SURGICAL STRIKE

CHAPTER 1

INTRODUCTION

1.1 Introduction to Computer Graphics

Computer Graphics is concerned with all aspects of producing pictures or images using a computer. The field began humbly almost 50 years ago, with the display of a few lines on a cathode-ray tube (CRT); now, we can create images by computer that are indistinguishable from photographs of real objects. We routinely train pilots with simulated airplanes, generating graphical displays of a virtual environment in real time. Feature-length movies made entirely by computer have been successful, both critically and financially. Massive multiplayer games can involve tens of thousands of concurrent participants.

Perhaps the dominant characteristic of this new millennium is how computer and communication technologies have become dominant forces in our lives. Activities as wide -  
ranging as filmmaking, publishing, banking and education continue to undergo revolutionary changes as these technologies alter the ways in which we conduct our daily activities. The combination of computers, networks, and the complex human visual system, through computer graphics, has led to new ways of displaying information, seeing virtual worlds, and communicating with people and machines.

The Computer Graphics is one of the most effective and commonly used methods to communicate the processed information to the user. It displays the information in the form of graphics objects such as pictures, charts, graphs and diagram instead of simple text. In computer graphics, pictures or graphics objects are presented as a collection of discrete picture elements called pixels. The pixel is the smallest addressable screen element.

Computer graphics today is largely interactive: The user controls the contents structure, and appearance of objects and their displayed images by using input devices, such as a keyboard, mouse, or touch-sensitive panel on the screen. Computer graphics concerns with the pictorial synthesis of real or imaginary objects from their computer based models, where the related field of image processing treats the converse process, the analysis of scenes, or the Reconstruction of models of 2D or 3D objects from their pictures. The image processing can be classified in various divisions based on their functions and the way in which it is used. They are implemented in various sections. Some of their classifications are,

DEPT OF CS&E, SCEM, MANGALURU 1

SURGICAL STRIKE

• Image enhancement.

• Pattern detection and recognition.

• Scene analysis and computer vision.

The image enhancement deals with the improvement in the image quality by eliminating noise   
or by increasing image contrast. Pattern detection and recognition deals with the detection and   
clarification of standard patterns. And finding deviations from these patterns .The optical   
character recognition (OCR) technology is a practical example for pattern detection &   
recognition. Scene analysis deals with the recognition and reconstruction of 3D model of scene   
from several 2D images.

1.2 Introduction to OpenGL

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. OpenGL is easy to learn, and it possesses most of the characteristics of other popular computer graphics system.

OpenGL is 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. Similarly,   
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.

OpenGL provides us with fairly direct control over the fundamental operations of two- and   
three-dimensional graphics. This includes specifications of such parameters and transformation   
matrices, lighting equation coefficients and pixel updated operators. However, it does not   
provide us with a means for describing modeling complex geometric objects.   
The model for interpretation of OpenGL commands is client-server. An application issues   
commands, which are interpreted and processed by OpenGL. The server may or may not   
operate on the same computer as the client. In this sense, OpenGL network is transparent.

DEPT OF CS&E, SCEM, MANGALURU 2

SURGICAL STRIKE

GLU

GL Frame

OpenGL buffer

Application   
Program

GLUT Xlib, Xtk

GLX

Figure 1.2.1 OpenGL Library Organization

A sophisticated library that provides these features could certainly be built on top of OpenGL.   
The OpenGL Utility Library (GLU) provides many of the modeling features, such as quadric   
surfaces and NURBS curves and surfaces. GLU is a standard part of every OpenGL   
implementation. Also, there is a higher-level, object-oriented toolkit, Open Inventor, which is   
built atop OpenGL, and is available separately for many implementations of OpenGL.

Basically, our project is about environmental pollution with its different types of pollution   
along with the movement which contains some of the objects like sky, mountain, road, tree etc.   
in the background. Here we are using various OpenGL geometric primitives such as   
GL\_LINES, GL\_POLYGON, GL\_LINE\_LOOP, and GL\_POINTS. By using 3D   
transformations like Translation, Rotation. We have used different interactions like Mouse,   
Keyboard and Menu.

DEPT OF CS&E, SCEM, MANGALURU 3

SURGICAL STRIKE

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 Project Requirements

The package is designed such that users with a computer having minimum configuration can also use it. It does not require complex graphics packages.

The package requires simple in-built functions found in the <GL/glut.h> header file along with a few user defined functions.

2.2 Hardware Requirements

✓ Processor - Intel Core 5 processor. Processor Speed - 2.40 GHz.

✓ RAM - 8 MB or above Storage Space - 2 MB or above.

✓ Monitor resolution - A color monitor with a minimum resolution of 1000\*700

2.3 Software Requirements

✓ Operating System : Linux/Ubuntu/Windows 7 ✓ Language Tool: OpenGL

✓ Language used: C

✓ Compiler: Visual C Compiler.

