RENAISSANCE

[illegible] Software

PRO[illegible]

Submitted By

DINESH SARAVANAN. K 97BCT[illegible]
HARI PRASANNA . S 97BC[illegible]
V[illegible] SHANKAR . S 97BC[illegible]

Under the Guidance of

Mr. ANAND RAV. DESHPANDE MSc, MBA

In partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF SCIENCE

IN APPLIED SCIENCE – COMPUTER TECHNOLOGY
Of the Bharathiar University, Coimbatore

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

(Affiliated to Bharathiar University)

COIMBATORE – 641 006

Kumaraguru College of Technology

(Affiliated to Bharathiar University)

Department of Computer Science and Engineering

COIMBATORE – 641 006

CERTIFICATE

This is to certify that the project work report entitled

RENAISSANCE – Resource Planning Software

is a bonafide record work carried out by

Register No: 9727G0011, 9727G00[illegible], [illegible]

Name: DINESH SARAVANAN.K
HARI PRASANNA .S
VIJAY SHANKAR [illegible]

Of [illegible] Applied Science of Computer Technology

During the academic year 1999-2000

S. Ilango 27/3/2000

Head of the Department

(Mr. T. [illegible] ME)

Guide

(Mr. ANAND RAJ DESHPANDE)

Submitted for the Viva-Voce Examination held on

Internal Examiner

External Examiner

PROF. R. MEENAKSHI, B.Sc., M.A.,,
ILAIGNARKAL EDUCATION CENTRE
(Sri Aurobindo International Institute of Educational Research)
Matrimandir Nursery Gardens,
Auroville - 605101.
☎ 0413-622141/622037
E-mail: tamil@auroville.org.in and *Meenakshi@auroville.org.in*

Auroville
10.03.2000

To Whomsoever It May Concern

P-457

The following students of **Kumaraguru College of Technology, Coimbatore,** studying final year B.Sc. (C.T) have completed their project for our organisation for the year 1999 - 2000.

Mr. K. Dinesh Saravanan
Mr. S. Hari Prasanna
Mr. S. Vijay Shankar

Name of the project	: **"Renaissance "**
Operating system	: Windows 3.x and above.
Hardware configuration	: Pentium class, Min. 1MB HDD free space and Sound card.
Technical utility of the project	: Allows the user to develop his resource management powers through some exercises.
Duration of the project	: 4 months
Project implementation date	: 10.03.2000

They have put up many hours of hard work to achieve perfection in their project work. We appreciate their efforts and wish them a bright future and success in all their endeavours.

R Meenakshi

(R. MEENAKSHI)

❖ Bharati - Transit School ❖ Heritage School ❖ Rural Teachers' Training Programme
❖ Youth Counselling and Guidance ❖ Unending Education Programme for Auroville Workers
❖ Satellite Schools ❖ Publishing House

Synopsis

The environment in which we are living now is packed to the gunwales all kinds of ill factors, which are created by man himself. The best examples for this can be attributed to pollution, materialism, poverty, etc., this situation gives rise to a world where passion for humanity has a meagre role to play.

To overcome all these defects Auroville an international city was born which aims in eradicating materialistic form of life by giving utmost importance to human values & relations.

Our project, Renaissance hopes to emulate this idea of Auroville in bringing out an error free world where in work will not be termed as a profession or a way of earning one's own environment or surroundings.

The result of our project is in better understanding of auroville's concepts by the user, which is enabled by the interactive manner in which we have developed the project. We hope to see this in reality in near future.

Contents

CHAPTER PAGE NO.

1

INTRODUCTION ABOUT ORGANIZATION

The project Renaissance is accredited by Ilaignarkal, an integral part of SAIIER (Sri Aurobindo International Institute of Education and Research). Ilaignarkal is actually a school working for the rural people of the villages located near and around Auroville. This school mainly focuses on Tamil studies and literatures. The Ilaignarkal also converges the rural youths and provides them with various job opportunities in Auroville. Here the medium of teaching or education is purely Tamil based.

This school aims at providing life-oriented education rather than the normal knowledge oriented one. It strives hard in reducing the percentage of illiteracy among the rural folks by conducting both morning and evening schools, the latter being for the children going to work.

The school provides value-based education to the village youths like making handicrafts, carpentry, electrical works etc., Thus this approach makes the youth to also contribute their own cause for Auroville but pertaining to their will and wish.

2

About the Source of insight for this Project

Auroville

THE CITY THE EARTH NEEDS

"FOR MILLENNIA, WE HAVE DEVELOPED OUTER MEANS, OUTER INSTRUMENTS, OUTER TECHNIQUES OF LIFE - AND EVENTUALLY THESE MEANS AND TECHNIQUES ARE CRUSHING US.

THE SIGN OF THE NEW HUMANITY IS A REVERSAL OF PERSPECTIVE AND THE UNDER STANDING THAT THE INNER MEANS, THE INNER KNOWLEDGE AND INNER TECHNIQUE CAN CHANGE THE WORLD AND MASTER IT WITH OUT CRUSHING IT.

AUROVILLE IS THE PLACE WHERE THIS NEW WAY OF LIFE IS BEING WORKED OUT; IT IS A CENTER OF ACCELERATED EVOLUTION WHERE MAN HAS TO START CHANGING HIS WORLD BY THE POWER OF THE INNER SPIRIT."

(**THE MOTHER** 3.6.68)

2.1 THE GLOBAL CRISIS

The global Crisis, which concerns almost every one of us, has three main, closely interconnected aspects:

1. **We are unable to provide for every ones well being.**

- An enormous proportion of the world population lives below the "Poverty Line" while at the same time individual and collective greed continue to increase: 23% of the inhabitants of our planet allocate to themselves 85% of the earth gross global product. This has lead to severe pressure on the resources of our planet and to differences with in countries and between them that are both unacceptable and unaccepted. Our in capacity to share poses a major problem.

- A large proportion of the world population is still illiterate (roughly 50% in India); and when "education" is provided, it is largely only "Instruction". Many parents are forfeiting their responsibility to educate their children leaving it to the schools, where true "Value oriented education" hardly exists.

- Better medical facilities, lack of education and poverty are the three main factors responsible for the dramatic increase of the world population, which has more than tripled during the past century. As most of this increase occurred in the worlds poorest regions, these countries are caught in a vicious circle where population increase only provokes a greater lack of education and poverty.

- With the rise of un employment, the basic right to work is denied to a growing section of the world's population; in addition, most of those who do have a work rarely find it fulfilling.

- With the collapse of communism, crisis of capitalism and the persistent inability of many third world countries to improve their living standard, hundreds of millions of people are in disarray, not knowing any more which model of development to believe in.

In short: we have developed systems that imprison us, and where men's deepest aspirations are not fulfilled and all aims of society failed motivate us.

2. **We are unable to live together harmoniously**. In an increasingly competitive world, which is confronted with over population, unemployment and diminishing resources of land, water, energy, raw materials, etc., people all over the world tend to perceive their neighbors as competitors especially if they have a different cultural, religious, ethnic, linguistic or social background. Splits along communal lines are taking place in many parts of the world where people try to deprive, evict or even kill who ever is "different" from them.

At the same time our societies are becoming more polycultural, which triggers surges of xenophobia, nationalism and religious fundamentalism. There is a loss of cultural identity and a true Renaissance does not seem to take place.

3. **Man's activities have become so affective at destroying the global environment** - our common life support system - and at consuming our natural resources that we are over shooting a number of our limits. Our planet is now clearly threatened by ecological collapse.

These three failures are the source of a growing number of conflicts, which have become very violent.

2.2 Auroville, A Laboratory for the future

A few days prior to Auroville's inauguration ceremony, The Mother gave the following message:

"*India has become the symbol representing all the difficulties of modern humanity. India will be the land of its resurrection, the resurrection to a higher truer life.*"

With the following comments:

> "*And the clear vision: the same thing which in the history of the universe so as to be able to concentrate the work on one point, the same phenomenon is taking place now: India represents all the terrestrial human difficulties. And it is India that there will be the.... cure. And it is for this that I have been made to start Auroville.*
>
> *I had often asked myself the question, and all became clear at once like that. It was truly interesting. And then came immediately: well, this why you started Auroville.*"
>
> (**The Mother**, 3.2.68)

On 28th February 1968, Auroville was inaugurated in the presence of 5000 people from all over the world: representatives of 121 countries and 23 Indian states placed a hand full of soil from their respective countries in a lotus shaped urn at the center of Auroville. Afterwards The Mother read on all India radio the charter of Auroville and the following message:

> "*Greetings from Auroville to all men of good will. Are invited to Auroville all those who thirst for progress and aspire to a higher and truer life.*"

2.3 Auroville Charter

1. Auroville belongs to no body in particular. Auroville belongs to humanity as a whole. But to live in Auroville, one must be willing servitor of the Divine Consciousness.
2. Auroville will be the place of an unending education, of constant progress, and youth that never ages.
3. Auroville wants to be the bridge between the past and the future. Taking advantage of all discoveries from with out and from with in. Auroville will boldly spring towards future realizations.
4. Auroville will be a state of material and spiritual researches for a living embodiment of an actual Human Unity.

2.4 Aims and Guiding Principles

The main principle guiding the Auroville experiment is that the only way to effect the revolution the earth needs is by under going one self an inner change.

To create the circumstances propitious to this inner change, and to make the Auroville into a laboratory and a model where material and spiritual solutions to the difficulties of modern humanity are being searched and demonstrated, the Mother has given a set of practical principles:

Inner discovery

To be a true Aurovillian, the first necessity is the inner discovery by which one learns who and really is behind the social, moral, cultural, racial and hereditary appearances. At our inmost center there is free being, wide and knowing, who awaits our discovery and who ought to become the acting center of our being and our life in Auroville.

At the Service of Truth

Auroville should be at the service of the Truth, beyond all social, political and religious convictions. Auroville is the effort towards peace, in sincerity and Truth.

Beauty

In the world of farms, a violation of beauty is as great a fault as a violation of Truth in the world of ideas. For beauty is the worship of Nature to the supreme Master of universe; Beauty is the divine language in forms. And a consciousness of the divine, which is not translated externally by an understanding, and expression of Beauty would be an incomplete consciousness.

Unending Education

Auroville will be a place of an unending education, of constant progress, and a youth that never ages.
A place where education would be given not for passing examinations or obtaining certificates and posts but to enrich existing faculties and bring forth new ones.

Human Unity

Auroville wants to be a universal township where men and women of all countries are able to live in peace and progressive harmony, above all creeds, all politics and nationalities.

Freedom

The freedom we want to achieve in Auroville is not a license - each one doing what he pleases with out concern for the well being of the organization of the whole.
For those who want to be free, there is only one freedom: to be united to the supreme; and to unite with the supreme, there must no longer be any desires.

No private ownership

To be a true Aurovillian... one must lose the proprietary sense of possession. For our passage in the material world, that which is indispensable to our life and to our action is put at our disposal according to the place we should occupy there. The more conscious our contact is with our inner being, the more exact are the means given.
Auroville is the ideal place for those who want to know the joy and liberation of not having personal any more.

Work as service, not a way to earn one's living

A place work would not be a way to earn one's living but a way to express one self and to develop one's capacities and possibilities while being of service to the community as a whole, which for it own part, would provide for each individual's subsistence and sphere of action.

No circulation of money

Auroville have money relations only with out side world (1965).
The idea is that those who live in Auroville won't have money - there is no circulation of money but to eat, for example, everyone has the right to eat, naturally... those who want to eat will have to do something in exchange.

A place organized by an inner consciousness and where flexible rules and laws will get formulated progressively

No rules or laws are being framed. Things will get formulated as the underlying Truth of the town ship emerges and takes shape progressively. We do not anticipate. It is the experience of the LIFE ITSELF that should slowly work out rules that are AS SIMPLE and AS WIDE as possible, in such a way as to be always progressive. Nothing should be fixed. The probiem finally reduces itself to this: To replace the mental government of intelligence with the government of the spiritualized consciousness.

Research and experiments

Basically it should be a city for study, for study and research in to a way of life, which is both, simplified and in which the higher qualities will have more time to develop. To insist on the fact that it is an experiment, it is for making experiments...experiments, research, study.

Collaboration

A place where human relationships which are normally based almost exclusively on completion and strife, would be replaced by relationship of emulation in doing well, of collaboration and real brotherhood.
Auroville is an attempt towards world peace, friendships, fraternity and unity.
And above is the only way to do good work.

True collective Life

Auroville will provide a model for all those who aspire for a better and higher collective life everywhere.
Auroville is an experiment in collective realization.

Self supporting township

Auroville will be a city that will try to be, or will tend to become, or attempt to be "self supporting".

Sustainable development

"We want to be an example for true living in the world...
With what is at its disposal the financial could... oh transform the earth so quickly! Transform it, put it in to contact, truly into contact with the supramental forces that would make life bountiful and indeed, constantly renewed- instead of becoming withered, stagnant, shriveled up: a future dead moon...
The movement should be the opposite: the earth should become more and more a resplendent sun, but a sun of life. Not a sun that burns but a sun that illumines- a radiant glory".

- THE MOTHER.

3

Introduction about VJ++

VJ++ is an IDE for JAVA™. It is a general-purpose development tool; using it we can quickly and easily build many different types of JAVA applications, applets, enterprises components and so on. For us VJ++ is just a peripheral and appreciation of JAVA itself. It is What You See is What You Get (WYSWYG) package for JAVA.

VJ++ uses JAVA architecture for its compiler Jvc (JVC.exe). Jvc is actually a powerful compiler from Microsoft®. Jview (Jview.exe) is the run time JAVA interpreter from Microsoft®. Any program when compiled using Jvc gives out .class file, which is a Unicode file. This Unicode file has to be interpreted in order to execute it. For this interpretation purposes Jview is used.

3.1 Similarities between JAVA™ and VJ++:

- Structure of programming language:

Both Jvc and java compiler compile *.java files. These java files are the ones created using ASCII text editors. These *.java files contain keywords. control statements, looping structure, variables, methods etc., which are similar to both the compilers. The programs written for javac compiler can be compiled using Jvc and vice versa.

- Object Orientation:

Using most of the compilers we can get executable codes according to the operating system platform. This reduces the portability of the program can be run only in the platform. But in javac and Jvc the compiler will give .class files as output. These .class files are Unicode files. This can be transported to any platform and then interpreted using run time interpreters or Java Virtual Machines (JVM).

3.2 How VJ++ differs from JAVA^{TM}:

Java programs need to be written in an editor, which is not the one, and particular editor for java so. Every time if you want to compile a java program we must exit from the editor and compile it out side the editor. This increases the time consumed. But in VJ++ all these facilities are integrated in a single place i.e., editor, compilation, execution can all be done with out exiting from it. Also errors are checked for then and there i.e., is during the process of typing itself.

Java has an Abstract Windowing Toolkit (AWT) using, which we have to write separate codes for each and everything. For example even if we want to create a tool bar we have to do it by writing appropriate codes for it. But in VJ++ we have a WFC i.e., Windows Foundation Classes in which all predefined. The only job of the user is to transfer the required resource from WFC to the appropriate form. This reduces time consumed and also makes the job of the user lot easier. Further there are several facilities and advantages, which cannot be fully listed here.

