CHAPTER 1

INTRODUCTION

1.1 Computer Graphics

Computer graphics is concerned with all aspects of producing images using a computer. It concerns with the pictorial synthesis of real or imaginary objects from their computer-based models. In other words, it may be a collection, combination and representation of imaginary objects from their computer-based models. It is one of the most exciting and rapidly growing computer fields. It is also an effective media for communication between humans and the computer. Graphical interfaces have replaced textual interfaces as the standard means of computer interaction. Years of research and development were made to achieve the goals in the field of computer graphics. In 1950 the first computer driven display was used to generate only simple pictures. This display made use of a cathode ray tube similar to the one used in television sets. Although the term often refers to the study of three-dimensional computer graphics, it also encompasses two-dimensional graphics and image processing. Computer graphics studies the manipulation of visual and geometric information using computational techniques. It focuses on the mathematical and computational foundations of image generation and processing rather than purely aesthetic issues. Computer graphics is often differentiated from the field of visualization, although the two fields have many similarities.[1]

Computer graphics are pictures and films created using computers. Usually, the term refers to computer-generated image data created with help from specialized graphical hardware and software. It is a vast and recent area in computer science. The phrase was coined in 1960, by computer graphics researchers Verne Hudson and William Fetter of Boeing. It is often abbreviated as CG, though sometimes erroneously referred to as computer-generated imagery (CGI).

Some topics in computer graphics include user interface design, sprite graphics, vector graphics, 3D modelling, shaders, GPU design, implicit surface visualization with ray tracing, and computer vision, among others. The overall methodology depends heavily on the underlying sciences of geometry, optics, and physics.

Our project is based on creating awareness among the people about the proper disposal of waste. The scenario consists of dustbins, car, trees, house and a person. These are drawn by using some GL functions like GL_POLYGON , GL_LINES , GL_POINTS etc . The built - in function glutPostRedisplay( ) is used to redraw the element on the screen when they are translated or rotated.[2]

1.2 Open GL

Open Graphics Library (OpenGL) is cross-language, cross-platform application programming interface (API) for rendering 2D and 3D vector graphics. The API is typically used to interact with a graphics processing unit (GPU), to achieve hardware accelerated rendering.

Silicon Graphics Inc., (SGI) began developing OpenGL in 1991 and released it on June 30, 1992, applications use it extensively in the fields of computer-aided design (CAD), virtual reality, scientific visualization, information visualization, flight simulation, and video games.

Open GL emerged from Silicon Graphics Inc, in 1992, and has become a widely adopted graphics programming interface (API). It provides the actual drawing tools through a collection of functions that are called within an application. It is easy to install and learn, and its longevity as a standard API is being nurtured and overseen by the OpenGL Architecture Review Board (ARB), an industry consortium responsible for guiding its evolution.[3]

One aspect of OpenGL that suits it so well for use in computer graphics course is its device independence or portability. A student can develop and run a program on any available computer. OpenGL offers rich and highly usable API for 2D graphics and image manipulation, but its real power emerges with 3D graphics.

1.3 Swatch India

This scenario helps us to realize about the common mistakes which we commit in our day to day life and the bad effects caused by these mistakes. So , it helps us to have a good environment surrounding us by giving an awareness about the harmful effects caused by these wrong doings.

CHAPTER 2

REQUIREMENT SPECIFICATIONS

Requirement specification is focused specially on functioning of the system, functions to be carried out and performance levels to be obtained and corresponding interfaces to be established. This report gives the description of the roles of users, the functional overviews of the project, input and output characteristics and also the hardware and software for the project.

2.1 Functional Requirements

The graphics has been written in C language and many simple user defined functions are used. The project requires the access of OpenGL utility toolkit with the use of the header file "GL/glut.h" . This header file, in addition to the usual header files is needed for the working of the project. For running the program, any basic C running compatible version of Code Blocks or Linux based platform is sufficient.

2.2 Non-Functional Requirements

The software should not accept wrong inputs and produce outputs. It should use memory as less as possible, dynamic memory allocation is preferable to accomplish this task.

2.3 Hardware Requirements

The minimum/recommended hardware configuration required for developing the proposed software is given below:

- INTEL dual core and above compatible systems.
- 2GB RAM.
- Approximately 10MB free space in the hard disk.