✓ IDE: Microsoft Visual Studio 10.0/Linux gcc compiler

✓ Libraries: Supporting glut32.lib, opengl32.lib & glu32.lib

DEPT OF CS&E, SCEM, MANGALURU 4

SURGICAL STRIKE

CHAPTER 3

**DESIGN**

3.1 Algorithm

Step 1: Start

Step 2: Call display function to display the cover page.

Step 3: Use keyboard function to assign keys for various operations.

Step 4: Press key X to display the next screen by calling display function.

Step 5: In display function call different functions for displaying background objects and   
 movement of fighter jet.

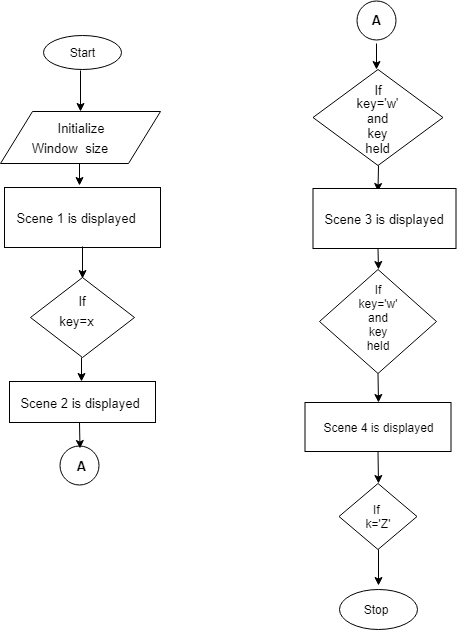
Step 6 Press the corresponding key displayed on the screen to move forward to the next slide. Step 7: Press key ‘Z’ to exit the display.

Step 8: Stop

DEPT OF CS&E, SCEM, MANGALURU 5

SURGICAL STRIKE

3.2 Flowchart



DEPT OF CS&E, SCEM, MANGALURU 6

SURGICAL STRIKE

CHAPTER 4

IMPLEMENTATION

4.1 OpenGL Commands

glVertex2i (…)

Library Basic commands Number of arguments Types of arguments

gl Vertex 2-(x,y) b-byte

glu Color 3-(x,y,z) un-unsigned byte

Clear 4-(x,y,z,w) or(r,g,b,a) s-short

glut Flush us-unsigned short

i-int

ui-unsigned int   
f-float

d-double

4.2 Buffers and their uses

An OpenGL () system can manipulate the following buffers:

4.2.1 Color Buffer

The color buffers are usually the ones you draw to. They contain either color-index or RGB color   
data and may also contain alpha values. An OpenGL implementation that supports stereoscopic   
viewing has left and right color buffers for the left and right stereo images. If stereo isn’t supported,   
Only the left buffers are used. Similarly, double-buffered systems have front and back buffer and a   
single.

DEPT OF CS&E, SCEM, MANGALURU 7

SURGICAL STRIKE

4.2.2 Clearing Buffers

The OpenGL clearing commands are structured to take advantage of architecture like this. First, you specify the values to be written into each buffer to be cleared. Then you issue a single command to perform the Clear Operation, passing in a list of all the Buffers to be cleared. If the hardware is capable of simultaneously clears, they all occur at once; otherwise each buffer is cleared sequentially. The following commands set the clearing values for each buffer.

glClearColor (GLclampf red,GLClampf blue,GLclampf alp);

4.3 OpenGL Functions

4.3.1 Specifying Simple Geometry 1.void glBegin (glEnum mode)

Initiates a new primitive of type mode and starts the collection of vertices. Values of mode include GL\_POLYGON, GL\_POINTS and GL\_LINES.

2.void glEnd()

Terminates a list of vertices. The set of vertex which has been drawn using the Begin statement will end when this function is executed.

DEPT OF CS&E, SCEM, MANGALURU 8

SURGICAL STRIKE

4.3.2 Attributes

1. void glClearColor(GLclampf r, GLclampf g, GLclampf b, Glclampf a)

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

2. void glPointSize(GLfloat size)

Sets the point size attribute in pixels.

4.3.3 Working with the window 1.void glFlush()

Forces any buffered OpenGL commands to execute.

2. void glutInit(int \*argc, char \*\*argv)

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

3. int glutCreateWindow(char \*title)

Creates a window on the display. The string title can be used to label the window.

4. void glutInitDisplayMode(unsigned int mode)

Requests a display with properties in mode. The value of mode is determined by logical OR of options including the color model (GLUT\_RGB,GLUT\_INDEX) and buffering   
(GLUT\_SINGLE,GLUT\_DOUBLE).

5. glutInitWindowSize(int width,int height)

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

6. void glutInitWindowPosition(int x,int y)

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

7. void glutMainLoop()

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

8. void glutDisplayFunc(void (\*func)(void))

Registers the display function func that is executed after the current callback returns.

9. void glutPostRedisplay()

Requests that the display callback be executed after the current callback returns.

4.3.4 Interactions

1. 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.

