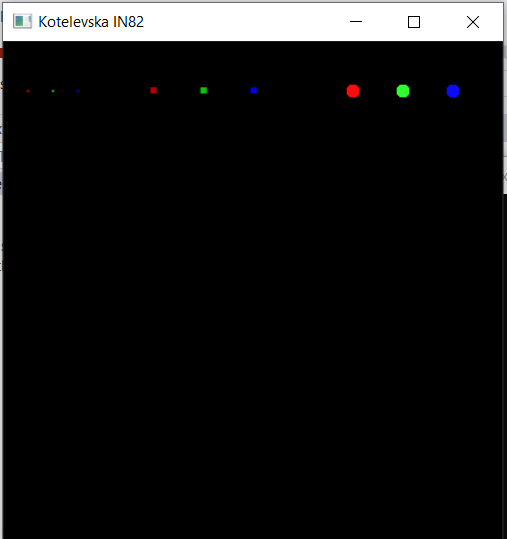
Лабораторна робота 3

Котелевська Вероніка ІН-82

Завдання 1.



// Simple\_objects.cpp : This file contains the 'main' function. Program execution begins and ends there.

//

#include <windows.h>

#include <GL/gl.h>

#include <GL/glu.h>

#include <GLAux.h>

void CALLBACK resize(int width, int height)

{

glViewport(0, 0, width, height);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-5, 5, -5, 5, 2, 12);

gluLookAt(0, 0, 5, 0, 0, 0, 0, 1, 0);

glMatrixMode(GL\_MODELVIEW);

}

void CALLBACK display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glPointSize(2);

glBegin(GL\_POINTS);

glColor3d(1, 0, 0);

glVertex3d(-4.5, 4, 0); // перша точка

glColor3d(0, 1, 0);

glVertex3d(-4, 4, 0); // друга точка

glColor3d(0, 0, 1); // третя

glVertex3d(-3.5, 4, 0);

glEnd();

glPointSize(5);

glBegin(GL\_POINTS);

glColor3d(1, 0, 0);

glVertex3d(-2, 4, 0); // перша точка

glColor3d(0, 1, 0);

glVertex3d(-1, 4, 0); // друга точка

glColor3d(0, 0, 1); // третя

glVertex3d(0, 4, 0);

glEnd();

glPointSize(10);

glEnable(GL\_POINT\_SMOOTH);

glBegin(GL\_POINTS);

glColor3d(1, 0, 0);

glVertex3d(2, 4, 0); // перша точка

glColor3d(0, 1, 0);

glVertex3d(3, 4, 0); // друга точка

glColor3d(0, 0, 1); // третя

glVertex3d(4, 4, 0);

glEnd();

glDisable(GL\_POINT\_SMOOTH);

auxSwapBuffers();

}

int main()

{

float pos[4] = { 3,3,3,1 };

float dir[3] = { -1,-1,-1 };

GLfloat mat\_specular[] = { 1,1,1,1 };

auxInitPosition(50, 10, 400, 400);

auxInitDisplayMode(AUX\_RGB | AUX\_DEPTH | AUX\_DOUBLE);

auxInitWindowA("Kotelevska IN82");

auxIdleFunc(display);

auxReshapeFunc(resize);

glEnable(GL\_DEPTH\_TEST);

glEnable(GL\_COLOR\_MATERIAL);

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glLightfv(GL\_LIGHT0, GL\_POSITION, pos);

glLightfv(GL\_LIGHT0, GL\_SPOT\_DIRECTION, dir);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, mat\_specular);

glMaterialf(GL\_FRONT, GL\_SHININESS, 128.0);