2.4 Software Requirements

Ubuntu Operating System is being used and the entire project code is developed using geditor and gcc compiler. OpenGL Functions and Primitives are used to draw various objects. The functions being used are mentioned later in this report.

CHAPTER 3

SYSTEM DESIGN

3.1 Flowchart

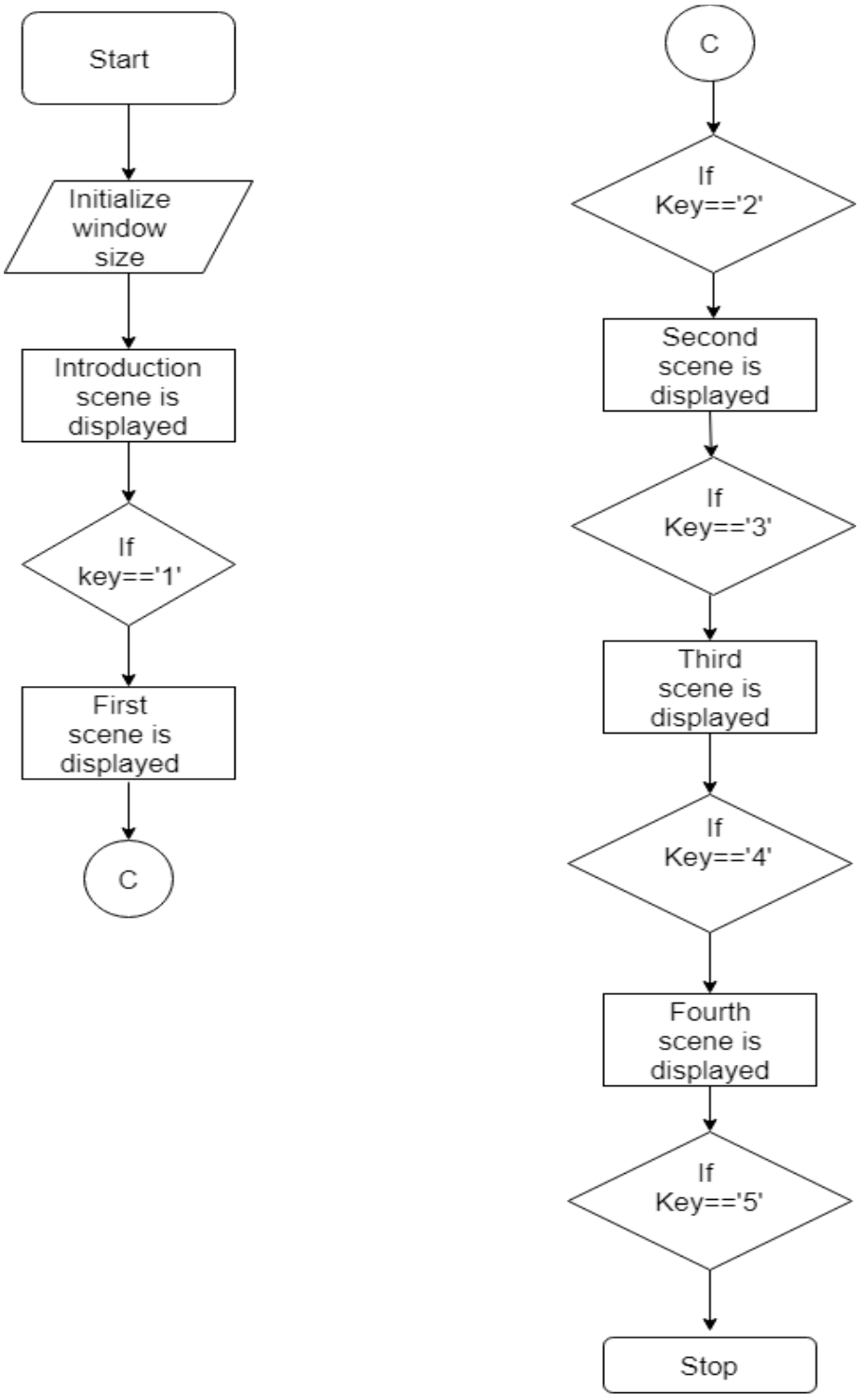

Figure 3.1 : Flowchart of the project

A Flowchart is a graphical representation of an algorithm, workflow or process. The symbols represent the steps and the connections represents the flow of instructions or steps. This representation of solution to a problem statement helps to design, document it efficiently.

