CHAPTER 1

INTRODUCTION

It would be amazing to have a simple game with a fantastic interface, rigidity and flexibility to serve the gameplay of anybody who intends to play the same. This project serves the very purpose. Now coming to the structure of the project, there are simple implications and instruction that are used to build and inherit certain aspects of the same. The game play includes the user to select from different difficulty levels and playing options that include keyboard and mouse interface as input devices. The details will be dealt with, in the further sections. Different splash screens have been integrated as well to enhance the user experience of the player. Maximum usage of OpenGL functions has been promoted in building this project and to the developers' knowledge of the same in order to give a great finish to the game.

1.1 Problem Statement

To create a simple game based on openGL libraries. The game must be based on a weightlifting challenge where the necessary components must be designed and respective functions are to be included to support required inputs and process the same to produce the desired output. All implications must be visually supported and viewer oriented.

1.2 Objectives

The objective would include the implementation of visual effects to display all the components of the game such as the weightlifting bar, the scaling standards, winning positions, etc. Including the use of mouse and keyboard input processing functions to allow a clear interfacing with the project.

Splash screens are to be included to allow the navigation within the game and select between the choices as provided by the developers.

The major objective would be the showcasing of computer graphics usage to create or recreate ideal situations for development of visual applications such as games, visual softwares, etc.

1.3 Scope

Computers have become a powerful tool for the rapid and economical production of pictures. There is virtually no area in which graphical displays cannot be used to some advantage, and so it is not surprising to find the use of computer graphics so widespread. Although early applications in engineering and science had to rely on expensive and cumbersome equipment, advances in computer technology have made interactive computer graphics a practical tool. Today, we find computer graphics used routinely in such diverse areas as science, engineering, medicine, business, industry, government, art, entertainment, advertising, education, and training. Computer graphics is concerned with all aspects of synthesizing and manipulating images. All of these have an adverse relation with the animation and gaming industries where we can observe an extensive usage of computer graphics and its wide varieties.

CHAPTER 2

LITERATURE SURVEY

People use the term “computer graphics” to mean different things in different context. Computer graphics are pictures that are generated by a computer. Everywhere you look today, you can find examples, especially in magazines and on television. Some images look so natural you can’t distinguish them from photographs of a real scene. Others have an artificial look, intended to achieve some visual effects.

There are several ways in which the graphics generated by the program can be delivered.

- Frame- by- frame: A single frame can be drawn while the user waits.
- Frame-by-frame under control of the user: A sequence of frames can be drawn, as in a corporate power point presentation; the user presses to move on to the next slide, but otherwise has no way of interacting with the slides
- Animation: A sequence of frames proceeds at a particular rate while the user watches with delight.
- Interactive program: An interactive graphics presentation is watched, where the user controls the flow from one frame to another using an input device such as a mouse or keyboard, in a manner that was unpredictable at the time the program was written. This can delight the eye.

2.1 History

OpenGL was developed by **‘Silicon Graphics Inc‘**(SGI) on 1992 and is popular in the gaming industry where it competes with the Direct3D in the Microsoft Windows platform. OpenGL is broadly used in CAD (Computer Aided Design), virtual reality, scientific visualization, information visualization, flight simulation and video games development.

OpenGL is a standard specification that defines an API that is multi-language and multi-platform and that enables the codification of applications that output computerized graphics in 2D and 3D.

The interface consists in more than 250 different functions, which can be used to draw complex tridimensional scenes with simple primitives. It consists of many functions that help to create a real world object and a particular existence for an object can be given.

2.2 Characteristics

- OpenGL is a better documented API.
- OpenGL is also a cleaner API and much easier to learn and program.
- OpenGL has the best demonstrated 3D performance for any API.
- Microsoft's Direct3D group is already planning a major API change called Direct Primitive that will leave any existing investment in learning Direct3D immediate mode largely obsolete.

2.3 Computer Graphics Library Organisation

OpenGL stands for Open Source Graphics Library. Graphics Library is a collection of APIs (Application Programming Interfaces).

Graphics Library functions are divided in three libraries. They are as follows-

i. GL Library (OpenGL in Windows)

ii. GLU (OpenGL Utility Library)

iii. GLUT (OpenGL Utility Toolkit)

Functions in main GL library name function names that begin with the letter ‘gl’.

