CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO PROJECT

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.

Everyone likes the fountains, so presenting the OpenGL Projects using c++ on it. In the perspective of computer graphics creating the fountain is a uphill task. There is need of many items in the project design either it's holding the water, the flow of water and the water reservoir. Rendering these is tough task to make fountain look real. Most important point is to take care of rise of water upward, its angle of falling in parabolic fashion

This Computer graphics project on fountains has used the concept of OOP (Object Oriented Programming) principle in some aspect. It is C++ project rather a normal c project. The use of Class and the object in only used for the fountain's water fallings, while rest was under normal functions. The OOP principle is basically used in the water drops, steps of fountain, speed of fountain etc. A fountain has several steps, each with its own height.

1.2 OPEN GL (Open Graphics Library):

OpenGL is a premier environment for developing portable, interactive 2D and 3D graphics applications. Since it introduced in 1992, OpenGL has become the industry's most widely used and supported 2D and 3D graphics applications programming interface(API), bringing the thousands of applications to a wide variety of computer platforms.

OpenGL fosters innovation and speeds application development by incorporating a broad set of rendering, texture mapping, special effects, and other powerful visualization functions.

Developers can leverage the power of OpenGL across all popular desktop and workstation platforms, ensuring wide application deployment.

OpenGL runs on every major operating system including Mac OS, OS/2, UNIX, Windows. OpenGL is callable from Ada, C, C++, Fortran, Python and Java and offers complete independence from network protocols and topologies.

The project Dancing Doll will be designed to access OpenGL directly through functions in 3 libraries namely: GL, GLU, GLUT.

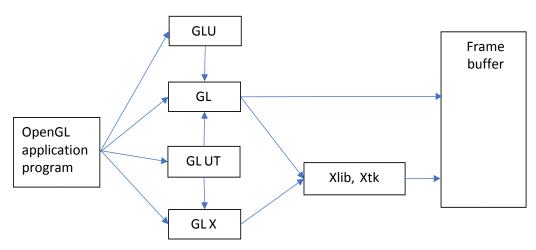


Fig 1.1. OpenGL Library tools

1.3 PROBLEM STATEMENT

The problem is mainly concerned with modeling a fountain and highlighting some of its unique features and qualities. We have limited ourselves to visualizing and not generating any level flow or user interaction as in a game. The fountain is subjected to many OpenGL transformations to bring out certain effects like changing the colors and changing the views like top view ,front view .

OpenGL is a software interface to graphics hardware. This interface consists of distinct commands, which we use to specify the objects and operations needed to produce interactive applications. It can be used to implement several visual effects. Hence this project has become more colorful because of the use of the versatile, robust and useful library called OpenGL.

1.4 OBJECTIVES OF THE PROJECT

- View Defined in the Project Top, Front, Away, Near and Back.
- Random Color of water can be there in fountains.
- Both Keyboard and Mouse Control for Users is added to this Project.
- There is need of many items in the project design either it's holding the water, the flow of water and the water reservoir.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 HARDWARE REQUIREMENTS

The standard output device is assumed to be a Color Monitor. It is quite essential for any graphics package to have this, as provision of color options to the user is must. The mouse is a main input device, has to be functional i.e. used to give input to the project. A keyboard is used for controlling and inputting the data in the form of characters, numbers i.e. to change the user views. Apart from these requirements the system should possess the minimum hardware requirements as follows:

- Hard disk- Sufficient hard disk space and primary memory available for proper working of the packages to execute the program.
- **Processor** Pentium 3 or higher processor.
- **RAM** 96 MB or more.
- 120 MB graphic card for better performance.
- Minimum 1.5 GHz of frequency.

2.2 SOFTWARE REQUIREMENTS

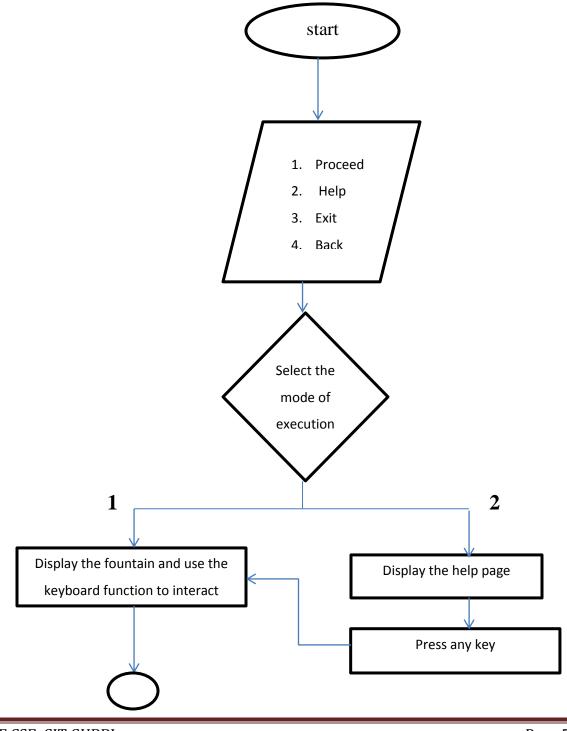
Minimum software specification as follows:

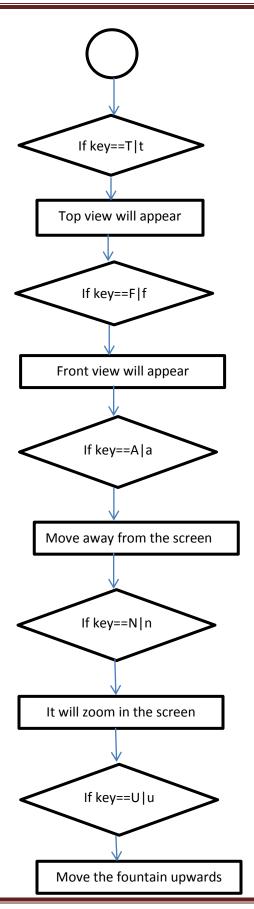
- Operating system- Windows XP
- **Programming language** C/C++ with OpenGL.
- **IDE-** Microsoft visual C++ 6.0

CHAPTER 3

SYSTEM DESIGN

3.1 ARCHITECTURE OF THE SYSTEM





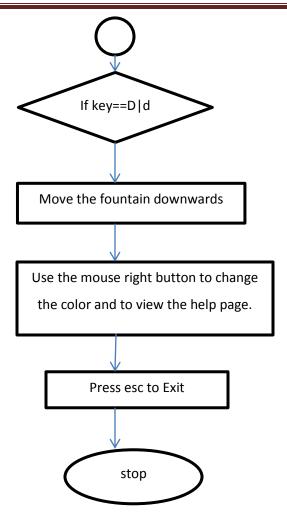


Fig 3.1overview of Flowing Fountain

The architecture of the Flowing Fountain is shown above. The user can interact with the flowing fountain using keyboard and mouse interfaces. The keyboard keys are-

T or t: The user can view the top view of the fountain.

F or f: The user can view the front view of the fountain.

A or a: The fountain moves away from the user.

N or n: The fountain moves near the user.

D or d: The fountain moves downward.

U or u: The fountain moves upwards.

Menus like color, help are attached to mouse Right button. Using color submenu user can change the color of the water and with the help submenu user can find help to interact with fountain.

CHAPTER 4

IMPLEMENTATION

4.1 HEADER FILES USED

#include <GL/glut.h>: The OpenGL utility Toolkit (GLUT) is a library of utilities for OpenGL programs, which primarily performs the system-level I/O with the host operating system. Functions performed include window definition, window control, and monitoring of keyboard and mouse input. Routines for drawing number of primitives (both solid and wireframe mode) are also provided, including cubes, spheres, and teapot.

#include<stdlib.h>:It is the header of the general purpose standard library of c programming language which includes function involving memory allocation, process control, conversions and others. It also includes four variables types, several macros and various functions for performing general functions.

#include<math.h>:. It supports all the mathematical related functions.

#include<**stdio.h>:** It provides many standard library functions for input and output. In program, printf () is a standard output function, and scanf () is standard input function, these are not defined by the programmer. These functions are already defined inside some language libraries, to use these kinds of predefined functions; we have to include this header file.

#include<**string.h**>: It defines one variable type,one macro and various functions for manipulating arrays of characters.

#include<iostream.h>: Like stdio.h iostream provides basic input and output services for c++ programs.

4.2 DATA TYPES USED

int: It contains whole numbers that can have both positive and negative values but no decimal values.

GLint: Signed 2's complement binary integer.

Unsigned int: unsigned types only store the positive value and thus can be quite larger than signed values.

GLfloat: An IEEE-754 floating-point value.

Struct: It is a user-defined data type that can store multiple related items. It contains multiple data types related to a single entity.