DEPT OF CS&E, SCEM, MANGALURU 9

SURGICAL STRIKE

4.3.5 Enabling Features

1.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. 2.void glDisable(GLenum feature)

Disables an openGL feature.

4.3.6 Transformations

1.void glMatrixMode(GLenum mode)

Specifies which matrix will be affected by subsequent transformations. Mode can be GL\_MODELVIEW, GL\_PROJECTION or GL\_TEXTURE.

2.void glLoadIdentity()

Sets the current transformation matrix to an identity matrix.

3.void glPushMatrix() & void glPopMatrix()

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

4.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).

5.void glTranslate[fd]( TYPE x, TYPE y, TYPE z)

Alters the current matrix by a displacement of(x, y, z).

4.3.7 Viewing

1.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.

2.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.

DEPT OF CS&E, SCEM, MANGALURU 10

SURGICAL STRIKE

4.4 User Defined Functions

1. 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.

2. void scene1()

This function is used to display the characters on the screen. This function is called only once.

3. void scene2()

This function is used to draw the background in various scenes in the project. This function is

called various times using different names but all does the same task and that is to draw the

background.

4. void scene3()

This function is used to draw the background for the 3rd scene. This function is called various times using different names but does the same task and that is to draw background for 3rd scene.

5. void scene4()

This function is used to draw the background for the 4th scene. This function is called once at the end of the program.

6. void building1()

This function is used to draw the 1st building. This function is called various times using

different names but does the same task and that is to draw 1st building.

7. void buliding2()

This function is used to draw the 2nd building. This function is called various times using

different names but does the same task and that is to draw 2nd building.

8. void building3()

This function is used to draw the 3rd building. This function is called various times using

different names but does the same task and that is to draw 3rd building.

9. void keys()

This is a keyboard function which does the task as specified when the key is pressed. This   
function takes three parameters that is, key, k and where k is the key pressed. X and Y are the   
coordinates.

DEPT OF CS&E, SCEM, MANGALURU 11

SURGICAL STRIKE

10. void jet()

This function is used to draw the fighter jet along with giving its color. This function is called various times using different names but does the same task and that is to draw the fighter jet.

11. void indian\_jet()

This function is used to draw the Indian fighter jet along with giving its color. This function is called various times using different names but does the same task and that is to draw the Indian fighter jet.

12. void pakistan\_jet()

This function is used to draw the Pakistan fighter jet along with giving its color. This function is called various times using different names but does the same task and that is to draw the Pakistan fighter jet.

13. void missile()

This function is used to draw two missiles. This function is called various number of times using same name but does the same task.

14. void explosion()

This function is used to show the explosion effect. This function is called various number of times but does the same task.

15. void stars()

This function is used to draw the stars in the background. This function is called various number of times with same name but does the same task.

16. void group\_stars()

This function is used to draw multiple stars in the background. This function is called various number of times with same name to perform the same task.

17. void move\_jet()

This function is used to move the fighter jet from one position to another position in 1st scene. This function is called various number of times with same name to perform same task.

18. void move\_jet1()

This function is used to move Indian fighter jet and Pakistan fighter jet in 2nd scene.

This function is called various number of times with same name to perform same task.

DEPT OF CS&E, SCEM, MANGALURU 12

SURGICAL STRIKE

18. void flight\_name()

This function is used to name the Indian fighter jet. This function is called various number of times with same name to perform same task.

19. void pak\_flight\_name()

This function is used to name the Pakistan fighter jet. This function is called once.

20. void draw\_circle()

This function is used to draw circle. It takes three parameter that is, r is the radius, and h and k are the coordinates.

DEPT OF CS&E, SCEM, MANGALURU 13

SURGICAL STRIKE

CHAPTER 5

RESULT

Following pages shown below are the graphical output obtained.