- GLU library uses only GL functions but contains code for creating objects and simplify viewing.

- To interface with the window system and to get input from external devices GLUT library is used, which is a combination of three libraries GLX for X windows, 'wgl' for Windows and 'agl' for Macintosh.
- These libraries are included in the application program using preprocessor directives. E.g.: #include<GL/glut.h>
- The following figure shows the library organization in OpenGL.

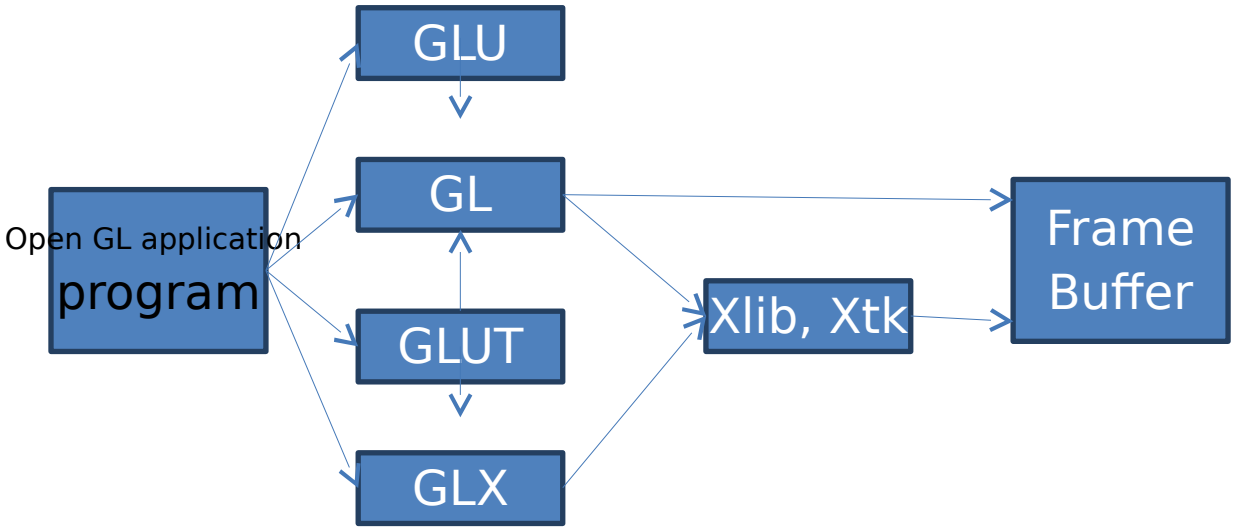

Fig 2.1 Library Organization

1.3 Graphics System and Functions

- Graphics system and functions can be considered as a black box, a term used to denote a system whose properties are only described by its inputs and output without knowing the internal working.
- Inputs to graphics system are functions calls from application program, measurements from input devices such as mouse and keyboard.
- Outputs are primarily the graphics sent to output devices.

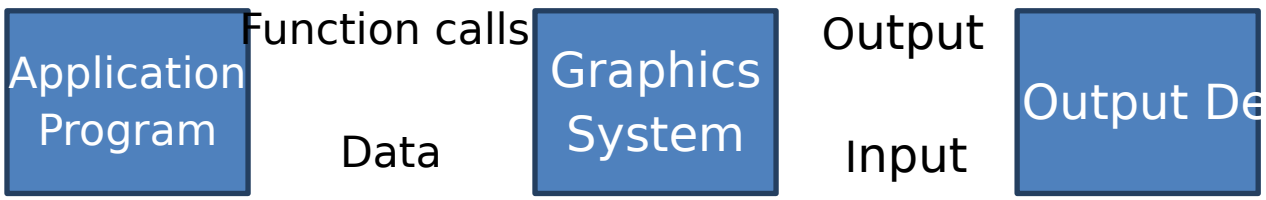

Fig 2.2 Graphics System as a Black Box

API's are described through functions in its library. These functions are divided into seven major groups.

1) Primitive Functions:

   Primitive functions define the low level objects or atomic entities that a system can display, the primitives include line segments, polygons, pixels, points, text and various types of curves and surfaces.