4.3 BUILT-IN FUNCTIONS

4.3.1 Void glBegin (glEnum mode);

Initiates a new primitive of type mode and starts the collection of vertices. Values of mode include GL_POINTS, GL_LINES and GL_POLYGON.

4.3.2 Void glEnd ();

Terminates a list of vertices.

4.3.3 Void glColor3f [I f d] (TYPE r, TYPE g, TYPE b);

Sets the present RGB colors. Valid types are int (i), float (f) and double (d). The maximum and minimum values of the floating-point types are 1.0 and 0.0, respectively.

4.3.4 Void glClearColor (GLclampf r, GLclampf g, GLclampf b, GLclampf a);

Sets the present RGBA clear colors used when clearing the color buffer. Variables of GLclampf are floating-point numbers between 0.0 and 1.0.

4.3.5 int glutCreateWindow (char *title);

Creates a window on the display. The string title can be used to label the window. The return value provides a reference to the window that can be used where there are multiple windows.

4.3.6 Void glutInitWindowSize (int width, int height);

Specifies the initial height and width of the window in pixels.

4.3.7 Void glutInitWindowPosition (int x, int y);

Specifies the initial position of the top-left corner of the window in pixels.

4.3.8 Void glutInitDisplayMode (unsigned int mode);

Request a display with the properties in mode. The value of mode is determined by the logical OR of operation including the colors model (GLUT_RGB, GLUT_INDEX) and buffering (GLUT_SINGLE, GLUT_DOUBLE);

4.3.9 Void glFlush ();

Forces any buffered any OpenGL commands to execute.

4.3.10 Void glutInit (int argc, char **argv);

Initializes GLUT. The arguments from main are passed in and can be used by the application.

4.3.11 void glutMainLoop();

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

4.3.12 void glutDisplayFunc (void (*func) (void));

Registers the display function func that is executed when the window needs to be redrawn.

4.3.13 gluOrtho2D (GLdouble left, GLdouble right, GLdouble bottom, GLdouble top);

Defines a two-dimensional viewing rectangle in the plane Z=0;

4.3.14 void glutBitmapCharacter (void *font, int char);

Renders the character with ASCII code char at the current raster position using the raster font given by Fonts include GLUT_BITMAP_TIMES_ROMAN_10 and GLUT_BITMAP_TIMES_ROMAN_8_Y_13. The raster position is incremented by the width of the character.

4.3.15 Void glClear (GL_COLOR_BUFFER_BIT);

To make the screen solid and white.

4.3.16 Void MouseFunc (myMouse);

It is used for the implementation of mouse interface. Passing the control to void myMouse (int button, int state, int x, int y);

4.3.17 Void KeyboardFunc (key);

It is used for the implementation of keyboard interface. Passing control to void key (unsigned char key, int x, int y);

4.3.18 void translate [fd] (TYPE x, TYPE y, TYPE z);

Alters the current matrix by displacement of (x, y, z). Type is either GL float or GL double

4.3.19 void glPushMatrix ();

Void glPopMatrix ();

Pushes to and pops from the matrix stack corresponding to current matrix mode.

4.3.20 void glLoadMatrix [fd](TYPE *m);

Loads the 16 element array of TYPE GLfloat or GLdouble as a current matrix.

4.4 USER DEFINED FUNCTION

4.4.1 DrawTextXY()

Syntax:

void DrawTextXY(double x, double y, double z, double scale, char *s)

Explanation:

This function is used to write the text on the output screen. This function takes the arguments such as co-ordinates of x-axis,y-axis,z-axis and the text which is to be print. For example DrawTextXY(-2.5,1.2,0.0,0.001,"Harshitha k") prints the name at the corresponding positions of x,y and z-axis.

4.4.2 cover()

Syntax:

void cover()

Explanation:

This function is used to design the main page of the program. The main page of the project displays the name of the project, the name and usn's of the students handled in this project, name of the lecturer who guide the students and institution name. In this function, the background color is set using glClearColor() and color for the text using glColor3f().

4.4.3 menu1()

Syntax:

void menu1()

Explanation:

This function is used to design the menu page .In this the user can select the action which they need to perform. We created 4 menus are given proceed, help, back, exit.In this we user select the proceed menu then it will go to the fountain, if user select the help menu then it will go to the help page, back means it will go to the previous page and lastly the exit menu then it will exit from the program.

4.4.4 menu2()

Syntax:

void menu2()

Explanation:

This Function is used to design the next page that is help page of the project. This page will help the user to handle the fountain with keyboard interactions. How the fountain will work? ,for which key is assigned and other information. In this function the background color is set using glClearColor() and color for the text using glColor3f().