/\*

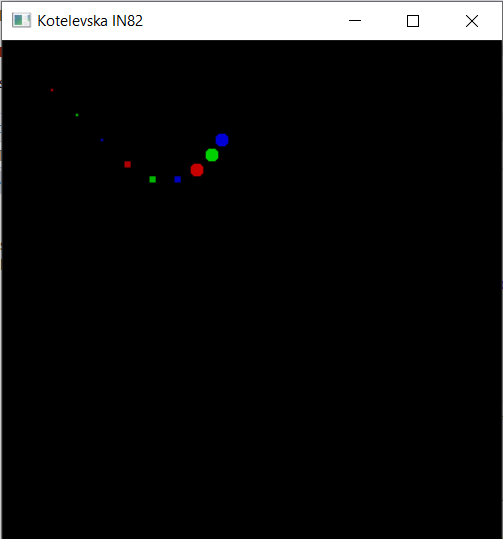
\* Enter your cod here

\*/

auxMainLoop(display);

}

Потім змініть проект так, щоб точки розташовувалися по вертикалі (горизонталі посередині, знизу, по діагоналі) екрана, змінився їх розмір і колір. Збережіть отриманий проект, копію екрану і властивостей файлу вставте в звіт.



void CALLBACK display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glPointSize(2);

glBegin(GL\_POINTS);

glColor3d(1, 0, 0);

glVertex3d(-4, 4, 0); // перша точка

glColor3d(0, 1, 0);

glVertex3d(-3.5, 3.5, 0); // друга точка

glColor3d(0, 0, 1); // третя

glVertex3d(-3, 3, 0);

glEnd();

glPointSize(5);

glBegin(GL\_POINTS);

glColor3d(1, 0, 0);

glVertex3d(-2.5, 2.5, 0); // перша точка

glColor3d(0, 1, 0);

glVertex3d(-2, 2.2, 0); // друга точка

glColor3d(0, 0, 1);

glVertex3d(-1.5, 2.2, 0);// третя

glEnd();

glPointSize(10);

glEnable(GL\_POINT\_SMOOTH);

glBegin(GL\_POINTS);

glColor3d(1, 0, 0);

glVertex3d(-1.1, 2.4, 0); // перша точка

glColor3d(0, 1, 0);

glVertex3d(-0.8, 2.7, 0); // друга точка

glColor3d(0, 0, 1); // третя

glVertex3d(-0.6, 3, 0);

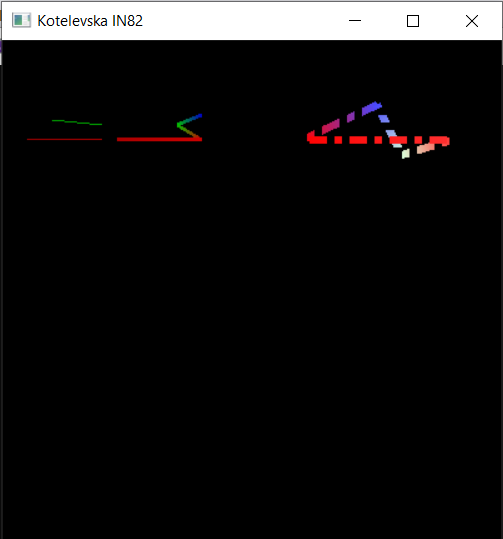
glEnd();

glDisable(GL\_POINT\_SMOOTH);

auxSwapBuffers();

}

Завдання 2 ***Малювання ліній***.



void CALLBACK display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glLineWidth(1); // ширину лінії встановлюємо 1

glBegin(GL\_LINES);

glColor3d(1, 0, 0); // червоний колір

glVertex3d(-4.5, 3, 0); // перша лінія

glVertex3d(-3, 3, 0);

glColor3d(0, 1, 0); // зелений

glVertex3d(-3, 3.3, 0); // друга лінія

glVertex3d(-4, 3.4, 0);

glEnd();

glLineWidth(3); // ширина 3

glBegin(GL\_LINE\_STRIP); //

glColor3d(1, 0, 0);

glVertex3d(-2.7, 3, 0);

glVertex3d(-1, 3, 0);

glColor3d(0, 1, 0);

