield
Market Sector	6	11	16
Workforce	15000	15000	10000
Sale Price	9500.00	12000.00	36000,00
Basic Cost	5187.39	7663.20	11200.06
Cost of Extras	783.18	1014.80	2485.35
Gross Margin (%)	25.67	18.60	31.67
Produced	155806	155806	10380
Cars Sold	114230	142475	5983
Cars in Stock	62007	23241	8683
Market Share (%)	3.17	4.61	0.60

Team 2 - Hai Tek Motors

Model Name	Ha1 6	Haj 11	Hai 16
Market Sector	6	11	16
Workforce	7500	7500	5000
Sale Price	9250.00	11500.00	32000.00
Basic Cost	5187.39	7663,20	11200.06
Cost of Extras	0.00	0.00	0.00
Gross Margin (%)	37.00	27.80	44,99
Produced	155344	155344	10350
Cars Sold	162084	173554	9610
Cars in Stock	5505	6618	2412
Market Share (%)	4.49	5.61	0.97

Team 3 - Expansion Motors Inc.

Model Name	Exp 5	Exp 6	Exp 11	Exp 7	Exp 10
Market Sector	5	. 6	- 11	. 7	10
Workforce	12000	12000	12000	12000	12000
Sale Price	5000.00	9000.00	11500.00	8500.00	8000.00
Basic Cost	3183.17	5187.39	7663.20	7663.20	5187 39
Cost of Extras	0.00	0.00	0.00	0.00	0.00
Gross Margin (%)	15.27	30.66	24.20	-2.55	21,99
Produced	125311	125311	125311	125311	125311
Cars Sold	110245	115801	122911	137156	122235
Cars in Stock	17851	9510	2400	4885	3076
Market Share (%)	4.84	3.21	3.98	4.44	3.39

.'-

DATA ON COMPETITION - Year 3

1. SALES	DISTRIBUTION			
Team		Sales By	Size Group	
	Sma 11	L. Med	U. Med	Luxury
1 1 1		114230 *	142475 *	5983 *
2 2 2		162084 *	173554 *	9610 *
3333	110245 *	115801 *	122911 * 137156 *	
3		122235 *	15/150	
Foreign	Sales 2165398	3092196	2515229	974787

* Models in stock.

2. MARKET DISTRIBUTION Product Prices (£) / Market Share (%)

Team	Sma 11	I	L. Me	ed	U. Me	ed	Luxur	у
1 1 1			9500	3.17	12000	4.61	36000	0.60
2 2 2			9250	4,49	11500	5.61	32000	0.97
3 3 3	5000	4.84	9000	3.21	11500	3.98		
3			8000	3,39	8500	4.44		

3. MARKET SECTORS

Team Market Sector Of Model

	1	2	3	4	5
1. 2. 3.	6 6 5	11 11 6	16 16 11	7	10

4, COSTS

		Basic	Materials Cos	ts (£)			
ſeam	Mode 1						
	1	2	3	4	5		
1 2 3	5970.57 5187.39 3183.17	8677.99 7663.20 5187.39	13685.41 11200.06 7663.20	7663.20	5187.3		

5. MODEL SPECIFICATIONS

Team 1		EXTRAS SELECTED			
		MOL 1 Felix	MDL 2 Tom	MDL 3 Garfield	
1.	Larger Engine Option	Yes	Yes	Yes	
2.	5-Speed gearbox	Yes	Yes	Yes	
3.	Fuel Injection	Yes	Yes	Yes	
4.	Alloy Wheels	Yes	Yes	Yes	
5.	Sunroof	Yes	Yes	Yes	
6.	Headlamp Washwipes			Yes	
7.	Spoiler Set	Yes	Yes	Yes	
8.	Metallic Paint	Yes	Yes	Yes	
9.	Electric Windows	Yes	Yes	Yes	
10.	Electric Aerial	Yes	Yes	Yes	
11.	Automatic Transmission			Yes	
12.	Anti-lock Braking (ABS)	Yes	Yes	Yes	
13.	Electric Seat Heating			Yes	
14.	Electric Wing Mirrors			Yes	
15.	Air Conditioning			Yes	
16.	Power Steering		Yes	Yes	
17.	Fog Lamps	Yes	Yes	Yes	
18.	Cruise Control			Yes	
19.	Heated Front Screen	Yes		Yes	
20.	Leather Upholstery			Yes	
21.	Run Flat Tyres			Yes	
22.	Electrical Adjustable Seats	5		Yes	
23.	Adjustable Steering Column			Yes	
24.	Central Locking	Yes	Yes	Yes	
25.	Self Leveling Suspension			Yes	
26.	Towing Package			Yes	
27.	Limited Slip Differential			Yes	
28.	Sports Steering Wheel	Yes			
29,	Low Profile Tyres		Yes	Yes	
30.	CD Player	Yes	Yes	Yes	

Team 1 has no R & D projects online.

No extras have been selected by team 2.

Team 2 RESEARCH AND DEVELOPMENT PROJECTS

		MOL 1	MDL 2	MDL 3	
		ilai 6	Hai 11	Ha1 16	
1.	Aerodynamic Remodelling	Online	Online	Online	
з.	Facelift	Online	Online	Online	
4.	Turbo Charging	Online	Online	Online	
8.	Four Wheel Drive	Online	Online		
9.	Diesel Engine	Online	Online	Online	
10.	Electronic Engine Control	Online	Online	Online	
12.	Multi Valve Engine	Online 0	Online	Online	
13.	Improved Build Quality	Online	Online	Online	
14.	Automatic Crash Restraint	Online	Online	Online	
15,	Stronger Passenger Cell	Online	Online	Online	
16.	Catalytic Converter	Online	Online	Online	
18.	Anti Engine Noise System	Online	Online	Online	
20.	Breathalyser Interlock	Online	Online	Online	
21.	Drowsiness Detection	Online	Online		
22.	Automatic Window Wipers		Online	Online	

No extras have been selected by team 3.

Team 3 RESEARCH AND DEVELOPMENT PROJECTS

	MDL 1	MDL 2	MDL 3	MDL 4	MDL 5
	Exp 5	Exp 6	Exp 11	Exp 7	Exp 10
2. Feelift 21. Drowsiness Detection	Online	Online	Online Online	Online Online	Online

٠

1.1. 3. C.

and the state of the second state and the second state

Team	Total Workforce	Productivity Index	Productivity (cars/man/year)	Days Lost To Strikes	Weekly Salary (Em)
1	40000	1.04	8.05	10	235	
2	20000	2.07	16.05	7	270	
3	60000	1.04	10.44	2	220	
	Sales (£)	Pre Tax Pre	ofit /			
Team	Employee	Employee (E)	·		
1	75256.82	6545.4	1			
2	190133.40	20088.9	5			
3	85843.61	1164.2	6			

Team	(£m)

1	68.10
2	53.30
3	126.00

Forewords

Bissim is a realistic and detailed simulation of a manufacturing company developed by Management Games Ltd. The BISSIM simulation/game allows up to 100 participants to take decisions in all the major areas of finance, production, and marketing. The structure of the game enables teams to move from the simple to the complex.

Game's description

The description of BISSIM will be done in the coming pages.

BISSIM PROSPECTUS

· · · · · · · · · · · ·

S to work

and the state of the second

A to the second se

. . . .

5.0

3 march 1



States and the second



PROSPECTUS

Directors: Pam Sykes, Philip Sykes A.C.I.S. Barrister

Regd. London 746499

The Master Game

CONCEPTS

BISSIM is the most realistic and detailed simulation of a manufacturing company ever offered to the general public. Its unique modular structure enables participants to move from the simple to the complex and from the known to the unknown. So, for such a complex simulation, it is very easy to assimilate. Adjudication is simple and speedy through use of a computer program written in "BASIC".

LEARNING OBJECTIVES

At the end of the simulation, participants will be able to:

- * describe the relationship between supply and demand in a competitive market
- describe the relationship of investment, risk and reward (or profit)
- * list the factors which can influence a company's sales and profits
- * outline at least three basic business strategies
- * describe the construction of the financial accounts of a company
- explain the process of business decision-making and the use of market and financial information

SIMULATION DESIGN

The exercise is run over 4 to 10 rounds, each representing one year's trading. In each round, each team of participants is presented with three or four new decisions in addition to their reconsideration of previous decisions. In this way, the detail is built up progressively during the exercise:

					Module 6
				Module 5	
			Module 4		
		Module 3			
	Module 2				
Module 1					

Time

W. 2 which a Com

The Master Game

SIMULATION DESIGN (cont.)

In each round the following procedure is repeated:



- 1) Firstly, the situation is analysed by participants.
- Then participants make their decisions concerning the running of their "company", and enter them on the relevant "Decision Form".
- 3) Those decisions are analysed by the computer model.
- The computer prints out the financial and other results for reappraisal. A sample print-out is attached to this prospectus.

Participants do not need to make a complete set of decisions in every round. If they do nothing, the computer will interpret that as an intention to pursue the "standard" decisions which are stored in the program. The program is interactive, i.e. the decisions of one company will have an impact on its competitors.

Page 2

The Master Game

and the second of the second

CONTENT

BISSIM comprises six modules of increasing complexity which are cumulative in the way they are built up, each containing the following decisions:

I

Module One: Introduction

- 1. Factory Output
- 2. Pricing
- 3. Wage Rates

Module Two: Production

- 4. Research & development
- 5. Productivity improvements
- 6. Quality control
- 7. Staffing levels and overtime
- Capital expenditure appraisal (factory extension and/or new equipment)
- 9. Industrial relations

Module Three: Marketing

- 10. Market segmentation and marketing strategies
- 11. Distribution channels
- 12. Media selection
- 13. Promotional expenditure

Module Four: Sales Management

- 14. Trade discounts
- 15. Salesforce size
- 16. Calling rates and market coverage
- 17. Sales training
- 18. Sales incentives

Moduel Five: Finance

- · 19. Equity and loan financing
- · 20. Share issues
- · 21. Dividend Rates
 - 22. Trade credit given and taken

Module Six: Customers & Society

- 23. After sales service (guarantees & service contracts)
- 24. Public relations
- 25. Donations
- 26. Pollution

The Master Game

Page 4

Reading to a fair and

在下来来,在这些情况,这些情况,就是"不少"。"这些"这",""。""。""。""。""。""。""。""。""。""。"""","""",""""。"""",""""。"")。""(")。"")。""

CONTENT (cont.)

All modules require the analysis of financial and market research data. The interpretation of financial accounts and management information is also featured in every module. A fascinating by-product of the exercise is the light which it sheds on teamwork, leadership and other facets of human relationships.

BENEFITS

BISSIM offers the following benefits to instructors:

- * "Model-based" simulations of this type have a reputation for being difficult to handle. Not so with BISSIM. This is the first such exercise which is as much fun for the instructor as for the participants!
- * This is achieved through ten years of development, an unusually flexible computer program and very thorough discussion notes.
- * BISSIM is flexible in its timings. A single round requires one or two hours, so as many rounds as you have time for can be worked into your programme (not necessarily consecutively). The total time occupied by the exercise can range from three to thirty hours.
- * It is flexible in terms of the number of participants from 3 to over 100 people!
- * The adjudication between rounds is simple, quick and accurate (an average of 3 minutes per team is all that is required!).

BISSIM offers the following benefits to participants:

- * It is an exciting, stimulating and challenging exercise (trainee motivation is never a problem).
- * It is easy to get to grips with at the beginning and is "graded" (becoming more difficult as it progresses).
- * It does not require high levels of numeracy to participate.

VALIDATION

BISSIM has been developed over 10 years from the practical experience gained with over 200 "live" runs of its manually operated predecessors - "Tycoon", "Finansim" and "Sellem", and from five previous editions of BISSIM itself.

The simulation is able to discriminate between teams of different abilities. Generally speaking, more able groups achieve higher profits. Users are asked to submit details of the results achieved so that a data bank of norms relating the financial results to age/salary or participants can be published. The detailed questions enable the instructor to judge the learning which has been achieved by participants.

The Master Game

and the second second second second

SCENARIO

BISSIM represents a typical company manufacturing a hypothetical consumer-durable product in the year 2021. This is a DOMESTIC ROBOT of (more or less!) human appearance, which is designed to carry out a wide range of domestic chores. At the beginning of the exercise, it looks like this:



During the exercise, participants have the opportunity to develop the specification of the machine and to offer different versions of it aimed at specialised market segments.

The market in which participants operate is a typical Western European "developed" economy of about 30 million people. Over the past five years, the company has established itself in the market but its financial position is still very weak.

The sales volume of the product over that period was as follows:

2016	2017	2018	2019	2020
500	1100	1400	1600	1500

The market demand fluctuates in accordance with a 4 to 5 year economic cycle but the combined marketing effort of all competing companies will affect the size of the market to some extent.

All the competing teams are identical at the beginning of the exercise. The financial and other results which the company achieved in the year 2020 are described in the attached computer printout. The existing policies of the company can be identified from the Decision Form for 2020, which is also attached.

The Master Game

SIMPLIFIED EXERCISE

For those who wish to carry out financial or "business appreciation" training at a less sophisticated level, our simple computerised exercise, "TYCOON" is designed to illustrate the basic concepts of supply and demand, competition and profit or loss in a manufacturing industry. The decisions required include: pricing, market research, promotional expenditure, staffing levels, wage rates, production schedule and capital expenditure (i.e. extending factory capacity). Each team then receives a print-out of a profit and loss account and balance sheet. The exercise requires from 6 to 10 hours.

OPEN LEARNING

"BISSIM" and "TYCOON" can be run for you as a postal competition. Used this way, participants can work in teams at their own place of work and in their own time. At regular intervals (e.g. weekly or monthly) they post their decisions to MGL who process them and send back to them a printout of their results (profit and loss account, balance sheet, indices of performance, etc.). The process is repeated over 5 rounds, each of which represents one year's trading.

MODIFICATIONS

Although most of the business principles illustrated in this context apply to every business, some instructors may find that the simulation is more readily accepted by participants if the scenario is closer to their own situation. Accordingly, the publishers will modify the rules and the program to suit ANY BUSINESS. A great deal of experience has been accumulated by MGL in the design of such simulations. In fact, simulations already exist covering:

> Airlines Banks Building & Construction Companies **Building Societies** Computer Companies Consultancy-type Businesses Engineering Companies Exporters Food Manufacturers Insurance Companies (General Business and Life) Motor Manufacturers Motor Traders Pharmaceutical Manufacturers Retail Traders (Newsagencies, etc.) Service Businesses (e.g. Linen Supply) Transport Companies Wholesalers

Any business environment can be simulated using the BISSIM structural framework of rules and computer program. Such modifications cost less than may be imagined.

and the second se

a li to a to the sail to the the to

The Master Game

THE PAGKAGE

The package comprises:

One Instructor's Manual - which provides a comprehensive guide to the use of the simulation.

<u>Twenty Participant Manuals</u> - containing the background, rules and decisions of the exercise. Alternatively, a copyright waiver is given.

Six Sets of Decision Forms

One Computer Program - written in "BASIC" and supplied as a disc for an IBM PC (Discs for many other machines may be supplied; please check with us for the current availability).

SUPPORT SERVICES

The publishers will provide all advice and assistance necessary to ensure that the simulation works well for you. Two days of training for your instructors OR conducting the first "live" run for you is available as part of the package.

Page 7

the second state of the second state of the second states

DECISION FORM

1

Team No:

Team Name: 21st CENTURY ELECTRONICS PLC. Year: 2020

14. A. A. B. B.

The second is a second and the second second

DECISIONS		AMO	UNT]
PRODUCTION DECISIONS		Model <i></i>	Mode	1 <11	>	1
Production Schedule	(Units)	1600			0	
Raw Material Quality	(1 - 4)	XXX_3	XX	X		1
Capital Expenditure: * Flow Line	(0003)				0	
Flex. Manufacture	(£000) ·	-+-			0	
R & D Key Project	(1 - 22)			1	3	
R & D 2nd Project	(1 - 22)		r	1	4	1
Productivity Expenditure *	(£000)		-		0	
Miscellaneous Expenditure * (including Market Research Donations, etc.)	(£000)				0	
PERSONNEL DECISIONS]
Production - Direct				7	5	
Production - Indirect (inc. Quality and Maintenance)				5	0	
Sales				1	0	1
Research and Development				3	5	
Administration (inc. Service)				3	0	1
Average Wage Rate	(£000)			1	0	0
Overtime Hours	(per Person per Week)			-	0	
				1	<u> </u>	1
MARKETING DECISIONS		Model <i></i>	Mode	1 <11	>	
Average Price of all Versions	(£)	5000			0	
Principal Market Segments		XX 3 0	XX			
Promotional Expenditure	(£000)	1 0				
Sales Incentives	(000£)	1 0				
Price (Contract)	(£)	0				
Training Expenditure	(000£)	1				1
Key Outlets	(1 - 7)	XXX4	XX	X		1
Trade Discount	(%)	XX_{20}	XX	T		
Guarantee Period	(Months)	XX_06	XX			1
R & D Projects Allocated		0				
No. of Versions		XX 1	XX	1		1

Form 2

DECISION FORM (Cont.)

Team No:

Team Name: 21st CENTURY ELECTRONICS PLC.

Year: 2020

2.7.

Sundary

Canada - Sautha - Sautha

The work

Allen & L. L. L.

and a state of the second and a second second

* 16 . N. . N. Ye ...

DECISIONS	AMOUNT					
FINANCE DECISIONS						
Shares Offered *	(000's)					0
Offer Price *	(£)					0
Dividend	(%)				1	0
Debtor Period (Customers)	(Weeks)				0	4
Creditor Period (Suppliers)	(Weeks)				1	2
Overdraft Application	(000£)		1	0	0	0
FOR ADJUDICATOR'S USE ONLY						
Marketing Index						
Staff Morale						
Productivity Index						
Quality Index						
Credit Rating						
Interest Rate						
Social Index						
Sales Management	•					
Cash Flow	(+/- £000)					
Government Contract Volume	(Units)					
Government Contract Product	(I or II)					

BUDGET FORM			
Forecast of Net Profit (Pre Tax) *	(000£)		
Forecast of Bank Overdraft or Cash Balance *	(+/- £000)		

Write on this form only those decisions which you wish to CHANGE. Otherwise, the computer program assumes the continuance of present policies (except for items marked with an asterisk which are of a "one-off" nature).

Form 2 (Cont.)

21st. CENTURY ELECTRONICS Pic.

PROFIT & LOSS	ACCOUNT		BALA	NCE SHE	ET
#000'S	#000'S	1	ting any pay par par for the and all and and any and had be	#000'S	#000'S
SALES TURNOVER (after discount 1507)	6028)		SHARE CAPITAL RESERVES	-	100 2332
(Incl. Incelest o	7		CAP. & RESVS.		2432
EXPENSES:					
Prodn. labour 1250			FIXED ASSETS		2430
Marketing 121 Research 350			TRADE INVESTI CURRENT ASSETS	IENTS	0
Guarantee costs 340 Admin & service 300			Stock Debtors	854 772	
Financial 49			Cash	0	
Distribution 177 Overhead 995			Total	1626	
TOTAL EXPENSES	6387		CUR. LIABILIT	IES:	
			Creditors	792	
NET PROFIT	-359		Overdraft	832	
laxation			Total	1624	
FROFIT AFTER TAX	-359				~
Dividends			NET COR. ASSE	15	
RETAINED PROFIT	-359		NET ASSETS		2432
			THUSTOPO-		

KEY RATIOS:

NET MARGIN	-6.0	%	
GROSS MARGIN	54	%	
R. O. C. E.	-14.8	%	
MARKET SHARE		%	
OUTPUT, PER PERS.	7.7	units	
LIQUIDITY RATIO	47	%	
STOCK TURN	6	times p.a.	
EARNINGS P. SHARE	0	P.	

PRODUCTION & FINANCE:

FACTORY	PROD.	(1> 15	50 units
	< 1	(1 > 0)	units
UNSOLD	STOCK	<1> 30	0 units
UNSOLD	STOCK	(1> 0	units
UNIT PR	DD COST	(I> 18	87 (#)
UNIT PRO	DDCOST<:	II > 0	(井)
CAPACIT	Y UTILN	. 10	0 %
FUTURE	CAPAC	ITY 15	50 units
SHARE	PRICE	# 26	.75
R & D KI	EY PROJE	ECT 13	= 95 yrs.
R & D 2	nd PROJ	ECT 14	= 1 yrs.
COMPLET	ED R &	D	

			a case only provide pr
MARKET	IMAG	ΞE	5
FRODUCT	QUA	LITY	3
STAFF	MORAL	.E	З
CREDIT	RATI	NG	3
PUBLIC	IMAG	θE	3
SALES	EFFIC	LENCY	1
AFTER-S	BALES	SERVICE	0
VALUE	FOR	MONEY	10

いいいいいの あいても ちょうちょうちょう あいろいろ

and the second state of the second state and the second second second second second second second second second

MARKETING	& PE	RSONNE	L:	
SALES	<1>	1507	uni	ts
SALES	<11>	0	uni	ts
OST SALES	<15	0	uni	ts
OST SALES	< F1 >	0	uni	ts
CUSTMS. & PROSP	TS.	1992		
CALLING RATE		600	pp	pa
DIRECT PRODN.	STAFF	75		
INDCT. PRODN. S	TAFF	50		
SALES STAFF		9		
R. \$ D. STAFF		33		
ADMIN. STAFF		30		

30 14 %

198

Ş

9

SEGMENTS

CONCLUSION

The workshop of Management Simulations held at the Management Centre, University of Bradford, was a very interesting experienced.

Its interest was based on the possibility to see four of the most successful simulationgames.

Among the simulations shown, Network Proteus was, in my opinion, the most interesting one. The reasons for that are :

◆ Network Proteus was shown as a non-numerical based simulation (modern approach), where the main issue to experience was the existing departmental interaction. It was not, as the other three, a mere calculation game (traditional approach).

• Network Proteus is a dynamic interactive simulation, where interactions among roles in a factory take place while running the simulation, not before (i.e. fill in a form or enter information on a computer).

• Network Proteus is a "real" simulation, where the product being produced has a real bill of materials, its produced in different machines for "real" known customers, and in a real manufacturing market.

• Network Proteus is able to generate "real" problems, that the different teams will have to face.

There are also some disadvantages associated :

• Network Proteus has a very complicate interrelation structure, there two many people to contact.

• Network Proteus has a too long cycle time, something that leads to a very long teaching/training process. Something not many courses can afford.

◆ Although it is said to work in a real time environment were people can make decisions or not, that methodology is virtually the same as changing or not changing a decision - a capacity provided in virtually all games [77].

Other contacts

The next pages show some of the contacts held during the fulfilment of this project.

OTHER CONTACTS

all's a server a "synds is a shirthall as

.

hter ville server

1

5.14.1 9 94.14

·* ·** ·

4 10

1. 2 2 5

Managment Workshop Programme, Management Centre, Emm Lane, Heaton, Bradford, BD9 4JL.

Jaun Ignacio Igartua, Nottingham Polytechnic, Dept. Industrial & Production Engineering, Burton Street, Nottingham, NG1 4BU.

17th. January 1991

あんとういいいい してあたち ちちちちち ちちちちち あいろう ひんちち またたり たいち きちゅう ちちたいちょう しゅうしょ

Dear Jaun,

Referring to your letter of 9th. January. Since the publication of the article in *"Perspectives on gaming and simulation 14"*, a new BUSGAM version 2 has been completed and is in use at the Management Centre on undergraduate Business Studies and Electrical Engineering, postgraduate MBA and Manufacturing Technology, and Executive Development courses. The new game is multi-product / market segment and adaptable for a number of different case studies.

In response to your questions, this type of game rapidly becomes technically difficult to design and develop once one wishes to do anything non-trivial, as the problem is to implement a real-time multi-user database, with considerations of response, locking, etc. On our network (286 PCs, 386 Server, MSNET, EtherNet) performance becomes unacceptably slow with around eight teams. Having said that, we can still put 50 or more participants through at one session (with one trainer) which is extremely resource effective compared to conventional games.

Evaluation of the game through participant questionnaire surveys, observers, etc. has lead us to conclude that generally students prefer BUSGAM to other business games they have taken part in, although this may of course, be because of its novelty value. The time dimension, with the relentless increasing clock is a strong motivating factor to maintain interest and heighten enjoyment, but we suspect this can lead to reactive behaviour; i.e. responding to events without planning. One of the key objectives of education is to enable the student to become proactive, aware and in control. Hence we now incorporate a practice session before the game, and also split up the sessions to allow reflection; i.e. thinking as well as doing. Careful working through of learning objectives is very important.

As BUSGAM is a strategic level business game and each team / company has its own computer, you may find that your plans have more in common with other games. For example, Network Proteus at Manchester Business School is being developed and will be a rewrite of their mainframe version. There are four roles for participants to run a company: Production, Finance, Marketing and Managing Director, plus a trainer's interface. Another contact may be the Production and Operations Management group, here at the Management Centre. My colleague Malcolm Afferson has worked on the development of a computerised Production Management system and associated training package for local companies, funded by the Training Agency.

I hope that I have been of help.

Yours sincerely,

C.L.N. Gray

(Mr. Carey Gray, Enterprise Initiative Fellow)



-

UNIVERSITY OF BRADFORD

MANAGEMENT CENTRE EMM LANE BRADFORD WEST YORKSHIRE BD9 4JL UK

Telex 51309 UNIBFD G Fax (0274) 546866 E-mail ghictiaUK.AC,BRADFORD

COMPUTERS IN TEACHING INITIATIVE CENTRE FOR BUSINESS AND MANAGEMENT STUDIES Coordinator: RICHARD WELFORD BA MA PhD PGCE ABIM

1 February 1991

Mr J I Igartua Nottingham Polytechnic Industrial & Production Engineering Dept Burton Street NG1 4BU

Director and Professor of Management DAVID T H WEIR MA DipPSA FBIM

Dear Mr Igartua

Workshop on Management Simulations - 6th February 1991

I am pleased to be able to confirm your place at the above workshop has been reserved.

I enclose a map showing the location of the Management Centre. The workshop will be held in the Heaton Mount building, which is accessed via Keighley Road.

The timing of some of some of the sessions has had to be amended slightly, which will probably mean finishing the day approximately half an hour later than publicised, and I trust this will not inconvenience you too much.

I look forward to seeing you on the 6th.

Yours sincerely

Holnes

Cillian Holmes CTI Information Officer

Sponsored by the Institute of Chartered Accountants in England and Wales and the Computer Board for Universities and Research Councils

APPENDIX C

GAME'S SOFTWARE

Factory Running Sub-program » SYS11.PRG Production Planning/Control Sub-program » PLA4.PRG Marketing/Sales Sub-program » SAL6.PRG

FACTORY RUNNING SUB-PROGRAM

SYS11.PRG

1	*.*	********
2	*	This program is the one that simulates
3	*:	Program: SYS11.PRG all the factory activities.Together with
4	*:	the Sales and Planning ones conforms the
5	*.	System: SISIL RG Notingnam Polytechnics's
7	*	Copyright (c) 1991. NOTTINGHAM POLYTECHNIC
8	*	Last modified: 08/05/91 17:36
9	*:	
10	*:	Proces & Fnotes: ERR FIX
12	*:	: GAMCONF • OPEN
13	*	
14	*:	: FACTCONF
15	*:	: STARTGAM
16	*	: RESET
19		: SEI - Setneyt
19	*	· NEXTDAY
20	*:	: EXITGAME
21	*:	: QUITGAME
22	*:	: CONTINUE
23	*.	: RUNFACI - RACTSTOD
25	*	: SAVEDAY
26	*	RESTART
27	*:	: DAYREPOR
28	*:	: SALREPO
29	- T:	: PLAREPO
31	*	- GOODSERV
32	*	RUNSCREN
33	*:	: SAYTIME0
34	*:	: SAYTIME1
35		: TIMECAL
37	*	: ENOGEN
38	*	SHIPPING
39	*:	: REPOMENU
40	*:	: FREQUEN
41		SALPAST
42	÷.	: SALPERP • DI ADAST
44	*	: MACHBROW
45	*	: MACH_DISP
46	*:	: NEWMACH
47	*:	: MACHEDIT
40	*.	
50	*	: MACH REPL
51	*:	: MACH RSTR
52	*:	: MACH_STOR
53	*:	: REINDEX
55	*.	COPENSAL
56	*:	: OPENDIA
57	*:	: SYSTEM
58	*:	: GETKEY
39	*: *:	: EXITSYS
61	*	; FIELDD • DESETTID
62	*:	: ONTIME
63	*:	: ENQUSET
64	*:	: ADDREC
65	*:	: RESTVAR
67	*	
68	*	: CREATSTAT
69	*	: RENSTAT
70	*:	: REPLSYS
71	*:	: CREATSTOR
72	*:	: CKEAIKEPL - DEI ETC
74	*	
75	*:	Calls: ERROR() (FOXBASE+ function)
76	*:	: MESSAGE0 (FOXBASE + function)
77	*	: EKK FIX (procedure in SYS11.PRG)
/8 70	*	: CIOD() (FOXBASE+ function) : GAMCONE (providence in \$2511 DPG)
80	*	: OPEN (procedure in S1511.PRG)
81	*:	: MAINMENU (procedure in SYS11.PRG)
82	*:	
83	*:	Uses: SYSTEM.DBF

S

and a string to war a to the string

14. m.

•

N.

· · · · · · ·
Documented 08/05/91 at 18:22 85 FoxDoc version 1.0 86 * **** 87 also also 88 set escape on 89 set talk off 90 91 92 93 94 95 96 97 98 99 set heading off set bell off set scoreboard off set status off set color to set delete on clear set color to bu +/n@ 7,24 to 12,54 double 100 set color to gr+/n @ 9,30 say "SETTING THE SYSTEM" 101 2,50 say SETTING THE S @ 10,34 say "PLEASE WAIT" ** 102 103 104 105 set date british set procedure to sys11.prg 106 107 108 on error do err_fix with error(), message() set default to v: ** public mfound,mresourcod,mdescript,mcapacity,msetup1,msetup2,msetup3,; msetup4,msetup5,srnlines,difclock,ptlines,pout,factclock,factclock1 public facthh,factdd,factmn,factyy,produce,ctde1,confdate,seed,; factmin,factstart,tempdd,startstop,mmfacttime,mmfactdate,number,newclock public difstop,slack,plow,tempdd1,facthhh,reso,actsetup,lstsetup; r,c,screenatr,statusatr,windowatr,promptair,hiliteatr,factdaten,daysenqu public startday,startmon,startyy,enqurec,conffdate,mwaittime,oldrec,toprec,; botrec,tt,xx,mpla,mmar,mmac,mstk,mbil,mstu,msta,msalp,msalf,mstkp,mmacp,mplap public msys,pfiles,first,mloop,var,fil 109 110 111 112 113 114 115 116 117 118 screenatr="R+/N,N/W" statusatr="BU/N,N/W" windowatr="R+/N,N/W" promptatr="GR+/N,N/W" hiliteatr="N/W" 119 120 121 122 123 124 125 mloop=0126 127 128 var= ** srnlines = 20129 prtlines=55 130 pout=srnlines 131 seed=0 132 daysenqu=0133 134 135 136 oldrec = 1 botrec = 1factdaten = ctod("00/00/00") first=.t. 137 138 facthh=0 139 *mwaittime=0 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 select 10 set exclusive off use system do gamconf do open close databases set view to run select 10 use system replace gam_stop with .f. unlock 156 157 158 159 do mainmenu ******* *1 * Procedure: MAINMENU This procedure shows the available options when starting the program. 160 +! 161 *i 162 *! Called by: SYS11.PRG *! *! 163 (procedure in SYS11.PRG) (procedure in SYS11.PRG) 164 Calls: FACTCONF 165 : STARTGAM *! 166 167 : EXITSYS (procedure in SYS11.PRG) * 168

Sand Rearies & Very Sand Strate State State

Sugar.

いたい あいたい いちの いち あいのい い

The set of the set of the set of

and a set of a large

APRIL L. Martak

procedure mainmenu

The state of the s

Set.

169

170 171 ***** set color to 172 173 174 175 176 177 178 clear do while .t. clear m_menu=0 set color to g/n@ 3,16,19,62 box " r = | = | = | = | = |(g) 5,10,19,02 b0x
(g) 5,10,19,02 b0x
(g) 5,20 to 7,53 double
set color to bu/n
(g) 6,29 say "SYSTEM GAME WORKAREA"
set color to br/n
(g) 10,041 to 54 179 180 181 182 183 184 185 186 187 @ 10,24 to 16,54 @ 10,24 to 10,34
@ 12,27 prompt "1.FACTORY CONFIGURATION."
@ 13,27 prompt "2.RUN THE GAME. "
@ 14,27 prompt "3.EXIT FROM SYSTEM. " 188 menu to m_menu 189 =do case 190 191 192 193 194 case m menu=1 do factconf case m_menu=2 do startgam case m_menu=3 195 do exitsys 196 exit 197 F =endcase L 198 enddo 199 200 201 * *!*** ***** *1 Procedure: FACTCONF This procedure allows the game manager modifying the machines' capacity and setup. Called by: MAINMENU (procedure in SYS11.PRG) 202 +j 203 * 204 205 *! *! *1 206 Calls: MACHBROW (procedure in SYS11.PRG) 207 208 *į procedure factconf 209 210 211 212 do while .t. clear 213 214 215 m_menu=0 set color to g/n@ 3,16,17,62 box " _ | = = | " set color to gr/n@ 5,26 to 7,53 double 216 217 218 219 220 221 222 set color to bu/n @ 6,28 say " FACTORY CONFIGURATION" set color to br/n @ 10,24 to 16,54 set color to bg/n @ 12,29 prompt "1.MACHINE FILE." @ 13,29 prompt "2.EXIT. " 223 224 225 226 227 228 229 230 menu to m menu do case =case m_menu=1 do machbrow case m_menu=2 < =return 231 232 233 234 235 236 237 238 endcase enddo * **** *! *1 Procedure: STARTGAM This procedure allows the system manager *1 to start running the gaming-simulation. Called by: MAINMENU (procedure in SYS11.PRG) *! 239 240 241 242 +į E (procedure in SYS11.PRG) (FOXBASE+-function) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) (procedure in SYS11.PRG) *! **Calls: CONTINUE** : INTO : VALO *1 *! 243 244 245 *1 : SUBSTRO * : DTOCO +1 : LEFTO 246 *1 : RIGHŤ() 247 *! : RUNFACT 248 * 249 *1* ***** and the state and sta 250 251 252 procedure startgam =do while .t. 253 clear 254 $m_{menu} = 0$

208

2471.78 . Ry. 6. 12.04

astra with State

255 256 257 258 set color to g/n @ 3,16,17,62 box " [a | === = | " set color to gr/n @ 5,26 to 7,53 double 259 set color to bu/n 260 @ 6,33 say "RUN THE GAME" 261 set color to br/n 262 @ 10,24 to 16,54 263 set color to bg/n @ 12,29 prompt "1.RUN THE GAME. " @ 13,29 prompt "2.EXIT BACK TO MAIN." 264 265 266 267 menu to m_menu =do case 268 case m menu=1 269 do continue 270 271 272 273 store int(val(substr(dtoc(confdate),4,2))) to startmon store int(val(left(dtoc(confdate),2))) to startday store int(val(right(dtoc(confdate),2))) to startyy do runfact 274 case m_menu=2 275 276 =return L endcase 277 278 279 ≈nddo ** *! 280 *! 281 +į Procedure: RESET This procedure resets the gaming-simulation's 282 *į time/date and production variables. 283 *! 284 *! 285 *! Called by: GAMCONF (procedure in SYS11.PRG) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) 285 286 287 Calls: INT() *! : VALO : SUBSTRO *! (FOXBASE+ function) (FOXBASE+ function) : DTOC() : LEFT() 288 * 289 *! 290 *! : RIGHŤ() (FOXBASE+ function) 291 *! *1* 292 293 294 procedure reset 295 store 0 to factmin, newclock, factclock1, factclock, facthh, difclock 296 store 0 to difstop, startstop, produce, tempdd, tempdd1, daysenqu 297 store 1 to plow store i-brequency to slack store int(val(substr(dtoc(j->factdate),4,2))) to factmm store int(val(left(dtoc(j->factdate),2))) to factdd store int(val(right(dtoc(j->factdate),2))) to factyy 298 299 300 301 302 return ** 303 304 305 ***** ***** *1 306 *1 306 *1 307 *1 308 *1 Procedure: SET This procedure resets the gaming-simualtion's running elements when the game is exit or quit. Called by: EXITGAME (procedure in SYS11.PRG) : QUITGAME (procedure in SYS11.PRG) 309 +j 310 *1 : EXITSYS (procedure in SYS11.PRG) 311 * 312 *! Calls: SPACE() (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) 313 *! : EOF() 314 315 * : .NOT.EOF() *1 316 *þ 317 procedure set 318 319 clear 320 set color to bu+/n 321 @ 8,24 to 12,54 double 322 323 324 325 set color to gr+/n @ 10,29 say "THE GAME HAS FINISHED" select 10 326 327 replace gam stop with .t. replace sys stop with .t. replace salreport with .f. 328 329 replace plareport with .f. 330 331 332 333 replace neweng with .f. replace neweng with .f. replace newengrec with space(9) replace goodserv with .f. replace factime with "0:00" replace residue with .f. replace restatus with .f. 334 335 336 337 replace statreso with replace statcode with "----" 338 replace waittime with (factdate-startdate)*480 339

and the second of the second s

A and Participant of a lost

the state of the second s

340 341 342 factdaten=factdate replace factdate with factdaten replace throughput with 0 replace systock with 0 343 344 345 unlock select 1 346 347 348 349 350 351 352 353 354 355 356 357 358 359 goto top to top edo while .not, eof() replace ord_stat_0 with ord_stat_1 replace product 0 with product 0+product_1 f ord_stat_1="FINISHED" replace areduct 1 with product_0 replace product_1 with product_0 endif unlock -if .not.eof() skip endif =enddo select 3 replace all t avail with 0 360 unlock 361 select 5 362 replace all num_enqu with 0 363 364 365 unlock return 366 * **** ***** This procedure sets the gaming-simulation's Procedure: SETNEXT running elements in order to start the next 367 *j 368 *! 369 *! running day. Called by: NEXTDAY 369 370 371 *i (procedure in SYS11.PRG) *į 372 373 (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) *! Calls: EOF() *! : .NOT.EOF() *! : SPACE() 374 +į 375 *!******** 376 377 378 procedure setnext 379 clear 380 select 1 381 goto-top 382 383 384 385 replace product_1 with product_0 endif 386 387 388 unlock 389 -if .not.eof() 390 391 392 skip endif enddo 393 select 3 394 replace all t_avail with 0 395 unlock 396 397 select 5 replace all num_enqu with 0 398 399 400 401 unlock select 10 replace sys_stop with .f. replace salreport with .f. replace plareport with .f. replace newenq with .f. replace newenqrec with space(9) 402 403 404 405 406 407 408 replace goodserv with .f. replace facttime with "0:00" replace restative with f. replace restative with f. replace restative with f. replace statreso with "---409 410 replace statcode with "----" 411 412 replace waittime with (factdate-startdate)*480 413 factdaten = factdate 414 415 416 417 replace factdate with factdaten replace throughput with 0 replace systock unlock with 0 418 select 1 419 set color to 420 clear 421 return ** 422 423 ****** 424 *1

210

Make and some

Procedure: NEXTDAY This procedure calls the procedures needed to run the next running day. Called by: REPOMENU (procedure in SYS11.PRG) 425 *! +į 426 *! 427 428 *į 429 * Calls: GAMCONF (procedure in SYS11.PRG) 430 *! : SETNEXT (procedure in SYS11.PRG) 431 * 432 procedure nextday 433 434 435 do gamconf 436 do setnext 437 return 438 *!***** 439 **************** 440 441 442 *! This procedure allows the system manager to Procedure: EXITGAME exit the gaming-simulation once a day has * finished. *! 443 * Called by: REPOMENU (procedure in SYS11.PRG) 444 *1 *1 445 Calls: SET (procedure in SYS11.PRG) 446 447 448 *1 * ******* ****** procedure exitgame 449 450 do set 451 close all 452 clear all 453 return to master 454 455 456 *1 This procedure allows the system manager 457 *į Procedure: QUITGAME to quit the gaming-simulation after it has 458 *! been frozen. *! 459 Called by: FACTSTOP (procedure in SYS11.PRG) 460 *! 461 *! Calls: SET (procedure in SYS11.PRG) *1 462 463 * ******* procedure quitgame 464 465 466 do set 467 468 close all clear all 469 return to master 470 471 ***** This procedure allows the system manager Procedure: CONTINUE to continue with the gaming-simulation after *! 472 473 *1 *! 474 it has been frozen. *! 475 476 Called by: STARTGAM (procedure in SYS11.PRG) (procedure in SYS11.PRG) * : FACTSTOP 477 *! 478 * ****** 479 procedure continue 480 481 select 10 482 replace sys_stop with .f. 483 unlock 484 set color to 485 clear 486 return 487 488 *!**** ********** This is the main "running" procedure, which controls Procedure: RUNFACT the different tasks of the gaming-simulation. Its functioning is thorougly explained in one of the thesis's sections. 489 *! 490 *1 491 *1 * 492 493 Called by: STARTGAM (procedure in SYS11.PRG) *! (procedure in SYS11.PRG) (procedure in SYS11.PRG) (FOXBASE+ function) (FOXBASE+ function) 494 *! Calls: RUNSCREN 495 *! : SAYTIME0 496 *İ : VALO 497 *! : SYSO *! *! (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) 498 : INTŎ 499 : LTRIMO * 500 : STR() : TIMECAL 501 *i (procedure in SYS11.PRG) (procedure in SYS11.PRG) 502 *1 : RUNPROD (procedure in SYS11.PRG) 503 *j : GOODSERV (procedure in SYS11.PRG) 504 *1 : SHIPPING *! 505 : RENSTAT (procedure in SYS11.PRG) *1 506 : DAYREPOR (procedure in SYS11.PRG) * 507 ****** 508 procedure runfact 509

211

the second of the second s

Statt water water second to be a survey of a

and a set for set

510	*****
511	clear
512	do runscren
513	store val(svs(2)) to factstart
515	do while .t.
516	newclock = int((val(sys(2))-factstart-difstop)*(160/60))
518	var = "LOTMS" + ltrim(str(mloop)) + ltrim(str(tempdd))
519	public &var
520	$\delta var = val(sys(2))$
522	var = "LOTMF" + ltrim(str(mloop)) + ltrim(str(tempdd))
523	public &var
524	dcvar=val(sys(2)) if tempdd1=tempdd
526	do saytime0
527	
528	var = LOPDS + trim(str(intoop)) + trim(str(ienpod))
530	&var = val(sys(2))
531	do runprod ver="I_OPDF"+!trim(str(mloon))+!trim(str(tempdd))
533	public &var
534	&var=val(sys(2))
535	$mioop \approx mioop + i$
537	set color to gr*+/bu
538	@ 10,57 say "8:00"
540	select 10
541	replace sys_stop with .t.
542 543	unlock ver="LOGDS"+ltrim(str(tempdd))
544	public &var
545	&var = val(sys(2))
540	var = "LOGDF" + ltrim(str(tempdd))
548	public &var
549	$\begin{aligned} &\& var = val(sys(2)) \\ &var = "I OSPS" + itrim(str(tamp dd)) \end{aligned}$
551	public &var
552	&var=val(sys(2))
553 554	do shipping var="LOSPF"+ltrim(str(tempdd))
555	public &var
556	&var=val(sys(2))
558	do renstat
559	set color to
560	clear
562	@ 8,24 to 12,54 double
563	set color to gr+/n
565	@ 10,2/ Say "GENERATING THE PAST FILES"
566	do dayrepor
567	do runscren
569	endif
570	endif
572	**
573 *!	***********************
574 *!	This procedure is called when the gaming-simulation
576 *1	(quit or continue with the game).
577 *!	Called by: SAYTIME0 (procedure in SYS11.PRG)
578 *!	Calle: CONTINUE (procedure in SVS11 PDG)
580 *!	: QUITGAME (procedure in SYS11.PRG)
581 *!	*****
582 *1	procedure factstop
584	******
585	select 10
587	unlock
588	**
589	** ** SEND THIS SIGNAL TO THE DIFFEDENT DEDADTMENTS IF IT IS T IN EVENY
591	** COMPUTER OF THE GAME A BLINKING MENU WILL APPEAR SAYING THAT THE GAME
592	** IS TEMPORALY FROZEN IN ORDER TO DISPLAY THE DAY'S RESULTS. ALSO ALL THE
593 504	** DATABASES WILL BE CLOSED.