Figure 5.1

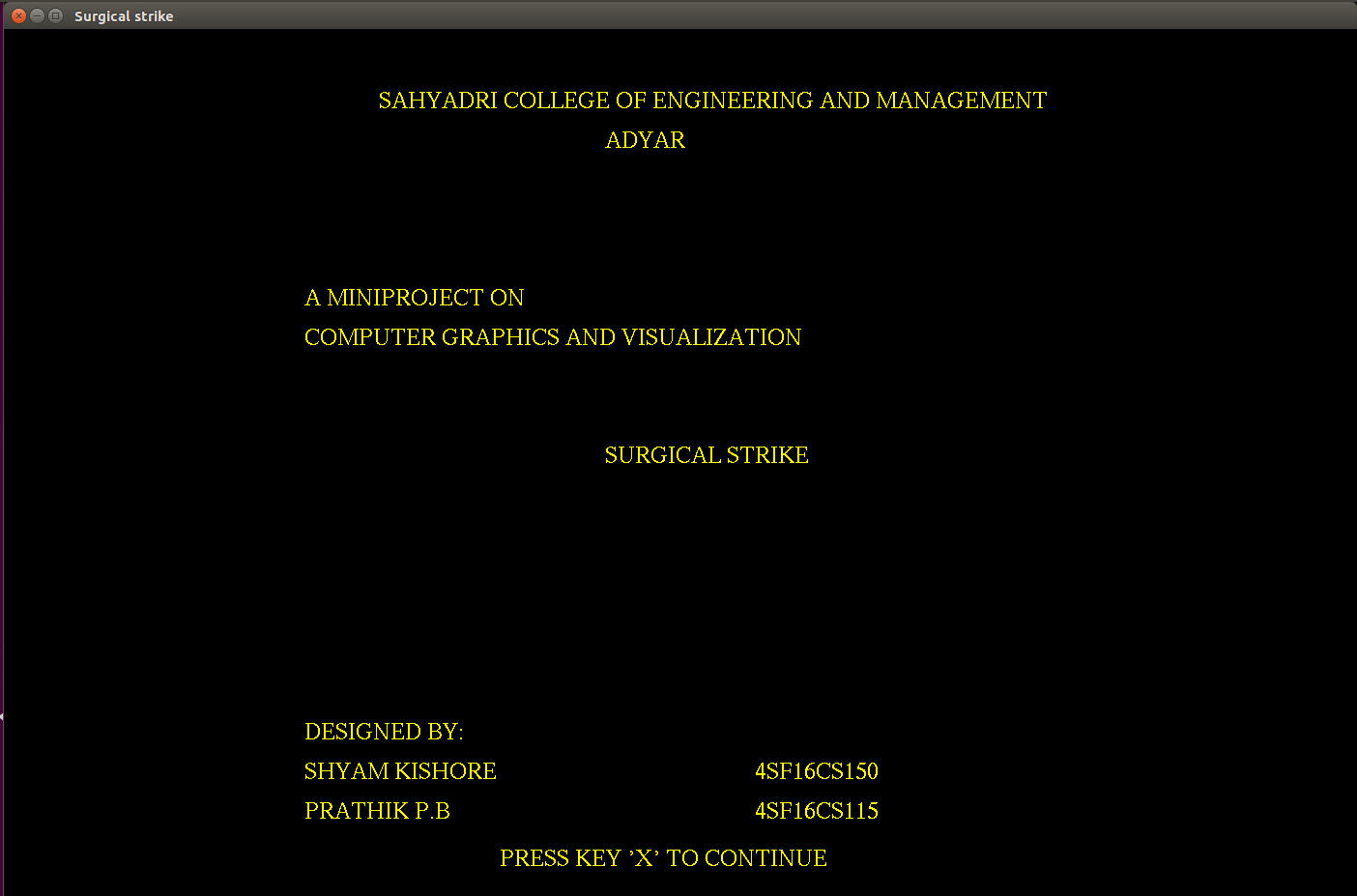


Figure 5.1: Page indicating the introduction scene

Figure 5.2

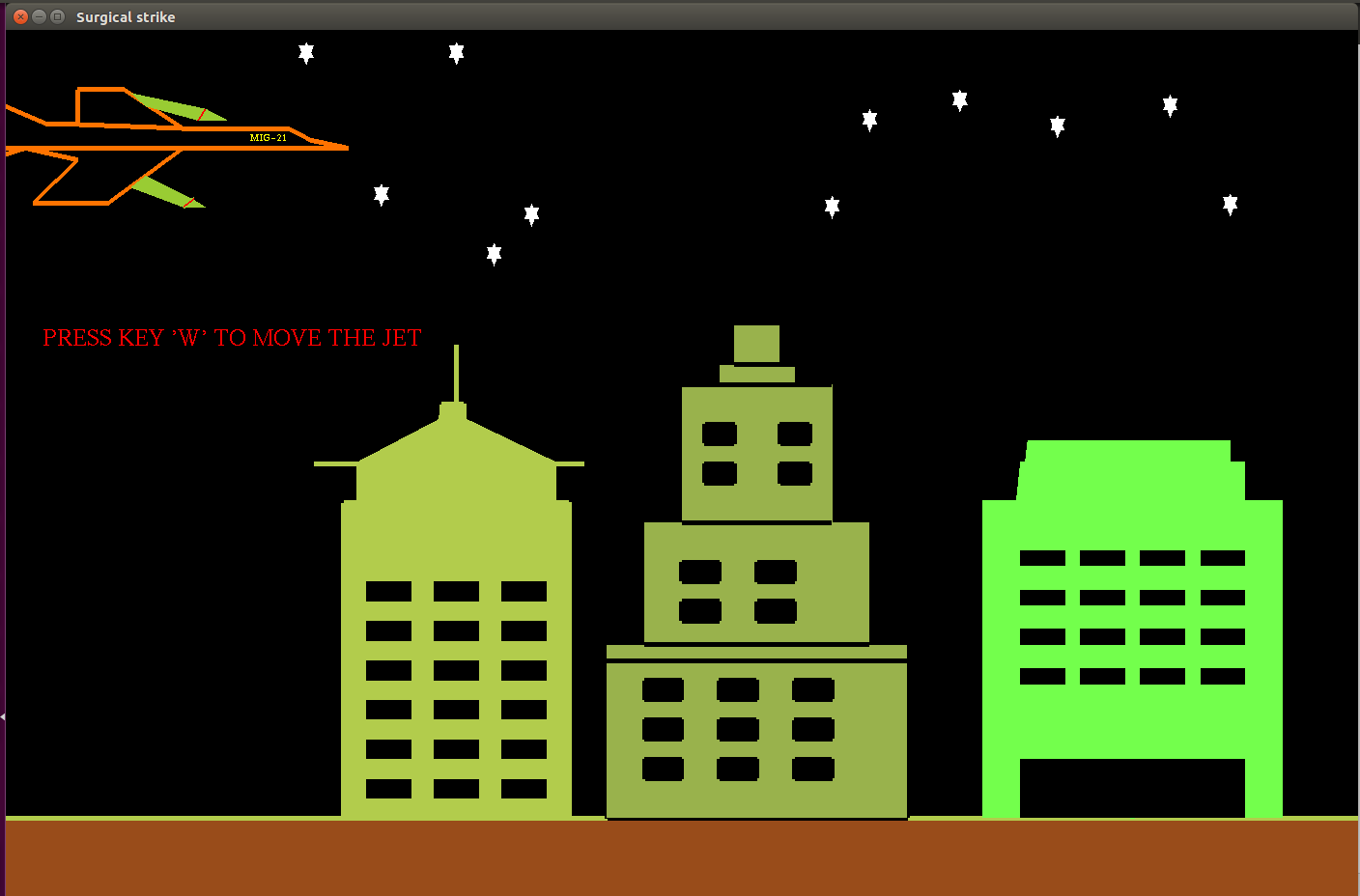


Figure 5.2: Fighter jet is ready