4.4.5 Dis()

Syntax:

void Dis()

Explanation:

This function is used to display the menu page. If the f is equal to 0 means then it will call the function menu1, if f is equal to 1 means then it will call the function menu2, if f is equal to 2 means then it wil display the main page and lastly it wil call the function of display for displaying.

4.4.6 Display()

Syntax:

void Display(void)

Explanation:

This function is used to display the founatin. Inside this we are specified the key values to which it should be change the values of axis when we pressed that key with respect to the x-axis, y-axis and z-axis.

4.4.7 DrawFountain()

Syntax:

void DrawFountain(void)

Explanation:

This function is used to give the color for the fountain drops. In this we are giving the color pink, blue, white and also many more colors for the drops.

4.4.8 CreateList()

Syntax:

void CreateList(void)

Explanation:

This function is used to draw the base of the fountain. In this the height, width and also the radius of that base is defined. In this the fountain base is visible like a hexagon.

4.4.9 InitFounatin()

Syntax:

void InitFountain(void)

Explanation:

This function is used to draw the founatin drops. In this the water flow level and angle of the drops in which it is flowing. In this the raypersteps and also the timespeed of the drops is calculated. And also it will calculate how much drops is need to flow in one ray and also how much rays is need to be there in steps.

4.4.10 randcolor()

Syntax:

void randcolor():

Explanation:

This function is used to give the different colors for the drops in the fountain. When we click the random color in the color menu it will gives the different color for the drop. In this for each click also it will gives the random color differently.

4.4.11 colours()

Syntax:

void colours(int id)

Explanation:

This function is takes the value that specified in the id. If the flag is equal to id then if flag is equal to 1 then it calls the randcolor function. Then the random colors will display.

4.4.12 flow()

Syntax:

void flow(int id)

Explanation:

This function is used to flow the founatin drops. In this it will take the value from the id and flow the drops which is equal to raypersteps.

4.4.13 NormalKey()

Syntax:

void NormalKey(GLubyte key, GLint x, GLint y)

Explanation:

This function is used to gives the actions of the keyword which is used to gives the top view, front view, back, near, away and also the program will exit when we press escape.

CHAPTER 5

SNAPSHOTS

5.1 Main screen:

It is the first page appears when the program runs. It tells who are handled this project under whom guidance and from which institute.

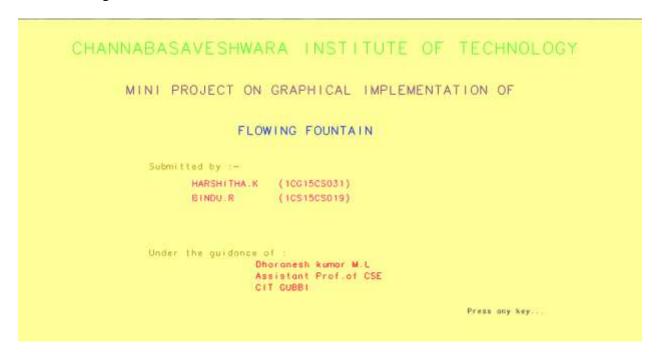


Fig 5.1 Main Screen

5.2 Menu Screen:

This is the page where the user can decide to proceed further or not . This page will display the menu screen to select the option from the keyboard interactions.

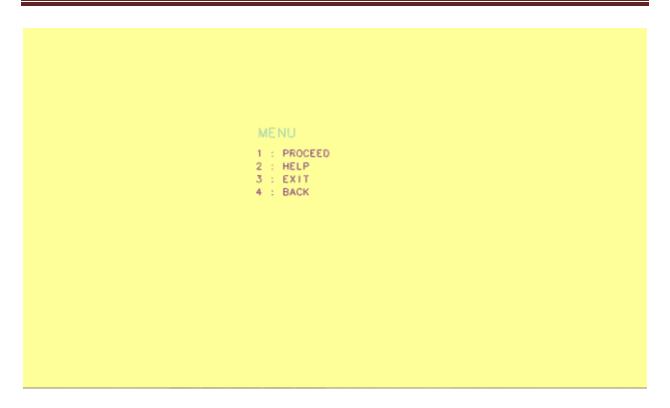


Fig 5.2 Menu Screen

5.3 The Flowing Fountain (Initially):

Initially the founatin looks this when we pressed the proceed button in menu.

