

CHAPTER 1

INTRODUCTION

COMPUTER GRAPHICS

Computer graphics is sub-field of computer science and is concerned with digitally synthesizing and manipulating visual content. Although the term refers to three-dimensional computer graphics, it also encompasses two dimensional graphics and image processing. Computer graphics is often differentiated from the field of visualization, although the two have some similarities.

Computer graphics started with the display of data on hardcopy plotters and cathode ray tube (CRT) screens soon after the introduction of computers. It has grown to include the creation, storage and manipulation of models and images of objects. These models come from a diverse and expanding set of fields, and includes physical, mathematical, engineering, architecture and even conceptual structures.

Graphics interfaces have replaced textual interfaces as the standard means for user-computer interaction. Graphics has also become a key technology for communicating ideas, data and trends in most areas of commerce, science, engineering and education. Much of the task of creating effective graphic communication lies in modelling the objects whose image we want to produce.

The aim of this project is to create a JUMPING CAR. Here the user interface is made with clear attractive pictures using graphics. In this scenario the car is made to move over the road by jumping over the obstacles by pressing 's' key. So that if the user fails to press the specified key then, it leads the car to crash to the obstacle. After this a dialogue appears as "GAME OVER" in the user display. Then again the game starts from beginning. Here we have used 'void chardraw()' function, so that the game accepts the specified user input to run the game. This function accepts three parameters like GLubyte key, GLint x, GLint y. Here x and y are the axis of the user display.

OpenGL

OpenGL is the abbreviation for OpenGL Graphics Library. It is a software interface for graphics hardware. This interface consists of several hundred functions that allow you, a

graphics programmer, to specify the objects and operations needed to produce high quality actually simple variations of each other, so in reality there are about 120 substantially different functions. The main purpose of OpenGL is to render two-dimensional and three-dimensional objects into frame buffer. These objects are defined as sequence of vertices or pixels. OpenGL performs several process on this data to convert it to pixel to form the final desired image in the frame buffer.

APPLICATIONS OF OPENGL

1. Display information:
 - Medical imaging poses intersecting and important data- analysis.
 - Modern imaging technology such as computed tomography(CT), magnetic resonance imaging(MRI).
2. Design:
 - Design is an iterative process.
 - Professions such as engineering and architecture are concern with design.
3. Simulation and animation:
 - Graphical flight simulator has proved to increase safety and to reduce training expenses.
 - The simulator can be used for designing the robot.
4. User interface:
 - Our interaction with computer has become dominated by visual paradigm.
 - Graphical user interface used on most workstations.

CHAPTER 2

LITERATURE SURVEY

A basic library of functions is provided in OpenGL for specifying primitives, attributes, geometric transformations, viewing transformations, and many other operations.

OpenGL is designed to be hardware independent, so many operations such as input and output routines are not included in the basic library.

BASIC OPENGL SYNTAX

Function names in the OpenGL basic library are prefixed with `gl` and each component word within the function name has its first letter capitalized, for example:

`glBegin`, `glClear`, `glColor3f`, `glClearColor`, `glCopyPixel`, `glPolygonMode`.

`glColor3f(float, float, and float)`: - This function will set the current drawing color

`glClear()`: - Takes a single argument that is the bitwise OR of several values indicating which buffer is to be cleared?

`glClearColor()`: - Specifies the red, green, blue, and alpha values used by `glClear` to clear the buffers.

Certain functions require that one of their arguments be assigned a symbolic constant specifying, for instance, a value for a parameter or a particular mode. All such constants begin with an uppercase letter `GL`.

For example: `GL_RGB`, `GL_POLYGON` etc

The OpenGL functions also specify data types. To indicate a specific data type, OpenGL uses built-in data types such as

`GLbyte`, `GLshort`, `GLint`, `GLfloat`, `GLdouble`, `GLboolean`

RELATED LIBRARIES

In addition to OpenGL basic library, there are a number of associated libraries for handling special operations. The OpenGL Utility (GLU) provides.

Routines for setting up viewing and projection matrices, describing complex objects with line and polygon approximations, displaying quadrics and B-splines using linear approximations, processing the surface rendering operations and complex tasks.