595 596 597 ** LOCK THE DEPARTMENTS DATA INPUT IN ORDER NOT TO LOSE THE TIME/DATE ****** SEQUENCE. 598 ** 599 ** 600 set color to 601 clear 602 do while .t. 603 clear 604 m menu=0605 set color to g/n @ 3,16,17,62 box " ┌──┐ | = = └= | " 606 607 set color to gr/n @ 5,26 to 7,53 double set color to bu/n 608 609 610 @ 6,33 say "GAME FROZEN" 611 set color to br/n 612 @ 10,24 to 16,54 613 614 615 set color to bu/bg @ 20,5 say "THE GAME HAS BEEN FROZEN.SELECT FROM MENU." set color to bg/n @ 12,32 prompt "1.CONTINUE GAME." @ 13,32 prompt "2.QUIT GAME. " 616 617 618 menu to m menu 619 -do case 620 621 622 623 case m_menu=1 do continue restore screen from runscr =exit 624 625 case m menu=2 do quitgame 626 627 628 endcase enddo return 629 630 * * * * * This procedure allows to save the data of the just Procedure: SAVEDAY finished day to files, in order to be restored if 631 *1 632 *! 633 634 635 *1 wanted. *į Called by: REPOMENU (procedure in SYS11.PRG) *į 636 *1 Calls: RESTVAR (procedure in SYS11.PRG) 637 *1 638 *! Uses: TEMP.DBF *1 639 *!** 640 641 642 ****** procedure saveday 643 clear set color to gr+/bu @ 5,6 to 18,72 double 644 645 set color to g + /n@ 6,24 to 8,54 646 647 648 649 650 651 set color to r*+/n @ 7,31 say "SAVE DAY FOR RESTART." set color to bg+/n @ 11,21 say "THE DATE TO SAVE FROM IS:" 652 ?? (mmfactdate-1) 653 654 655 656 657 658 set color to ,n/w do restvar with (mmfactdate-1) close databases copy file system.dbf to temp.dbf select 10 use temp copy file system.dbf to &msys copy file plann.dbf to &mspla copy file market.dbf to &mmar copy file machine.dbf to &mmar 659 660 661 662 copy file stock.dbf to &mstk copy file stock.dbf to &mstk copy file setup.dbf to &mstu copy file setup.dbf to &mstu copy file salpast.dbf to &msta 663 664 665 666 667 668 copy file salperf.dbf to &msalf 669 670 671 672 copy file stkpast.dbf to &mstkp copy file machpast.dbf to &mmacp copy file plapast.dbf to &mplap set view to run 673 return 674 675 This procedure allows the game manager to restore Procedure: RESTART the data previously saved, in order to run the next day with that data. Called by: GAMCONF (procedure in SYS11.PRG) 676 677 * *| *| 678 +j 679

the gas is a to a sure is which is in the " with the add the said of the said of w when a deal a trainer of in the state of the second in the second

" Were the state of a failed and and and the state of the

680	*1	
681	*!	Calls: DTOC() (FOXBASE+ function)
682	*!	: KEPLSIS (procedure in SISII.PKG)
684	*1*	*******
685	•	procedure restart
686		******
687		clear
688		set color to gr + /bu
600		(a) 5,6 to 18,72 double
601		a = 524 to 8.54
692		set color to $r^{+} + /n$
693		@ 7.28 say "RESTART DAY FOR NEW RUN."
694		set color to bg+/n
695		@ 11,18 say "THE DATE TO RESTART FROM IS :"
696		?? dtoc(mmfactdate)
608		set color to ,n/w
699		replace restart with .t.
700		unlock
701		do replsys
702		close databases
703		erase plann.dbf
704		copy file compla to plann.doi
706		conv file &mmar to market.dbf
707		erase machine.dbf
708		copy file &mmac to machine.dbf
709		erase stock.dbf
710		copy file & mstk to stock.dbf
712		conv file & mhil to hill dhf
713		erase setup.dbf
714		copy file &mstu to setup.dbf
715		erase status.dbf
716		copy file consta to status.dbf
718		conv file & mesh to relove tdbf
719		erase salperf.dbf
720		copy file &msalf to salperf.dbf
721		erase stkpast.dbf
722		copy file &mstkp to stkpast.dbf
724		conv file & mmach to mach ast dbf
725		erase plapast.dbf
726		copy file & mplap to plapast.dbf
727		return
728	**	~~
730	*	This procedure calls the procedures that will create
731	*i	Procedure: DAYREPOR and store in the report datafiles, the report variables
732	*!	to be shown in the plannig department.
733	*!	Called by: RUNFACT (procedure in SYS11.PRG)
734	*1	Calles ITEMA (EOVERSE)
736	*1	: STRO (FOXBASE+ function)
737	*	: VAL() (FOXBASE+ function)
738	*ţ	: SYSO (FOXBASE+ function)
739	*!	: SALREPO (procedure in SYS11.PRG)
740	+! #!	: PLAKEPU (procedure in SYS11, PKG)
742	*1	· OR SYSTEMA (FOXBASE+ function)
743	+	: REPOMENU (procedure in SYS11.PRG)
744	*!	
745	*i*	*************************
740		procedure dayrepor
748		private scr
749		sor=.t.
750		var="LOSRS"+ltrim(str(tempdd))
751		public &var
752		acvar=vai(sys(2))
754		var="LOSRF" + ltrim(str(tempdd))
755		public &var
756		&var=val(sys(2))
757		var="LOPKS"+ltrim(str(tempdd))
138		public ocvar &var=val(eve(2))
760		do plarepo
761		var="LOPRF" + ltrim(str(tempdd))
762		public &var
763		&var=val(sys(2))
/04		select 10

....

765 766 767 768 replace salreport with .t. replace plareport with .t. unlock ? throughput, systock, tot_overh, assets -----do while system("salreport").or.system("plareport") 769 770 771 -if scr clear clear set color to gr+/bu @ 7,17 clear to 15,61 @ 7,17 to 15,61 double set color to r*+/bu @ 11,24 say "Please wait to finish the reports" @ 12,26 say "In the Departmental areas." 772 773 774 775 776 777 scr=.f. endif 778 779 780 enddo 781 782 do repomenu return 783 784 *!** ****** 785 *1 786 *1 787 *1 This procedure calls the procedure that generates and Procedure: SALREPO stores the report variables concerning the sales department. 788 789 *i Called by: DAYREPOR (procedure in SYS11.PRG) *i 790 791 792 793 794 Calls: SALPAST : ONTIME * (procedure in SYS11.PRG) *1 (procedure in SYS11.PRG) *1 : FREQUEN (procedure in SYS11.PRG) *! * ** 795 796 797 798 799 800 801 procedure salrepo . . set view to sales do salpast do ontime do frequen return 802 803 ****** ************ 804 *1 This procedure calls the procedures that generate and 805 +1 Procedure: PLAREPO store the report variables concerning the planning department. *į 806 807 *i Called by: DAYREPOR (procedure in SYS11.PRG) 808 *! 809 *į Calls: PLAPAST (procedure in SYS11.PRG) 810 +j 811 812 procedure plarepo 813 814 815 816 set view to plann do plapast return 817 818 * ***** ********** This procedure manages the possible problems that could Procedure: ERR_FIX arise from the fact that we are running in a multi-user 819 *! 820 821 822 *1 *1 environment. *j Cailed by: SYS11.PRG 823 *į 824 *Ì Calls: RLOCK() (FOXBASE+ function) 825 *1 826 * ********************* procedure err fix 827 828 829 830 831 parameters errnum, mess ** Error: File in use by another. if errnum=108 832 833 save screen to screen 834 set color to gr+/bu @ 7,17 clear to 15,61 @ 7,17 to 15,61 double 835 836 837 838 set color to r*+/bu @ 11,24 say "Please wait to append a record."
 @ 12,26 say "Press any key to continue." 839 840 read 841 842 restore screen from screen retry endif 843 844 845 846 847 ** ** Error: Record in use by another. if errnum=109 save screen to screen 848 time = 0849 set color to gr+/bu

and the second second second second

Carlo Carlo March

and the second s

12. martin and a list

and the second of the second of the second

215

that to the this date is an the feet of

@ 7,17 clear to 15,61 @ 7,17 to 15,61 double set color to r*+/bu @ 11,26 clear to 11,54 @ 11,26 say mess @ 12,26 say "Press any key to continue." 850 851 852 853 854 855 856 857 858 859 read do while .not. rlock().and.time < 1000 time = time + 1-enddo 860 -if time < 1000 861 restore screen from screen 862 retry 863 else set color to gr+/bu@ 7,17 clear to 15,61 @ 7,17 to 15,61 double set color to r^+/bu 864 865 866 867 @ 11,19 say "Record cannot be locked.Try again later." @ 12,26 say "Press any key to continue." 868 869 870 871 read restore screen from screen 872 873 -endif -endif 874 875 ** ** Error: Record is not locked. 876 if errnum=130 877 save screen to screen 878 879 time=0 do while .not. rlock().and.time < 1000 880 time = time + 1881 enddo 882 -if time < 1000 883 restore screen from screen 884 885 retry -else 886 set color to gr+/bu @ 7,17 clear to 15,61 @ 7,17 to 15,61 double set color to r*+/bu @ 11,19 say "Record cannot be locked.Try again later." @ 12,26 say "Press any key to continue." 887 888 889 890 891 892 893 read restore screen from screen 894 endif 895 -endif 896 * 897 *!**** *********** *1 898 This procedure calculates pseudo-random numbers. 899 *j Procedure: RND 900 *j 901 *j Called by: ENQGEN (procedure in SYS11.PRG) 902 *1 903 * Calls: MOD() (FOXBASE+ function) 904 *1 905 *!* ***** ***** 905 906 907 procedure rnd 908 store mod(seed *7137421+21132487,10000000)to seed 909 return seed/1000000 910 911 912 ************************* *! This procedure calculates the sales department's Procedure: GOODSERV quotation performance and its fail or success. Its 913 *! 914 *! 915 *! functioning is thorougly explained in one of the thesis's sections. VFACT (procedure in SYS11.PRG) Called by: RUNFACT 916 * ls: EOF() (FOXBASE+ function) : QUDELIVER > ENDEL(FOXBASE+ function) 917 *j Calls: EOF() 918 *! 919 * 920 921 922 923 procedure goodserv select 10 924 replace goodserv with .t. 925 unlock 926 927 928 select 2 goto top =do while .not. eof() ______if status='QUOT' 929 930 931 932 -if quprice < = e- > standprice p=100 else -if quprice >e->standprice.and.quprice <e->price_limt p=((quprice-e->price_limt)/(e->standprice-e->price_limt))*100 933 934

1 X 2 4 1 4 4 2 5 1

1. 1 M. 2. ...

1. 7 " "Ban

216

where it is a state to an extension of the state of the

and the stand of a second state of the state of a state of the state of the state of the second state of t

935 else 936 p=0endif 937 -endif 938 dis = 100939 940 -if qudeliver < = endeliver d=100 941 942 else , -if qudeliver>endeliver.and.qudeliver<(endeliver+e->deliv_limt) d ≂(((endeliver+e->deliv_limt)-qudeliver)/(e->deliv_limt))^{*}100 943 944 945 946 947 else d=0 endif 948 949 950 951 952 953 953 954 955 endif replace succeprice with p replace succedeliv with d unlock -if (rnd()*100) < =p*d*dis replace status with "FIRM" else replace status with "LOST" 955 956 957 958 959 960 endif unlock endif -if .not. eof() skip endif 961 962 enddo 963 select 10 964 965 966 967 replace goodserv with .f. unlock return 968 ***** This procedure creates the running screen displayed Procedure: RUNSCREN while the gaming-simulation is running. 969 *] 970 *1 970 *1 971 *1 972 *1 973 *1 974 *1 Called by: RUNFACT (procedure in SYS11.PRG) 975 procedure runscren 976 977 store 0 to check set color to $r^* + /bu$ (@ 6,31 say "RUNNING FACTORY" set color to gr + /bu(@ 3,7 to 21,71 double set color to r + /bu978 979 980 981 982 @ 10,13 say "DATE: @ 10,48 say "TIME: 983 984 985 986 987 988 @ 19,19 say "PRESS <END> OR Ctrl-F TO FREEZE THE GAME." return 989 ****** This procedure displays the time and date values Procedure: SAYTIME0 the first time the gaming-simulation starts running. 990 *! 990 991 992 993 993 994 995 996 *! *! *! Called by: RUNFACT (procedure in SYS11.PRG) Calls: INKEY() : VAL() : SYS() (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) *! *! 997 *! 998 *! : FACTSTOP (procedure in SYS11.PRG) 999 * 1000 *!* ***** ******* 1000 1001 1002 1003 1004 procedure saytime0 1005 1006 1007 1008 store val(sys(2)) to startstop 1009 1010 save screen to runscr do factstop difstop = difstop + (val(sys(2))-startstop) 1011 1012 endif 1013 return 1014 1015 *!** 1016 *1 1017 *1 This procedure displays the time and date values Procedure: SAYTIME1 every time these are recalculated. * 1018 *į 1019 Called by: TIMECAL (procedure in SYS11.PRG)

and a state of the state

Martin Star Date in

a start and a start and a start and the second start and a
Mar addition in the

217

Let have de state

The same of the spectrum weat of more than a the

12. 1. 1. 1. 2. 15 3

1020 *! 1021 *! +14 1022 procedure saytime1 1023 1024 set color to gr*+/bu @ 10,22 say mmfactdate picture "@E" @ 10,57 say mmfacttime 1025 1026 1027 1028 return 1029 This procedure calculates the gaming-simulation's time Procedure: TIMECAL and date values. Its functioning is thoroughly explained in one of the thesis's sections. 1030 *! 1031 *! 1032 *! 1033 *! 1034 *! 1035 *! Called by: RUNFACT (procedure in SYS11.PRG) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) Calls: MOD() : INT() : STR() 1036 *! 1037 *! : STR() (FOXBASE+ function) : CTOD() (FOXBASE+ function) : SAYTIME1 (procedure in SYS11.PRG) : RECNO() (FOXBASE+ function) : .AND..NOT.EOF()(FOXBASE+ function) : TRIM() (FOXBASE+ function) : SYSTEM (procedure in SYS11.PRG) : .NOT.TRANSFORM((FOXBASE+ function) : TRANSFORM() (FOXBASE+ function) : SPACE() (FOXBASE+ function) 1038 *! 1039 *! 1040 *! 1041 *! 1042 *! 1043 *! 1044 *! 1045 *! 1046 *! 1047 *1 1048 * ***************** 1049 procedure timecal 1050 1051 factclock=factclock1 1052 factclock1 = newclock 1053 1054 1055 difclock=factclock1-factclock factmin=mod(factclock1,60) facthhh = facthh facthh = int(factclock1/60)tempdd = factdd factdd = int(facthh/8) + factdd 1056 1057 1058 1059 tempdd1 = factdd 1060 facthh=mod(facthh,8) 1061 factmm=int(factdd/31)+factmm 1062 factdd=mod(factdd,31) 1063 1064 -if factdd=0 factdd = 11065 endif 1066 factyy=int(factmm/13)+factyy 1067 factmm=mod(factmm,13) 1068 -if factmm=0 1069 factmm=1 1070 1071 endif factyy=mod(factyy,100) if factyy=0 factyy=1 endif 1072 1073 1074 1075 if factmin < 10 1076 mmfacttime = str(facthh, 1) + ":0" + str(factmin, 1)1077 else 1078 mmfacttime=str(facthh,1)+":"+str(factmin,2) endif mfactdate = str(factdd,2) + "/" + str(factmm,2) + "/" + str(factyy,2) mmfactdate = ctod(mfactdate) 1080 1081 1082 if tempdd1=tempdd do saytime1 endif 1083 1084 1085 select 2 1086 1087 set order to 2 goto bottom 1088 oldrec = recno() 1089 mwaittime = ((mmfactdate-confdate)*480) + (facthh*60) + factmin 1090 1091 goto bottom botrec=recno() 1092 if (oldrec < > botrec.or.first).and..not.eof() 1093 -if botree = 1 1094 1095 tt=botrec first = .f.1096 else 1097 tt = oldrec + 11098 endif xx=trim(system("newenqrec")) do while tt < = botrec 1099 1100 -if .not.transform(tt, "999") \$ xx xx=transform(tt, "999") + xx 1101 1102 endif 1103 1104 tt = tt + 1

and a start of a

6.4

1. . .

218

all a compared on the case of the second of the second of the second second of the second second second second

- - - -

With half and the also

1105	enddo				
1106	else				
1107	xx=space(9)				
1109	set order to 1				
1110	select 10				
1112	replace factdate with mmfactdate				
1113	replace waittime with mwaittime				
1114	replace newengree with XX				
1116	return				
1117	**				
1119 *	This procedure calculates the gaming-simulation's production				
1120 *	Procedure: RUNPROD values for each of the existing resources. Its functioning is				
1121 *	Called by: RUNFACT (procedure in SYS11 PRG)				
1123 *					
1124 *	Calls: EOF() (FOXBASE+"function)				
1126 *	: MOD() (FOXBASE+ function)				
1127 *	: INT() (FOXBASE+ function)				
1128 *	STR() (FOXBASE+ function)				
1130 *	: ANDNOT.EOF()(FOXBASE+ function)				
1131 *					
1132 *	nrocedure runnrod				
1134	*****				
1135	select 3				
1130	do while .not, eof()				
1138	reso = resour_cod				
1139	select 1 set filter to resour cod=reso and ord stat $1 < >$ "FINISHED"				
1141	goto top				
1142	if .not.eof()				
1144	actsetup = $f > type$				
1145	do case				
1146	$= case c - > t_avail = 0$				
1148	timepro=(quantit-product 0)*e->procestime				
1149	else				
1150	endif				
1152	$=$ case c->t_avail <>0				
1153	if actsetup = c - > lastsetup				
1155	else				
1156	timepro=(quantit*e->procestime)+f->setup				
1157	endrase				
1159	$=$ do while newclock > = (c->t_avail+timepro)				
1160	if ord_stat_l="NO_START"				
1162	select 3				
1163	replace t avail with factclock				
1165	select 1				
1166	endif				
1167	replace start_date with j->factdate				
1169	mfactmin=mod(c->t avail.60)				
1170	mfacthh=int(c->t_avail/60)				
1172	= 11 mfactmin < 10				
1173	-else				
1174	mfacttime = str(mfacthh, 1) + ":" + str(mfactmin, 2)				
1176	replace start time with mfacttime				
1177	replace ord_stat_1 with "STARTED"				
1178	unlock seléct 10				
1180	replace statreso with a->resour cod				
1181	replace statcode with a->code				
1182	uniock select 3				
1184	replace avail with "NO "				
1185	unlock salest 1				
1187	endif				
1188	if tempdd1 > tempdd				
1189	replace finis_date with (j->factdate)-1				

". Believe to

Ward a very

and the state that and and and

The start and the start

うち たいおか 一日の はない とうまたい しょうせい アイ・デオ・デオ たいち しゅうけんがかたいよう

「人うちんう」、 あいので いろん

1190 else 1190 1191 1192 replace finis date with (j-> factdate) endif 1193 unlock mfactmin=mod((c->t_avail+timepro),60) mfacthh=int((c->t_avail+timepro)/60) _____if mfactmin < 10 1194 1195 1196 1197 mfacttime=str(mfacthh,1)+":0"+str(mfactmin,1) 1198 else 1199 mfacttime=str(mfacthh,1)+":"+str(mfactmin,2) 1200 endif 1201 1202 1203 select 4 replace stock with stock+(a->quantit) unlock 1203 1204 1205 select 3 replace avail with "YES" replace t_avail with (t_avail+timepro) 1206 1207 unlock 1208 1209 1210 select 1 replace finis_time with mfacttime replace product_1 with quantit replace ord_stat_1 with "FINISHED" 1211 1212 unlock 1213 -if .not. eof() skip endif 1214 1215 1216 1217 -if eof() select 3 1218 replace lastsetup with actsetup 1219 unlock 1220 select 1 1221 1222 1223 exit -endif L select 3 1224 replace lastsetup with actsetup 1225 1225 1226 unlock select 1 1227 actsetup=f->type 1228 1229 1230 if actsetup = c->lastsetup timepro=quantit*e->procestime else 1231 timepro=(quantit*e->procestime)+f->setup 1232 endif 1233 enddo 1234 1235 1236 1237 1238 replace t_avail with factclock 1239 unlock 1240 select 1 1241 1242 endif replace start_date with (j->factdate) 1243 1244 1245 unlock mfactmin=mod(c->t_avail,60) mfacthh=int(c->t_avail/60) 1246 1247 mfacttime=str(mfacthh,1)+":0"+str(mfactmin,1) 124/ 1248 1249 1250 1251 1252 1253 -else mfacttime=str(mfacthh,1)+":"+str(mfactmin,2) endif replace start_time with mfacttime replace ord_stat_1 with "STARTED" unlock 1254 select 10 1255 1256 replace statreso with a->resour_cod replace statcode with a->code 1257 unlock 1258 select 3 1259 1260 replace avail with "NO " unlock 1261 select 1 1262 endif 1263 -if actsetup = c->lastsetup -if tempdd1 > tempdd produce=int((480-(c->t_avail))/(e->procestime)) 1264 1265 1266 1267 else produce=int((newclock-(c->t_avail))/(e->procestime)) 1268 endif 1269 else 1270 if tempdd1 > tempdd produce=int((480-(c->t_avail)-(f->setup))/(e->procestime)) 1271 1272 else 1273 produce=int((newclock-(c->t_avail)-(f->setup))/(e->procestime)) 1274 endif

all the start of the last of the and the show the lot the set of the set of the set i well and in maker i freite NA SAN ALTER ALE ALTER A

· she

and the second second states with a state of the second state of the second states the second s

Strate and a second second with the second
1275		endif					
1276		select 1					
1277		replace product_1 with produce + product_0					
1278		unlock					
1279		endif					
1280		endif					
1281		set filter to					
1282		select 3					
1283		if .not. eof()					
1284		skip					
1285		endif					
1286		enddo					
1287		return					
1288		**					
1289	*	***************************************					
1290	*!	This procedure generates the enquires that will be shown					
1291	- 11	Procedure: ENQGEN to the sales department's people the next coming day. Its					
1292	<u>.</u> 1	functioning is thoroughly explained in one of the thesis's sections.					
1293	1	Called by: GAMCONF (procedure in STS11.PKG)					
1294	긢	Calles I FNO (FOVDASE America)					
1295	긞	Calls: LEN() (FOXBASE+ function)					
1290	1	VAL() (FOABASE+ Rection)					
1297	꿃	TRATE (FOXBASE + MUNCTION)					
1290		POUNDO (POABASET NUCLION)					
1299	1	: ROUND() (FOXBASE+ miction)					
1201	1	(procedure in STSTL.PRG)					
1202		SUBSIK() (FUXBASET IUnction)					
1202	*1	MMFACIDATES() (FOXBASET function)					
1204		· MIMFACIDATE () (FOADASET INICION)					
1204	- 14	DED (FOADASET INICION)					
1205	-	: INTO (FOADASET (Interior)					
1207	*	(FORDASET function)					
1209	*1	****					
1300		neoradura angran					
1310		**************************************					
1311		nrivete mindd meydd gendete					
1312							
1313		select 5					
1314		store val(right(time(1) 2)) to seed					
1315		store round((rnd()*(lengurec)-1))+1.0) to element					
1316		store val(substr(engurec.element.1)) to select					
1317		goto select					
1318							
1319		maxdd=maxdelivdd					
1320		mindd=mindelivdd					
1321		gendate = (confdate + (mindelivdd-1))					
1322		else					
1323		if mmfactdate > (confdate + (maxdelivdd-1))					
1324		replace tot enqu with num enqu					
1325		else					
1326		maxdd = (factdd + i - > enguperiod)					
1327		maxdd=iif(maxdd>maxdelivdd,maxdelivdd,maxdd)					
1328		mindd = factdd + 1					
1329		gendate = mmfactdate + 1					
1330		endif					
1331		endif					
1332		if num_enqu < tot_enqu					
1333		store val(right(time(1),2)) to seed					
1334		store round((rnd()*(slack-1))+plow,0) to number					
1335		number=number+((mmfactdate-confdate)*480)					
1336		plow=plow+slack					
1337		replace num_enqu with (num_enqu+1)					
1338		unlock					
1339		store code to mcode					
1340		store descript to mdescript					
1341		store standprice to mstprice					
1342		store val(right(time(1),2)) to seed					
1343		store round((rnd)*(maxdd-mindd)),0)+gendateto mendate					
1344		store int((rnd()*(maxquantit-minquantit)) + minquantit) to menquant					
1345		select 9					
1346		append blank					
1347		replace code with mcode					
1348		replace descript with mdescript					
1349		replace endate with mmfactdate					
1350		replace enquantit with menguant					
1351		replace endeliver with mendate					
1352		replace enprice with mstprice					
1353		replace qudeliver with mendate					
1354		replace quprice with mstprice					
1355		replace status with 'ENQU'					
1356		replace wait with number					
1357		unlock					
1358		else					
1359		enqurec=stuff(enqurec,element,1,"")					

The section of the se

Suca ...

1360 1 -endif 1361 enddo-1362 select 9 1363 use 1364 return 1365 1366 **** This procedure shipes the orders with delivery date equal or Procedure: SHIPPING less than the recently finished day. Its functioning is thoroughly explained in one of the thesis's sections. Called by: RUNFACT (procedure in SYS11.PRG) 1367 *! 1368 *1 1369 *1 1370 *1 1371 *1 1372 *1 ls: EOF() (FOXBASE+ function) : .NOT.FOUND() (FOXBASE+ function) : RECNO() (FOXBASE+ function) Calls: EOF() 1373 ÷₽ 1374 *! 1375 *! 1376 *!*********** ********* 1377 procedure shipping 1377 1378 1379 private quantdeliv, yeskip 1380 select 2 1381 goto top 1382 1383 do while .not. eof() yeskip = .t. if ((j - stattate) - 1) > = qudeliver if d - stock > = enquantit 1384 1385 1386 quantdeliv=enquantit recdeliv=qudeliver replace realdeliv with ((j->factdate)-1) 1387 1388 replace delivered with .t. 1389 1390 unlock 1391 select 4 1392 replace stock with (stock-quantdeliv) 1393 unlock 1394 select 2 1395 unlock 1396 1397 seek recdeliv -if .not.found() 1398 goto recno(0) -endif 1399 1400 yeskip=.f. 1401 endif 1402 endif 1403 1404 1405 -if .not. eof().and.yeskip skip endif 1406 enddo 1407 return 1408 1409 *!*** ****** 1410 *1 1411 *1 1412 *1 1413 *1 This procedure allows the game manager to make a choice among three Procedure: REPOMENU menu choices after a day has finished (save the day's data, run the next day or quit the game). Called by: DAYREPOR (procedure in SYS11.PRG) 1414 *! 1415 *! Calls: SAVEDAY (procedure in SYS11.PRG) 1416 *! : NEXTDAY (procedure in SYS11.PRG) 1417 *! 1418 *! 1419 *! : EXITGAME (procedure in SYS11.PRG) Memory Files: FIL.MEM 1420 *! 1421 *!*** 1422 procedure repomenu 1423 e sele sele sele sele sele sele ** 1424 ** 1425 1426 1427 1428 ** MENU OF THE DAY REPORT, ALLOWING TO QUIT, SAVE DAY'S RESULTS OR CONTINUE. ** ** 1429 set view to run 1430 set color to 1431 do while .t. 1432 clear 1433 m menu=0 1434 1435 1436 set color to g/n @ 3,16,17,62 box " ┌───┐ | ज = ⊨ | " set color to gr/n @ 5,26 to 7,53 double 1437 1438 set color to bu/n 1439 @ 6,29 say "DAY RUN FINISHED" 1440 1441 1442 set color to br/n @ 10,24 to 16,54 set color to bg/n @ 12,30 prompt "1.SAVE DAY'S DATA." @ 13,30 prompt "2.RUN NEXT DAY. " 1443 1444

N. O'S P. S. P. S.

And a week

1. 8. 18 1 AV

Nº water an mar

Street the star of

222

wedness. Berginse

A. Mathier .

a black of the state of the state of the second structure of the

1445	1	@ 14,30 prompt "3.QUIT GAME. "
1446		save to fil all like lo*
1447		menu to m_menu
1448		do case
1449	1	do revedev
1451		=case m menu=2
1452		do nextday
1453	v	exit
1454		case m_menu=3
1455		do exitgame
1456		endcase
1457	1	enddo
1450		**
1460	*!*	************
1461	*1	This procedure calculates the frequency of generation
1462	*1	Procedure: FREQUEN of enquires. Its functioning is thoroughly explained in one
1463	*!	of the thesis's sections.
1404	*!	Called by: SALKEPO (procedure in STS11.PKG)
1465	*	Calls: EOEO (EOXBASE + function)
1467	+į	: MIN() (FOXBASE+ function)
1468	*į	: INT() (FOXBASE+ function)
1469	*!	
1470	*!*	,
1471		procedure frequen
1472		calant 1
1474		solot 1
1475		nday = (mmfactdate-(i-> startdate))
1476		do while .not. eof()
1477		if d->ave_qprice<=mrkt_price
1478		pe=1
14/9		if d- >ave annice > mrkt price and d->ave annice < mrkt limt
1481		pe=((mrkt limt)-(d->ave aprice))/((mrkt limt)-(mrkt price))
1482		else
1483		pe=0
1484		endif
1485		endif
1480		d_{a-1} for time delv > = 100
1488		
1489		r-if ontimedely > ontimlimt and ontimedely < 100
1490		de = ((ontimedely) - (ontimlimt))/(100 - ontimlimt)
1491		-else
1492		de=0
1493		endif
1494		f = 1 + (((6*nday - 1))((5*nday))*(d > ave enqu)*min(ne de))
1496		replace tot enqu with int(f)
1497		unlock
1498		if .not. eof()
1499		skip
1500		
1502		sum fot engli to sum engli
1503		select 10
1504		replace frequency with int(480/sum_enqu)
1505		unlock
1506		select 1
1507		return
1509	*	***********
1510	*!	This procedure calculates and stores the report variables
1511	*!	Procedure: SALPAST concerning the sales department's daily performance.
1512	*1	
1513	*! *!	Called by: SALREPO (procedure in SYS11.PRG)
1514	*1	Calls: EOE() (FOXBASE+ function)
1516	*1	; IIF() (FOXBASE+ function)
1517	*1	: SALPERF (procedure in SYS11.PRG)
1518	*!	
1519	*!	************
1520		procedure salpasi
1521		select 1
1523		goto top
1524		do while .not. eof()
1525		store 0 to numeng, numquo, numfirm, mavqprice, mqprice, mqdeliv, numontime,;
1526		numdeliv, timdeliv, mfprice, mfdeliv, mvol, mthrough, dvol, tobedeliv
1527		store code to mcode
1520		store descript to modescript
1529		

. . .

「日本の日本にはない



ふうち ひろうちょう

the support . The second

19 X 44

1615 *! 1616 1617 1618 *11 procedure salperf 1619 select 4 1620 goto top 1621 do while .not. eof() 1622 store code to mcode 1623 1624 1625 store (mmfactdate-1) to mdate average for code=mcode to mave enqu,mave_quot,mper_quot,mave_firm,; mper_firm,movr_deltim,mper_ontime,mave_qprice,mper_qprice,; mper_qdeliv,mper_fprice,mper_fdeliv,mdeliveries,mtodeliv,mvolume select 4 1626 1627 1628 1629 1630 replace date with mdate replace ave_enqu with mave_enqu replace ave_quot with mave_quot 1631 replace per quot with mper quot replace ave firm with mave firm 1632 1633 1634 replace per_firm with mper_firm 1635 replace ovr_deltim with movr_deltim replace ovr_denim with movr_denim replace per_ontime with mper_ontime replace ave_oprice with mave_oprice replace per_oprice with mper_oprice replace per_oprice with mper_oprice replace per_force with mper_force replace per_follow with mper_follow replace deliveries with motioveries melace todaliw, with motioveries 1636 1637 1638 1639 1640 1641 1642 1643 1644 with mtodeliv with mvolume replace todeliv replace volume 1645 unlock 1646 -if .not. eof() 1640 1647 1648 1649 1650 1651 skip endif enddo return 1652 This procedre calculates and stores the report variables Procedure: PLAPAST concerning the planning department's daily 1653 *1 1655 *1 1655 *1 1656 *1 1657 *1 and overall performance. Called by: PLAREPO (procedure in SYS11.PRG) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) 1658 *! Calls: EOF() : .NOT.EOF() : VAL() 1659 *! 1660 *! 1661 *! : STRO (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) 1662 *! : LEFT() 1663 *! 1664 *! 1665 *! * : RIGHT() : CTODO 1666 * 1667 procedure plapast 1668 1669 1670 1671 1672 select 2 store 0 to totprod store 0 to mmwip goto top 1673 1674 1675 do while .not. eof() store 0 to product, util, nset, tset store " " to lset, sett 1676 store resour cod to cod 1677 store capacity to cap store redescript to descrp 1678 1679 1680 1681 select 1 set filter to resour_cod=cod.and.ord_stat_0<>"FINISHED".and.ord_stat_1<>"NO_START" goto top 1682 do while .not.eof() 1683 sett=d->set up 1684 product=product+((product 1-product 0)*d->procestime)&& production in minutes if lset < > sett 1685 1686 1687 tset=tset+h->setup -endif 1688 1689 lset=sett -if product_1 < quantit product=480-tset 1690 1691 1692 endif 1693 -if .not. eof() 1694 skip 1695 1696 endif enddo 1697 totprod = totprod + product store (product/cap) to util 1698 set filter to 1699

7.1. . 344

and the second of the second states of the

1700 1701 1702 select 5 append blank replace resour cod with cod replace descript with descrp replace capacity with cap 1703 1703 1704 1705 1706 1707 1708 1709 1710 replace production with product replace productiv with util*100 && Utilization in percentage. replace setuptime with tset replace setuptum with nset replace date with (j->factdate)-1 unlock 1711 1712 select 2 if .not. eof() skip endif 1713 1714 1715 1715 1716 1717 enddo 1718 1719 select 1 set relation to 1720 1721 1722 1723 1724 1725 1726 select 4 goto top do while .not. eof() store code to cod store descript to descrp 1727 1728 1729 store procestime to mprocstim store matcost to minatcost store 0 to incomp.start,batch,prod,last,pprod store 0 to cost,finish,mwip,lasting,avelas 1729 1730 1731 1732 1733 select 1 set filter to code=cod goto top content of conten 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 endif -if ord_stat_0="NO_START".and.ord_stat_1="STARTED" start=start+1 mwip=mwip+product_1 pprod=pprod+((product_1-product_0)*d->procestime) endif -endif -if finis_date = (j-> factdate)-1 store val(str(start_date-finis_date,3)) to difdays store val(left(start_time,1)) to starthour store val(right(start_time,2)) to startmin store val(right(finis_time,2)) to finishour store val(right(finis_time,2)) to finismin store (finishour-starthour) to difhour store (finishour-starthour) to difmin last = (difdays #840) + (difhour*f60) + difmin 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 last=(difdays*8*60)+(difhour*60)+difmin lasting = lasting + last finish = finish + 11756 1756 1757 1758 1759 prod = prod + quantit pprod = pprod + ((product_1-product_0)*d-> procestime) endif -if .not. eof() 1760 skip endif 1761 enddo 1762 1763 -if pprod=0 1763 1764 1765 1766 1767 1768 cost=d->stand_cost else cost=((mprocstim)*(j->hour_rate))+(mmatcost)+(((j->tot_overh)*pprod)/totprod)&& overhead per day endif set filter to avelas=iif(finish=0,0,(lasting/finish)) 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 mmwip=mmwip+(mwip*mmatcost) mwip=iif((incomp+start)=0,0,(mwip/(incomp+start))) batch=iif((inish=0,0,(prod/finish)) select 6 append blank replace code with cod replace descript with descrp replace incomplete with incomp replace started with start replace finished with finish 1779 1780 1781 1782 replace production with prod replace wip with mwip replace ave_batch with batch 1783 replace setupnum with (incomp + start + finish)

12.

1 to other

ころうちをないていていていていたいですのできたいです。 ちょうちょう ちょうちょう ちょうちょう

Sec. Sec.

the beh .