glVertex3d(-1.5, 3.3, 0);

glColor3d(0, 0, 1);

glVertex3d(-1, 3.5, 0);

glEnd();

glLineWidth(5);

glEnable(GL\_LINE\_SMOOTH);

glEnable(GL\_LINE\_STIPPLE); // дозволяємо малювати переривчасту лінію

glLineStipple(2, 58360); // встановлюємо маску

glBegin(GL\_LINE\_LOOP);

glColor3d(1, 0, 0);

glVertex3d(1, 3, 0);

glVertex3d(4, 3, 0);

glColor3d(0, 1, 0);

glVertex3d(3, 2.7, 0);

glColor3d(0, 0, 1);

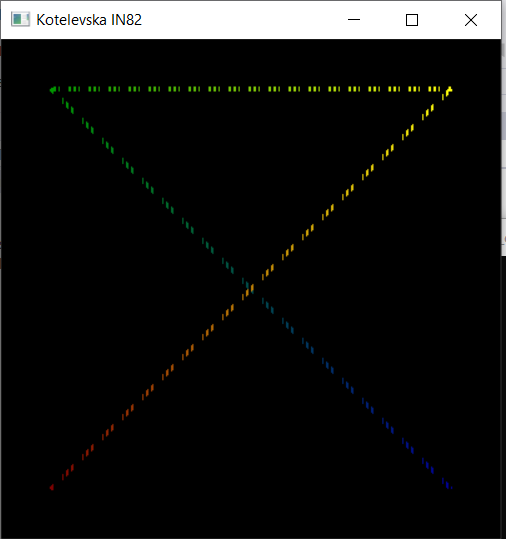
glVertex3d(2.5, 3.7, 0);

glEnd();

glDisable(GL\_LINE\_SMOOTH);

glDisable(GL\_LINE\_STIPPLE);

auxSwapBuffers();}



void CALLBACK display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glLineWidth(3);

glEnable(GL\_LINE\_SMOOTH);

glEnable(GL\_LINE\_STIPPLE);

glLineStipple(1, 12803);

glBegin(GL\_LINE\_STRIP);

glColor3d(1, 0, 0);

glVertex3d(-4, -4, 0);

glColor3d(1, 1, 0);

glVertex3d(4, 4, 0);

glColor3d(0, 1, 0);

glVertex3d(-4, 4, 0);

glColor3d(0, 0, 1);

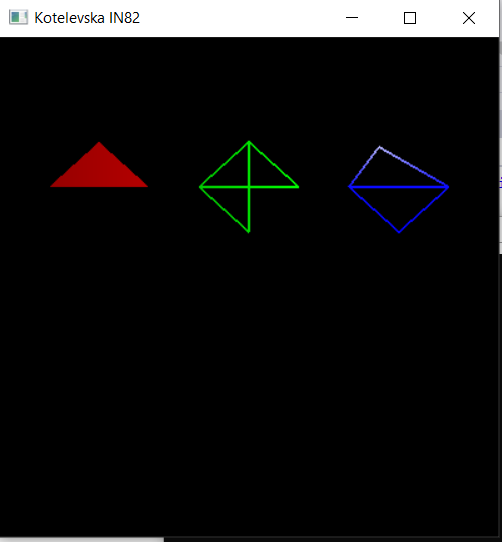
glVertex3d(4, -4, 0);

glEnd();

auxSwapBuffers();

}

Завдання 3. ***Малювання трикутників***



glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL); //

glBegin(GL\_TRIANGLES);

glColor3d(1,0,0); // малюємо трикутник

glVertex3d(-4,2,0);

glVertex3d(-3,2.9,0);

glVertex3d(-2,2,0);

glEnd();

glLineWidth(2);