4

Specific Features

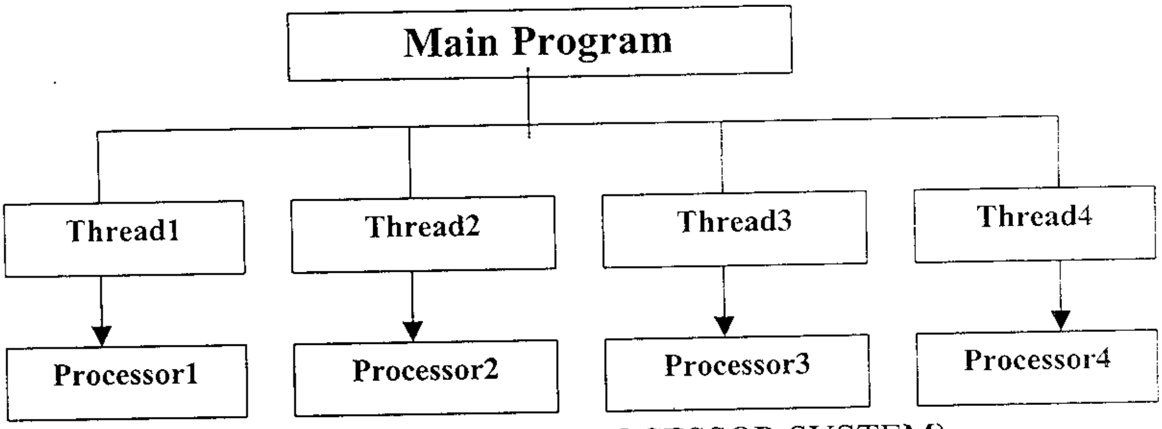
When we say specific features we does not mean about VJ++'s specific feature from other languages. But in turn the specific features of the VJ++, which are used effectively in our project.

The features made use of in our project are:

1. Multi Threading
2. Windows Foundation Classes (WFC)
3. Multimedia

In the past when multiprocessor system first came in to effect, each processor was assigned a separate program to be executed. Hence this reduced the fullest utilization of the multiprocessor system and its capabilities, if single program has to be run irrespective of its size. This method of execution allows only a single processor to the program, which increases the time consumed for a program in case of the program being big.

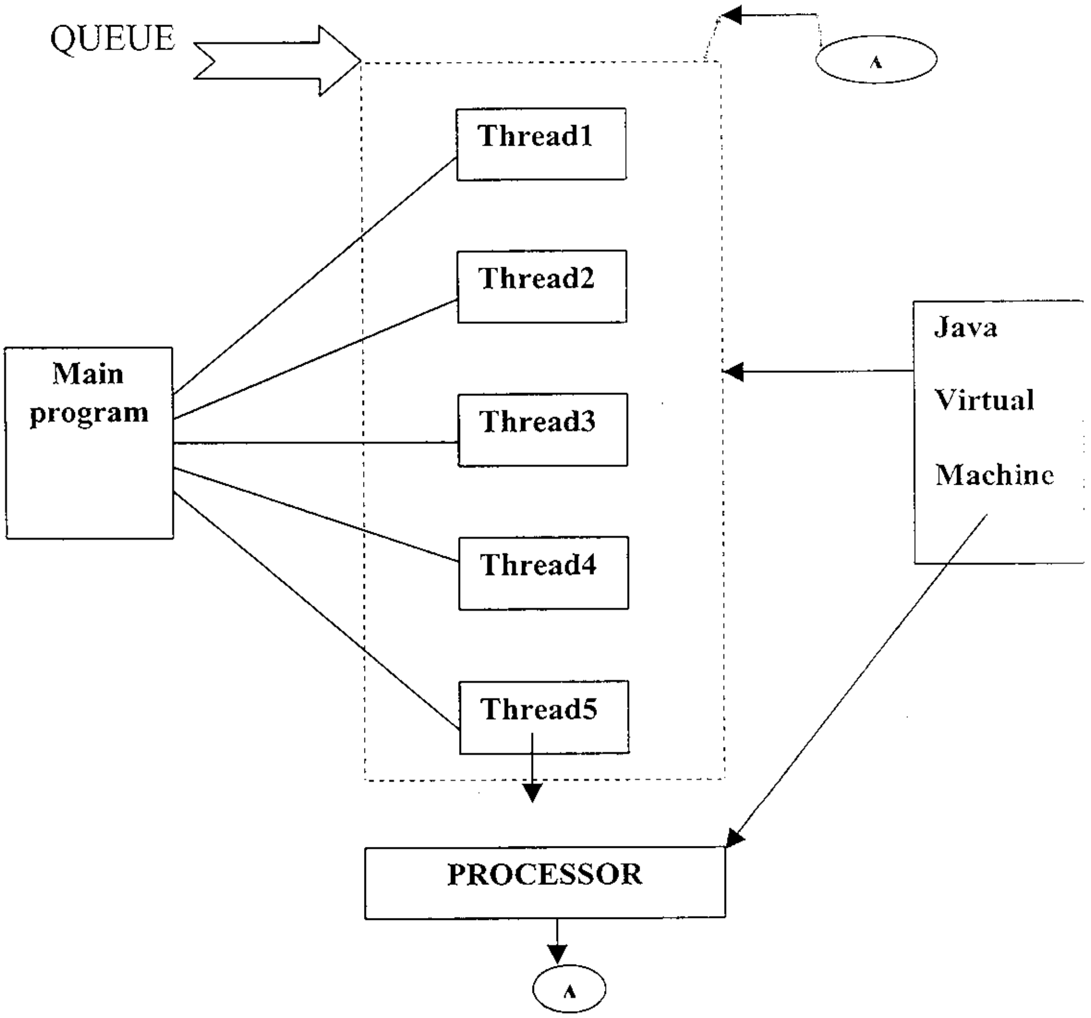
To extinguish this above said disadvantage a program is divided or separated in to different **threads**. A **thread** is nothing but an independent process with in a program. The following chart can represent this:



(THREADS IN A MULTIPROCESSOR SYSTEM)

Thread can also be useful and helpful in the case of systems with only one processor also. Unlike threads executed in multiprocessor systems, the threads here wait in a queue in which a separate time slot is assigned for each thread. This necessarily need not be done in a multiprocessor operating system but can also be done in a DOS system.

This allocation of threads to a single processor is managed by JVM (Java Virtual Machine). By management we mean the allocation of threads, time slice allotted to them, order of assigning it, etc. This differs from machine to machine based on JVM.This can be represented diagrammatically as:



From this, multithreading can be defined as a process in which two or more threads are processed simultaneously, hence the above explained process is the actual multithreading.

In our project there are a lot of independent processes as we are implementing the real time data. This includes time, population, available money, satisfactory level and various attributes, which are run based on the time.

For example, if we take the time thread i.e., month and year, it is implemented in our project as: hare every month is incremented for every minute. Hence initially the present month is left as it is and the thread goes to sleep for one minute, after which it increases the month. It also checks weather the month has increased to 12 or above if it is then the year is incremented by one. After this process the thread once again goes into sleep for one minute after which this process continues once again, till the quitting time of the software. As this process runs as a separate thread it will not interrupt the other processes of the software. In addition to this there are separate temporary threads that run along with this particular thread.

The Windows Foundation Classes are a large framework of classes built to give the java programmers complete access to the power and functionality of the win32 bit OS. It is a pretty remarkable set of classes. The functions exposed by this set of classes provide functionality equivalent to several of the packages available in the java core API. These functions include a user interface (UI) framework analogous to the java.awt. It also includes a set of packages dedicated to database programming. These are wfc.data, wfc.data.adodb and wfc.data.ui packages.

The WFC packages also include html and more importantly DHTML support in wfc.html package. This is a very unique and powerful package. But the most important factor that was useful in our project was the core functionality of WFC, which is stored in wfc.core.Component class. Through inheritance from this class all other WFC components inherit three attributes: properties, events and methods.

The WFC is a java implementation of Microsoft's Component S/W model. Another implementation of this model would be ActiveX's OLE automation control; through those controls are **COM** implementations of the model. The component S/W model defines components as objects that have three major capabilities: properties, event, and methods. All three are dynamically available to a program i.e., without static linking; an application can discover a component's properties, events and methods. The application can also discover the types and value of the each component's properties as well as modify the properties. Using the WFC, application can also subscribe to the Component's events and application can invoke the component's method all without having any static link to the component's code.

The component model is the most important for visual programming. A component can easily be added to the palette of the visual programming environment such as VJ++.

Since WFC and ActiveX™ components software definitions are both based on same model – in which each component has a set of dynamically discoverable properties, events and methods – it is very easy to bridge between WFC and ActiveX™ i.e., it becomes a very easy task to add a component developed in VJ++ to a VB application as well as to add an ActiveX™ component into a user interface developed in VJ++. The component model is language independent.

There are few important distinctions between java beans and WFC Components; most important is that the java beans specification was built from the beginning to be platform independent. The WFC's on the other hand are built to fit very well into the ActiveX™ component's model and will really only work using the Microsoft's Virtual Machines.

Ours is a resource planning software and hence has to be used till the end is reached. Hence in order to make the user feel tireless we have incorporated background music, which continuously goes on until the end is reached. This gesture acts as a response to the user as long as he/she is working.

The music files are normally stored in the extension .Wav i.e. as wave files. These wave files can be directly called in the program but instead of doing it so we have put the wave file in the HTML format & the respective HTML file is called inside the project.

Since our project needs to be a user interactive one, there arises a need for representing each & everything in a pictorial manner. Hence we have included a large number of picture files, which will make the project a more interactive one & also enables easy understanding. The pictures are included in the project using the WFC i.e. Windows Foundation Classes in which there is a picture controller.

5

INTRODUCTION ABOUT PROJECT

5.1 Purpose

The main purpose our project is to propagate the concepts of Auroville to the masses. Thus inducing them to develop an environment like that of Auroville.

This can be also taken as a propaganda for Auroville doing which the message of living in unison with nature & with fellowmen could be reached all over the world.

5.2 Scope

The resource planning software has a definite scope. We can increase the GUI by inducing DirectX™ programming where we can see 3D animations.

The software possesses more challenging when more real time facts are being added i.e. it makes the software more sensitive based on user's decisions.

5.3 ACRONYMS AND DEFINITIONS:

AWT	Abstract Windowing Toolkit
COM	Component Object Model
DHTML	Dynamic Hyper Text Mark up Language
DOS	Disk Operating System
GUI	Graphical User Interface
HTML	Hyper Text Mark up Language
JVM	Java Virtual Machine
UI	User Interface
WFC	Windows Foundation Classes
IDE	Integrated Development Environment

6

General Description

6.1 Product Perspective

The following few points are our products perspectives:

1. The system focuses on resource planning which is completely based on auroville's prospects.
2. It also acts as a means of propaganda for Auroville.
3. This system provides a user interactive, strategy based resource planning software.
4. It also inculcates in the users mind the urge in envisaging an error free world.
5. It brings out the leadership qualities in the user.

6.2 Product Functions

The project "Renaissance" starts from the year 1968, where in there are 120 people in Auroville. The main aim of the user should be in achieving the target of 55,000 people and 4300 acres of land. And bringing self-sufficiency in all aspects of life i.e., land, water, food, etc.,

The starting screen is displayed with option of opening a new project. When we click it all the real time attributes like land, transport facilities, education services, etc., are displayed, there runs a separate thread called time which increments the month for every minute and increments the year once in a every 12 minutes.

In the land attribute there is displayed a separate screen in which there are provisions for either buying or selling a land. There are also options for the land available and the land yet to be bought. At this juncture there is also another thread running called population this increases as the months pass by and also increases or decreases depending upon the availability of food, water and other attributes.

The constraint in selling a land is that it will not go to sales for the same cost for which it has been purchased but for the cost which is usually half or a quarter part of the original cost. Hence the user before selling the land must think twice. Similarly as the population increases there occurs more need for lands. It is not only necessary to buy a land but also the lands bought must be reclaimed for cultivation. This can be increased or decreased through buttons available for the available for the respective purposes.

As the reclaimed land increase, there occurs a condition to employ proportionate number of employees. Hence there is an increase in the intake. This in turn increases the need for food commodities. Hence more people need to be recruited.

The raise in the number o people or the increase in population increases the need for a proper transport facility. This in turn requires more volumes of land for roads, and also the need for an appropriate communication service. Hence the requirement for more buses needs to be fulfilled. This has to be done proportionate to the present population. The user must also be watchful in utilizing people as the population increases.

When there is a hike in the number of people, naturally there is an urging need for taking care for their health. Hence there arises the need for opening more number of health centres and sub centre. This factor can also be increased or decreased through the buttons provided for it. There is also an option for campaigning to the people about various diseases that are presently rocking them. It is nothing but a process of creating awareness between the masses about the various hazards brought by the disease and also various precautionary methods for it, which is specified implicitly.

There are lists of diseases given, from which the user can select one and campaigns on that. This is also a separate temporary thread, which begins and ends without any hindrances to the on going operation. For various diseases which are not included in the list there is a special option provided as **Others**, clicking which the user can enter the name of any disease through the keyboard and also can campaign on that one.

As the population increases there will be a raise in the number of children. The children need basic necessity of education. So, we have a separate attribute as education, in which there are options to construct primary schools and also evening schools. These primary schools are meant for the rural youths who are mostly illiterates. Further more education is not considered as a means of getting a degree but a basic idea about the works, which will be handy to them.

The evening schools are mainly for the ones who are working. The worst part is that, all are children and this education literates them about their needs. This creates awareness in them and also makes them to think of their own. These are the various methods of raising their satisfactory level.

In addition to the above said attributes there is another attribute called researches, which has a list of researches to be done on various fields such as land, infrastructure, water, etc., The user can simply chose a topic for research can add to the list of researches being done. The researches can be in the area of low cost construction, bio farming, etc. For all these actions the total money available is decreased and can be increased by the satisfaction level of the people.

There is also a separate attribute for the area of renewable resources. The resources falling under this category are solar energy and wind energy because these two resources are the ones which can be redeemed again and is also of low cost. Solar energy can be tapped using solar cells. These energies can be used to run a machine or for personal use also. Wind resource is consumed by installing numerous windmills, which are also an important factor in reducing the total amount spent without reducing the available facilities. These renewable resources can also be increased or decreased in number.

Above are the various product functions available in the project.

6.3 User Characteristics

Since, this project is based on the welfare of mankind, any one can utilize this software.

But it also depends on the mental health of the user. If the user is unable to recognize the resources then there is no point in using it. That is why we request the users to go through the **Scenario** and **HowTos** part before getting in to the resource planning events.

If the user is a child he should have the capacity to think about the identity of the resource. Unless he thinks, he can never succeed.

Mind maturity is one of the most influencing factors in using this software. In using this software.

6.4 General Constraints

Depending on the availability we have created this project. So, there will be limitations in its usage respectively.

The Basic constraints are:

- As it is developed in Windows OS, the software only runs in a GUI base OS i.e., Windows.
- No Keyboard in put is accepted.
- Some of the objects like administration services, Commercial activities etc., will not be in use as it's availability facts are less. So it is left for future expansion.
- Screen setting should be set to 800 x 600 pixel resolution or else the software will not be seen well.

7

REQUIREMENT SPECIFICATION

7.1 Fundamental Requirements

7.1.1 List Of Inputs

The only source of input in this project is mouse click.

There will be various buttons in the increasing and decreasing the resources from the screen. They act as inputs.

The inputs are listed according to Screens one by one.

Land Screen:

1. Land Bought in Green Belt area.
2. Land Bought in City area.
3. Cost of Land in Green Belt area.
4. Cost of Land in City area.
5. Number of People Working for Land Reclamation.
6. Number of People Working for Water Shed Management.
7. Number of People Working for Afforestation.

Community Services Screen:

1. Number of land used for food processing.
2. Number of people working for food processing.
3. Number of vehicles used for public transport.

Health Care Screen:

1. Number of main health centres.
2. Number of sub health centres.
3. Topic for campaigning.
4. Start Campaign Command.

Education Screen:

1. Number of active schools.
2. Number of Technologies used in schools.
3. Number of evening schools.
4. Number of month intervals for campaigning for evening schools.