2) Attribute Functions:
   Attribute Functions allow us to perform operations ranging from choosing the color to display a line segment, to packing a pattern to fill inside any solid figure.
3) Viewing Functions:
   Viewing functions allow us to specify various views.
4) Transformation Functions:
   Transformation functions allow us to carry out transformation of objects such as rotation, translation and scaling.
5) Input Functions:
   Input functions allow us to deal with the diverse forms of input that characterize modern graphics system. It deals with devices such as keyboard, mouse and data tablets.
6) Control Functions:
   Control Functions enable us to communicate with the window system, to initialize the programs, and to deal with any errors that occur during the execution of the program.
7) Query Functions:
   Query Functions provides information about the API.

CHAPTER 3

SYSTEM REQUIREMENTS

Requirements analysis is critical for project development. Requirements must be documented, actionable, measurable, testable and defined to a level of detail sufficient for system design. Requirements can be architectural, structural, behavioural, functional, and non-functional.A software requirements specification (SRS) is a comprehensive description of the intended purpose and the environment for software under development.

3.1Hardware Requirement

- Minimum of 2GB of main memory
- Minimum of 3GB of storage
- Keyboard
- Mouse
- Display Unit
- Dual-Core or AMD with minimum of 1.5GHz speed

3.2 Software Requirement

- Windows – XP/7/8
- Microsoft Visual Studio C/C++ 7.0 and above versions
- OpenGL Files
- DirectX 8.0 and above versions

Header Files

- glut.h

Object File Libraries

- glut32.lib

DLL files

- glut32.dll

CHAPTER 4

DESIGN AND IMPEMENTATION

4.1 Header Files Used

i. **The "stdafx.h" header file:** #include "stdafx.h"

This header file includes other header files such as <stdio.h> and <tchar.h>.

ii. **The <GL/glut.h> header file:** #include<glut.h>

OpenGL makes heavy use of macros to increase code readability and avoid the use of magic number. Thus, strings such as GL_POINTS and GL_LINES are defined in header (.h) files.

4.2 OpenGL API's Used

This gives a brief description of how the list of OpenGL API's used to achieve the exact solution of the project.

Function prototype	Usage
glutInit()	Used to initialize the GLUT library
glutInitWindowSize()	Used to set the initial Window Size
glutDisplayFunc(function)	Used to set the display call back function
glutInitDisplayMode()	Used to set the mode of display such as colour mode, no of buffers to be used etc.
glutMouseFunc(function)	Used to set Mouse call back function.
glAttachMenu()	Use to attach a mouse button for the current window to the identifier of the current

	menu.
glutPostRedisplay()	Marks the current window to be redisplayed, i.e. calling the display function.
glutCreateMenu()	Uses to create a new pop-up menu.
glutReshapeFunc(function)	Used to maintain the aspect ratio.
glutKeyboardFunc(function)	Used to set keyboard callback functions.
glutMainLoop()	Used to loop the main function infinitely to display constant output window.
glBegin()	Used to begin the plotting of vertices for desired structure.
glRasterPos2f(x,y)	Used to determine the position of the text to be displayed on output screen
glutBitmapCharacter(font, string)	Used to determine the fonts and character array.

4.3 User Defined Functions

This gives a brief description of how the various functions are used to achieve the exact solution of the project.

- **void welcome()**

  This function is used to display the welcome Screen. It displays the welcome message. To go to main menu we have to press “s” key on the keyboard.

- **void menug()**

  This function is used to display the main menu on screen. It displays the three different option to user. Options such as :

  1.Start Game.

  2.How to Play.

3.About Game.

- **void startg()**

  This function is used to start the game. In this function we draw different number of polygon using GL_POLYGON primitive .By calling this function we see an object with two stands on screen and a finish line.

  We have to use the left mouse button to lift the object till it crosses the finish line to complete the objective. In this function we also display the chances left for user to complete the objective.

- **void controlsg()**

  This function is used to display the message about the controls of the game.

- **void aboutg()**

  This function is used to display the message about the project.

- **void wing()**

  This function is used to display the "You Won !" message when you have completed the objective.It also shows the time taken by user to complete the objective.

