**Chapter 1**

**INTRODUCTION**

* 1. **OVERVIEW:**

The Project consisting of **“Ping Pong Game”** which contains of two paddles on either sides of the screen and a ball between them. Players have to save the ball from going off the screen. The game starts with a menu on the screen giving you three options

1.Single Player easy.

2.Single Player hard.

3.Two player.

4.Exit.

In a Single player game one person controls one of the paddle using UP and DOWN arrow keys and the other paddle is controlled automatically. In a two player game one paddle is controlled using UP and DOWN keys and the other paddle using ‘a’ and ‘z’ keys.

The project is based on Simple window coordinates and using recursive techniques in OpenGL

* 1. **Computer Graphics:**

**Computer graphics** deals with creating pictures or images using computer and more generally the representation and manipulation of images data by computer with the help from specialization software and hardware. The field began humble almost 50 years ago, with the display of few lines on the cathode-ray tube(CRT); now, we can create image using computer that are indistinguishable from photographs from the real objects. We routinely train pilots with simulated airplane, generating graphical display of the virtual environment in the real time. Feature length movies made entirely by computer have been successful, both critically and financially, massive multiplayer game can involve tens of thousands of concurrent participants.

**Visualization** is any technique for crating images, diagrams or animation to communicate a message.

* 1. **OpenGL:**

**OpenGL** is a software interface to graphics hardware. This interface consists of about120distinct commands, which you use to specify the objects and operations needed to produce interactive three-dimensional applications. OpenGL is designed to work efficiently even if the computer that displays the graphics you create isn’t the computer that runs your graphics program. This might be the case if we work in a networked computer environment where many computers are connected to one another by wires capable of carrying digital data. In this situation, the computer on which our program runs and issues OpenGL drawing commands is called the client, and the computer that receives those commands and performs the drawing is called the server.

The format for transmitting OpenGL commands (called the protocol) from the client to the server is always the same, so OpenGL programs can work across a network even if the client and server are different kinds of computers. If an OpenGL program isn't running across a network, then there's only one computer, and it is both the client and the server.

Most of our applications will be designed to access OpenGL directly through functions in three libraries. They are:

1. **MainGL:** library has names that begin with the letter gl and stored in library usually referred to as GL.
2. **OpenGL Utility Library (GLU):** this library uses only GL functions but contains code for creating common objects and simplifying viewing.
3. **OpenGL Utility Toolkit (GLUT):** this provides the minimum functionality that should be accepted in any modern windowing system.

**OpenGL Overview:**

* Open graphics library is the interface between the graphics program and graphics hardware. ***It is streamlined.***
* OpenGL supports features like 3-dimensions, lighting, anti-aliasing, shadows, textures, depth effects, etc.
* ***It is system independent.*** It does not assume anything about the hardware or operating system and is only concerned with efficiently rendering mathematically described scenes.
* ***It is a state machine.*** At any moment during execution of a program, there is a current model transformation.
* ***It is a rendering pipeline.*** The rendering pipeline consists of the following steps:
* Defines objects mathematically.
* Arranges objects in space relative to a viewpoint.
* Calculates the color of objects.
* Rasterizes the objects.
  1. **Applications of Computer Graphics:**

The development of computer graphics has been driven both by the needs of the user community and by the advances in hardware and software. The applications of computer graphics are many and varied; it is used in almost all areas from science, engineering, medicine, business, industries, art, government, education and training.

Following are some of the applications of computer graphics:

* **Display of information:** Computer graphics is a good medium to convey the information among people. We have computer plotting packages that provide a variety of plotting techniques and color tools that can handle multiple large data sets.This technique is used to generate scanning reports in medical field, outcomes in research field, live game scores etc…
* **Design:** Professional such as engineering and architecture are concerned with design. The architects use the tool to design the upcoming plot, automobile and mechanical engineers can design the vehicle and machines by using CAD. The use of interactive graphical tools in computer-aided design (CAD) pervades field including such as architecture and the design of very-large-scale integrated (VLSI) circuits.
* **Simulation and animation:** Once graphics systems evolved to be capable of generating sophisticated images in real time, engineers and researchers began to use them as simulators which mimics the real world situation with animation. The most important uses are graphical flight simulators to train pilots, water reservoir simulator, games and educational software.
* **User interfaces:** Interaction with computers has become dominated by a visual paradigm that includes windows, icons, menus, and plotting device, such as mouse. User interfaces demonstrate the variety of the tools available in high level modeling packages and the interactive devices the user can employ in modeling geometric objects.
* **Entertainment:** Although the methods used in CG image processing overlap, the two areas are concerned with fundamentally different operations. In CG a computer is used to create picture which includes techniques to modify existing pictures such as photo scans, TV scans for entertainment purpose.