In the following Flowchart Figure 3.1, rectangle indicates the pages of the project, Oval indicates switch statement, rhombus indicates decision and the round-edged rectangle indicates the terminal.

When we run the program, introduction page is generated. To continue further we should press '1' and we reach the first page. If we press '2' we will move to the second page. If we press '3' we will move to the third page. If we press '4' we will move to the fourth page. If we press '5' we will move to the end of the project.

CHAPTER 4

IMPLEMENTATION

4.1 Implementation of OpenGL Built-In functions:

- **glutCreateWindow(char *name) –** Creates the 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 when there are multiple windows.

- **glutInitWindowSize(int width, int height) –** It specifies the initial height and width of the window in pixel.

- **glutInitWindowPosition(int x, int y) –** It specifies the initial position of the top left corner of the window.

- **glColor3 [b i f d ub ui] (TYPE r, TYPE g, TYPE b) –** sets the present RGB (or RGBA) colors. Valid types are byte(b), int(i), float(f), double(d), unsigned byte(ub), unsigned int(ui). The maximum and minimum values of the floating-points types are 1.0 and 0.0 respectively.

- **glutInit(&argc,argv) –** It takes arguments.

- **glutKeyBoardFunc(void *(unsigned char key, int x, int y)func) -** It registers the keyboard callback function, takes the ASCII code of the key pressed and the position of the mouse.

- **glutMainLoop() –** It causes the program to enter tan event-processing loop, It should be the last statement in the main.

- **glutDisplayFunc(void(*func)(void)) –** Registers the display function that is executed when the window needs to be redrawn.

- **glClear(GLbitfieldmask) –** Clears the specified buffers to their current clearing values. The mask argument is a bitwise logical OR combination of values i.e. GL_COLOR_BUFFER_BIT,GL_DEPTH_BUFFER_BIT.
- **glLoadIdentity() –** Replaces the current matrix with the identity matrix.

- **glClearColor(GLfloatR,GLfloatG,GLfloatB,GLfloatalpha)-** This clears the background color to current clearing values.

- **voidgluOrtho2D(GLfloatleft,GLfloatright,GLfloattop,GLfloatbotom)-** Sets the ortho as specified values in the function argument.

- **glFlush() –** Forces all previously issued OpenGL commands to begin execution thus guarantying that they complete in finite time.

- **glVertex2f(GLfloat x,GLfloat y) –** specifies a vertex in 2D.

- **glRasterpos3f(TYPE * coordinates) –** Specifies the raster position. Parameters are same as glVertex.

- **glutBitmapCharacter(void *font int char) –** Renders the character with ASCII code char at the current raster position using the raster font given in the font.

- **glBegin(GL_LINE_LOOP) –** Draws a hollow diagram. Takes the end points from the glertex function.

- **glMatrixMode(GLenum mode) -** specifies which matrix will be affected by subsequent transformations. Mode can be GL_MODELVIEW,GL_PROJECTION.

- **glPushMatrix() -** It pushes the matrix stack corresponding to the current matrix mode onto the screen.

- **glPopMatrix() –** It pops the matrix stack corresponding to the current matrix mode onto the screen.
- **glBegin(GLenum mode) –** It initiates a new primitive of type mode and starts the collection of vertices. Values of mode includes GL_POINTS, GL_POLYGON, GL_TRIANGLE, GL_POINTS.
- **void glutPostRedisplay() –** Requests that the display callback be executed after the current callback returns.
- **void glTranslate[fd](TYPE x, TYPE y, TYPE z) –** Alters the current matrix by a displacement of(x, y, z).
- **void glutKeyboardFunc(void *f(char key, int width, int height) –** The callback function returns the ASCII code of the key pressed and the position of the mouse.
- **void glRotate[fd](TYPE angle, TYPE dx, TYPE dy, TYPE dz) –** Alters the current matrix by a rotation of angle degrees about the axis(dx, dy, dz).
- **void glEnable(GLenum feature) –** Enables an openGL feature. Features that can be enabled include GL_DEPTH_TEST, GL_LIGHTING, GL_TEXTURE_1D, GL_TEXTURE_2D, and GL_TEXTURE_3D.
- **void glDisable(GLenum feature)–** Disables an openGL feature.
- **void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far) –** Defines an orthographic viewing volume with all parameters measured from the centre of projection plane.
- **void gluPerspective(GLfloat fovy, GLfloat aspect ,GLfloat near, GLfloat far) –** Defines a perspective viewing volume with all parameters measured from the centre of projection plane
- **glEnd() –** Indicates the end of glBegin().