Research Screen:

1. Topic to be researched.
2. Amount funded for the research.
3. Number of people working for research.

Infrastructure Screen:

1. Number of windmills installed.
2. Number of solar cells installed.
3. Number of biogas plants installed.
4. Number of men working for water sanitation.
5. Number of men working for roads and traffic.
6. Number of parks and gardens with in city area.

Accommodation Screen:

1. Land allocated for a single Aurovillian.

Art and Culture Screen:

1. Category of the dance classes that have to be conducted.
2. Category of the music classes that have to be conducted.
3. Conduct Bon Fire Command.

7.1.2 List of Outputs:

The outputs in our projects will more be in a graphical representation like progress bars rather than in numbers. Some times there will also be numbers.

1. Money, this shows the amount the user have with him to develop the city. (TEXT)
2. Month, this shows which period of the software at present user is using. (TEXT)
3. Year, this shows which year of the software at present the user is using. (TEXT)
4. Population, this shows the number people living in user's city according to the software. (TEXT)

These are the outputs that are common to all screens. And following are the outputs that order according to the screen they appear.

Main Screen

1. Current Aurovillians. (TEXT)
2. New Aurovillians. (TEXT)
3. State of Lands. (PROGRESS BAR)
4. State of Community Services. (PROGRESS BAR)
5. State of Health Care Provided. (PROGRESS BAR)
6. State of Education Provided. (PROGRESS BAR)
7. State of Infrastructure. (PROGRESS BAR)
8. State of Accommodation. (PROGRESS BAR)
9. Progress of the research. (PROGRESS BAR)

Land Screen:

1. Number of available lands for the price user is ready to pay. (TEXT)
2. Level of land reclamation. (PROGRESS BAR)
3. Level of Water Sanitation. (PROGRESS BAR)
4. Level of Afforestation. (PROGRESS BAR)
5. Money Spent on land reclamation, water sanitation, afforestation. (TEXT)

Community Services Screen:

1. Utilization of land used for Food Processing. (PROGRESS BAR)
2. Diet level of the people. (PROGRESS BAR)
3. Amount spent on food processing. (TEXT)
4. Transportation need. (PROGRESS BAR)
5. Amount spent on transportation. (TEXT)

Health Screen:

1. Availability of health centres. (PROGRESS BAR)
2. Time since last campaign. (TEXT)
3. Topic of the current campaign. (TEXT)
4. Progress of the current campaign. (PROGRESS BAR)
5. Amount spent on health care. (TEXT)

Education Screen:

1. Number of people wanting education. (TEXT)
2. Number of people provided by education. (TEXT)
3. Level of education provided. (PROGRESS BAR)
4. Service provided by evening schools. (PROGRESS BAR)
5. Amount spent on education. (TEXT)

Research Screen:

1. Topics for research. (LIST BOX)
2. Name of the research that is in progress. (TEXT)

Infrastructure Screen:

1. Energy consumed by Auroville per month. (TEXT)
2. Amount spent on infrastructures. (TEXT)
3. Level of water and sanitation. (PROGRESS BAR)
4. Level of roads and traffic. (PROGRESS BAR)

Accommodation Screen:

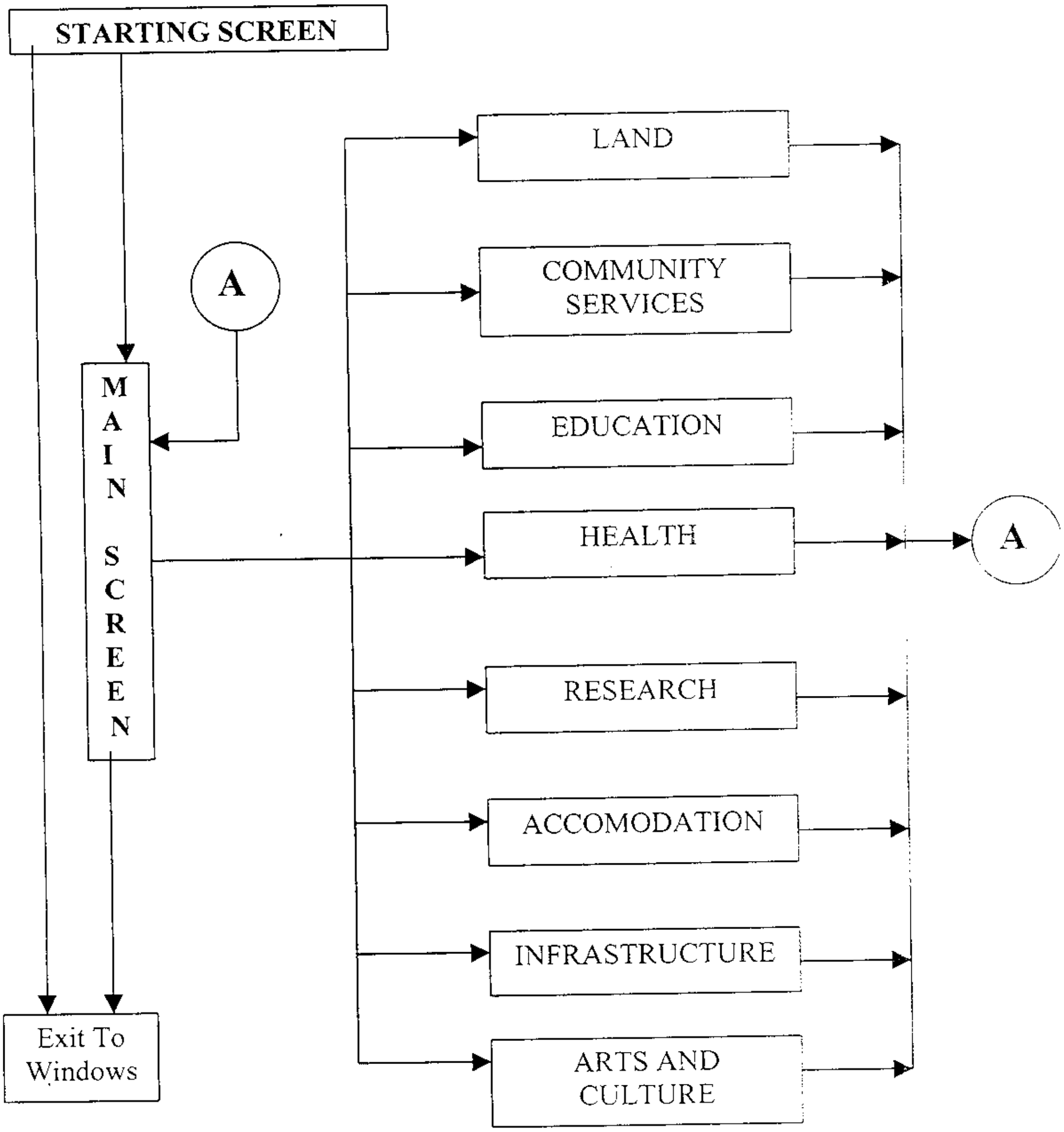
1. Population. (TEXT)
2. Land Accommodated by population. (TEXT)
3. Amount spent on accommodation. (TEXT)
4. Land available for accommodation. (TEXT)
5. Satisfaction of population on land accommodation. (PROGRESS BAR)

Art and Culture Screen:

1. Happiness of the people. (PROGRESS BAR)
2. Progress in dance class. (PROGRESS BAR)
3. Progress in music class. (PROGRESS BAR)
4. Time since last bon fire. (TEXT)

8

MENU MAPPING



9

DESIGN CONSTRAINTS

9.1 Hardware Limitations

1. Pentium processor and Above.
2. Colour monitor that supports 800 x 600 resolution.
3. Mouse.
4. Keyboard.
5. Hard disk with minimum 6 MB free space.
6. Sound Card. (optional)
7. Minimum 16 MB RAM.

9.2 External Requirements:

9.2.1 Software:

1. Java Virtual Machine.
2. Internet Explorer® 5.0.
3. Windows® 95 and above.
4. Windows media player®.

9.3 Screen Formats

In this software you, could see a main screen where eight screens are accommodated. Those screens are

1. Land.
2. Community Services.
3. Infrastructure.
4. Education.
5. Research.
6. Health.
7. Accommodation.
8. Arts and Culture.

Apart from this there are two more screens, Commercial Activities and Administration Services, which have been left for future expansions.

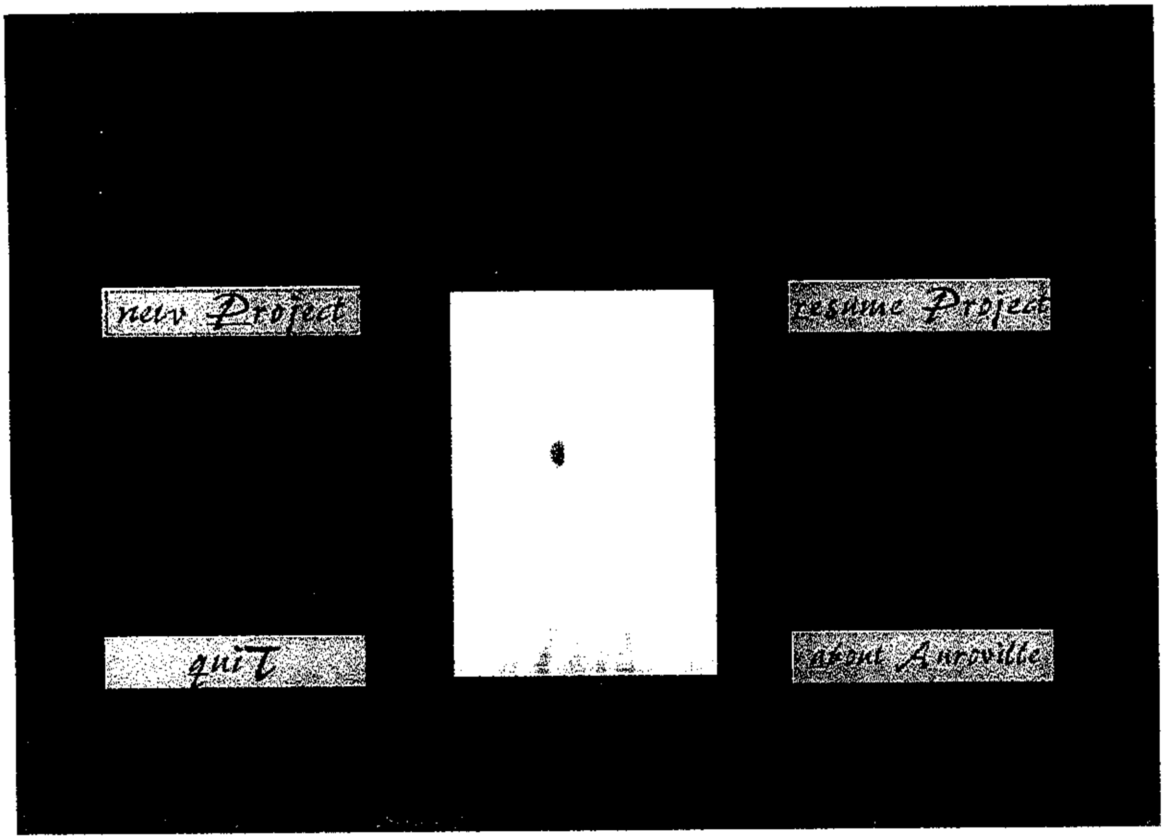
Loading Screen

The World is preparing for a big change. Will you help?

-The Mother.

Since our resource planning software is based on real time facts of Auroville we have used this screen as the loading screen. This photo was taken on 28th February 1968 the day when Auroville was inaugurated. So this photo also depicts the scenario of the project.

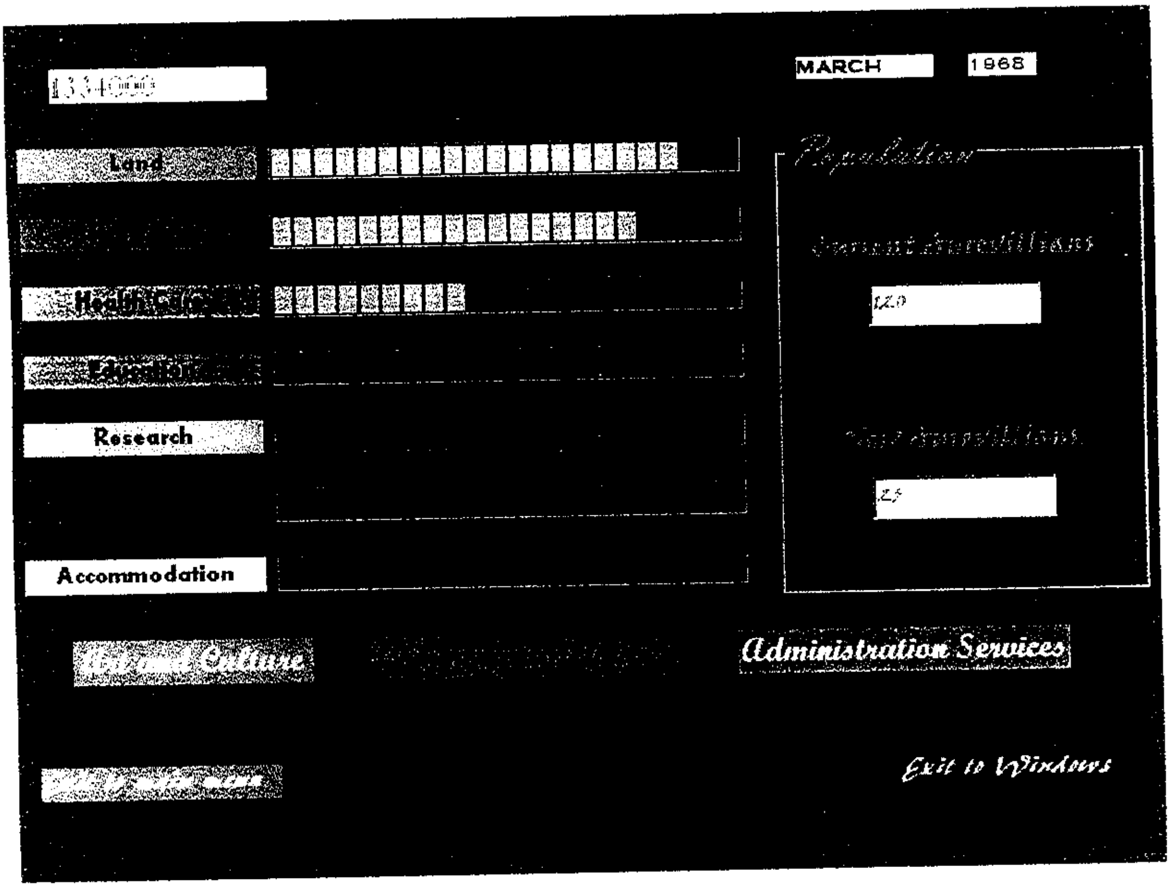
Opening Screen



This Screen is the main screen of the project. In this screen there are four choices.

1) Load a new project
2) Resume to the project to which we were working previously.
3) To see the full details about Auroville's concepts this also allows us to understand the project fully.
4) Quit to windows.