- **void loseg()**

  This function is used to display the "You lose !" message when you are unable to complete the objective.

- **void display()**

  This function is the main display function. It checks for various conditions to display the screen and call the functions mentioned above accordingly.

- **void keyboard(unsigned char key, int mx, int my)**

This function is used to check for keyboard interrupt and perform the action accordingly. Parameter key stores the value of the key pressed and mx and my are the x and y coordinated respectively.

This function is used to change the screen. 'S' or 's' to go to main menu from welcome screen ,'1' to start the game,'2' to display the "controls" screen,'3' to display "About game" screen and 'b' or 'B' to move back to main menu.

- **void mouse(int btn,int state,int x,int y)**

  This function is used to check which mouse button is clicked along with the coordinates (x,y) and perform the action accordingly. If the Left mouse button is pressed and the state is down the object moves up .If the there is more than 1 second of delay between to mouse clicks then object will resets to its initial postion and chance will be decremented by 1.

- **void myReshape(int w,int h)**

  This function is used to reshape the display .

- **void menu(int id)**

  This function is used to display the menu on right mouse click. It takes one parameter id which shows the action to be performed on menu. If id is 1 then it takes the screen to main menu, if id is 2 then it changes the level to Level 1, if id is 3 then it changes the level to Level 2, if id is 4 then it changes the level to Level 4.if id is 5 then it exits the game.

CHAPTER 5

SNAPSHOTS

5.1 Welcome Screen

Fig 5.1 Welcome Screen

- This is the welcome screen .It displays the welcome message and shows us the option to press 'S' key to start the game and go to main menu. This is the first screen to be displayed.

5.2 Main menu Screen

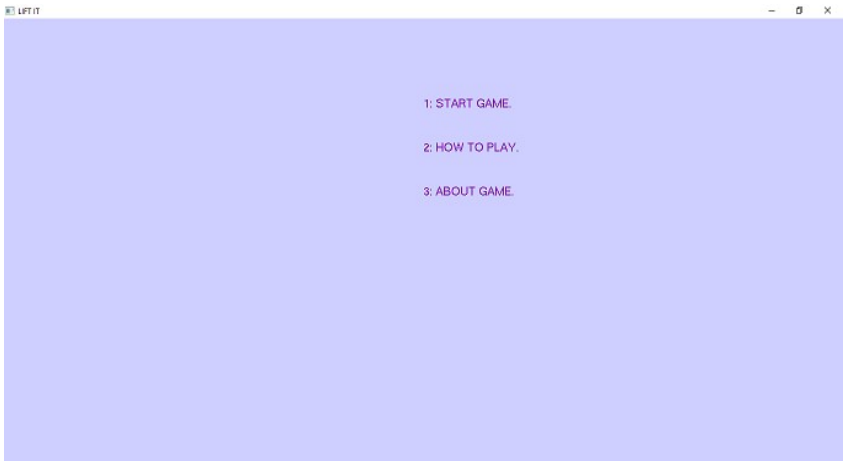

Fig 5.2 Main Menu Screen

- Fig 5.2 shows the main screen .It shows the different options. Press 1 to start the game, press 2 to go to how to play screen and press 3 to show about game screen.