1784 replace leadtime with avelas

226

网络小麦 化二氟乙酸 化二乙酰氨基乙基乙酰基 化结节子 机装饰的 化正式放发机 网络新山市 化分离子 医静脉管 化热热管 化热

1785 1786 1787 1788 replace stand_cost with cost replace date with (j->factdate)-1 unlock select 4 1789 replace stand cost with cost 1790 unlock 1791 -if .not. eof() 1792 skip 1793 1794 1795 endif enddo 1795 1796 1797 select 3 1798 goto top do while .not. eof() 1799 1800 1801 stkcode=code stkdes=descript 1802 stk=stock mmwip=mmwip+(stk*d->matcost) select 7 1803 1804 1805 append blank 1806 replace code with stkcode 1807 1808 1809 replace descript with stkdes replace stock with stk replace date with (j->factdate)-1 for date=ctod(" / / ") 1810 unlock 1811 select 3 1812 if .not.eof() 1813 skip 1814 1815 1816 endif enddo select 10 1817 replace systock with mmwip 1818 unlock 1819 select 3 1820 return 1821 1822 *!* 1823 *! This procedure displays the resources' information, allowing Procedure: MACHBROW also its edition. It's based on the "browse" algorithm which is thoroughly explained in one of the thesis's sections. Called by: FACTCONF (procedure in SYS11.PRG) 1824 *! 1825 *! 1825 *! 1826 *! 1827 *! 1828 *! 1829 *! (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) (procedure in SYS11.PRG) (procedure in SYS11.PRG) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) L (procedure in SYS11.PRG) (procedure in SYS11.PRG) (procedure in SYS11.PRG) Calls: CHR() : REPLICATE() 1830 *! : RECNOO : MACH_DISP : GETKEY 1831 *! 1832 *! 1833 *! : BOF() 1835 *! 1834 *! 1835 *! 1836 *! 1837 *! : EOF() : MACH RSTR : MACHGETS : UPDATED() : MACH_REPL : MACH_STOR 1838 *! 1839 *! 1840 *1 : NEWMACH (procedure in SYS11.PRG) NOT.DELETED() (FOXBASE+ function) 1841 *! 1842 *! 1843 *! 1844 *! : RESETUP (procedure in SYS11.PRG) 1845 procedure machbrow 1846 1847 select 3 1848 1849 1850 1851 1852 1853 private recnumtop, recnumlast, skiprecs, home, endkey, uparrow, downarrow private row, rowtop, rowbottom, rowprompt, keystrokes, pagepaint, recnum private promptrow, promptbar, keyst2, choice * ---Initialize constants. choice = " " uparrow = chr(5)downarrow = chr(24)1854 1855 1855 1856 1857 1858 1859 1860 returnkey = chr(13)pgdn = chr(3)pgan = cnr(3)
pgup = chr(18)
delrecord = chr(7)
keystrokes = "EN" + uparrow + downarrow +;
pgdn + pgup + delrecord + returnkey
keyst2 = "R" + delrecord + returnkey
rowtop = 1
southed to m = 20 1861 1862 1863 1864 1865 1866 1867 rowbottom = 20rowprompt = rowbottom + 3 promptrow=22 promptbar=replicate(chr(196),80) 1868 skiprecs = rowbottom - rowtop + 1 1869 goto top

The second second the second second

i Vinder " Line

and the second s

ちんないのち いいいちのない あき

1870	*Initialize local variables
1871	row = rowton
1872	recourt = recool
1873	recountion = recours
1874	pagepaint = .t.
1875	isedited $= .f.$
1876	*Perform BROWSE.
1877	set color to & screenatr
1878	clear
1879	* The following loop is really a "REPEAT/UNTIL < cond >".
1880	do while .t.
1881	jif pagepaint
1882	recnum = recno()
1883	goto recnumtop
1004	do mach_disp with (rowtop), skiprecs
1003	* . Reposition record pointer when repainting current nage
1897	row = rowtop
1888	nagenaint = f
1889	endif
1890	set color to &promptatr
1891	@ rowprompt-1,0 say promptbar
1892	
1893	@ rowprompt,0 say ;
1894	"BROWSE: (E)xit <arrows> (N)ew <return></return></arrows>
1895	@ row,4 say chr(16)
1896	do getkey with choice, keystrokes
189/	Reposition record pointer.
1898	do while choice > uparrow + downarrow
1000	<i>(@ row,4 say if a balance in uncompany)</i>
1001	skin_1
1002	SKIP -1
1903	and case
1904	goto top
1905	case row > rowtop
1906	row = row - 1
1907	
1908	skip
1909	endcase
1910	else
1012	
1913	
1914	goto bottom
1915	case row < rowbottom
1916	row = row + 1
1917	motherwise
1918	skip -1
1919	endcase and if
1920	a row 4 rev chr(16)
1921	do getkey with choice keystrokes
1923	enddo
1924	*Prompt line selections.
1925	do case
1926	case choice = "E"
1927	vexit
1928	==case choice = returnkey
1929	do mach_rstr
1930	do machgets with .f.
1931	do mach cont
1032	uo mach_icpi
1034	do mach disp with row 1
1935	case choice = "N"
1936	@ row.4 say " "
1937	do while $r > 20$
1938	goto recnumtop
1939	skip skiprecs
1940	if eof()
1941	goto bottom
1942	endif
1943	recnum op = recno()
1944	recrum = recruo
1945	do much disp with (rowston) skinsters
1947	enddo
1948	goto recnum
1949	row = rowtop
1950	set color to &promptatr
1951	@ row,4 say chr(16)
1952	do mach_stor
1953	do newmach
1954	goto recnumtop

1. 1. 24 . 5 . 5 . W. W.

and the state of the second se



Then welling

and the states

ちいのようないないないないないである

and the well of the second of the

A service and the service of the ser

1. C. Raine Lane

229

in all the stand with the

Set Bach

difficia al Chi

2040 do while .t. 2041 do machedit 2042 -if choice < > "R" 2042 2043 2044 2045 2045 2046 2047 2048 =exit -endif set color to &promptatr @ rowprompt-1,0 say promptbar @ rowprompt.0 say ; "BROWSE: (E)xit <Arrows> (N)ew <Return> " 2049 enddo 2050 return 2051 2052 2053 This procedure allows the game manager the edition of Procedure: MACHEDIT the input fields of a newly created resource. *1 2054 *! 2055 *! 2056 *! 2057 *! Called by: NEWMACH (procedure in SYS11.PRG) 2058 *! 2059 *! (procedure in SYS11.PRG) (procedure in SYS11.PRG) (procedure in SYS11.PRG) Calls: MACHGETS : GETKEY : INCLUDE * 2060 2061 *1 2062 procedure machedit 2063 2064 2065 do machgets with .t. 2066 2067 2068 set color to &promptatr @ rowprompt,0 clear 2069 @ rowprompt,0 say ; "NEW MACHINE: without saving (R)ewrite <Return> saving " 2070 do getkey with choice, keyst2 2071 2072 2073 do include 2074 2075 -endif @ rowprompt,0 clear 2076 return 2077 2078 *!************** ***** This procedure gets the values input by the game manager Procedure: MACHGETS when editing the resource's input fields. 2079 *1 2080 *! 2081 *! Called by: MACHBROW : MACHEDIT (procedure in SYS11.PRG) 2082 *! 2083 *! (procedure in SYS11.PRG) 2084 * 2085 * ********* procedure machgets 2086 2087 2088 parameter newget 2089 set color to n/w 2090 -if newget if newget @ r,c get mresourcod picture "@!NNNN" @ r,c+9 get mdescript picture "@!" @ r,c+27 get mcapacity picture "9999" @ r,c+33 get msetup1 picture "999.99" @ r,c+40 get msetup2 picture "999.99" @ r,c+47 get msetup3 picture "999.99" @ r,c+54 get msetup4 picture "999.99" @ r,c+61 get msetup5 picture "999.99" 2091 2092 2093 2094 2095 2096 2097 2098 2099 else else @ row,c say mresourcod @ row,c+9 say mdescript @ row,c+27 get mcapacity picture "9999" @ row,c+33 get msetup1 picture "999.99" @ row,c+40 get msetup2 picture "999.99" @ row,c+47 get msetup3 picture "999.99" @ row,c+54 get msetup4 picture "999.99" @ row,c+61 get msetup5 picture "999.99" 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 endif read 2110 return 2111 +!++++ 2112 2113 ****************** This procedure includes a new record in the resource Procedure: INCLUDE datafile. 2114 *! 2115 *! 2116 * Called by: MACHEDIT (procedure in SYS11.PRG) 2117 *! 2118 *! 2119 *! Calls: MACH_REPL (procedure in SYS11.PRG) 2120 ***** procedure include 2121 2122 2123 append blank 2124 do mach_repl

and the second
114 . Back

230

Sec. in

with respect to reachest the section of the little

The second the second of the second frances

こう ちょうちょうち あるかい ふしいちょう

Catt in the Weak

return 2125 2126 2127 *** 2128 *! This procedure replaces in the newly included record the 2129 *! Procedure: MACH_REPL values input in the resource input procedure. 2130 +1 2131 *! 2132 *! Called by: MACHBROW : INCLUDE (procedure in SYS11.PRG) (procedure in SYS11.PRG) 2133 *İ 2134 +j Calls: EOF() (FOXBASE+ function) 2135 *1 ** 2136 **** procedure mach repl 2137 2138 2139 -if .not. eof() 2140 ---Replace only if there is an available record 2141 2142 replace; resour_cod with mresourcod,; 2143 2144 2145 redescript with mdescript,; capacity with mcapacity,; lastsetup with lstsetup,; with msetup1;; with msetup2;; 2146 2147 set_up1 set_up2 2148 set up3 with msetup3,; 2149 set_up4 with msetup4,; 2150 set_up5 with msetup5 2151 2152 2153 unlock endif return 2154 2155 *1 2156 *! 2157 *! 2157 *! 2158 *! 2159 *! 2160 *! This procedure stores the values of a resource record Procedure: MACH_RSTR to be reedited. Called by: MACHBROW (procedure in SYS11.PRG) +1 2161 ******* 2162 procedure mach rstr 2163 2164 store resour_cod to mresourcod 2165 store redescript to mdescript 2166 2167 2168 store capacity to mcapacity store lastsetup to istsetup store set_up1 store set_up2 to msetup 1 2169 2170 to msetup2 store set_up3 to msetup3 2171 store set_up4 to msetup4 2172 2173 2174 store set_up5 to msetup5 return 2175 + ***** This procedure stores the default values of a resource Procedure: MACH_STOR record to be edited. 2176 *İ 2177 *! 2178 *! *i 2179 2180 Called by: MACHBROW (procedure in SYS11.PRG) *j 2181 *i Calls: SPACE() (FOXBASE+ function) 2182 *1 * 2183 2184 procedure mach_stor 2185 store space(4) to mresourcod store space(15) to mdescript store 0 to mcapacity store "NO_SET " to lstsetup 2186 2187 2188 2189 2190 2191 store 0 to msetup1 store 0 to msetup2 2192 store 0 to msetup3 2193 store 0 to msetup4 2194 store 0 to msetup5 2195 return 2196 2197 2198 *!******* ******* *1 This procedure calls the procedures that will create the 2199 *į Procedure: OPEN different necessary "views" to run the gaming-simulation. 2200 *! 2201 * Called by: SYS11.PRG 2202 *! 2203 2204 2205 *1 **Calls: OPENRUN** (procedure in SYS11.PRG) * : OPENSAL (procedure in SYS11.PRG) *į : OPENPLA (procedure in SYS11.PRG) 2206 *1 *! 2207 procedure open 2208 2209

See State Leave See Now

Starting the start of the start of the

and the bar have been

2210	do openri	in the second second second second second second second second second second second second second second second
2211	do opens	al I-
2212	do openp	18
2214	**	
2215	* *******	*********
2216	*!	This procedure reindexes the used files in case
2217	*! Proce	dure: REINDEX there has been any indexing error.
2210	*! Caller	the GAMCONE (procedure in SYS11 PRG)
2220	*!	by, Gradoona procession of Strinkey
2221	*! Ca	ills: DELETC (procedure in SYS11.PRG)
2222	*	
2223	*! U	Ses: SYSTEM.DBF
2225	+1	: MARKET.DBF
2226	*i	: MACHINE.DBF
2227	*!	: STOCK.DBF
2228	*!	: BILL.DBF
2229	* 1	SALPAST.DBF
2231	*1	· MACHPAST DRF
2232	*i	: PLAPAST.DBF
2233	+1	: STKPAST.DBF
2234	*!	: SETUP.DBF
2235	*! Inde	ANN INY
2237	*!	: NAPLANN.IDX
2238	*!	: DEMARK.IDX
2239	*!	: WAIT.IDX
2240	平! 由1	
2241	*1	: CODESTK.IDX
2243	*į	: BILLNA.IDX
2244	*1	: CDSAL.IDX
2245	*!	: CPRFSAL IDX
2240	*! #1	CDMACH.IDX
2247	*1	· CDSTK IDX
2249	*i	: SETUP.IDX
2250	*1	
2251	* * ******	· · ·
2252	procedur	e reindex
2254	set exclu	sive off
2255	select 10	
2256	use syste	m
2257	replace r	eindex with .t. && inform the users that the databases
2250	uniock	dide can not be used by them.
2260	use plann	index orplann.naplann
2261	pack	· · · · · · · · · · · · · · · · · · ·
2262	use mark	et index demark, wait, saldeliv
2263	reindex	
2265	nack	
2266	use mach	ine index resour
2267	reindex	
2268	use stock	index codestk
2269	reindex	ndar hillen
2271	use on n reindex	NGA UIIIIA
2272	use salpa	st index cdsal
2273	reindex	
2274	use salpe	rf index cprfsal
22/5	reindex	nast index admach
2277	reindex	past muex comach
2278	use plapa	st index cdpla
2279	reindex	
2280	use stkpa	st index cdstk
2281	reindex	index estus
2283	reindex	much soup
2284	set exclu	sive off
2285	select 10	
2286	use syste	M
2288	volock	while a set of the set
2289	return	cele can be used by moni.
2290	**	
2291	* *******	*****************
2292	*] Dec.co	This procedure creates the "view" to be used when
2294	*1	oute. Or DIAKOIA Tunining the gaming-simulation,

Said Said and a la the la the

2295	*!	Called by: OPEN (procedure in SYS11.PRG)
2296	*!	LIGHT DI ANN DE
2298	*	: MARKET.DBF
2299	*1	: MACHINE.DBF
2300	7] *	STOCK.DBF
2302	*!	: SETUP.DBF
2303	*	: SYSTEM.DBF
2304	7] #[Indexes: ORPLANN.IDX
2306	*İ	: NAPLANN.IDX
2307	*	: DEMARK.IDX
2309	*1	: SALDELIV.IDX
2310	*	: RESOUR.IDX
2311	*!	: CODESTK.IDX
2313	*	: SETUP.IDX
2314	*1	
2315	*!	**************************************
2317		*****
2318		
2319		** DATAFILES USED TO KUN THE FACTORY. ** **
2321		set exclusive off
2322		select 1
2323		select 2.
2325		use market index demark, wait, saldeliv
2326		set filter to wait < = mwaittime
2327		select 5 use machine index resour
2329		select 4
2330		use stock index codestk
2332		use bill index billina
2333		select 6
2334		use setup index setup
2336		use system
2337		
2338		** RELATIONS USED WHILE RUNNING THE FACTORY SIMULATION/GAME **
2340		select 2
2341		set relation to code into d
2342		select 1
2344		set relation to code into d
2345		set relation to code into a additive
2347		select 5
2348		set relation to code into f
2349		set relation to resour_cod into c additive
2351		create view run from environment all
2352		set safety on
2353		ciose databases return
2355		**
2356	*!	****************
2358	*!	Procedure: OPENSAL calculating the sales department's past performance
2359	*	variables.
2360	*!	Called by: OPEN (procedure in SYS11.PRG)
2362	*	Uses: BILL.DBF
2363	뼴	: MARKET.DBF
2364	*! *!	SALPAST.DBF
2366	*	: SYSTEM.DBF
2367	*!	
2360	*! *!	Indexes: BILLNA.IDX : SALDELIV.IDX
2370	*	: DEMARK.IDX
2371	*!	: WAIT.IDX
2373	*1	: CDSAL.IDX : CPRFSAL.IDX
2374	+!	
2375	*!	***************************************
2377		aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
2378		**
2379		** DATAFILES USED TO CALCULATE THE SALES PERFORMANCE. **

and the set of a set

a station a

2380		**
2381		set exclusive off
2301		
2382		
2383		use bill index bilina
2384		select 2
2385		use market index saldeliv, demark, wait
2386		set filter to .not.delivered.or.realdeliv=conffdate
2387		select 3
2388		vse salnast index cdsal
2200		
2309		select 4
2390		use salpert index oprisal
2391		select 10
2392		use system
2393		**
2394		** RELATIONS USED WHILE CALCULATING THE SALES DEPT. PERFORMANCE.
2395		**
2306		select 1
2207		set relation to gode into d
2371		
2390		Belect 4
2399		set relation to code into c
2400		set safety off
2401		create view sales from environment all
2402		set safety on
2403		close databases
2404		rejuta
2405		
2406	*!*	*****
2400		
2407	-T1	Inis procedure creates the "view" to be used when
2408	*!	Procedure: OPENPLA calculating the planning department's past performance
2409	*!	variables.
2410	*!	Called by: OPEN (procedure in SYS11.PRG)
2411	*1	
2412	*	Uses PI ANN DRE
2412	*	· MACHINE DEF
2413	*	
2414	1	SIOCA.DBF
2415	- 71	: BILL.DBF
2416	*1	: MACHPAST.DBF
2417	*!	: PLAPAST.DBF
2418	*1	: STKPAST.DBF
2419	*1	· SETUP DBE
2420	*1	· SVSTEM DBE
2420	-	, 5151EM.DDF
2421		LAND ODDI ANN IDV
2422	- 71	Indexes: OKPLANN.IDX
2423	*!	: RESOUR.IDX
2424	*!	: CODESTK.IDX
2425	*!	: BILLNA.IDX
2426	*1	: CDMACH.IDX
2427	*1	
2429		
2420	1	CDSTRADA
2429		: SETUP.IDX
2430	-#1	
2431	*!	****************
2432		procedure openpla
2433		*****
2434		**
2435		** DATAFILES USED TO CALCULATE THE PLANNING PERFORMANCE. **
2436		
2430		not evolution off
2437		
2438		select 1
2439		use plann index orplann
2440		select 2
2441		use machine index resour
2442		select 3
2443		use stock index codestk
2444		select 4
2445		use hill index hillon
2446		
2440		Soleti J
2447		use macnpast index comacn
2448		select o
2449		use plapast index cdpla
2450		select 7
2451		use stkpast index cdstk
2452		select 8
2453		use setup index setup
2454		select 10
2455		
2433		use system
2430		
245/		*** RELATIONS USED WHILE CALCULATING THE PLANNING DEPT, PERFORMANCE.
2458		
2459		select 1
2460		set relation to code into d
2461		set relation to code into h additive
2462		select 3
2463		set relation to code into d
		set safety off
2464		

.

2465 create view plann from environment all 2466 set safet on 2467 close databases 2468 return 2470 * 1 2471 *! This procedure obtains the most updated 2472 *! Procedure: SYSTEM the Gold to a This procedure obtains the most updated value of Procedure: SYSTEM the fields in the system datafile. 2473 *1 2474 *! Called by: DAYREPOR R (procedure in SYS11.PRG) (procedure in SYS11.PRG) (procedure in SYS11.PRG) 2475 *! 2476 *! 2477 *! 2478 *! 2479 *! : TIMECAL : RENSTAT (FOXBASE+ function) (FOXBASE+ function) Calls: STR() : SELECT() 2480 *! 2482 procedure system 2483 2484 2485 2486 parameters fielddd base = str(select(), 2)set exclusive off 2487 select 10 goto top system=&fielddd select &base 2488 2489 2490 2491 2492 return system 2493 *!************ ****** 2494 *! This procedure gets the input of a key stroke Procedure: GETKEY on the keyboard. 2495 *! 2495 *! 2496 *! 2497 *! 2498 *! 2499 *! Called by: MACHBROW (procedure in SYS11.PRG) (procedure in SYS11.PRG) : MACHEDIT 2500 *! Calls: INKEY() (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) : UPPER() 2501 *! 2502 *! : CHR() 2503 * 2505 2506 2507 procedure getkey parameter choice, keychars 2508 private keycode choice = "*" 2509 2509 2510 2511 2512 2513 2514 2515 do while .not. (choice \$ keychars) keycode = inkey() if keycode > 0 choice = upper(chr(keycode)) endif enddo 2516 return 2517 2520 *! 2521 *! 2522 *! Procedure: EXITSYS it has started. Called by: MAINMENU (procedure in SYS11.PRG) 2523 * 2524 *! Calls: SET (procedure in SYS11.PRG) 2525 *1 2526 2527 procedure exitsys 2528 2529 2530 do set set color to 2531 clear 2532 close all 2533 clear all 2534 return 2535 ** 2536 ** 2537 *!*** ** ***** 2538 *! This procedure obtains the value of the setup time Procedure: FIELDD assign to a certain setup field. 2539 *! 2540 *j 2541 *! Called by: RESETUP (procedure in SYS11.PRG) 2542 *! 2543 *! 2544 *! 2545 *! Calls: FCOUNT() (FOXBASE+ function) (FOXBASE+ function) : FIELD() 2546 ***** 2547 2548 procedure fieldd 2549 parameter fieldd

2550 2551 2552 n = fcount(3)nn = 1fiel=field(nn,3) 2553 do while nn < n.and.fiel < > fieldd 2555 2554 2555 2556 2557 nn = nn + 1fiel=field(nn,3) enddo return c->&fiel 2558 2559 ********* ************ 2560 *! 2561 *! This procedure updates the values of the setup datafile Procedure: RESETUP in case there has been an update of the resource datafile. 2562 *1 2563 *1 2564 *1 2565 *1 Called by: MACHBROW (procedure in SYS11.PRG) Calls: EOF() (FOXBASE+ function) 2566 *! : FIELDD (procedure in SYS11.PRG) 2567 *1 2568 *!* 2569 ****** ***** procedure resetup 2570 2571 select 5 goto top do while .not. cof() 2572 2573 2573 2574 2575 2576 2577 2578 2578 2579 select 6 replace type with e->set_up replace setup with fieldd(e->set_up) unlock select 5 -if .not. eof() 2580 skip 2581 2582 2583 endif enddo return 2584 2585 *!*** 2585 *! 2586 *! 2587 *! 2588 *! 2589 *! This procedure calculates the number of deliveries Procedure: ONTIME done on time. Called by: SALREPO (procedure in SYS11.PRG) 2590 *i 2591 +j Calls: EOF() (FOXBASE+ function) 2592 * 2593 *!* e ste sie de 2594 2595 2596 procedure ontime select 1 2597 goto top 2598 do while .not. eof() 2599 replace ontimedely with d->per_ontime 2600 unlock 2601 2602 2603 -if .not. eof() skip endif 2604 -enddo 2605 return 2606 2607 *1*** This procedure calculates the variable (enqurec) where all the Procedure: ENQUSET record numbers of the enquires to be generated will be stored. 2608 *! +į 2609 2610 *! 2611 *! 2612 *! (procedure in SYS11.PRG) Called by: GAMCONF (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) 2613 *! Calls: EOF() 2614 *! 2615 *! : LTRIMO : STR() 2616 *! : RECNO() (FOXBASE+ function) *1 2617 2618 2619 procedure enquset 2620 enqurec = "" 2621 2622 select 5 2623 2624 2625 goto top do while .not. eof() enqurec = enqurec + ltrim(str(recno())) 2626 2627 2628 -if .not. eof() skip endif 2629 enddo 2630 2631 2632 return This procedure configurates the gaming-simulation to Procedure: GAMCONF a certain starting date. 2633 *! 2634 *!

2635	*!	
2636	*!	Called by: SYS11.PRG
2637		: NEXTDAY (procedure in SYSTI.PRG)
2639	**	Calls: CTODO (FOXBASE+, function)
2640	*1	: RESTVAR (procedure in SYS11.PRG)
2641	*1	: CHEKFILE (procedure in SYS11.PRG)
2642		: DTOC() (FOXBASE+ function)
2043	*	(EOXBASE + function)
2645	*1	: RESTART (procedure in SYS11.PRG)
2646	*	: REINDEX (procedure in SYS11.PRG)
2647	*!	: RESET (procedure in SYS11.PRG)
2648	- #1	: NUT:FILE() (FOXBASE+ function)
2650	*!	: ENGEN (procedure in SYS11.PRG)
2651	*!	: ADDREC (procedure in SYS11.PRG)
2652	*!	
2053	- T - #	Uses: SYSTEM.DBF
2655	*	: CHIAR.DDI
2656	*!	********
2657		procedure gamconf
2658		eseta antidata man milidata
2039		clear
2661		select 10
2662		store ctod("01/07/91") to confdate
2663		store factdate to confidate
2665		∞ 5 6 to 18 72 double
2666		set color to $g +/n$
2667		@ 6,24 to 8,54
2668		set color to $r^* + /n$
2669		@ 7,29 say "CONFIGURE DATE TO RUN."
2671		@ 12:16 say "ENTER THE GAME'S STARTING DATE :"
2672		@ 12,50 say confdate picture "@E"
2673		read
2674		set color to bg +/n
2676		@ 13.50 set confidate nicture "@E" valid confidate >=confidate
2677		read
2678		mmfactdate = conffdate
2680		do restvar with (confidate-1)
2681		mffdate=dtoc(mmfactdate)
2682		cmffdate = left(mffdate,2) + substr(mffdate,4,2)
2683		mar=cmffdate + "mar.dbf"
2084		if updated().or.confidate=confidate
2686		do restart
2687		do reindex
2688		select 10
2009		replace restart with f.
2691		unlock
2692		store j->waittime to mwaittime
2693		set view to run
2694		do reset
2696		set color to gr+*/n
2697		@ 14,25 say "CREATING THE ENQUIRES FILE"
2698		select 2
2099		copy stru to &mar
2701		use &mar
2702		do enquset
2703		do enggen
2704		do addree
2706		else
2707		set color to
2708		clear
2709		(@ 12,50 say "No existing past files." @ 13 28 say "Press any key to continue "
2711		Wait
2712		store j->waittime to mwaittime
2713	<	return
2715		else
2716		set color to
2717		@ 12,7 clear to 13,70
2/18		set color to $gr + /bu$ @ 5.6 to 18.72 double
~ ~ ~ ~ /		

the state water and weeks as an

a service and the service of the ser

- 5. 2 P. W.C. -

Same.

2720		set color to $g + /n$
2721		@ 6,24 to 8,54
2723		@ 7.25 say "CONTINUE RUN FROM ACTUAL DAY."
2724		set color to $bg + /n$
2725		@ 11,18 say "THE DATE TO CONTINUE FROM IS :" 22 mffdete
2727		set color to ,n/w
2728		close databases
2729		select 10
2731		replace restart with .t.
2732		unlock
2733		do reindex select 10
2735		use system
2736		replace restart with .f.
2738		store i->waittime to mwaittime
2739		set view to run
2740		do reset
2742		set color to gr+*/n
2743		@ 14,20 say "CREATING THE ENQUIRES FILE"
2744		select 2
2746		select 9
2747		use &mar
2748		do enquet
2750		endif
2751		do addrec
2753		mloon=0
2754		return
2755	*1	**
2757	*	This procedure adds an empty record to a datafile.
2758	*!	Procedure: ADDREC
2759	*!	Called by: GAMCONE (procedure in SVS11 DDG)
2761	*	Cance by, GAMCONIC (procedure in 31311.1 KG)
2762	*!	Uses: &MAR.DBF
2764	71 *!	*******
2765		procedure addrec
2766		######################################
2768		append from &mar
2769		return
2770	*!	***
2772	*i	This procedure calculates the name of the files to restore
2773	*1	Procedure: RESTVAR from or save to by the "saveday" and "restore" procedures.
2774	*! *!	Called by: GAMCONE (procedure in SYS11 PRG)
2776	*İ	: SAVEDAY (procedure in SYS11.PRG)
2777	켎	C-II DTOGO COVELOT L Andres
2779	*!	Calls: DIOC() (FOXBASE+ function) : CONFDATE() (FOXBASE+ function)
2780	*!	: LEFT() (FOXBASE+ function)
2781	*!	: SUBSTR() (FOXBASE+ function)
2783	+	********
2784		procedure restvar
2785		****
2787		vardate == dtoc(vardate)
2788		if conffdate=confdate.and.(mmfactdate-1) < > confdate
2789		vardate = "000/"
2791		vardate = left(vardate, 2) + substr(vardate, 4, 2)
2792		l-endif
2794		msys = varuale + "sys. ddl" mpla = vardate + "pla.dbf"
2795		mmar=vardate+"mar.dbf"
2796		mmac = vardate + "mac.dbf"
2798		$m_{sik} = vardate + "sik.ddf"$ mbil = vardate + "bil.dbf"
2799		mstu=vardate+"stu.dbf"
2800		msta = vardate + "sta.dbf"
2802		msalf=vardate + "salf.dbf"
2803		mstkp = vardate + "stkp.dbf"
2804		mmacn = vardate + "macn.dbf"

Variation and the state of the second

1 ... 2 ... 5

and the second second

a lot in this is

and the state of t

2805 mplap=vardate+"plap.dbf" 2806 return 2807 ** 2808 * *** 2809 *! This procedure calculates the screen position of the last 2810 *! enquire to be displayed. Procedure: POSIT 2811 *! Calls: .NOT.EOF() (FOXBASE+ function) : EOF() (FOXBASE+ function) 2812 *! 2813 *! 2814 *! ****** 2815 *!* 2816 2817 procedure posit 2818 2819 mposit = 1goto top do while .not.cof() 2820 2821 replace posit with mposit 2822 mposit=mposit+1 2823 2824 -if .not. eof() skip endif 2825 2826 =enddo 2827 2828 return 2829 *1 2830 *1 This procedure checks the existance of previously saved 2831 *! 2832 *! 2833 *! Procedure: CHEKFILE files in order to be restored. Called by: GAMCONF (procedure in SYS11.PRG) 2834 *İ 2835 * Calls: FILE() (FOXBASE+ function) 2836 *1 2837 2838 2839 ******* *11 procedure chekfile file1 = file("&msys") file2 = file("&mpla") file3 = file("&mpla") file4 = file("&mmac") file5 = file("&mstk") file6 = file("&mstk") file6 = file("&msalp") file7 = file("&msalp") file9 = file("&msalp") file11 = file("&msalp") file12 = file("&msalp") file12 = file("&msalp") file13 = file("&msalp") file14 = file("&msalp") file15 = fi 2840 2841 2842 2843 2844 2845 2846 2847 2848 2849 2850 2851 2852 2853 2854 2855 2856 pfiles=.t. else 2857 pfiles=.f. 2858 endif 2859 else pfiles=.f. 2860 2861 endif 2862 ** 2863 *!**** ********** 2864 *! 2865 *! This procedure creates the status file which is based on the Procedure: CREATSTAT sales orders and is used by the planning department to do the 2865 *! 2866 *! 2867 *! 2868 *! scheduling. Called by: RENSTAT (procedure in SYS11.PRG) Calls: .NOT.EOF() : CREATSTOR 2869 *į (FOXBASE + function) (procedure in SYS11.PRG) 2870 * 2871 * : CREATREPL (procedure in SYS11.PRG) 2872 *1 2873 *! Uses: TEMPSTAT.DBF 2874 *! 2875 *! Indexes: TEMPSTAT.IDX 2876 * 2877 *1 procedure creatstat 2878 2879 2880 private mtottime, crcode, crquant, crstk, crdate 2881 private credate, crresour, crcap, crtime, crsetup, crtottime 2882 store 0 to mtottime, crcode, crquant, crstk, crdate 2883 store 0 to credate, crresour, crcap, crtime, crsetup, crtottime 2884 set exclusive on 2885 select 7 2886 use tempstat index tempstat 2887 zap 2888 use 2889 set exclusive off

All the second and the

2890 2891 select 7 use tempstat index tempstat 2892 select 2 2893 2894 goto top =do while .not.eof() ______if status="FIRM".and..not.delivered 2895 2896 do creatstor 2897 do creatrepl 2898 select 2 2899 2900 2901 endif -if .not.eof() skip endif 2902 2902 2903 2904 2905 2906 2907 enddo select 7 total to status on cdate fields quantit, tottime use return 2908 2909 *!**** ****************************** 2910 *! This procedure freezes the departmental programs while the 2911 *! Procedure: RENSTAT system is re-calculating the status datafile. 2912 *1 2913 *1 2913 *1 2914 *1 2915 *1 Called by: RUNFACT (procedure in SYS11.PRG) (procedure in SYS11.PRG) (FOXBASE+ function) Calls: SYSTEM 2916 *! : LTRIM() : STRO : VALO 2917 *! (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) 2918 *! 2919 *! : SYSO 2920 *! : CREATSTAT (procedure in SYS11.PRG) * 2921 *! 2923 procedure renstat 2924 2925 select 10 2926 replace restatus with .t. 2927 unlock 2928 2929 2930 -do while system("restatus") var="LOCRS"+ltrim(str(tempdd)) public &var &var=val(sys(2)) 2931 2932 2933 2934 do creatstat var="LOCRF"+ltrim(str(tempdd)) 2935 2936 2937 public &var &var=val(sys(2)) select 10 2937 2938 2939 2940 2941 2942 *!' 2943 *! replace restatus with .t. unlock return ** *1**** This procedure updates the system datafile's fields 2944 *! after a restart from previously saved files has been done. Procedure: REPLSYS 2945 *! 2946 *! Called by: RESTART (procedure in SYS11.PRG) 2947 *! 2948 *! Uses: &MSYS 2949 *! 2950 *! : SYSTEM.DBF 2951 *!** procedure replsys 2952 2953 2954 set exclusive off 2954 2955 2956 2957 2958 2959 select 9 use &msys select 10 replace; startdate with i->startdate;; factdate with i->factdate;; 2960 2961 hour rate with i->hour rate,; 2962 tot overh with i->tot overh,; 2962 2963 2964 2965 2966 2967 frequency with i-> frequency,; waittime with i->waittime replace; throughput with i->throughput,; assets with i->assets,; systock with i->systock,; enquperiod with i->enquperiod 2968 2969 2970 unlock 2971 select 9 2972 use select 10 2973 2974 use system

ALL LAND

240

Are tool bak

V. S. 1. 6. 65

けいし いんか ないないない あいいろかいのな

2975 return 2975 10 2976 *** 2977 *!*** 2978 *! 2979 *! This procedure calculates the status datafile's fields Procedure: CREATSTOR based on the existing sales orders. 2980 *! 2981 *! Called by: CREATSTAT (procedure in SYS11.PRG) 2982 *! lls: DTOC() (FOXBASE+ function) : TRANSFORM() (FOXBASE+ function) : INT() (FOXBASE+ function) : MOD() (FOXBASE+ function) 2983 *! Calls: DTOC() 2984 *! 2985 *! 2986 *i 2987 *1 2988 ****** * *** 2989 procedure creatstor 2990 2990 2991 2992 2993 2994 store (enquantit*e->procestime)+f->setup to mtottime store code to crcode store enquantit to crquant store d->stock to crstk 2995 store gudeliver to crdate 2995 2996 2997 2998 2999 store code + dtoc(qudeliver) to crcdate store c->resour_cod to crresour store c->capacity to creap store e->procestime to crtime store f->setup to crsetup store transform(int(mtottime/60), "9") + "h" + transform(mod(mtottime,60), "99") + "m" to crtottime 3000 3001 3002 ** 3003 3004 *! 3005 *! This procedure stores the values calculated by the "creatstor" Procedure: CREATREPL procedure in the status datafile 3006 *! 3007 *! Called by: CREATSTAT (procedure in SYS11.PRG) 3008 *! 3009 * 3010 procedure creatrepl 3011 3012 select 7 3013 append blank replace code with crcode replace quantit with crquant 3014 3015 replace stock with crstk replace date with crdate 3016 3017 3018 replace cdate with credate 3019 replace resource with crresour 3020 3021 3022 replace capacity with crcap replace time with crtime replace setup with crsetup replace tottime with crtottime unlock 3023 3024 3025 ** 3026 *!****** 3027 *! 3028 *! Pro ******** This procedure deletes the records which havn't been Procedure: DELETC quoted before their delivery date. 3029 *] 3030 *! Called by: REINDEX (procedure in SYS11.PRG) procedure deletc 3033 3034 3035 goto top 3036 delete for endeliver < mmfactdate.and.status = "ENQU" 3037 return 3038 3040 *: EOF: SYS11.ACT

Sa 14

State sto

as actorises to be a chief to the

System: SYS11.PRG Author: JUAN IGNACIO IGARTUA 08/05/91 18:23:39 Tree Diagram

SYS11.PRG
ERROR() (function in ?)
ERR_FIX (procedure in SYS11.PRG)
RLOCK() (function in ?)
CIOD() (function in ?)
SYSTEM.DBF (database)
&MAR.DBF (database)
CTOD() (function in ?)
RESTVAR (procedure in SYS11.PRG)
CONEDATE AND (function in ?)
LEFT() (function in ?)
SUBSTR() (function in ?)
CHEKFILE (procedure in SYS11.PRG)
FILE() (function in ?)
LICCU (function in ?)
RESTART (procedure in SYS11.PRG)
DTOC() (function in ?)
REPLSYS (procedure in SYS11.PRG
SYSTEM.DBF (database)
REINDEX (procedure in SYS11.PRG)
SYSTEM.DBF (database)
PLANN.DBF (database)
MACHINE DBF (database)
STOCK.DBF (database)
BILL.DBF (database)
SALPAST.DBF (database)
MACHPAST DBF (database)
PLAPAST.DBF (database)
STKPAST.DBF (database)
SETUP.DBF (database)
DELEIC (procedure in SYS11.PRG)
INTO (function in ?)
VALO (function in ?)
SUBSTR() (function in ?)
LEFTO (function in ?)
RIGHT() (function in ?)
.NOT.FILE() (function in ?)
ENQUSET (procedure in SYS11.PRG)
EUP() (function in ?)
STR() (function in ?)
RECNO() (function in ?)
ENQGEN (procedure in SYS11.PRG)
VALO (function in ?)
RIGHT() (function in ?)
TIME() (function in ?)
ROUND() (function in ?)
KND (procedure in SYS11.PRG)
SUBSTR() (function in ?)
MMFACTDATE<() (function in ?)
MMFACTDATE>() (function in ?)
IIF() (function in ?)
STUFF() (function in ?)
ADDREC (procedure in SYS11.PRG)
&MAR.DBF (database)
OPEN (procedure in SYS11.PRG)
PLANN.DBF (database)
MARKET.DBF (database)
MACHINE.DBF (database)
BILL DBE (database)
SETUP.DBF (database)
SYSTEM.DBF (database)

.....

1. 20 - ---and the second

(and the second

S. Ash 200

2. Same

1 Sin int

12 .

S. 2. J.

1 4 6

all a start


a the second state of the second

n". Annär were etter Standeterne Seffer en som etter med Standarf. Daravsterat vära som er bi

an and B. Canton

and the state of t

No North Part

W. W. Denn Ma Vill Bart

243

الأقوان ومستعد منتجار المراسية للتلاطية والرابع ومستوريت بمراجع المتناط بالمتحال والمتحار المتكافر مراجع أراجه أوالا والمتحاد

-SAYTIME1 (procedure in SYS11.PRG) -RECNO() (function in ?) -.AND..NOT.EOF() (function in ?) DAYREPOR -ONTIME -FREOUEN -EOF() (function in ?) -.NOT.EOF() (function in ?) -VAL() (function in ?) -STR() (function in ?) -LEFT() (function in ?) -RIGHT() (function in ?) -CTOD() (function in ?) SYSTEM (procedure in SYS11.PRG) ——STR() (function in ?) ——SELECT() (function in ?) .OR.SYSTEM() (function in ?) REPOMENU (procedure in SYS11.PRG) ——SAVEDAY (procedure in SYS11.PRG) SYSTEM REPOMENU SAVEDAY (procedure in SYS11.PRG) &MAR.DBF (database) CTOD() (function in ?)

and see as the second second second second second second second second second second second second second second

A State of the second second second second

2.5. 1

in an an the second the second to the second of the second state and the second of the second to be the second



The state of the s

State of the state

No. S. ... Dan

and and an and and

description in .

