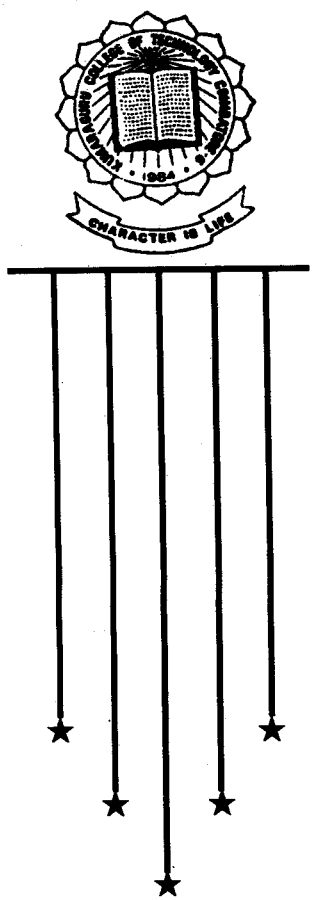
LOSE YOUR MARBLES
(COMPUTER SIMULATION AND GRAPHICS)

PROJECT REPORT



1999 - 2000

DONE AT

DHRUVA INFOTECH

SUBMITTED BY

C. B. SHALINI

GUIDED BY

INTERNAL

Mr. K.R. BASKARAN, B.E., M.S.,

EXTERNAL

Mr. MAHESH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE OF

MASTER OF SCIENCE IN

APPLIED SCIENCES-COMPUTER TECHNOLOGY

OF THE BHARATHIAR UNIVERSITY, COIMBATORE.

Department of Computer Science and Engineering

Kumaraguru College of Technology

Coimbatore - 641 006.

KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore – 641 006

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the report entitled

LOSE YOUR MARBLES

(COMPUTER SIMLATION & GRAPHICS)

has been submitted by

Mr./Ms. Miss. C.B. SHALINI

in partial fulfillment of the requirements for the award of degree of Master of Science-Applied Sciences (Computer Technology) in the Computer Science & Engineering branch of the Bharathiar University, Coimbatore-641 046 during the year 1999-2000.

(Guide) (Head of the Department) 26/4

Certified that the candidate was examined by us in the project viva- voce examination held on [redacted] and the University Register number is 9837Q0031

(Internal Examiner) (External Examiner) 28/4

DHRUVA INFOTECH (P) LTD.

Regd. Office : 946, 12th Main, 5th Cross, HAL 2nd Stage, Indiranagar, Bangalore - 560 008, INDIA.
Ph : +91 (80) 526 7091 Fax: +91 (80) 526 3281 E-Mail : info@dhruva.com Web: http://www.dhruva.com

Letter of Approval

C. B. Shalini

This is to certify that Ms. C. B. Shalini has completed all her modules towards the game project titled "Lose Your Marbles" with Dhruva Infotech Private Limited, Indiranagar, Bangalore. The project was completed during the period January 2000 to April 2000. She will be returning in May 2000 to integrate the product and optimize code.

Her work was found above satisfactory in all regards and we recommend that appropriate credit to be given to her work. This is particularly in view of the fact that she has picked up technical and team-work skills very effectively over a short span of time and is able to utilize the same in her daily work.

The project report has been reviewed by us and has been found to be satisfactory regarding the content and is ready to be submitted to the university.

Mr. Mahesh Khambadkone
C.E.O.
Dhruva Infotech Pvt. Ltd.
Bangalore

Date: 17/4/2000

DECLARATION

I here by declare that this project work entitled

"Lose Your Marbles (Computer Simulation & Graphics)"

Submitted to Kumaraguru College of Technology, Coimbatore –6, (Affiliated to Bharthiar University) is a record of original work done by me under the supervision and guidance of **Mr. K. R. Baskaran,** B.E., M.S., Senior Lecturer, Department of Computer Science & Engineering, Kumaraguru College of Technology and his project work has not formed the basis for the award of **any Degree / Diploma / Associate ship / Fellowship or similar with to any candidate of any University.**

Place : Coimbatore

Date : 28.04.2000

Shalini C.B.
Signature of the candidate

(C. B. Shalini)

Counter signed by

Signature of Staff

Mr. K. R. Baskaran
Senior Lecturer

Kumaraguru College of Technology.

ACKNOWLEDGEMENT

I wish to express my heartfelt gratitude to ***Principal Dr K.K Padmanaban***, for his encouragement and counseling. I also thank ***Prof. Dr. S. ThangaSwamy B.E (Hon's)., P.hd., Head of the Department***, Department Of Computer Science & Engineering, for his valuable guidance.

I express my gratitude to ***Mr. K. R. Baskaran, B.E. M.S., Senior Lecturer***, Department of Computer Science & Engineering, for providing me constant support and encouragement for successfully completing this project. I also express my sincere thanks to all the staff members for their guidance and advice.

There are no adequate words to express my sincere gratitude to ***Mr. Raju B. Patil, Dhruva InfoTech***, for kindly recommending me for the project work at Dhruva InfoTech.

I express my sincere thanks and gratitude to ***Mr. Mahesh, Chief Executive Officer, Dhruva InfoTech***, for providing me this opportunity to do my project in Dhruva InfoTech. I also express sincere thanks to ***Mr. Pramod Ram Prasad***, for giving me this opportunity to work under his supervision and has been a constant source of guidance and support throughout my project.

Let me admit that I really enjoyed carrying out my project work in the friendly and informal atmosphere of this group. I thank whole-heartedly all my team members and friends in this organization for their co-operation and timely help.

ABSTRACT

The project entitled, as **"Lose Your Marbles"** is an event-based multi-player game and has been designed for **Dhruva InfoTech (Pvt.) Ltd.**, Bangalore. This is a game involving marbles. The game can be played between both human and computer-simulated players against each other.

The system includes implementing

- Computer-simulated player and
- Graphics support required for the whole game.

Computer-simulated player deals with emulating a human player using Artificial Intelligence. This is implemented using Strategic AI, as the game is played based on a fixed set of well-defined rules that the computer has to follow to play the game and behavioral AI based on some primary behaviors.

Graphics support for the game is used to provide faster and good rendering effects for images. A single GIF file is used to store all the images required for the game from which images are cropped at runtime.

This game has been developed in Java due to its main feature, platform - Independent language because there should be no barrier for the game to be played across the Internet throughout the world.

CONTENTS

Chapter 1

Introduction

1.1 Organization Profile

Dhruva Infotech, a registered STP (Software Technology Park) unit, based in Bangalore, India, offers products and services in the areas of location-independence computing, Internet infrastructure-space and multimedia.

Products:

Broadly in the domain of location-independence computing, they are working on applications that utilize wireless and PDA technologies. They are setting up a permanent presence in the Valley that would help ensure better visibility and faster time-to-market. The R&D unit would be housed there with development executed in India. They also undertake collaborative development of products for the web.

Services:

Having had over 3 years experience in offshore-software development, they are now ramping operations and processes to capture a larger slice of market-share. Their strength lies in adapting emerging technologies and tools to provide better and more effective solutions to customers. Clients can leverage on their project management expertise and attractive costing, enabling solutions on strict time and quality parameters.

1.2 Overview of the Project

Lose Your Marbles is a multi-player web-based game. It is developed for ***Dhruva InfoTech (Pvt.) Ltd.***, situated in Bangalore. It is a game involving marbles. The game consists of both human and computer-simulated players. In this game players across the net can play together at the same time. This is an event-based game hence each player can play by itself.

This game includes three main modules human player, computer-simulated player and graphics support. Computer-simulated player involves emulating a human player; here the computer is made to play by itself without any human interference. This part is included in game mainly because it will make the human player's instinct to win the game.

Computer-simulated player is designed based on some fixed set of rules that the computer has to follow to play the game. These rules are just the appropriate moves the computer have to make on coming across some set of conditions. Strategic Artificial Intelligence is used to implement the game, as it is played based on a fixed set of well-defined rules that the computer has to use to play the game. Apart from this behavioral AI is also used because to play the game the computer has to look for some primary behaviors based on which the moves are made.

The game engine involves a human and a computer player playing side-by-side. The main adjective of the game is to destroy three, four or five adjacent marbles of same color in the center row. The more marbles destroyed leaves the player's grid with less number of marbles. Marbles keep rolling down into grid and the players

Chapter 2

System Analysis & Design

2.1 Existing System Limitations

The game engine of existing computer-simulated player is:

- Platform-specific because it is developed in Visual C++ and it can be run in Windows based operating system only.
- The graphics support for game is not faster and efficient.
- The moves made by the computer player become slow when played on a slower machine by which the human player can always win the game on slower machines.

2.2 User Characteristics

Since, the computer plays the module computer-simulated player itself; it does not involve any human interface. We can even say that the computer-simulated player is user-independent, but it inspires the human opponent to thrive for victory. Similarly, the other module – Graphics support for the game also requires no user input.