HEADER FILES

In all our graphics programs, we will need to include the header file for the OpenGL core library. For most applications we will also need GLU, and on many systems we will need to include the header file for window system. The header file must be listed before the OpenGL and GLU header files. So the source files in this case would begin with

```
#include<windows.h>,#include<GL/gl.h>,#include<GL/glu.h>
```

However, if we use GLUT to handle the window managing operations, we do not need to include gl.h and glu.h because GLUT ensures that these will be included correctly. Thus we can replace the header files for OpenGL and GLU with

```
#include<GL/glut.h>
```

GLUT initialization functions could also process any command line arguments, we perform the GLUT initialization with statements.

```
glutInit(&argc,argv);
```

Next, we can state that a display window is to be created on the screen with a given caption for the title bar.

```
glutCreateWindow("Flag Hosting");
```

Then we need to specify what the display window is to contain. For this we create a picture using OpenGL functions and pass picture definition to the GLUT routine glutDisplayFunc.

```
glutDisplayFunc(line segment);
```

But the display window is not yet on the screen. We need one more GLUT function to complete the window processing operations, after execution of the following statement, all the display windows that we created, including their content, are now activated.

```
glutMainLoop();
```

This function must be last one in our program. We use `glutInitWindowPosition` function to give an initial location for the upper left corner of the display window.

```
glutInitWindowPosition();
```

Similarly, the `glutInitWindowSize` function is used to set the initial pixel width and height

```
glutInitWindowSize();
```

We can also set the number of other options for the display window, such as buffering and a choice of color of modes, with the `glutInitDisplayMode` function.

```
glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
```

The values of the constants passed to these functions are combined using logical operations.

CHAPTER 3

OBJECTIVE

1. The aim of the project is to demonstrate the jumping car with views.
2. As linux does not provide graphics editor.
3. It should be easy to understand,user interactive interface.
4. Creation of primitives,i.e.polygons.
- 5.Providing human interaction through keyboard.

ADVANTAGES

- 1.Users can feel the gaming environment set up in user screen,and can enjoy the the game by jumping the car over the obstacles by some pressing user specified key.
2. Attracts the users to have such a game in different levels.So that they can have a lot of fun.
- 3.Engineers who build games can design their basic level of games by looking thisproject.
4. It will be helpful for them in building similar games upto advanced levels. So this project will be benefited by Engineers and users

CHAPTER 4

DESIGN

4.1 INITIALIZATION

- Initialize the display mode that is double buffer and RGB color system.
- Initialize window position and window size.
- Initialize and create the window to display the output.
- Initialize to interact with windows.

4.2 DISPLAY

- The operations are performed by OpenGL.
- The program is to display a tree using recursive function.
- We use the parameters scaling and angle.
- Scaling:Scaling is an affine non rigid body transformation by which we can make an object bigger or smaller.
- Angle:Angle specifies the start position of tree.

CHAPTER 5

SYSTEM REQUIRMENTS

The minimum software and hardware requirments which are used for project are as follows.

MINIMUM HARDWARE REQUIREMENTS

- Processor : 1.6 GHZ
- RAM :1GB(32-bit),2GB(64-bit)
- Hard Disk : 3GB (approx) , 5400 RPM hard disk drive
- Display : DirectX 9 capable video card running at 1024 x768 or higher display

MINIMUM SOFTWARE REQUIREMENTS

- Operating System : Windows 7
- Language : OpenGL using C/C++
- Compiler : Microsoft Visual studio 2010 pro

CHAPTER 6

IMPLEMENTATION

OVERVIEW

This Project is on “JUMPING CAR” computer graphics using OpenGL functions. It is an interactive program where the User can view the required display by making use of the input devices like keyboard. The jumping movement of car can be done with help of the keyboard.

USER INTERFACE

1. User can select the view of the car by keyboard.
2. Movement of car is controlled by S.
3. Different types of objects are used.
4. Car jumps when we press the key button S.
5. Score of the game stops after the end of game.
6. When car hits the object game ends.

MODELS

Void DisplayFunc():-Contains all the functions that are used to create and movement.

Void ReshapeFunc():-Used to change the shape of the object.

Void IdleFunc():-Idle callback for OpenGL.

Void SpecialFunc():-Used to make modification.

Void KeyboardFunc():-Used to make movement and rotation.

Void init(args):-Init function for OpenGL

CONCLUSION

Our project is only a humble venture to satisfy the needs in JUMPING CAR prescribed under the Computer Graphics and Visualization lab mini project. Several users friendly coding has also adopted. This project shall prove to be a powerful package in satisfying all the requirements of the user. We found designing and developing this jumping car as very interesting and learning experience. It helped us to learn about computer graphics, design of Graphical User Interfaces, interface to the user, interaction handling and screen management.

REFERENCES

1. Donald Hearn & Pauline Baker: Computer Graphics-OpenGL Version,3rd Edition, Pearson Education,2011.
2. Edward Angel's Interactive Computer Graphics Pearson Education 5th Edition.
3. Interactive computer graphics- A Top Down approach using OpenGL- by Edward angle.