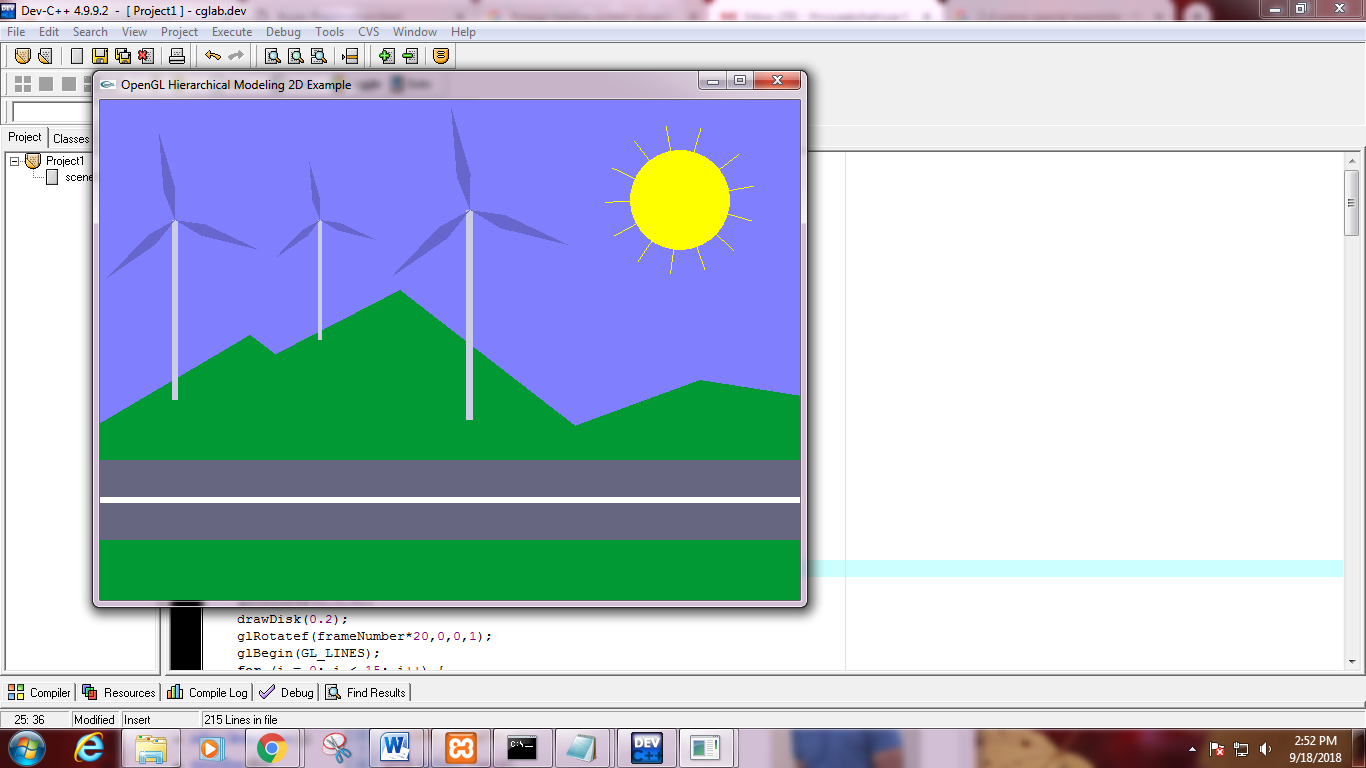
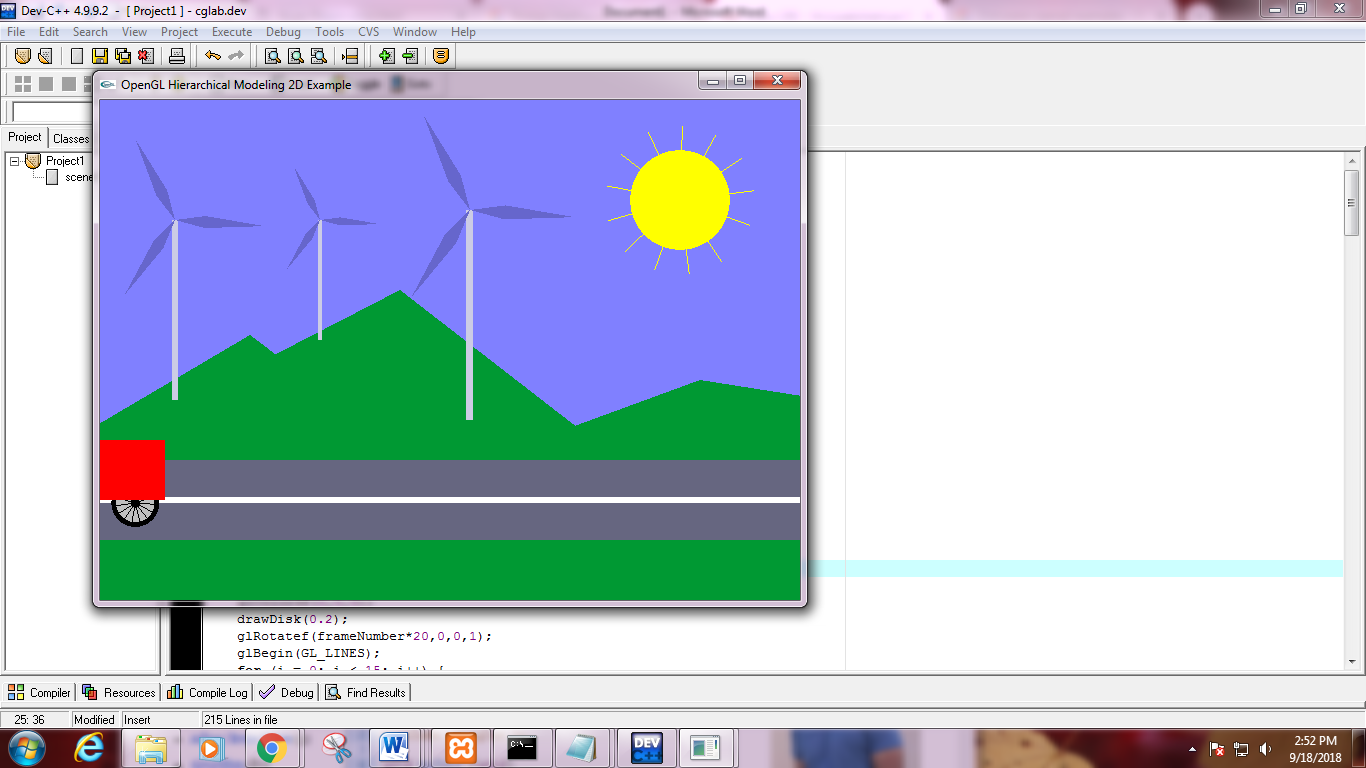
