**CHAPTER 1**

**INTRODUCTION**

### 1.1 COMPUTER GRAPHICS:

Computer Graphics is concerned with all aspect of producing pictures or image using computer. 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.Graphics is created using [computers](http://en.wikipedia.org/wiki/Computer) and, more generally, the [representation](http://en.wikipedia.org/wiki/Representation) and [manipulation](http://en.wikipedia.org/wiki/Manipulation) of [pictorial](http://en.wikipedia.org/wiki/Pictorial)[data](http://en.wikipedia.org/wiki/Data) by a computer. The development of computer graphics has made computers easier to interact with and better for understanding and interpreting many types of data. Developments in computer graphics have had a profound impact on many types of media and have revolutionized the [animation](http://en.wikipedia.org/wiki/Animation) and [video game](http://en.wikipedia.org/wiki/Video_game) industry. The phrase “Computer Graphics” was coined in 1960 by William Fetter, a graphic designer for Boeing.

In today’s world advanced technology, interactive computer graphics has become a powerful tool for the production of realistic features. Today’s we find computer graphics used in various areas that include science, engineering, medicine, business, industry, art, entertainment etc. The main reason for effectiveness of the interactive computer graphics is the speed with which the user can understand the displayed information.The graphics in openGL provides a wide variety of built-in function. The computer graphics remains one of the most exciting and rapidly growing computer fields. It has become a common element in user interface, data visualization, TV commercials, motion picture and many other applications. The current trend of computer graphics is to incorporate more physics principles into 3D graphics algorithm to better simulate the complex interactions between objects and lighting environment.

**1.2 OPEN GL** (Open Graphics Library)**:**

OpenGL has become a widely accepted standard for developing graphics application. OpenGL is easy to learn, and it possesses most of the characteristics of other popular graphics system. It is top-down approach. OpenGL is a standard specification defining a cross-language, cross-platform API for writing applications that produce 2D and 3D computer graphics. The interface consists of over 250 different function calls which can be used to draw complex three-dimensional scenes from simple primitives. OpenGL was developed by Silicon Graphics Inc. (SGI) in 1992and is widely used in CAD, virtual reality, scientific visualization, information visualization, and flight simulation. It is also used in video games, where it competes with Direct3D on Microsoft Windows platforms.The interface between the application program and the graphics system can be specified through that set of function that resides in graphics library. The specification is called the APPLICATION PROGRAM INTERFACE (API). The application program sees only the API and is thus shielded from the details both the hardware and software implementation of graphics library. The software driver is responsible for interpreting the output of an API and converting these data to a form that is understood by the particular hardware.

Most of our applications will be designed to access openGL directly through functions in three libraries. Function in the main GL library have name that begin with the letter gl and stored in the library. The second is the openGL utility Library (GLU). Thislibrary uses only GL function but contains codes for creating common object and viewing. Rather than using a different library for each system we used available library called openGL utility toolkit (GLUT). It used as #include<glut.h>.A graphics editor is a computer program that allows users to compose and edit pictures interactively on the computer screen and save them in one of many popular “bitmap” or “raster” a format such as TIFF, JPEG, PNG and GIF.

Graphics Editors can normally be classified as:

* 2D Graphics Editors.
* 3D Graphics Editors.

A 3D Graphics Editor is used to draw 3D primitives Rectangles, Circle, polygons, etc. and alter those with operations like cut, copy, paste. These may also contain features like layers and object precision etc.

3D Graphics Editor should include the following features:

* Facilities: Cursor Movement, Editing picture objects.
* Good User Interface: GUI / Toolbars / Icon based User Interface.

Computer Graphics is concerned with all aspects of producing pictures or images using a computer. A particular graphics software system called OpenGL, which has become a widely accepted standard for developing graphics applications.The applications of computer graphics in some of the major areas are Display of information, design, simulation and animation,User interfaces.OpenGL is a software interface to graphics hardware. This interface consists of about 150 distinct commands that you use to specify the objects and operations needed to produce interactive three-dimensional applications.

The project named “Floating ball” uses OpenGL software interface and develops 2D images. This project uses the techniques like Translation, motion, display list, transformation techniques, etc.

In proposed system, the OpenGL is a graphic software system designed as a streamlined, hardware-independent interface to be implemented on many different hardware platforms. To achieve these qualities, no commands for performing windowing tasks or obtaining user input are included in OpenGL; instead, you must work through whatever windowing system controls the particular hardware you're using.OpenGL doesn't provide high-level commands for describing models of three-dimensional objects. Such commands might allow you to specify relatively complicated shapes such as automobiles, parts of the body, airplanes, or molecules. With OpenGL, you must build up your desired model from a small set of *geometric primitives* - points, lines, and polygons.