System: SYS11.PRG Author: JUAN IGNAC 08/05/91 18:23:2 Database Structure 	IO IGARTUA 5 Summary e system					
TEMPSTAT.DBF &MSYS						
Structure for data	base : SYSTE	M.DBF				
Number of data rec Last upd Field Field name 1 STARTDATE 2 FACTDATE 3 FACTTIME 4 SYS STOP 5 GAM STOP 6 HOUR RATE 7 TOT OVERH 8 NEWENQ 9 NEWENQREC 10 FREQUENCY 11 GOODSERV 12 REINDEX 13 STATRESO 14 STATCODE 15 WAITTIME 16 RESTART 17 RESTATUS 18 THROUGHPUT 19 ASSETS 20 SYSTOCK 21 SALREPORT 23 ENQUPERIOD ** Total **	ords : lated : 07/01 Type Date Date Date Character Logical Logical Character Numeric Logical Character Numeric Logical Logical Logical Logical Numeric Numeric Numeric Numeric Numeric Numeric Logical Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric	/91 at Width 8 4 1 7 10 9 3 1 1 4 4 4 1 10 10 10 10 10 10 102	9:17 Dec 2 2 2 2	Start 97 222 300 410 554 559 67 899 101	End 860 221 2299 3492 5534 82667 888 990 101	
FoxDoc did not fir	nd any associ	ated ind	ex files	3		
Used by: SYS11.PRG : GAMCONF : REINDEX : OPENRUN : OPENSAL : OPENPLA : REPLSYS	(proce (proce (proce (proce (proce (proce	dure in a dure in a dure in a dure in a dure in a dure in a	SYS11.PF SYS11.PF SYS11.PF SYS11.PF SYS11.PF SYS11.PF	3G) 3G 3G 3G 3G 3G 3G 3G 3G 3G 3G 3G 3G 3		
Structure for data Number of data rec Last upo Field Field name 1 STARTDATE 2 FACTDATE 3 FACTTIME 4 SYS STOP 5 GAM STOP 6 HOUR RATE 7 TOT OVERH 8 NEWENQ 9 NEWENQREC 10 FREQUENCY 11 GOODSERV	base : TEMP. ords : lated : 07/01 Type Date Date Character Logical Numeric Logical Character Numeric Logical Character Numeric Logical	DBF 1 /91 at Width 8 4 1 7 10 1 9 3 1	8:50 Dec 2 2	Start 1 9 17 21 23 30 40 41 50 53	End 8 16 20 21 22 29 39 40 49 52 53	

and the state of the second second state of the state

and a start of the second second second second second second second second second second second second second s

So to a strain

and the of the state of the sta

12 REIN 13 STAT 14 STAT 15 WAIT 16 REST 17 REST 18 THRC 19 ASSI 20 SYST 21 SALH 22 PLAH 23 ENOU ** Total **	NDEX LC IRESO Ch ICODE Ch ITIME NU IATUS LC DUGHPUT NU ETS NU IOCK NU REPORT LC REPORT LC UPERIOD NU	ogical maracter meric ogical meric meric ogical ogical ogical ogical ogical ogical	1 4 4 1 10 10 10 10 10 10 10 10 10 10 10 10 1	2 22 2	54 55 63 67 68 69 79 99 900 01	54 58 62 66 67 68 78 88 99 99 100 101
FoxDoc did Used by: SA	not find a AVEDAY	iny associated (procedure	index f	files (1.PRG)		
Structure f Number of c I Field Fiel 2 DESC 3 QUAN 4 STAN 5 STAN 6 ORD 7 SEQU 8 RESC 9 PROI 10 PROI 11 ORD 12 ORD 13 FIN 14 FIN 15 NEEN ** Total ** This databa : 01 : NZ	for databas data record Last update Last update CRIPT Ch NTIT NU RT DATE Da RT TIME CH ER NUM NU UENCE NU DUCT 1 NU DUCT 1 NU DUCT 1 NU STAT 1 CH STAT 1 CH STAT 1 CH STAT 1 CH STAT 1 CH STAT 1 CH STAT 1 NU STAT 1 CH STAT 0 CH STAT 1 CH	se : PLANN.DBF ls : 5 cd : 07/01/91 vpe Wid haracter maracter maracter meric haracter meric haracter haracter haracter haracter haracter haracter to haracter	at 9:17 th De 20 6 8 4 4 4 4 10 10 8 8 8 4 6 09 ated wit CHR (SEQU	2 2 2 2 2 2 2 2 2 2 1 2 1 2 1 2 1 2 1 2	file(s):	End 4 30 38 42 46 50 54 64 74 82 90 98 102 108
Used by: RI : Ol : Ol	EINDEX PENRUN PENPLA	(procedure (procedure (procedure	in SYSI in SYSI in SYSI	1.PRG) 1.PRG) 1.PRG)		
Structure f Number of o Field Fiel 2 DESG 3 ENO 4 ENF 5 END 6 ENT 7 END 8 QUP 9 QUD 10 QUT 11 QUD 12 STA 13 REA 14 DEL 15 SUCC 16 SUCC 16 SUCC 17 WAI ** Total **	for databas data record Last update la name Ty CRIPT Cr UANTIT NU RICE NU ATE Da IME Cr ELIVER Da IME Cr ELIVER Da IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME Cr ELIVER DA IME CR ELI	ae : MARKET.DB ls : 10 ed : 07/01/91 wid haracter haracter meric haracter hara	F at 9:17 th De 20 46 84 86 84 86 84 86 84 81 66 410	2 2 2 2 2 2	art 1 25 229 35 43 43 43 45 55 61 69 73 88 5 93 940 00 00	End 4 24 28 34 42 46 54 60 68 72 80 84 92 93 93 93 93 93 93
This databa : DI : WA : SA	ase appears EMARK.IDX AIT.IDX (F ALDELIV.IDX	to be associ (QUDELIVER) RECNO()) C (QUDELIVER)	ated wit	h index	file(s):	
Used by: RI	EINDEX	(procedure	in SYS1	1.PRG)		

5 1 12 . Ch ?!

i there are a street

in the

and a start of a substanting of the second starts that a

And a star of the second of the second

(procedure in SYS11.PRG) (procedure in SYS11.PRG) : OPENRUN : OPENSAL Structure for database : MACHINE.DBF Number of data records : 3 Last updated : 07/01/91 at Field Field name Type Width 1 RESOUR COD Character 4 2 REDESCRIPT Character 15 3 NUM 9:13 Start End Dec 4 19 22 27 31 38 44 1 1234 15 3 5 20 23 28 39 AVAIL Character AVAIL T AVAIL CAPACITY LASTSETUP SET UP1 SET UP2 SET UP3 5 Numeric 476 5 Numeric Character Numeric 6 7 8 22222 66 50 45 Numeric 51 57 63 ē Numeric **Š**6 SET UP4 SET UP5 62 68 10 Numeric 6 īī Numeric 6 Total ** 69 ** This database appears to be associated with index file(s): : RESOUR.IDX (RESOUR_COD) (procedure in SYS11.PRG) (procedure in SYS11.PRG) (procedure in SYS11.PRG) Used by: REINDEX : OPENRUN ż **OPENPLA** Structure for database : STOCK.DBF Number of data records : 5 Last updated : 07/01/91 at Field Field name Type Width 1 CODE Character 4 9:13 Dec Start End 20 23 24 32 DESCRIPT Character 5 25 STOCK Numeric 8 ** Total ** зā This database appears to be associated with index file(s): : CODESTK.IDX (CODE) Used by: REINDEX : OPENRUN (procedure in SYS11.PRG) (procedure in SYS11.PRG) (procedure in SYS11.PRG) : OPENPLA Structure for database : BILL.DBF Number of data records : 5 Last updated : 07/01/91 at Field Field name Type Width 9:09 Field name Type Character Dec Start End CODE 4 1 4 23 24 26 28 34 40 20226 DESCRIPT Character 5 MINDELIVDD MAXDELIVDD 222344556677889 Numeric Numeric 4 MAXQUANTIT MINQUANTIT STANDPRICE 5 Numeric 2222 67 Numeric 6664 46 52 Numeric MATCOST TOT ENQU PRICE LIMT DISCOUNT QUANT DISC DELIV-LIMT MRKT PRICE MRKT-LIMT ONTIMEDELY 8 9 Numeric Numeric 5626726 76288 94 2 2 2 10 11 12 13 Numeric 64646666 Numeric Numeric Numeric 14 NNNNNNN Numeric Numeric 16 17 ONTIMEDELV Numeric ONTIMLIMT Numeric 100 STANDHOURS OVERHEAD 101 107 113 106 112 118 18 Numeric ē 19 20 22 23 23 23 24 Numeric 6 STAND COST Numeric 6 NUM ENQU RESOUR COD PROCESTIME SET_UP 119 123 127 132 122 Numeric 4 126 131 457 Character 2 Numeric Character 138 Total ** ** 139

248

Section for

and a ke land thing to that a

This database appe ; BILLNA.ID	ars to be as X (CODE)	sociated	l with i	ndex file(в):	
Used by: REINDEX : OPENRUN : OPENSAL : OPENPLA	(proce (proce (proce (proce	dure in dure in dure in dure in	SYS11.P SYS11.P SYS11.P SYS11.P	RG) RG) RG) RG)		
Structure for data Number of data rec Last upd Field Field name 1 CODE	base : SALPA ords : ated : 07/01 Type Character Character	ST.DBF 1 /91 at Width 4	9:17 Dec	Start 1	End 4	
3 DATE 4 AVE ENQU 5 AVE OUOT 6 PER OUOT 7 AVE FIRM 8 PER FIRM 9 OVR DELTIM 10 PER OPRICE 11 AVE OPRICE 12 PER OPRICE 13 PER OELIV 14 PER FPRICE	Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric	2866666666666666	~~~~~~	2339 2339 553 739 567 785 7851	32840 56262 77840 96	
15 PER [_] FDELIV 16 DELIVERIES 17 VOLUME ** Total **	Numeric Numeric Numeric	6 6 10 119	2 2	97 103 109	102 108 118	
This database appe : CDSAL.IDX	ars to be as (CODE+DTOC	sociated (DATE))	l with i	ndex file(s):	
Used by: REINDEX : OPENSAL	(proce (proce	dure in dure in	SYS11.P SYS11.P	RG) RG)		
Structure for data Number of data rec	base : SALPE ords : ated : 07/01	RF.DBF	9.09			
Field Field name 1 CODE 2 DESCRIPT 3 DATE 4 AVE ENQU 5 AVE OUOT 6 PER OUOT 7 AVE FIRM 8 PER FIRM 9 OVR DELTIM 10 PER OPRICE 11 AVE OPRICE 12 PER OPRICE 13 PER ODELIV	Type Character Date Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric	Widt4 208666666666666666666666666666666666666	Dec 222222222222222222222222222222222222	Start 5 25 339 45 57 63 67 73 85	End 242 344 562 626 78 80 78 80	
14 PER ⁻ FPRICE 15 PER ⁻ FDELIV 16 DELIVERIES 17 VOLUME ** Total **	Numeric Numeric Numeric Numeric	6 6 10 119	2 2 2	91 97 103 109	96 102 108 118	
This database appe : CPRFSAL.I	ars to be as DX (CODE)	sociated	l with i	ndex file(s):	
Used by: REINDEX : OPENSAL	(proce (proce	dure in dure in	SYS11.P SYS11.P	RG) RG)		
Structure for data Number of data rec Last upd Field Field name 1 RESOUR COD 2 DESCRIPT 3 CAPACITY 4 PRODUCTION	base : MACHP ords : ated : 07/01 Type Character Character Numeric Numeric	AST.DBF 3 /91 at Width 4 15 4 7	9:09 Dec 2	Start 5 20 24	End 4 19 23 30	

.....

11.4

and the second second of the second s

the second to be the second with a second se

and a second

249

aline in a state of the second second second second second second second second second second second second se

5 6 7 8 ** Tota	PRODUCTIV SETUPTIME SETUPNUM DATE 1 **	Numeric Numeric Numeric Date	6 7 2 8 54	22	31 37 44 46	36 43 45 53	
This da	tabase appea : CDMACH.ID	rs to be as (RESOUR_C	sociated	l with i (DATE))	index file(∃):	
Used by	: REINDEX : OPENPLA	(proce	dure in dure in	SYS11.1 SYS11.1	PRG) PRG)		
Structu Number Field 1 2 3 4 5 6 7 8 9 10 11 12 ** Tota	are for data of data reco Last upda Field name CODE DESCRIPT INCOMPLETE STARTED FINISHED PRODUCTION WIP AVE BATCH SETUPNUM LEADTIME STAND COST DATE Al **	pase : PLAPA ords : ited : 07/01 Type Character Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Date	ST.DBF 0 /91 at Width 20 3 3 10 10 6 8 83	9:09 Dec 2 2 2 2 2 2	Start 1 5 25 28 31 34 44 54 60 63 69 75	End 4 24 27 30 33 43 59 62 68 74 82	
This da	tabase appea : CDPLA.IDX	ars to be as (CODE+DTOC	sociated	i with i	index file(8):	
Used by	: REINDEX : OPENPLA	(proce (proce	dure in dure in	SYS11.1 SYS11.1	PRG) PRG)		
Structu Number Field 1 2 3 4 ** Tota	ire for data of data reco Last upda Field name CODE DESCRIPT STOCK DATE al **	Dase : STKPA Drds : Ated : 07/01 Type Character Character Numeric Date	ST.DBF 5 /91 at Width 4 20 8 8 41	9:09 Dec	Start 1 25 33	End 4 24 32 40	
This da	tabase appea	irs to be as	sociated	1 with i	index file(5):	
Used by	: REINDEX : OPENPLA	(proce (proce	dure in dure in	SYS11.1 SYS11.1	PRG) PRG)		
Structu Number Field 2 3 ** Tota	are for data of data reco Last upda Field name CODE TYPE SETUP al **	Dase : SETUF Drds : ated : 07/01 Type Character Character Numeric	.DBF 5 ./91 at Width 4 7 6 18	9:09 Dec 2	Start 1 5 12	End 4 11 17	
This da	: SETUP.IDX	ars to be as (CODE)	sociated	l with i	index file(B):	
Used by	: REINDEX : OPENRUN : OPENPLA	(proce (proce (proce	dure in dure in dure in	SYS11.F SYS11.F SYS11.F	PRG) PRG) PRG)		

and a state of the second of the state of the

1. 2va

while a lot of the second second second

als a religion of success in thick a third and the how we are an

" the Breed around 22 the second

10.00

See. of

&MAR.DBF is a macro unknown to FoxDoc

FoxDoc did not find any associated index files Used by: GAMCONF (procedure in SYS11.PRG) : ADDREC (procedure in SYS11.PRG) &MAR is a macro unknown to FoxDoc FoxDoc did not find any associated index files Structure for database : TEMPSTAT.DBFNumber of data records : 7Last updated : 07/01/91 at 9:13Field Field name Type Width Dec Start1 CODE Character 42 OUANTIT Numeric 43 STOCK Numeric 84 DATE Date 86 CAPACITY Numeric 47 TIME Numeric 5 28 SETUP Numeric 6 29 TOTTIME Character 59 TOTTIME Character 42 OUANTI 89 TOTTIME Character 49 TOTTIME Character 59 TOTTIME Character 59 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME Character 49 TOTTIME 110 CDAT End 48 148233348 60 This database appears to be associated with index file(s): : TEMPSTAT.IDX (CDATE) Used by: CREATSTAT (procedure in SYS11.PRG) &MSYS is a macro unknown to FoxDoc FoxDoc did not find any associated index files (procedure in SYS11.PRG) Used by: REPLSYS

- adding the

a for an en la carto se esta su de la carto de la contra de la contra de la contra de la contra de la contra d

Construction of the state of the second of the second second with the second second second second second second

المحاج والمحاج والمحاج المحاج والمحاج والمحاج والمحاج والمحاج والمحاج والمحاج المحاج والمحاج والمحاج والمحاج والمحاج

System: SYS11.PRG Author: JUAN IGNACIO IGARTUA 08/05/91 18:23:29 Data Dictionary

and the set while the same where the set of the set of the set of the set of the set of the set of the set of t

1. A.

Alter William Later

Gen a service de services

Field Name ASSETS	Type N	Len 10	Dec 2	Database SYSTEM.DBF
λ Υλ ΤΤ .	C	2	0	MACUTHE DEF
AVALD	N	5	2	DIADAGE DEF
AVE DAICH	N	6	4	CALDEDE DEE
AVE_ENQU	14	0	2	CATDACT DDC
AVE ETOM	N	6	2	CAT DACM DOF
AVE_FIRM	11	6	2	CALPAST DEF
AVE ODDICE	NT	e	2	CALPERF.DDF
AVE_QFRICE	14	0	2	CALPAST .DDF
AVE OTOT	N	E	2	CALDEDE DEE
AVE_QUUT	11	0	2	CALPERF.DDF
CADACTON	NT		~	SALPAST DBF
CAPACITY	11	4	U	TEMPSTAT.DBF
				MACHPAST.DBF
OD A THE	0	10	•	MACHINE.DBF
CDATE	C	12	0	TEMPSTAT.DBF
CODE	C	4	0	SALPAST UBF
				PLANN. DBF
				SETUP.DBF
				STRPAST.DBF
				BITT DBL
				TEMPSTAT.DBF
				SALPERF.DBF
				PLAPAST.DBF
				MARKET DBF
D3.000		•	•	STOCK DBF
DATE	D	8	0	SALPAST.DBF
				PLAPAST.DBF
				TEMPSTAT.DBF
				MACHPAST.DBF
				SALPERF.DBF
	-	-	-	STKPAST.DBF
DELIVERED	Ц.	1	0	MARKET.DBF
DELIVERIES	N	6	0	SALPAST.DBF
			_	SALPERF.DBF
DELIV LIMT	N	4	0	BILL.DBF
DESCRIPT	C	20	0	STKPAST.DBF
				PLANN.DBF
				PLAPAST.DBF
				SALPAST.DBF
				STOCK.DBF
				MARKET.DBF
				BILL.DBF
				SALPERF.DBF
DESCRIPT	C	15	0	MACHPAST.DBF
DISCOUNT	N	4	2	BILL.DBF
ENDATE	D	8	0	MARKET.DBF
ENDELIVER	D	8	0	MARKET.DBF
ENPRICE	N	6	2	MARKET.DBF
ENQUANTIT	N	4	0	MARKET.DBF
ENQUPERIOD	N	1	0	SYSTEM.DBF
				TEMP.DBF
ENTIME	С	4	0	MARKET.DBF
FACTDATE	D	8	0	SYSTEM.DBF
				TEMP.DBF
FACTTIME	С	4	0	SYSTEM.DBF
				TEMP.DBF
FINISHED	N	3	0	PLAPAST.DBF
FINIS DATE	D	8	0	PLANN, DBF
FINIS TIME	С	4	0	PLANN.DBF
FREQUENCY	N	3	0	TEMP.DBF

252

1 18 . m. m. 1 1992 to . . .

CAN CHOD	Ŧ	-	0	SYSTEM.DBF
GAM_STOP	L.	T	0	TEMP.DBF SYSTEM.DBF
GOODSERV	\mathbf{L}	1	0	TEMP.DBF
HOUR_RATE	N	7	2	TEMP.DBF
TNCOMDLETE	N	2	0	SYSTEM.DBF
LASTSETUP	C	7	ŏ	MACHINE, DBF
LEADTIME	Ň	6	2	PLAPAST.DBF
MATCOST	N	6	2	BILL.DBF
MAXDELIVDD	N	2	0	BILL.DBF
MINDELTVDD	N	2	20	BILL DBF
MINOUANTIT	Ň	õ	2	BILL DBF
MRKT LIMT	N	6	2	BILL.DBF
MRKT_PRICE	N	6	2	BILL.DBF
NEED TIME NEWENO	N T	6	2	PLANN.DBF
NEWENQ	ш	–	U	SYSTEM, DBF
NEWENQREC	С	9	0	TEMP.DBF
		_		SYSTEM.DBF
NUM ENQU	N	4	0	BILL.DBF
ONTIMEDELV	N N	6	2	BILL DBF
ORDER NUM	Ň	4	õ	PLANN, DBF
ORD STAT 0	Ĉ	8	ŏ	PLANN.DBF
ORD STAT 1	С	8	0	PLANN.DBF
OVERHEAD	N	6	2	BILL.DBF
OVR_DELITIM	N	4	2	SALPAST DBF
PER FDELIV	N	6	2	SALPAST DBF
		-	-	SALPERF.DBF
PER_FIRM	N	6	2	SALPAST.DBF
DED FODTOF	N	6	2	SALPERF.DBF
FER_FFRICE	IN	0	2	SALPAST.DDF
PER ONTIME	N	6	2	SALPAST.DBF
		~	-	SALPERF.DBF
PER_QDELIV	N	6	2	SALPAST.DBF
PER OPRICE	N	6	2	SALPERF. DBF
				SALPAST.DBF
PER_QUOT	N	6	2	SALPERF.DBF
DI AREDORT	т.	1	0	SALPAST.DBF
		T	0	SYSTEM, DBF
PRICE LIMT	N	6	2	BILL.DBF
PROCESTIME	N	5	2	BILL.DBF
PRODUCTION	N	10	2	MACHPAST.DBF
PRODUCTIV	N	10	2	MACHPAST DBF
PRODUCT 0	Ñ	10 1	2	PLANN.DBF
PRODUCT 1	N	10	2	PLANN.DBF
QUANTIT OUR	N	6	2	PLANN.DBF
OUANTIT OUANT DISC	N N	4	0	TEMPSTAT.DBF
OUDATE	D	8	õ	MARKET, DBF
QUDELIVER	D	8	ō	MARKET . DBF
QUPRICE	N	6	2	MARKET.DBF
QUTIME DEALDELTY	C	4	0	MARKET DBF
REDESCRIPT	C	15	0	MARKET.DBF MACHINE DBE
REINDEX	Ľ	1	ŏ	TEMP. DBF
				SYSTEM.DBF
RESOURCE	c	4	0	TEMPSTAT.DBF
VF200K_COD	C	4	0	PLANN. UBF MACHDACT DEF
				BILL.DBF
				MACHINE.DBF

Section 18 and section 1

いたいはくいい たんじゅうい 大学学 一年 そういだい じゅうう ドレング あいたみ レンドリー たんじん ちょう かんかい しょう

Contraction on the second second

6 ...

RESTART	L	1	0	TEMP.DBF
RESTATUS	L	1	0	TEMP.DBF
SALREPORT	т.	1	0	SYSTEM.DBF
		1	U	TEMP.DBF
SEQUENCE	N N	4	02	PLANN.DBF SETUP DBF
			~	TEMPSTAT.DBF
SETUPNUM	N N	2	0	MACHPAST.DBF
SETUPTIME	Ň	7	ž	MACHPAST DBF
SET UP	ĉ	7	ō	BTLL, DBF
SET UP1	Ň	6	2	MACHINE DBF
SET UP2	N	6	2	MACHINE.DBF
SET UP3	N	6	2	MACHINE.DBF
SET UP4	N	6	2	MACHINE.DBF
SET UP5	N	6	2	MACHINE.DBF
STANDHOURS	N	6	2	BILL.DBF
STANDPRICE	N	6	2	BILL.DBF
STAND_COST	N	6	2	PLAPAST.DBF
				BILL.DBF
STARTDATE	D	8	0	TEMP.DBF
		-	-	SYSTEM.DBF
STARTED	N	3	0 0	PLAPAST DBF
START DATE	D	8	U U	PLANN DBF
START TIME	C	4	0	PLANN.DBF
STATCODE	C	4	0	SYSTEM.DBF
CUINDECO	0		0	TEMP.DBF
STATRESO	C	4	0	SISTEM.DBF
STATUS	C	٨	0	MADEFT DEF
STOCK	Ň	a a	ŏ	STERAST DEF
DIOOR		0	U	STOCK DBF
				TEMPSTAT, DBF
SUCCSDELIV	N	6	2	MARKET, DBF
SUCCSPRICE	N	Ğ	$\overline{2}$	MARKET, DBF
SYSTOCK	N	10	2	TEMP, DBF
				SYSTEM. DBF
SYS STOP	\mathbf{L}	1	0	SYSTEM. DBF
				TEMP.DBF
THROUGHPUT	N	10	2	SYSTEM.DBF
				TEMP.DBF
TIME	N	5	2	TEMPSTAT.DBF
TOTTIME	C	5	0	TEMPSTAT.DBF
TOT ENQU	N	4	0	BILL.DBF
TOT_OVERH	N	10	2	SYSTEM.DBF
my D P	0	-	-	TEMP.DBF
TIPE M AVATT	C	/	0	SETUP.DBF
	N	10	ő	MACHINE DBF
VOLUME	N	10	2	SALPERF.DBF
ωατπ	N	٨	0	DALFAST. UBF
WATTTTMF	NI IN	4	0	TARALI. DDF
	14	'1	U	SVSTEM DRE
WTP	N	10	2	DIADAST DEF
· · · · · · · · · · · · · · · · · · ·	11	T 0	دعه	T TULE VOL . DDL

an enter tri u de l'herene de nord a num a construction de la construction de la construction de la constructio

PLANNING/CONTROL SUB-PROGRAM

PLA4. PRG

1	*.*	******************
2	*:	This program is the one that simulates all the activities
3	*:	Program: PLA4.PRG performed by the planning department. Together with the Sales
4	Ţ.	and System ones, it conforms the Notlingham Polytechnic's
6	*	Author: IUAN IGNACIO IGARTUA
7	*	Copyright (c) 1991, NOTTINGHAM POLYTECHNIC
8	*:	Last modified: 08/05/91 18:06
9	*:	Prove & Franker FDD, FRV
11	*	Proce & Proces ERK PIA
12	*:	OPEN
13	*:	: CHEKSTOP
14	*:	: INPTMPS
15	1	: MAINMENU
17	*.	· NEWPLAN
18	*:	: PLEDIT
19	*:	: PL_GETS_
20		: INCLUDE
22		
23	*:	: PL STOR
24	*:	: CALORD
25	*:	COUNTDEL
27	+.	SAM DISP
28	*:	STA DISP
29	*:	: PLA_AREA0
30	*:	PLA_AREA1
32	*.	
33	*:	RESOURCE
34	*:	: FILT_CODE
35	*:	: FILT RESO
37	*.	· POSITION
38	*:	: PLANREPO
39	*:	: PLAREPO
40	*:	: PLAAVE
41	*.	: MACHAVE
43	*:	: PRT
44	*:	: PLAPAST
45	*:	: MACHPAST
40	*.	· OFFANEW
48	*:	
49	*:	Calls: ERROR() (FOXBASE + function)
50		: MESSAGE() (FOXBASE+ function)
52		: BKR_FLA ((FOCCENER IN FLA4.FKG)
53	+:	CHR() (FOXBASE+ function)
54	*:	: SYSTEM() (function in PLA4.PRG)
33	-	: OPEN (procedure in PLA4.PRG)
57	*:	: CHEKSTOP (noredure in PLA4 PRG)
58	*:	: INPTMPS (procedure in PLA4.PRG)
59	*:	
61	7	Uses: SYSTEM.DBF
62	*	Documented 08/05/91 at 18:26 FoxDoc version 1.0
63	*:	************************
64		The set of the set of
66		set procedure to pla4 pro
67		on error do err fix with error(), message()
68		set default to v:
69		T and a secolar and
71		
72		set delete on
73		set status off
75		set talk oli
76		set scoreboard off
77		set heading off
78		*
80		puone pgun, countrey, pgup, acirecoru, recnum, cnoice, promptrow, promptoar;; oldrecnum.screenatr.statusatr.windowatr.nhrase.nhrase.nonew.emoty
81		public promptatr, hiliteatr, filtreso, filtcode, reschoice, actfilt, lphrase;
82		codchoice, rowbottom, rowprompt, listlast, reccord, lfiltcode, lfiltreso
84		public morder num, mresour co, mneedlim, mquantit, mord stat 1, mord stat 0, otdel,; mdeliver, moode, mdescript, mfound, nout indsel indmas maspaul stangut :

antheres as a

18. 16 al

· · ·

the court first in a

and the state of t

mproduct 1,mproduct 0,signal,srnlines,prtlines,r,c,msequence,pfilt 85 public sys_start, signal 1, statpage, pause, start, daystat, editing 86 87 88 89 90 srnlines=6 prtlines=55start=.f. 91 92 93 94 95 96 97 signal = .f.signal 1 = .f.sys start=.t. pause = .f.daystat = .f.nonew = .f. empty=.f. 98 99 editing=.f. 100 store "orplann" to indmps 101 store scalines to mpspout 102 store srnlines to stopout store "namark" to indsel 103 104 store srnlines to pout 105 screenatr="R+/N,N/W" statusatr="BU/N,N/W" windowatr="R+/N,N/W" promptatr="GR+/N,N/W" hiliteatr="N/W" 106 107 108 109 110 111 112 rowbottom = 20113 rowprompt = rowbottom + 3114 115 store 0 to recnum,oldrecnum store " " to choice store "1" to reschoice store "0" to codchoice 116 117 118 store 0 to msequence promptrow=22 119 120 121 promptbar=replicate(chr(196),80) pgdn=chr(3) pgup=chr(18) 122 123 124 125 returnkey=chr(13) delrecord=chr(7) signal=.f. 126 127 128 set exclusive off 129 130 select 10 use system 131 132 scr=.f. dowhile system("reindex") && wait until the reindex is finished —if .not.scr && in the system program. 133 134 set color to bu+/n @ 7,24 to 12,54 double set color to gr+/n @ 9,25 say "REINDEXING THE MASTER SYSTEM" @ 10,34 say "PLEASE WAIT" 135 136 137 138 139 scr=.t. endif 140 141 142 143 144 145 146 enddo ** set color to clear set color to bu +/n @ 7,24 to 12,54 double set color to gr +/n @ 9,25 say "SETTING THE PLANNING SYSTEM" @ 10,34 say "PLEASE WAIT" 147 147 148 149 150 151 152 153 154 155 do open set view to pladep do chekstop with select() 156 157 158 set color to clear do inptmps 159 160 *!** 161 *! 162 *! 163 *! This is the main procedure that will call to very Procedure: MAINMENU different activities (Display/Input or Exit), depending on the menu choice selected by the player. Calls: SELECT() (FOXBASE+ function) 164 165 *! (procedure in PLA4.PRG) (procedure in PLA4.PRG) * : CHEKSTOP 166 *! : INPTMPS 167 * : QUIT (procedure in PLA4.PRG) 168 *j 169

S 2.44

W.

144.5

· 12.24.

N.S. 8. 4.

· · · · · · · · · ·

-

1. 1. 1. 1. C. C.

256

en' to de winde the first tim

A 450 6 2.

The stand of the William in march of

170 171 procedure mainmenu 172 do while .t. 173 174 175 set color to clear m_menu=0 176 177 178 179 180 181 182 g 5,10,17,02 b0x
g 5,26 to 7,53 double set color to bu/n
g 6,34 say "PLANNING WORKAREA" set color to br/n @ 11,24 to 14,54 183 184 set color to bg/n @ 12,27 prompt "1.DISPLAY/INPUT OF PLANN." @ 13,27 prompt "2.EXIF. " 185 186 187 188 189 menu to m_menu do chekstop with select() do case 190 191 192 case m menu=1 do inptmps case m menu=2 do quit 192 193 194 195 196 set color to exit -endcase 197 enddo 198 199 +1++ This procedure calls the display/input procedure used to edit Procedure: INPTMPS the scheduling, based on the information obtained from orders accepted by the sales department. Called by: PLA4.PRG 200 201 202 203 *1 *į *i *j 204 205 *į : MAINMENU (procedure in PLA4.PRG) *1 206 +į Calls: PLA_BROW (procedure in PLA4.PRG) 207 208 209 *į procedure inptmps 210 211 212 213 214 215 216 217 218 219 220 221 *! 222 *! 223 *! clear select 5 goto top filtreso=resour_cod filtcode="ALL" lfiltreso="ALL" select 3 do pla brow return **** ********** This procedure displays the scheduling, as well as the sale Procedure: PLA_BROW information, allowing the edition of the former. It is based on the "browse" algorithm which Called by: INPTMPS (procedure in PLA4.PRG) is thoroughly explained in one of 224 225 226 227 228 229 230 231 *! ***** the thesis's sections. Calls: CHR() (FOXBASE+ function) : RECNO() : FILT_RESO : NOT.EOF() : SAM_DISP (FOXBASE+ function) (FOXBASE+ function) (procedure in PLA4.PRG) (FOXBASE+ function) (procedure in PLA4.PRG) (FOXBASE+ function) (procedure in PLA4.PRG) (function in PLA4.PRG) (procedure in PLA4.PRG) (FOXBASE+ function) (mocedure in PLA4.PRG) : EOFO : STA_DISP : SYSTEMO 232 233 234 235 236 236 237 238 *[*i *1 GETKEY : BOF() : CREANEW *1 : CREANEW (procedure in PLA4.PRG) : .AND..NOT.SYSTE(FOXBASE+ function) : POSITION (procedure in PLA4.PRG) : FILT_CODE (procedure in PLA4.PRG) * *į 239 *j 240 *Ì 241 242 243 244 245 246 247 248 249 250 251 252 253 254 *1 *!* ******* procedure pla_brow * Notes...: BROWSE program for PLANN.DBF private pancol, panmax, panlast, recnumtop, recnumlast, skiprecs private home, endkey, uparrow, downarrow, leftarrow, rightarrow private row, rowtop, rowbottom, rowprompt, keystrokes, pagepaint private low;how once, target, tarsed, soursed, keyst1, keyst2 private keyst3, press1, press2, lastdel * ---Initialize constants. home = chr(1)endkey = chr(6)

16 4. Sec. 1. Sec. 1. 1. 1.

S. S. S. S.

A. 1. 2 . 2 . 14. 14.

and the ship was a state

· . .

Sail and Angell Start Hat Sa

Start Contraction and

S. A. Store diff.

Level and the a last lines

255 uparrow = chr(5)256 257 258 259 260 261 262 downarrow = chr(24)leftarrow = chr(19)lettarrow = chr(19)
rightarrow = chr(4)
keystrokes = uparrow+downarrow+home+leftarrow+;
rightarrow+endkey+pgdn+pgup
keyst1 = "RNCS" + keystrokes + delrecord + returnkey
keyst2 = "R" + delrecord + returnkey
keyst3 = "PC" + keystrokes
contact = 1 263 264 265 rowtop = 1rowbottom = 20266 rowprompt = rowbottom + 3267 268 skiprecs = rowbottom - rowtop + 1 goto top * ---Initialize local variables. 269 270 271 272 273 273 274 275 276 press1 = " press2="P" row = rowtop recnum = recno() recnumtop = recnum pagepaint = .t. statpage = .f. isedited = .f. 277 278 279 280 281 282 lastdel=.f. pancol = 1 panlast = 1panmax = 2 * ---Perform BROWSE. set color to &screenatr 283 284 285 clear 286 do filt reso with .t. 287 288 * --- The following loop is really a "REPEAT/UNTIL <cond>". =do while .t. 289 if pagepaint f-if press2 <> press1.or.statpage 289 290 291 292 293 294 295 goto top recnumtop=recno() press1 = press2 endif recnum = recno() 296 -if .not.eof() 297 298 goto recnumtop ndif 299 300 301 302 if press2="P" select 3 do sam_disp with (rowtop), skiprecs goto top 303 -if eof() 304 305 empty = .t.else 305 306 307 308 309 310 311 -if empty empty =.f. -endif -endif else select 2 312 313 do sta_disp with (rowtop), skiprecs endif 314 315 -if .not.eof().or.select()=2 goto recnum 316 317 318 endif -if pancol = panlast * ---Reposition record pointer when repainting current page. 319 row = rowtop 320 321 322 323 324 325 326 327 328 endif panlast = pancol pagepaint = .f. set color to &promptatr @ rowprompt-1,0 say promptbar _____if press2="P" @ rowprompt,0 clear if empty @ rowprompt,0 say ; "NO RECORDS AVAILABLE: (R)esource (N)ew (S)tatus " 329 330 331 332 333 334 335 else -if system("plareport") @ rowprompt,0 say ; "BROWSE: (E)xit (N)ew (C)ode (R)esource (S)tatus <Arrows> <Return> " else 336 @ rowprompt,0 say ; "BROWSE: (N)ew (C)ode (R)esource (S)tatus <Arrows> <Return> " 337 338 endif endif 339

A STATISTICS AND A STATISTICS AND A STATISTICS AND THAT

St. 1. 1. 1.

a state way a set of the set of

1. 2. 2. 2. - Sin

258

and a strange to a state of a second strange of a strange of the second strange

340 341 342 343 344 345 346 346 347 348 349 350 else @ rowprompt,0 clear -if system("plareport") @ rowprompt,0 say ; "STATUS: (E)xit (P)lann (C)ode <Arrows> " else @ rowprompt,0 say ; "STATUS: (P)lann (C)ode <Arrows> " endif endif et color to bu+,n @ rowprompt+1,18 say "CURRENT SELECTION: CODE=" ?? filtcode 351 352 353 354 355 356 357 @ rowprompt+1,47 say " RESOURCE=" ?? filtreso set color to &promptatr @ row,0 say chr(16) if press2="P" 358 -if empty 359 do getkey with choice, "RNS" 360 else 361 do getkey with choice, keyst1 362 363 364 365 endif else do getkey with choice, keyst3 endif 366 * --- Reposition record pointer. do while choice \$ uparrow+downarrow @ row,0 say " " 367 368 369 370 371 372 373 374 375 -if choice = uparrow skip -1 do case case bof() goto top case row > rowtop row = row - 1376 377 378 otherwise recnumtop=recno() * ---Scroll window down. scroll rowtop,0,rowbottom,79,-1 379 380 381 do sam_disp with row,1 382 else 383 384 385 do sta_disp with row,1 endif endcase 386 else 387 skip 388 do case 389 390 391 392 case eof() goto bottom case row < rowbottom row = row + 1393 otherwise 394 * --- Adjust top-of-page record pointer. recnum=recno() 395 396 397 goto recnumtop skip 398 399 400 recnumtop = recno() 401 402 403 404 else 405 406 407 408 do sta_disp with row, 1 endif endcase endif 409 410 set color to &promptatr @ row,0 say chr(16) if press2="P" 411 412 413 414 415 415 416 417 do getkey with choice, keyst1 else do getkey with choice, keyst3 endif enddo ----Prompt line selections. 418 419 do case case choice = "N" 420 do creanew case choice = returnkey if ord_stat_1 = "NO_START".and..not.system("SYS_STOP") 421 422 423 424 @ rowprompt,0 clear set color to &promptatr

4. A. T.

120 . 15'4 1

Sec. We and

Vi Burning Star

and all all all a

4. 50m

Site Presalition was

The ist and some the ask of the second second

diet in the

@ rowprompt,0 say ; "SELECT : (E)dit/Re-edit (R)eplan " do getkey with choice, "ER" 425 426 427 428 do case 429 430 case choice = "E" store, Read and Replace the quantity of an existing order. store quantit to mquantit 431 @ row,c+26 get mquantit picture "@Z 999" valid mquantit <>0 432 433 434 435 read replace quantit with mquantit unlock do sam disp with row,1 case choice="R" 436 437 438 do position 439 440 441 442 443 444 445 444 445 444 445 450 451 452 453 454 455 456 pagepaint = .t. endcase endif =case choice = delrecord —if ord_stat_1="NO_START".and..not.system("SYS_STOP") * ---Delete the record. delete -Adjust the record position skip ----if eof() skip -1 row=row-1 r=r-1 lastdel=.t. else lastdel=.f. endif -if recno()=recnumtop 457 recnumtop = recno() 458 endif 459 460 461 462 463 recnum=recno() * ---Scroll window up. -if lastdel scroll row+1,0,rowbottom,79,1 else 464 scroll row,0,rowbottom,79,1 465 endif goto recnumtop skip (skiprecs-1) 466 467 468 469 470 471 472 473 474 475 476 477 do sam_disp with rowbottom,1 goto recnum endif case choice = pgdn _____if .not. eof() goto recnumtop skip skiprecs _____if eof() goto bottom endif 478 479 recnumtop = recno() pagepaint = .t. 480 endif case choice = pgup _____if .not. bof() 481 482 483 484 485 goto recnumtop skip -skiprecs -----if bof() goto top 486 487 -endif 488 recnumtop = recno() 489 490 491 492 493 pagepaint = .t. endif case choice = "C" lphrase = phrase lfiltcode = filtcode 494 495 ---Set FILTER for the Codes. do filt_code with .f. case choice = "R" 496 497 498 499 500 501 Iphrase = phrase Ifiltreso = filtreso * --Set FILTER for the Resources. ---Set FILTER for do filt_reso with .f. case choice = "S" press1=press2 press2=choice 502 503 504 pfilt=actfilt 505 pfiltcode=filtcode 506 507 508 pcodchoice=codchoice select 2 set filter to &actfilt 509 goto top