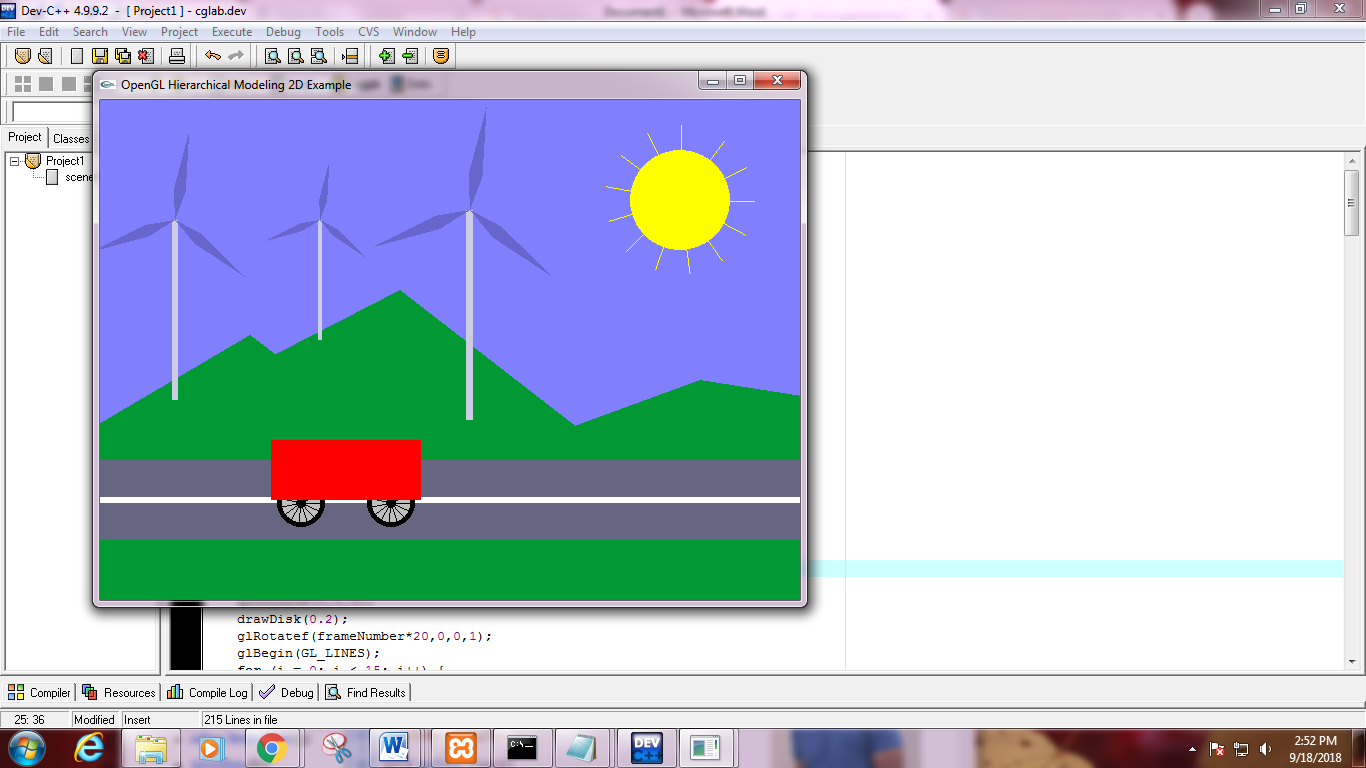