Main Screen



1334000

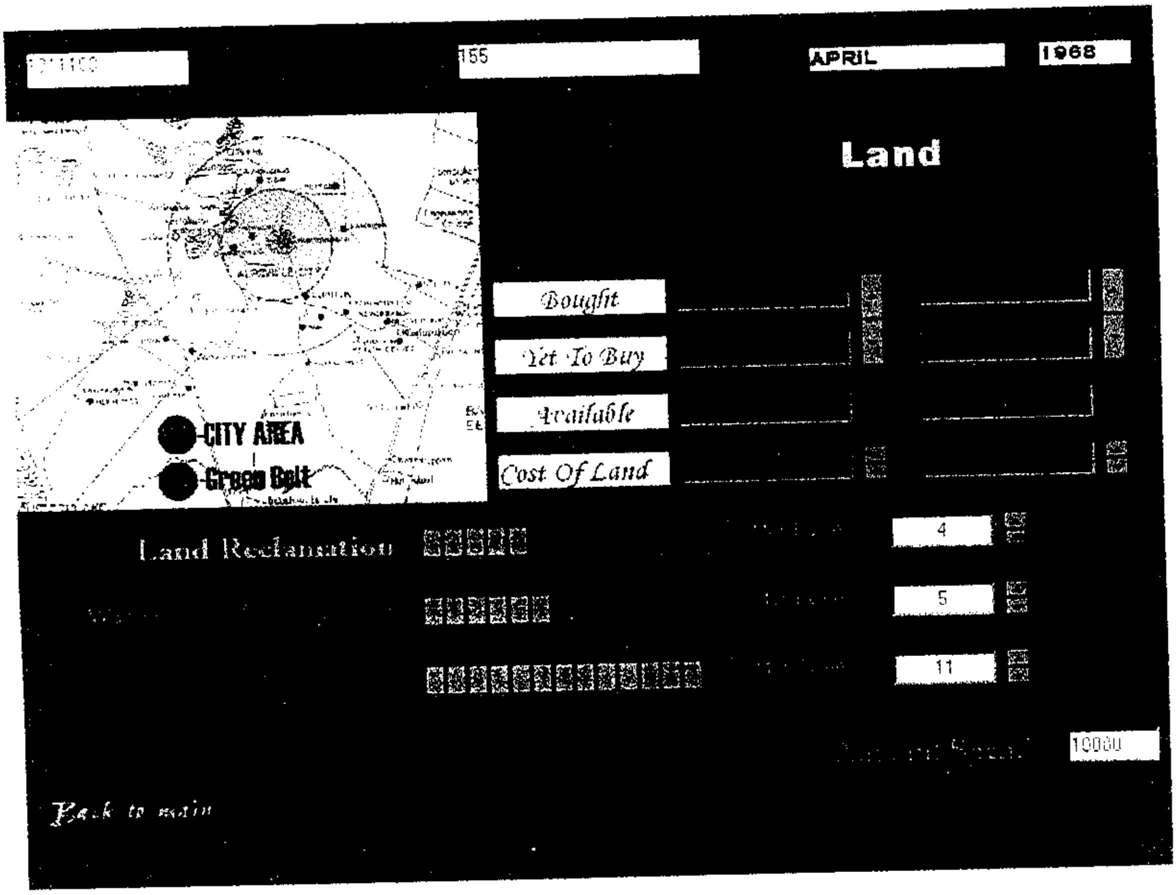
MARCH 1968

Current Aurovillians: 120

New Aurovillians: 25

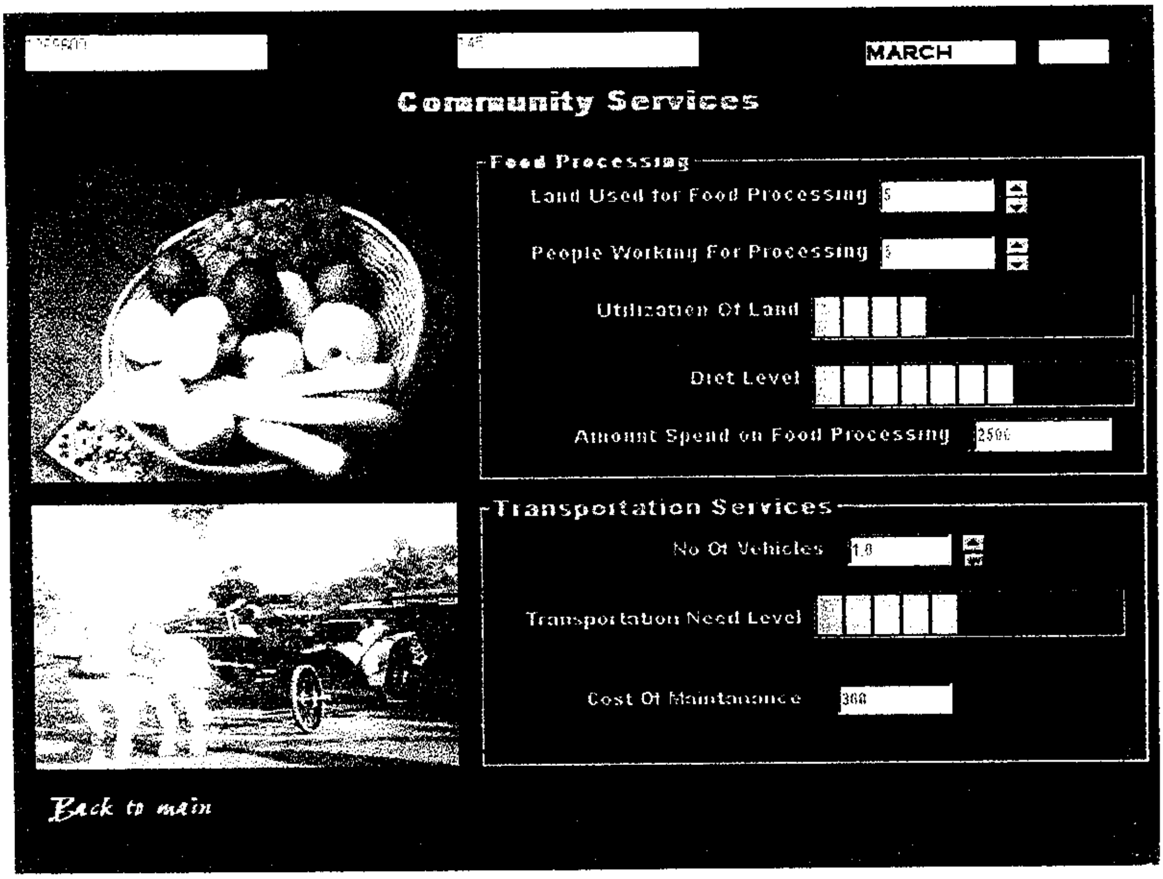
This is the main screen of project. From this screen we have direct access to all other attributes such as Land, Community Services, Health care, Education, Research, Infrastructure, Accommodation, Art and Culture. By clicking the name of the attributes present in this screen we can go to the screen of that attribute. And there is also two more options namely exit to windows, exit to main menu. In this screen information about year, month, money in hand and population etc., are also available. Two vital information of population namely Current Aurovillians and new Aurovillians are also present in this screen.

Land Screen



In this screen there are two main objects of land. They are **City area** and **Green belt area**. They both have similar elements and they are **Bought**, **Yet to Buy**, **Available**, **Cost of Land**. The three progress bars gives an idea about progress of the factors **Afforestation, Water Shed Management** and **Land Reclamation**. Each progress bar has an associated **manpower** with it. There is also text box which shows the amounts spend for the land activities.

Community Services Screen



MARCH

Community Services

Food Processing

Land Used for Food Processing 5

People Working For Processing 5

Utilization Of Land

Diet Level

Amount Spend on Food Processing 2500

Transportation Services

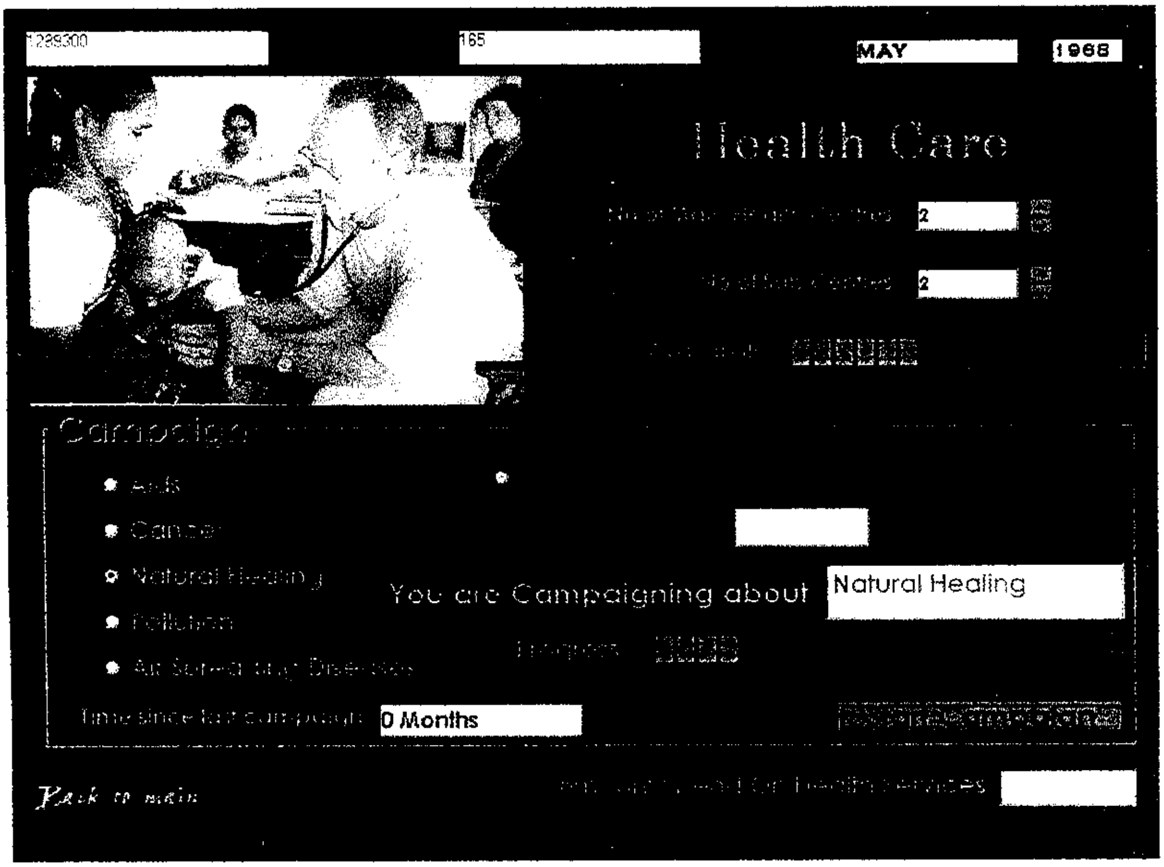
No Of Vehicles 1.0

Transportation Need Level

Cost Of Maintanance 300

This screen contains two units, one is **Food-processing** unit and the other is **Transportation service**. In the food-processing unit we have **Land used for food processing** and **People working for food processing** which acts as input. There are also two progress bar **Utilization of land** and **Diet level** which shows their relevant progress. **Amounts spend for food processing** is also shown in a text box. In the other unit, Transportation service we have input for number vehicles. The progress bar in this unit shows the transportation need. Here also there is a text box to show **cost of maintenance**.

Health Screen



1289300

165

MAY

1968

Health Care

No of Main Health Centres 2

No of Sub Centres 2

Availability

Campaign

- Aids
- Cancer
- Natural Healing
- Pollution
- Air Spreading Diseases

You are Campaigning about Natural Healing

Progress

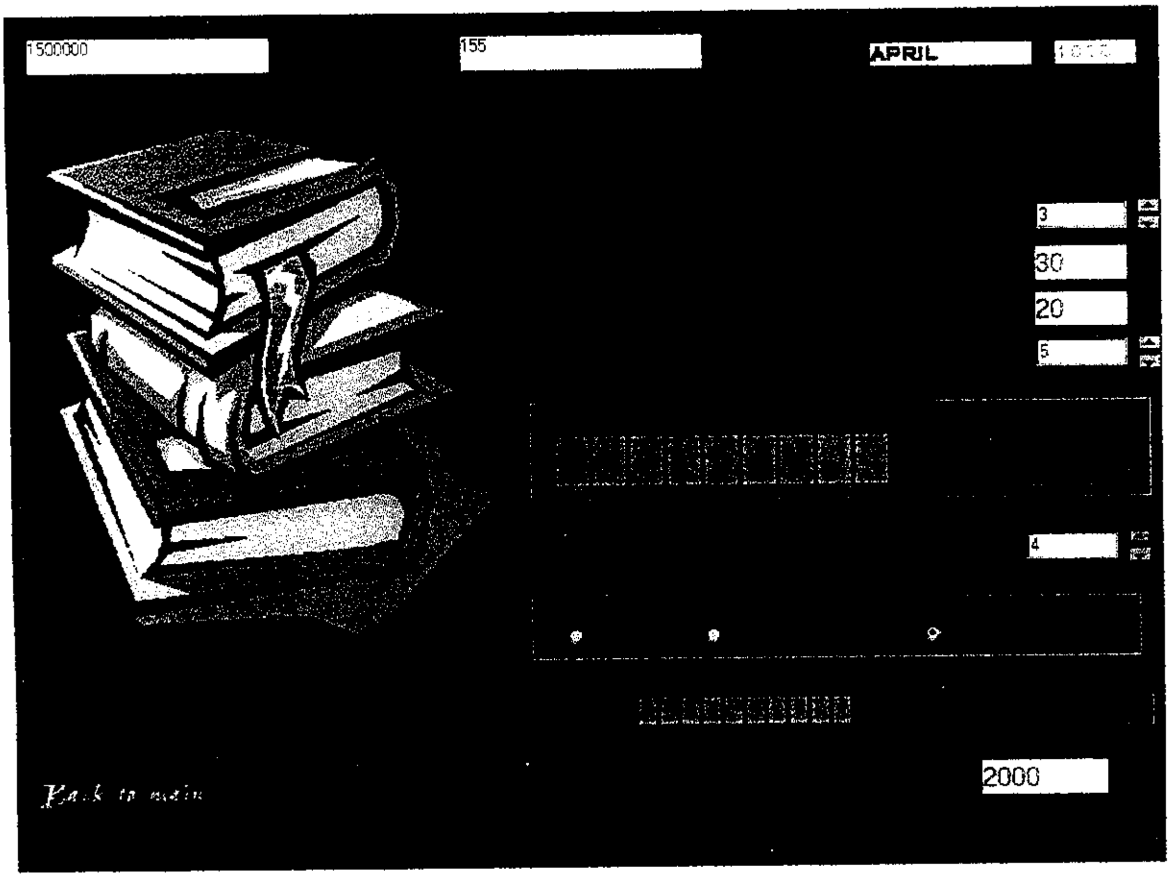
Time since last campaign 0 Months

Back to main

Amount spend for health services

In the **Health Care** screen we have Number of **Main Health Centres** and **Number of Sub Centres** as inputs. There is a progress bar for **Availability** that shows the accessible status of the health centres. There is a **Campaign** group box in which there are a number of topics to be selected for campaigning and a text box is included for campaigning on user created topic. A progress bar is included to show the status of campaign. **Time since last campaign** is also shown, in months. Finally **Amounts spend** on Health services can also be seen in this screen.