1. 1 Be y

in the said

1 41 - 1 Tes

No. of

1. 1. 2. 4. 2. L.

at the state of a loss of the second state of a second state with

260

The take the water stand the to the to the

510 511 512 513 514 515 if .not.eof() pagepaint = .t. else press1=press2 press2="P" select 3 516 517 518 519 520 521 522 523 524 525 526 527 528 529 endif case choice = "P" press1=press2 press2=choice filtcode=pfiltcode actfilt=pfilt codchoice=pcodchoice select 3 set filter to &actfilt pagepaint = .t. case choice = home pagepaint = (pancol <> 1) pancol = 1 case choice = endkey pagepaint = (pancol <> panmax) 530 531 pancol = panmax 532 533 534 535 endcase enddo set filter to goto top 536 return 537 538 *!**** ******* 538 *! 539 *! 540 *! 541 *! 542 *! 543 *! 544 *! 545 *! 546 *! 547 *! This procedure allows the input of the article code to be Procedure: NEWPLAN scheduled and calls the edition procedures if that input code exists, exiting the procedure otherwise. Called by: CREANEW (procedure in PLA4.PRG) (FOXBASE+ function) (procedure in PLA4.PRG) Calls: FOUND() : PL_STOR : PLEDIT (procedure in PLA4.PRG) ************ *** 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 procedure newplan **** mcode = " editing=.t. do while .t. @ r,c get mcode picture "@!NNNN" valid mcode <> " ".and.mcode\$phrase read select 1 find &mcode if found() store descript to mdescript store resour_cod to mresour_co select 3 do pl_stor 565 566 editing = .f.exit 567 568 569 570 571 572 573 574 575 576 577 578 L -endif else select 3 clear @ 8,33 say "INEXISTENT ARTICLE" wait exit= Ł endif enddo return *14 This procedure calls the procedures responsible for the Procedure: PLEDIT input of the schedule information and the replace of it in the <PLAN > datafile. 579 *1 580 *! 581 582 * * * * * * Called by: NEWPLAN (procedure in PLA4.PRG) 583 584 585 586 586 587 Is: PL_GETS (procedure in PLA4.PRG) ; GETKEY (procedure in PLA4.PRG) ; RETURNKEY() (FOXBASE+ function) ; INCLUDE (procedure in PLA4.PRG) ; DELRECORD() (FOXBASE+ function) Calls: PL GETS : GETKEY *! *1 588 *! 589 *1 590 591 592 593 *!* ******* procedure pledit choice = " " 594 do pl_gets

10.00

261

AD aver of start

Consistence of the Stand of the second standing of a construction

595 set color to &promptatr 596 597 @ rowprompt,0 clear @ rowprompt,0 say; "NEW ORDER: without saving (R)ewrite <Return> saving " 597 598 599 600 601 602 do include -else 603 -if choice=delrecord.and..not.system("sys_stop") 603 604 605 606 607 nonew = .t.endif -endif return 608 *!**** 609 This procedure allows the input of the schedule Procedure: PL_GETS information attached to each order. 610 *1 611 *) 612 *! 613 *! 614 *! 615 *! Called by: PLEDIT (procedure in PLA4.PRG) 616 procedure pl_gets 617 **** set color to n/w @ r,c+5 say mdescript @ r,c+26 get mquantit picture "@Z 999" valid mquantit < >0 618 619 620 621 622 623 624 625 read mmeedtim=mquantit*a->procestime @ r,c+35 say mneedtim picture "999.99" @ r,c+47 say mresour_co picture "@!NNNN" @ r,c+52 say mproduct_1 626 627 628 @ r,c+66 say mord_stat_1 return 629 This procedure calls the procedures that will replace Procedure: INCLUDE the input schedule information in the <PLAN > datafile. 630 *1 631 * . *! *! *! 632 633 634 635 Called by: PLEDIT (procedure in PLA4.PRG) *! Calls: CALORD (procedure in PLA4.PRG) (FOXBASE+ function) (procedure in PLA4.PRG) : RECNO() : PL_REPL 636 *j 637 * 638 *! 639 *!********* ****** 640 641 642 procedure include do calord 643 -if reccount()=0 644 recnumtop=recno() 645 -endif 646 647 648 append blank do pl_repl return 649 650 651 ***** *1 This procedure replaces the input schedule information 652 *! Procedure: PL_REPL in the <PLAN> datafile. 653 *! *! 654 655 656 Called by: INCLUDE (procedure in PLA4.PRG) * Calls: EOF() (FOXBASE+ function) 657 *į 658 * **************** procedure pl_repl 659 660 661 if .not. eof() * ----Replace only if there is an available record 662 663 664 665 replace; code with mcode,; descript with mdescript,; order_num with morder_num,; resour_cod with mresour_co,; 666 667 668 quantit with mquantit 669 replace; replace, need time with mneedtim,; ord_stat_1 with mord_stat_1,; ord_stat_0 with mord_stat_0;; product_1 with mproduct_1;; sequence with msequence 670 671 672 673 674 675 unĺock 676 endif 677 return 678 *1 679

Marrie 1. Contraction of the second

and a series of the

1. 1. and 2.4

Salar Salar

262

2.

in all

This procedure exits the Sales Department's program IT before it has been started. 680 *! 681 *! Procedure: QUIT 682 *! 683 *! Called by: MAINMENU (procedure in PLA4.PRG) 684 *! 685 *! *[** procedure quit 686 687 688 clear 689 close all 690 691 692 set color to set scoreboard on set bell on 693 set talk on 694 return 695 696 697 698 ***** ****** *Ì This procedure stores default values in the schedule Procedure: PL_STOR variables to be edit. *i 699 +į 700 701 702 703 704 705 706 707 708 709 710 711 *į Called by: NEWPLAN (procedure in PLA4.PRG) *) * *** procedure pl_stor to mquantit to mproduct_1 store 0 store 0 store 0 to mproduct 0 store 0 store "NO_START" store "NO_START" to mneedtim to mord_stat_1 to mord_stat_0 return 726 727 728 729 730 731 732 733 734 735 736 737 738 739 store sequence +1 to msequence return **************** *1 This procedure manages the possible conflicts that could *i Procedure: ERR_FIX arise from the fact that we are working in a multi-user environment. Its functioning is thoroughly explained in one Called by: PLA4.PRG of the thesi's sections. * * *** Calls: RLOCK() (FOXBASE+ function) *********************** procedure err_fix parameters errnum, mess 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 ** Error: File in use by another. -if errnum=108 save screen to screen set color to gr+/bu @ 7,17 clear to 15,61 @ 7,17 to 15,61 double set color to r*+/bu @ 11,24 say "Please wait to append a record." @ 12,26 say "Press any key to continue." read restore screen from screen retry endif ** ** Error: Record in use by another. -if errnum=109 save screen to screen time=0set color to gr+/bu @ 7,17 clear to 15,61 @ 7,17 to 15,61 double set color to r^{+} +/bu 763 764 @ 11,26 clear to 11,54

enter Mark.

Si Para Lat?

I was a second state of the state of the state of the state of the state of the state of the state of the state

A second a second

@ 11,26 say mess @ 12,26 say "Press any key to continue." 765 766 767 read 768 do while .not. rlock().and.time < 1000 769 770 771 772 773 774 time = time + 1enddo if time < 1000 restore screen from screen retry else set color to gr+/bu@ 7,17 clear to 15,61 @ 7,17 to 15,61 double set color to r^*+/bu 775 776 777 778 779 @ 11,19 say "Record cannot be locked.Try again later." @ 12,26 say "Press any key to continue." 780 781 782 783 784 read restore screen from screen endif endif 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 ** ** Error: Record is not locked. if errnum=130 save screen to screen time = 0do while .not. rlock().and.time < 1000 time=time+1 enddo if time < 1000 restore screen from screen retry else else set color to gr+/bu @ 7,17 clear to 15,61 @ 7,17 to 15,61 double set color to r*+/bu @ 11,19 say "Record cannot be locked.Try again later." @ 12,26 say "Press any key to continue." 800 801 802 803 read 804 restore screen from screen 805 endif 806 -endif 807 808 *1*** 809 *1 810 811 812 * Procedure: COUNTDEL *i * Calls: EOFO (FOXBASE+ function) 813 *! : DELETED() (FOXBASE+ function) 814 *1 815 *!* ***** procedure countdel 816 817 818 set delete off 819 store 0 to ctdel 820 goto top 821 do while .not. eof() 822 -if deleted() 823 ctdel=ctdel+1 824 825 -endif -if .not. eof() 826 skip 827 -endif 828 enddo 829 goto top 830 831 832 set delete on return 833 *!** 834 835 This procedure checks if the game manager has started, Procedure: CHEKSTOP frozen, or quit the game. It is the procedure th * * frozen, or quit the game. It is the procedure that checks For the status of the running gaming simulation and forces the PRG planning program to do so. Its functioning is thoroughly explained ENU (procedure in PLA4.PRG) in one of the thesis's sections. 836 *! 837 *! Called by: PLA4.PRG *! 838 : MAINMENU 839 840 *! : GETKEY (procedure in PLA4.PRG) *! 841 842 *! Calls: SYSTEM() ls: SYSTEM() (function in PLA4.PRG) : .AND..NOT.SYSTE(FOXBASE + function) : PLANREPO (procedure in PLA4.PRG) *! 843 *! 844 *! : PRT (procedure in PLA4.PRG) : PKT (proceaure in FLAT.FAG) : .AND..NOT.SHUT.(FOXBASE+ function) : .NOT.SYSTEM() (FOXBASE+ function) : .AND.SYSTEM() (FOXBASE+ function) : STR() (FOXBASE+ function) +1 845 846 *! 847 *! 848 *! *1

C. Buch cr. NY BURNER The best of the state of the

849

She by . To want

850 *! Uses: SYSTEM.DBF 851 +! 853 procedure chekstop 854 855 parameters aselect 856 private shut shut=.f. 857 858 859 signal=.f. -do while .t. 860 861 862 =do case case system("sys_stop").and..not.system("gam_stop") 863 -if system("plareport") 864 do planrepo 865 do prt select 10 866 867 868 869 replace plareport with .f. unlock start=.t. 870 871 872 endif if .not.signal save screen to screen1 873 874 875 set color to clear set color to gr+/bu @ 5,14 clear to 14,64 @ 5,14 to 14,64 double 876 877 @ 9,17 say "THE GAME HAS BEEN FROZEN OR A DAY HAS ENDED." @ 11,30 say "WAIT UNTIL RESTART." 878 879 880 sys_start=.f. 881 signal=.t. endif 882 883 884 if system("restart").and .. not.shut.and .. not.system("gam_stop") 885 close databases 886 select 10 887 use system 888 889 shut = .t.else 890 -if .not.system("restart").and.shut 891 set view to pladep 892 shut = .f.893 894 895 -endif endif else set color to gr+*/bu @ 24,70 say "STOPPED" 896 897 898 set color to 899 exit 900 901 L -endif case .not.system("gam_stop").and..not.system("sys_stop") 902 -if signal.and..not.start.and..not.pause 903 restore screen from screen1 904 905 endif -if start 906 907 908 909 daystat =.f. endif start = .f.sys_start=.f. 910 911 set color to @ 24,70 clear to 24,76 912 =exit 913 914 915 916 917 918 919 920 case system("gam_stop").and.system("sys_stop") -if sys_start _____if .not.signal1 set color to clear 921 922 923 924 925 signal1 = .t. start=.t. endif 926 -else 927 928 close all clear all 929 set color to 930 931 932 clear set color to bu +/n @ 8,24 to 12,54 double 933 set color to gr+/n @ 10,29 say "THE GAME HAS FINISHED" 934

11

dere and the second

1. 135

all the allow

2.99 . 9.00 16. 19 . 18

The state of the state of the internet by it in Bart is and they are not and the self of the

" A Advant Constant on Level ASC A Coll of State B or Second March 19

and a second second states of the states and the second second second states and the second second second second

025	E	L states and the
935 936		set console off
937	<	cancel
938		Lendif
939		enddo
941		sys_start=.f.
942		aselect = str(aselect)
943		select acaselect
945		水 中
946	*!*	***************************************
947	*1	Procedure: GETKEY done by the user
949	*j	
950	*!	Called by: PLA BROW (procedure in PLA4.PRG)
951	*!	: PLA CODE (procedure in PLA4.PRG)
953	ŧ!	: PLA_RESO (procedure in PLA4, PRG)
954	*!	: POSITION (procedure in PLA4.PRG)
955	*!	Calis: SYSTEM() (function in PLA4.PRG)
957	*!	: .NOT.SYSTEM() (FOXBASE+ function)
958	*!	: CHR() (FOXBASE + function)
959	*1	: UPPER() (FOXBASE+"function)
961	*1	: .AND.SYSTEM() (FOXBASE+ function)
962	*!	: SELECT() (FOXBASE + function)
964	*!	: CHERSTOP (procedure in FLAT. FRO)
965	*!	Uses: STATUS.DBF
966	*1	******
968	•	procedure getkey
969		*****
970		parameter choice, key chars
972		scrr=.t.
973		choice = "*"
974		
976		select 2
977		use
979		replace restatus with .f.
980		unlock
981		do while .not.system("restatus")
983		set typeahead to 0
984		save screen to screen2
985		set color to $bu+/n$ a 11.25 to 13.55 double
987		set color to gr+*/n
988		@ 12,29 say "calculating status file"
990		scrr=.1.
991		enddo
992		set typeahead to 20
993		set color to & promptatr
995		@ row,0 say chr(16)
996		select 2
998		select 10
999		replace restatus with .f.
1000		unlock
1001		select 5
1003		keycode = inkey()
1004		if .not.system("sys_stop").andnot.daystat
1005		$\begin{array}{c} \text{daystal} = .t.\\ \text{nagenaint} = .t. \end{array}$
1007		statpage=.t.
1008		
1009		select 5
1011		select 2
1012		
1013		set mier to ocacimit /exit
1015		L-endif
1016		if press2="P".and.system("STATRESO")=filtreso.andnot.editing
1017		pagepaint=.t.
1019		statpage =. t.

1020 exit 1021 L-endif 1022 -endif 1023 if keycode > 0choice = upper(chr(keycode)) -if choice \$ "E".and.system("plareport") pause = .f. 1024 1025 1026 alse 1028 -if system("FACTTIME") < > "8:00" 1029 pause = .t.1030 endif endif 1031 1032 1033 do chekstop with select() endif 1034 inddo 1035 return 1036 1037 *!* This procedure displays on the screen the scheduled 1038 *! 1038 *! 1039 *! 1040 *! 1041 *! 1042 *! 1043 *! 1044 *! 1045 *! 1046 *! 1047 *! 1048 *! Procedure: SAM_DISP orders. Called by: PLA_BROW : POSITION (procedure in PLA4.PRG) (procedure in PLA4.PRG) (procedure in PLA4.PRG) : CREANEW (FOXBASE+ function) Calls: ROW() (FOXBASE+ function) : COLO 1049 procedure sam_disp 1050 1051 parameter row, listrees 1052 1053 1054 if listrecs > 1 * ---Display heading when listing the entire page. set color to &statusatr 1055 @ rowtop-1,1 1056 =do case coase pancol = 1
?? "CODE DESCRIPT------ QUANTIT NEED_TIME RESOUR_COD PRODUCT_1 ORD_STAT_1" 1057 1058 1059 endcase * --- Clear the window area. 1060 1061 set color to &windowatr @ rowtop,0 clear to rowbottom,79 1062 1063 -endif * --- Display the records. 1064 set heading off set color to &windowatr 1065 1066 1067 1068 @ row-1,0 say ' =do case 1069 case pancol = 1 1070 list next listrees code, descript, quantit, ", need_time," ", resour_cod, product_1," ", ord_stat_1," " off 1071 endcase 1072 set heading on 1073 1074 1075 -if listrecs > 1 r=row() c = col0+11076 -if statpage 1077 select 10 replace statreso with "--1078 1079 replace statcode with "----" 1080 unlock 1081 select 3 1082 statpage = .f. 1083 endif 1084 endif 1085 return 1086 1087 *!****** ***** This procedure displays the sales orders to be Procedure: STA_DISP delivered. 1088 *! 1089 *1 1090 * 1091 *! Called by: PLA_BROW (procedure in PLA4.PRG) 1092 *! 1093 *! Calls: ROW() (FOXBASE+ function) 1094 *! : COLO (FOXBASE+ function) 1095 *! 1096 *!* ******* ****** 1097 procedure sta_disp 1098 1099 parameter row, listrecs if listrecs > 1 1100 * --- Display heading when listing the entire page. set color to &statusatr 1101 1102 1103 @ rowtop-1,1 1104 =do case

The art is a we have

......

267

Comments in 18

and a state a liter in a state of the state of states of

1105 case pancol = 1 ?? "CODE OUANTIT STOCK DATE RESOURCE CAPACITY UNTIME SETUP TOTALTIME" 1106 1107 endcase 1108 * ----Clear the window area. 1109 set color to &windowatr @ rowtop,0 clear to rowbottom,79 1110 1111 1112 endif * -Display the records. 1113 set heading off 1114 1115 set color to g+/n,n/w @ row-1,0 say 1116 do case 1117 case pancol = 1 1118 1119 list next listrees code," ",quantit,stock," ",date," ",resource," ",capacity," ",time,setup," ",tottime off endcase 1120 set heading on 1121 1122 r = row()c = col() + 11123 1124 1125 1126 return *1 This procedure calculates all the posible article codes Procedure: PLA_AREA0 to filter the <PLAN> and <STATUS> datafiles by. 1127 1128 *1 1129 *1 1130 *1 1131 *1 1132 *! Called by: FILT_CODE (procedure in PLA4.PRG) Calis: STR() : EOF() : PLA_CODE (FOXBASE+ function) (FOXBASE+ function) (procedure in PLA4.PRG) 1133 *į *1 1134 *1* 1135 ****** procedure pla_area0 1136 1137 1138 1139 parameter base private i,ii 1140 select 1 1141 goto top i=0 1142 ii=str(0,1) code&ii="ALL" 1143 1144 1145 1146 i=i+1ii=str(i,1) do while .not. eof() 1147 1148 -if code < > " public code&ii 1149 1150 1151 1152 1153 1154 code&ii=code endif -if .not. eof() i=i+1ii=str(i,1) skip endif 1155 1155 1156 1157 1158 1159 1160 1161 =enddo base=str(base) select &base do pla_code with i return 1162 1163 ******* 1164 *! 1165 *! This procedure calculates all the possible resource codes Procedure: PLA_AREA1 to filter the <PLAN> and <STATUS> datafiles by. * 1166 *į Called by: FILT_RESO 1167 (procedure in PLA4.PRG) +j 1168 (FOXBASE+ function) (FOXBASE+ function) 1169 *j Calls: STR() 1170 *į : EOF() *1 1171 : PLA RESO (procedure in PLA4.PRG) *1 1172 1173 1174 1175 procedure pla_area1 1176 parameter base1 1177 private i,ii 1178 select 5 goto top i=0 1179 1180 1181 ii=str(i,1) 1182 1183 =do while .not. eof() -if resour_cod < > public reso&ii 1184 1185 reso&ii=resour_cod 1186 endif 1187 if .not. eof() 1188 i=i+1ii = str(i, 1)1189

1190 1191 1192 1193 skip endif enddo base1 = str(base1) 1194 1195 select &base1 do pla_reso with i 1196 1197 1198 return 1199 *! 1200 *! This procedure calculates the filter of the <PLAN> and <STATUS> Procedure: PLA_CODE datafiles based on the selected article code. 1201 ÷į + 1202 Called by: PLA AREA0 (procedure in PLA4.PRG) *! *! *! 1203 Calls: STR() (FOXBASE+ function) 1204 1205 1206 1207 : GETKEY (procedure in PLA4.PRG) 1208 1209 1210 procedure pla_code parameter numbcode 1210 1211 1212 1213 1214 private n,nn,codekey,sayphrase n=0sayphrase = "" codekey = "" 1215 nn = str(n, 1)1216 set color to &promptatr 1217 1217 1218 1219 1220 do while n < numbcode sayphrase=sayphrase+" "+nn+"-"+code&nn codekey=codekey+nn n=n+1nn = str(n, 1)1221 1222 1223 enddo @ rowprompt-1,0 clear @ rowprompt-1,0 say promptbar
 @ rowprompt,0 clear
 @ rowprompt,0 say "SET CODE:"+sayphrase
 do getkey with codchoice,codekey+returnkey 1224 1225 1226 1227 1228 =do case 1229 =case codchoice = returnkey 1230 case codchoice \$ codekey 1230 1231 1232 1233 1234 filtcode=code&codchoice endcase return 1235 1236 ***** ****** This procedure calls the procedure to calculate the filter Procedure: PLA_RESO of the <PLAN> and <STATUS> datafiles based on the selected 1237 *1 1238 1239 1240 *1 *i resource code. *j Called by: PLA_AREA1 (procedure in PLA4.PRG) 1240 1241 1242 *İ (FOXBASE+ function) (procedure in PLA4.PRG) (procedure in PLA4.PRG) *į Calls: STR() 1243 1244 *1 : GETKEY *1 : RESOURCE 1245 * 1246 1247 ***** ****** procedure pla reso 1248 1249 parameter numbreso private n,nn,codekey,sayphrase n=0 1250 1251 1252 sayphrase = "" codekey="" 1253 1254 1255 nn = str(n, 1)nn=sr(n, 1) set color to &promptatr do while n < numbreso sayphrase=sayphrase + " "+nn+"-"+reso&nn codekey=codekey + nn 1256 1250 1257 1258 1259 n=n+11260 nn = str(n, 1)1261 enddo @ rowprompt-1,0 clear 1262 @ rowprompt-1,0 star
 @ rowprompt-1,0 clear
 @ rowprompt,0 clear
 @ rowprompt,0 say "SET RESOURCE:"+sayphrase
 do getkey with reschoice,codekey+returnkey
 if reschoice \$ codekey 1263 1264 1265 1266 1267 1268 filtreso=reso&reschoice 1269 do resource 1270 1271 endif return 1272 1273 1274 *1 This procedure calculates the filter of the <PLAN> and <STATUS>

Procedure: RESOURCE datafiles based on the selected resource code. 1275 *! *! *! 1276 Called by: PLA_RESO : FILT_RESO (procedure in PLA4.PRG) 1277 1278 1279 1280 (procedure in PLA4.PRG) ÷į (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) *į Calls: .NOT.EOF() : STR() : EOF() 1281 *į 1282 *j 1283 * 1284 *14 ****** ***** 1285 procedure resource 1286 public field 1288 m=0phras="" 1289 1290 phrase = "" 1291 select 1 1292 1293 1294 goto top do while .not.eof() -if resour_cod=filtreso mm=str(m,1) 1295 1296 1297 public field field=code 1298 phras="&field" 1299 m=m+1 1300 1301 -if m>1 phrase = phrase + phras 1302 else 1303 phrase=phras 1304 endif 1305 endif 1306 1307 1308 -if .not. eof() skip endif 1309 enddo 1310 select 3 1311 return 1312 1313 *!' 1314 *! 1315 *! *1** This procedure filters the <PLAN> and <STATUS> datafiles Procedure: FILT_CODE based on the selected article code. 1316 * * * * * 1317 Called by: PLA_BROW (procedure in PLA4.PRG) 1318 Calls: SELECT() 1319 (FOXBASE+ function) 1320 1321 1322 *i (procedure in PLA4.PRG) (FOXBASE+ function) (FOXBASE+ function) : PLA_AREA0 *1 : EOFO *į : RECNO() 1323 *i 1324 +11 procedure filt code 1325 1326 1327 1328 1329 parameter start -if .not.start do pla_area0 with select() 1330 endif 1331 -if codchoice = "0" 1332 actfilt="Code <> FiltCode.AND.Code\$phrase" 1333 else actfilt="Code=FiltCode.AND.Code\$phrase" 1334 1335 1336 1337 endif set filter to &actfilt goto top 1338 if eof() 1339 set color to &promptatr 1340 @ rowprompt,0 clear 1341 @ rowprompt,0 say "No matching records." 1341 1342 1343 1344 1345 wait @ rowprompt,0 clear phrase = lphrase filtcode = lfiltcode -if filtcode="ALL' codchoice="0" 1346 1347 1348 actfilt="Code <> FiltCode.AND.Code\$phrase" 1349 endif 1349 1350 1351 1352 1353 1354 set filter to &actfilt goto top endif recnumtop=recno() pagepaint=.t. 1355 set color to &screenatr 1356 -if press2 = "S" 1357 select 2 1358 endif 1359 clear

- 14 4 6 F.V

1360 return 1361 1362 1363 *!* This procedure filters the <PLAN> and <STATUS> datafiles Procedure: FILT_RESO based on the selected resource code. +i 1364 *! 1365 *! Called by: PLA_BROW 1366 *! (procedure in PLA4.PRG) +1 1367 1368 Calls: SELECT() : PLA_AREA1 : RESOURCE (FOXBASE+ function) (procedure in PLA4.PRG) (procedure in PLA4.PRG) *į 1369 1370 +į *1 +į : RECNOO (FOXBASE+ function) 1371 1372 *1 1373 . 1374 1375 1376 1377 procedure filt_reso parameter start mmstart=start 1378 if .not.start do pla_area1 with select() 1379 1380 else 1381 1382 do resource endif 1383 -if codchoice = "0" 1384 actfilt="Code <> FiltCode.AND.code\$phrase" 1385 else 1386 actfilt="Code=FiltCode.AND.code\$phrase" 1387 endif 1388 set filter to &actfilt 1389 goto top 1390 recnumtop=recno() pagepaint = .t. 1391 1392 set color to &screenatr 1393 clear 1394 return 1395 1396 This procedure allows the player to define the source Procedure: POSITION and target positions of the order to be re-sequenced. 1397 *] 1398 *! 1399 +į 1400 1401 1402 1403 1404 *! Called by: PLA_BROW (procedure in PLA4.PRG) *! (FOXBASE+ function) (FOXBASE+ function) (procedure in PLA4.PRG) (FOXBASE+ function) Calls: CHR() : RECNO() *į GETKEÝ 1405 *į BOF() 1406 *1 SAM DISP (procedure in PLA4.PRG) 1407 *1 : EOFO (FOXBASE+ function) 1408 1409 1410 1411 1412 * : RLOČK() (FOXBASE+ function) *1 : REPLANN (procedure in PLA4.PRG) +1 procedure position 1413 1414 rsour=row 1415 @ row,0 say " " 1416 set color to &promptatr 1417 1418 1419 @ row,0 say chr(17) source = recno() sourseq = sequence @ rowprompt,0 clear @ rowprompt,0 say ; "SELECT SEQUENCE : < Arrows> < Return> " 1420 1421 1422 1423 do while .t. 1424 1425 do getkey with choice, uparrow + downarrow + returnkey do while choice \$ uparrow + downarrow if row <> rsour 1426 1427 @ row,0 say 1428 else 1429 @ rsour,0 say chr(17) 1430 endif 1431 if choice=uparrow 1432 skip -1 1433 1434 do case ≈case bof() 1435 goto top case row > rowtop 1436 1437 row = row - 11438 otherwise 1439 1440 1441 1442 recnumtop = recno() * --- Scroll window down. scroll rowtop,0,rowbottom,79,-1 do sam_disp with row,1 1443 endcase else

1445 skip 1446 do case 1447 1448 1449 1450 case eof() goto bottom case row < rowbottom row = row + 1 1451 otherwise 1452 * --- Adjust top-of-page record pointer. 1452 1453 1454 1455 1455 1456 1457 1458 1459 recnum=recno() goto recnumtop skip recnumtop = recno() goto recnum ---Scroll window up. scroll rowtop,0,rowbottom,79,1 1459 1460 1461 1462 1463 1464 do sam disp with row,1 endcase -endif set color to &promptatr @ row,0 say chr(16) 1464 1465 1466 1467 1468 1469 1470 1471 do getkey with choice, uparrow + downarrow + returnkey enddo target=recno() tarseq=sequence -if tarseq < sourseq 1472 1473 position = .t. && UP else 1474 1475 1475 1476 1477 1478 && DOWN position=.f. endif do replann endif exit 1479 L... -endif 1480 @ rowprompt,0 clear @ rowprompt,0 say ; "IT IS NOT POSSIBLE TO RESEQUENCE THAT" 1481 1482 1483 wait 1484 1485 Walt @ rowprompt,0 clear @ rowprompt,0 say ; "SELECT SEQUENCE : <Arrows> <Return> " 1486 1487 enddo 1488 return 1489 1490 *1** This procedure re-schedules (re-sequence or re-edit) Procedure: REPLANN the selected order. 1491 *! 1492 *į 1493 *! 1494 1495 *| *| *| Called by: POSITION (procedure in PLA4.PRG) Calls: SEQUENCE>TARSEQ(FOXBASE+ function) : <>SOURCE() (FOXBASE+ function) : SEQUENCE<TARSEQ(FOXBASE+ function) : <>SOURCE() (FOXBASE+ function) : EOF() (FOXBASE+ function) : RECNO() (FOXBASE+ function) 1496 1497 + 1498 *į 1499 *1 1500 *1 1501 * 1502 1503 1504 *1 * procedure replann 1505 1506 -if target < > source 1507 1508 goto source 1509 replace sequence with tarseq 1509 1510 1511 1512 1513 goto target replace sequence with (sequence + 1) -else replace sequence with (tarseq-1) -endif 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 unlock if position filt="SEQUENCE>tarseq.AND.RECNO() <>source.AND.SEQUENCE<sourseq.AND.RECNO() <>target" else filt="SEQUENCE<tarseq.AND.RECNO() <> source.AND.SEQUENCE>sourseq.AND.RECNO() <> target" endif count for &filt to rec if rec < >0 dimension rr(rec) goto top 1526 1527 n=1 =do while .not. eof() 1528 -if &filt 1529 rr(n) = recno()

16 19 12 20 20

1530 n=n+11530 1531 1532 endif if .not. eof() 1533 skip endif 1534 1535 1536 enddo n=11537 do while n < = rec1538 goto rr(n) 1539 -if position 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1555 1556 1557 1558 replace sequence with (sequence + 1) else replace sequence with (sequence-1) endif unlock n=n+1enddo endif endif goto top recnumtop = recno() return **** This procedure creates the "view" used during all the * * * * * Procedure: OPEN planning program. Called by: PLA4.PRG Uses: BILL.DBF : STATUS.DBF : PLANN.DBF : STOCK.DBF : MACHINE.DBF 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 *j : PLAPAST.DBF : STKPAST.DBF : MACHPAST.DBF *į **** : SETUP.DBF Indexes: BILLNA.IDX : ORPLANN.IDX : CODESTK.IDX : RESOUR.IDX *! * 1572 1573 1574 1575 1576 1577 : CDPLA.IDX : CDSTK.IDX *! * * * * : CDMACH.IDX : SETUP.IDX *i 1578 1579 1580 1581 1582 1583 1584 1584 1585 1586 * procedure open set exclusive off select 1 use bill index billna select 2 use status select 3 1587 1588 use plann index orplann select 4 1589 use stock index codestk 1590 1591 1592 1593 select 5 use machine index resour select 6 use plapast index cdpla select 7 1595 1594 1595 use stkpast index cdstk select 8 1596 1597 use machpast index cdmach 1598 1599 1600 select 9 use setup index setup select 2 set safety off create view pladep from environment all 1601 1602 1603 set safety on 1604 close databases return ** 1605 1606 1607 This procedure calls the report procedures for Procedure: PLANREPO <PLAN>, <STOCK>, and <MACHINE> datafiles. 1608 *į 1609 *j 1610 *į 1611 *! Called by: CHEKSTOP (procedure in PLA4.PRG) 1612 *! 1613 *! : PRT (procedure in PLA4.PRG) *į 1614 Calls: PLAREPO (procedure in PLA4.PRG)

273

Sugar, Arth 1

the tet steam br the

1615 *! 1616 1617 procedure planrepo 1618 1619 do plarepo with 6 1620 1621 do plarepo with 7 do plarepo with 8 1622 signal = .f. 1623 1624 return This procedure manages all the report generations Procedure: PLAREPO for the different datafiles to be report. 1625 * 1626 *j *1 1627 1628 *! 1629 *! 1630 *! Called by: PLANREPO (procedure in PLA4.PRG) Calls: STR() : VAL() : EOF() (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) 1631 *! 1632 +j (FOXBASE+ function) (FOXBASE+ function) 1633 *į INTO *Ì SUBSTR() 1634 : DTOC() : PLAAVE : STKAVE : MACHAVE : PLAPAST : STKPAST 1635 *! 1636 *! (FOXBASE+ function) (procedure in PLA4.PRG) (procedure in PLA4.PRG) (procedure in PLA4.PRG) (procedure in PLA4.PRG) (procedure in PLA4.PRG) (procedure in PLA4.PRG) *! 1637 1638 *j 1639 *į 1640 *1 1641 *1 : MACHPAST 1642 *1 1643 1644 1645 1646 +1+ ******** procedure plarepo parameter pastbase 1647 set color to 1648 =do case 1649 1650 1651 1652 case pastbase=6 store 0 to n,mave inc,mper_inc,mave_star,mper_star,mave_fin store 0 to mper_fin,mave_prod,mave_wip,mave_batch,mave_setn store 0 to mave_lead,mave_cost,per_inc,per_star,per_fin,ave_star;; 1653 mave tot.ave tot 1654 1655 case pastbase = 7store 0 to n, mave_stk 1655 1656 1657 1658 1659 case pastbase=8 store 0 to n, mave_cap, mave_prod, mave_prot, mave_sett, mave_setn,; per_sett,mper_sett endcase 1660 1661 pastbase = str(pastbase, 2)1662 select &pastbase 1663 pastbase = val(pastbase) 1664 1665 1666 1667 goto top =do while .not. eof() -do case =case pastbase=6.or.pastbase=7 1668 actcode=code 1669 mdescript=descript fieldcod="CODE" 1670 1671 case pastbase = 8 1672 actcode=resour_cod 1673 1674 1675 mdescript=descript fieldcod="RESOUR_COD" endcase 1676 do while &fieldcod=actcode actweek = int(val(substr(dtoc(date), 1,2))/5) = do while int(val(substr(dtoc(date), 1,2))/5) = actweek.and.&fieldcod = actcode 1677 1678 1679 n=n+11680 do case 1681 1682 1683 case pastbase=6 do plaave case pastbase=7 1684 do stkave 1685 case pastbase = 8 1686 do machave 1687 endcase 1688 if .not. eof() 1689 skip 1690 else 1691 =exit 1692 ι -endif 1693 enddo 1694 skip -1 1695 set color to 1696 clear 1697 =do case case pastbase=6 do plapast 1698 1699

27 . 3

1700 1701 1702 1703 case pastbase = 7 do stkpast case pastbase = 8 do machpast 1704 endcase 1705 1706 skip do case case pastbase=6 1707 1708 1709 1710 store 0 to n, mave inc, mper_inc, mave_star, mper_star, mave_fin store 0 to mper_fin, mave_prod, mave_wip, mave_batch, mave_setn store 0 to mave_lead, mave_cost, per_inc, per_star, per_fin, ave_star;; 1711 1712 mave tot, ave tot case pastbase=7 1713 store 0 to n, mave_stk 1715 1714 1715 1716 1717 case pastbase=8 store 0 to n,mave_cap,mave_prod,mave_prot,mave_sett,mave_setn,; per_sett,mper_sett endcase 1718 1719 enddo enddo return 1720 1721 1722 1723 *1 This procedure calculates the averages of the 1724 *į <PLAN> datafile's key fields. Procedure: PLAAVE 1725 *į 1726 * Called by: PLAREPO (procedure in PLA4.PRG) 1727 ** *!** ***** 1728 1729 procedure plaave 1730 mave_inc = (mave_inc + incomplete)/n mave_star = (mave_star + started)/n mave_fin = (mave_fin + finished)/n 1731 1732 1733 mave_fin=(mave_fin+finished)/n mave_prod=(mave_prod+production)/n mave_batch=(mave_wip+wip)/n mave_batch=(mave_batch+ave_batch)/n mave_batch=(mave_setn+setupnum)/n mave_lead=(mave_lead+leadtime)/n mave_cost=(mave_cost+stand_cost)/n mave_tot=(mave_inc+mave_star+mave_fin) ave_tot=(incomplete+started+finished) per_inc=iif(ave_tot=0,0,(incomplete/ave_tot)*100) per_star=iif(ave_tot=0,0,(started/ave_tot)*100) mper_inc=iif(mave_tot=0,0,(mave_inc/mave_tot)*100) mper_star=iif(mave_tot=0,0,(mave_star/mave_tot)*100) mper_fin=iif(mave_tot=0,0,(mave_fin/mave_tot)*100) mper_fin=iif(mave_tot=0,0,(mave_fin/mave_tot)*100) mper_fin=iif(mave_tot=0,0,(mave_fin/mave_tot)*100) 1734 1735 1736 1737 1738 1739 1740 1741 1741 1742 1743 1744 1745 1746 1747 1748 return 1749 1750 1751 This procedure calculates the average of the <MACHINE> Procedure: MACHAVE datafile's key fields. *į 1752 *i 1753 *į 1754 * Called by: PLAREPO (procedure in PLA4.PRG) 1755 * 1756 1757 1758 1759 ******* ** procedure machave mave_cap =: (mave_cap + capacity)/n
mave_prod =: (mave_prod + production)/n
mave_prot =: (mave_prot + productiv)/n
mave_sett =: (mave_sett + setuptime)/n
per_sett =: (setuptime/capacity)*100 && the capacity should be per day.
mper_sett =: (mave_sett/capacity)*100 && the capacity should be per week. 1760 1761 1762 1763 1764 1765 1766 mave_setn=(mave_setn+setupnum)/n return 1767 1768 *!*** ************ 1769 * This procedure calculates the average of the <STOCK> 1770 * Procedure: STKAVE datafile's key fields. *|
*| 1771 1772 Called by: PLAREPO (procedure in PLA4.PRG) *j 1773 1774 *!* ******* 1775 procedure stkave 1776 1777 $mave_stk = (mave_stk + stock)/n$ 1778 return 1779 1780 ***** *j 1781 1782 This procedure allows the user to select the report's * Procedure: PRT output device (screen or printer). *İ 1783 1784 *! Called by: CHEKSTOP (procedure in PLA4.PRG)