Keyboard

Graphics library (API)

Drivers

Application program

Mouse

Fig1.1: Application programmers model of graphics system

The interface between an application program and a graphics system can be specified through a set of functions the resides in a graphics library .These specification are called the application programmer’s interface (API).The application programmer see only the API and is thus shielded from the details of both the hardware and thesoftware implementation of the graphics library. The software drivers are responsible for interpreting the output of the API and converting this data to a form that is understood by the particular hardware.

**CHAPTER 2**

**REQUIREMENT SPECIFICATIONS**

**2.1 Hardware requirements**:

* Pentium or higher processor.
* 16 MB or more RAM.
* A standard keyboard, and Microsoft compatible mouse
* VGA monitor.
* If the user wants to save the Created files a secondary storage medium can be used.

**2.2 Software requirements**:

* Programming Language- C/C++ using OpenGL
* Operating System- Linux
* Compiler- C Compiler
* Graphics library- <GL/glut.h>
* OpenGL 2.0

**CHAPTER 3**

**ABOUT THE PROJECT**

Apple Gardenis a 2-dimentional apple game, wherein the user has to catch the apples in a net. The apples fall from three trees, with different speeds. The user has to catch these apples by moving the net either to the left, centre, or to the right. The scores keep incrementing as the apples are caught.

The user is provided with a menu and the options are to be selected using the RIGHT MOUSE BUTTON. The menu is implemented to select the options as mentioned below.

* START: To start the game.
* LEVEL: To change the difficulties of the game. It consists of two levels, in which the speed of the falling apples increases with the increase in level.
* INSTRUCTIONS: A guide to how to play the game.
* EXIT: To quit the game.

The project is based on a simple window co-ordinates and using recursive techniques in OpenGL.

* The package is developed using computer graphics with OpenGL
* Migration from text editor to OpenGL
* To show that implementation of translation is easier with OpenGL
* To show that implementation of rotation is easier in OpenGL

**3.1 Movement of the Net:**

The net can be moved in two different ways on the screen:

* RIGHT
* LEFT

These movements are designed for the user to catch the falling apples correctly from the trees, in a given speed.The position of the net has been marked using the coordinate system. We make use of X coordinate and Y coordinate.

**CHAPTER 4**

**IMPLEMENTATION**

**4.1 Major Functions:**

* **SCORE DISPLAY**

This function is used to display the score. We make use of Raster Position, wherein the position is maintained with subpixel accuracy; as openGL maintains a 3-D position in window coordinate system. It is used to position the pixel and BitMap write operations. This consists of two parameters of integer type. The scoreboard is set to an appropriate position on the screen and the value keeps on increasing as the apples are caught into the net.

* **START SCREEN**

This function is used as the cover screen of the game. The key, G has to be pressed from the keyboard to start the game. We make use of opnGl primitive functions, glClear, glClearColor and glColor3f. The string function is used to print the characters- “GRAB THE APPLES” and to start the game- “PRESS G TO START” on the screen.

* **DRAWING THE OBJECTS**

The trees, the apples and the net are drawn onto the screen using the Matrix functions, primitive functions like glColor, glVertex and glutSolidSphere to make the apples fall from their respective trees.

* **DISPLAY**

For displaying, we first display the background. The background consists of one polygons, whichis color filled polygon which give the appearance of ground and sky when implemented appropriately. In the display function we call drawtree() and net() functions. It directs to the scoredisplay() function if the apples are caught.

* **MAIN FUNCTION**

Here we specify the initial display mode, window size and position. Create a new window where the output is rendered. We provide a menu driven which provides options such as: start, level, instruction and exit. The functions displayfunc,glutSpecialFunc(), glutKeyboardFunc() are called here.

glutMainLoop(): This is used here in order to call the main function repeatedly and constantly refreshes the display at a sufficient rate to avoid flickering.

**4.2 Important Functionalities:**

**4.2.1 Display functionality:**

It Renders the program onto the screen.Uses the following functions:

* GlClear (GL\_COLOR\_BUFFER\_BIT)- Indicates the buffers currently enabled for color writing and also indicates the depth buffer.
* glFlush () — force execution of GL commands in finite time.
* GlutSwapBuffers ()-Swap the buffers ->make the result of rendering visible.
* glMatrixMode (GL\_PROJECTION) -applies subsequent matrix operations to the projection matrix stack.
* glMatrixMode (GL\_MODELVIEW)-applies subsequent matrix operations to the model view matrix stack.We adjust the viewing volume. We use the whole window for rendering and adjust point size to window size.
* The API glutBitMapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24) render the character one at a time in the 24-point proportional speed Times Roman font. glRasterPos2i(x, y) specifies the x and y co-ordinate for the current raster position.
* glBlendFunc(GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA)– defines the operation of blending for all draw buffers when it is enabled by glEnable(GL\_BLEND).