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE); //мадюємо

// дротові трикутники

glBegin(GL\_TRIANGLE\_STRIP); // зверніть увагу на порядок вершин

glColor3d(0,1,0);

glVertex3d(1,2,0);

glVertex3d(0,2.9,0);

glVertex3d(-1,2,0);

glVertex3d(0,1.1,0);

glEnd();

glEnable(GL\_LINE\_STIPPLE);

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);

glBegin(GL\_TRIANGLE\_FAN);

glColor3d(0,0,1);

glVertex3d(4,2,0);

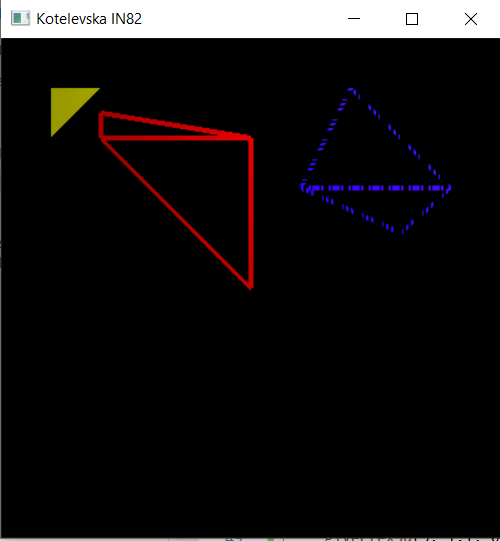
glVertex3d(2.6,2.8,0);

glVertex3d(2,2,0);

glVertex3d(3,1.1,0);

glEnd();

glDisable(GL\_LINE\_STIPPLE);



void CALLBACK display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);

glBegin(GL\_TRIANGLES);

glColor3d(1, 1, 0);

glVertex3d(-4, 4, 0);

glVertex3d(-3, 4, 0);

glVertex3d(-4, 3, 0);

glEnd();

glLineWidth(4);

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);

glBegin(GL\_TRIANGLE\_STRIP);

glColor3d(1, 0, 0);

glVertex3d(0, 0, 0);

glVertex3d(0, 3, 0);

glVertex3d(-3, 3, 0);

glVertex3d(-3, 3.5, 0);

glEnd();

glEnable(GL\_LINE\_STIPPLE);

glLineStipple(1, 12803);

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);

glBegin(GL\_TRIANGLE\_FAN);

glColor3d(0.2, 0, 1);

glVertex3d(4, 2, 0);

glVertex3d(2, 4, 0);

glVertex3d(1, 2, 0);

glVertex3d(3, 1.1, 0);

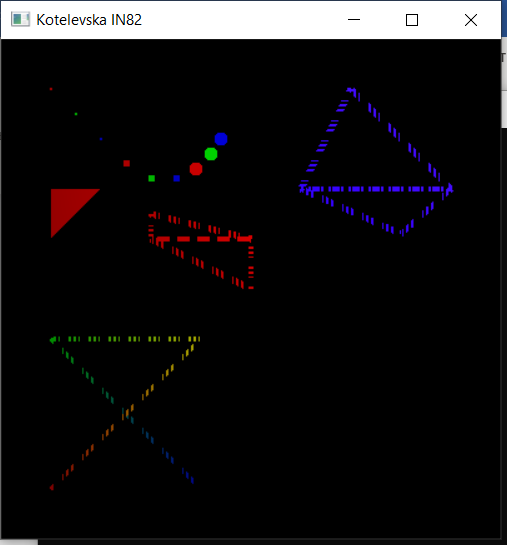
glEnd();

glDisable(GL\_LINE\_STIPPLE);

auxSwapBuffers();