Education Screen



1500000

155

APRIL

3

30

20

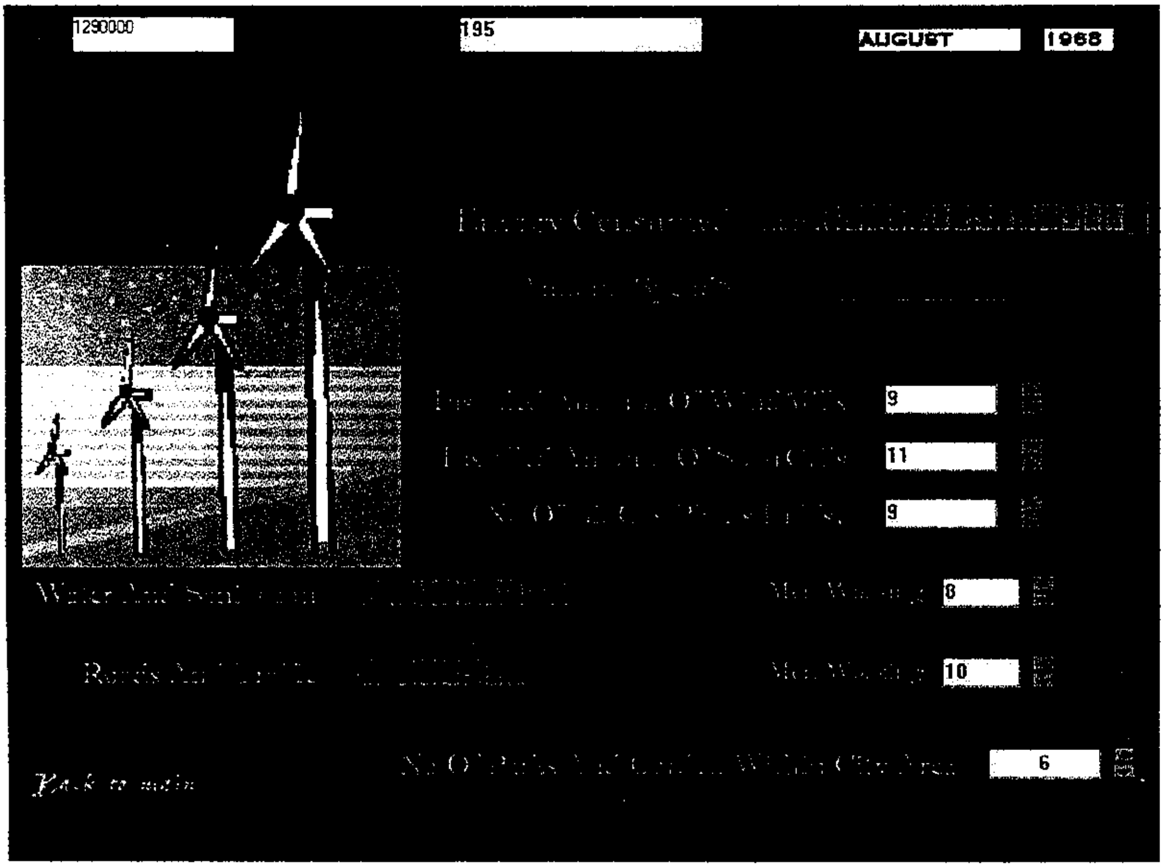
5

4

2000

The **Education screen** has two main inputs. They are **Number of Active Schools** and **Number of Technologies Used**. **Number of Population Needing Education** and **Number of Population Provided By Education** factors depends upon population attribute. A progress bar for **Satisfaction on Education Provided** can also be seen in this screen. We also have included the **Number of Evening Schools** as an input. There is a group box on campaign for Evening Schools present here. **Amount Spent on Education** is also provided here in text box.

Infrastructure Screen



1290000

195

AUGUST 1968

Energy Consumed

Amount Spent

Installed Amount Of Windmills 9

Installed Amount Of Solar Cells 11

No Of Biogas Plants Used 9

Water And Sanitation

Men Working 8

Roads And Traffic

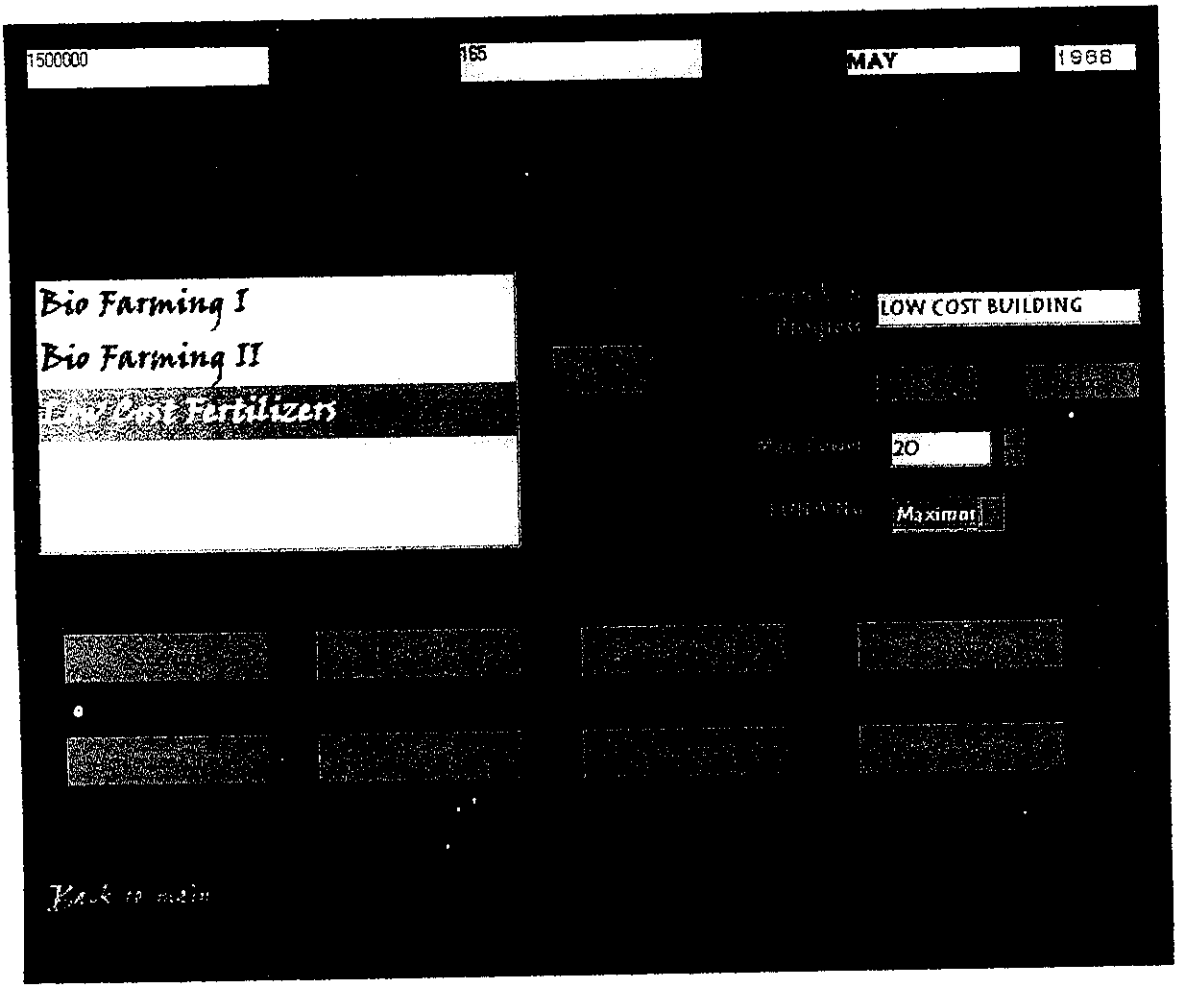
Men Working 10

No Of Parks And Gardens Within City Area 6

Back to main

Infrastructure Screen mainly focuses on power. There is a progress bar to show the power consumed. Amount spent for it can also be obtained. Installed Amount of Windmills, Installed amount of Solar Cells and Number of Biogas Plants Used are the user inputs in this screen. In this screen men can be allotted to work for Water and Sanitation as well as Road and Traffics, whose conditions can be viewed in separate progress bars. Parks and Gardens can also be built in city area.

Research Screen



1500000

165

MAY

1988

Bio Farming I

Bio Farming II

Low Cost Fertilizers

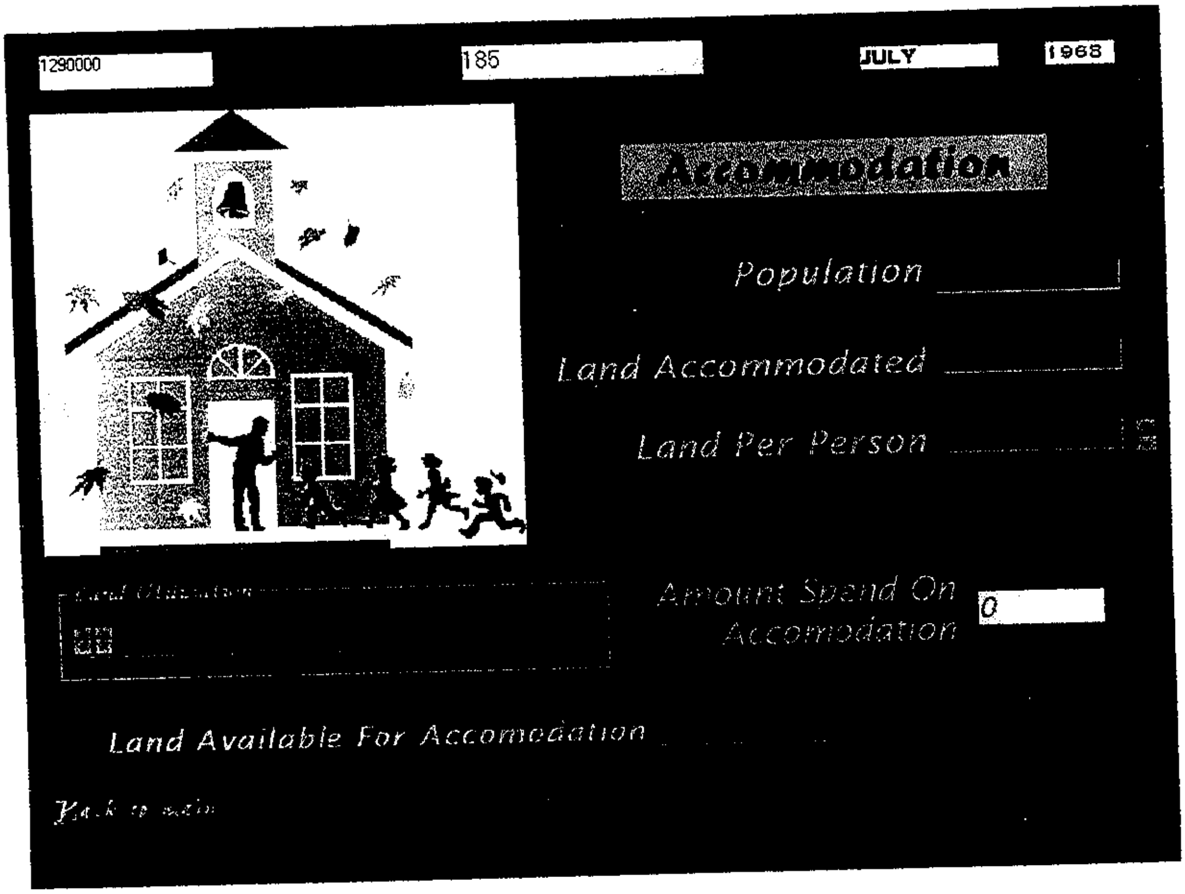
Progress: LOW COST BUILDING

20

Maximal

In this there are various fields such as Scientific, Land & Water, Bio Farming etc., on which researches could be performed. The topics under each field are listed in a list box. The topic of currently under going research is also shown. The fund for the research can be selected between MAXIMUM, MINIMUM & MEDIUM. Manpower for research can also be allocated in this screen.

Accommodation Screen



1290000 185 JULY 1968

Population

Land Accommodated

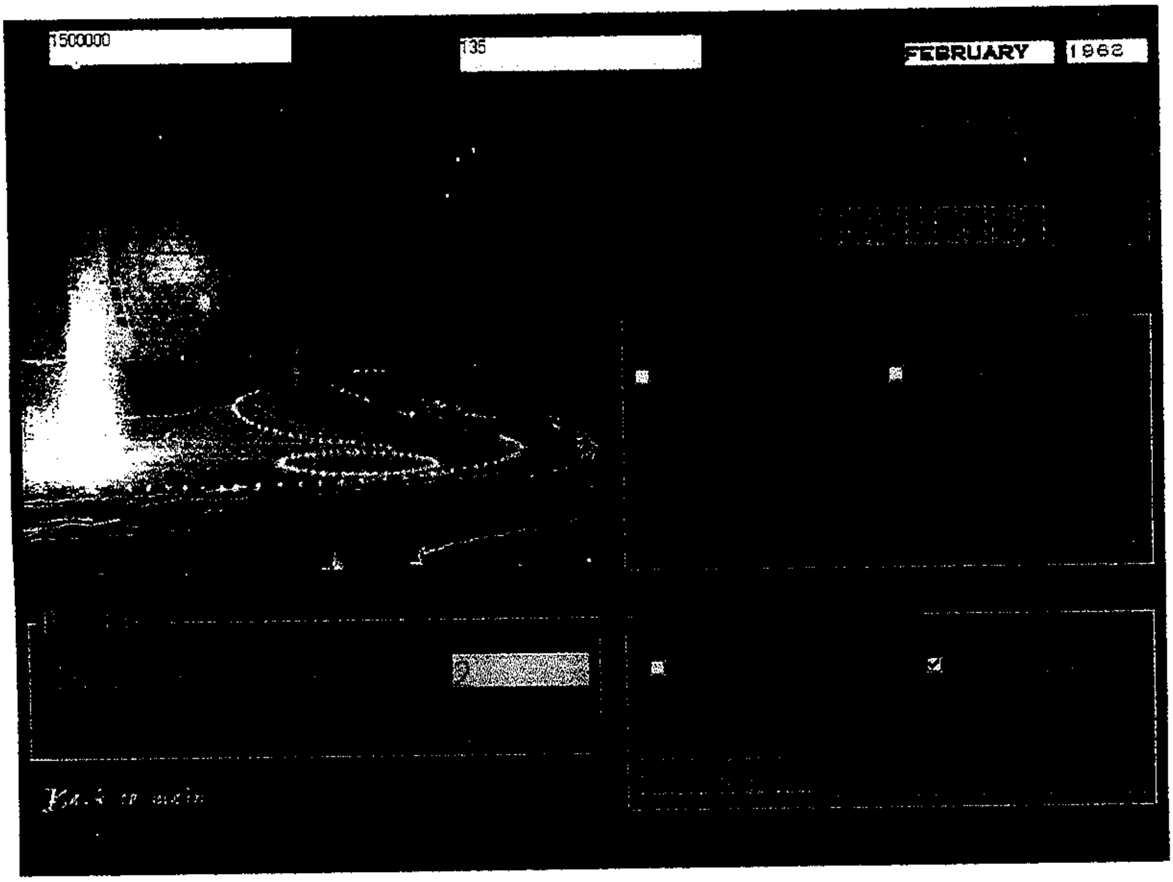
Land Per Person

Amount Spend On Accomodation 0

Land Available For Accomodation

Accommodation need of the population is sought for in this screen. User must assign Land Per Person according to which Total Land Accommodated is shown. The satisfaction of people over amount of land accommodated can be derived from the progress bar. And the attributes like Population, amount spent on Accommodation are also displayed in this Screen.

Art and Culture Screen



1500000

135

FEBRUARY 1962

The progress bar in this screen shows the happiness of the people. To increase happiness we have to conduct gatherings like Bon Fire. And improving Cultural activities, by teaching music, dance to the people will also be very helpful to increase the happiness of the People. The progress bars in Music, Dance group boxes shows the progress in coaching classes.

