REVIEW OF RESEARCH
IMPACT FACTOR : 5.7631(UIF) UGC APPROVED JOURNAL NO. 48514 ISSN: 2249-894X
VOLUME - 8 | ISSUE - 1 | OCTOBER - 2018

FIGHTER PLANE

Roopa R. L.
Gulbarga University,Gulbarga , PDA College of Engineering, Gulbarga.

ABSTRACT

This project illustrates the transformation and viewing API's using various functions. It describes three angles of viewing such as roll, pitch and yaw efficiently.Opengl can process the geometry of the plane faster by using the model view matrix. The keyboard keys are used to perform the various operations on the plane. Most of transformations used in computer graphics are affine. These transformations include rotation, translation and scaling.

KEYWORDS: *OpenGL, GLUT, Graphics, Rasterization, vector, matrix.*

INTRODUCTION TO OPENGL AND COMPUTER GRAPHICS

Computer graphics comprises the creation and representation of simple graphical elements and images, as well as modern techniques for rendering a virtual reality. To apply these techniques correctly, one requires a basic understanding of the fundamental concepts in graphics. **Computer graphics** are graphics created using computers and, more generally, the representation and manipulation of image 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 animation, movies and the video game industry.

The term *computer graphics* refers to several different things:

- The representation and manipulation of image data by a computer
- The various technologies used to create and manipulate images
- The images so produced, and
- The sub-field of computer science which studies methods for digitally synthesizing and manipulating visual content, see study of computer graphics

FUNCTIONS USED:

glutMainloop()
glutPostRedisplay ()
glVertex3fv()
glColor3fv()
glFlush()
glutSwapBuffers()
glutPostRedisplay()
glMatixMode()
glLoadIdentity()
glTranslatef()

glRotatef()
glClear()
glPushMatrix ()
glPopmatrix ()
glScalef()
glutInit()
glutCreateWindow()
glutInitWindowSize ()

TYPES OF COMPUTER GRAPHICS

There are two types of computer graphics: raster graphics, where each pixel is separately defined (as in a digital photograph), and vector graphics, where mathematical formulas are used to draw lines and shapes, which are then interpreted at the viewer's end to produce the graphic. Using vectors results in infinitely sharp graphics and often smaller files, but, when complex, vectors take time to render and may have larger file sizes than a raster equivalent.

GRAPHICS SYSTEM

It is a computer system and it must have all the components of a general purpose computer system.It consists of major elements like

- Input devices
- Processor
- Memory
- Frame buffer
- Output devices

OpenGL (Open Graphics Library) is a specification defining a cross-language cross-platform API for writing applications that produce 3D computer graphics (and 2D computer graphics as well). The interface consists of over 250 different function calls which can be used to draw complex three-dimensional scenes from simple primitives.

It is very popular in the video games industry where it competes with Direct3D on Microsoft Windows platforms. OpenGL is widely used in CAD, virtual reality, scientific visualization, information visualization and video game development.

OpenGL's basic operation is to accept primitives such as points, lines and polygons, and convert them into pixels. This is done by a graphics pipeline known as the OpenGL state machine. Most OpenGL commands either issue primitives to the graphics pipeline, or configure how the pipeline processes these primitives.

OpenGL Rendering Pipeline

All starts with Commands, which specify geometric objects and control how they are drawn and handled during the various processing stages. Rather than having all commands proceed immediately through the pipeline, you can choose to accumulate some of them in a display list for processing at a later time.

The evaluator stage of processing provides an efficient means for approximating curve and surface geometry.

During per-vertex operations and primitive assembly, OpenGL processes geometric primitives: points, line segments, and polygons, all of which are described by vertices. Vertices are transformed and lit, and primitives are clipped to the viewport.

Rasterization produces a series of frame buffer addresses and associated values, called fragments, using a two-dimensional description of a point, line segment, or polygon.

Each fragment is fed into the last stage, per-fragment operations, which performs the final operations on the data before it is stored as pixels in the frame buffer. These operations include conditional updates to the frame buffer, blending of incoming pixel colors with stored colors, as well as masking and other logical operations on pixel values.

Input data can be in the form of pixels rather than vertices. Such data, which might describe an image for use in texture mapping, skips the first stage of processing and instead is processed as pixels, in the pixel operations stage. The result of this stage is either stored as texture memory, for use in the rasterization stage, rasterized and the resulting fragments merged into the frame buffer just as if they were generated from geometric data.

GLUT and GLU and GLX and WGL

The **OpenGL Utility Toolkit (GLUT)** by Mark J. Kilgard is a cross-platform library of utilities for OpenGL programs:

- system-level I/O
- window control
- keyboard and mouse input
- Geometric primitives (solid and wireframe mode):cubes, spheres, and the Utah teapot

All GLUT functions start with the glut prefix, for example, glutPostRedisplay.
Freeglut and its spin-off, OpenGLUT, are open source alternatives to GLUT.