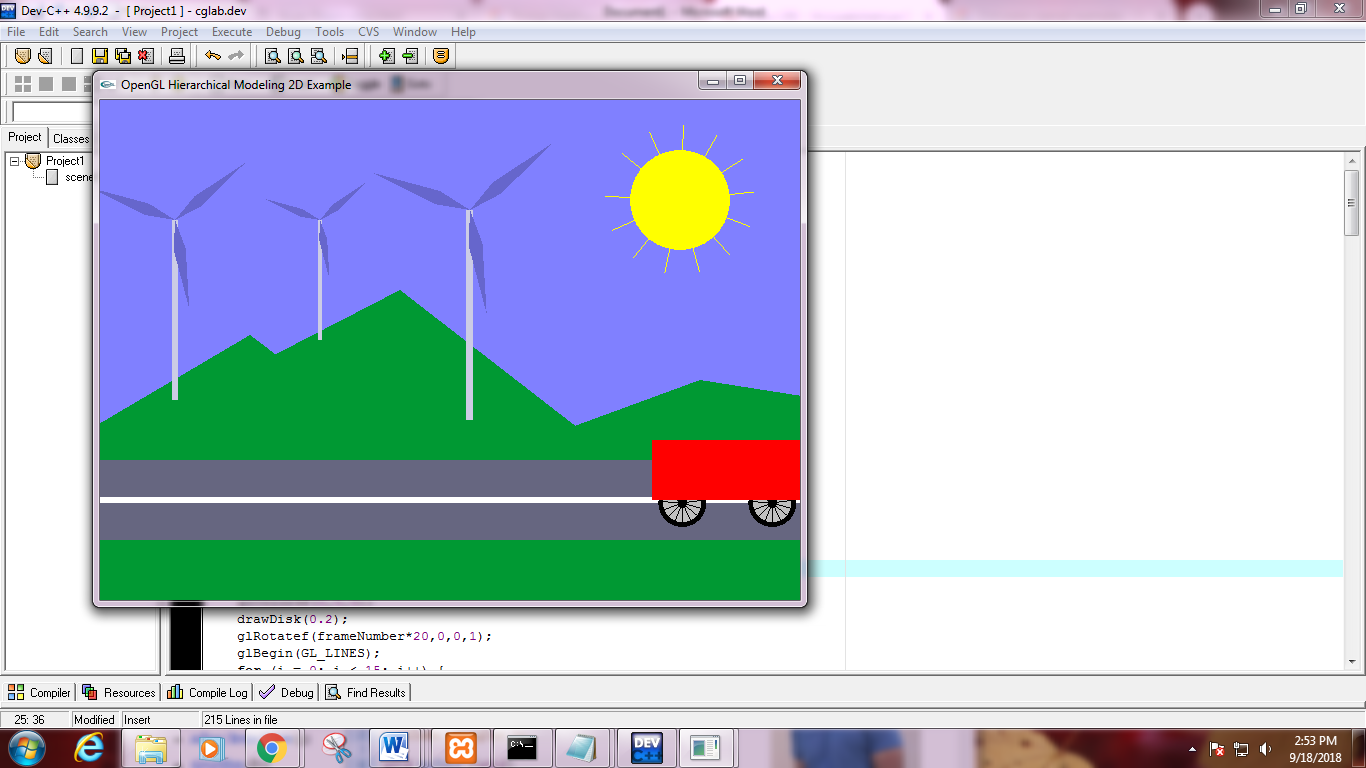
**OUTPUT:**