10

Sample Code

10.1 Sample code for a Form

```
import com.ms.wfc.app.*;
import com.ms.wfc.core.*;
import com.ms.wfc.ui.*;
import com.ms.wfc.html.*;

/**
 * This class can take a variable number of parameters on the command
 * line. Program execution begins with the main() method. The class
 * constructor is not invoked unless an object of type 'LandScr'
 * created in the main() method.
  */
public class LandScr extends Form
{
        private                          static                          int
        GreenAvail,CityAvail,GreenBought,CityBought,GreenYetBuy,CityYetBu
        y,GreenCost,CityCost;
        private static int RecalMan,WaterMan,AfforestMan;
        public LandScr()
        {
                super();

                // Required for Visual J++ Form Designer support
                initForm();
                MonthTxt.setText(Time.Months);
                YearTxt.setText(""+Time.year);
                PopTxt.setText(""+Population.TotPop);
                GreenAvail = LandClass.GetGreenAvail();
                GreenBought = LandClass.GetGreenBought();
                CityBought = LandClass.GetCityBought();
                CityAvail = LandClass.GetCityAvail();
                GreenYetBuy =LandClass.GetGreenYet();
                CityYetBuy = LandClass.GetCityYet();
                GreenCost = LandClass.GetGreenCost();
                CityCost = LandClass.GetCityCost();
                CityAvaiTxt.setText(""+CityAvail);
                GrAvaiTxt.setText(""+GreenAvail);
                GrBoughtTxt.setText(""+GreenBought);
                CityBoughtTxt.setText(""+CityBought);
                CityYet2Btxt.setText(""+CityYetBuy);
                GrYet2BTxt.setText(""+GreenYetBuy);
                GrCstTxt.setText(""+GreenCost);
                CityCstTxt.setText(""+CityCost);
                MoneyTxt.setText(""+Money.TotalMoney);
                AmoLand.setText(""+Money.Land);
                RecalTxt.setText(""+LandClass.GetLndHumanAssighned());
                WatSheTxt.setText(""+LandClass.GetWaterHumanAssighned());
                AfforTxt.setText(""+LandClass.GetAfforestHumanAssighned());
                if(LandClass.GetLndHumanNeeded() == 0)
                {
                        progressBar1.setValue(100);
                }
                else
                {
                        progressBar1.setValue((int)(100*LandClass.GetLanRe()));
                }
                if(LandClass.GetAfforestHumanNeeded() == 0)
                {
                        progressBar3.setValue(100);
                }
                else
                {

progressBar3.setValue((int)(100*LandClass.GetAfforest()));
                }
                if(LandClass.GetWaterHumanNeeded() == 0)
                {
                        progressBar2.setValue(100);
                }
                else
                {

progressBar2.setValue((int)(100*LandClass.GetWaterShe()));
                }
        }

        /**
         * LandScr overrides dispose so it can clean up the
         * component list.
         */
        public void dispose()
        {
                super.dispose();
                components.dispose();
```

```
}



private void label19_mouseEnter(Object source, Event e)
{
        label19.setBackColor(Color.BROWN);
}

private void label19_mouseLeave(Object source, Event e)
{
        label19.setBackColor(Color.BLACK);
}

private void label19_click(Object source, Event e)
{
        this.dispose();
        Application.run(new NewStart());
}



 private void GreenUp_click(Object source, Event e)
 {
         if(GreenAvail > 0 && GreenYetBuy >0)
         {
         LandClass.FoodLand = LandClass.FoodLand + 1;
         GreenBought = GreenBought + 1;
         GrBoughtTxt.setText(""+GreenBought);
         LandClass.SetGreenBought(GreenBought);
         GreenYetBuy = GreenYetBuy - 1;
         GreenAvail = GreenAvail - 1;
         GrAvaiTxt.setText(""+GreenAvail);
         GrYet2BTxt.setText(""+GreenYetBuy);
         LandClass.SetGreenAvail(GreenAvail);
         LandClass.SetGreenYet(GreenYetBuy);
         Money.TotalMoney = Money.TotalMoney - GreenCost;
         MoneyTxt.setText(""+Money.TotalMoney);

LandClass.SetLndHumanNeeded(LandClass.GetLndHumanNeeded()+2);

LandClass.SetWaterHumanNeeded(LandClass.GetWaterHumanNeeded()+2);

LandClass.SetAfforestHumanNeeded(LandClass.GetAfforestHumanNeeded()+2);
```

```
  LandClass.SetLanRe((float)LandClass.GetLndHumanAssighned()/(float)LandClass.Ge
tLndHumanNeeded());
                        if(LandClass.GetLanRe() > 1)
                        {
                                LandClass.SetLanRe(1);
                        }

        LandClass.SetAfforest((float)LandClass.GetAfforestHumanAssighned()/(float)La
ndClass.GetAfforestHumanNeeded());
                        if(LandClass.GetAfforest() > 1)
                        {
                                LandClass.SetAfforest(1);
                        }

        LandClass.SetWaterShe((float)LandClass.GetWaterHumanAssighned()/(float)La
ndClass.GetWaterHumanNeeded());
                        if(LandClass.GetWaterShe() > 1)
                        {
                                LandClass.SetWaterShe(1);
                        }
                        progressBar1.setValue((int)(100*LandClass.GetLanRe()));
                        progressBar3.setValue((int)(100*LandClass.GetAfforest()));
                        progressBar2.setValue((int)(100*LandClass.GetWaterShe()));
                        }
                }


                private void GreenDown_click(Object source, Event e)
                {
                        if(GreenBought > 0)
                        {
                        LandClass.FoodLand = LandClass.FoodLand - 1;
                        if(ComuScr.LndUsedFoodPro == GreenBought)
                        {
                                ComuScr.LndUsedFoodPro = ComuScr.LndUsedFoodPro -
 1;
                                FoodProcessing.HumanNeeded=
 FoodProcessing.HumanNeeded - 2;
                                FoodProcessing.SetLandUsed();
                                FoodProcessing.SetFoodProduced();
                                FoodProcessing.SetDietLevel();
                        }
                        GreenBought = GreenBought - 1;
                        GrBoughtTxt.setText(""+GreenBought);
                        LandClass.SetGreenBought(GreenBought);
                        GreenYetBuy = GreenYetBuy + 1;
                        GrYet2BTxt.setText(""+GreenYetBuy);
                        LandClass.SetGreenYet(GreenYetBuy);
                        GreenAvail = GreenAvail + 1;
                        GrAvaiTxt.setText(""+GreenAvail);
                        LandClass.SetGreenAvail(GreenAvail);
                        Money.TotalMoney = Money.TotalMoney + (GreenCost/2);
                        MoneyTxt.setText(""+Money.TotalMoney);

        LandClass.SetLndHumanNeeded(LandClass.GetLndHumanNeeded()-2);

        LandClass.SetWaterHumanNeeded(LandClass.GetWaterHumanNeeded()-2);

        LandClass.SetAfforestHumanNeeded(LandClass.GetAfforestHumanNeeded()-2);

        LandClass.SetLanRe((float)LandClass.GetLndHumanAssighned()/(float)LandCla
ss.GetLndHumanNeeded());
                        if(LandClass.GetLanRe() > 1)
                        {
                                LandClass.SetLanRe(1);
                        }

        LandClass.SetAfforest((float)LandClass.GetAfforestHumanAssighned()/(float)La
ndClass.GetAfforestHumanNeeded());
                        if(LandClass.GetAfforest() > 1)
                        {
                                LandClass.SetAfforest(1);
                        }

        LandClass.SetWaterShe((float)LandClass.GetWaterHumanAssighned()/(float)La
ndClass.GetWaterHumanNeeded());
                        if(LandClass.GetWaterShe() > 1)
                        {
                                LandClass.SetWaterShe(1);
                        }
                        progressBar1.setValue((int)(100*LandClass.GetLanRe()));
                        progressBar2.setValue((int)(100*LandClass.GetAfforest()));
                        progressBar3.setValue((int)(100*LandClass.GetWaterShe()));
                        }

                }


                private void GreenMonUp_click(Object source, Event e)
                {
                        GreenCost = GreenCost+1000;
        LandClass.SetGreenCost(GreenCost);
        GreenAvail=GreenAvail + 10;
        LandClass.SetGreenAvail(GreenAvail);
        GrCstTxt.setText(""+GreenCost);
        GrAvaiTxt.setText(""+GreenAvail);
}

private void GreenMonDown_click(Object source, Event e)
{
        GreenCost = GreenCost-1000;
        LandClass.SetGreenCost(GreenCost);
        GreenAvail=GreenAvail - 10;
        LandClass.SetGreenAvail(GreenAvail);
        GrCstTxt.setText(""+GreenCost);
        GrAvaiTxt.setText(""+GreenAvail);
}

private void CityMonUp_click(Object source, Event e)
{
        CityCost = CityCost+1000;
        LandClass.SetCityCost(CityCost);
        CityAvail=CityAvail + 10;
        LandClass.SetCityAvail(GreenAvail);
        CityCstTxt.setText(""+CityCost);
        CityAvaiTxt.setText(""+CityAvail);
}

private void CityMonDown_click(Object source, Event e)
{
        CityCost = CityCost-1000;
        LandClass.SetCityCost(CityCost);
        CityAvail=CityAvail - 10;
        LandClass.SetCityAvail(GreenAvail);
        CityCstTxt.setText(""+CityCost);
        CityAvaiTxt.setText(""+CityAvail);
}

private void CityUp_click(Object source, Event e)
{
        if(CityAvail > 0 && CityYetBuy > 0)
        {
        CityBought = CityBought + 1;
        CityBoughtTxt.setText(""+CityBought);
        LandClass.SetCityBought(CityBought);
        CityYetBuy =CityYetBuy - 1;
        CityAvail = CityAvail - 1;
```

```
                        CityAvaiTxt.setText(""+CityAvail);
                        CityYet2Btxt.setText(""+CityYetBuy);
                        LandClass.SetCityAvail(CityAvail);
                        LandClass.SetCityYet(CityYetBuy);
                        Money.TotalMoney = Money.TotalMoney - CityCost;
                        MoneyTxt.setText(""+Money.TotalMoney);

        LandClass.SetLndHumanNeeded(LandClass.GetLndHumanNceded()+2);

        LandClass.SetWaterHumanNeeded(LandClass.GetWaterHumanNeeded()+2);

        LandClass.SetAfforestHumanNeeded(LandClass.GetAfforestHumanNeeded()+2);

        LandClass.SetLanRe((float)LandClass.GetLndHumanAssighned()/(float)LandCla
ss.GetLndHumanNeeded());
                        if(LandClass.GetLanRe() > 1)
                        {
                                LandClass.SetLanRe(1);
                        }

        LandClass.SetAfforest((float)LandClass.GetAfforestHumanAssighned()/(float)La
ndClass.GetAfforestHumanNeeded());
                        if(LandClass.GetAfforest() > 1)
                        {
                                LandClass.SetAfforest(1);
                        }

        LandClass.SetWaterShe((float)LandClass.GetWaterHumanAssighned()/(float)La
ndClass.GetWaterHumanNeeded());
                        if(LandClass.GetWaterShe() > 1)
                        {
                                LandClass.SetWaterShe(1);
                        }
                        progressBar1.setValue((int)(100*LandClass.GetLanRe()));
                        progressBar3.setValue((int)(100*LandClass.GetAfforest()));
                        progressBar2.setValue((int)(100*LandClass.GetWaterShe()));
                        }
                }


                private void CityDown_click(Object source, Event e)
                {
                        if(CityBought > 0 )
                        {
                        CityBought = CityBought - 1;
                        CityBoughtTxt.setText(""+CityBought);
                        LandClass.SetCityBought(CityBought);
                        CityYetBuy =CityYetBuy + 1;
                        CityAvail = CityAvail + 1;
                        CityAvaiTxt.setText(""+CityAvail);
                        CityYet2Btxt.setText(""+CityYetBuy);
                        LandClass.SetCityAvail(CityAvail);
                        LandClass.SetCityYet(CityYetBuy);
                        Money.TotalMoney = Money.TotalMoney +(CityCost/2);
                        MoneyTxt.setText(""+Money.TotalMoney);

        LandClass.SetLndHumanNeeded(LandClass.GetLndHumanNeeded()-2);

        LandClass.SetWaterHumanNeeded(LandClass.GetWaterHumanNeeded()-2);

        LandClass.SetAfforestHumanNeeded(LandClass.GetAfforestHumanNeeded()-2);

        LandClass.SetLanRe((float)LandClass.GetLndHumanAssighned()/(float)LandCla
ss.GetLndHumanNeeded());
                        if(LandClass.GetLanRe() > 1)
                        {
                                LandClass.SetLanRe(1);
                        }

        LandClass.SetAfforest((float)LandClass.GetAfforestHumanAssighned()/(float)La
ndClass.GetAfforestHumanNeeded());
                        if(LandClass.GetAfforest() > 1)
                        {
                                LandClass.SetAfforest(1);
                        }

        LandClass.SetWaterShe((float)LandClass.GetWaterHumanAssighned()/(float)La
ndClass.GetWaterHumanNeeded());
                        if(LandClass.GetWaterShe() > 1)
                        {
                                LandClass.SetWaterShe(1);
                        }
                        progressBar1.setValue((int)(100*LandClass.GetLanRe()));
                        progressBar2.setValue((int)(100*LandClass.GetAfforest()));
                        progressBar3.setValue((int)(100*LandClass.GetWaterShe()));
                        }
                }


                private void ReclaUp_click(Object source, Event e)
                {
                    if(LandClass.GetLndHumanAssighned() !=
LandClass.GetLndHumanNeeded())
                    {
                            Money.Land = Money.Land+500;
                            AmoLand.setText(""+Money.Land);

        LandClass.SetLndHumanAssighned(LandClass.GetLndHumanAssighned()+1);
                    RecalTxt.setText(""+LandClass.GetLndHumanAssighned());

        LandClass.SetLanRe((float)LandClass.GetLndHumanAssighned()/(float)LandCla
ss.GetLndHumanNeeded());
                    if(LandClass.GetLanRe() > 1)
                    {
                            LandClass.SetLanRe(1);
                    }
                    progressBar1.setValue((int)(100*LandClass.GetLanRe()));
                    }
                }

                private void ReclaDown_click(Object source, Event e)
                {
                    if(LandClass.GetLndHumanAssighned() > 0)
                    {
                            Money.Land = Money.Land-500;
                            AmoLand.setText(""+Money.Land);

        LandClass.SetLndHumanAssighned(LandClass.GetLndHumanAssighned()-1);
                    RecalTxt.setText(""+LandClass.GetLndHumanAssighned());

        LandClass.SetLanRe((float)LandClass.GetLndHumanAssighned()/(float)LandCla
ss.GetLndHumanNeeded());
                    if(LandClass.GetLanRe() > 1)
                    {
                            LandClass.SetLanRe(1);
                    }
                    progressBar1.setValue((int)(100*LandClass.GetLanRe()));
                    }
                }

                private void WatSheUp_click(Object source, Event e)
                {
                    if(LandClass.GetWaterHumanAssighned() !=
LandClass.GetWaterHumanNeeded())
                    {
                            Money.Land = Money.Land+500;
                            AmoLand.setText(""+Money.Land);
        LandClass.SetWaterHumanAssighned(LandClass.GetWaterHumanAssighned()+1
);
                        WatSheTxt.setText(""+LandClass.GetWaterHumanAssighned());

        LandClass.SetWaterShe((float)LandClass.GetWaterHumanAssighned()/(float)La
ndClass.GetWaterHumanNeeded());
                        if(LandClass.GetWaterShe() > 1)
                        {
                                LandClass.SetWaterShe(1);
                        }
                        progressBar2.setValue((int)(100*LandClass.GetWaterShe()));
                        }
                }

                private void WatSheDown_click(Object source, Event e)
                {
                        if(LandClass.GetWaterHumanAssighned() > 0)
                        {
                                Money.Land = Money.Land-500;
                                AmoLand.setText(""+Money.Land);

        LandClass.SetWaterHumanAssighned(LandClass.GetWaterHumanAssighned()-
1);
                        WatSheTxt.setText(""+LandClass.GetWaterHumanAssighned());

        LandClass.SetWaterShe((float)LandClass.GetWaterHumanAssighned()/(float)La
ndClass.GetWaterHumanNeeded());
                        if(LandClass.GetWaterShe() > 1)
                        {
                                LandClass.SetWaterShe(1);
                        }
                        progressBar2.setValue((int)(100*LandClass.GetWaterShe()));
                        }
                }

                private void AfforUp_click(Object source, Event e)
                {
                        if(LandClass.GetAfforestHumanAssighned() !=
LandClass.GetAfforestHumanNeeded())
                        {
                                Money.Land = Money.Land+500;
                                AmoLand.setText(""+Money.Land);

        LandClass.SetAfforestHumanAssighned(LandClass.GetAfforestHumanAssighned
()+1);
```

```
                        AfforTxt.setText(""+LandClass.GetAfforestHumanAssighned());

        LandClass.SetAfforest((float)LandClass.GetAfforestHumanAssighned()/(float)La
ndClass.GetAfforestHumanNeeded());
                        if(LandClass.GetAfforest() > 1)
                        {
                                LandClass.SetAfforest(1);
                        }
                        progressBar3.setValue((int)(100*LandClass.GetAfforest()));
                        }
                }

                private void AfforDown_click(Object source, Event e)
                {
                        if(LandClass.GetAfforestHumanAssighned() > 0)
                        {
                                Money.Land = Money.Land-500;
                                AmoLand.setText(""+Money.Land);

        LandClass.SetAfforestHumanAssighned(LandClass.GetAfforestHumanAssighned
()-1);
                        AfforTxt.setText(""+LandClass.GetAfforestHumanAssighned());

        LandClass.SetAfforest((float)LandClass.GetAfforestHumanAssighned()/(float)La
ndClass.GetAfforestHumanNeeded());
                        if(LandClass.GetAfforest() > 1)
                        {
                                LandClass.SetAfforest(1);
                        }
                        progressBar3.setValue((int)(100*LandClass.GetAfforest()));
                        }
                }

                /**
                 * NOTE: The following code is required by the Visual J++ form
                 * designer.  It can be modified using the form editor.  Do not
                 * modify it using the code editor.
                 */
                Container components = new Container();
                PictureBox pictureBox1 = new PictureBox();
                Label label1 = new Label();
                static Label MoneyTxt = new Label();
                static Label MonthTxt = new Label();
                static Label YearTxt = new Label();
                Label label5 = new Label();
                Label label6 = new Label();
Label label7 = new Label();
Edit GrBoughtTxt = new Edit();
Button GreenDown = new Button();
Edit CityBoughtTxt = new Edit();
Button CityUp = new Button();
Label label8 = new Label();
Edit GrYet2BTxt = new Edit();
Edit GrCstTxt = new Edit();
Edit CityYet2Btxt = new Edit();
Label label10 = new Label();
Label label9 = new Label();
Edit GrAvaiTxt = new Edit();
Edit CityCstTxt = new Edit();
Edit CityAvaiTxt = new Edit();
Button GreenMonUp = new Button();
Button GreenMonDown = new Button();
Label label11 = new Label();
Label label12 = new Label();
Label label13 = new Label();
ProgressBar progressBar1 = new ProgressBar();
ProgressBar progressBar2 = new ProgressBar();
ProgressBar progressBar3 = new ProgressBar();
Label label14 = new Label();
Label label15 = new Label();
Label label16 = new Label();
Edit RecalTxt = new Edit();
Edit WatSheTxt = new Edit();
Edit AfforTxt = new Edit();
static Label PopTxt = new Label();
Label label18 = new Label();
Edit AmoLand = new Edit();
Label label19 = new Label();
Button GreenUp = new Button();
Button CityDown = new Button();
Button CityMonDown = new Button();
Button CityMonUp = new Button();
Button ReclaUp = new Button();
Button ReclaDown = new Button();
Button WatSheDown = new Button();
Button WatSheUp = new Button();
Button AfforUp = new Button();
Button AfforDown = new Button();

private void initForm()
{
        // NOTE:  This form is storing resource information in an
```

```
                         // external file.  Do not modify the string parameter to any
                         // resources.getObject() function call.  For example, do not
                         // modify "foo1_location" in the following line of code
                         // even if the name of the Foo object changes:
                         //
foo1.setLocation((Point)resources.getObject("foo1_location"));

                         IResourceManager resources = new ResourceManager(this,
"LandScr");
                         this.setBackColor(Color.BLACK);
                         this.setFont(new Font("MS Sans Serif", 10.0f, FontSize.POINTS,
FontWeight.NORMAL, false, false, false, CharacterSet.DEFAULT, 0));
                         this.setText("Renaissance");
                         this.setAutoScaleBaseSize(16);
                         this.setBorderStyle(FormBorderStyle.NONE);
                         this.setClientSize(new Point(800, 572));
                         this.setStartPosition(FormStartPosition.CENTER_SCREEN);
                         this.setWindowState(FormWindowState.MAXIMIZED);

                         pictureBox1.setLocation(new Point(0, 64));
                         pictureBox1.setSize(new Point(328, 272));
                         pictureBox1.setTabIndex(0);
                         pictureBox1.setTabStop(false);
                         pictureBox1.setText("pictureBox1");

      pictureBox1.setImage((Bitmap)resources.getObject("pictureBox1_image"));

      pictureBox1.setSizeMode(PictureBoxSizeMode.STRETCH_IMAGE);

                         label1.setFont(new Font("Arial Black", 20.0f, FontSize.POINTS,
FontWeight.BOLD, false, false, false, CharacterSet.DEFAULT, 0));
                         label1.setForeColor(new Color(224, 224, 224));
                         label1.setLocation(new Point(512, 80));
                         label1.setSize(new Point(208, 40));
                         label1.setTabIndex(1);
                         label1.setTabStop(false);
                         label1.setText("Land");
                         label1.setTextAlign(HorizontalAlignment.CENTER);

                         MoneyTxt.setBackColor(Color.WHITE);
                         MoneyTxt.setForeColor(new Color(0, 192, 0));
                         MoneyTxt.setLocation(new Point(16, 16));
                         MoneyTxt.setSize(new Point(112, 24));
                         MoneyTxt.setTabIndex(2);
                         MoneyTxt.setTabStop(false);
                         MoneyTxt.setText("Rs.000000");
```

```
                    MonthTxt.setBackColor(Color.WHITE);
                    MonthTxt.setFont(new Font("Copperplate Gothic Bold", 11.0f,
FontSize.POINTS, FontWeight.NORMAL, false, false, false, CharacterSet.DEFAULT,