3. PROGRAMMING ENVIRONMENT

Chapter 3

Programming Environment

3.1 Hardware Configuration

Client Side

- Intel Pentium 100 MHz & Above
- 16 MB RAM
- Cache Memory 512 KB

3.2 Description of Software and Packages Used

Client Side

- Operating System Independent
- Java Development Kit 1.2.1 & Above
- 256-Color Graphics
- Internet & Intranet Connection Tools
- Internet Explorer 5.0 & Netscape Navigator

Distributed:

Java is designed in such a way to run across the network; hence it has extensive TCP/IP and HTTP capabilities. Though Java was primarily used as the web language, it is an excellent substitute for C++ in the corporate shops.

Dynamic:

Java programs carry with them substantial amount of run-time type information that is used to verify and resolve accesses to objects at run time. This makes it possible to dynamically link code in a safe and expedient manner.

Rich Set of Class Libraries:

Java class libraries provide high level interfaces for GUI programming, I/O, Server programming, multithreading and networking. Apart from this, the new Java media APIs support 2D and 3D graphics, playback and recording of media data and speech analysis. Together these libraries enable the development of powerful Java applications using native OS functions. Along with these interfaces Java's lightweight components provides benefits of transparency, ToolTips, and a luggable look and feel and JAR (Java Archive) file, serves as a self-contained, reusable component.

Key Concepts of Java

- Applets are platform independent, which means the same applet, can execute on PC running Windows 95 or a Mac or UNIX based system.
- Java Applet runs within the boundaries of a browser called "Java enabled Browser". Examples: Netscape Navigator, Internet Explorer.
- Look & Feel of Graphics User Interface is pluggable, i.e., the look & feel of UI can be changed according to the platform on which Java applications or applets are running.

- Stand-alone applications do not run within the limits of the browser.
- Unlike other programs that create programs that are processor specific, Java creates code for virtual machine called Byte code. The browser then converts this code into the binary code the processor understands.
- Java compiler is a special compiler; it is used to translate Java source file into a Class (byte code) file.

Applications:

Java language is broadly classified into:

- Applets (mini applications)
- GUI applications
- Command line applications (Stand alone)
- Packages (libraries)

An applet is an application designed to be transmitted over the Internet and executed by a Java-compatible Web browser. An applet is a small program that is intended not to be run on its own, but rather to be embedded inside another application and it can be downloaded across the network. Applets are accessed on an Internet server, transported over the Net, installed and run part of a web page. On the client side, applets have limited access to resources, so that it can produce an arbitrary multimedia User Interface.

The second type is a typical GUI application, such as the Windows Notepad Application, which does not require a Web browser to execute it.

The JSL includes hundreds of classes and methods grouped into six different packages:

- ***Language Support Package:*** This package consists of the basic language functions.
- ***Utilities Package:*** This package contains some of the most exciting enhancements such Calendar, Dictionary classes etc.
- ***Input / Output Package:*** This package provides support for I/O operations.
- ***Networking Package:*** This package is used to provide networking support for Java.
- ***AWT Package:*** It is one of Java's largest packages and consists of Java's support for User Interface.
- ***Applet Package:*** This packaage contains the Applet class and applet consists of several methods that gives detailed control over the execution of the applet.

Java's Support for Graphics

Graphics are at the heart of most games. Knowing this, you need to understand a certain degree of Java graphics fundamentals before you get into full-blown game graphics.

The Graphics Coordinate System

All graphical computing systems use some sort of coordinate system to specify the nature of points in the system. Coordinate systems typically spell out the

wide range of computer systems. The GIf89a format supports up to 256 colors with a transparency color, data compression, and interlacing.

Transparency colors are colors in an image that are unused, meaning that they aren't drawn when the rest of the colors in the image are drawn. The significance of transparency colors is that they allow the background behind an image to show through.

Data compression is a technique involving the conversion of data into a smaller, more efficient format so that it takes up less space in memory or on a hard drive or takes less time to transfer over a modem connection.

Interlaced images are images that can be displayed in incremental stages as they are being loaded. You've no doubt witnessed this effect in Web pages when images go from blurry to clear as they are being loaded.

JPEG stands for **J**oint **P**hotographic **E**xpert **G**roup. This group created a format to store full-color-spectrum, continuous-tone images. These images, when properly created, can be of much higher fidelity as well as more compressed than a GIF encoding of the source image.

Colors & Palettes:

Bitmapped computer images are composed of pixels that describe the colors at each location of an image. Each pixel in an image has a unique color that is usually described using the RGB color system. Java provides support for working with 32-bit images, which means that each pixel in an image is described as using 32 bits. The red, green, and blue components of a pixel's color are stored in these 32 bits, along with an alpha component. The alpha component of a pixel refers to the transparency or opaqueness of the pixel. A 32-bit Java image pixel is therefore composed of red,

components like the pixels used in a direct color model. An index color model pixel contains an index into an array of fixed colors.

3.2.3 Browser

A web browser is a application which is used to view and navigate the pages on World Wide Web. Any web browser job is two fold: Given a pointer to a piece of information on the Net (URL), the browser has to able to access that information or operate in the same way based on the contents of the pointer.

Browser is the application used to access any **H**yper **T**ext **M**arkup **L**anguage (HTML) based information. That information usually called pages can be on the Internet or on Intranet. In addition to HTML, browsers can interpret and display information stored in other formats, including sounds, graphics etc.

Hot Java, Internet Explorer and Netscape Navigator are Java enabled browsers. A Java enabled web browser contains its virtual machine. Web documents containing name embedded Java applets must specify the location of the main applet class file. The web browser then starts up the virtual machine and passes the location of the applet class file to a class loader. Each class file knows the names of any files coming from the client machine. This may require the class loader to make a number of additional classes loading operations before the applet starts. After loading the class file, execution begins and the applet is asked to draw itself in the browser window.

Java is strongly associated with the Internet because of the fact that the first application program written in Java was Hot Java. Internet users can use Java to create applet programs and run them locally using a "Java-enabled browser" such as Hot Java. We can also use a Java-enabled browser to download an applet located on a

computer color system in use, there are others. Another popular color system is HSB, which stands for Hue Saturation Brightness. In this system, colors are defined by varying degrees of hue, saturation, and brightness.

Chapter 4

Artificial Intelligence

4.1 About Artificial Intelligence

This is one of the challenging areas in gaming. Creating truly engaging games is often a matter of effectively mimicking human thought within the confines of software constructs. Because you no doubt want your games to be engaging, you need at least a basic understanding of how to give your games some degree of brainpower.AI is often useful to control many of the background aspects of the game using simple AI.

Artificial intelligence (AI) is defined simply as techniques used on a computer to emulate the human thought process. This is a pretty general definition for AI, as it should be; AI is a very broad research area, with game-related AI being a relatively small subset of the whole of AI knowledge. Human thought is no simple process to emulate, which explains why AI is such a diverse area of research. Even though there are many different approaches to AI, all of them basically boil down to attempting to make human decisions within the limitations of a computer.

Most traditional AI systems use a variety of information-based algorithms to make decisions, just as people use a variety of previous experiences and mental rules to make a decision. In the past, the information-based AI algorithms were completely

4.2 Types of Game AI

There are many different types of AI systems and even more specific algorithms implementing those systems. Even when you limit AI to the world of games, there is still a wide range of information and options from which to choose when it comes to adding AI to a game of your own. Many different AI solutions are geared toward particular types of games, with a plethora of different possibilities that can be applied in different situations.

It makes more sense to give the theoretical background on a few of the most important types of AI, and by which we can figure out how they might apply to particular gaming needs. Game-related AI can be classified into three fundamental types:

- Roaming
- Behavioral, and
- Strategic.

The three types of AI discussed here are simply the most common types.

Roaming AI

Roaming AI refers to AI that models the movement of game objects-that is, the decisions game objects make that determine how they roam about the game world. A good example of roaming AI is in shoot-em up space games, where aliens often tend to track and go after the player. Basically, roaming AI is used whenever a computer-controlled object must make a decision to alter its current path, either to achieve a desired result in the game or simply to conform to a particular movement pattern. In

Behavioral AI

Although the types of roaming AI strategies are pretty neat in their own right, a practical gaming scenario often requires a mixture of all three. *Behavioral AI* is another fundamental type of gaming AI that often uses a mixture of roaming AI algorithms to give game objects specific behaviors. To implement behavioral AI, you need to establish a set of behaviors for the game objects. Giving game objects behaviors is pretty simple, and usually just involves establishing a ranking system for each type of behavior present in the system, and then applying it to each object. A typical implementation simply involves a `switch` statement or nested `if-else` statements to select a particular behavior.

Strategic AI

The final fundamental type of game AI you're going to learn about is strategic AI. *Strategic AI* is basically any AI that is designed to play a game with a fixed set of well-defined rules. For example, a computer-controlled chess player would use strategic AI to determine each move based on trying to improve the chances of winning the game. Strategic AI tends to vary more based on the nature of the game, because it is so tightly linked to the rules of the game.

Strategic AI, especially for board games, typically involves some form of weighted look-ahead approach to determining the best move to make. For a look-ahead to make sense, however, there must be a method of looking at the board at any state and calculating a score. This is known as *weighting* and is often the most difficult part of implementing strategic AI in a board game. *Weighting* is a method of looking at a game at any state and calculating a score for each player. This approach

typically involves a lot of tinkering with the AI code, but it can result in very good computer players.