4.2 Functions used in "Swatch India" :

- **void display() –** This function is used to display various scenes present in the project. This function is called various times using different names but does the same task and that is to display.
- **void main(int agrc,char **agrv) –** The int main function is the main routine of the program.
- **void intro() –** This function is used to display the introduction screen.
- **void road() –** This function is used to draw the road on the screen. This function is called various times using different names but does the same task and that is to draw the road.
- **void hill() –** This function is used to draw the hills on the first scene.
- **void circle() –** This function is used to draw the circle on the screen. This function is called various times to do the same task.
- **void home() –** This function is used to draw the house on the screen. This function is called various times using different names but does the same task and that is to draw the house.
- **void sickman() –** This function is used to draw the sickman on the first scene.
- **void car() –** This function is used to draw the car on the screen. This function is called various times using different names but does the same task and that is to draw the car.
- **void dustbin() –** This function is used to draw the dustbin. This function is called various times using different names but does the same task and that is to draw the dustbin.
- **void man_in_car() –** This function is used to draw the man inside the car.
- **void waste() –** This function is used to draw the wastes of differnet shapes. This function is called various times using different names but does the same task and that is to draw the waste.
- **void pond() –** This function is used to draw the pond on the first scene.
- **void board() –** This function is used to draw the board near the pond.

- **void move_car() –** This function is used to translate the car from one point to another point.
- **void waste_with_man() –** This function is used to draw the man carrying a waste who dumps it into dustbin.
- **void move_waste_and_man() –** This function is used to provide movement for the man carrying a waste.
- **void man_throw_waste() –** This function is used to the draw the waste which will be thrown to the pond.
- **void bone() –** This function is used to draw the waste bone in the pond.
- **void tree() –** This function is used to draw the tree on the screen. This function is called various times to do the same task.
- **void cut_tree() –** This function is used to draw the half cut tree on the screen. This function is called various times to do the same task.
- **void bench() –** This function is used to draw the bench near the house on first scene.
- **void school() –** This function is used to draw the school in the second scene.
- **void write_content(float a,float b,char *aa,char font) –** This function is used to print the characters on any object. This function is called various times using different names but does the same task and that is to print character.
- **void plotpixels(GLint h,GLint k, GLint x,GLint y) –** This function is used to plot the pixels. This function helps us to draw circles.
- **void busstand2() –** This function is used to draw the busstand on the second screen.
- **void vehicle2() –** This function is used to draw the vehicle on the second screen.
- **void man_throw_waste2() –** This function is used to draw the man on the second screen.
- **void compound() –** This function is used to draw the compound for the school on the second screen.

CHAPTER 5

RESULTS

Following pages shown below are the graphical output obtained.