5.3 Main Screen

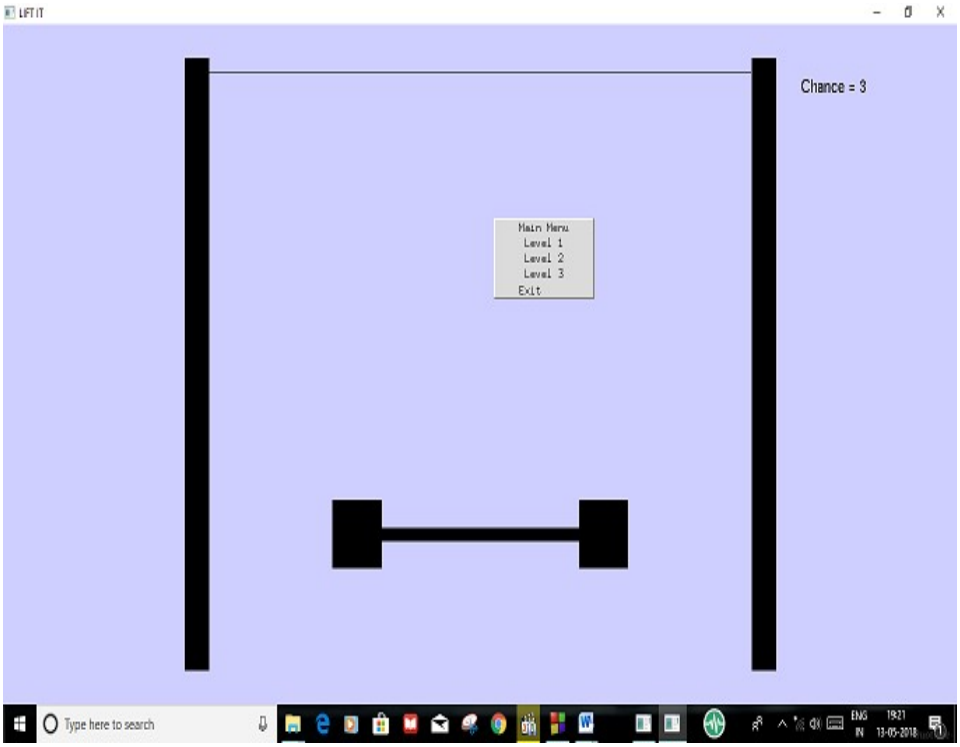

Fig 5.3 Main Screen

- This shows the main screen of the game where an object is displayed at the bottom and it shows the two pillars and a finish line at top, which the object have to cross to complete the objective. To move the object you have to click the left mouse button repeatedly with not more than 1 second of delay. It also shows the menu on right mouse button click. First option of the menu is to go to main menu screen followed by change of levels followed by option to exit the game.

5.4 Win Screen

- Fig 5.4 shows the Win screen it displays the “YOU WIN” message and also displays the time taken to complete the objective. This screen is displayed when user has completed the objective.