AI in Commercial Games

So far, adventure and strategy games are the only commercial games to have a great deal of success in implementing complex AI systems. Some examples for games involving AI are:

- Battlecruiser: 3000AD first game to feature neural networks
- Cloak, Dagger, and DNA, first game to use Genetic algorithms

4.3 Java's support for AI

Java provides a wide support for artificial intelligence. In Java it is easier to implement any kind of AI. As already said Roaming AI models movement of gaming objects, which can be implemented by changing the x, y location of the objects. Graphics has a wide support in Java because there are separate classes for 2-Dimensional objects and 3-Dimensional objects, so, rendering objects is easy in Java.

Behavioral AI can also be implemented in java because it deals with specific set of behaviors. These behaviors can be established by using either nested if-else conditions or switch-case statements.

Strategic AI can be implemented by establishing a fixed set of rules, which should be followed in the game. These rules will be the moves to be made in the game.

- If there are no marbles displace the marbles below the dissolved ones to the center row.
- Otherwise create new marbles in the center row.
- If there are no adjacent marbles of same color, check whether there are two adjacent marbles of same color. If so, then search for the same marble in the third column.
- If there is a marble of that property, bring that marble to center row and destroy them.
- Else if the adjacent marbles are of different colors, look for the color of the marble in the first column in the second and third columns. If so, destroy otherwise repeat the same step for second, third and fourth positions.
- Continue this game loop until the opponent's grid is fully filled with marbles.

In the game apart from the Strategic AI we have to use some Behavioral AI also in the game. In order to continue the game we have to look for the for the following behaviors:

- Whether there are more than three marbles of the same color adjacent to each other.
- Else look for two adjacent marbles of same color in the center row.
- Look for the above behavior in all positions.

- Else look for marbles of same color in alternative positions.
- If any of these behaviors is satisfied look for the third marble of same color in the adjacent columns and dissolve them.

Chapter 5

System Design

Software design is an iterative process through which requirements are translated in to a "blue print" for developing the software. Initially, the blue print depicts a holistic view of software i.e., the design is represented at a high level of abstraction. It is the level that can be directly traced to specific data, functional and behavioral requirements. As design iterations occur, subsequent refinement leads to design representation for lower level of abstractions. The evaluation of good design is made as follows:

- The design implements all of the explicit requirements contained in the analysis model and it accommodates all the implicit requirements desired by the customer.
- The design is a readable, understandable guide for user and for those who test and subsequently maintain the software.
- Design provides a complete picture of the software, functional and behavioral domains form an implementation perspective.

The quality representation of design is achieved by the following criteria:

- The design exhibits a hierarchical structure that makes intelligent use of control among various elements of software.

- The design is modular, that is, the software could be logically partitioned into elements that perform specific functions and sub-functions.
- The design contains both data and procedural abstraction.
- The design leads to modules that exhibit independent functional characteristics.
- The design leads to interfaces that reduce the complexity of connections between modules and external environment.
- The design is derived using a repeatable method that is driven by information obtained during software requirement analysis.

The software design is both a process and a model. The design process is a set of iterative steps that enables the designer to describe all aspects of the software to be developed. This system design is traceable to analysis model, because a single element of the design model often traces to multiple requirements.

This design will minimize the intellectual distance between the software and the problem as it exists in the real world. That is, the structure of the software design should minimize the structure of the problem domain. The design should also exhibit uniformity and integration. The structure of the design should be:

- Adaptive
- Degrade gently
- Assessed for quality, as it is being created and not after that.
- Reviewed to minimize conceptual errors

The proposed system will be developed using JDK 1.2. The system will involve some basic concepts of Artificial Intelligence. Java is used efficiently in this

system. An applet is designed to be transmitted over the net and executed by a Java compatible web browser. Future enhancements can also be made to the existing system design.

Since the project deals with computer-simulated player, it does not require any input to the game and so there would be no output from the system.

5.1 Detailed Design

The design part of the system include two modules:

- Graphics support for the whole game.
- Computer-simulated player

For a game engine to be efficient the graphics support for the engine should be faster and effective since graphics is the heart of the game. This is achieved by storing all images required for the game in a single file. At the time of play the required images are cropped from the base image as rectangular portions. The boundaries of the marbles are then made transparent so that the exact marble is obtained. Since only one big file will be used, the space required for storing the images will be less and so accessing images at runtime will be easier. Due to this reasons the time taken to download the game will be less.

Graphics support for the Game

For a game engine to work efficiently the graphics support provided to the engine must be good and faster. In this game the images required are stored in a single file from which images are cropped and used at runtime.

The game uses a base image called CommonImgballs.jpg to store all the marbles required for the game. This file consists of five different balls along with the GIF images required when the marbles are dissolved.

The base image consists of five different marbles each one having twelve different transforms of the marble. The x, y co-ordinates, width and height of each marble are stored in four different are arrays along with the color of the image. Each color is assigned a different number as given below:

- Green: 1
- Yellow: 2
- Red: 3
- Blue: 4

These values are stored in a separate array and they, can be accessed by specifying that particular index. All these arrays are serialized and stored in a separate data file. Serialized data can be accessed easily by de-serializing them. Data can be stored in series by implementing the interface Serializable and this is a way to store data and restore them easily. To access data in a file which, is serialized, the file has to de-serialized first.

In the game coordinates of the marbles are serialized by implementing Serializable interface to a class and using it by creating subclass for that class.The classes and methods used are:

1. ***SerializeObject:***
   - This class implements Serializable interface with out this separate class a NotSerializableException is thrown.

- Its constructor assigns the required data. This class will provide a way to store data and restore it at runtime.
- Subclasses of this class are also serializable. In this class the current objects x and y coordinates, width, height and flag objects are assigned.

2. ***ObjWrite:***

This class imports the SerializeObject to serialize the data. This class is used to write data into a data file which will be serialized.

- A fileoutputstream is created to create a data file and this file is given as input to the ObjectOutputstream.
- Then data is written in to the data file using writeObject method in the ObjectOutputstream. The data written in to file is stored as objects.

3. ***ObjRead:***

This class also imports SerializeObject class to restore stored data.

- Here the same data file is opened as FileInputStream inorder to read the data form the file.
- This stream is converted into a ObjectInputStream from which objects can be read using readObject method. We can de-serialize the data in the file and extract the information required from the file.

Computer-simulated Player

This part of the system is used to make the computer play by itself. The computer is made to play as the opponent to human player. The computer is made to play by itself using artificial intelligence. This system uses Strategic AI and Behavioral AI. Strategic AI is used as the game involves some fixed set of rules to be

followed. The game also involves Behavioral AI as the game proceeds based on some behaviors. The data structures used in the game are:

1. Vector:

This class implements a growable array of objects. It also contains components that can be accessed using an integer index. The size of a Vector can

grow or shrink as needed to accommodate adding and removing items after the Vector has been created. Due to this reason a vector is used to store all marbles on the grid. In this game an empty vector is constructed with the initial capacity as 25 and capacity increment as one is created.

2. Thread:

Each marble is stored in the vector as a thread, threads are used because it would be easier to load all the 25 marbles initially and at the time of dissolving marbles of same property it would be easier to delete them.

The flow of game is as below:

1. An applet is created which implements Runnable interface.
2. Images required for the game are initially loaded.
3. The thread to begin the game is started.
4. Initially 25 different marbles are placed on to the grid.
5. Then the center row is checked for three, four or five adjacent marbles of same color from the initial selector position.
6. If the above rule is satisfied, those marbles are dissolved and the marbles in the above and below rows are repositioned.

3. ***run()***

   - This method starts a thread and executes the commands within it.
   - In this method a method createInitMarbles() is called to create 25 marbles initially.

4. ***createInitMarbles()***

   - This method creates the initial 25 marbles to start the game.
   - This method is used to create new marbles and place them on to the grid initially.
   - This method calls createMarble() 25 times to place them on to the screen.

5. ***createMarble(int x, int y)***

   - This method creates a new marble and paints it on to the screen and it places the new marble on to the specified x, y location.
   - Here a random number is generated and the image at that index in the object file is created.
   - To create a new marble placeImage1 class constructor is called
   - Each new marble on creation is added on to the vector as its element.

6. ***Autorun()***

   - This method is called by the run method in this class
   - The method initially looks for the selector position and calls the Destroy() method to dissolve adjacent marbles of same property.
   - If the marbles are dissolved, the screen is repainted and the marbles in the above and below rows are repositioned.
   - If there are no marbles to dissolve movespace_check() method is called.

- If the number of marbles on the grid reduces to 18 the method Roll_Down is called.
- The above steps are repeated for all the selector positions continuously until the specified duration.