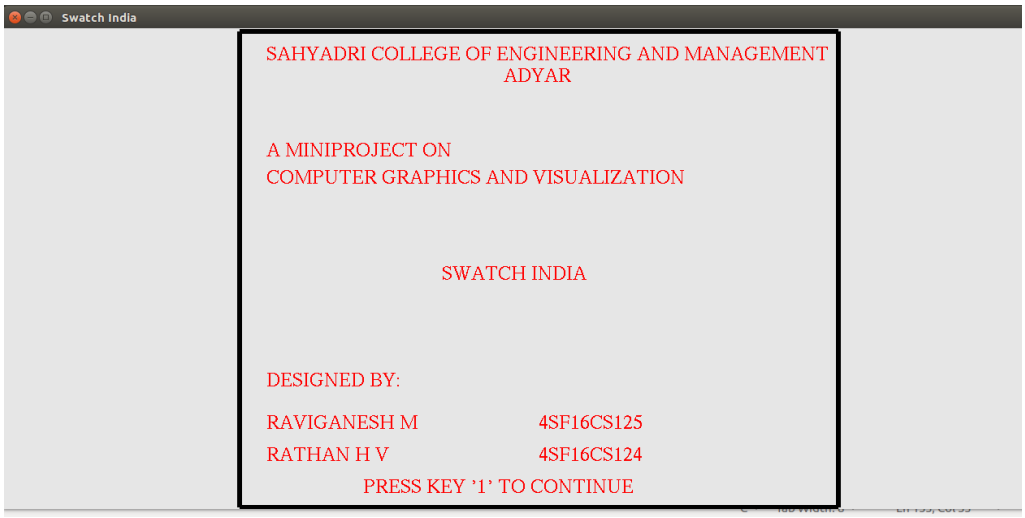

Figure 5.1 : Introduction Page

Figure 5.1 displays the project title and also the names of the all the members. Upon pressing the key '1' it transitions to the next scene.

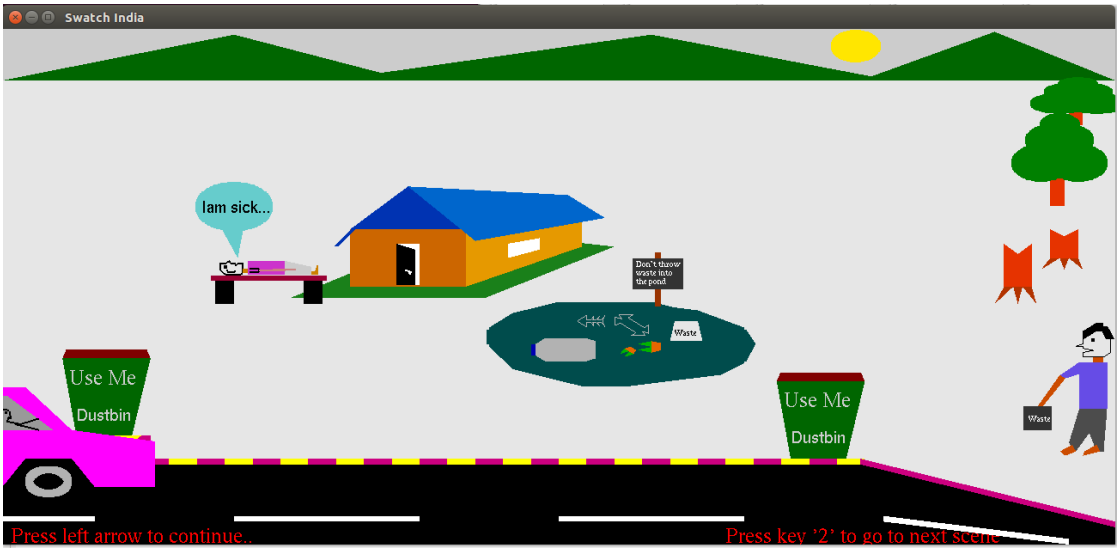

Figure 5.2 : First Scene

Figure 5.2 shows the beginning of the first scene where the pond is filled by the wastes.