1785	*!	
1786	*!	Calls: SYS() (FOXBASE+ function)
1788	*1	, I LAINER O (DIOCOMIC IN I LAN. NO)
1789	*!*	iseenedum of
1791		#***************
1792	1	
1793		clear
1795		m_menu=0
1797		set color to $bu + /n$ @ 4.18 to 17.61
1798		@ 6,24 to 8,54 double
1799		set color to br +/n 7.29 say "PRINT OUT OF REPORTS"
1801		set color to $bu + /n$
1802		$(2, 10, 24, 14, 50 \text{ box } \ _{\Pi^{$
1804		@ 11,29 prompt "1.TO PRINTER."
1805		@ 12,29 prompt "2.PAST BEHAVIOUR."
1807		$(g_{1})_{3,2}$ prompt "3.8A11. set color to bu +/n
1808		@ 16,29 say "DEVICE :"
1809		-?? sys(101) menu to m menu
1811		do case
1812		==case m menu=1
1814		=case m_menu=2
1815		do planrepo
1817		set color to
1818	v	exit
1820		set device to screen
1821		set color to
1822	v	mendcase exit
1824		enddo
1825		return *
1827	*!*	***********
10.07		
1828	*! *1	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance
1828 1829 1830	*! *! *!	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance.
1828 1829 1830 1831 1832	*****	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG)
1828 1829 1830 1831 1832 1833	*****	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function)
1828 1829 1830 1831 1832 1833 1834 1835	+++++++++++++++++++++++++++++++++++++++	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function)
1828 1829 1830 1831 1832 1833 1834 1835 1836	+ + + + + + + + + + + + + + + + + + + +	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function)
1828 1829 1830 1831 1832 1833 1834 1835 1836 1837	+ + + + + + + + + + + + + + + + + + + +	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast
1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839	* * * * * * * *	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast set color to r/n @ 4.23 say "
1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840	++++++++	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast set color to r/n @ 4,23 say " PLANNING DEPARTMENT'S PAST "
1828 1829 1830 1831 1832 1833 1834 1835 1836 1835 1836 1837 1838 1839 1840 1841 1842	********	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast set color to r/n @ 4,23 say " @ 5,23 say " PLANNING DEPARTMENT'S PAST " @ 6,23 say " PERFORMANCE (Page 1) "
1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843	******	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast set color to r/n @ 4,23 say " @ 5,23 say " @ 5,23 say " PLANNING DEPARTMENT'S PAST " @ 6,23 say " PERFORMANCE (Page 1) " @ 10, 0 say "Week"
1828 1829 1830 1831 1832 1833 1834 1835 1835 1835 1837 1838 1839 1840 1841 1842 1843 1844 1845	*****	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast ************************************
1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1843 1844 1843 1844	*****	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast ************************************
1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1846 1848	*****	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast ************************************
1828 1829 1830 1831 1833 1834 1835 1836 1837 1838 1836 1838 1839 1840 1843 1844 1843 1844 1845 1846 1848 1848 1848 1848 1848 1846 1848 1848 1848 1846 1848 1848 1848 1846 1848 1848 1846 1848 1848 1846 1848 1848 1846 1848 1846 1848 1846 1848 1846	*****	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast ************************************
1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1837 1838 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850	*******	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast ************************************
1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1840 1841 1842 1844 1845 1846 1847 1848 1849 1850	******	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast ************************************
1828 1829 1830 1831 1831 1832 1833 1834 1835 1836 1837 1838 1840 1841 1842 1843 1844 1845 1846 1847 1848 1847 1848 1849 1851 1851 1851 1851 1851 1851 1851 185	*****	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast ************************************
1828 1829 1830 1831 1831 1833 1834 1835 1836 1837 1838 1837 1838 1839 1840 1841 1842 1843 1844 1845 1847 1848 1847 1852 1851 1854 1855	*****	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast set color to r/n @ 4,23 say " PLANNING DEPARTMENT'S PAST " @ 6,23 say " @ 7,23 say " @ 10,0 say "Week" @ 10,0 say "Week" @ 10,15 say actweek picture "99" @ 10,20 say "Date" @ 10,30 say date @ 12,13 say actcode @ 12,20 say "Descript" @ 15,25 say "DAY" @ 15,55 say "DAY" @ 15,72 say "OVERALL" @ 15,72 say "OVERALL" @ 15,72 say "OVERALL" @ 15,72 say "OVERALL" @ 15,72 say "OVERALL" @ 15,72 say "Num. of incomplete orders"
1828 1829 1830 1831 1831 1833 1834 1835 1836 1837 1838 1837 1838 1839 1840 1841 1842 1843 1844 1845 1847 1848 1847 1852 1853 1854 1855 1854	*****	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast set color to r/n @ 4,23 say " PLANNING DEPARTMENT'S PAST " @ 6,23 say " @ 10,0 say "Week" @ 10,0 say "Week" @ 10,15 say actweek picture "99" @ 10,20 say "Descript" @ 12,20 say "Code" @ 12,20 say "Code" @ 12,20 say "Descript" @ 15,27 say "DYERALL" @ 15,55 say "DYERALL" @ 15,65 say "DAY" @ 15,75 say "Num. of incomplete orders" @ 16,27 say incomplete picture "99" @ 16,27 say incomplete picture "99"
1828 1829 1830 1831 1831 1833 1834 1835 1836 1837 1838 1837 1838 1849 1841 1843 1844 1845 1847 1848 1849 1851 1855 1856 1857 1858	*****	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast ************************************
1828 1829 1830 1831 1831 1832 1833 1834 1835 1836 1837 1838 1837 1838 1839 1840 1841 1842 1843 1844 1845 1854 1855 1856 1855 1856 1857 1858 1857 1858 1857	****	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast ************************************
1828 1829 1830 1831 1831 1833 1834 1835 1836 1837 1838 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1851 1852 1855 1856 1857 1858 1859 1867	******	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) Calls: INKEY() (FOXBASE+ function) rocedure plapast ************************************
1828 1829 1830 1831 1831 1833 1834 1835 1836 1837 1838 1837 1838 1837 1838 1839 1840 1841 1842 1843 1844 1845 1851 1852 1855 1855 1855 1857 1857 1857 1857 1857	*****	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function)
1828 1829 1830 1831 1831 1832 1833 1834 1835 1836 1837 1838 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1855 1856 1857 1857 1857 1857 1857 1857 1857 1857	******	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast ************************************
1828 1829 1830 1831 1831 1832 1833 1834 1835 1836 1837 1838 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1857 1858 1857 1858 1857 1858 1857 1857	******	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast ************************************
1828 1829 1830 1831 1831 1832 1833 1834 1835 1836 1837 1838 1840 1841 1842 1843 1844 1845 1854 1855 1856 1857 1858 1859 1850 1851 1852 1853 1854 1855 1856 1860 1861 1862 1864 1865 1864	******	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast
1828 1829 1830 1831 1831 1831 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1847 1848 1849 1850 1851 1855 1855 1857 1858 1857 1858 1857 1856 1860 1861 1867 1866	******	This procedure displays the reports concerning the Procedure: PLAPAST schedule performance. Called by: PLAREPO (procedure in PLA4.PRG) Calls: INKEY() (FOXBASE+ function) procedure plapast

Ħ

*

1871 @ 18,43 say "% over total orders" @ 18,65 say per_fin picture "999.99" @ 18,74 say mper_fin picture "999.99" ? inkey(3) set color to clear set color to r/n set color to r/n @ 4,23 say " @ 5,23 say " @ 6,23 say " @ 7,23 say " @ 10, 2 say "Week" PLANNING DEPARTMENT'S PAST . PERFORMANCE (Page 2) (a) 10, 2 say "Week"
(a) 10, 2 say "Week"
(a) 10, 22 say "Date"
(a) 10, 22 say "Date"
(a) 10, 32 say date
(a) 12, 22 say "Code"
(a) 12, 32 say "Descript"
(a) 12, 32 say "Descript"
(a) 12, 32 say "Descript"
(a) 12, 32 say "DAY"
(a) 15, 32 say "OVERALL"
(a) 15, 70 say "OVERALL"
(a) 15, 70 say "OVERALL"
(a) 15, 70 say "Production"
(a) 16, 3 say "Production picture "9999999.99"
(a) 16, 30 say mave_prod picture "9999999.99" 1891 @ 10,18 say production picture "9999999.99"
@ 16,30 say mave_prod picture "9999999.99"
@ 16,44 say "Num. of setups "
@ 16,63 say setupnum picture "999"
@ 16,72 say mave setn picture "999"
@ 17,3 say "Batch size "
@ 17,20 say ave batch picture "999.99"
@ 17,32 say mave batch picture "999.99"
@ 17,44 say "Lead-time"
@ 17,20 say linead time nicture "909.99" 1904 1905 1906 1907 (a) 17,44 say "Lead-time"
(b) 17,61 say leadtime picture "999.99"
(c) 17,70 say mave lead picture "999.99"
(c) 18,30 say mave picture "9999999.99"
(c) 18,30 say mave wip picture "9999999.99"
(c) 18,44 say "Standard cost"
(c) 18,61 say stand cost picture "999.99"
(c) 18,70 say mave cost picture "999.99"
(c) 18,70 say mave cost picture "999.99"
(c) 18,70 say mave cost picture "999.99"
(c) 18,70 say mave cost picture "999.99" 1912 1915 ? inkey(3) set color to clear return *!*** ***** This procedure displays the reports concerning the Procedure: MACHPAST resources performance. *1 1922 *! *! *! Called by: PLAREPO (procedure in PLA4.PRG) * *! Calls: INKEY() (FOXBASE+ function) *! *!***** ***** ******* procedure machpast 1930 set color to r/n

 get
 4,23 say "

 get
 5,23 say "

 get
 6,23 say "

 get
 6,23 say "

 get
 7,23 say "

 get
 10, 0 say "Week"

 PLANNING DEPARTMENT'S PAST 11 " PERFORMANCE (Page 3) I (a) 7,25 say "Week"
(a) 10,0 say "Week"
(a) 10,20 say "Date"
(a) 10,20 say "Date"
(a) 10,20 say "Date"
(a) 10,20 say "Date"
(a) 12,20 say "Resource"
(a) 12,20 say "Descript"
(a) 12,20 say "Descript"
(a) 12,20 say "Machine Capacity"
(a) 12,20 say "Machine Capacity"
(a) 12,20 say "Descript"
(a) 12,20 say "Descript"
(a) 12,20 say "Descript"
(a) 12,20 say "Descript"
(a) 12,20 say "Descript"
(a) 12,20 say "Descript"
(a) 12,20 say "Descript"
(a) 12,20 say "Descript"
(a) 12,20 say "Descript"
(a) 12,20 say "Descript"
(a) 12,20 say "Descript"
(b) 12,20 say "Descript"
(c) 12,20 say "Number of Set-Ups"
(a) 16,25 say setupnum picture "99"
(a) 16,25 say setupnum picture "99"
(a) 16,25 say "OVERALL"
(a) 16,35 say "OVERALL"
(a) 16,35 say "OVERALL"
(a) 16,35 say "Machine Production"
(a) 17,03 say "Machine Production"
(a) 17,32 say mave_prod picture "9999.99" 1937 1938 1941 1942 1943 1944 1945 1945 1950 1951 1952 1953

@ 17,42 say "Machine Productivity" @ 17,65 say productiv picture "999.99" @ 17,73 say mave_prot picture "999.99" @ 18, 0 say "Time spend in Set-Ups" @ 18,23 say setuptime picture "9999.99" @ 18,42 say "% of Set-Ups" @ 18,42 say "% of Set-Ups" @ 18,42 say "% of Set-Ups" 1955 1956 1957 1958 1959 1960 1961 1962 @ 18,65 say per_sett picture "999.99" @ 18,73 say mper_sett picture "999.99" ? inkey(3) set color to 1963 1964 1965 1966 clear 1967 1968 1969 return *!**** This procedure displays the reports concerning Procedure: STKPAST the stock performance. 1970 1971 *Ì 1972 *İ *! 1973 Called by: PLAREPO (procedure in PLA4.PRG) 1974 * 1975 Calls: INKEY() (FOXBASE+ function) 1976 1977 +1 1978 procedure stkpast 1979 1980 set color to r/n 4,23 say " 5,23 say " 6,23 say " 7,23 say " 1981 ø (27,23 say " (26,23 say " (27,23 say " (210,0 say "Week" (210,15 say actweek (210,20 say "Date" (210,20 say "Date" (210,30 say date PLANNING DEPARTMENT'S PAST PERFORMANCE (Page 4) || " . 1982 1983 1984 1985 1986 1987 ^(a) 10,20 say "Date"

 ^(a) 10,30 say date

 ^(a) 12,10 say "Code"

 ^(a) 12,13 say actcode

 ^(a) 12,20 say "Descript"

 ^(a) 12,30 say mdescript

 ^(a) 12,30 say mdescript

 ^(a) 12,30 say mdescript

 ^(a) 15,32 say mdescript

 ^(a) 15,32 say mdescript

 ^(a) 15,32 say "OVERALL"

 ^(a) 16,0 say "Level of stocks"

 ^(a) 16,21 say stock picture "999999999"

 ^(a) 16,21 say mave_stk picture "999999999"

 ^(a) 16,21 say mave_stk picture "9999999999"

 ^(a) 16,21 say stock picture "999999999"
 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 clear 2001 return 2002 ** 2003 2004 2005 *1**** ***** This procedure obtains the most updated value of Function: SYSTEM() the fields of the <SYSTEM> datafile. *į *1 2006 +1 Called by: PLA4.PRG 2007 *! 2008 *! : CHEKSTOP (procedure in PLA4.PRG) : PLA_BROW : GETKEY (procedure in PLA4.PRG) (procedure in PLA4.PRG) 2009 *1 +j 2010 *! 2011 2012 (FOXBASE+ function) (FOXBASE+ function) Calls: STR() 2013 * : SELECTO 2014 *į 2015 *14 procedure system 2016 2017 parameters fielddd 2018 2019 store str(select(),2) to base 2020 select 10 2021 goto top system=&fielddd select &base 2022 2023 2024 return system 2025 ** * *********** 2026 ****** This procedure calls the procedures used to create Procedure: CREANEW a new order in the <PLAN> datafile. 2027 *1 +į 2028 *į 2029 2030 *į Called by: PLA_BROW (procedure in PLA4.PRG) +! 2031 Calls: .NOT.SYSTEM() (FOXBASE+ function) : RECCOUNT() (FOXBASE+ function) : EOF() (FOXBASE+ function) : RECNO() (FOXBASE+ function) : RECNO() (FOXBASE+ function) 2032 *! *! 2033 2034 *! +1 2035 : SAM DISP : CHR() 2036 *! (procedure in PLA4.PRG) (FOXBASE+ function) (procedure in PLA4.PRG) 2037 *! 2038 *! : NEWPLAN 2039 *1
2040	*!************************
2041	procedure creanew
2042	*******
2043	if .not.system("SYS STOP")
2044	f if reccount() $< >0.$ and $r > 20$
2045	do while .not. eof()
2046	goto recnumtop
2047	skip skiprecs
2048	
2049	goto bottom
2050	endif
2051	recnumtop = recno()
2052	recnum = recno()
2053	goto recnumtop
2054	enddo
2055	goto recnumtop
2056	do sam_disp with (rowtop), skiprecs
2057	goto recnum
2058	row=rowtop
2059	set color to &promptatr
2060	@ row,0 say chr(16)
2061	endif
2062	do newplan
2063	goto top
2064	recnumtop = recno()
2065	pagepaint = .t.
2066	endif
2067	return
2069	*: EOF: PLA4.ACT

* 7 si

.....

5. 2 3.14

e

System: PLA4.PRG Author: JUAN IGNACIO IGARTUA 08/05/91 18:27:26 Tree Diagram

PLA4.PRG
SYSTEM.DBF (database)
ERROR() (function in ?)
MESSAGE() (function in ?)
ERR FIX (procedure in PLA4.PRG)
DEDLICATEO (function in ?)
CHP() (function in ?)
SYSTEM() (function in PLA4.PRG)
STR() (function in ?)
SELECT() (function in ?)
OPEN (procedure in PLA4.PRG)
BILL.DBF (database)
DI ANN DEE (database)
STOCK DBE (database)
MACHINE.DBF (database)
PLAPAST.DBF (database)
STKPAST.DBF (database)
MACHPAST.DBF (database)
SETUP.DBF (database)
CHEKSTOP (noncedure in DI A4 BPC)
SYSTEM DBF (database)
SYSTEM() (function in PLA4.PRG)
STRO (function in ?)
SELECT() (function in ?)
.AND., NOT.SYSTEM() (function in ?)
PLANKEPO (procedure in PLA4.PRG)
STR() (function in ?)
VALO (function in ?)
EOFO (function in ?)
INT() (function in ?)
SUBSTR() (function in ?)
PI A AVE (proceedings in DI A 4 DD G)
STKAVE (procedure in PLA4.PRG)
MACHAVE (procedure in PLA4.PRG)
PLAPAST (procedure in PLA4.PRG)
SIKPASI (procedure in PLA4.PRG)
MACHPAST (nrocedure in PLA4, PRG)
INKEY() (function in ?)
PRT (procedure in PLA4.PRG)
SYSO (function in ?)
PLANKEPO (procedure in PLA4.PKG)
TRO (function in ?)
VALO (function in ?)
EOFO (function in ?)
SUBSTR() (function in ?)
DIOC() (function in ?)
STKAVE (procedure in PLA4.PRG)
MACHAVE (procedure in PLA4.PRG)
PLAPAST (procedure in PLA4.PRG)
INKEY() (function in ?)
STKPAST (procedure in PLA4.PRG)
MACHPAST (procedure in PI A4 PPG)
INKEY() (function in ?)
.ANDNOT.SHUT.ANDNOT.SYSTEM() (function in ?)
.NOT.\$YSTEM() (function in ?)
AND.SYSTEM() (function in ?)
TNDTMPS (necessary in PLA4 DP(1)
PI A BROW (procedure in PLA4.PRG)
CHR() (function in ?)
RECNO() (function in ?)
FILT RESO (procedure in PLA4.PRG)
SELECT() (function in ?)
PLA_AREA1 (procedure in PLA4.PRG)
EOF() (function in ?)
PLA_RESO (procedure in PLA4.PRG)

280



281

11 10 12 51 3 23

Sache Land - States and the second in the second and the

---INKEY() (function in ?) ---UPPER() (function in ?) ---AND.SYSTEM() (function in ?) ---SELECT() (function in ?) ---CHEKSTOP (procedure in PLA4.PRG) SYSTEM.DBF (database) SYSTEM.DBF (database) PO (procedure in PLA4.PRG) AREPO (procedure in PLA4.PRG) STR() (function in ?) -VAL() (function in ?) -BOF() (function in ?) -INT() (function in ?) -SUBSTR() (function in ?) -DTOC() (function in ?) -PLAAVE (procedure in PLA4.PR(-STKAVE (procedure in PLA4.PR((procedure in PLA4.PRG) (procedure in PLA4.PRG) (procedure in PLA4.PRG) STKAVE MACHAVE PLAPAST (procedure in PLA4.PRG) L PRT -SYS() (function in ?) -PLANREPO (proc (procedure in PLA4.PRG) AREPO (procedure in PLA4.PRG) AREPO (procedure in PLA4.PRG) -STR() (function in ?) -VAL() (function in ?) -EOF() (function in ?) -SUBSTR() (function in ?) -DTOC() (function in ?) PLAREPO DTOC() (function in ?) (procedure in PLA4.PRG) (procedure in PLA4.PRG) PLAAŬĚ STKAVE (procedure in PLA4.PRG) (procedure in PLA4.PRG) MACHAVE PLAPAST -INKEY() (function in ?) PAST (procedure in PLA4.PRG) STKPAST MACHPAST (procedure in PLA4.PRG) MACHPA .NOT.SYSTEM() (function in ?) .AND.SYSTEM() (function in ?) ANEW (procedure in PLA4.PRG) -NOT.SYSTEM() (function in ?) -RECCOUNT() (function in ?) -EOF() (function in ?) -RECNO() (function in ?) -SAM DISP (procedure in P -STR() (function in ?) -BOF() (function in ?) -CREANEW (proce SAM DISP (procedure in PLA4.PRG) —ROW() (function in ?) —COL() (function in ?) CHR() (function in ?) NEWPI AN NEWPLAN (procedure in PLA4.PRG) -FOUND() (function in ?) PL STOŘ (procedure in PLA4.PRG) EDIT (procedure in PLA4.PRG) -PL_GETS (procedure in PLA4.PRG) PLEDIT (procedure in PLA4.PRG) STR() (function in ?)

14. 14 M 14 14 14 14 14 14 14





See. 3. 4.



System: PLA4.PRG Author: JUAN IGNACIO IGARTUA 08/05/91 18:27:17 Database Structure Summary 10 databases in the system SYSTEM.DBF STATUS.DBF BILL.DBF BILL.DBF PLANN.DBF STOCK.DBF MACHINE.DBF PLAPAST.DBF STKPAST.DBF MACHPAST.DBF SETUP.DBF Structure for database : SYSTEM.DBF Number of data records : 1 Last updated : 07/01/91 a 9:17 at LABU UPC Field name STARTDATE FACTDATE FACTDATE FACTTIME SYS STOP GAM-STOP HOUR RATE TOT OVERH NEWENO Type Date Date Width End Field Dec Start 8 8 19 17 122 230 860212299090492334492355482667888899001 1001 123 Character Logical Logical ã 1 1 7 10 4 56789 2 2 Numeric Numeric NEWENO NEWENÔREC FREQUÊNCY GOODSERV Logical Character Numeric Logical Logical Character 40 41 50 53 19311444 10112345678 GOODSERV REINDEX STATRESO STATCODE WAITTIME RESTART RESTART RESTATUS THROUGHPUT 54 55 59 Character 63 67 68 69 79 Numeric Logical Logical Numeric 1 10 10 222 19 20 22 22 23 ASSETS Numeric 89 99 SYSTOCK Numeric īõ Logical Logical SALREPORT 1111 PLAREPORT 100 101 ENQUPERIOD Numeric 102 * * Total FoxDoc did not find any associated index files Used by: PLA4.PRG : CHEKSTOP (procedure in PLA4.PRG) Structure for database : STATUS.DBF Number of data records : 7 Last updated : 07/01/91 at Field Field name Type Width 9:13 Type Character Dec Start End CODE 4 1 **4** 8 1234567 OUANTIT STOCK DATE RESOURCE Numeric 4 5 9 Numeric 8 16 28 32 37 43 48 17 25 29 38 Date 8 Character 4456521 161 CAPACITY Numeric Numeric TIME 22 8 9 SETUP TOTTIME Numeric Character 44 10 CDATE Character 6Õ ** Total ** FoxDoc did not find any associated index files (procedure in PLA4.PRG) (procedure in PLA4.PRG) Used by: OPEN : GETKEY Structure for database : BILL.DBF Number of data records : 5

Last upd Field Field name 1 CODE 2 DESCRIPT 3 MINDELIVDD 4 MAXDELIVDD 5 MAXQUANTIT 6 MINQUANTIT 7 STANDPRICE 8 MATCOST 9 TOT ENQU 10 PRICE LIMT 11 DISCOUNT 12 QUANT DISC 13 DELIVILIMT 14 MRKT PRICE 15 MRKTTLIMT 16 ONTIMEDELV 17 ONTIMLIMT 18 STANDHOURS 19 OVERHEAD 20 STAND COST 21 NUM ENQU 22 RESOUR COD 23 PROCESTIME 24 SET UP ** Total *T This database appe : BILLNA, ID	ated : 07/01/ Type Character Character Numeric Character	91 at 9 Width 20 22 26 66 66 46 46 46 46 66 66 66 66 66 66 44 57 139 0ciated	9:09 Dec 2222 2222 2222 2222 2222 2222 2222	Start 1 25 27 29 35 41 47 53 57 63 67 73 77 83 89 95 101 107 113 119 123 127 132 ndex file(s	End 24 26 28 34 40 46 56 62 66 72 76 82 88 100 106 112 126 131 138 3):	
Used by: OPEN	(proced	ure in 1	PLA4.PRO	})		
Structure for data Number of data rec Last upd Field Field name 1 CODE 2 DESCRIPT 3 QUANTIT 4 START DATE 5 START TIME 6 ORDER NUM 7 SEQUENCE 8 RESOUR COD 9 PRODUCT 1 10 PRODUCT 1 10 PRODUCT 1 10 PRODUCT 0 11 ORD STAT 0 13 FINIS DATE 14 FINIS TIME 15 NEED TIME ** Total ** This database appe : ORPLANN.I	base : PLANN. ords : ated : 07/01/ Type Character Character Numeric Date Character Numeric Character Numeric Character Character Date Character Date Character Date Character Date Character Sars to be ass DX (RESOUR_C	DBF 5 91 at 9 Width 20 6 8 4 4 4 10 10 8 8 4 10 10 9 00+CHR(9:17 Dec 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	Start 1 5 25 31 43 47 55 65 75 83 91 99 103 ndex file(s	End 24 30 38 42 46 50 54 64 74 82 90 98 102 108 5):	
Used by: OPEN	(proced	ure in 1	PLA4.PRO	3)	la de teste	
Structure for data Number of data rec Last upd Field Field name 1 CODE 2 DESCRIPT 3 STOCK ** Total **	base : STOCK. ords : ated : 07/01/ Type Character Character Numeric ars to be ass	DBF 91 at 9 Width 20 8 33 ociated	9:13 Dec with in	Start 1 25 25	End 4 24 32	
: CODESTR.I	DX (CODE)	ure in 1	DT.34 DD		-,.	
obed by. OFER	(Proced		DAT.FR	*/		

S 42 1 2 2 3

×.

-2 Stel

1.6

......

Structure for data Number of data rec Last updi Field Field name 1 RESOUR COD 2 REDESCRIPT 3 AVAIL 4 T AVAIL 5 CAPACITY 6 LASTSETUP 7 SET UP1 8 SET UP1 9 SET UP3 10 SET UP3 10 SET UP5 ** Total **	Dase : MACHIN Drds : ated : 07/01/ Type Character Character Character Numeric Numeric Numeric Numeric Numeric Numeric Numeric	E.DBF 391 at 9 Width 15 3 4 7 6 6 6 6 9 001 at 9 001	22222222222222222222222222222222222222	Start 1 20 23 28 32 39 45 51 57 63 rdov filo(End 4 19 227 31 38 44 50 56 62 68	
: RESOUR.ID		D)	PLA4. PR	G)	5,.	
	(Proced					
Structure for data Number of data rec Last upd Field Field name 2 DESCRIPT 3 INCOMPLETE 4 STARTED 5 FINISHED 6 PRODUCTION 7 WIP 8 AVE BATCH 9 SETUPNUM 10 LEADTIME 11 STAND COST 12 DATE ** Total **	base : PLAPAS ords : ated : 07/01/ Type Character Character Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Date	T.DBF 0 91 at 9 Width 20 3 10 10 6 8 83	2 2 2 2 2 2 2 2 2	Start 5 25 28 31 34 44 60 63 63 63 75	End 24 27 30 33 43 59 62 68 74 82	
This database appe : CDPLA.IDX	ars to be ass (CODE+DTOC(ociated DATE))	with i	ndex file(8):	
Used by: OPEN	(proced	ure in H	PLA4.PR	G)		
Structure for data Number of data rec Last upd Field Field name 1 CODE 2 DESCRIPT 3 STOCK 4 DATE ** Total **	base : STKPAS ords : ated : 07/01/ Type Character Character Numeric Date	T.DBF 5 91 at 9 Width 20 8 8 41	9:09 Dec	Start 1 5 25 33	End 4 24 32 40	
This database appe : CDSTK.IDX	ars to be ass (CODE+DTOC(ociated DATE))	with i	ndex file(8):	
Used by: OPEN	(proced	ure in H	PLA4,PR	G)		
Structure for data Number of data rec Last upd Field Field name 1 RESOUR COD 2 DESCRIPT 3 CAPACITY 4 PRODUCTION 5 PRODUCTIV 6 SETUPTIME 7 SETUPNUM	base : MACHPA ords : ated : 07/01/ Type Character Character Numeric Numeric Numeric Numeric Numeric	ST.DBF 391 at 9 Width 4 15 4 7 6 7	2 2 2 2	Start 1 5 20 24 31 37	End 4 19 23 30 36 43	

Bater

Saud & Bassier J.

A. 13.

12

S PATTER STATE

Nº S

This database appears t : CDMACH.IDX (F	o be associate ESOUR_COD+DTOC	d with inde: (DATE))	k file(s)	:
Used by: OPEN	(procedure in	PLA4.PRG)		
Structure for database Number of data records Last updated Field Field name Type 1 CODE Char 2 TYPE Char 3 SETUP Nume ** Total **	: SETUP.DBF : 5 : 07/01/91 at width acter 4 acter 7 eric 6 18	9:09 Dec S [.] 2	tart 1 5 12	End 4 11 17
This database appears t : SETUP.IDX (CC	o be associate DDE)	d with inde	κ file(s)	:
Used by: OPEN	(procedure in	PLA4.PRG)		

- 14 C

St. C. Ser

3......

A 18 10. 16

System: PLA4.PRG Author: JUAN IGNACIO IGARTUA 08/05/91 18:27:19 Data Dictionary

Field Name ASSETS AVAIL AVE BATCH CAPACITY	Type N C N N	Len 10 3 6 4	Dec 2 0 2 0	Database SYSTEM.DBF MACHINE.DBF PLAPAST.DBF MACHPAST.DBF STATUS DBF
CDATE CODE	cc	12 4	0 0	MACHINE.DBF STATUS.DBF PLANN.DBF BILL.DBF SETUP.DBF PLAPAST.DBF STKPAST.DBF
DATE	D	8	0	STATUS.DBF STOCK.DBF PLAPAST.DBF STKPAST.DBF STATUS.DBF
DELIV LIMT DESCRIPT	N C	4 20	0 0	MACHPAST.DBF BILL.DBF STKPAST.DBF PLAPAST.DBF STOCK.DBF PLANN.DBF
DESCRIPT DISCOUNT ENQUPERIOD FACTDATE FACTDATE FACTTIME FINIS DATE FINIS TIME FREQUENCY GAM STOP GOODSERV HOUR RATE INCOMPLETE LASTSETUP LEADTIME MATCOST MAXDELIVDD MAXOUANTTT	С N N D C N D C N L L N N C N N N	1541 843 843117376626	02000000002002200	BILL.DBF MACHPAST.DBF BILL.DBF SYSTEM.DBF SYSTEM.DBF PLAPAST.DBF PLAPAST.DBF PLANN.DBF SYSTEM.DBF SYSTEM.DBF SYSTEM.DBF SYSTEM.DBF PLAPAST.DBF PLAPAST.DBF BILL.DBF BILL.DBF BILL.DBF
MINDELIVDD MINOUANTIT MRKT LIMT MRKT PRICE NEED TIME NEWENQ NEWENQREC NUM ENQU ONTIMEDELV ONTIMLIMT ORDER NUM ORD STAT 0 ORD STAT 0 ORD STAT 1 OVERHEAD PLAREPORT PRICE LIMT PROCESTIME PRODUCTION PRODUCTION PRODUCTIV	NNNNNLCNNNNCCNLNNNN	026666194664886165706 10	202222000222000222222	BILL.DBF BILL.DBF BILL.DBF BILL.DBF PLANN.DBF SYSTEM.DBF BILL.DBF BILL.DBF BILL.DBF PLANN.DBF PLANN.DBF PLANN.DBF BILL.DBF BILL.DBF BILL.DBF BILL.DBF BILL.DBF BILL.DBF MACHPAST.DBF

PRODUCT 0 PRODUCT 1 QUANTIT QUANTIT QUANT DISC REDESCRIPT REINDEX RESOURCE RESOUR_COD	NNNNNC LCC	10 10 6 4 6 15 1 4 4	2 2 2 0 2 0 0 0 0 0	PLANN.DBF PLANN.DBF PLANN.DBF STATUS.DBF BILL.DBF MACHINE.DBF SYSTEM.DBF BILL.DBF MACHINE.DBF MACHINE.DBF
RESTART RESTATUS SALREPORT SEQUENCE SETUP	L L N N N	1 1 1 4 6	0 0 0 2	PLANN.DBF SYSTEM.DBF SYSTEM.DBF PLANN.DBF SETUP.DBF SETUP.DBF
SETUPNUM SETUPTIME SET UP SET UP1 SET UP2 SET UP3 SET UP3 SET UP4 SET UP5 STANDHOURS STANDPRICE STAND COST	N N N N N N N N N N N N N N N N N N N	2377666666666	002022222222222222222222222222222222222	MACHPAST.DBF PLAPAST.DBF MACHPAST.DBF BILL.DBF MACHINE.DBF MACHINE.DBF MACHINE.DBF MACHINE.DBF BILL.DBF BILL.DBF BILL.DBF
STARTDATE STARTED START DATE START TIME STATCODE STATRESO STOCK	D N D C C C N	8 3 8 4 4 8	0 0 0 0 0 0	PLAPAST.DBF SYSTEM.DBF PLAPAST.DBF PLANN.DBF SYSTEM.DBF SYSTEM.DBF STKPAST.DBF STATUS.DBF
SYSTOCK SYS STOP THROUGHPUT TIME TOTTIME TOT ENQU TOT OVERH TYPE T AVAIL WAITTIME WIP	N L N N C N N C N N N	10 10 5 4 10 7 5 4 10	2 0 2 2 0 0 2 0 0 0 0 0 2	STOCK.DBF SYSTEM.DBF SYSTEM.DBF STATUS.DBF STATUS.DBF BILL.DBF SYSTEM.DBF SETUP.DBF MACHINE.DBF SYSTEM.DBF PLAPAST.DBF

MARKETING/SALES SUB-PROGRAM

SAL6.PRG

1	*:*	This anoram is the one that simulates all the activities
3	*	Program: SAL6.PRG of the sales department. Together with the Planning and System
4	*	ones, conforms the Nottingham Polytechnic's gaming-simulation.
5	*:	System: SAL6.PRG
6	*:	Author: JUAN IGNACIO IGARTUA Convright (a) 1001 NOTTINGHAM POL VIECHNIC
8	*	Last modified: (08/05/91 17:48
ğ	•	
10	*:	Procs & Fncts: ERR_FIX
11	*:	: SYSTEM()
13		CHEKSTOP
14	*:	: MAINMENU
15	*:	: INPTQUO
10	*.	SAL FORM
18	+	: SAL GETS
19	*:	: SAL_STOR
20	*:	: MARKING
22	*.	· SAL KEFL
23	*:	SAL EDIT
24	*:	: SAL DISP
25	-	: GETKEY
27		: SAL AREAO
28	*:	: SAL_AREA1
29	*:	: SAL_CODE
31	*.	SAL SIAI
32	*:	: FILT STAT
33	*:	: SALREPO
34	*:	SALAVE
36	+	: PRT
37	*:	: SAL_BROW
38	*:	: POS
40	*.	ACCEPT
41	*:	Calls: ERROR() (FOXBASE+ function)
42	*:	: MESSAGE() (FOXBASE + function)
43	*.	: EKK_FIX (procedure in SALO.PKG) : CTOD() (FOXBASE+ function)
45	*:	: REPLICATE() (FOXBASE+ function)
46	*:	: CHR() (FOXBASE+ function)
47	*.	: OPEN (nunction in SAL6 PRG)
49	*:	: SELECT() (FOXBASE+ function)
50	*:	: CHEKSTOP (procedure in SAL6.PRG)
52	-	: MAINMENU (procedure in SALO.PRG)
53	*	Uses: SYSTEM.DBF
54	*:	
55	*	
57		•
58		set date british
60		on error do err fix with error() message()
61		set default to v:
62		set status off
64		set talk off
65		set scoreboard off
66		set heading off
67		* authlic signal material desire managements around the second stars of
69		moutime.moudeliver.mcode.ngdn.returnkey.ngun
70		public delrecord, recnum, choice, promptrow, promptbar,;
71		oldrecnum, screenatr, statusatr, windowatr, signal1
73		public promptatr, hiliteatr, fillstat, fillcode, ordchoice,;
74		public firstpos, lastpos, neweng, pause, start, waittime, last disp.;
75		browrec, move, bkey, xx, filtstat0, inedit, mmfactdate, chgorder
76		public tempdd, tempdd1
78		screenstr="R+/N.N/W"
79		statusatr = "BU/N,N/W"
80		windowatr="R+/N,N/W"
81		promptatr="GK+/N,N/W" bilitestr="N/W"
83		*
84		filtstat = "ENQU" + "QUOT"

.....

filtstat0="ENQU"+"QUOT" 85 86 filtcode = "ALL 87 88 89 90 91 92 93 rowbottom = 20rowprompt = rowbottom + 3store 0 to recnum,oldrecnum,browrec,lastdisp store ctod("01/01/99") to firstpos,lastpos store " to choice,xx store "2" to ordchoice store "0" to codchoice 94 95 96 97 98 99 promptrow=22 promptbar=replicate(chr(196),80) promptoar = replicat pgdn=chr(3) pgup=chr(18) returnkey=chr(13) delrecord=chr(7) 100 101 bkey = .f.inedit = .f. 102 103 104 105 start=.f. signal=.f. 106 signal1 == .f. 107 sys start=.t. newenq = .f.pause = .f. 108 109 110 move=.t. 111 112 113 chgorder=.f. set exclusive off 114 select 10 use system 115 116 117 scr=.f. 118 119 120 =do while system("reindex") && wait until the reindex is finished && in the system program. if .not.scr set color to bu+/n 121 122 @ 7,24 to 12,54 double 9,25 to 10,57 to 000
 9,25 say "REINDEXING THE MASTER SYSTEM"
 (a) 10,34 say "PLEASE WAIT" 123 124 125 scr=.t. 126 endif enddo 127 128 ** 129 waittime=system("waittime") 130 131 132 set color to clear 133 134 set color to bu + /n@ 7,24 to 12,54 double @ 1,24 to 12,34 double
set color to gr+/n
@ 9,27 say "SETTING THE SALES SYSTEM"
@ 10,34 say "PLEASE WAIT" 135 136 137 138 139 do open 140 141 set view to saldep 142 do chekstop with select() 143 144 145 146 store j->factdate to tempdd, tempdd1 set color to clear do mainmenu 147 * 148 *!*** 149 150 151 152 153 154 155 This procedure allows the user to call to two different Procedure: MAINMENU activities depending on the menu selection done. * * * * * Called by: SAL6.PRG *1 Calls: SELECT() (FOXBASE+ function) (procedure in SAL6.PRG) (procedure in SAL6.PRG) : CHEKSTOP *1 156 : INPTQUO 157 : INPTPRI (procedure in SAL6.PRG) 158 *1 *!* 159 ***** procedure mainmenu 160 161 162 do while .t. 163 set color to 164 165 clear m_menu=0 set color to g/n @ 3,16,17,62 box " == 166 167 ╕ |╛═╘| " 168 set color to gr/n @ 5,26 to 7,53 double 169

170 set color to bu/n 171 172 @ 6,32 say "SALES WORKAREA" set color to br/n 173 174 175 @ 11,24 to 14,55 (a) 11,21 to bg/n
 (b) 12,26 prompt "1.DISPLAY/INPUT OF QUOTATION"
 (c) 13,26 prompt "2.INPUT OF STANDARD PRICE "
 (c) menu to m_menu 176 177 178 179 do chekstop with select() do case 180 case m_menu=1 181 do inptquo 182 case m menu=2 183 do inptpri 184 endcase 185 enddo 186 187 188 * *1*** This procedure calls the display/input procedures used Procedure: INPTQUO to quote the customers' enquires received. *! 189 *! *! *! 190 191 Called by: MAINMENU (procedure in SAL6.PRG) 192 *1 193 Calls: SAL_BROW (procedure in SAL6.PRG) 194 *į 195 *!* 196 procedure inptquo 197 198 clear 199 select 1 goto top do sal_brow 200 201 202 return 203 204 *!***** 205 *! This procedure creates the editing form used to quote 205 206 207 208 * + + + + Procedure: SAL_FORM the enquires. Called by: SAL_EDIT (procedure in SAL6.PRG) 209 *į 210 *! Calls: SYSTEM() (function in SAL6.PRG) 210 211 212 213 214 215 216 * * ******* procedure sal_form system1=system("factdate") @ 0,72 say system1 set color to &screenatr 217 218 clear 219 220 221 222 223 set color to &promptatr @ promptrow-1,0 say promptbar set color to bu/n @ 5,24 to 7,54 double 224 set color to r/n set color to r/n @ 6,31 say "QUOTATIONS INPUT" set color to br/n @ 10, 8 say "Article Code " @ 10,37 say "Description " @ 12, 8 say "Quantity " @ 13, 8 say "Quantity " @ 13,37 say "Quoted Price set color to bg/n @ 13,37 say "Quoted Price set color to br/n @ 14 & say "Delivery Date " 225 226 227 228 229 230 231 232 233 234 @ 14, 8 say "Delivery Date (@ 14, 8 say "Derivery Date set color to bg/n @ 14,37 say "Quoted Delivery Date set color to g/n @ 17, 8 say "Enquire Date @ 17,37 say "Quotation Date @ 18,38 say "Enquire Time @ 18,37 say "Quotation Time 235 236 237 238 239 240 241 242 return 243 244 245 246 * *********** *į This procedure shows the information attached to Procedure: SAL_SAYS each one of the enquires. *į 247 +į 248 249 250 251 252 *1 Called by: SAL_EDIT (procedure in SAL6.PRG) *! *! procedure sal_says 253 set color to n/w 254 @ 10,27 say code