}

***Завдання 4 Примітиви***



void CALLBACK display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glPointSize(2);

glBegin(GL\_POINTS);

glColor3d(1, 0, 0);

glVertex3d(-4, 4, 0); // перша точка

glColor3d(0, 1, 0);

glVertex3d(-3.5, 3.5, 0); // друга точка

glColor3d(0, 0, 1); // третя

glVertex3d(-3, 3, 0);

glEnd();

glPointSize(5);

glBegin(GL\_POINTS);

glColor3d(1, 0, 0);

glVertex3d(-2.5, 2.5, 0); // перша точка

glColor3d(0, 1, 0);

glVertex3d(-2, 2.2, 0); // друга точка

glColor3d(0, 0, 1);

glVertex3d(-1.5, 2.2, 0);// третя

glEnd();

glPointSize(10);

glEnable(GL\_POINT\_SMOOTH);

glBegin(GL\_POINTS);

glColor3d(1, 0, 0);

glVertex3d(-1.1, 2.4, 0); // перша точка

glColor3d(0, 1, 0);

glVertex3d(-0.8, 2.7, 0); // друга точка

glColor3d(0, 0, 1); // третя

glVertex3d(-0.6, 3, 0);

glEnd();

glDisable(GL\_POINT\_SMOOTH);

glLineWidth(3);

glEnable(GL\_LINE\_SMOOTH);

glEnable(GL\_LINE\_STIPPLE);

glLineStipple(1, 12803);

glBegin(GL\_LINE\_STRIP);

glColor3d(1, 0, 0);

glVertex3d(-4, -4, 0);

glColor3d(1, 1, 0);

glVertex3d(-1, -1, 0);

glColor3d(0, 1, 0);

glVertex3d(-4, -1, 0);

glColor3d(0, 0, 1);

glVertex3d(-1, -4, 0);

glEnd();

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);

glBegin(GL\_TRIANGLES);

glColor3d(1, -1, 0);

glVertex3d(-4, 2, 0);

glVertex3d(-3, 2, 0);

glVertex3d(-4, 1, 0);

glEnd();

glLineWidth(4);

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);

glBegin(GL\_TRIANGLE\_STRIP);

glColor3d(1, 0, 0);

glVertex3d(0, 0, 0);

glVertex3d(0, 1, 0);

glVertex3d(-2, 1, 0);

glVertex3d(-2, 1.5, 0);

glEnd();

glEnable(GL\_LINE\_STIPPLE);

glLineStipple(1, 12803);

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);

glBegin(GL\_TRIANGLE\_FAN);

glColor3d(0.2, 0, 1);

glVertex3d(4, 2, 0);

glVertex3d(2, 4, 0);

glVertex3d(1, 2, 0);

glVertex3d(3, 1.1, 0);

glEnd();

glDisable(GL\_LINE\_STIPPLE);

auxSwapBuffers();

}

***Завдання 5 Багатогранники***



void drowmytriangle() {

glBegin(GL\_TRIANGLES);

glVertex3d(0.5, 1, 0);

glVertex3d(0, 3, 0);

glVertex3d(-0.5, 1, 0);

glEnd();

}

void CALLBACK display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);

glColor3d(1, 0, 0);

drowmytriangle();

glPushMatrix();

glRotated(-90, 0, 0, 1);

drowmytriangle();

glRotated(-90, 0, 0, 1);

drowmytriangle();

glRotated(-90, 0, 0, 1);

drowmytriangle();

glPopMatrix();

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);

glColor3d(1, 1, 1);

glEnable(GL\_LINE\_SMOOTH);

glEnable(GL\_LINE\_STIPPLE);

glPointSize(2);

glLineWidth(3);

glLineStipple(1, 12803);

glPushMatrix();

glRotated(-45, 0, 0, 1);

drowmytriangle();

glRotated(-90, 0, 0, 1);

drowmytriangle();

glRotated(-90, 0, 0, 1);

drowmytriangle();

glRotated(-90, 0, 0, 1);

drowmytriangle();

glPopMatrix();

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);

glColor3d(1, 1, 0);

glDisable(GL\_LINE\_SMOOTH);

glDisable(GL\_LINE\_STIPPLE);

drowmytriangle();

glPushMatrix();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

drowmytriangle();

glRotated(-20, 0, 0, 1);

glPopMatrix();

auxSwapBuffers();

}