7. ***moveSpace_check()***

- This method is called by AutoRun() method.
- The elements in the center are extracted and they checked for adjacent marbles of same color.
- Based on the position of the selector, the following conditions are checked:
  - ✓ If there are three, four or five adjacent marbles of same color, they are dissolved and marbles are repositioned.
  - ✓ Else if two adjacent marbles are of same color, the same marble is searched in the next adjacent positions. If the same marble exists that particular column is moved vertically either up or down by calling ud_check().
  - ✓ Else if there are same color marbles in the alternative positions, again the same marble is searched in the middle column. If exists it is moved to center row using the method ud_check() and they are dissolved and repositioned.
  - ✓ Else the marble in the selector position is searched in the adjacent columns and if exists they are moved in to the center row and destroyed.
  - ✓ The above behaviors are checked for all selector positions and if the rule is satisfied they are dissolved and repositioned.
- The above behaviors are checked for each call to this method.

- Based on the number of elements in that column the functions are performed.
- If the number of elements is less than 7 then this column can be moved until there is a element in the center row.
- Else, the column can be slided until the last element touches the bottom boundary of the grid.
- After each movement the screen is updated with the new screen.

11. Space_check()

- This method is called movespace_check() method.
- In this method the elements in the center row are extracted and made to shift to left by one position.
- If the marble in the last column position, on next shift it will the first element in the center row.
- Here also on each updation to the grid the screen is repainted.

12. update_Images(Graphics g)

- This method is called either form the paint() method of the applet whenever updations are made or called directly.
- This method first paints the container with the background image and then places the selector on to the screen.
- Then the elements in the vector i.e., the threads which are alive are painted individually on to the screen by calling placeImage's paint method.

placeImage1

This class is called whenever a marble is created. Each element in the vector is an object of this class and object is a thread.

Constuctor:

This constructor assigns the screen position of the new marble, its color and its index to the current objects x, y, flag and index along with the parent of the marble. It then starts that particular thread.

This class contains the following methods:

1. ***toLoadImg()***
   - This method is called from the run method of this class
   - In this method the main image which consists of all the marbles and GIF images is loaded.
   - This image is loaded only once even though there are a number of calls to this method.
2. ***waitToDie()***
   - This method is used to make all the threads to wait until all the images are loaded.
   - This is a synchronized method.
3. ***toCropImageFilter()***
   - In this method the appropriate rectangle is cropped form the main image. The coordinates of the rectangle are obtained from the index value assigned and the values are obtained from the object file.
   - Then the cropped rectangle is sent as input to the constructor of DissolveEdgeFilter, where the background of the marble is made transparent and the new image is returned.
   - A new image is created for that filtered image and that is drawn on to the screen each time the paint method is called.

- If the number of marbles created initially reaches 25, the repaint() method of the parent is called.

4. ***dissapear()***
   - This method is used by Destroy() method of the parent class to make the marbles satisfying the rule die.
   - In this method that specific thread is stopped from further execution.
5. ***timeToDie()***
   - This method notifies all the threads in the game that a particular thread is going to stop its execution.
6. ***paint(Graphics g)***
   - It is called by the parents update_Images(g) method to draw that particular image on to the screen in the specified x, y location.

DissolveEdgeFilter

This is a class used to dissolve the edges of the images. In this class the background of the image sent is made transparent. This method extends RGBFilter.

Constructor

The constructor of this class assigns the color model used as canIndexColorModel. It takes as input the cropped image and then calls the built-in method for RGBFilter class.

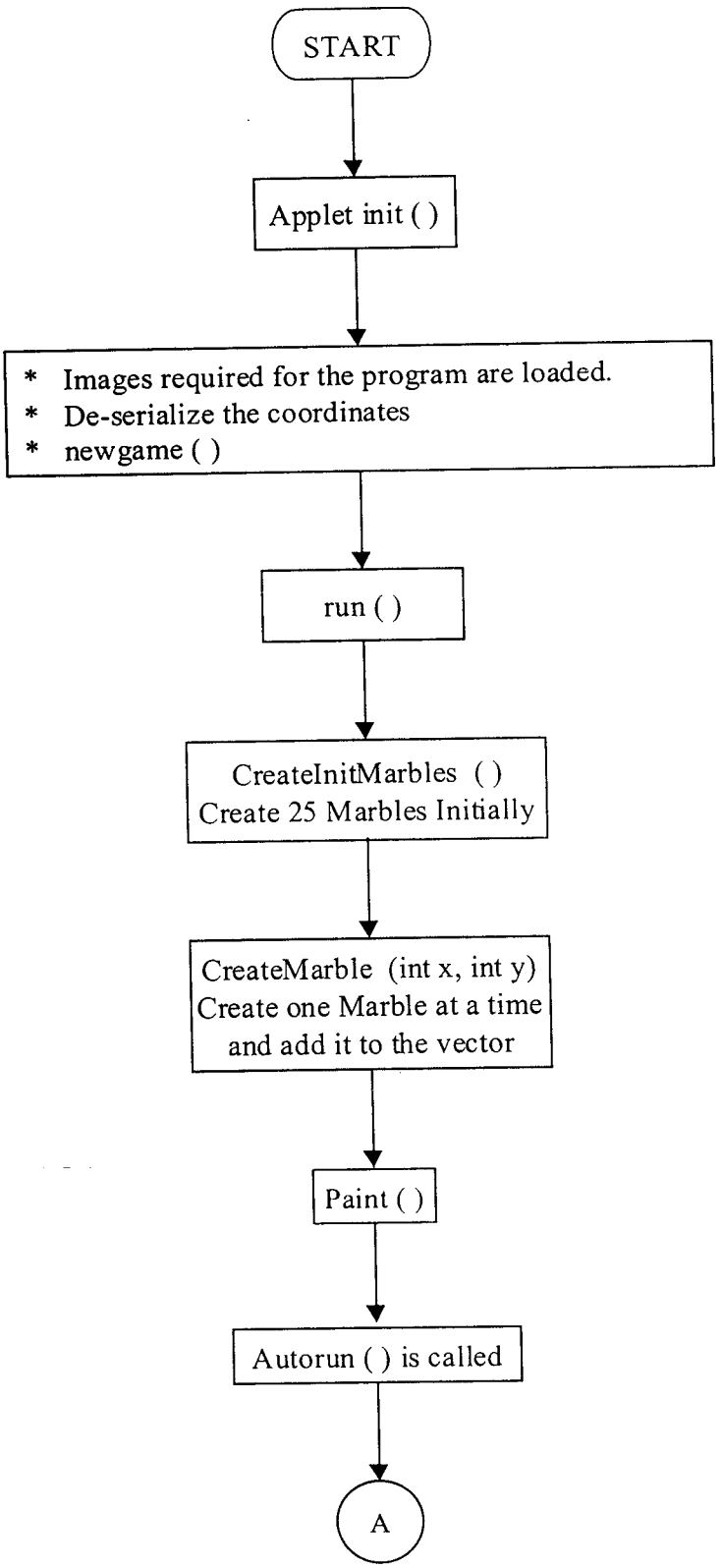
Method

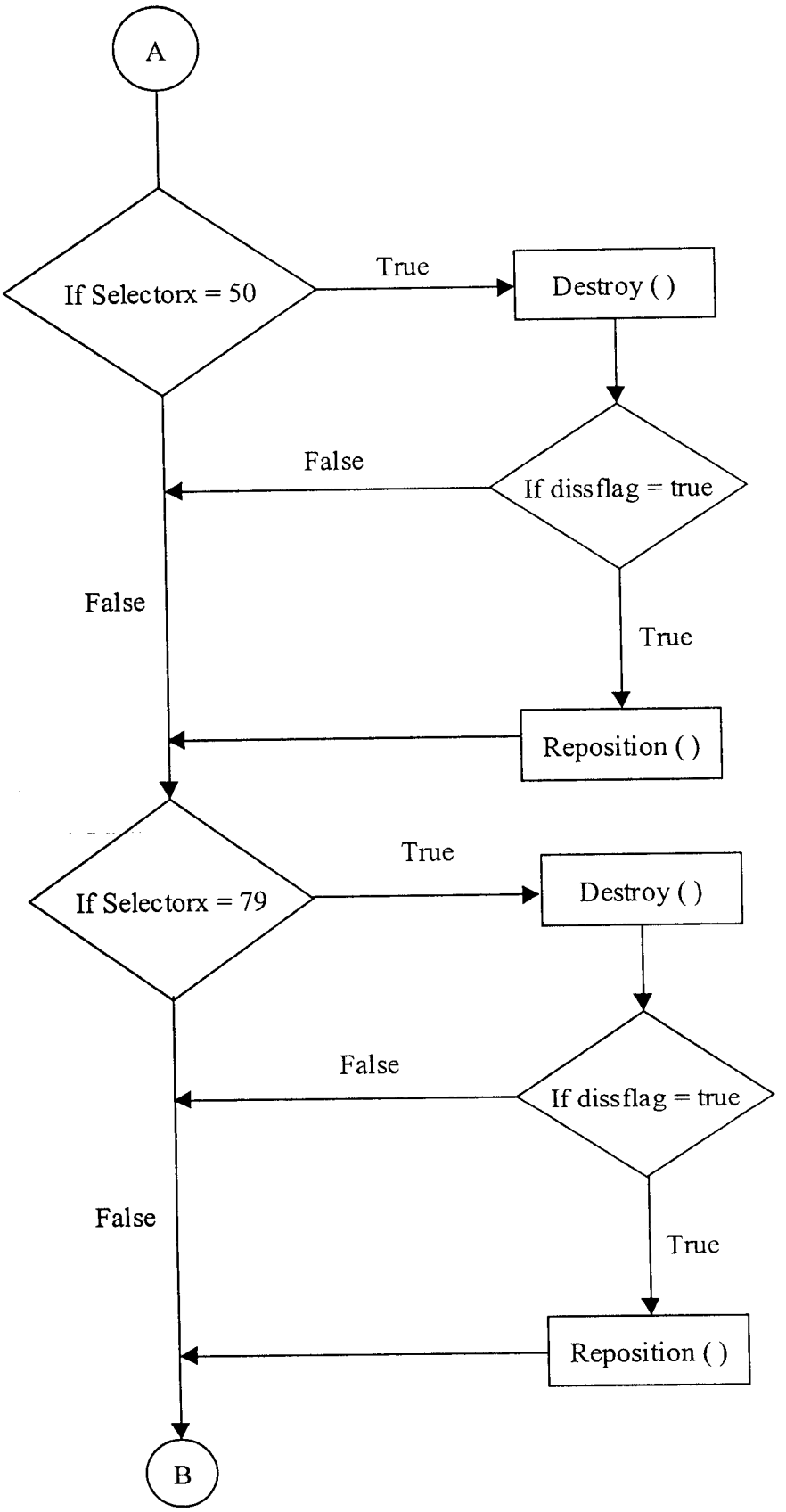
filterRGB(red,green,blue)