@ 10,50 say descript
@ 12,26 say enquantit picture "999.99"
@ 13,26 say enprice picture "999.99" 255 256 257 258 259 260 set color to ,n/w picture "999.99" @ 13,64 get mquprice @ 14,24 say endeliver @ 14,62 get mqudeliver picture "@E" @ 17,24 say endate 261 262 263 @ 17,62 say qudate 264 @ 18,27 say entime 265 266 267 @ 18,65 say qutime clear gets return 268 ********** 269 270 271 272 273 274 This procedure allows the input of the quotation Procedure: SAL_GETS information attached to each enquire. *1 *į *j *! Called by: SAL_EDIT (procedure in SAL6.PRG) *į 275 276 277 278 +1 ******** procedure sal_gets set color to ,n/w @ 13,64 get mquprice picture "999.99" @ 14,62 get mqudeliver picture "@E" 279 280 281 read 282 return 283 285 *! 285 *! 286 *! 287 *! ***** * **** This procedure stores the default values in the Procedure: SAL_STOR quotation variables to be edit. 288 *j Called by: SAL_EDIT (procedure in SAL6.PRG) 289 + 290 291 292 293 ************ procedure sal stor * --- Initialize memvars with field contents and date(). 294 store quprice to mauprice store qudeliver to maudeliver 295 296 return 297 298 *!*** ***** This procedure marks the enquires after being Procedure: MARKING quoted. 300 +i 301 *Ì 302 *i Called by: SAL EDIT (procedure in SAL6.PRG) 303 *į SAL BROW (procedure in SAL6.PRG) *i *i 304 305 Calls: QUPRICE <>0() (FOXBASE+ function) *1 306 307 ***** 308 procedure marking 309 -if quprice < > 0.and.qudeliver < > ctod('00/00/00') replace status with 'QUOT' 310 311 312 unlock 313 314 315 -endif return 316 *!*** This procedure replaces the quotation information Procedure: SAL_REPL entered in the <SALES > datafile. 317 *! 318 319 *! *! 320 Called by: SAL EDIT (procedure in SAL6.PRG) 321 322 *1 *į Calls: EOF() (FOXBASE+ function) 323 *) 324 325 procedure sal_repl 326 327 mqudate=j->factdate mqutime=j->facttime if tempdd1>tempdd mqudate=(j->factdate-1) 328 329 330 331 332 endif -if .not. eof() 333 ---Replace only if there is an available record 334 replace; 335 qudate with mqudate,; 336 qutime with mqutime,; 337 quprice with mquprice,; 338 qudeliver with mqudeliver 339 unlock

340	L	endif
342	4	tuni *
343	+ **	*************
344	*	This procedure manages the possible conflicts that could Procedure: FPP, FIX, arise from the feat that we are working in a multi-user
345	*	environment. Its functioning is thoroughly explained in
347 348	*1 *1	Called by: SAL6.PRG one of the thesis's sections.
349 350	*1 *1	Calls: RLOCK() (FOXBASE+ function)
351	+]+4	***********
352	Ę	rocedure err fix
354	T	parameters errnum.mess
355	3	
350		•• Error: File in use by another.
358		save screen to screen
359		clear
361		a 7.17 clear to 15.61
362		@ 7,17 to 15,61 double
363		set color to r ⁺ +/bu @ 11.24 ray. "Please weit to append a record."
365		restore screen from screen
366		retry
367		endif
369		** Error: Record in use by another.
370	Г	if errnum=109
371		save screen to screen
373		time=0
374		set color to gr+/bu
376		(a) 7.17 to 15.61 double
377		set color to $r^* + /bu$
378		@ 11,26 clear to 11,54
380		@ 12.26 say "Press any key to continue."
381		read
382		do while .not. rlock().and.time < 1000
384		enddo
385		if time < 1000
387		restore screen from screen
388		else
389		clear
391		@7.17 clear to 15.61
392	- 1	@ 7,17 to 15,61 double
393		set color to r ⁺ +/bu @ 11.19 say "Record cannot be locked Try again later "
395		@ 12,26 say "Press any key to continue."
396		read
398		restore screen from screen
399	L	-endif
400		** Error: Record is not locked
402	Г	if errnum=130
403		time=0 (1000)
404		time = time + 1
406		enddo
407		retry
409		endif
410	5	endif
411	+ **	}************************************
413	*!	This procedure checks for the status of the game forcing the
414	*1	Procedure: CHEKSTOP Sales program to do different things in each case. Its functioning
416	*1	Called by: SAL6.PRG
417	*!	: MAINMENU (procedure in SAL6.PRG)
418	*1	: GEIKEY (procedure in SALO.PRG)
420	*!	Calls: SYSTEM() (function in SAL6.PRG)
421	*!	: .ANDNOT.SYSTE(FOXBASE + function)
423	+1	: PRT (procedure in SAL6.PRG)
424	*1	: .ANDNOT.SHUT.(FOXBASE + function)

: .NOT.SYSTEM() (FOXBASE+ function) : .AND.SYSTEM() (FOXBASE+ function) : STR() (FOXBASE+ function) 425 * 426 420 *! 429 *! Uses: SYSTEM.DBF 430 *1 431 procedure chekstop 432 433 434 435 parameters aselect private shut shut = .f. 436 signal = .f. 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 do while .t. do case coase system("sys_stop").and..not.system("gam_stop") _____if .not.pause if system("salreport") do salrepo do prt select 10 replace salreport with .f. unlock start=.t. endif if .not.signal save screen to screen1 set color to clear set color to gr+/bu @ 5,14 clear to 14,64 @ 5,14 to 14,64 double @ 3,14 to 14,64 double set color to bg +/bu @ 9,17 say "THE GAME HAS BEEN FROZEN OR A DAY HAS ENDED." @ 11,30 say "WAIT UNTIL RESTART." sys_start=.f. 458 459 460 461 462 463 signal=.t. endif if system("restart").and..not.shut.and..not.system("gam_stop") close databases 464 465 466 467 468 469 470 select 10 use system shut – .t. else -if .not.system("restart").and.shut set view to saldep 471 472 shut = .f.endif 473 endif 474 475 476 -lse filtstat = "FIRMLOSTENQU" set color to gr + */bu @ 24,70 say "STOPPED" 477 478 set color to 479 =exit 480 L -endif 481 case .not.system("gam_stop").and ..not.system("sys_stop") 482 483 -if signal.and..not.start.and..not.pause restore screen from screen1 484 485 -endif start=.f. 486 sys_start=.f. 487 set color to 488 @ 24,70 clear to 24,76 489 exit 490 case system("gam_stop").and.system("sys_stop") 491 492 -if sys_start -if .not.signal1 set color to 493 494 clear 495 set color to bu+/n 496 497 @ 7,24 to 12,54 double @ 9,26 say "SETTING THE MASTER SYSTEM" @ 10,34 say "PLEASE WAIT" 498 499 500 signal1 = .t. 501 start=.t. 502 endif 503 else 504 505 506 507 close all clear all set color to clear 508 set color to bu+/n 509 @ 8,24 to 12,54 double

297

Sugar Sec. Sec.

South & Contract of Street

A 10 10 10

\$10		
210	set color to gr+/n	
511	@ 10,29 say "THE GAME HAS FINISHED"	
512	set color to	
513	set console off	
514	<cancel< td=""><td></td></cancel<>	
515	endif	
516	endcase	
517		
518	sys start=.f.	
519	aselect = str(aselect)	
520	select &aselect	
521	return	
522		
523	`* <u></u> ***********************************	-
524	*! This procedure allows the input of the standard price	
525	*! Procedure: INPTPRI for the selected article.	
526	*	
527	*! Called by: MAINMENU (procedure in SAL6.PRG)	
528		
529	*) Calls: GETKEY (procedure in SAL6.PRG)	
530	+! : FOUND() (FOXBASE+ function)	
531	*	
532		
333	procedure inpipri	
534	manda	
535	mcode =	
530		
532	nrivate innchoice	
530	innchoice="*"	
540	set color to r/n	
541	@ 6.24 to 8.54 double	
542	set color to bu/n	
543	@ 7.28 say "INPUT OF STANDARD PRICE"	
544	set color to g/n	
545	@ 10.13 to 17.65	
546	set color to br/n	
547	@ 12,18 say "Code "	
548	@ 12,31 say "Descript "	
549	@ 15,18 say "Standcost "	
550	set color to bg/n	
551	@ 15,40 say "Standprice "	
552	•	
553	set color to gr+/n,n/w	
554	@ 22,0 say " EDIT: {E}xit <return> "</return>	
555	•	
220		
	do while .t.	
331	do getkey with inpchoice, "EC" + returnkey	
558	do while .t. do getkey with inpchoice, "EC" + returnkey	
557 558 559	do while .t. do getkey with inpchoice, "EC" + returnkey do case case inpchoice = "E"	
557 558 559 560	do while .t. do getkey with inpchoice, "EC" + returnkey edo case case inpchoice = "E" v	
557 558 559 560 561	do while .t. do getkey with inpchoice,"EC" + returnkey case inpchoice = "E" v====================================	
557 558 559 560 561 562	do while .t. do getkey with inpchoice, "EC" + returnkey case inpchoice = "E" v====================================	
557 558 559 560 561 562 563 564	do while .t. do getkey with inpchoice, "EC" + returnkey do case case inpchoice = "E" vexit vexit get color to ,n/w @ 12,25 get mcode picture "@!NNNN" valid mcode < >" "	
557 558 559 560 561 562 563 564 565	do while .t. do getkey with inpchoice, "EC" + returnkey do case case inpchoice = "E" vexit case inpchoice = returnkey set color to ,n/w @ 12,25 get mcode picture "@!NNNN" valid mcode < > " " read select 2	
557 558 559 560 561 562 563 564 565 565	<pre>do while .t. do getkey with inpchoice,"EC" + returnkey do case case inpchoice = "E" v=</pre>	
557 558 559 560 561 562 563 564 565 566 565	<pre>do while .t. do getkey with inpchoice,"EC"+returnkey ==do case case inpchoice="E" v======="case inpchoice=returnkey set color to ,n/w @ 12,25 get mcode picture "@!NNNN" valid mcode <>" " read select 2 find &mcode -==ff found()</pre>	
557 558 559 560 561 562 563 564 565 566 566 567 568	<pre>do while .t. do getkey with inpchoice,"EC"+returnkey do case case inpchoice="E" vertice v</pre>	
557 558 559 560 561 562 563 564 565 566 567 568 569	<pre>do while .t. do getkey with inpchoice,"EC" + returnkey do case case inpchoice = "E" v===================================</pre>	
558 559 560 561 562 563 564 565 566 566 566 567 568 569 570	<pre>do while .t. do getkey with inpchoice,"EC" + returnkey do case case inpchoice = "E" v========"">****************************</pre>	
557 558 559 560 561 562 563 564 565 566 567 568 569 570 571	<pre>do while .t. do getkey with inpchoice,"EC" + returnkey</pre>	
357 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572	do while .t. do getkey with inpchoice, "EC" + returnkey do case case inpchoice = "E" 	
558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573	<pre>do while .t. do getkey with inpchoice,"EC" + returnkey</pre>	
357 558 559 560 561 562 563 564 565 566 566 566 566 566 566 567 570 571 572 573 574	do while .t. do getkey with inpchoice, "EC" + returnkey do case case inpchoice = "E" v=	
357 558 559 560 561 562 563 564 565 566 566 566 566 566 566 570 571 572 573 574 575	do while .t. do getkey with inpchoice, "EC" + returnkey case inpchoice = "E" v======="texturnkey" set color to ,n/w @ 12,25 get mcode picture "@!NNNN" valid mcode <>" " read select 2 find & mcode 	
357 558 559 560 562 563 564 565 566 566 566 566 570 571 573 574 575 576	do while .t. do getkey with inpchoice, "EC" + returnkey case inpchoice = "E" case inpchoice = returnkey set color to ,n/w @ 12,25 get mcode picture "@!NNNN" valid mcode <>" " read select 2 find &mcode 	
557 558 559 560 561 562 563 564 565 566 565 566 570 571 572 573 574 575 576 577	<pre>do while .t. do getkey with inpchoice, "EC" + returnkey</pre>	
557 558 559 560 562 563 564 565 566 567 576 577 578 577 578	<pre>do while .t. do getkey with inpchoice, "EC" + returnkey</pre>	
557 558 559 560 562 563 564 565 566 565 566 566 568 570 571 572 573 574 575 577 577 577 577 577 577 577	do while .t. do getkey with inpchoice, "EC" + returnkey case inpchoice = "E" v====================================	
557 558 559 560 561 562 563 565 565 565 566 567 568 570 571 572 573 574 577 577 578 577 578 577 578 578 579 580	<pre>do while .t. do getkey with inpchoice, "EC" + returnkey case inpchoice = "E" v=</pre>	
557 558 559 560 561 562 563 564 5565 566 576 577 578 577 578 577 578 577 578 579 5580 5579 5580 5579 5580 5579 5580 577	<pre>do while .t. do getkey with inpchoice, "EC" + returnkey</pre>	
557 558 559 560 561 562 563 564 5565 566 565 566 570 571 572 573 574 577 577 577 577 577 577 577 577 577	<pre>do while .t. do getkey with inpchoice, "EC" + returnkey</pre>	
557 558 560 562 563 565 565 565 566 567 570 577 578 577 577 577 577 577 577 578 579 580 581 577 577 578 579 580 581 552 563 579 579 570 577 579 579 579 570 579 570 570 570 570 570 570 570 570 570 570	do while .t. do getkey with inpchoice, "EC" + returnkey —do case —case inpchoice = "E" vexit —case inpchoice = returnkey set color to ,n/w @ 12,25 get mcode picture "@!NNNN" valid mcode <>" " read select 2 find & mcode —if found() store standprice to mstandprice @ 12,42 say descript @ 15,30 say stand_cost @ 15,42 get mstandprice picture "999.99" read replace standprice with mstandprice unlock —else set color to gr+/bu @ 9,14 to 13,64 @ 9,14 to 13,64 @ 9,14 to 13,64 @ 9,14 to 13,64 @ 9,14 to 13,64 @ 9,14 to 13,64 Market Descent Start," @ 15,29 say "" wait "TRY ANOTHER ONE" retry	
557 558 559 560 561 5562 5563 5563 5563 5564 5563 5565 5566 5578 5770 5778 5776 5573 5778 5778 5778 5579 5581 5573 5579 5581 5579 5579 5579 5579 5579 5579 5579 557	<pre>do while .t. do getkey with inpchoice, "EC" + returnkey</pre>	
557 558 558 5563 5563 5563 5563 5563 5563 5	<pre>do while .t. do getkey with inpchoice, "EC" + returnkey</pre>	
557 558 5563 5563 5563 5563 5563 5563 5563	<pre>do while .t. do getkey with impchoice, "EC" + returnkey</pre>	
335 558 559 5560 5562 5563 5565 5565 5565 5566 5570 5570 5572 5574 5575 5576 5577 5578 5579 5580 5582 5582 5582 5582 5582 5582 5583 5582 5583 5582 5583 5582 5583 5583	do while .t. do getkey with inpchoice, "EC" + returnkey —do case case inpchoice = "E" v	
337 558 558 558 558 558 558 558 558 558 55	do while .t. do getkey with inpchoice, "EC" + returnkey case inpchoice="E" case inpchoice="E" case inpchoice=returnkey set color to ,n/w @ 12,25 get mcode picture "@!NNNN" valid mcode <>" " read solect 2 find &mcode find &mcode @ 12,42 say descript @ 15,30 say stand_cost @ 15,42 get mstandprice picture "999.99" read replace standprice with mstandprice unlock else set color to gr+/bu @ 9,14 clear to 13,64 @ 9,14 clear to 13,64 @ 9,14 clear to bg +/bu @ 11,21 say "THE CODE INPUT DOES NOT EXIST." @ 15,39 say "" wait "TRY ANOTHER ONE" retry endif endcase return *	
3357 558 559 560 5563 5564 5565 5665 5565 5565 5577 578 578 578 578 578 578 578 578 5	do while .t. do getkey with inpchoice, "EC" + returnkey do case case inpchoice = "E" vertified exit case inpchoice = returnkey set color to ,n/w @ 12,25 get mcode picture "@!NNNN" valid mcode <>" " read select 2 find &mcode 	*
558 559 5560 5561 5559 5561 5563 5564 5565 5565 5565 5565 5565 5577 578 5770 5772 578 5780 5581 5582 5583 5583 5585 5583 5583 5583 5583	<pre>do while .t. do getkey with impchoice, "EC" + returnkey do case case inpchoice = "E" vertice = returnkey set color to ,n/w @ 12,25 get mcode picture "@!NNNN" valid mcode <> " " read select 2 find &mcode if found() store standprice to mstandprice @ 12,42 say descript @ 15,30 say stand_cost @ 15,42 get mstandprice picture "999.99" read replace standprice with mstandprice unlock else set color to gr + /bu @ 9,14 clear to 13,64 @ 9,14 to 13,64 double set color to bg + /bu @ 11,21 say "THE CODE INPUT DOES NOT EXIST." @ 15,39 say "" wait "TRY ANOTHER ONE" return * This procedure calls the procedures used to edit/quote</pre>	•
337 558 559 550 550 550 550 550 550 550 550 550	<pre>do while .t. do getkey with inpchoice, "EC" + returnkey do case case inpchoice = "E" vertif case inpchoice = returnkey set color to ,n/w @ 12,25 get mcode picture "@!NNNN" valid mcode <> " " read select 2 find &mcode if found() store standprice to mstandprice @ 12,42 say descript @ 15,42 get mstandprice picture "999.99" read replace standprice with mstandprice unlock else set color to gr +/bu @ 9,14 clear to 13,64 @ 9,14 to 13,64 double set color to bg +/bu @ 11,21 say "THE CODE INPUT DOES NOT EXIST." @ 11,23 say "" wait "TRY ANOTHER ONE" return * * * * * * * * * * * * * * * * * * *</pre>	•
3357 558 559 5560 5562 5563 5565 5565 5565 5566 5570 5572 5576 5577 5578 5579 5580 5582 5583 5582 5583 5582 5583 5582 5583 5582 5583 5582 5583 5583	<pre>do while .t. do getkey with inpehoice, "EC" + returnkey do case case inpehoice = "E" view Xit case inpehoice = returnkey set color to ,n/w @ 12,25 get meode picture "@!NNNN" valid meode <> " " read select 2 find & meode if found() store standprice to mstandprice @ 12,42 say descript @ 15,30 say stand_cost @ 15,42 get mstandprice picture "999.99" read replace standprice with mstandprice unlock else set color to gr +/bu @ 9,14 to 13,64 double set color to bg +/bu @ 11,21 say "THE CODE INPUT DOES NOT EXIST." @ 15,39 say "" wait "TRY ANOTHER ONE" return * This procedure calls the procedures used to edit/quote * Procedure: SAL_EDIT a selected enquire.</pre>	•

Calls: RLOCK() (FOXBASE+ function) 595 **** 596 597 598 599 600 : SAL_FORM : SAL_STOR : SAL_SAYS (procedure in SAL6.PRG) (procedure in SAL6.PRG) (procedure in SAL6.PRG) (procedure in SAL6.PRG) : GETKEY .AND..NOT.SYSTE(FOXBASE+ function) 601 602 *! *! : SAYLINE (procedure in SAL6.PRG) SAL_GETS (procedure in SAL6.PRG) (procedure in SAL6.PRG) 603 604 605 *i (procedure in SAL6.PRG) 606 607 procedure sal_edit 608 609 do while .not. rlock() 610 enddo private row, lastpage, editchoice, ndxchoice private isblank, isunique, isedited 611 612 613 row = promptrow expr = 614 615 store .f. to isedited, isblank, isunique 616 do sal_form 617 do sal_stor 618 619 do sal_says editchoice = "*" 620 ndxchoice = "*" 621 -Loop until {Return} is pressed. 622 ÷ ... -The following loop is really a "REPEAT/UNTIL <cond>". 623 do while .t. @ row,0 say "EXIT/VIEW: {B}ack <Return> " do getkey with editchoice, "B"+returnkey 624 625 626 627 628 do case case editchoice = "B" 629 unlock 630 exit case editchoice = returnkey if status="ENQU".and..not.system("SYS_STOP") * ---Edit the record. 631 632 633 634 isedited = .t.635 do sayline with row, "Press {Ctrl-W} to Exit" 636 do sal_gets 637 638 do sal_repl endif 639 endcase 640 =enddo 641 -if isedited 642 do marking 643 644 645 646 pagepaint=.t. endif return 647 **** 648 649 This procedure displays all the existing enquires regardless Procedure: SAL_DISP of their status (quoted, firm or lost). * * * * * 650 651 652 653 Called by: SAL BROW (procedure in SAL6.PRG) Calls: ROW() (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) 654 *i : EOFO 655 * : SPACEO 656 * 657 658 procedure sal_disp 659 660 parameter row, listrecs ——if listrecs > 1 661 * ----Display heading when listing the entire page. set color to &statusatr 662 663 @ rowtop-1,0 @ rowtop-1,0 say "' 664 665 666 667 668 =do case =case pancol = 1 ?? "ARTICLE_CODE-DESCRIPTION-----PRICE----QUANTITY----DELIVERY_DATE--STATUS" =case pancol = 2 669 670 =endcase 671 * --- Clear the window area. 672 set color to &windowatr 673 @ rowtop,0 clear to rowbottom,79 674 675 endif * --- Display the records. 676 set heading off set color to &windowatr 677 @ row-1,0 say 678 679 =do case

680		
682		ist next instrees code, , descript, , quprice, , enquantit, , quuenver,
683		lastdisp = row()-2
684		endif
080		
687		set heading on
688		if eof()
689		goto bottom
691		
692		lastpos=qudeliver
693		endif
694		
696		replace newengree with space(9)
697		unlock
698		select 1
699		endif rotum
701		*
702	*!	************************
703	*!	This procedure gets a keyboard input done by
704	*1	Procedure: GETKEY the user.
706	*1	Called by: INPTPRI (procedure in SAL6.PRG)
707	*1	: SAL_EDIT (procedure in SAL6.PRG)
708	*1	: SAL CODE (procedure in SAL6.PRG)
709	*1	: SAL STAT (procedure in SALO.PRG)
711	+1	: SAL_BROW (procedure in SALO.PRO)
712	*!	Calls: SYSTEM() (function in SAL6.PRG)
713	*!	: SUBSTR() (FOXBASE+ function)
714	- T] + 1	: RECNO() (FOXBASE+ function)
716	*	: IIF() (FOXBASE+ function)
717	*!	: INKEY() (FOXBASE+ function)
718	*!	: UPPER() (FOXBASE+ function)
719	- T I - # 1	: CHK() (FOXBASE + function)
721	*	: CHEKSTOP (procedure in SAL6.PRG)
722	*!	· · · · · · · · · · · · · · · · · · ·
723	*!	######################################
725		**************************************
726		parameter choice, keychars
727		private keycode
720		choice = "+"
730		tempdd=tempdd1
731		tempdd1=j->factdate
732		if system("sys_stop")
734		store it to ser
735		do while system ("goodserv")
736		if scr
737		set typeahead to 0
739		m = 11.25 to 13.55 double
740		set color to $gr + */n$
741		@ 12,27 say "calculating sales response"
742		scr=.f.
743		$rac{1}{2}$
745		enddo
746		
747		if .not.goodfin
770		
749		if .not.goodfin set typeahead to 20 goodfin=.t. store (system("FACTDATE")-1) to mmfactdate
749 750		if .not.goodfin set typeahead to 20 goodfin=.t. store (system("FACTDATE")-1) to mmfactdate select 1
749 750 751		if .not.goodfin set typeahead to 20 goodfin=.t. store (system("FACTDATE")-1) to mmfactdate select 1 set filter to qudate=mmfactdate.or.status\$ filtstat0
749 750 751 752 752		if .not.goodfin set typeahead to 20 goodfin=.t. store (system("FACTDATE")-1) to mmfactdate select 1 set filter to qudate=mmfactdate.or.status\$ filtstat0 goto top select 10
749 750 751 752 753 754		<pre>if.not.goodfin set typeahead to 20 goodfin=.t. store (system("FACTDATE")-1) to mmfactdate select 1 set filter to qudate=mmfactdate.or.status\$ filtstat0 goto top select 10 replace goodsery with .f.</pre>
749 750 751 752 753 754 755		<pre>if.not.goodfin set typeahead to 20 goodfin=.t. store (system("FACTDATE")-1) to mmfactdate select 1 set filter to qudate=mmfactdate.or.status\$ filtstat0 goto top select 10 replace goodserv with .f. unlock</pre>
749 750 751 752 753 754 755 756		if .not.goodfin set typeahead to 20 goodfin=.t. store (system("FACTDATE")-1) to mmfactdate select 1 set filter to qudate=mmfactdate.or.status \$ filtstat0 goto top select 10 replace goodserv with .f. unlock select 1
749 750 751 752 753 754 755 756 757		<pre>if.not.goodfin set typeahead to 20 goodfin=.t. store (system("FACTDATE")-1) to mmfactdate select 1 set filter to qudate=mmfactdate.or.status\$ filtstat0 goto top select 10 replace goodserv with .f. unlock select 1 pagepaint=.t. replace goodserve is a select 1 </pre>
749 750 751 752 753 754 755 756 757 758 759		<pre>if .not.goodfin set typeahead to 20 goodfin=.t. store (system("FACTDATE")-1) to mmfactdate select 1 set filter to qudate=mmfactdate.or.status\$ filtstat0 goto top select 10 replace goodserv with .f. unlock select 1 pagepaint=.t. newenq=.t.</pre>
749 750 751 752 753 754 755 756 757 758 759 760		<pre>if .not.goodfin set typeahead to 20 goodfin=.t. store (system("FACTDATE")-1) to mmfactdate select 1 set filter to qudate=mmfactdate.or.status \$ filtstat0 goto top select 10 replace goodserv with .f. unlock select 1 pagepaint=.t. newenq=.t. exit</pre>
749 750 751 752 753 754 755 756 757 758 759 760 760	v	<pre>if .not.goodfin set typeahead to 20 goodfin=.t. store (system("FACTDATE")-1) to mmfactdate select 1 set filter to qudate=mmfactdate.or.status \$ filtstat0 goto top select 10 replace goodserv with .f. unlock select 1 pagepaint=.t. newenq=.t. exit</pre>
749 750 751 752 753 754 755 756 757 758 759 760 761 762 763		<pre>if .not.goodfin set typeahead to 20 goodfin=.t. store (system("FACTDATE")-1) to mmfactdate select 1 set filter to qudate=mmfactdate.or.status\$ filtstat0 goto top select 10 replace goodserv with .f. unlock select 1 pagepaint=.t. newenq=.t. endif waittime=system("waittime") xx=system("newengree") </pre>
749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764		<pre>if.not.goodfin set typeahead to 20 goodfin=.t. store (system("FACTDATE")-1) to mmfactdate select 1 set filter to qudate=mmfactdate.or.status \$ filtstat0 goto top select 10 replace goodserv with .f. unlock select 1 pagepaint=.t. newenq=.t. exit </pre>

1.1.1

1

1.00

1. 2. 3.

1. 1. 1. 1.

",status off

1.

1.



This procedure calculates the possible statuses Procedure: SAL_AREA1 to filter the <SALES > datafile by. 850 * * * * * * 851 852 853 854 855 Called by: FILT_STAT (procedure in SAL6.PRG) *İ Calls: SAL STAT (procedure in SAL6.PRG) 856 *j 857 procedure sal areal 858 859 public status0, status1, status2, status3, status4 860 861 status0 = "ENQU" status1 = "ENQU" + "QUOT" + "FIRM" 862 863 status2="LOST" 864 865 status3 = "FIRM" select 1 866 do sal_stat 867 return 868 869 ******* *********** ***** This procedure calculates the filter of the <SALES> Procedure: SAL_CODE datafile based on the selected article code. 870 *1 871 * * * * 872 873 874 Called by: SAL_AREA0 (procedure in SAL6.PRG) 875 * Calls: STR() (FOXBASE+ function) 876 * : GETKEY (procedure in SAL6.PRG) 877 *1 878 *1* ***** 879 880 procedure sal code 881 parameter numbcode private n,nn,codekey,phrase n=0 882 883 884 phrase = "" codekey="" 885 n=str(n,1) set color to &promptatr do while n<numbcode phrase=phrase+" "+nn+"-"+code&nn 886 887 888 889 890 codekey=codekey+nn 891 n=n+1892 nn = str(n, 1)893 =enddo @ rowprompt-1,0 clear
 @ rowprompt-1,0 say promptbar
 @ rowprompt,0 clear
 @ rowprompt,0 say "SET CODE:"+phrase 894 895 896 897 898 do getkey with codchoice, codekey + returnkey 899 do case 900 =case codchoice = returnkey 901 return case codchoice \$ codekey filtcode=code&codchoice 902 902 903 904 endcase 905 return 906 907 908 909 ***** *] This procedure calculates the filter of the <SALES> Procedure: SAL_STAT based on the selected status. 910 *! 911 * Called by: SAL_AREA1 (procedure in SAL6.PRG) 912 *! 913 * Calls: GETKEY (procedure in SAL6.PRG) 914 915 916 917 * + |+ ******* ****** procedure sal stat 918 set color to &promptatr @ rowprompt-1,0 clear @ rowprompt-1,0 say promptbar 919 920 @ rowprompt,0 clear @ rowprompt,0 clear @ rowprompt,0 say "SET STATUS: 0-ENQU+QUOT 1-ENQU+QUOT+FIRM 2-LOST 3-FIRM " do getkey with ordchoice, "0123"+ returnkey 921 922 923 924 =do case 925 =case ordchoice = returnkey 926 < =return 927 928 =case ordchoice = "0" filtstat=status&ordchoice 929 case ordchoice = "1" 930 931 932 filtstat = status&ordchoice case ordchoice = "2" filtstat=status&ordchoice case ordchoice = "3" 933 934 filtstat=status&ordchoice

935 936 endcase return 937 938 939 +1 This procedure filters the <SALES > datafile based Procedure: FILT_CODE on the selected article code. *i 940 *į 941 942 943 944 945 945 946 947 948 949 * * * * Called by: SAL BROW (procedure in SAL6.PRG) Calis: SAL_AREA0 : EOF() ((: RECNO() 0 (procedure in SAL6.PRG) (FOXBASE+ function) (FOXBASE+ function) *j * procedure filt_code 950 951 952 953 954 ***** do sal_area0 -if codchoice = "0" set filter to wait < = waittime.and.code < > filtcode.and.status \$ filtstat -else 955 set filter to wait < = waittime.and.code = filtcode.and.status\$ filtstat 956 957 958 959 960 -endif goto top -if eof() set color to &promptatr @ rowprompt,0 clear @ rowprompt,0 say "No matching records." 961 962 wait 963 964 @ rowprompt,0 clear set filter to wait < = waittime.and.status \$ filtstat0 965 goto top 966 967 endif recnumtop == recno() firstpos=qudeliver pagepaint=.t. 968 969 970 set color to &screenatr 971 972 973 974 975 clear return This procedure filters the <SALES > datafile based Procedure: FILT_STAT on the selected status. *1 976 977 978 979 980 *! *! *! Called by: SAL BROW (procedure in SAL6.PRG) *! 1 (procedure in SAL6.PRG) (FOXBASE+ function) (FOXBASE+ function) Calls: SAL_AREA1 981 *1 : EOFO 982 * : RECNO() 983 * *1 984 ****** ***** ****** procedure filt stat 985 986 987 do sal areal 988 -if codchoice = "0" 989 set filter to wait < = waittime.and.code < > filtcode.and.status \$ filtstat 990 991 992 993 994 else set filter to wait < = waittime.and.code = filtcode.and.status \$ filtstat -endif goto top -if eof() 995 set color to &promptatr @ rowprompt,0 clear 996 @ rowprompt,0 say "No matching records." 997 998 wait 999 @ rowprompt,0 clear 1000 set filter to wait < = waittime.and.status \$ filtstat0 goto top endif 1001 1002 1003 recnumtop = recno() firstpos=qudeliver pagepaint=.t. 1004 1005 1006 1007 set color to &screenatr clear 1008 1009 return 1010 *! 1011 *! 1012 *! 1013 *! This procedure manages all the report generations Procedure: SALREPO for the Sales department. 1015 *! 1014 *! 1015 *! 1016 *! 1017 *! 1018 *! Called by: CHEKSTOP (procedure in SAL6.PRG) : PRT (procedure in SAL6.PRG) Calls: EOF() (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) : INTO : VALO 1019 *!

1020 *! 1021 *! : SUBSTR() (FOXBASE+ function) : DTOCO : SALAVE (FOXBASE+ function) 1022 *! (procedure in SAL6.PRG) * 1023 1024 : SALFMT (procedure in SAL6.PRG) *1 1025 ******* 1026 procedure salrepo 1027 store 0 to n,mave_enqu,mave_quot,mper_quot,mave_firm,; mper_firm,movr_delti,mper_ontim,mave_qpric,mper_qpric,; mper_qdeli,mper_fpric,mper_fdeli,mdeliverie,mtodeliv,mvolume select 3 1028 1029 1030 1031 1032 1033 goto top 1034 do while .not. eof() 1035 actcode=code 1036 1037 mdescript=descript do while code=actcode 1038 actweek=int(val(substr(dtoc(date),1,2))/5) 1039 do while int(val(substr(dtoc(date), 1, 2))/5) = actweek.and.code = actcode 1040 n=n+11040 1041 1042 1043 1044 do salave -if .not. eof() skip -else 1045 =exit 1046 н -endif 1047 ≠nddo 1048 skip -1 1049 1050 1051 do salfmt skip store 0 to n,mave_enqu,mave_quot,mper_quot,mave_firm,; mper_firm,movr_delti,mper_ontim,mave_qpric,mper_qpric,; mper_qdeli,mper_fpric,mper_fdeli,mdeliverie,mtodeliv,mvolume 1052 1053 1054 enddo 1055 enddo 1056 1057 1058 signal=.f. return 1059 1060 *! This procedure calculates the average of the <SALES> Procedure: SALAVE datafile's key fields. 1061 1062 1063 1064 1065 *! *! *! Called by: SALREPO (procedure in SAL6.PRG) *į *i 1066 procedure salave 1067 1068 mave_enqu=(mave_enqu+ave_enqu)/n 1069 mave_quot=(mave_quot+ave_quot)/n mave_quot=(mave_quot+ave_quot)/n mper_quot=(mper_quot+per_quot)/n mave_firm=(mave_firm+ave_firm)/n movr_delti=(movr_delti+ovr_deltim)/n movr_delti=(movr_delti+ovr_deltim)/n mave_qpric=(mave_qpric+ave_qprice)/n mave_qpric=(mave_qpric+ave_qprice)/n 1070 1071 1072 1073 1074 1075 mper_qpric=(mper_qpric+ave_qprice)/n mper_qdeli=(mper_qdeli+per_qdeliv)/n mper_fpric=(mper_fpric+per_fprice)/n mper_fdeli=(mper_fdeli+per_fdeliv)/n mdeliverie=(mdeliverie+deliveries)/n 1076 1077 1078 1079 1080 1081 mtodeliv = (mtodeliv + todeliv)/n 1082 mvolume=(mvolume+volume)/n 1083 return 1084 1085 *!*** ***** 1086 1087 *] This procedure displays the reports concerning the + Procedure: SALFMT Sales department's performance. 1088 *! 1089 *! Called by: SALREPO (procedure in SAL6.PRG) 1090 *1 * 1091 Calls: INKEY() (FOXBASE+ function) 1092 *j+ 1093 1094 1095 procedure salfmt 1096 set color to 1097 clear 1098 set color to r/n set color to 1/n @ 1,24 say " @ 2,24 say " @ 3,24 say " @ 4,24 say " @ 7, 0 say "Week" @ 7,15 say actweek picture "99" 1099 SALES DEPARTMENT'S PAST || " PERFORMANCE (Page 1) || " 1100 1101 1102 1103 1104

7,20 say "Date" 7,30 say date 9, 0 say "Code" 9,13 say actcode 9,20 say "Descript" 1107 1108 ē ē 9,20 say "Descript" 9,30 say mdescript 12,27 say "DAY OVERALL" 13, 0 say "Number of enquires" 13,25 say ave_enqu picture "999.99" 13,34 say mave_enqu picture "999.99" 13,65 say "DAY OVERALL" 14,0 say "Number of quotations" 14,25 say ave_quot picture "999.99" 14,34 say mave_quot picture "999.99" 14,34 say may "% over enquires" 14,63 say per_quot picture "999.99" 14,72 say mper_quot picture "999.99" ĕ õ ã õ õ õ ě õ 14,03 say per_quot picture "999.99" 14,72 say mper_quot picture "999.99" 15,0 say "Number of sales orders" 15,25 say ave firm picture "999.99" 15,34 say mave_firm picture "999.99" 15,43 say "% over quotations" 15,63 say per_firm picture "999.99" 15,72 say mper_firm picture "999.99" õ 1125 õ 1136 1141 1142 1143 1145 ? inkey(3) set color to 1149 1150 clear set color to r/n @ 4,23 say @ 5,23 say . SALES DEPARTMENT'S PAST PERFORMANCE (Page 2) 1 " . ě 6,23 say 7,23 say ĕ @ 7,23 say " @ 10, 2 say "Week" @ 10,17 say actweek picture "99" @ 10,22 say "Date" @ 10,32 say date @ 12, 2 say "Code" @ 12,22 say "Descript" @ 12,22 say mbescript 1157 @ 12,22 say "Descript"
@ 12,32 say mdescript
@ 14, 2 say "Quoted Enquires' Success % :"
@ 14,45 say "Achieved orders success % :"
@ 15,22 say "DAY"
@ 15,32 say "OVERALL"
@ 15,70 say "OVERALL"
@ 16,3 say "Price"
@ 16,3 say "Price" 1164 ā 16,21 say per_qprice picture "999.99" (@ 16,21 say per_qprice picture "999,99",
(@ 16,33 say mper_qpric picture "999,99",
(@ 16,44 say "Price")
(@ 16,62 say per_fprice picture "999,99",
(@ 17,3 say mper_fprice picture "999,99",
(@ 17,21 say per_qdeliv picture "999,99",
(@ 17,33 say mper_qdeli picture "999,99",
(@ 17,44 say "Delivery Date",
(@ 17,44 say "Delivery Date",
(@ 17,44 say "Delivery Date",
(@ 17,44 say "Delivery Date",
(@ 17,44 say "Delivery Date", 1172 1173 @ 17,62 say per_fdeliv picture "999.99" @ 17,71 say mper_fdeli picture "999.99" 1180 ? inkey(3) set color to clear return 1187 *! This procedure allows the user to select the reports' *! Procedure: PRT output device (screen or printer).

*

1190	*1	
1191	*1	Called by: CHEKSTOP (procedure in SAL6.PRG)
1193	*1	Calls; SYS() (FOXBASE+ function)
1194	*1	: SALREPO (procedure in SAL6.PRG)
1195	*!	***************************************
1197		procedure prt
1198		*********
1199		et color to
1201	- 1	clear
1202		m menu=0
1203		set color to $bu + h$
1205		a 6,24 to 8,54 double
1206		set color to br+/n
1207		$(g_{\ell})_{\ell,2}$ say relation of the relation set color to but $/n$
1209		@ 10,24,14,50 box "
1210		set color to bg +/n
1212		@ 11,22 prompt "2.PAST BEHAVIOUR."
1213		@ 13,29 prompt "3.EXIT. "
1214		set color to bu +/n @ 16 - 29 say "DEVICE :"
1216		?? sys(101)
1217		menu to m_menu
1219		==uo case ==case m menu = 1
1220		set device to print
1221		memorase m_menu=2
1223		set device to screen
1224		set color to
1225	ĭ	\rightarrow
1227		set device to screen
1228		set color to
1230	Ĭ	
1231		-enddo
1232		return *
1234	*!*	***************************************
1235	*!	This procedure displays the enquires allowing also
1237	*	thoroughly explained in one of the thesis's sections.
1238	*!	Called by: INPTQUO (procedure in SAL6.PRG)
1239	- TI ++	Calls: CHRA (EOXRASE+ function)
1241	*!	: SYSTEM() (function in SAL6.PRG)
1242	*!	: EOFO (FOXBASE + function)
1243	*!	: $XX <> SPACEO (FOXBASE + function)$
1245	*1	: POS (procedure in SAL6.PRG)
1246	*]	: SAL DISP (procedure in SAL5.PRG)
1248	*!	: GETKEY (procedure in SAL6.PRG)
1249	*!	: BOFO (FOXBASE+ function)
1250	+1	: SAL_BDII (procedure in SALD.PKG) : FILT STAT (modedure in SAL6.PRG)
1252	*!	: FILT_CODE (procedure in SAL6.PRG)
1253	+1	: NOT.SYSTEM() (FOXBASE+ function)
1255	*!	: MARKING (procedure in SAL6.PRG)
1256	*!	
1257	-1-	nmeedure sal brow
1259		************
1260		private parcol, panmax, panlast, recnumlop, recnumlast, skiprecs
1262		private row, rowtop, keystrokes, rowlast, pagepaint
1263		*Initialize constants.
1264		none = cn(1) endkey = chr(6)
1266		uparrow = $chr(5)$
1267		downarrow = $chr(24)$
1268		rightarrow = chr(4)
1270		keystrokes = "EACL" + uparrow + downarrow + home + leftarrow +;
1271		rightarrow+endkey+pgdn+pgup+returnkey rowton = 1
1273		skiprecs = rowbottom - rowtop + 1
1274		scrr≖.f.