to launch missiles on buildings

DEPT OF CS&E, SCEM, MANGALURU 14

SURGICAL STRIKE

Figure 5.3

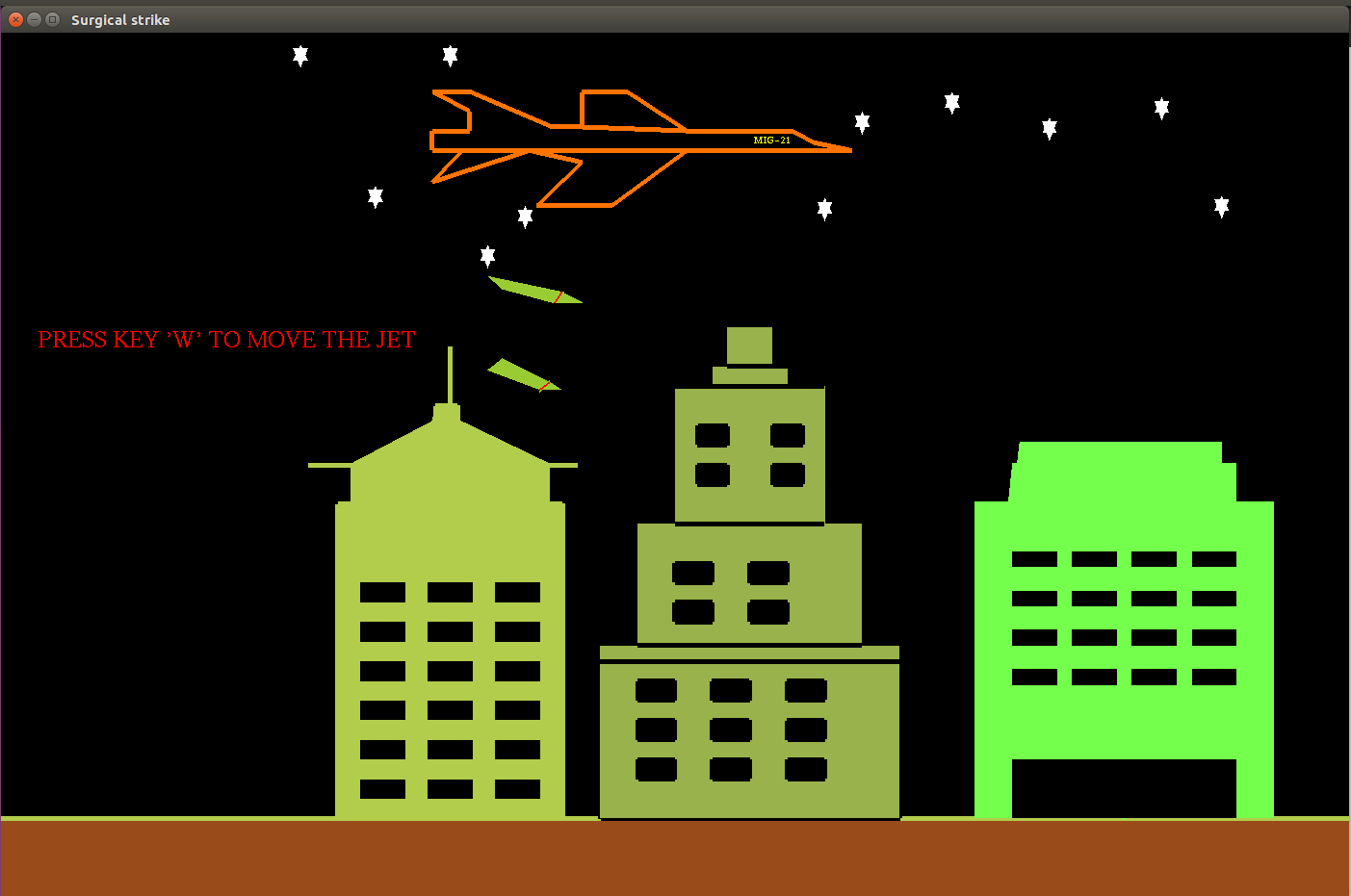


Figure 5.3: Fighter jet launches missiles on buildings