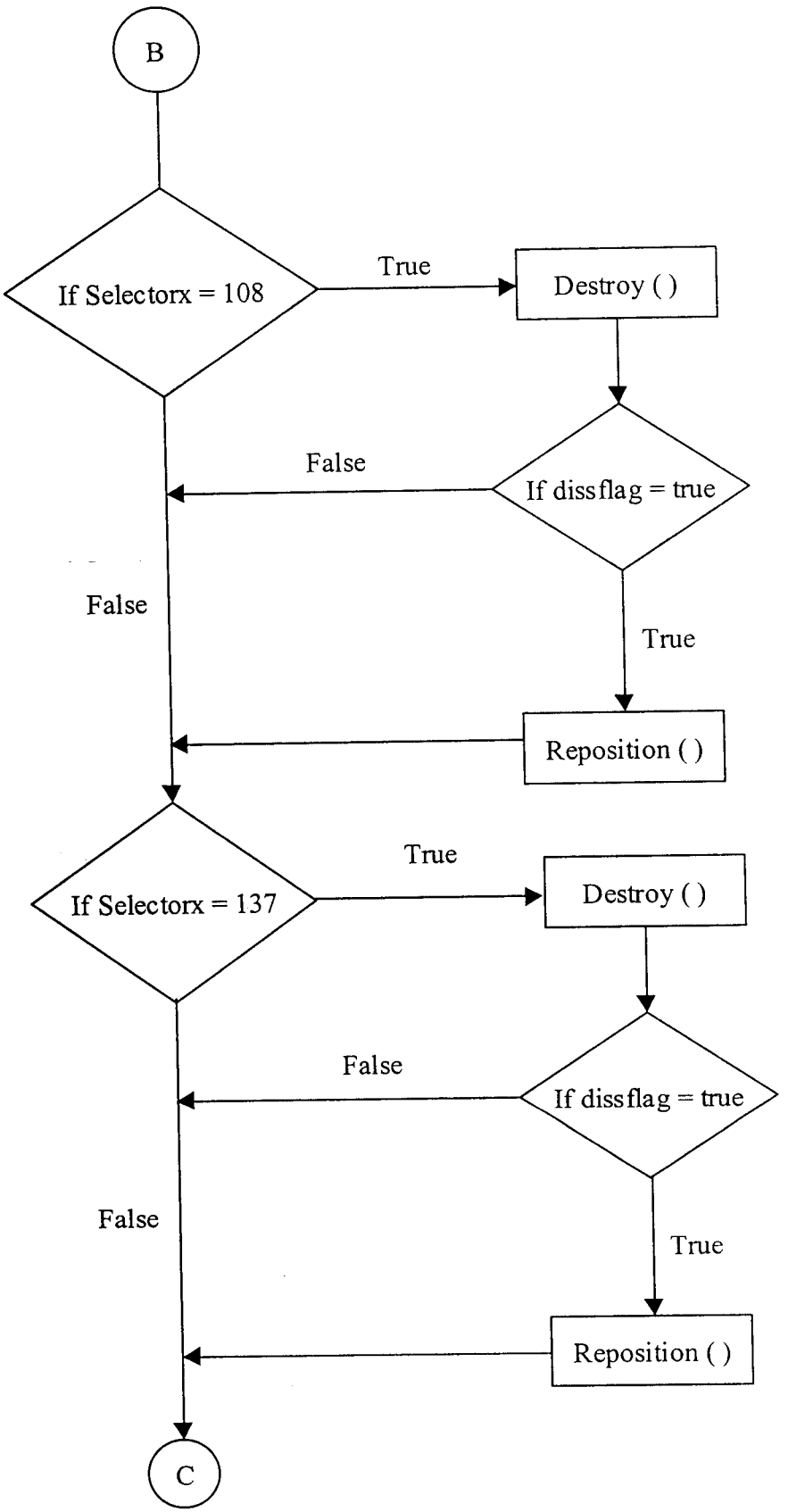
- This method takes as input the filtered image and extracts RGB value for each pixel in the image.

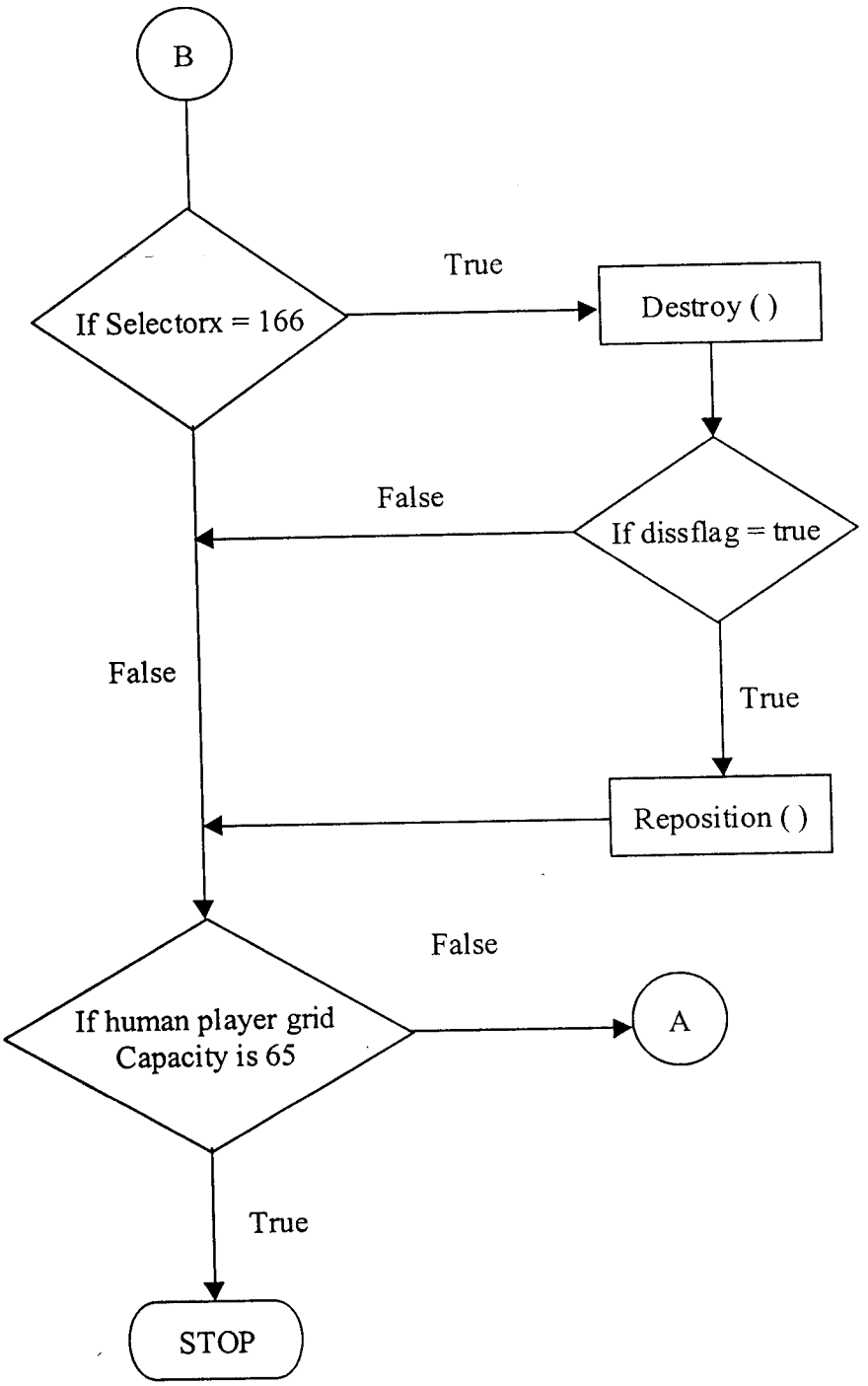
marble is created. As the computer plays by itself it does not require any input hence there will be no output from the system.