**4.2.2 Keyboard functionality:**

Manages operations by various keys pressed on the key board.This function is the one that provides the user to control the net to move on the screen to catch the falling apples.

**4.2.3 Mouse functionality**:

This is a function which is used to provide the user the option to select from the given menu. The user can check the menu options by clicking the RIGHT BUTTON.

* 1. **OPENGL APIs USED:**

**4.3.1 glutInit(&argc, argv):**

* glutInit is used to initialize the GLUT library.
* glutInit will initialize the GLUT library and negotiate a session withthe window system. During this process glutInit may cause the termination of the GLUT program with an error message to the user if

GLUT cannot be properly initialized. Examples of this situation include the failure to connect to the window system, the lack of window system support for OpenGL, and invalid command line options.

* glutInit also processes command line options, but the specific options

**4.3.2 glutInitDisplayMode(GLUT\_DOUBLE|GLUT\_RGB):**

* glutInitDisplayMode sets the initial display mode.
* GLUT\_DOUBLE

Bit mask to select a double buffered window. This overrides GLUT\_SINGLE if it is also specified.

* GLUT\_RGB

Bit mask to select an RGBA mode window.

**4.3.3 glutInitWindowPosition() and** **glutInitWindowSize() :**

* + set the initial window position andsizerespectively.
  + Usage:

VoidglutInitWindowSize(int width, int height);

VoidglutInitWindowPosition(int x, int y);

**4.3.4 glutCreateWindow():**

* + creates a top-level window.
  + Usage:

glutCreateWindow(char \*name)

**4.3.5 glutDisplayFunc():**

* + Usage:

voidglutDisplayFunc(void (\*func)(void));

* Parameter:

func : The new display callback function.

* + Description:

glutDisplayFunc sets the display callback for the current window.

**4.3.6 glutKeyboardFunc(keyboard):**

* + This OpenGL will teach you how to add keyboard interaction to your OpenGL application through the use of glut and glut's keyboard calls.

**4.3.7 glutMainLoop():**

* Causes the program to enter an event processing loop. It should be the last statement in main.

**4.3.8 glClear(GL\_COLOR\_BUFFER\_BIT):**

* void **glClear**(GLbitfield mask);

mask: Bitwise OR of masks that indicate the buffers to be cleared. The three masks are GL\_COLOR\_BUFFER\_BIT.

The values are as follows:

* GL\_COLOR\_BUFFER\_BIT

Indicates the buffers currently enabled for color writing.

**4.3.9 glutBitmapCharacter(**void \*font, int character**):**

Without using any display lists, glutBitmapCharacter renders the character in the named bitmap font. There are various font available, the one used in the program is

GLUT\_BITMAP\_TIMES\_ROMAN\_24

A 24-point proportional spaced Times Roman font.

**4.3.10 glTranslatef():**

* Usage

glTranslatef(GLfloatx,Glfloaty,GLfloat z); glTranslated(GLdoublex,GLdoubley,GLdouble z);

* Parameters

x, y, z: Specify the x, y, and z coordinates of a translation vector

* Description

glTranslate produces a translation by x y z . The current matrix is multiplied by this translation matrix, with the product replacing the current matrix. If the matrix mode is either GL\_MODELVIEW or GL\_PROJECTION, all objects drawn after a call to glTranslate are translated

**4.3.11 glLineWidth():**

* Usage:

void glLineWidth(GLfloat width);

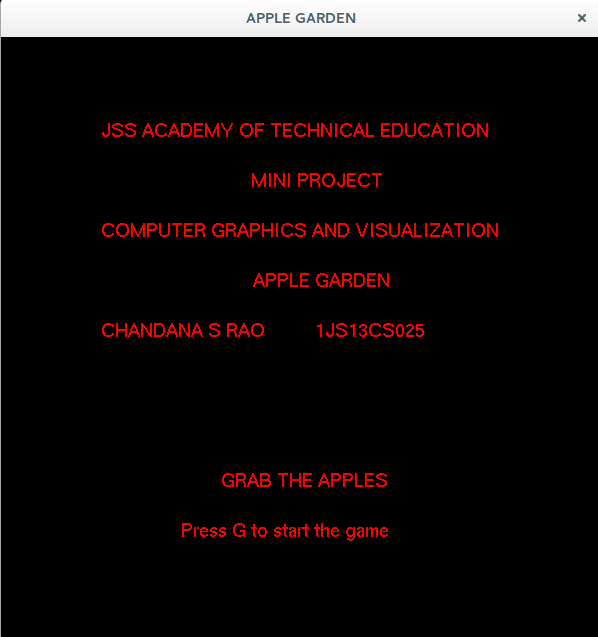
* Parameters:

Width: Specifies the width of rasterized. The initial value is 1.