Figure 5.4

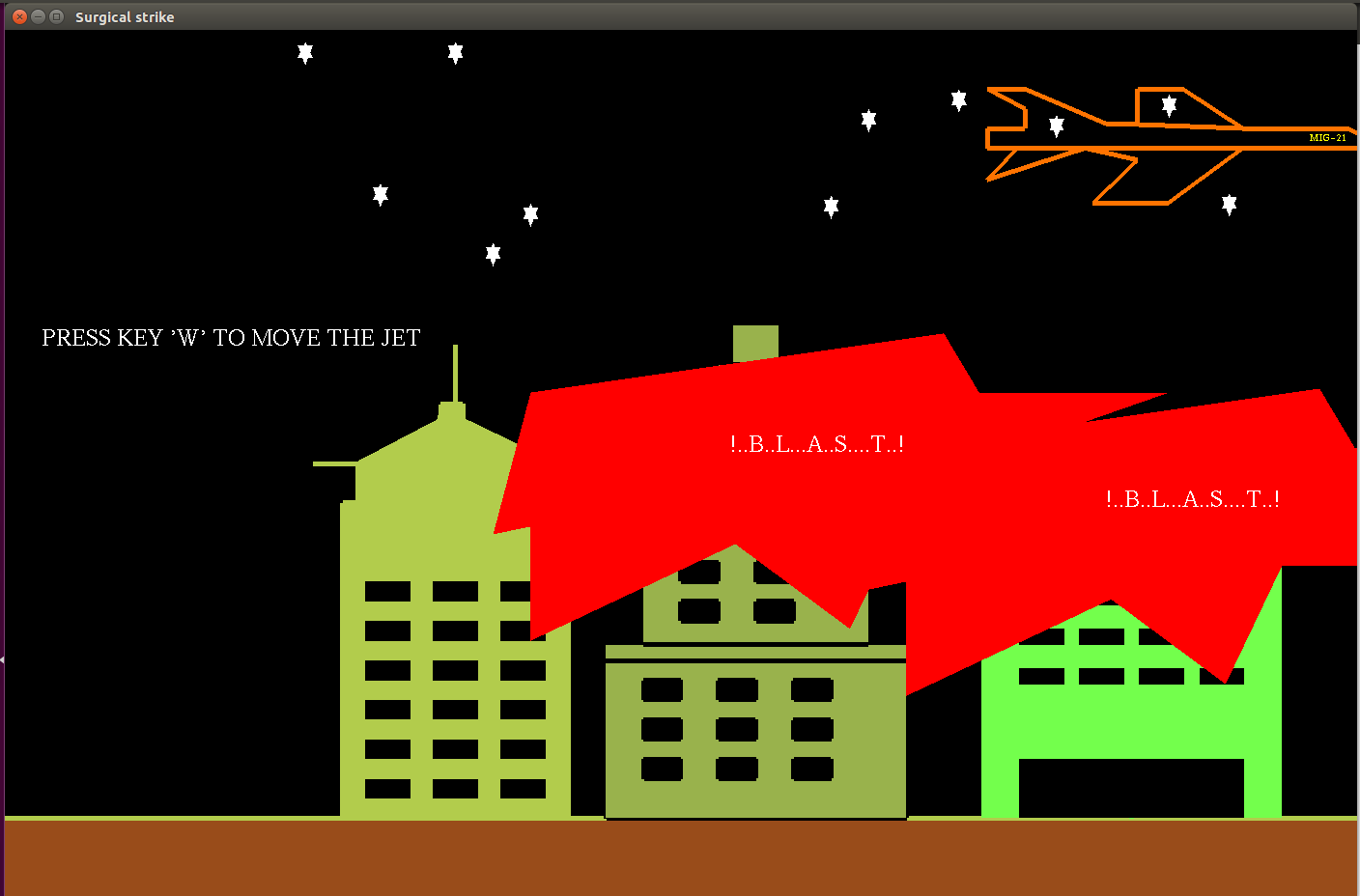


Figure 5.4: Launched missiles destroys enemy camp

DEPT OF CS&E, SCEM, MANGALURU 15

SURGICAL STRIKE

Figure 5.5

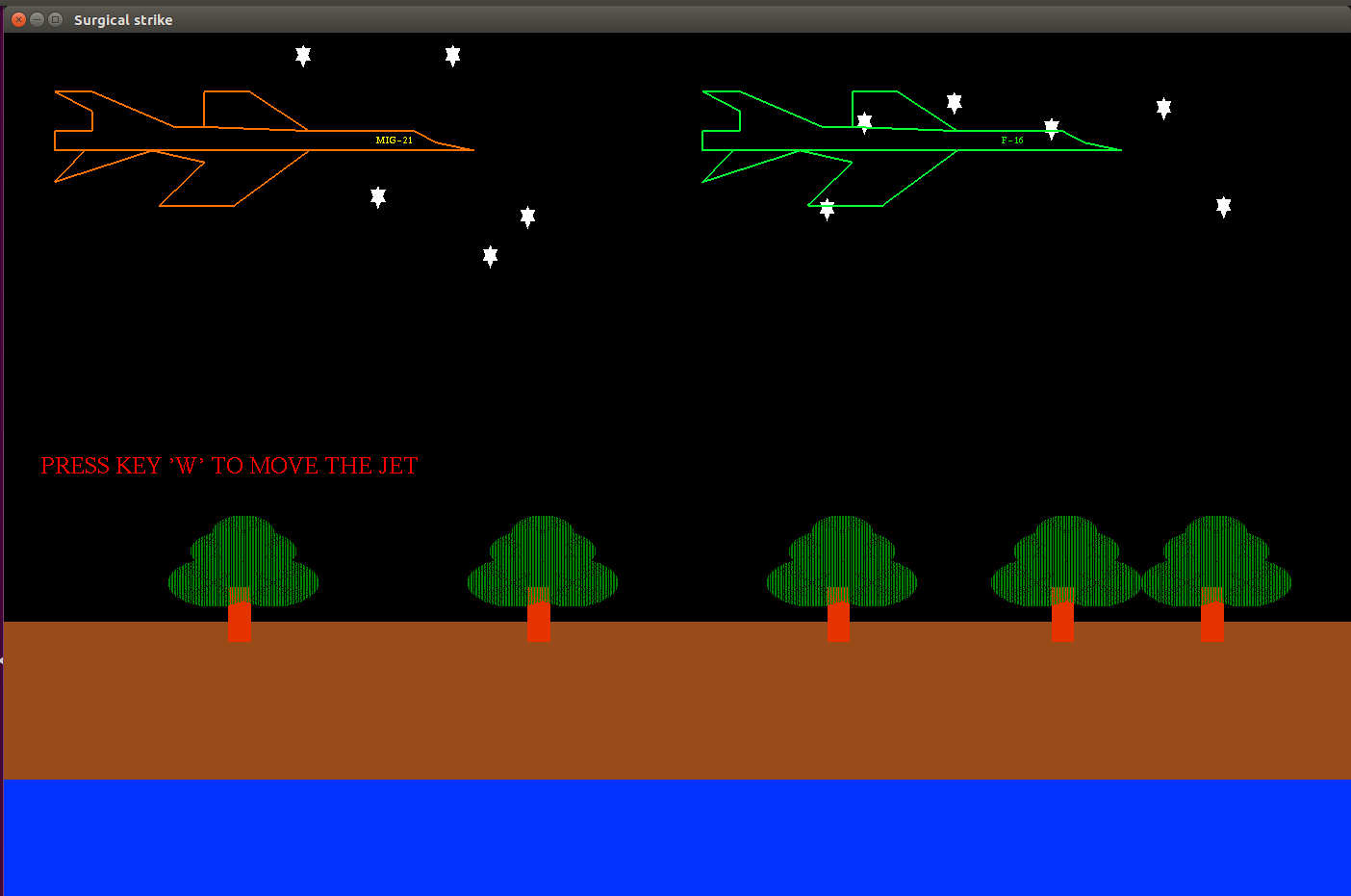


Figure 5.5: Indian fighter jet MIG-21 chases down Pakistan fighter jet F-16

Figure 5.6



Figure 5.6: Indian fighter jet MIG-21 hits Pakistan fighter jet F-16 causing explosion