**ADVANTAGES OF COMPUTER GRAPHICS**

* Industry standard
* Stable
* Reliable and Portable
* Evolving
* Scalable
* Easy to use
  1. **GLUT:**

GLUT is a complete API written by Mark Kilgard which lets you create windows and handle the messages. It exists for several platforms, that means that a program which uses GLUT can be compiled on many platforms without (or at least with very few) changes in the code.

* 1. **OPENGL RENDERING PIPELINE:**

Most implementations of OpenGL have a similar order of operations, a series of processing stages called the OpenGL rendering pipeline. This ordering, as shown in Figure 1-2, is not a strict rule of how OpenGL is implemented but provides a reliable guide for predicting what OpenGL will do.

The following diagram shows the Henry Ford assembly line approach, which OpenGL takes to processing data. Geometric data (vertices, lines, and polygons) follow the path through the row of boxes that includes evaluators and per-vertex operations, while pixel data (pixels, images, and bitmaps) are treated differently for part of the process. Both types of data undergo the same final steps (rasterization and per-fragment operations) before the final pixel data is written into the framebuffer.

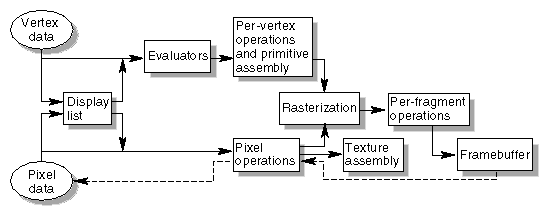


Figure 1.5: Order of Operations

**Chapter 2**

**SYSTEM REQUIREMENT**

A software requirement definition is an abstract description of the services which the system should provide, and the constraints under which the system must operate. It should only specify the external behavior of the system.

* 1. **Functional Requirements:**

Functional requirements defines the internal working of the software i.e. the calculations, technical details, data manipulation and processing and other specific functionality that shows how the cases are to be satisfied and how they are support by non-functional requirement, which impose constraints on the design or the implementation.

The following are the functional requirements:

* The ability to perform correct operation when the key is pressed.
* When the corresponding menu is selecting option should be performed.
  1. **Non Functional requirements:**

Nonfunctional requirement are requirement which specify criteria that be used to judge the operation of the system, rather than behaviors.

The following are the Non Functional Requirements:

* The application should provide a simple interface.
* Adding the different type of objects the same time.
* Specifying the type of move navigation
  1. **Hardware and Software Specification used in the project:**
     1. **Minimum Hardware Specification:**
* 64Mb RAM
* 500Mb Hard Disk space
* Pentium III and above
* 24 bit color monitor
  + 1. **Software Specification:**
* Windows 98 / XP / 7
* Microsoft Visual Studio 10.0
* VC++

**Chapter 3**

**PROJECT DESIGN**

**3.1 FLOW CHART:**

The figure explains the flow of the project ‘**PING PONG’.**

Start

Main

Initialize Open GL Call back functions

Welcome Screen Display

Menu

3. Two Players

4. Exit

2. Single Player Hard

1. Single Player Easy

Events

End

The window setup is done by certain functions called in the main function. The menu function is invoked through DisplayFunc which displays the options and in turn invokes the following functions based on the choice of the user: Single player easy, Single player hard, Two players and Exit. When any of the menu options is selected the particular event occurs.

**Chapter 4**

**Implementation**

This project can be developed by Visual C++ using OpenGL, which has become a widely accepted standard for developing graphics applications.

**4.1 Algorithm for Different Functions:**

**4.1.1main () Function:**

Step1. Create window and display the menu on the screen.

Step2. Wait for the input from the keyboard. As any key is pressed from the keyboard the corresponding function in the menu is performed.

Step3. After the completion of the game the game is closed.

**4.1.2 menu () Function:**

Step1. If no key is pressed till now from the key board the menu is displayed on the screen.

Step2. If the user opts for “Single player easy game” or “Two player game” then two paddles of equal size and a ball is displayed on the screen. The size, colour and position of the paddle and the ball are specified in this function.

Step3. If the user opts for “Single Player hard” game then the at the user side(right) a small paddle is displayed and at the CPU side medium sized paddle is shown.

Step4. If the user opts for “Exit” then a break is introduced and we come out of the program.

**4.1.3 TimerFunction() Function:**

Step1. If no key is pressed till now from the keyboard then it waits for any key to be pressed by calling itself recursively.