GLU is the OpenGL Utility Library, it provides higher-level drawing routines:

- Additional primitives including spheres, cylinders and disks
- Mapping between screen- and world-coordinates
- Generation of texture mipmaps
- Drawing of quadric surfaces
- NURBS
- Tessellation of polygonal primitives
- Interpretation of OpenGL error
- Extended transformations for viewing volumes camera

GLU functions have glu as a prefix, for example, gluOrtho2D(), which defines a two dimensional orthographic projection matrix.

GLX is a X-Windows application for OpenGL. It does Low-level Frame buffer, Color handling, etc.
WGL does the same MS Windows (DOS).

DESCRIPTION

In this project we use translation and rotation viewing APLS .we apply sequence of rotations and translation to the model view matrix.in the fighter plane,we can rotate the plane in three angles . Those are ROLL ,PITCH AND YAW. These angles are specified relative to the center of mass of the vehicle and to coordinate system aligned along the axes of the vehicle. Hence the pilot sees an object in terms of three angles and of the distance from the object to the center of the mass of pilots vehicle.

Flight simulation can be done by following keys

X or x – vertical rotation of the flight
Y or y - horizontal rotation of the flight.
U or u – Upward or downward movement of the flight

F or f – forward or backward movement of the flight
W- firing of bullets
Z – rotation of the flight.
Esc- exit
To perform all these functions, we need to adjust the plane in particular direction.

Applications

1. This project can be used for researches to construct the plane in wars.
2. It uses the effect of animation so it can be used as game application for children.
3. OpenGL provides options such as rotation, translation, scaling, color. By using these, we can create other animated structures efficiently.
4. By using OpenGL we can have an efficient view of an object by moving it and positioning of camera in particular direction.
5.It widely helps in animations ,games.

Implementation

Functions used in implementation:

- glutPostRedisplay(): Requests that the display callback be executed after the current call back returns.
- glPushMatrix(): Push the current matrix into the current matrix stack.
- glPopMatrix():Pop the current matrix from the current matrix stack.
- glLoadIdentity():set the current matrix to the identity matrix.
- glScalef():multiply the current matrix by general scaling matrix.

Syntax:

Void glScale(x,y,z);
Parameters:
x,y,z
scale factors along the x,y and z axes,respectively.

- glutInit():interaction between the windowing system and OPENGL is initiated.
- glutInitDisplayMode(): used when double buffering is required and depth information is required.
- glutCreateWindow(): this opens the OPENGL window and displays the title at top of the window.
- glutWindowSize():specifies the size of the window.
- glutInitWindowPosition():specifies the position of the window in screen coordinates.
- glutKeyboardFunc(): handles normal ascii symbols.
- glutIdleFunc(): this handles the processing of the background.
- glutDisplayFunc(): this handles redrawing of the window.
- glutMainLoop(): this starts the main loop,it never returns.
- glVertex3fv(): used to set up the points or vertices in three dimensions.
- glColor3fv(): used to render color to faces.
- glFlush(): used to flush the pipeline.
- glutSwapBuffers(): used to swap the front and back buffers.
- glMatrixMode(): used to set up the required mode of the matrix.
- glLoadIdentity(): used to load or initialize the Identity matrix.
- glTranslatef(): used to translate or move the otation centre from one point to another in three dimensions.
- glRotatef(): Used to rotate an object through a specified rotation angle.
- glClear(): used to clear the rendering window.
- glBegin():Describes what type of primitive you want to draw.

- glEnd(): Indicates when you wish to stop drawing that type of primitive.
- gluCylinder(): Draws the cylinder.

Syntax:

```
void gluCylinder(GLUquadric* quad,
GLdouble base,
GLdouble top,
GLdouble height,
GLint slices,
GLint stacks)
```

Description :

The gluCylinder subroutine draws a cylinder that is oriented along the z axis. the base of the cylinder is placed at z=0;the top of the cylinder is placed at z height. like a sphere, the cylinder is subdivided around the z axis into slices and along the z axis into stacks.

- gluQuadricDrawStyle (): specifies the desired quadric drawing style.

Syntax:

```
Void glutQuadricDrawStyle(GLUquadric* qobj,GLenum draw)
```

Description:

The gluQuadricDrawStyle subroutine specifies the draw style for quadrics rendered with the quad parameter.

System specification

Software requirements:

1.Microsoft visual C++
2.OPENGL

Hardware requirements:

1.Graphics system
2.Intel pentinum processor
3.256 RAM
4.Operating system:WINDOWS XP

Libraries required:

1.GL/glut
2.stdlib
3.math
4.stdio

CONCLUSION

This project concludes that the graphics primitives can be rendered efficiently and can be rotated effectively using OpenGL. It illustrates a simple use of OpenGL to animate motion.

Thus we come to know that how we can use fighter plane in order to implement 3D geometric primitives in space , and also rotate it by changing direction through keys of keyboard and using the functions provided by OpenGL GLUT library.

REFERENCES

We have obtained information from many resources to design and implement our project successively .we have acquired most of the knowledge from related websites .

The following are some of the resources:

- Interactive Computer Graphics a top down approach –by Edward angel
- Computer Graphics By James.D.Foley

The websites which were searched are

- www.wikipedia.com
- www.goole.com