DEPT OF CS&E, SCEM, MANGALURU 16

SURGICAL STRIKE

Figure 5.7

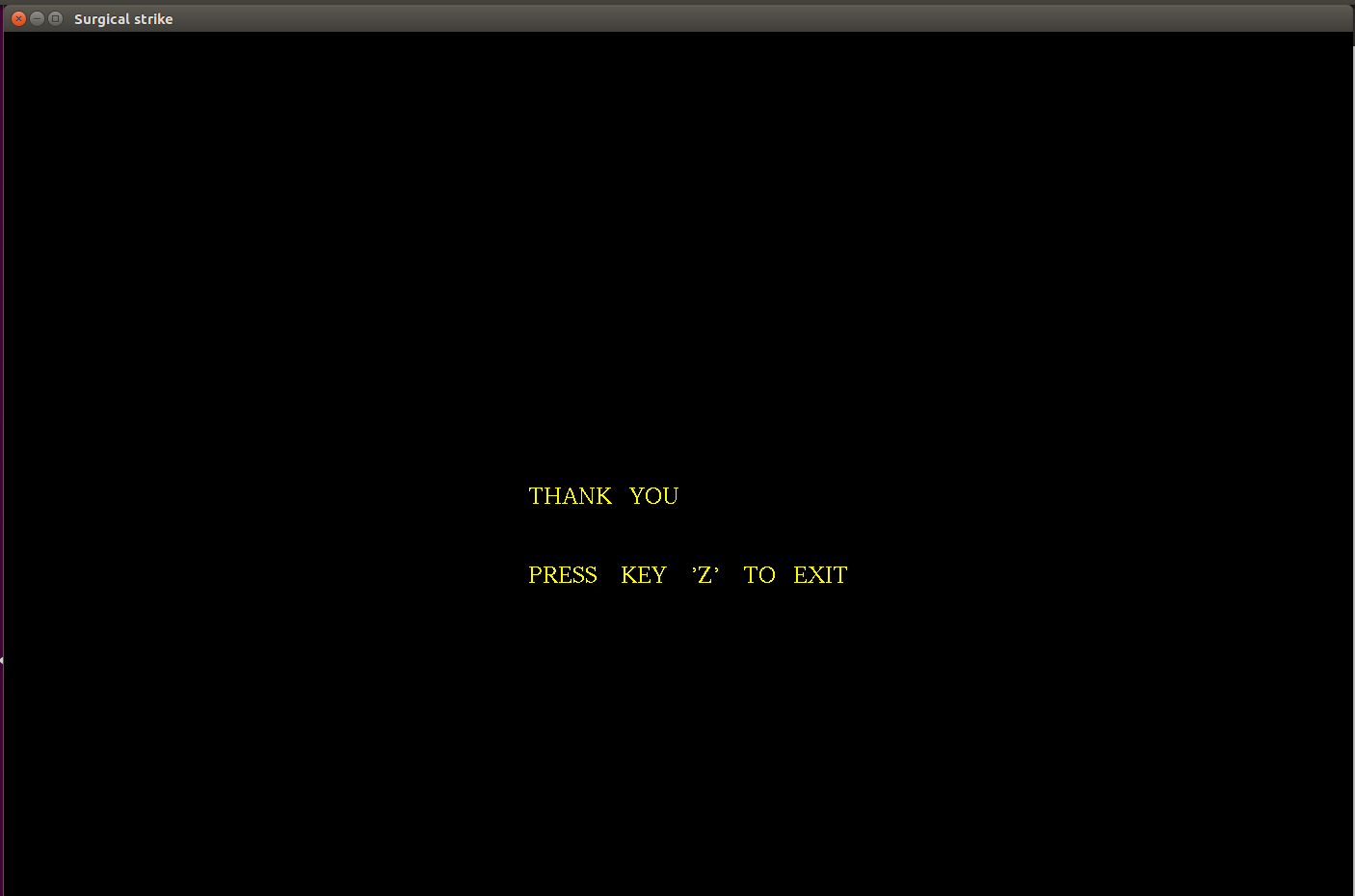


Figure 5.7: Page indicating the last scene

DEPT OF CS&E, SCEM, MANGALURU 17

SURGICAL STRIKE

CHAPTER 6

CONCLUSION

This project is based on the Air strike performed by our Indian Air force with several scenes’ depicting the cause, each scene which has a graphical output. The system provides a complete overview of the story.

It has several functions used to make the project look real and functions to make it more   
understandable. During the duration of the construction of the project we learnt various functions of   
OpenGL library and also how to optimize and develop a graphics project to depict certain concepts.

DEPT OF CS&E, SCEM, MANGALURU 18

SURGICAL STRIKE

REFERENCES

1. Donald D. Hearn,M.Pauline Baker, Warren Carithers, “Computer Graphics with OpenGL”,   
 Pearson Fourth Edition,2014

2. [www.openglprojects.in](http://www.openglprojects.in/)

3. [www.opengl-tutorial](http://www.opengl-tutorial/) .org

4. [www.opengl-redbook.com](http://www.opengl-redbook.com/)

5. [www.glprogramming.com](http://www.glprogramming.com/)

DEPT OF CS&E, SCEM, MANGALURU 19