0));
                    MonthTxt.setForeColor(new Color(0, 0, 128));
                    MonthTxt.setLocation(new Point(560, 24));
                    MonthTxt.setSize(new Point(136, 16));
                    MonthTxt.setTabIndex(4);
                    MonthTxt.setTabStop(false);
                    MonthTxt.setText("FEBRUARY");

                    YearTxt.setBackColor(Color.WHITE);
                    YearTxt.setFont(new Font("Copperplate Gothic Bold", 11.0f,
FontSize.POINTS, FontWeight.NORMAL, false, false, false, CharacterSet.DEFAULT,
0));
                    YearTxt.setForeColor(Color.BLUE);
                    YearTxt.setLocation(new Point(720, 24));
                    YearTxt.setSize(new Point(64, 16));
                    YearTxt.setTabIndex(5);
                    YearTxt.setTabStop(false);
                    YearTxt.setText("Year");

                    label5.setBackColor(new Color(0, 192, 0));
                    label5.setFont(new Font("Viner Hand ITC", 12.0f,
FontSize.POINTS, FontWeight.BOLD, false, false, false, CharacterSet.DEFAULT, 0));
                    label5.setForeColor(new Color(64, 64, 0));
                    label5.setLocation(new Point(464, 136));
                    label5.setSize(new Point(144, 24));
                    label5.setTabIndex(6);
                    label5.setTabStop(false);
                    label5.setText("Green Belt");
                    label5.setTextAlign(HorizontalAlignment.CENTER);

                    label6.setBackColor(new Color(64, 64, 0));
                    label6.setFont(new Font("Viner Hand ITC", 12.0f,
FontSize.POINTS, FontWeight.BOLD, false, false, false, CharacterSet.DEFAULT, 0));
                    label6.setForeColor(new Color(0, 192, 0));
                    label6.setLocation(new Point(632, 136));
                    label6.setSize(new Point(144, 24));
                    label6.setTabIndex(7);
                    label6.setTabStop(false);
                    label6.setText("City Area");
                    label6.setTextAlign(HorizontalAlignment.CENTER);

                    label7.setBackColor(Color.WHITE);
```

```
                label7.setFont(new Font("Monotype Corsiva", 16.0f,
FontSize.POINTS, FontWeight.NORMAL, false, false, false, CharacterSet.DEFAULT,
0));
                label7.setLocation(new Point(336, 184));
                label7.setSize(new Point(120, 24));
                label7.setTabIndex(8);
                label7.setTabStop(false);
                label7.setText("Bought");
                label7.setTextAlign(HorizontalAlignment.CENTER);

                GrBoughtTxt.setBackColor(new Color(0, 192, 0));
                GrBoughtTxt.setEnabled(false);
                GrBoughtTxt.setForeColor(new Color(64, 64, 0));
                GrBoughtTxt.setLocation(new Point(464, 184));
                GrBoughtTxt.setSize(new Point(120, 23));
                GrBoughtTxt.setTabIndex(9);
                GrBoughtTxt.setText("000000");
                GrBoughtTxt.setTextAlign(HorizontalAlignment.CENTER);

                GreenDown.setFont(new Font("Marlett", 12.0f,
FontSize.CHARACTERHEIGHT, FontWeight.NORMAL, false, false, false,
CharacterSet.DEFAULT, 0));
                GreenDown.setLocation(new Point(592, 216));
                GreenDown.setSize(new Point(16, 32));
                GreenDown.setTabIndex(34);
                GreenDown.setText("u");
                GreenDown.addOnClick(new
EventHandler(this.GreenDown_click));

                CityBoughtTxt.setBackColor(new Color(64, 64, 0));
                CityBoughtTxt.setEnabled(false);
                CityBoughtTxt.setForeColor(new Color(0, 192, 0));
                CityBoughtTxt.setLocation(new Point(632, 184));
                CityBoughtTxt.setSize(new Point(120, 23));
                CityBoughtTxt.setTabIndex(10);
                CityBoughtTxt.setText("000000");
                CityBoughtTxt.setTextAlign(HorizontalAlignment.CENTER);

                CityUp.setFont(new Font("Marlett", 12.0f,
FontSize.CHARACTERHEIGHT, FontWeight.NORMAL, false, false, false,
CharacterSet.DEFAULT, 0));
                CityUp.setLocation(new Point(760, 184));
                CityUp.setSize(new Point(16, 32));
                CityUp.setTabIndex(22);
                CityUp.setText("t");
                CityUp.addOnClick(new EventHandler(this.CityUp_click));
```

```
                        label8.setBackColor(Color.WHITE);
                        label8.setFont(new Font("Monotype Corsiva", 16.0f,
FontSize.POINTS, FontWeight.NORMAL, false, false, false, CharacterSet.DEFAULT,
0));
                        label8.setLocation(new Point(336, 224));
                        label8.setSize(new Point(120, 24));
                        label8.setTabIndex(11);
                        label8.setTabStop(false);
                        label8.setText("Yet To Buy");
                        label8.setTextAlign(HorizontalAlignment.CENTER);

                        GrYet2BTxt.setBackColor(new Color(0, 192, 0));
                        GrYet2BTxt.setEnabled(false);
                        GrYet2BTxt.setForeColor(new Color(64, 64, 0));
                        GrYet2BTxt.setLocation(new Point(464, 224));
                        GrYet2BTxt.setSize(new Point(120, 23));
                        GrYet2BTxt.setTabIndex(12);
                        GrYet2BTxt.setText("000000");
                        GrYet2BTxt.setTextAlign(HorizontalAlignment.CENTER);

                        GrCstTxt.setBackColor(new Color(0, 192, 0));
                        GrCstTxt.setEnabled(false);
                        GrCstTxt.setForeColor(new Color(64, 64, 0));
                        GrCstTxt.setLocation(new Point(464, 304));
                        GrCstTxt.setSize(new Point(120, 23));
                        GrCstTxt.setTabIndex(19);
                        GrCstTxt.setText("000000");
                        GrCstTxt.setTextAlign(HorizontalAlignment.CENTER);

                        CityYet2Btxt.setBackColor(new Color(64, 64, 0));
                        CityYet2Btxt.setEnabled(false);
                        CityYet2Btxt.setForeColor(new Color(0, 192, 0));
                        CityYet2Btxt.setLocation(new Point(632, 224));
                        CityYet2Btxt.setSize(new Point(120, 23));
                        CityYet2Btxt.setTabIndex(13);
                        CityYet2Btxt.setText("000000");
                        CityYet2Btxt.setTextAlign(HorizontalAlignment.CENTER);

                        label10.setBackColor(Color.WHITE);
                        label10.setFont(new Font("Monotype Corsiva", 16.0f,
FontSize.POINTS, FontWeight.NORMAL, false, false, false, CharacterSet.DEFAULT,
0));
                        label10.setLocation(new Point(336, 304));
                        label10.setSize(new Point(120, 24));
                        label10.setTabIndex(18);
                        label10.setTabStop(false);
                        label10.setText("Cost Of Land");

                        label9.setBackColor(Color.WHITE);
                        label9.setFont(new Font("Monotype Corsiva", 16.0f,
FontSize.POINTS, FontWeight.NORMAL, false, false, false, CharacterSet.DEFAULT,
0));
                        label9.setLocation(new Point(336, 264));
                        label9.setSize(new Point(120, 24));
                        label9.setTabIndex(14);
                        label9.setTabStop(false);
                        label9.setText("Available");
                        label9.setTextAlign(HorizontalAlignment.CENTER);

                        GrAvaiTxt.setBackColor(new Color(0, 192, 0));
                        GrAvaiTxt.setEnabled(false);
                        GrAvaiTxt.setForeColor(new Color(64, 64, 0));
                        GrAvaiTxt.setLocation(new Point(464, 264));
                        GrAvaiTxt.setSize(new Point(120, 23));
                        GrAvaiTxt.setTabIndex(15);
                        GrAvaiTxt.setText("000000");
                        GrAvaiTxt.setTextAlign(HorizontalAlignment.CENTER);

                        CityCstTxt.setBackColor(new Color(64, 64, 0));
                        CityCstTxt.setEnabled(false);
                        CityCstTxt.setForeColor(new Color(0, 192, 0));
                        CityCstTxt.setLocation(new Point(632, 304));
                        CityCstTxt.setSize(new Point(120, 23));
                        CityCstTxt.setTabIndex(20);
                        CityCstTxt.setText("000000");
                        CityCstTxt.setTextAlign(HorizontalAlignment.CENTER);

                        CityAvaiTxt.setBackColor(new Color(64, 64, 0));
                        CityAvaiTxt.setEnabled(false);
                        CityAvaiTxt.setForeColor(new Color(0, 192, 0));
                        CityAvaiTxt.setLocation(new Point(632, 264));
                        CityAvaiTxt.setSize(new Point(120, 23));
                        CityAvaiTxt.setTabIndex(16);
                        CityAvaiTxt.setText("000000");
                        CityAvaiTxt.setTextAlign(HorizontalAlignment.CENTER);

                        GreenMonUp.setFont(new Font("Marlett", 12.0f,
 FontSize.CHARACTERHEIGHT, FontWeight.NORMAL, false, false, false,
 CharacterSet.DEFAULT, 0));
                        GreenMonUp.setLocation(new Point(592, 304));
                        GreenMonUp.setSize(new Point(16, 12));
                        GreenMonUp.setTabIndex(28);
                        GreenMonUp.setText("t");
                        GreenMonUp.addOnClick(new
EventHandler(this.GreenMonUp_click));

                        GreenMonDown.setFont(new Font("Marlett", 12.0f,
FontSize.CHARACTERHEIGHT, FontWeight.NORMAL, false, false, false,
CharacterSet.DEFAULT, 0));
                        GreenMonDown.setLocation(new Point(592, 316));
                        GreenMonDown.setSize(new Point(16, 12));
                        GreenMonDown.setTabIndex(33);
                        GreenMonDown.setText("u");
                        GreenMonDown.addOnClick(new
EventHandler(this.GreenMonDown_click));

                        label11.setFont(new Font("Imprint MT Shadow", 16.0f,
FontSize.POINTS, FontWeight.BOLD, false, false, false, CharacterSet.DEFAULT, 0));
                        label11.setForeColor(Color.LIGHTGRAY);
                        label11.setLocation(new Point(72, 352));
                        label11.setSize(new Point(192, 32));
                        label11.setTabIndex(35);
                        label11.setTabStop(false);
                        label11.setText("Land Reclamation");
                        label11.setTextAlign(HorizontalAlignment.RIGHT);

                        label12.setBackColor(Color.BLACK);
                        label12.setFont(new Font("Imprint MT Shadow", 14.0f,
FontSize.POINTS, FontWeight.BOLD, false, false, false, CharacterSet.DEFAULT, 0));
                        label12.setForeColor(Color.CYAN);
                        label12.setLocation(new Point(32, 400));
                        label12.setSize(new Point(232, 32));
                        label12.setTabIndex(36);
                        label12.setTabStop(false);
                        label12.setText("Water Shed Management");
                        label12.setTextAlign(HorizontalAlignment.RIGHT);

                        label13.setFont(new Font("Imprint MT Shadow", 16.0f,
FontSize.POINTS, FontWeight.BOLD, false, false, false, CharacterSet.DEFAULT, 0));
                        label13.setForeColor(new Color(0, 128, 0));
                        label13.setLocation(new Point(64, 448));
                        label13.setSize(new Point(200, 32));
                        label13.setTabIndex(37);
                        label13.setTabStop(false);
                        label13.setText("Afforestation");
                        label13.setTextAlign(HorizontalAlignment.RIGHT);
```

```
progressBar1.setLocation(new Point(280, 352));
progressBar1.setSize(new Point(216, 24));
progressBar1.setTabIndex(38);
progressBar1.setText("progressBar1");

progressBar2.setLocation(new Point(280, 400));
progressBar2.setSize(new Point(216, 24));
progressBar2.setTabIndex(39);
progressBar2.setText("progressBar2");

progressBar3.setLocation(new Point(280, 448));
progressBar3.setSize(new Point(216, 24));
progressBar3.setTabIndex(40);
progressBar3.setText("progressBar3");

label14.setForeColor(Color.WHITE);
label14.setLocation(new Point(512, 352));
label14.setSize(new Point(88, 24));
label14.setTabIndex(41);
label14.setTabStop(false);
label14.setText("Man Power");

label15.setBackColor(Color.BLACK);
label15.setForeColor(Color.WHITE);
label15.setLocation(new Point(512, 400));
label15.setSize(new Point(88, 24));
label15.setTabIndex(42);
label15.setTabStop(false);
label15.setText("Man Power");

label16.setForeColor(Color.WHITE);
label16.setLocation(new Point(512, 448));
label16.setSize(new Point(88, 24));
label16.setTabIndex(43);
label16.setTabStop(false);
label16.setText("Man Power");

RecalTxt.setEnabled(false);
RecalTxt.setLocation(new Point(608, 352));
RecalTxt.setSize(new Point(72, 23));
RecalTxt.setTabIndex(44);
RecalTxt.setText("000");
RecalTxt.setTextAlign(HorizontalAlignment.CENTER);

WatSheTxt.setEnabled(false);
WatSheTxt.setLocation(new Point(608, 400));
```

```
                WatSheTxt.setSize(new Point(72, 23));
                WatSheTxt.setTabIndex(45);
                WatSheTxt.setText("000");
                WatSheTxt.setTextAlign(HorizontalAlignment.CENTER);

                AfforTxt.setEnabled(false);
                AfforTxt.setLocation(new Point(608, 448));
                AfforTxt.setSize(new Point(72, 23));
                AfforTxt.setTabIndex(46);
                AfforTxt.setText("000");
                AfforTxt.setTextAlign(HorizontalAlignment.CENTER);

                PopTxt.setBackColor(Color.WINDOW);
                PopTxt.setLocation(new Point(316, 16));
                PopTxt.setSize(new Point(168, 24));
                PopTxt.setTabIndex(3);
                PopTxt.setTabStop(false);
                PopTxt.setText("PopTxt");

                label18.setFont(new Font("Imprint MT Shadow", 16.0f,
FontSize.POINTS, FontWeight.NORMAL, false, false, false, CharacterSet.DEFAULT,
0));
                label18.setForeColor(new Color(192, 0, 0));
                label18.setLocation(new Point(392, 504));
                label18.setSize(new Point(312, 32));
                label18.setTabIndex(47);
                label18.setTabStop(false);
                label18.setText("Amount Spend");
                label18.setTextAlign(HorizontalAlignment.RIGHT);

                AmoLand.setEnabled(false);
                AmoLand.setLocation(new Point(728, 504));
                AmoLand.setSize(new Point(64, 23));
                AmoLand.setTabIndex(48);
                AmoLand.setText("");

                label19.setCursor(Cursor.UPARROW);
                label19.setFont(new Font("Pepita MT", 16.0f, FontSize.POINTS,
FontWeight.BOLD, false, false, false, CharacterSet.DEFAULT, 0));
                label19.setForeColor(Color.LIGHTGRAY);
                label19.setLocation(new Point(16, 536));
                label19.setSize(new Point(120, 24));
                label19.setTabIndex(17);
                label19.setTabStop(false);
                label19.setText("Back to main");
                label19.addOnClick(new EventHandler(this.label19_click));
```

```
                    label19.addOnMouseEnter(new
EventHandler(this.label19_mouseEnter));
                    label19.addOnMouseLeave(new
EventHandler(this.label19_mouseLeave));

                    GreenUp.setFont(new Font("Marlett", 12.0f,
FontSize.CHARACTERHEIGHT, FontWeight.NORMAL, false, false, false,
CharacterSet.DEFAULT, 0));
                    GreenUp.setLocation(new Point(592, 184));
                    GreenUp.setSize(new Point(16, 32));
                    GreenUp.setTabIndex(23);
                    GreenUp.setText("t");
                    GreenUp.addOnClick(new EventHandler(this.GreenUp_click));

                    CityDown.setFont(new Font("Marlett", 12.0f,
FontSize.CHARACTERHEIGHT, FontWeight.NORMAL, false, false, false,
CharacterSet.DEFAULT, 0));
                    CityDown.setLocation(new Point(760, 216));
                    CityDown.setSize(new Point(16, 32));
                    CityDown.setTabIndex(21);
                    CityDown.setText("u");
                    CityDown.addOnClick(new EventHandler(this.CityDown_click));

                    CityMonDown.setFont(new Font("Marlett", 12.0f,
FontSize.CHARACTERHEIGHT, FontWeight.NORMAL, false, false, false,
CharacterSet.DEFAULT, 0));
                    CityMonDown.setLocation(new Point(760, 316));
                    CityMonDown.setSize(new Point(16, 12));
                    CityMonDown.setTabIndex(29);
                    CityMonDown.setText("u");
                    CityMonDown.addOnClick(new
EventHandler(this.CityMonDown_click));

                    CityMonUp.setFont(new Font("Marlett", 12.0f,
FontSize.CHARACTERHEIGHT, FontWeight.NORMAL, false, false, false,
CharacterSet.DEFAULT, 0));
                    CityMonUp.setLocation(new Point(760, 304));
                    CityMonUp.setSize(new Point(16, 12));
                    CityMonUp.setTabIndex(24);
                    CityMonUp.setText("t");
                    CityMonUp.addOnClick(new
EventHandler(this.CityMonUp_click));

                    ReclaUp.setFont(new Font("Marlett", 12.0f,
FontSize.CHARACTERHEIGHT, FontWeight.NORMAL, false, false, false,
CharacterSet.DEFAULT, 0));
```

```
                        ReclaUp.setLocation(new Point(688, 352));
                        ReclaUp.setSize(new Point(16, 12));
                        ReclaUp.setTabIndex(27);
                        ReclaUp.setText("t");
                        ReclaUp.addOnClick(new EventHandler(this.ReclaUp_click));

                        ReclaDown.setFont(new Font("Marlett", 12.0f,
FontSize.CHARACTERHEIGHT, FontWeight.NORMAL, false, false, false,
CharacterSet.DEFAULT, 0));
                        ReclaDown.setLocation(new Point(688, 364));
                        ReclaDown.setSize(new Point(16, 12));
                        ReclaDown.setTabIndex(32);
                        ReclaDown.setText("u");
                        ReclaDown.addOnClick(new
EventHandler(this.ReclaDown_click));

                        WatSheDown.setFont(new Font("Marlett", 12.0f,
FontSize.CHARACTERHEIGHT, FontWeight.NORMAL, false, false, false.
CharacterSet.DEFAULT, 0));
                        WatSheDown.setLocation(new Point(688, 412));
                        WatSheDown.setSize(new Point(16, 12));
                        WatSheDown.setTabIndex(31);
                        WatSheDown.setText("u");
                        WatSheDown.addOnClick(new
EventHandler(this.WatSheDown_click));

                        WatSheUp.setFont(new Font("Marlett", 12.0f,
FontSize.CHARACTERHEIGHT, FontWeight.NORMAL, false, false, false,
CharacterSet.DEFAULT, 0));
                        WatSheUp.setLocation(new Point(688, 400));
                        WatSheUp.setSize(new Point(16, 12));
                        WatSheUp.setTabIndex(26);
                        WatSheUp.setText("t");
                        WatSheUp.addOnClick(new
EventHandler(this.WatSheUp_click));

                        AfforUp.setFont(new Font("Marlett", 12.0f,
FontSize.CHARACTERHEIGHT, FontWeight.NORMAL, false, false, false,
CharacterSet.DEFAULT, 0));
                        AfforUp.setLocation(new Point(688, 448));
                        AfforUp.setSize(new Point(16, 12));
                        AfforUp.setTabIndex(25);
                        AfforUp.setText("t");
                        AfforUp.addOnClick(new EventHandler(this.AfforUp_click));
```

```
                    AfforDown.setFont(new Font("Marlett", 12.0f,
FontSize.CHARACTERHEIGHT, FontWeight.NORMAL, false, false, false,
CharacterSet.DEFAULT, 0));
                    AfforDown.setLocation(new Point(688, 460));
                    AfforDown.setSize(new Point(16, 12));
                    AfforDown.setTabIndex(30);
                    AfforDown.setText("u");
                    AfforDown.addOnClick(new
EventHandler(this.AfforDown_click));

                    this.setNewControls(new Control[] {
                                                    AfforDown,
                                                    AfforUp,
                                                    WatSheUp,
                                                    WatSheDown,
                                                    ReclaDown,
                                                    ReclaUp,
                                                    CityMonUp,
                                                    CityMonDown,
                                                    GreenMonUp,
                                                    GreenMonDown,
                                                    CityUp,
                                                    CityDown,
                                                    GreenUp,
                                                    GreenDown,
                                                    label19,
                                                    AmoLand,
                                                    label18,
                                                    PopTxt,
                                                    AfforTxt,
                                                    WatSheTxt,
                                                    RecalTxt,
                                                    label16,
                                                    label15,
                                                    label14,
                                                    progressBar3,
                                                    progressBar2,
                                                    progressBar1,
                                                    label13,
                                                    label12,
                                                    label11,
                                                    CityCstTxt,
                                                    GrCstTxt,
                                                    label10,
                                                    CityAvaiTxt,
                                                    GrAvaiTxt,
                                                            label9,
                                                            CityYet2Btxt,
                                                            GrYet2BTxt,
                                                            label8,
                                                            CityBoughtTxt,
                                                            GrBoughtTxt,
                                                            label7,
                                                            label6,
                                                            label5,
                                                            YearTxt,
                                                            MonthTxt,
                                                            MoneyTxt,
                                                            label1,
                                                            pictureBox1});
        }

        /**
         * The main entry point for the application.
         *
         * @param args Array of parameters passed to the application
         * via the command line.
         */
        public static void main(String args[])
        {
                Application.run(new LandScr());
        }
}
```