Step2. If the user has opted for “Single player easy” game then he gets medium sized paddle.

Step3. If the user has opted for “Single player hard” game then he gets a small sized paddle.

Step4. User can move his paddle up and down using up and down arrow keys. The CPU paddle moves according to the movement of the ball, so for user it moves automatically. As the ball hits the users paddle it reverses its direction and also increases its speed. This is implemented by changing the coordinates of the ball on intersecting with the coordinates of the paddle.

Step5. If the user has opted for “Two player game” then two paddles of medium size are displayed on either sides of the screen. One paddle is moved by using up and down keys and the other is controlled using ‘a’ and ‘z’ keys. This is implemented by changing the coordinates of the paddle when the respective keys are pressed.

Step6. If the game is over then a winning or losing message is displayed in the message box.

Step7. If the game is not over then this function is repeated until the game is over.

**4.2Built-in Functions:**

* **glLoadIdentity():**
* Replaces the current matrix with the identity matrix. It is semantically equivalent to calling glLoadMatrix with the 4x4 identity matrix.
* **glClearColor():**
* Specifies the red, green, blue and alpha values used by glClear to clear the color buffer. Values specified by the glClearColor are clamped to the range [0-1].
* **glMatrixMode():**
* Sets the current matrix mode. Mode can be any of the values:

1. GL\_PROJECTION: Applies subsequent matrix operations to the projection matrix stack.
2. GL\_MODELVIEW: Applies subsequent matrix operations to model view the matrix stack

* **glutDisplayFunc():**
* Register the display function that is executed when the window needs to be redrawn.
* **glutPostRedisplay():**
* Request that the display callback be executed after the current callback returns.
* **glutMainLoop():**
* Cause the program to enter an event processing loop. It should be the last statement in main ().
* **glColor3f():**
* Sets the current color.
* **glVertex2f():**
* Specifies a vertex.
* **glutSwapBuffers():**
* glutSwapBuffers swaps the buffers of the current window if double buffered.
* **glutReshapeFunc():**
* glutReshapeFunc sets the reshape callback for the current window.
* **void glutInitDisplayMode(unsigned int mode);**
* Requests a display with the properties in the mode; the value of the mode is determined by the logical OR of options including the color model (GLUT\_RGB,GLUT\_INDEX) and buffering (GLUT\_SINGLE,GLUT\_DOUBLE).
* **void glutCreateWindow(char \*title);**
* Creates a window on display; the string title can be used to label the window. The return value provides a reference to the window that can be used when there are multiple.
* **void glutKeyboardFunc(void \*f(char key, int width, int height));**
* Registers the keyboard callback function f. The callback function returns the ASCII code of the key pressed and the position of the mouse.
* **void glViewPort(int x, int y, GLsizei width, GLsizei height);**
* Specifies the width\*height viewport in pixels whose lower left corner is at (x,y) measured from the origin of the window.
* **void glutInitWindowSize(int Width, int Height);**
* Specifies the initial height and width of the window in pixels.
* **glutBitmapCharacter();**
* This function is usedto put some bitmapped text on the screen.
* **glutTimerFunc**();
* Sets the Timer callback for the current window.

**4.3 User Defined functions:**

* **Output()**
* This function is used to display the strings on the screen.
* **SetupRC()**
* This function is used to set the background color on the screen.
* **ChangeSize()**
* This function is used to change the size of the game screen using window size of the game window

**Chapter 5**

**TESTING**

A brief description of the events that take place before, during and after gameplay with respect to the keys that are pressed.

**5.1 Welcome Screen:**

Table 5.1 Test cases for welcome screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.No** | **Test Case** | **Expected Result** | **Actual Result** | **Conclusion** |
|  | Press ‘1,2,3’ to start the game | It is expected to begin the game | Game commences | PASS |
|  | Press 4 to quit the game | It is expected to quit the game | Quit | PASS |

**5.2 Gameplay with Keyboard**

Table 5.2 Test cases during gameplay with keyboard

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.No** | **Test Case** | **Expected Result** | **Actual Result** | **Conclusion** |
|  | Press 1 to start game in single player easy mode | It is expected to open a game in single player easy mode | Game is started | PASS |
|  | Press 2 to start game in single player hard mode | It is expected to open a game in single player hard mode | Game is started | PASS |
|  | Press 3 to start game in two player mode | It is expected to open a game in two player mode | Game is started | PASS |

**5.3 Paddle movement with Keyboard**

Table 5.3 Test cases of paddle movement with keyboard

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.No** | **Test Case** | **Expected Result** | **Actual Result** | **Conclusion** |
|  | Press top and bottom arrow key to move the user paddle for single player mode | It is expected that user paddle will move up and down respectively | Paddle moves up and down | PASS |
|  | Press “a” and “z” for player 1 and top and bottom arrow key to move the user 2 paddle in two player mode | It is expected that both the paddle will move according to the key is pressed | Paddle moves for both the player | PASS |

**5.4 System testing**

System testing involves integrating two or more components that implement the system function or features and then testing this integrated system.System testing is concerned with testing an increment to be delivered to the customer in a waterfall process.