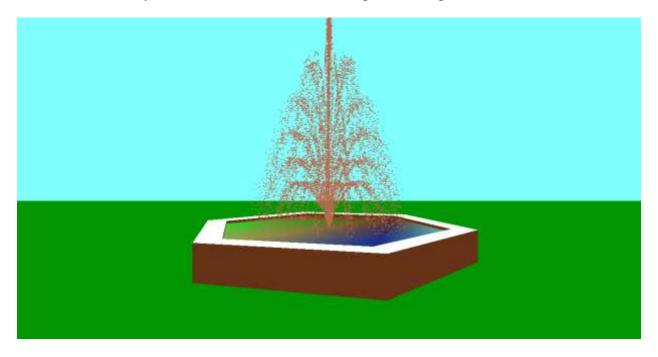


Fig 5.3 Flowing Fountain(Initially)

5.4 Help Page:

This is the page which helps the user to access the project applications to handle the fountain in the project with the keyboard interactions. How the fountain working here?, for which is the key assigned and other information.

```
Keyboard Controls:

Move Near -> N
Move Away -> A
Move Up -> U
Move Down -> D

Top View -> T
Front View -> F

Open HELP -> H
Open MENU -> B
```

Fig 5.4 Help Page

5.5 Top View:

When user press the key T or t keyword in keyboard the top view of the fountain will appear like this.

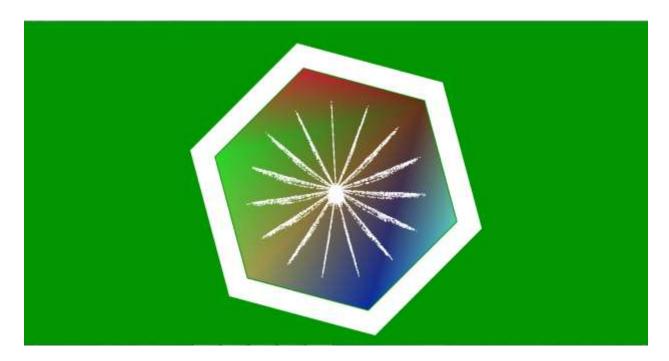


Fig 5.5 Top View

5.6 Result Of Pressing The Keyword A or a(away):

In this below screen we are displaying the fountain when we pressed the keyword A or a. It will move the fountain away the screen.

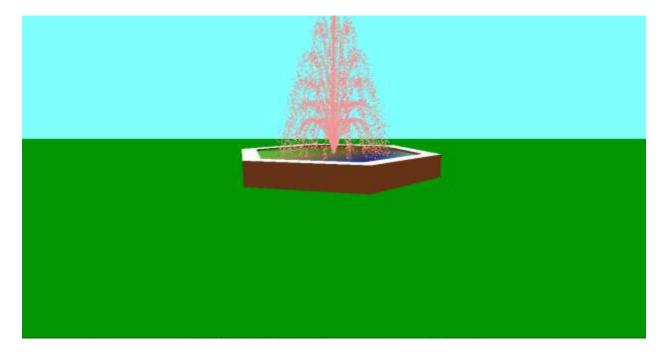


Fig 5.6 Zoom Out

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

This project has met its objectives to produce a Fountain. It has a user friendly interface. It is quite interactive with a good keyboard interface. The development of this project was very helpful to develop my programming skills and to know C-programming better. It also helped me to implement concepts in my project such as translation, drawing objects on the window screen using points, lines, polygons use of lighting effects, material properties, toggling effects on the objects.

This project is interesting in designing and developing and a good learning experience. It helps to learn about computer graphics, design of graphical interfaces, interface to the user, user interaction handling and screen management. Structure is defined for each of the directions and a class is created for dropping the water via fountain. The class declaration contains all data members and member functions.

This project may be useful to follow the architecture of Fountain and its further development in future with more options.. Even though demo designed is enriched with many options, it is a two dimensional demo, in future it can be redesigned with 3D animation and sound effects.

REFERENCES

- [1] Angel. E, OpenGL, A Primer, Third Edition, Addison-Wesley, Reading, MA, 2008.
- [2] Edward Angel, *Interactive Computer Graphics: A Top-Down Approach Using OpenGL, Fifth* Edition, Addison Wesley, 2008.
- [3] Francis S. Hill, Computer Graphics Using Open GL, Second Edition, Prentice Hall, 2000.
- [4] Hearn, D., and M.P. Baker, *Computer Graphics*, Second Edition, Prentice-Hall, Englewood Cliffs, NJ, 1994.