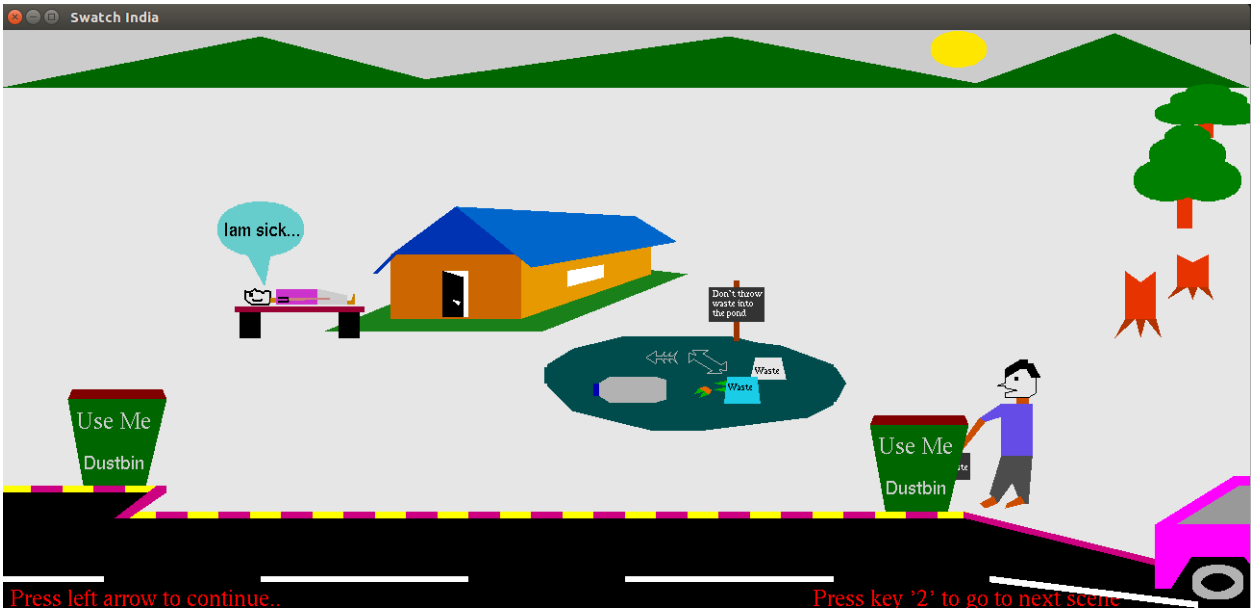

Figure 5.3:First scene with wastes thrown

Figure 5.3 shows that the person in the car throwing waste onto the pond. Upon pressing '2' it transitions to next scene.

Figure 5.4 : Second scene

Figure 5.4 shows the beginning of the second scene.

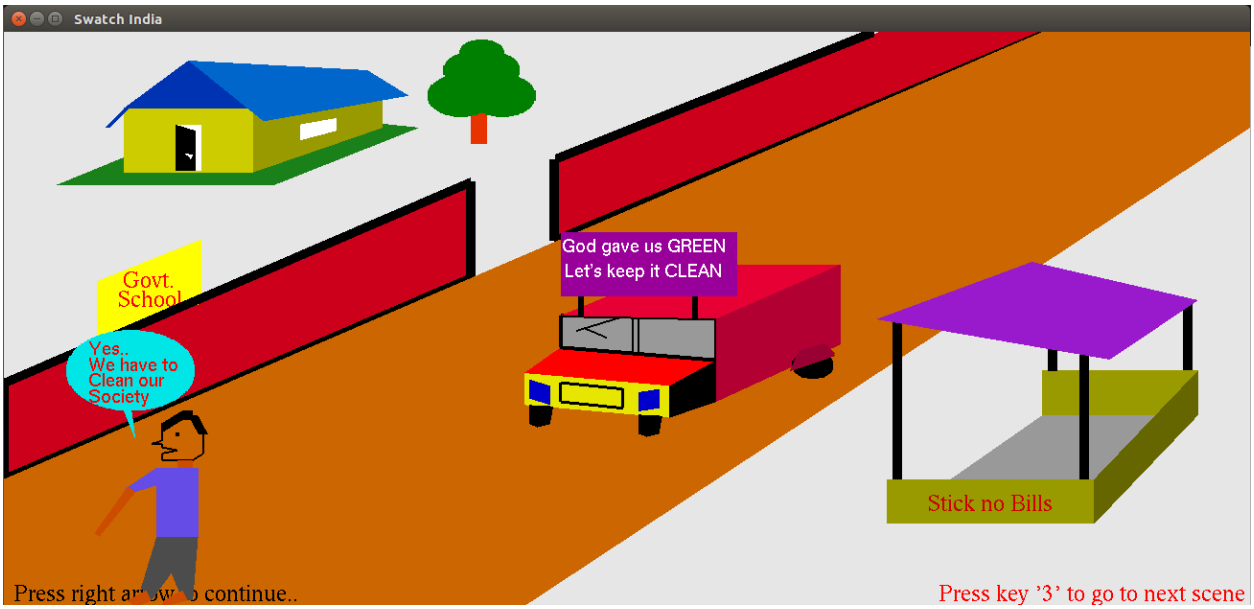

Figure 5.5 :Second scene with procession vehicle.