**PROGRAM:**

#include <GL/gl.h>

#include <GL/glut.h>

#include <math.h>

const double PI = 3.141592654;

int frameNumber = 0;

void drawDisk(double radius) {

int d;

glBegin(GL\_POLYGON);

for (d = 0; d < 32; d++) {

double angle = 2\*PI/32 \* d;

glVertex2d( radius\*cos(angle), radius\*sin(angle));

}

glEnd();

}

void drawWheel() {

int i;

glColor3f(0,0,0);

drawDisk(1);

glColor3f(0.75f, 0.75f, 0.75f);

drawDisk(0.8);

glColor3f(0,0,0);

drawDisk(0.2);

glRotatef(frameNumber\*20,0,0,1);

glBegin(GL\_LINES);

for (i = 0; i < 15; i++) {

glVertex2f(0,0);

glVertex2d(cos(i\*2\*PI/15), sin(i\*2\*PI/15));

}

glEnd();

}

void drawCart() {

glPushMatrix();

glTranslatef(-1.5f, -0.1f, 0);

glScalef(0.8f,0.8f,1);

drawWheel();

glPopMatrix();

glPushMatrix();

glTranslatef(1.5f, -0.1f, 0);

glScalef(0.8f,0.8f,1);

drawWheel();

glPopMatrix();

glColor3f(1,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-2.5f,0);

glVertex2f(2.5f,0);

glVertex2f(2.5f,2);

glVertex2f(-2.5f,2);

glEnd();

}

void drawSun() {

int i;

glColor3f(1,1,0);

for (i = 0; i < 13; i++) {

glRotatef( 360 / 13, 0, 0, 1 );

glBegin(GL\_LINES);

glVertex2f(0, 0);

glVertex2f(0.75f, 0);

glEnd();

}

drawDisk(0.5);

glColor3f(0,0,0);

}

void drawWindmill() {

int i;

glColor3f(0.8f, 0.8f, 0.9f);

glBegin(GL\_POLYGON);

glVertex2f(-0.05f, 0);

glVertex2f(0.05f, 0);

glVertex2f(0.05f, 3);

glVertex2f(-0.05f, 3);

glEnd();

glTranslatef(0, 3, 0);

glRotated(frameNumber \* (180.0/46), 0, 0, 1);

glColor3f(0.4f, 0.4f, 0.8f);

for (i = 0; i < 3; i++) {

glRotated(120, 0, 0, 1);

glBegin(GL\_POLYGON);

glVertex2f(0,0);

glVertex2f(0.5f, 0.1f);

glVertex2f(1.5f,0);

glVertex2f(0.5f, -0.1f);

glEnd();

}

}

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glLoadIdentity();

glColor3f(0, 0.6f, 0.2f);

glBegin(GL\_POLYGON);

glVertex2f(-3,-1);

glVertex2f(1.5f,1.65f);

glVertex2f(5,-1);

glEnd();

glBegin(GL\_POLYGON);

glVertex2f(-3,-1);

glVertex2f(3,2.1f);

glVertex2f(7,-1);

glEnd();

glBegin(GL\_POLYGON);

glVertex2f(0,-1);

glVertex2f(6,1.2f);

glVertex2f(20,-1);

glEnd();

glColor3f(0.4f, 0.4f, 0.5f);

glBegin(GL\_POLYGON);

glVertex2f(0,-0.4f);

glVertex2f(7,-0.4f);

glVertex2f(7,0.4f);

glVertex2f(0,0.4f);

glEnd();

glLineWidth(6);

glColor3f(1,1,1);

glBegin(GL\_LINES);

glVertex2f(0,0);

glVertex2f(7,0);

glEnd();

glLineWidth(1);

glPushMatrix();

glTranslated(5.8,3,0);

glRotated(-frameNumber\*0.7,0,0,1);

drawSun();

glPopMatrix();

glPushMatrix();

glTranslated(0.75,1,0);

glScaled(0.6,0.6,1);

drawWindmill();

glPopMatrix();

glPushMatrix();

glTranslated(2.2,1.6,0);

glScaled(0.4,0.4,1);

drawWindmill();

glPopMatrix();

glPushMatrix();

glTranslated(3.7,0.8,0);

glScaled(0.7,0.7,1);

drawWindmill();

glPopMatrix();

glPushMatrix();

glTranslated(-3 + 13\*(frameNumber % 300) / 300.0, 0, 0);

glScaled(0.3,0.3,1);

drawCart();

glPopMatrix();

glutSwapBuffers();

}

void doFrame(int v) {

frameNumber++;

glutPostRedisplay();

glutTimerFunc(30,doFrame,0);

}

void init() {

glClearColor(0.5f, 0.5f, 1, 1);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0, 7, -1, 4, -1, 1);

glMatrixMode(GL\_MODELVIEW);

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE);

glutInitWindowSize(700,500);

glutInitWindowPosition(100,100);

glutCreateWindow("OpenGL Hierarchical Modeling 2D Example");

init();

glutDisplayFunc(display);

glutTimerFunc(200,doFrame,0);

glutMainLoop();

return 0;

}