24
24
、意
14
16
. >
~ 25
7.0
4.2
E.
19
20
· .
25
1
1.42
1.1
1
E be
1
-
· · · ·
5.6
d.P
10
1
1
8
A.C.
5
1
18
1.0
7
1
大學
· si
1.65.7
14 14 14
41
194
124
1.3
17
1.12
1
6.5
· af
3
St
24
4.5
13.
. 91
500
10
44
47
. 57.
- 22
3
30
10
1
eres and
5
1.1
1
15
.9
7
17
2
1
A set of a
and the second second second second second second second second second second second second second second secon
and the second second second second second second
and the second second second second second second second second second second second second second second secon
and the second second second second second second second second second second second second second second secon
ىدىنىيەرىمەيلىك بىرى بىلىمەتلەرلىك ئىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىك بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى بىلىكى
ى ئى ئىرىمىدىنى بىرىمىيىتى بىرىمىيىلىغى بىرىمىيىلىغى بىرىمىيىلىغى بىرى بىرىمىيىلىغى بىرى بىرى بىرى بىرى بىرى بى بىرى بىرى
a the state of the
tatan da tanan sebagai ang sang sang sang sang sang sang sang
ir version in the state of the second state of the second second second second second second second second seco
de se estado de la consectión de la consectión de la consectión de la consectión de la
ىلىدىغىنىيە بىرىمىدىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرى بىرىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرىمىيە بىرى
al de la companya de la companya de la companya de la companya de la companya de la companya de la companya de La companya de la companya de la companya de la companya de la companya de la companya de la companya de la comp
de Astronomica de la construcción de la construcción de la construcción de la construcción de la
a de de acteur de la constant de la conservation en constant de la constant de la constant de la constant de la
te medicie des constants de la constant de la constant de la constant de la constant de la constant de la cons La constant de la constant de la constant de la constant de la constant de la constant de la constant de la con
ando zaskie dotani i statu i da sana sana sa sa sa sa sa sa sa sa sa sa sa sa sa
ىلىغىنى بىرىمىيى بىرىمىيىلىغىنىيىلىغىنىيىلىغىنىيىلىغىنىيىلىغىنىيىلىغىنىيىلىغىنىيىلىغى بىرىمىيىلىغى بىرىكى بىرى ئىمىيىلان بىرىغۇنغۇ بىرلارىدىغۇرىيە بىرىمەرىكەرىكەركى ئىرىكىدە بىرىكى بىرىمىيەتلەرلارىكى قەڭ بەرسىيەت بىرى بىرى
a statistica sector de la sector de la sector de la sector de la sector de la sector de la sector de la sector

ALC: NO

1275	waittime=system("waittime")
1276	valume – system (valume)
1277	if eofO
1278	do while .t.
1279	rest not serr
1281	as 20 to 13 54 double
1282	set color to gr+/n
1283	@ 10,29 say "NO RECORDS AVAILABLE"
1284	@ 11,34 say "PLEASE WAIT"
1285	sorr=1.
1287	waittime=system("waittime")
1288	goto top
1289	recnum=recno()
1290	xx = system('newengrec'')
1292	recnum=recno()
1293	do pos with xx, lastpos, firstpos
1294	endif
1295	
1297	v exit
1298	endif
1299	enddo
1300	*Initialize local variables.
1302	goto top
1303	row = rowtop
1304	recnum = recno()
1306	firstos=audeliver
1307	pagepaint = .t.
1308	pancol = 1
1310	panuas = 1 panuas = 1
1311	*Perform BROWSE.
1312	set color to &screenatr
1313	clear * The following loop is positive "DEDEAT/UNITH cound > "
1315	metonowing loop is really a REPERTION THE Cond > ,
1316	if pagepaint
1317	goto top
1319	
1320	row = rowtop
1321	endif
1322	panlast = pancol
1324	waitime = system("waitime")
1325	newenq=.f.
1326	goto top
1327	=
1329	bot=.f.
1330	do while recno() <> browrec.andnot.bot
1331	skip
1333	goto bottom
1334	bot=.t.
1335	else
1336	row=row+1
1338	enddo
1339	endif
1340	endif
1341	recnum = recno()
1343	do sal disp with (rowtop), skiprecs
1344	goto recnum
1345	pagepaint = .f.
1340	Drowrec = 0 —endif
1348	set color to &promptatr
1349	@ rowprompt-1,0 say promptbar
1350	(Comprompt, U say ; "BROWSE: (F)vit (A)coapt (C)oda fill tan (Amount) (D-D-) (D-T-) (D-T-)
1352	set color to bu +, n
1353	@ rowprompt+1,18 say "CURRENT SELECTION: CODE="
1354	7? filtcode
1355	2? filtstat
1357	set color to &promptatr
1358	@ row,0 say chr(16)
1322	(Q row, // say cnr(1/)

5 5 . 5 4



Salter and Miller & Construction

1445 1446 1447 1448 1449 1450 goto top endif recnumtop = recno() firstpos=qudeliver pagepaint = .t. endif 1451 case choice = "L" 1452 do filt stat 1452 1453 1454 1455 1456 1456 ase choice = "C" do filt_code case choice = home pagepaint = (pancol <> 1) pancol = 1 1458 case choice = leftarrow 1459 if pancol > 11460 pancol = pancol - 1 pagepaint = .t. endif 1461 1462 1462 1463 1464 case choice = rightarrow -if pancol < panmax pancol = pancol + 1 1465 pagepaint = .t.endif 1466 1467 case choice = endkey pagepaint = (pancol <> panmax) 1468 1469 1470 1471 pancol = panmax case choice = "A" -if .not.system("SYS_STOP").and.move _____if status="ENQU" 1472 1473 1473 1474 1475 1476 1477 1478 do accept do marking do sal disp with row,1 set color to &promptatr @ row,0 say chr(16) @ row,77 say chr(17) 1479 1480 1481 1482 1483 goto recnum endif -endif 1484 endcase 1485 enddo 1486 set filter to wait < = waittime.and.status \$ filtstat0 1487 goto top 1488 1489 1490 *! return 1491 *! This procedure obtains the most updated values of 1492 *! Function: SYSTEM() the fields of the <SYSTEM > datafile. 1492 *! 1493 *! 1494 *! 1495 *! 1496 *! 1497 *! Called by: SAL6.PRG : CHEKSTOP : SAL_FORM : GETKEY (procedure in SAL6.PRG) (procedure in SAL6.PRG) (procedure in SAL6.PRG) 1498 *1 : SAL BROW (procedure in SAL6.PRG) 1499 *i 1500 * Calls: STR() (FOXBASE+ function) 1501 * : SELECTO (FOXBASE+ function) procedure system 1507 store str(select(),2) to base 1508 select 10 1509 1510 goto top system=&fieldd 1511 1512 1513 select &base return system 1515 *! 1515 *! This procedure creates the "view" used during the 1515 *1 1516 *1 1517 *1 1518 *1 1519 *1 1520 *1 Procedure: OPEN run of the Sales program. Called by: SAL6.PRG Uses: MARKET, DBF 1521 1522 1523 * : BILL.DBF *į : SALPAST.DBF * 1524 * Indexes: DEMARK.IDX 1525 : BILLNA.IDX : CDSAL.IDX *j 1526 1527 * 1528 1529 procedure open

1530 ***** 1530 1531 1532 1533 1534 1535 set exclusive off select 1 use market index demark set filter to wait < = waittime.and.status \$ filtstat0 select 2 1536 use bill index billna 1537 select 3 1538 1539 1540 use salpast index cdsal set safety off create view saldep from environment all 1541 1542 set safety on close databases
 1542
 close databases

 1543
 **

 1544
 **

 1545
 *!

 1546
 *!

 1546
 *!

 1547
 *!
 ********** This procedure checks the screen position of the Procedure: POS ewly arrived enquire. 1547 *! 1548 *! 1549 *! 1550 *! 1551 *! 1552 *! 1553 *! Called by: GETKEY (procedure in SAL6.PRG) : SAL_BROW (procedure in SAL6.PRG) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) (FOXBASE+ function) Calls: VAL() : SUBSTR() : CTOD() : .OR.() : MIN() 1554 +į 1555 * 1556 * : SPAČE() (FOXBASE+ function) 1557 * 1558 *!* ***** 1559 1560 1561 procedure pos parameter yy,lpos,fpos rpos1=iif(substr(yy,1,3)=" rpos2=iif(substr(yy,4,3)=" rpos3=iif(substr(yy,7,3)=" ",0,val(substr(yy,1,3))) ",0,val(substr(yy,4,3))) ",0,val(substr(yy,7,3))) 1562 1563 1564 1565 -if rposl < >0 1566 1567 1568 goto rpos1 pos1 = qudeliver else 1569 pos1=ctod("01/01/99") 1570 endif 1571 if rpos2 <>0 1572 1573 1574 1575 goto rpos2 pos2 = qudeliverelse pos2=ctod("01/01/99") 1576 1577 endif -if rpos3 < >0 1578 goto rpos3 1579 pos3 = qudeliver 1580 else 1581 1582 pos3=ctod("01/01/99") endif 1583 -if pos1 < > ctod("01/01/99").and.((pos1 > = fpos.and.pos1 < = lpos).or.(lastdisp < 20)) 1584 pagepaint =.t. 1585 ndif 1586 1587 1588 1589 -if pos2 < > ctod("01/01/99").and.((pos2 > = fpos.and.pos2 < = 1pos).or.(lastdisp < 20)) pagepaint=.t. endif if pos3 < > ctod("01/01/99").and.((pos3 > = fpos.and.pos3 < = lpos).or.(lastdisp < 20)) pagepaint = .t. endif 1590 1591 1592 minpos=min(pos1,pos2) 1593 minpos=min(minpos,pos3) 1594 1595 1596 1597 if pos1 = minpos browrec=rpos1 else -if pos2=minpos 1598 browrec=rpos2 1599 -else 1600 browrec = rpos3 endif 1601 endif 1602 1603 1604 1605 select 10 replace newengree with space(9) unlock 1606 select 1 -if pagepaint newenq = .t. 1607 1608 1609 endif 1610 if .not.neweng 1611 goto recnum 1612 1613 endif return 1614

1615	* *************************************
1616	*! This procedure checks the acceptance or not
1617	*! Procedure: ACCEPT of the input done.
1618	*
1619	* Called by: SAL, BROW (procedure in SAL 6 PRG)
1620	
1621	Calle: FOFO (FOXBASE + function)
1622	
1622	******
1604	
1024	procedure accept
1025	
1626	store j-> factdate to mqudate
1627	store j-> facttime to mqutime
1628	if tempdd1 > tempdd
1629	store $(j > factdate-1)$ to mqudate
1630	endif
1631	if .not. cof()
1632	*Replace only if there is an available record
1633	replace gudate with moudate
1634	replace dutime with moutime
1635	unlock
1636	
1637	return
1620	
1022	"I EUF: SALUACI

System: SAL6.PRG Author: JUAN IGNACIO IGARTUA 08/05/91 18:32:05 Tree Diagram

....

CATE	DDC
	ING VSTEV DE (databasa)
1 *	EBDODO (detablase)
	ERROR() (lunction in 7)
	MESSAGE() (function in ?)
	ERR_FIX (procedure in SAL6.PRG)
	RLOCK() (function in ?)
	CTOD() (function in ?)
	REPLICATE() (function in ?)
	CHR() (function in ?)
	SYSTEMO (function in SAI 6 PPG)
	STD (Guatian in SALO.1 RG)
	STRUCHURCHON IN ()
	SELECTO (runction in 7)
	OPEN (procedure in SAL6.PRG)
	MARKET.DBF (database)
	BILL.DBF (database)
	SALPAST DBF (database)
	SELECT() (function in ?)
	CHEVSTOR (Introduce in SAI 6 PDG)
	SISIEM.DEF (database)
	SYSTEMU (function in SALO.PRG)
	STR() (function in ?)
	SELECT() (function in ?)
	ANDNOT.SYSTEM() (function in ?)
	SALREPO (procedure in SAL6.PRG)
	EOFO (function in ?)
	INTO (function in ?)
	VALO (Interior in 1)
	VALU (Idiction in t)
	SUBSTR() (function in 7)
	DIOCO (function in 7)
	SALAVE (procedure in SAL6.PRG)
	SALFMT (procedure in SAL6.PRG)
	INKEY() (function in ?)
	PRT (procedure in SAL6, PRG)
	SYSO (function in ?)
	SAUPERO (Inmedure in SAUE DDG)
	TOPO (files in 2)
	EUFO (function in ?)
	INTO (function in 7)
	VALO (function in ?)
	SUBSTR() (function in ?)
	DTOC() (function in ?)
	SALAVE (procedure in SAL6.PRG)
	SALEMT (procedure in SAL6 PRG)
	INKEY() (function in 2)
	AND NOT SHIFT AND NOT SYSTEM (Association in 2)
	NOT SYSTEMO (Antoine in 9)
	AND.SYSTEMU (function in ?)
	STRO (function in ?)
	——MAINMENU (procedure in SAL6.PRG)
- F	SELECT() (function in ?)
-	CHEKSTOP (procedure in SAL6, PRG)
	SYSTEM.DBF (database)
1	SYSTEM() (function in SALG DDG)
	STDA (function in 2)
	SELECTO (function in ()
	SELECT((unction in /)
	ANDNUT.SYSTEM() (function in ?)
	SALREPO (procedure in SAL6.PRG)
	EOF() (function in ?)
	INT() (function in ?)
	SUBSTR() (function in 2)
	DTOCO (function in 2)
	SALAVE (monodum in CALA DDC)
	CALENT (procedure in SALO.PKO)
	SALEMI (procedure in SALO.PKG)
	INKEY() (function in ?)
	PRT (procedure in SAL6.PRG)
	SYS() (function in ?)
	SALREPO (procedure in SAL6.PRG)
	EOF() (function in ?)
	INTO (function in ?)
	VALO (function in 2)
	SUBSTDA (function in 7)
	DIOC() (nunction in ?)
	SALAVE (procedure in SAL6, PRG)
	SALFMT (procedure in SAL6.PRG)
	INKEY() (function in ?)
	.NOT.SYSTEM() (function in ?)

AND.SYSTEM() (function in ?)	
STR() (function in ?)	
CHRO (function in ?)	
STR() (function in ?)	
EOEO (function in ?)	
RECNO() (function in ?)	
POS (procedure in SAL6.PRG)	
SUBSTR() (function in ?)	
CTOD() (function in ?)	
OR.() (function in ?)	
SAL DISP (procedure in SAL6.PRG)	
ROW() (function in ?)	
EOFO (function in ?)	
FIQ. (function in ?)	
GETKEY (procedure in SAL6.PRG)	
SYSTEM() (function in SAL6.PRG)	
SELECTED (function in ?)	
SUBSTRO (function in ?)	
RECNO() (function in ?)	
POS (procedure in SAL6.PRG)	
VAL() (function in ?)	
CTOD((function in ?)	
.OR.() (function in ?)	
SPACE() (function in ?)	
INKEY((function in ?)	
SELECT() (function in ?)	
SYSTEM DBF (database)	
STRO (function in ?)	
SELECT() (function in ?)	
SALREPO (procedure in SAL6 PRG)	
EOF() (function in ?)	
VAL() (function in ?)	
DTOCO (function in ?)	
SALAVE (procedure in SAL6.PRG)	
SALFMT (procedure in SAL6.PRG)	
INKEY() (function in ?)	
SYSO (function in ?)	
SALREPO (procedure in SAL6.PRG)	
EOFO (function in ?)	
INT() (function in ?)	
SUBSTR() (function in ?)	
DTOC() (function in ?)	
SALAVE (procedure in SAL6.PRG)	
SALFMI (procedure in SAL0.PKG)	
	in ?)
.NOT.SYSTEM() (function in ?)	
.AND.SYSTEM() (function in ?)	
=STR() (function in 7)	
SAL EDIT (procedure in SAL6.PRG)	
RLOCK() (function in ?)	
SAL FORM (procedure in SAL6.PRG)	
STR() (function in SALO.PKG)	
SELECT() (function in ?)	
SAL_STOR (procedure in SAL6.PRG)	
SAL SAYS (procedure in SAL6.PRG)	
SYSTEM() (function in SAL6 PPG)	
STR() (function in ?)	
SELECT() (function in ?)	
SUBSTR() (function in ?)	

1 54

2

1. 10

1	RECNO() (function in ?)
	POS (procedure in SAL6.PRG)
-	SUBSTR() (function in ?)
	CTOD() (function in ?)
	UF() (function in ?)
	CHR() (function in ?)
	SELECT() (function in ?)
	SYSTEM.DBF (database)
	SYSTEM() (function in SAL6.PRG)
	SIRU (Junction in ?) SELECT() (function in ?)
	ANDNOT.SYSTEM() (function in ?)
	EOF() (function in ?)
	INT() (function in ?)
	SUBSTRO (function in ?)
	DTOC() (function in ?)
	SALFMT (procedure in SAL6.PRG)
	PRT (procedure in SAL6.PRG)
	SYS() (function in ?)
	EOF() (function in ?)
	INTO (function in ?)
	SUBSTR() (function in ?)
	DTOC() (function in ?)
	SALAVE (procedure in SALL.PRG)
	INKEY() (function in ?)
	.NOT.SYSTEM() (function in ?)
	AND.SYSTEM() (function in ?)
	.ANDNOT.SYSTEM() (function in ?)
	SAYLINE (procedure in SAL6.PRG)
	SAL_REPL (procedure in SAL6.PRG)
	MARKING (procedure in SAL6.PRG)
	QUPRICE <> 0. AND. QUDELIVER <> CTOD() (function in ?)
	SAL_AREA1 (procedure in SAL6.PRG)
	GETKEY (procedure in SAL6.PRG)
	SYSTEM() (function in SAL6.PRG)
	STR() (function in ?)
	SUBSTR() (function in ?)
	RECNO() (function in ?)
	VAL() (function in ?)
	CTOD() (function in ?)
	OR.() (function in ?)
	SPACE() (function in ?)
	UPPER() (function in ?)
	CHEKSTOP (procedure in SAL6.PRG)
	SYSTEM.DBF (database) SYSTEM() (function in SAL6 PRG)
	STRO (function in ?)
	SELECT() (function in ?)
	SALREPO (procedure in SAL6.PRG)
	EOF() (function in ?)
	VALO (function in ?)
	DTOC() (function in ?)
	SALEMT (procedure in SAL6.PRG)

314

AS A CONTRACT


RECNOO (function in ?)
POS (myadura in SAL6 PPG)
VALU (function in 7)
SUBSTR() (function in ?)
CTOD() (function in ?)
OR (function in ?)
MINO (function in 2)
SPACED (function in 2)
STACED (Indicition in ?)
IFO (function in ?)
INKEY() (function in ?)
UPPER() (function in ?)
CHRO (function in ?)
SELECTO (function in ?)
CHEKSTOP (amadure in SAL6 PPG)
SISIEM.DEF (database)
STSTEMU (function in SALO.PRG)
STR() (function in ?)
SELECT() (function in ?)
SALREPO (procedure in SAL6.PRG)
EOF() (function in ?)
INTO (function in ?)
VALO (function in 2)
SUBSTRACTION IN 17
DTOCO (function in ?)
DIOCO (function in /)
SALAVE (procedure in SAL6.PRG)
SALFMT (procedure in SAL6.PRG)
INKEY() (function in ?)
PRT (procedure in SAL6.PRG)
SYSO (function in ?)
SAUPERO (procedure in SAU6 PPG)
INTO (runchon in 7)
VALO (function in 7)
SUBSTR() (function in ?)
DTOC() (function in ?)
SALAVE (procedure in SAL6.PRG)
SALFMT (procedure in SAL6.PRG)
INKEY() (function in 2)
AND NOT SHITT AND NOT SYSTEMO (function in 2)
NOT SVSTEMA (function in 2)
AND OVOTELAO (function in /)
AND SYSTEM() (function in ?)
STR() (function in ?)
———FOUND() (function in ?)

System: SAL6.PRG Author: JUAN IGNACIO IGARTUA 08/05/91 18:31:58 Database Structure Summary 4 databases in the system SYSTEM.DBF MARKET.DBF BILL.DBF SALPAST.DBF Structure for database : SYSTEM.DBF Number of data records : 1 Last updated : 07/01/91 at Field Field name Type Width 9:17 Last upo Field name STARTDATE FACTDATE FACTTIME SYS STOP GAM-STOP HOUR RATE TOT OVERH NEWENO Type Date Date Dec Start End 8 19 8601222990923482667888899 12222990923482667888889 12 8 Character Logical Logical ĩ 17 22 23 30 40 41 50 4 ā ī 17 5 2 2 Numeric 6789 10 Numeric NEWENO NEWENOREC FREQUENCY GOODSERV Logical Character Numeric 193114441100 110 10 53 53 55 55 55 55 55 11 12 13 14 Logical Logical Character REINDEX STATRESO STATCODE Character 15 16 17 63 67 68 69 79 WAITTIME Numeric RESTART RESTATUS THROUGHPUT Logical Logical 18 19 Numeric 222 ASSETS Numeric Logical Logical 20 SYSTOCK **8**9 10 21 22 23 SALREPORT ğģ PLAREPORT ī 100 100 ENQUPERIOD Numeric 101 101 Total ** 102 FoxDoc did not find any associated index files Used by: SAL6.PRG : CHEKSTOP (procedure in SAL6.PRG) Structure for database : MARKET.DBF Number of data records : 10 Last updated : 07/01/91 at Field Field name Type Width 1 CODE Character 4 9:17 Dec Start End 1 44842 342640820 780 DESCRIPT ENQUANTIT ENPRICE 20 23 Character 5 25 29 35 Numeric 4 **4** 5 68 2 Numeric ENDATE Date ENTIME ENDELIVER QUPRICE QUDATE 43 47 55 61 Character 6789 486848 Date Numeric 2 Date ÕUTIME ÕUDELIVER STATUS 69 73 1Õ Character 11 12 13 Date 84 92 93 105 Character **8**1 4 8 1 REALDELIV 85 93 94 Date 14 DELIVERED SUCCSPRICE SUCCSDELIV Logical 2 2 Numeric Numeric 66 16 17 100 17 WAIT Total ** 4 110 109 Numeric 106 This database appears to be associated with index file(s): : DEMARK.IDX (QUDELIVER) Used by: OPEN (procedure in SAL6.PRG) Structure for database : BILL.DBF Number of data records : 5 Last updated : 07/01/91 Field Field name Type Wid at 9:09 Width Dec Start End Belgi Banar Dissilati Strain din a sati

Carly Cours Brooks Course

317

They doe a light with the total

- mar hat

1 CODE 2 DESCRIPT 3 MINDELIVDD 4 MAXDELIVDD 5 MAXQUANTIT 6 MINQUANTIT 7 STANDPRICE 8 MATCOST 9 TOT ENQU 10 PRICE LIMT 11 DISCOUNT 12 QUANT DISC 13 DELIV LIMT 14 MRKT PRICE 15 MRKT LIMT 16 ONTIMEDELV 17 ONTIMLIMT 18 STANDHOURS 19 OVERHEAD 20 STAND COST 21 NUM ENQU 22 RESOUR COD 23 PROCESTIME 24 SET UP ** Total ** This database app : BILLNA.I	Character Character Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Character Numeric Character	4 20 2 6 6 6 6 4 6 4 6 6 6 6 6 6 6 6 6 6 6 6	2222 2222 2222 2222 2222 2222 2 2 2 with i	1 5 25 27 29 35 41 47 53 57 63 67 73 77 83 89 95 101 107 113 109 123 127 132	44 246 288 446 556 662 776 888 9006 1062 1118 1226 1118 138 138 (5):
Used by: OPEN	(proced	dure in S	SAL6.PR	G)	
Structure for dat	abase : SALPA	ST.DBF			
Number of data re Last up Field Field name 2 DESCRIPT 3 DATE 4 AVE ENQU 5 AVE OUOT 6 PEROUOT 7 AVE FIRM 9 OVR DELTIM 10 PER ONTIME 11 AVE OPRICE 12 PER OPRICE 13 PER FORLIV 14 PER FPRICE 15 PER FPELIV 16 DELIVERIES 17 VOLUME ** Total **	cords : dated : 07/01, Type Character Date Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric Numeric	/91 at 9 Width 20 8 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 10 119	22222222222222222222222222222222222222	Start 1 25 33 39 45 51 57 63 67 73 79 85 91 97 103 109	End 24 32 38 44 50 562 666 728 84 90 962 108 118
: CDSAL.IDX (CODE+DTOC(DATE))					
Used by: OPEN	(proced	dure in S	SAL6.PR	G)	

125

a the second second second second second second

· 2:

1

A 2.424.4 243

System: SAL6.PRG Author: JUAN IGNACIO IGARTUA 08/05/91 18:31:59 Data Dictionary

Riold Name	(Thum o	Tom	Dea	Databaga
FIEId Name	туре	Len	Dec	Dalabase
ASSETS	N	10	2	SYSTEM.DBF
AVE ENOU	N	6	2	SALDAST DEF
AVE ENOU	14	<u> </u>	4	SALFASI .DDF
AVE FIRM	N	6	2	SALPAST.DBF
AVE OPRICE	N	6	2	SALPAST DBF
AVE OTION	NT NT	č	5	
AVE QUOT	IN	0	2	SALPAST. DBF
CODE	С	4	0	BTLL, DBF
	-	-	•	CATDACE DDE
				SALPAST DDr
				MARKET.DBF
DA ጥ ም	n	Q	Δ	CALDACT DEF
DATE	<u>D</u>	0	U	SALFAST . DDr
DELIVERED	Г	1	0	MARKET DBF
DELIVERTES	N	6	0	SALDAST DBF
	17	, in the second s	š	
DELIV LIMI	N	4	0	BTTT DBL
DESCRTPT	С	20	0	SALPAST, DBF
	-		•	DITI DEF
				DITT.DDL
				MARKET DBF
DTSCOUNT	N	Λ	2	BTTT DEF
DISCOUNT	11	4	4	DTTTTODL
ENDATE	D	8	· 0	MARKET.DBF
ENDELTVER	р	8	0	MARKET DRF
	<i>D</i>	ç	š	
ENPRICE	N	6	2	MARKET DBF
ENOUANTTT	N	4	0	MARKET DRF
ENOUDEDTOD	1		ě	
ENQUPERIOD	N	1	0	SYSTEM.DBF
ENTIME	C	4	0	MARKET, DBF
FACTIDATE	ň	ō	õ	CVCTEN DDE
FACIDATE	D	0	U	SISTEM.DBF
FACTTIME	С	4	0	SYSTEM.DBF
FREQUENCY	N	2	Ō	CVCTEM DEF
TRECOBACT	1	5	<u>v</u>	SISTEM.DDF
GAM STOP	L	1	0	SYSTEM.DBF
COODSERV	Т.	1	0	SVSTEM DBE
	1	÷	Š	
HOUR RATE	N	/	2	SYSTEM.DBF
MATCOST	N	6	2	BTLL, DBF
MAYDELTUDD	M		õ	BTTT DDD
MANDEDIVDD	14	4	U	DILLI.DDL
MAXQUANTIT	N	6	2	BILL.DBF
MINDELTVDD	N	2	õ	BTTT DEF
MINDIDIVDD	11	4	0	DIDD.DBF
MINQUANTIT	N	6	2	BILL.DBF
MRKT LIMT	N	6	2	BTLL DBF
NDVOTDDTOP	Ň	ž	ã	DITT DDE
MRKT PRICE	IN	6	4	BTTT DBL
NEWENO	\mathbf{L}	1	0	SYSTEM, DBF
NEWENODEC	ā	ō	õ	CVCUEM DDE
MEMERQREC	<u> </u>	9	U	SISTEM. DDF
NUM ENQU	N	4	0	BILL.DBF
ONTIMEDELV	N	6	2	BTTT DEF
	11	ý	4	DIDD.DDF
ONTIMLIMT	N	6	2	BTTT'DBL
OVERHEAD	N	6	2	BTLL, DBF
OVD DET UTM	AT AT	ž	ž	
OVR DELITIM	11	4	2	SALPAST.DBr
PER FDELIV	N	6	2	SALPAST.DBF
DEDELETON	M	6	2	CATDACT DEF
	14	0	4	SALFASI.DDF
PER FPRICE	N	6	2	SALPAST.DBF
PERTONTIME	N	6	2	SALDAGT DBE
DED ODEL TU	11	Š	2	OXDEADI.DDF
PER QUELIV	N	6	2	SALPAST.DBF
PERFORME	N	6	2	SALPAST DBF
	ŇŤ	č	5	
FER QUUI	IN	0	4	SALPAST DBr
PLAREPORT	L	1	0	SYSTEM, DBF
DDTOF TIM	N	Ē	5	DITT DDE
	14	0	4	DTTT • DDL
PROCESTIME	N	5	2	BILL.DBF
OUANT DISC	N	6	2	BTLL DBF
AIDAME	1	0	4	DTDD.DDF
QUDATE	D	8	0	MARKET.DBF
OUDELIVER	D	8	0	MARKET DRF
OUDDICE	Ň	ć	2	MADUDE DOD
QUERTCE	IN	0	4	PIAKKET . DBF
QUTIME	С	4	0	MARKET DBF
REALDELTY	D	ø	ñ	MADERT DEF
DETNOEV	Ϋ́	0	v v	MARKEL DDF
REINDEX	Ь	1	0	SYSTEM.DBF
RESOUR COD	С	4	0	BTLL DBF
DECMADI	Ť		č	
REDIARI	با.	4	U	SISTEM.DBF

	RESTATUS SALREPORT SET UP STANDHOURS STANDPRICE STAND COST STARTDATE STATCODE STATRESO STATRESO STATRESO STATRESO STATUS SUCCSDELIV SUCCSPRICE SYSTOCK SYS STOP THROUGHPUT TOT ENOU TOT OVERH VOLUME WAITT WAITTIME	L L C N N N D C C C N N N L N N N N N N	1176668444660110400144 1014401044	000222000022202020200	SYSTEM.DBF SYSTEM.DBF BILL.DBF BILL.DBF BILL.DBF SYSTEM.DBF SYSTEM.DBF MARKET.DBF MARKET.DBF MARKET.DBF SYSTEM.DBF SYSTEM.DBF SYSTEM.DBF BILL.DBF SYSTEM.DBF SALPAST.DBF MARKET.DBF
	WATLITHE	N	4	U	SISTEM.DBF

1999 (B) 199

320

-

APPENDIX D Evaluation of the game

the states in



nenen in sin und auf Alfflühr eine Andere durch die Stellenbergen in der der Alfbergeren eine der Alfbergeren e



記



322

A Sauce . . .

and a state to a second to be the second and the second and

and the star while care in this was again to be the the set in a start the set is a start of the start of the



W.F. Blee, 2 west

State of the state of the state

al Smellin

Con 184.

is number nemera statistic for the second statistic second statistic products a survey statistic second film a statis





323

A M. M. Burner M.

Mark without the Oak to Bake the

Auger 2

That a condition a co

We the busy as I have determined to be and a



zka uster i z eraktere i umurum tehekstörikte i sourdå ana byster i sisskistemas i fra skisteressississississi





324

2' 1 4 " 1 " werten

At Ly chie

1. 21

What the middle a for the or the

2. Alex

APPENDIX E Seventh National Conference on Production Research

いいたかにないであるとうないですいない

AN INTERACTIVE GAMING-SIMULATION OF MANUFACTURING ORGANISATIONS.

Authors: G.TRANFIELD J. I. IGARTUA

ABSTRACT.

Manufacturing organisations are inevitably divided into individual departments at some level. These departments will have their own objectives and the success of the organisation depends on the extent to which these departmental objectives are in line with the overall objectives of the organisations.Sadly this is not the case in many western manufacturing companies.

Many companies would therefore like to realign their departmental objectives and an important step in this process is to educate their staff. They must understand how their decisions affect other parts of the company and through this the overall company performance. This is inevitably a dynamic process as decisions are being made throughout an organisation in response to a continually changing environment.

Work is therefore taking place at Nottingham Polytechnic to develop a dynamic, multi-user, computer-based gaming-simulation which is designed to highlight the interactions between departments. This paper reports the progress of that work.

1.-INTERNAL CO-ORDINATION OF MANUFACTURING ORGANISATIONS.

Many authors have argued that the main problems facing manufacturing industry, particularly in the West, are organisational rather than technological. Schonberger [1] gives a classic example comparing two factories manufacturing similar products. The Japanese factory uses older technology but is more successful by almost any basis of measurement which he attributes to their superior organisation.

More recently a study conducted by the management consultants A.T.Kearny [2] concluded that in Britain of the 9 billion pounds spend on factory automation each year, some 600 million are wasted.

To some extent organisational problems arise from the sheer scale of Manufacturing companies and for many years Burbidge [3,4,5] has been pointing out the benefits of breaking down shop floor activities into product based rather than functional elements. In fact some companies have taken this philosophy to extremes and created product based business units within a factory containing the production and supporting business functions (accounting, marketing, purchasing, etc) in physical distinct areas [6].

However it is unlikely that many companies will be able to pursue this option. Most will continue to have an organisation based on functional activities like marketing, production planning, purchasing and manufacturing. The challenge faced by these companies is then to co-ordinate the activities in the individual areas.

This is not an easy task. A main theme of Goldratt's work is that:-

"The total of local optima is not equal to the optimum of the total." [7]

and that the targets set for most managers encourages them move towards local optima [8] at the expense of the global optimum. Additionally many of the reasons given in the literature (i.e. [9]) for the problem with Computer Aided Production Management Systems can be viewed as problems arising from the lack of co-ordination between departments. An overstated Master Production Schedule is the lack of co-ordination of Production and Sales. Inaccurate data often arises because the department supplying the data does not realise the importance of the data to the functioning of other departments. Lack of education again means that one department does not realise what another department is doing with the system.

This lack of co-ordination is probably most serious when an organisation is trying to undergo major change [10]. A survey of UK industry last year [11] concluded:-

"Compartmentalisation in an organisation creates diversiveness and breeds conflict which is a significant obstacle for change. Some managers' clear awareness of this does not seem to have a major impact on the type of initiatives planned for the future."

2.-ROLE OF EDUCATION.

There is clearly no simple solution to this problem. It must be solved on many fronts by providing sufficient and appropriate resources, generating enthusiasm and commitment, and providing the right kind of leadership from the top of the organisation. However, the first basic step in solving the problem must be for people to understand the problem, and the importance of education in this respect has been the subject of at least one book [12].

Traditionally education in this area has involved lectures and discussions and more recently there has been a considerable number of videos produced on the subject by such companies as the Oliver Wight organisation. These are useful ways to provide information but are rather limited in the way they can help people understand a dynamic situation of the sort that exists within a manufacturing organisation. Additionally any activity that requires the students to just listen and not participate tends to get very boring.

More recently a number of computer aided learning programs have been developed and the use of 14 packages in the field of P.O.M. (Production and Operations Management) was reviewed by Wieters & Williams [13].Unfortunately the majority of these packages are simply programs that can check whether the user is able to perform the standard (and often unrealistic) calculations that have come to form a significant part of courses in this area, and do not contribute to an understanding of the dynamics of the interactions in a manufacturing organisation.

3.-GAMING-SIMULATION.

The idea of gaming-simulation is defined by Greenblat [14]. Like a game it provides the players with goals. At the same time it in some way reflects real life and is consequently a simulation.

The majority of gaming-simulations applied to industry are business games which concentrate on different companies competing for a single market i.e. they concentrate on the external interactions of the company rather than the internal ones.

Other games (eg TEAMSKILL [14]) are based on the internal decisions made in the company but a team is given the problem. They have to provide one "answer" for each simulated month which is a form defining the next month's plan including scheduling of production, maintenance and purchasing.

This kind of gaming-simulation, however, ignores two important aspects of the real manufacturing situation. Firstly in the game there are pressures to bring the participants together. They are jointly presented with the problem and they are required to present their answer jointly by filling in one form. This contrasts with the real situation where people work in separate offices and are judged independently.

Secondly there is a very artificial treatment of time. In the game one month's results are presented and decisions are made for the next month. In reality, events (eg the receipt of an order) occur throughout the working period.

4.-NOTTINGHAM POLYTECHNIC GAME.

The aim of the current work is to produce a gaming-simulation that overcomes these problems and will help people understand the nature of the interdepartmental conflict in manufacturing and how it affects the overall factory performance. In order to do this we require a game where the players are performing the different functional tasks in an organisation and where the decisions of one player affect the opportunities offered to other players.

In order to achieve this we are developing a game based on a network of personal computers with the main data stored on a central file server using a multi-user relational database management system (FOXBASE+). The program being run in one of the computers is used to control the game as well as to generate enquiries. The player who is performing the sales role is then required to respond to the enquiries by quoting a price and delivery date. The success of his quotation is dependent on these two factors based on a simple linear relationship as shown in Fig.1.



「「「「「「「「「」」」」



The resultant sales are then passed on to the production department which has to schedule the set-ups and production of the products. The current model is restricted to 5 products, 3 machines and a single operation to produce the existing products, although the basic structure of the system is designed to allow for more complex models to be built in the future.

Feedback from Production to Sales then exists within the game because the rate of enquiries is dependent on the success of the factory to meet the quoted delivery dates.

The game is played interactively with the screens being refreshed as enquiries are received and orders are accepted. At daily intervals the company performance is displayed on screen with the option of printed reports being created.

5.-CONCLUSION.

At present the game is very much at a prototype stage. This partly reflects the fact that although the FOXBASE+ system provided a very convenient means of creating certain features of the system (screens, menus, etc) it required quite a lot of effort to create an interactive multi-user

system with a common view of time.

We are hoping to use the game on participants from manufacturing companies and to do this two main factors have to be decided. Firstly, should the players be told the principles on which the simulation is operating or should it be viewed as a "black-box" whose characteristics need to be discovered by trail and error? Secondly, to what extent should the organisers of the game try to manipulate the interactions between the players?

What we originally intend to do is to be purposely divisive, warning each player that the other is unlikely to do very well and encourage them to seek their own independent goals. Later we hope to discuss the idea of co-operation and let then discover the effect of operating in this way.

The game also needs development into other departments but feedback from the use of a simple 2 departmental model is required before moving in this direction.

REFERENCES

[1] "World Class Manufacturing: The lessons of Simplicity applied.", R.Schonberger, 1986.

[2] "Computer Integrated Manufacture: Competitive Advantage or Technological Dead End?", A.T.Kearkey, 1989.

[3] "The Introduction of Group Technology.", J.L.Burbidge, London: Heinemann, 1975.

[4] "Group Technology in the Engineering Industry.", J.L.Burbidge, Mechanical Engineering Publication, 1979.

[5] "Production Control: The future choice.", 5th National Conference of Production Research, J.L.Burbidge, 1989.

[6] Private Communication, M.Mitchell, Siemens AG.

[7] "The Theory of Constraints Journal, volumes 1-6", .Goldratt and R.E.Fox, 1988-1989.

[8] "The Goal: A process of ongoing improvement.", E.M.Goldratt R.E.Fox, 1986.

[9] "MRP: A review of failure and a proposal of recovery using CBS.", BPICS Control, G.Archer, December 1990/January 1991.

[10] "Barriers to employee involvement in technological change. More than a case of the good guys and the bad guys.", Adv. Manuf. Eng., B.Burnes, April 1990.

[11] "Managing change in the 90's - People in Business.", May 1990.

[12] "Implementing CIM: Computer Integrated Manufacturing.", A.Kochan and D.Cowan, IFS 1986.

[13] "Software Supporting Introductory Courses in Production Operations Management.", Jnl. of Prod. and Inv. Man. 30(1), 1989.

[14] "Designing Games and Simulations - An Illustrated Handbook.", C.S.Greenblat, Sage, 1988.

[15] "Try your Production Ability - go in for TEAMSKILLI", Metalworking Production, July 1978.