5.4. *Process Design*




A
If Selectorx = 50
True
Destroy ( )
If dissflag = true
False
False
True
Reposition ( )
If Selectorx = 79
True
Destroy ( )
If dissflag = true
False
False
True
Reposition ( )
B


B
If Selectorx = 108
True
Destroy ( )
If dissflag = true
False
False
True
Reposition ( )
If Selectorx = 137
True
Destroy ( )
If dissflag = true
False
False
True
Reposition ( )
C


B
If Selectorx = 166
True
Destroy ( )
If dissflag = true
False
True
False
Reposition ( )
If human player grid
Capacity is 65
False
A
True
STOP

Chapter 6

System Implementation

6.1 System Implementation

The completion of design process initiates the next step of development, implementation and testing. System implementation is the process of making the newly designed system fully operational. The system is implemented after careful testing. The following steps have to be followed in the implementation of the system.

- Implementation planning
- Equipment acquisition and installation
- System conversion
- User training
- System Audit

6.1.2 Implementation Planning

The first task in system implementation is implementation planning. Planning means deciding on the methods and time scale to be adopted. A logical starting point for this type of planning involves knowledge of the following areas.

- Personal needs
- Programming equipment's selected
- Physical requirements
- Conversion activities

6.1.3 User Training

The training should include every one associated with the implementation, use, operation or maintenance of the new system. Since the system is a game and the module is computer-simulated player, it is doesn't require user's to be trained to use the system as the computer will play by itself.

6.1.4 System Audit

The system audit is an investigation to review the performance of an operational system. The performance of the system is compared with the planned performance, to verify that the stated objectives of the system are valid in the present environment and to evaluate the achievement of these objectives.

Chapter 7

System Testing

7.1 Testing Objective

Once the system implementation plan ahs been decided, the application has to be tested with sample data before giving it to the user. The objective of testing is as follows:

- Testing is the process of executing a program with the indent of finding an error.
- A good test case is one that has a high probability of finding an as-yet undiscovered error.
- A successful test is one that uncovers an as-yet-undiscovered error.

Since this is a computer-simulated player it doesn't require separate test cases to test the system, as there are no human players. In the testing process the main issues taken into consideration are:

- Process validation
- Data validation and security
- Preparing the feedback information on the testing process

After completing the testing process, the application can be downloaded to the user's site and the computer will start playing with opponent: human. Since integration of modules is yet to be done, there will be small changes that need to be incorporated into the application before implementation.

7.2 Black Box Testing

In Black box testing, the following are tested:

- Incorrect or missing functions
- Interface errors
- Errors in data structures
- Performance errors
- Initialization and termination errors

7.3 White Box Testing

In White Box testing, the following are tested successfully for this application.

- Check whether all independent paths within a module have been exercised at least once.
- Exercise all logical decision on their true and false sides.
- Execute all loops at their boundaries and within their bounds.
- Exercise internal data structures to ensure their validity.

7.4 Testing GUIs

Graphical User Interfaces (GUIs) present interesting challenges for software engineers. Because of reusable components provided by Java development environment, the user interface become less time-consuming and more precise. The GUIs in this project are tested as follows:

- The window opens properly based on related type commands.
- The window can be resized, moved and scrolled.
- The window is properly regenerated when it is overwritten and then recalled.

- Multiple windows are displayed; the name of the window is properly represented.
- The window is properly closed.
- The images in the window are correctly placed.

Chapter 8

Conclusion

The system has been developed according to the specifications and requirements given by the company. The project entitled Lose Your Marbles is a game. The modules specified include computer-simulated player and graphics support for the game. Here the computer is made to play the game by itself. It acts as an enthusiastic opponent to the human player, by which the player can play the game with interest and strive towards winning the game.

Apart from the above module the project also includes provision of graphics support to the game. This support is obtained by storing all the images required for the game in a single file and cropping the required images at runtime. Images, which are required, are cropped as rectangular portions and the edges of images are made transparent to obtain the exact image of the marble.

This system is part of a big game and the separate modules in the game are completed fully. The game is still under implementation, as it requires integration of all the modules in the game. Since it is a computer-simulated player the system was tested on various machines each one with different processor speed, so that the computer plays at the same pace on all the machines. As the system is still under implementation, the code will be optimized further during the integration of the

modules. Further the system can be executed on any platform since it is developed in Java using applets.

- Robert F. Sproull & William M. Newman **Principles of Interactive Computer Graphics**
  - Basic Concepts of Computer Graphics.

Appendix A

Graphics in General

Graphics is the heart of the game. If the graphics of the game is good and efficient then only the game engine will be faster. A *graphics engine* is essential to a well-designed game in Java. A graphics engine is an object that is given the duty of painting the screen. The graphics engine keeps track of all objects on the screen at one time, the order in which to draw the objects and the background to be drawn. By far, the most important function of the graphics engine is the maintenance of movable object blocks. Any graphics system should allow the programmer to define pictures that include a variety of transformations. It must be possible to transform one image into another image.

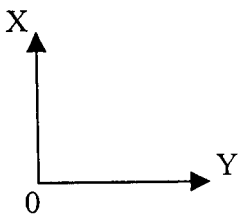
Types Of Graphics:

Generally speaking graphics composes of mesh and textures to be mapped on to the image. Graphics can be broadly classified as:

- 2D Graphics
- 3D Graphics

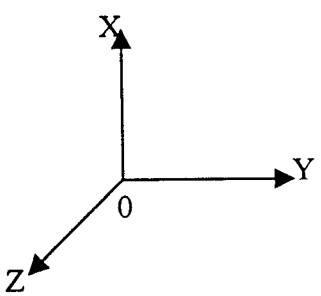
2D graphics deals with the x, y coordinates system only i.e., the height and width of the image. The images will not have any depth and they will be plane. These sorts of graphics have less reality effect but they are faster and easier to render. The dimensions include only the x & y coordinates. This type of graphics look as though there is no life to them.

Figure 5.1 shows the coordinate system for 2D graphics:



3D Graphies on the other hand deals with three coordinates. 3D graphics coordinate system involves x, y & z-axes. As described in 2D graphics the first coordinates denote the height and width of the graphics object and the third one denotes the depth of the object. If depth is more it gives an illusion that the object exists in reality more depth makes the object more realistic.

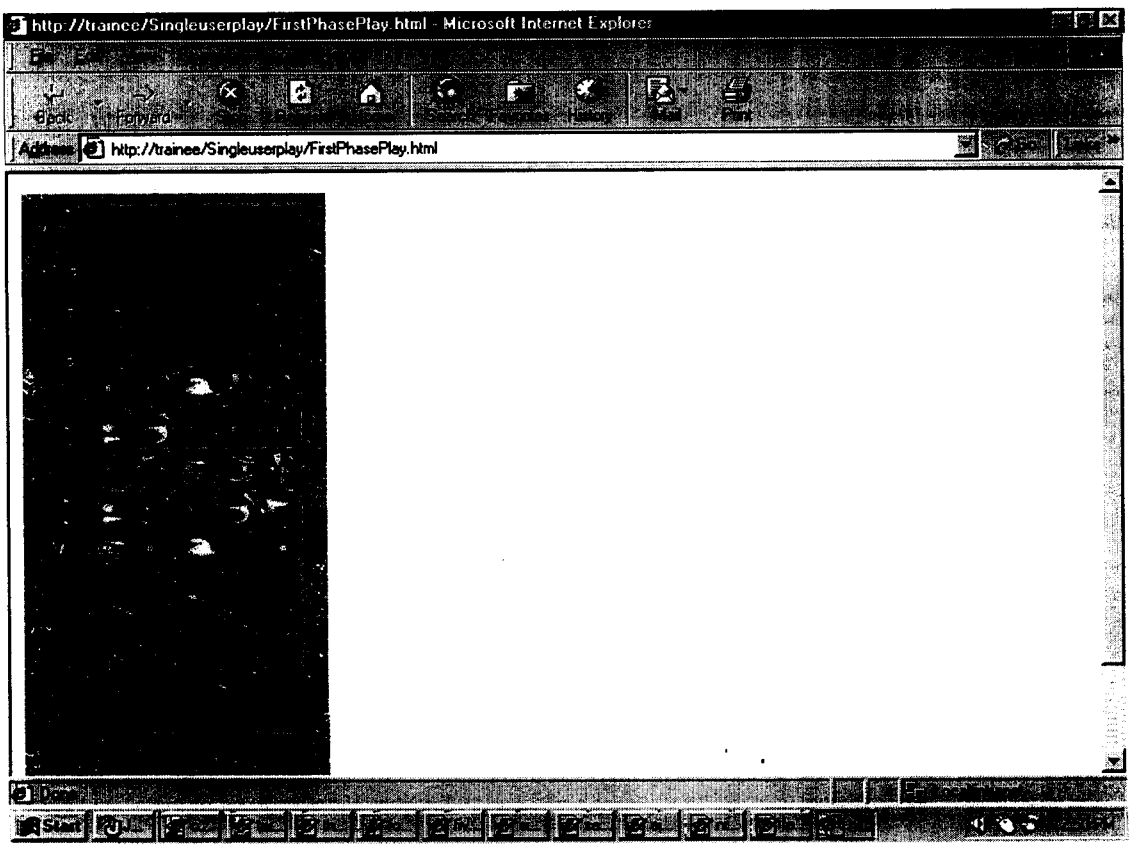
Figure 5.2 shows the coordinate system in 3D graphics:

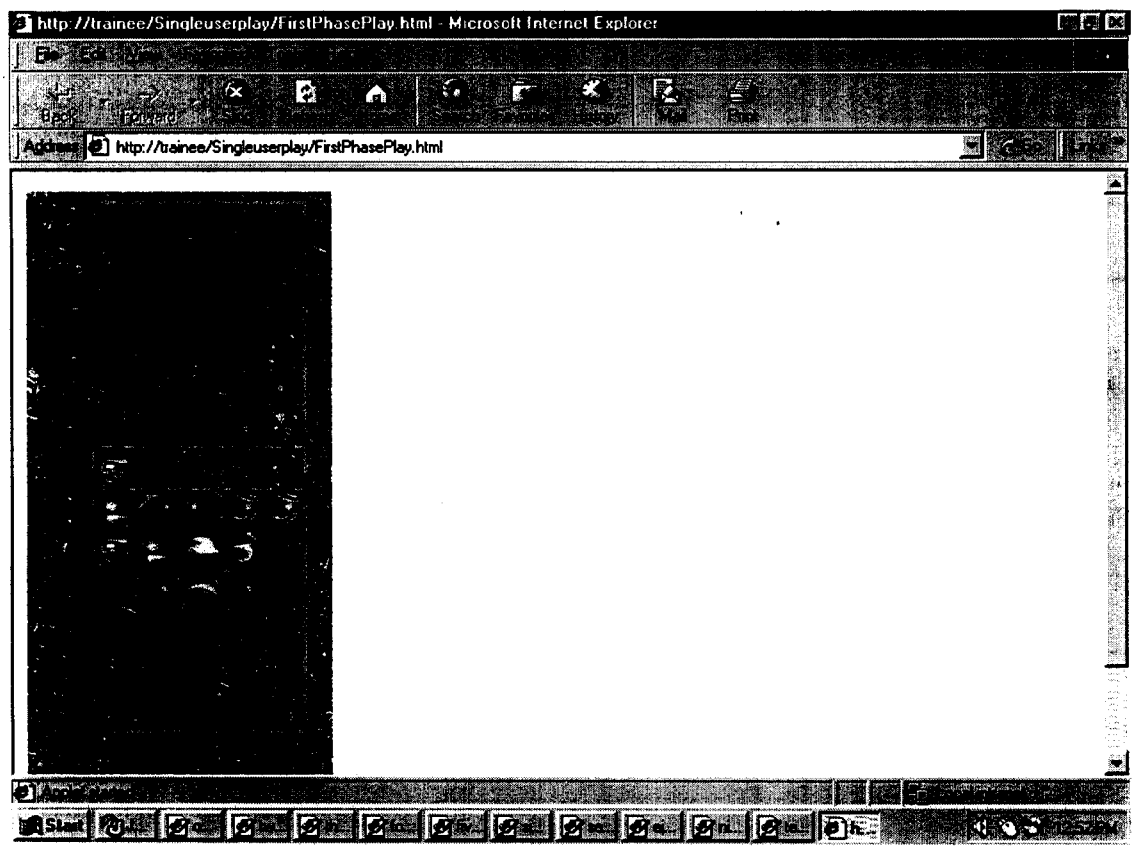


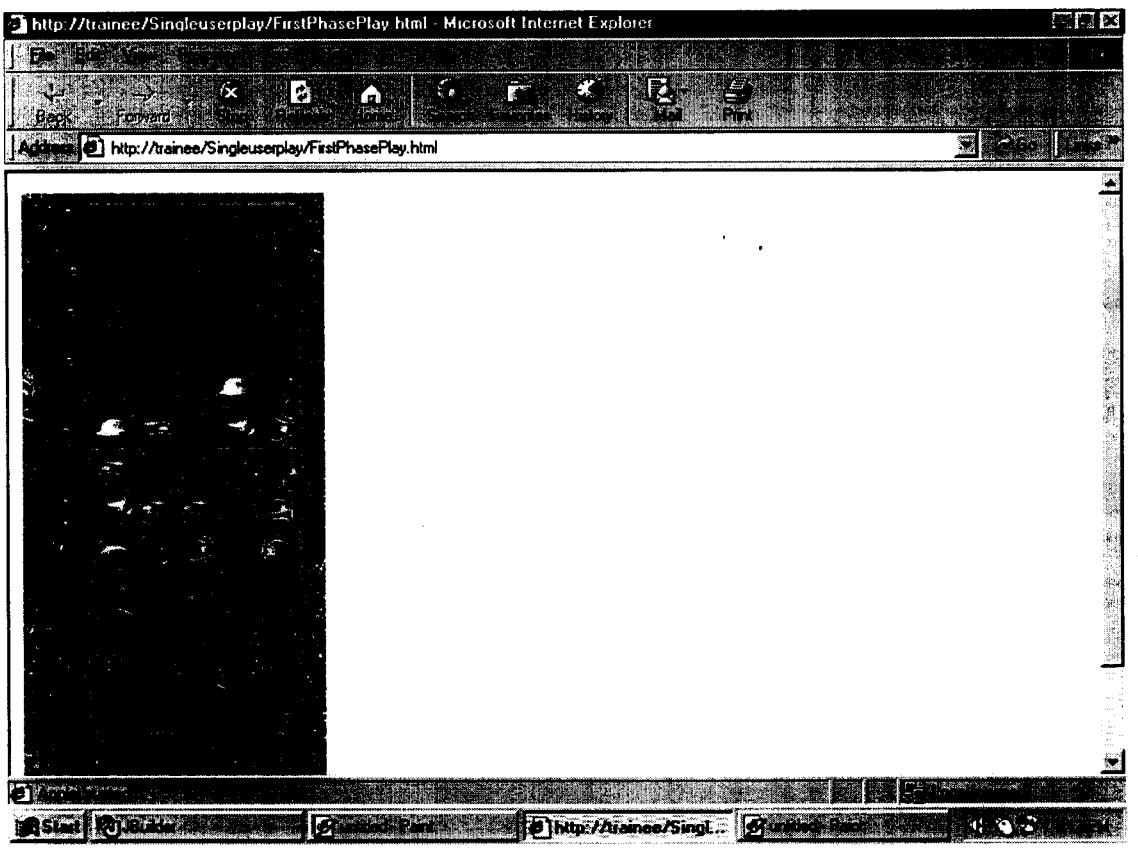
Apart from these two basic types, from the aspect of games graphics can be classified into:

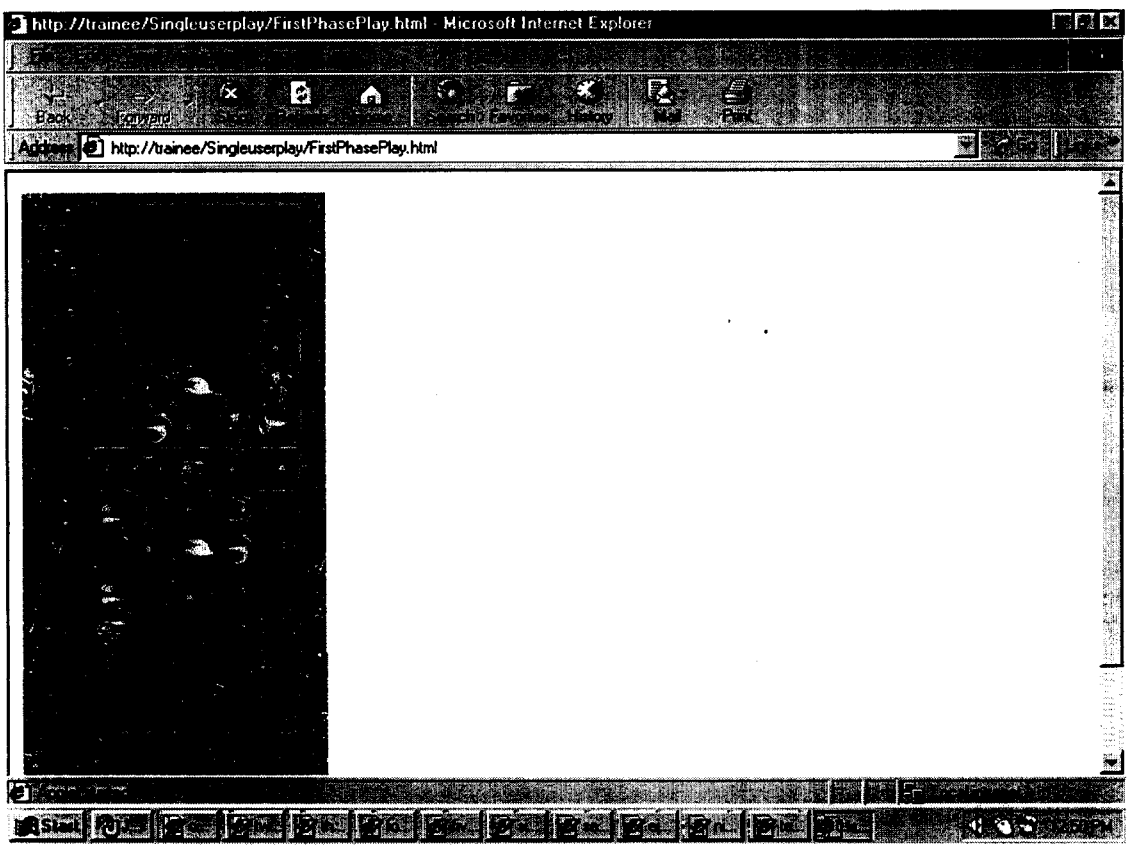
- **Line-art Graphics:** The first type graphics is called *line-art* graphics, it is called so because it encompasses practically all hand-drawn graphics, whether drawn and scanned from paper or drawn in a software paint program.
- **3D-Rendered Graphics:** 3-D rendered graphics have become popular in the game world. There is a reason for this; rendering provides the capability to create incredibly complex and realistic 3-D graphics that

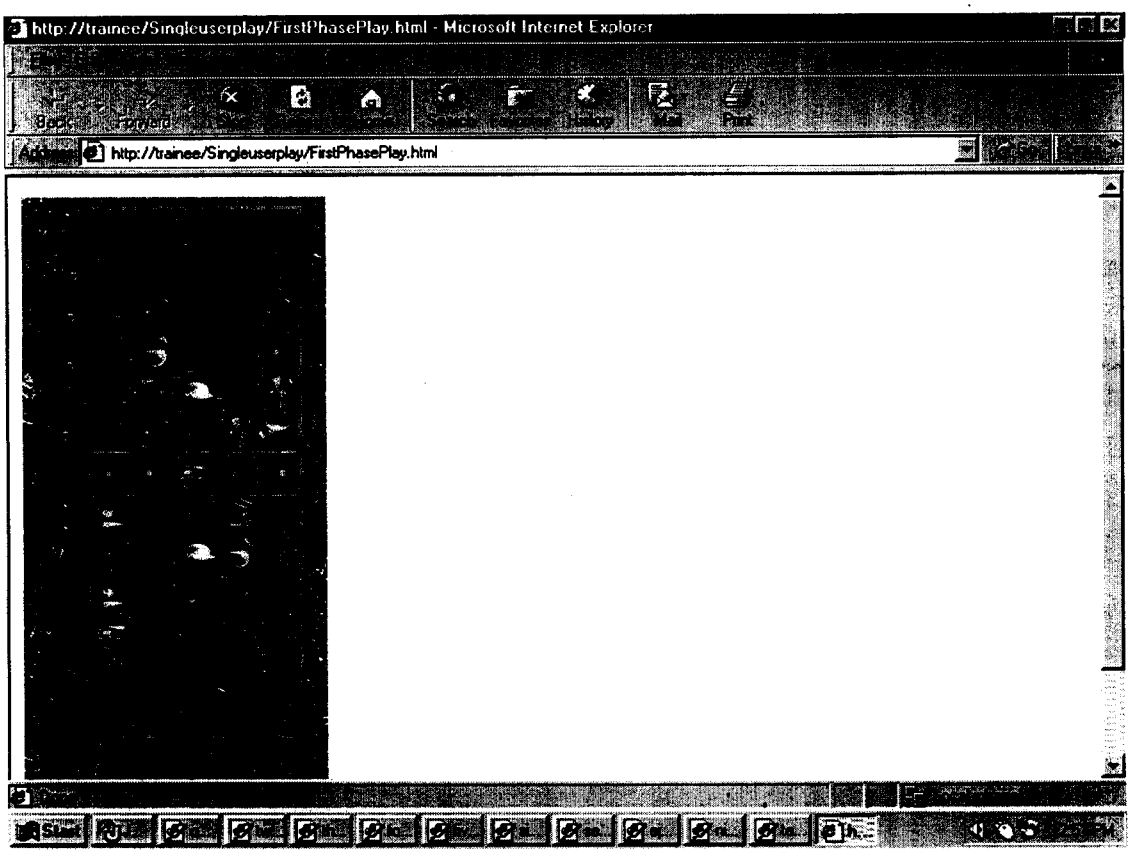

Shareware Version

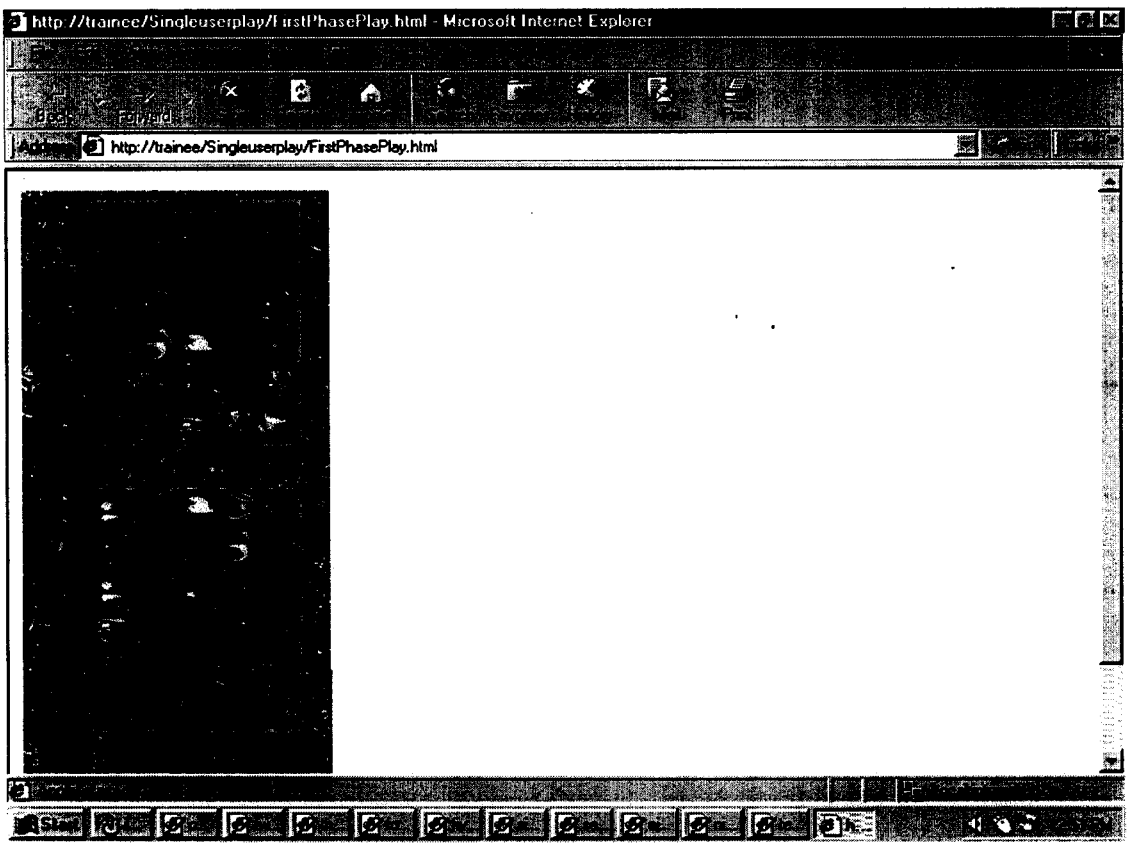

http://trainee/Singleuserplay/FirstPhasePlay.html - Microsoft Internet Explorer
Address
http://trainee/Singleuserplay/FirstPhasePlay.html


http://trainee/Singleuserplay/FirstPhasePlay.html - Microsoft Internet Explorer
Back
Forward
Address
http://trainee/Singleuserplay/FirstPhasePlay.html


http://trainee/Singleuserplay/FirstPhasePlay.html - Microsoft Internet Explorer
Back
Forward
Print
Address
http://trainee/Singleuserplay/FirstPhasePlay.html
Start


http://trainee/Singleuserplay/FirstPhasePlay.html - Microsoft Internet Explorer
Back
Forward
Address
http://trainee/Singleuserplay/FirstPhasePlay.html


http://trainee/Singleuserplay/FirstPhasePlay.html - Microsoft Internet Explorer
http://trainee/Singleuserplay/FirstPhasePlay.html



P-433