NAME: VAIBHAV DNYANESHWAR MOHITE

CLASS: SE IT A

ROLLNO: 25045

# ASSIGNMENT NO: 8

CODE:

#include <GL/glut.h> // Header File For The GLUT Library

#include <GL/gl.h> // Header File For The OpenGL32 Library #include <GL/glu.h> // Header File For The GLu32 Library //#include <unistd.h> // Header File For sleeping.

/\* ASCII code for the escape key. \*/ #define ESCAPE 27

/\* The number of our GLUT window \*/ int window;

/\* rotation angle for the triangle. \*/ float rtri = 0.0f;

/\* rotation angle for the quadrilateral. \*/ float rquad = 0.0f;

/\* A general OpenGL initialization function. Sets all of the initial parameters. \*/ // We call this right after our OpenGL window is created.

void InitGL(int Width, int Height)

{

// This Will Clear The Background Color To Black

glClearColor(0.0f, 0.0f, 0.0f, 0.0f); glClearDepth(1.0);

// Enables Clearing Of The Depth Buffer glDepthFunc(GL\_LESS); // The Type Of Depth Test To Do glEnable(GL\_DEPTH\_TEST); // Enables Depth Testing glShadeModel(GL\_SMOOTH); // Enables Smooth Color Shading

glMatrixMode(GL\_PROJECTION);

glLoadIdentity(); // Reset The Projection Matrix gluPerspective(45.0f,(GLfloat)Width/(GLfloat)Height,0.1f,100.0f);

glMatrixMode(GL\_MODELVIEW);

}

/\* The function called when our window is resized (which shouldn't happen, because we're fullscreen) \*/

void ReSizeGLScene(int Width, int Height)

{

if (Height==0) // Prevent A Divide By Zero If The Window Is Too Small Height=1; glViewport(0, 0, Width, Height); // Reset The Current Viewport And Perspective Transformation

glMatrixMode(GL\_PROJECTION); glLoadIdentity();

gluPerspective(45.0f,(GLfloat)Width/(GLfloat)Height,0.1f,100.0f);

glMatrixMode(GL\_MODELVIEW);

}

float ballX = -0.5f; float ballY = 0.0f; float ballZ = 0.0f;

void drawBall(void) {

glColor3f(0.0, 1.0, 0.0); //set ball colour

glTranslatef(ballX,ballY,ballZ); //moving it toward the screen a bit on creation

//glRotatef(ballX,ballX,ballY,ballZ); glutSolidSphere (0.3, 20, 20); //create ball.

glTranslatef(ballX+1.5,ballY,ballZ); //moving it toward the screen a bit on creation glutSolidSphere (0.3, 20, 20); //

}

/\* The main drawing function. \*/

void DrawGLScene()

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT); // Clear The Screen And The Depth Buffer

glLoadIdentity(); // Reset The View

glTranslatef(rtri,0.0f,-6.0f); // Move Left 1.5 Units And Into The Screen 6.0

//glRotatef(rtri,1.0f,0.0f,0.0f); // Rotate The Triangle On The Y axis

// draw a triangle (in smooth coloring mode) glBegin(GL\_POLYGON); // start drawing a polygon glColor3f(1.0f,0.0f,0.0f); // Set The Color To Red glVertex3f(-1.0f, 1.0f, 0.0f); // Top left glVertex3f(0.4f, 1.0f, 0.0f);

glVertex3f(1.0f, 0.4f, 0.0f);

glColor3f(0.0f,1.0f,0.0f); // Set The Color To Green glVertex3f( 1.0f,0.0f, 0.0f); // Bottom Right glColor3f(0.0f,0.0f,1.0f); // Set The Color To Blue glVertex3f(-1.0f,0.0f, 0.0f);// Bottom Left

//glVertex3f();

glEnd(); // we're done with the polygon (smooth color interpolation) drawBall();

rtri+=0.005f; // Increase The Rotation Variable For The Triangle if(rtri>2) rtri=-2.0f;

rquad-=15.0f; // Decrease The Rotation Variable For The Quad

// swap the buffers to display, since double buffering is used. glutSwapBuffers();

}

/\* The function called whenever a key is pressed. \*/

void keyPressed(unsigned char key, int x, int y)

{

/\* sleep to avoid thrashing this procedure \*/ // usleep(100);

/\* If escape is pressed, kill everything. \*/ if (key == ESCAPE)

{

/\* shut down our window \*/

glutDestroyWindow(window);

/\* exit the program...normal termination. \*/ exit(0);

}

}

int main(int argc, char \*\*argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_RGBA | GLUT\_DOUBLE | GLUT\_ALPHA | GLUT\_DEPTH);

/\* get a 640 x 480 window \*/ glutInitWindowSize(640, 480);

/\* the window starts at the upper left corner of the screen \*/ glutInitWindowPosition(0, 0);

/\* Open a window \*/

window = glutCreateWindow("Moving Car");

/\* Register the function to do all our OpenGL drawing. \*/ glutDisplayFunc(&DrawGLScene);

/\* Go fullscreen. This is as soon as possible. \*/ //glutFullScreen();

/\* Even if there are no events, redraw our gl scene. \*/ glutIdleFunc(&DrawGLScene);

/\* Register the function called when our window is resized. \*/ glutReshapeFunc(&ReSizeGLScene);

/\* Register the function called when the keyboard is pressed. \*/ glutKeyboardFunc(&keyPressed);

/\* Initialize our window. \*/

InitGL(640, 480);

/\* Start Event Processing Engine \*/ glutMainLoop();

return 1; }

OUTPUT: Moving Car

 