Figure 5.5 shows the movement of vehicle along the road and a person realizing his mistakes. Upon pressing '3' it transitions to next scene.

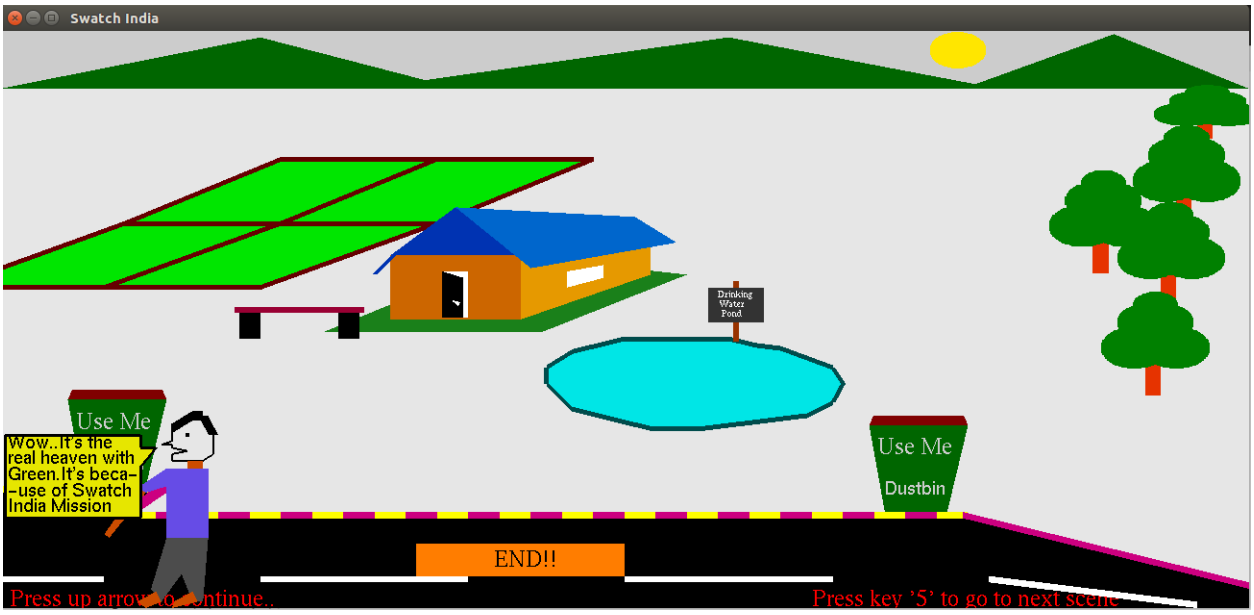

Figure 5.6 :Third scene

Figure 5.6 shows the scenario with unpolluted environment.

CHAPTER 6

CONCLUSION

The project "SWATCH INDIA" has helped in bringing awareness about the common mistakes committed by people against the nature using OpenGL and the various concepts, functions and methodologies required for the development of a graphics package. . The project illustrates the most basic concepts of OpenGL, how it can be interactive and exciting.

Computer Graphics plays a major role in today's world where visualization takes the upper hand as compared to text based interactions. It is also significantly proven that people remember the things which they see rather than read or listen. This is largely true as we can see user interfaces more attractive. Computer graphics is a powerful tool in this era of mass media.

The use of the keyboard driven interfaces, has helped to show the various features of our project. This reduces the complexity and improves the case with which any kind of user can run it. Project can be easily handled by anyone and changes can be made instantly because code used to run our programme is easily understandable. Various features of our project is displayed in such a way that, all the users can easily understand the concept. The project is made in such a way that it is interactive in all terms and exciting as well. We have given mouse interphase that has helped to exhibit its features and hence complexity of our project is greatly reduced.

BIBILOGRAPHY

[1] James D Foley, andries Van Dam, Steven K Feiner, John F Huges: ***Computer Graphics***, Pearson Education.

[2] https://en.wikipedia.org/wiki/Computer_graphics

[3] https://en.wikipedia.org/wiki/OpenGL