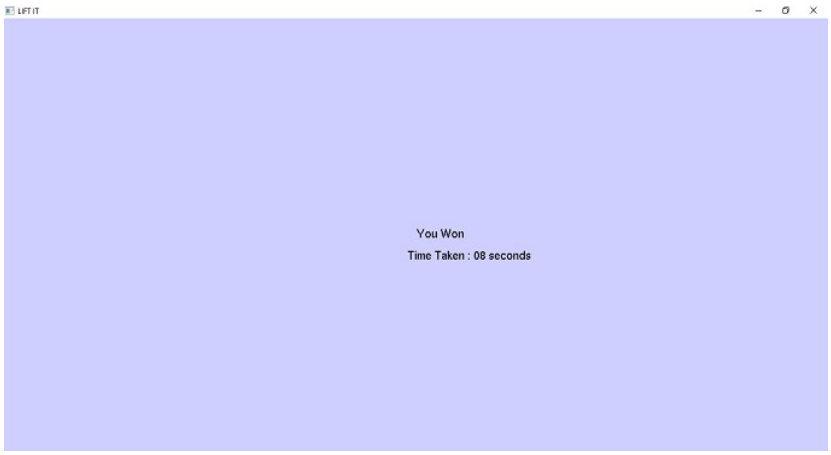

Fig 5.4 Win Screen

5.5 Lose Screen

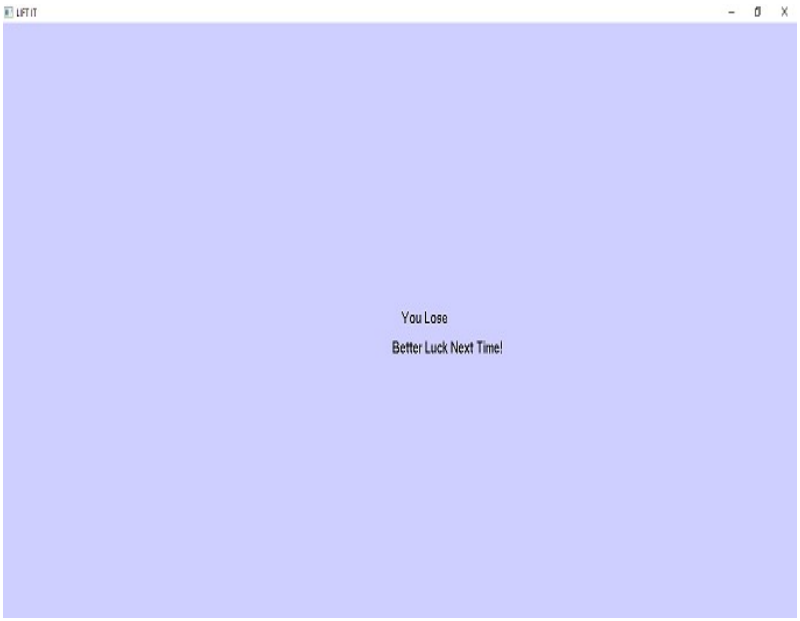

Fig 5.5 Lose Screen

- This Screen displays the “YOU LOSE” and “Better Luck Next Time!” message on screen .This screen is displayed when user is unable to complete the objective with given number of chances.

5.6 Controls Screen

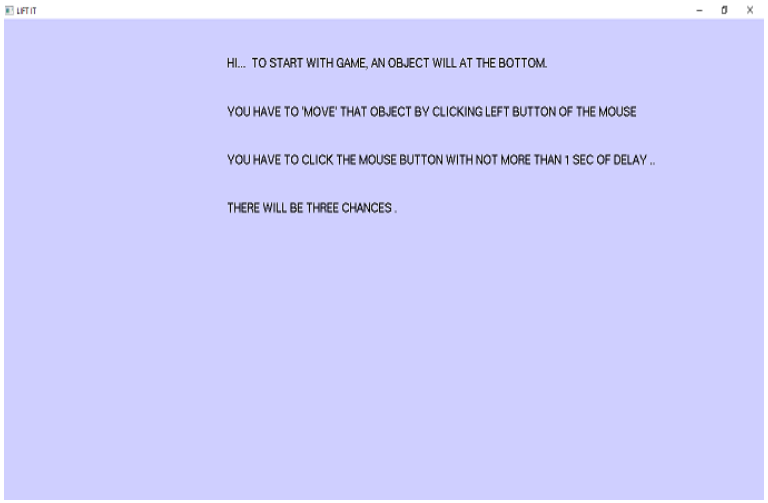

Fig 5.6 Controls Screen

- This page displays all the controls and instructions that can be used to handle the game.

5.7 About Screen

Fig 5.7 About Screen

- This screen describes and gives a brief overview of the game and its features.

CONCLUSION

An attempt has been made to develop an OpenGL package which meets necessary requirements of the user successfully. Since it is user friendly, it enables the user to interact efficiently and easily. The development of the mini project has given us a good exposure to OpenGL by which we have learnt some of the technique which help in development of animated pictures, gaming. Hence it is helpful for us even to take up this field as our career too and develop some other features in OpenGL and provide as a token of contribution to the graphics world.

FUTURE ENHANCEMENT

- The project can be enhanced in a number of ways including aspects such as design, speed and compatibility based on user requirements.
- Inputs devices such as joysticks also can used to accept inputs from users.
- The visual effects can be enhanced using complex display attributes such as shades, ambience, etc. But, this may also result in complexity of code that has to handled with superior knowledge of the working aspects.
- Additional characters or objects may be added to make the visuals more realistic and advanced.
- The front end design can also be changed if programmer wishes. There are other many things that can also be changed such as colours of objects, sizes, orientations, positions etc.
- More information can be included as a reference purposes such as different useful books and websites and journals.
-

BIBLIOGRAPHY

[1] Edward Angel: Interactive Computer Graphics A Top-Down Approach with OpenGL, 2nd Edition, Addison-Wesley, 2000.

[2] Donald Hearn and Pauline Baker: Computer Graphics- OpenGLVersion,3rd Edition, Pearson Education,2004.

[3] F.S. Hill Jr: Computer Graphics using openGl,3rd Edition, PHI,2009.

[4] James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes, Computer Graphics,

Pearson Education 1997.

[5] https://www.opengl.org/resources/