Table 5.4 Test Table for System Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl. No** | **Action** | **Input** | **Expected output** | **Actual output** | **Comment** |
| 1 | Testing of source code | Source code | Build successfully | Build successfully | Pass |

**Chapter 6**

**Screen Shots**

**6.1 Welcome Screen**

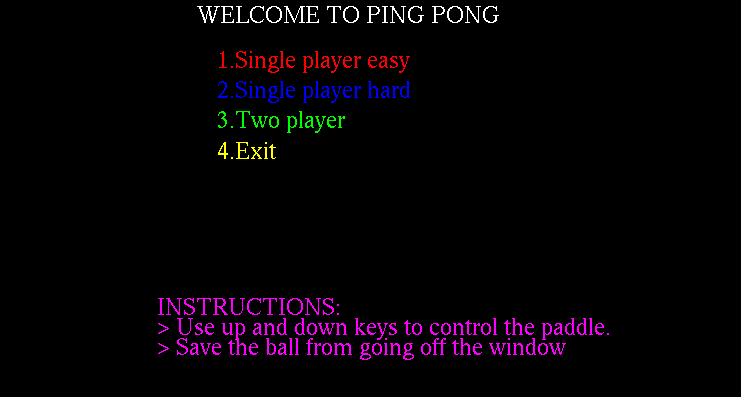
****

Figure 6.1 Display the options available to select from the welcome screen.