10.2 The Sample code for a Thread Program:

```
import java.util.*;
//import java.lang.*;
public class Time extends Thread
{
    static public int seconds,minutes,hours;
    static public int month,year=1968,TimeCamp = 0;
    static public String Months = new String("FEBRUARY");
    private boolean a5;
    public void run()
    {
        a5 = true;
        Months = new String("FEBRUARY");
        month = 2;
        year = 1968;
        while( a5 = true )
        {
            AccomScr.MonthTxt.setText(Months);
            ArtCult.MonthTxt.setText(Months);
            ComuScr.MonthTxt.setText(Months);
            EduScr.MonthTxt.setText(Months);
            HealthScr.MonthTxt.setText(Months);
            InfraScr.MonthTxt.setText(Months);
            LandScr.MonthTxt.setText(Months);
            ReseScr.MonthTxt.setText(Months);
            NewStart.MonthTxt.setText(Months);
            AccomScr.YearTxt.setText(""+year);
            ArtCult.YearTxt.setText(""+year);
            ComuScr.YearTxt.setText(""+year);
            EduScr.YearTxt.setText(""+year);
            HealthScr.YearTxt.setText(""+year);
            InfraScr.YearTxt.setText(""+year);
            LandScr.YearTxt.setText(""+year);
            ReseScr.YearTxt.setText(""+year);
            NewStart.YearTxt.setText(""+year);
            HealthScr.TimeCampTxt.setText(TimeCamp+" Months");
            TimeCamp = TimeCamp +1;
            LandClass.SetCityAvail(LandClass.GetCityAvail()+5);
            LandClass.SetGreenAvail(LandClass.GetGreenAvail()+5);
            try{
                this.sleep(60000);
                }
            catch(InterruptedException x)
            {
```

11

Conclusion

Our idea in this project of resource planning software has been successfully implemented and fulfilled though the real goal lies in making an impact in the user's mind to practise humanity.

It would not be justice to expect a sea change in the user's mind after working in this project but our ultimate aim or the real conclusion of our project lies in making this idea of ours to linger in even a small part of his/her heart.

11

Bibliography

1. Holzner, Steven Mastering™ VJ++, BPB publications, New Delhi, 1997.
2. Maso, Brian VJ++ 6 From Ground Up, Tata McGRAW HILL, New Delhi, 1999.
3. H.Murray, William, H.Pappas, Chris, The VJ++ Handbook, AP Professional, USA, 1997.
4. Hyde, Paul, JAVA™ Thread Programming, The Authoritative Solution, TECHMEDIA, New Delhi, 1999.
5. Guy, Auroville – Development Perspectives, Auroville Development Group, Auroville, 1993.
6. Age Of Empires, Microsoft® Corp and Ensemble Studios.
7. http://BRAVE_HEARTS.ZZN.COM