* Description:

It specifies the width of rasterized lines.

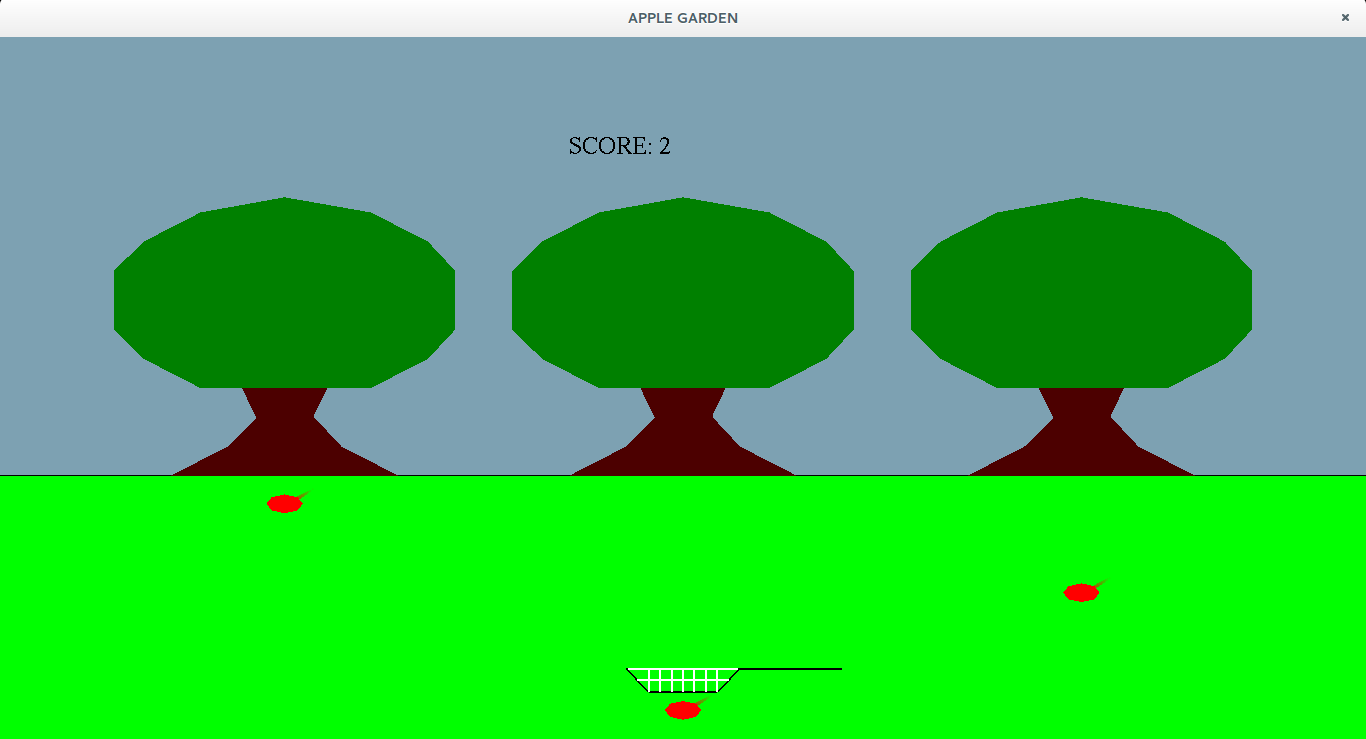
**CHAPTER 5**

** SNAPSHOTS**

Fig(5.1): Cover Page

****

Fig(5.2): Apple catch

****

Fig(5.4): Level Selection

Fig(5.3): Catch miss

Fig(b): Basket at CENTER and Catchmiss

**CHAPTER 6**

**CONCLUSION AND FUTURE SCOPE**

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.

**REFERENCES**

**WEBSITES:**

* [www.opengl.org](http://www.opengl.org)
* [www.sgi.com](http://www.sgi.com)
* [www.glprogramming.com](http://www.glprogramming.com)
* [www.openglbook.com](http://www.openglbook.com)

**BOOKS:**

* Edward Angel: Interactive Computer Graphics- A Top Down Approach with opnGL, 5th Edition, Addison-Wesley, 2008
* F.S.Hill, jrL: Computer graphics using openGL, 2nd edition, Pearson education 2001
* OpenGL Insights
* OpenGL Data Visualization Cookbook