**6.2 Single Player Easy**

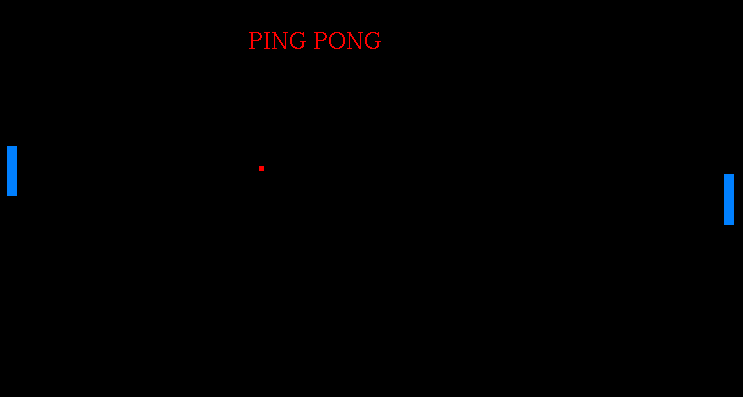
****

Figure 6.2 Single Player easy mode game in progress



Figure 6.2.1 User lost the game by the CPU in easy mode

**6.3 Single Player Hard**

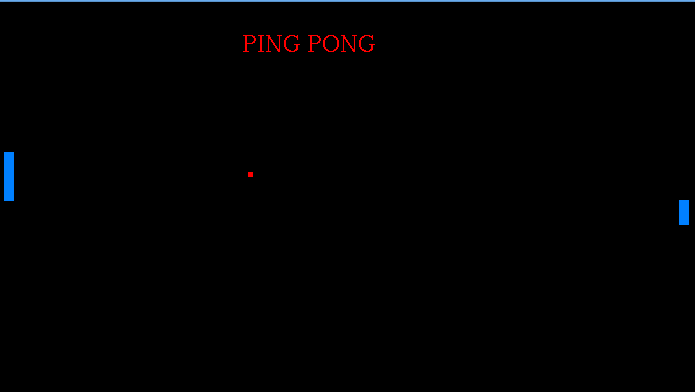


Figure 6.3 Single Player hard mode game in progress

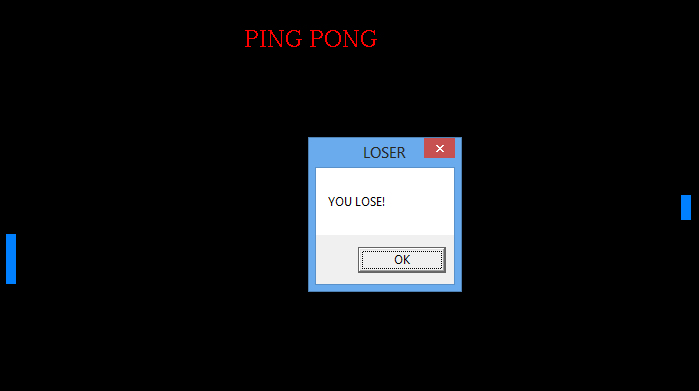


Figure 6.3.1 User lost the game by the CPU in hard mode

**6.4 Two Player**

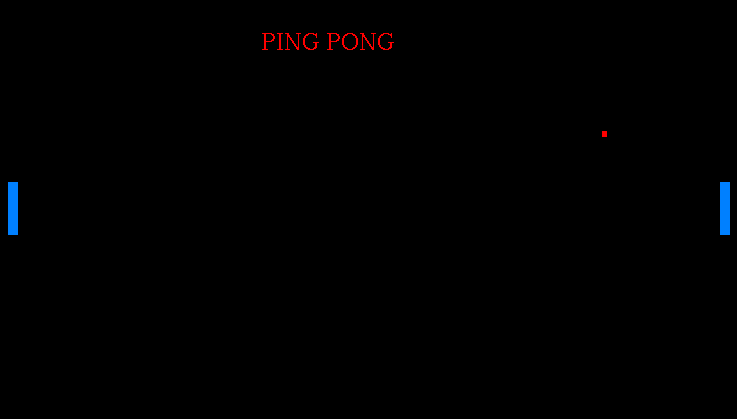
****

Figure 6.4 Two player game in progress

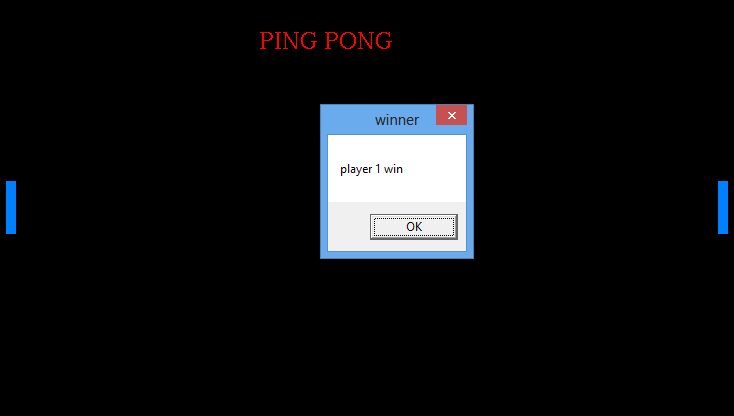


Figure 6.4.1 Player one won the game



Figure 6.4.2 Player two won the game

**6.5 Credits Page**

Figure 6.5 Final credit page



**Chapter 7**

**CONCLUSION**

We started with a modest aim, with no prior experience in any programming projects such as this, but ended up in learning many things, fine tuning my programming skills and getting into the real world of software development with an exposure to corporate environment. **During thedevelopment of any software of significant utility, we are faced with the trade-off between speed of execution and amount of memory consumed.** This is a simple interactive application. It is extremely user friendly and has the features, which makes simple graphics project. It has an open source code and no security features has been included. The user is free to alter the code for future enhancement. Checking and verification of all possible types of the functions are taken care. Care was taken to avoid bugs. Bugs may be reported to creator as the need may be. So, we conclude on the note that we are looking forward to develop such projects with an appetite to learn more in the field of computer science.

**Chapter 8**

**REFERENCES**

[1] Edward Angel, “Interactive Computer Graphics”,5th edition, Pearson Education,2005

[2]Jackie L. Neider, Mark Warhol, Tom R. Davies,” OpenGL Red Book”, 2nd Revised Edition,2005

[3]Donald D Hearn and M.PaulinenBaker,“Computer Graphics with OpenGL”,3rd edition.

[4] F.S.Hill and Stephen Kelly, “Computer Graphics using OpenGL”,3rd edition

[5]Richard.S.Wright(jr) and Michael sweet, ”OpenGL superbible”,2nd ,3rd and 5th edition.

**WEBSITES REFERRED:**

1. https://www.opengl.org/documentation/specs/glut/spec3/node20.html
2. https://www.opengl.org/documentation/specs/glut/spec3/node1.html
3. https://en.wikipedia.org/wiki/OpenGL
4. https://www.youtube.com/watch?v=-\_CqxqQHD-M