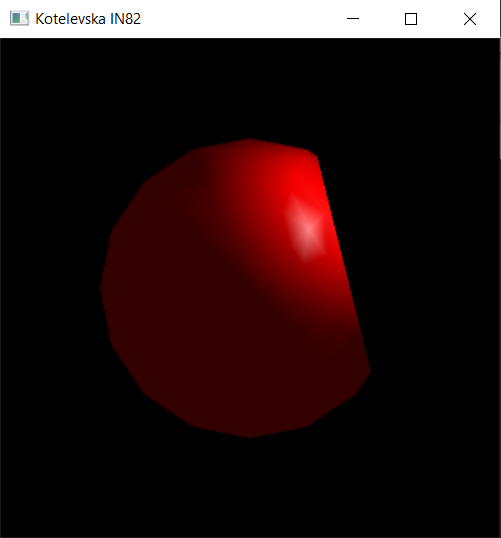
**Лабораторна робота №5**

**Котелевська Вероніка ІН-82**

**Завадння 1. Площини відсікання**



void CALLBACK display(void)

{

GLdouble equation[4] = { -1,-0.25,0,2 };

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

glEnable(GL\_CLIP\_PLANE0);

glClipPlane(GL\_CLIP\_PLANE0, equation);

glColor3d(1, 0, 0);

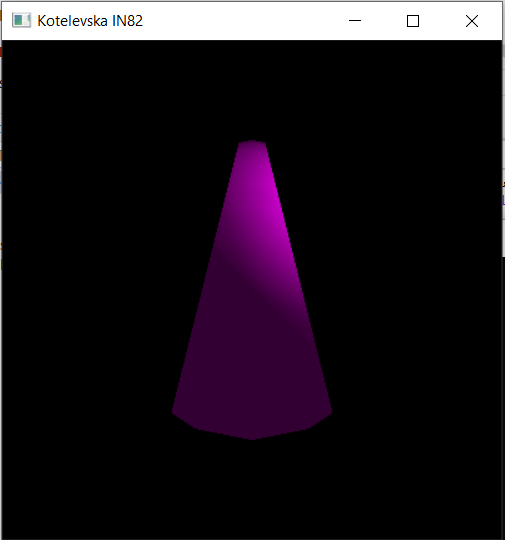
auxSolidSphere(3);

glDisable(GL\_CLIP\_PLANE0);

auxSwapBuffers();

}

**Завадння 2** **Вправа "Дві площини"**



void CALLBACK display(void)

{

GLdouble equation[4] = {-1, -0.25, 0, 1 };

GLdouble equation1[4] = { 1, -0.25, 0, 1 };

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

glEnable(GL\_CLIP\_PLANE0);

glClipPlane(GL\_CLIP\_PLANE0, equation);

glEnable(GL\_CLIP\_PLANE1);

glClipPlane(GL\_CLIP\_PLANE1, equation1);

glColor3d(1, 0, 1);

auxSolidSphere(3);

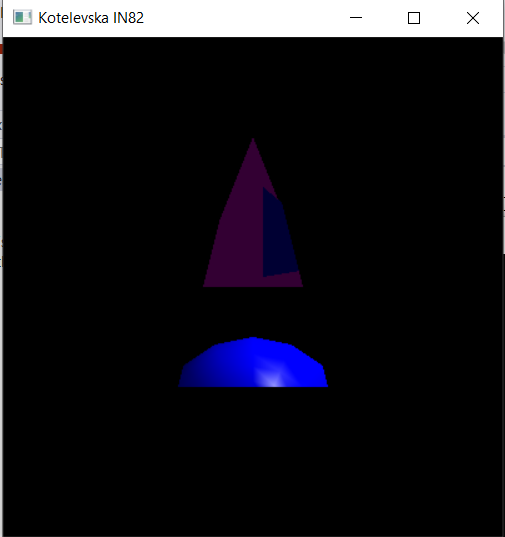
glDisable(GL\_CLIP\_PLANE0);

glDisable(GL\_CLIP\_PLANE1);

auxSwapBuffers();

}

**Завадння 3 Три площини**



GLdouble equation0[4] = {-1, -0.25, 0, 1 };

GLdouble equation1[4] = { 1, -0.25, 0, 1 };

GLdouble equation2[4] = { 0, 1, 0, 2 };

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

glEnable(GL\_CLIP\_PLANE0);

glClipPlane(GL\_CLIP\_PLANE0, equation0);

glEnable(GL\_CLIP\_PLANE1);

glClipPlane(GL\_CLIP\_PLANE1, equation1);

glEnable(GL\_CLIP\_PLANE2);

glClipPlane(GL\_CLIP\_PLANE2, equation2);

glColor3d(1, 0, 1);

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

glBegin(GL\_TRIANGLES);

glColor3d(1, 0, 1);

glVertex3d(-1.2, 0, 0);

glVertex3d(0, 3, 0);

glVertex3d(1.2, 0, 0);

glColor3d(0, 0, 1);

glVertex3d(0.2, 0.2, 1);

glVertex3d(0.2, 2, 1);

glVertex3d(2, 0.5, 1);

glEnd();

glTranslated(0, -3, 0);

auxSolidSphere(2);

glTranslated(0, 3, 0);

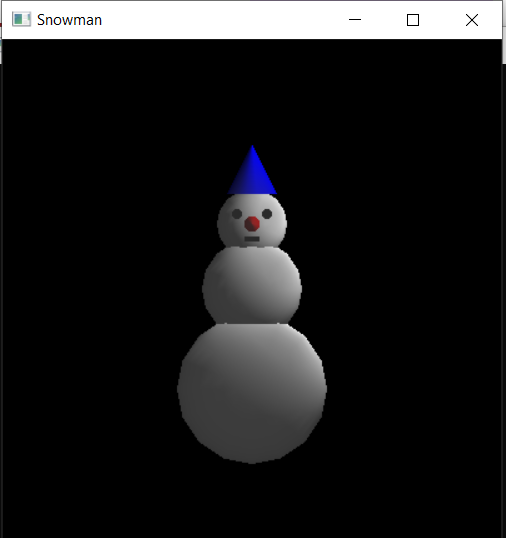
glDisable(GL\_CLIP\_PLANE0);

glDisable(GL\_CLIP\_PLANE1);

glDisable(GL\_CLIP\_PLANE2);

auxSwapBuffers();

**Завадння 4 Туман**



#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);

}

float density;

void CALLBACK Key\_UP(void)

{

density += (float)0.1;

glFogf(GL\_FOG\_DENSITY, density);

}

void CALLBACK Key\_DOWN(void)

{

density -= (float)0.1;

glFogf(GL\_FOG\_DENSITY, density);

}

void CALLBACK display(void)

{

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

glColor3d(1, 1, 1);

auxSolidSphere(1);

glPushMatrix();

glTranslated(0, -2, 0);

auxSolidSphere(1.5);

glPopMatrix();

glPushMatrix();

glTranslated(0, 1.3, 0);

auxSolidSphere(0.7);

glColor3d(1, 0, 0);

auxSolidCone(0.5, 1);

glColor3d(0, 0, 1);

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

glTranslated(0, 0, 0.6);

auxSolidCone(0.5, 1);

glColor3d(0, 0, 0);

glTranslated(0.3, -1, -0.4);

auxSolidSphere(0.1);

glTranslated(-0.6, 0, 0);

auxSolidSphere(0.1);

glPopMatrix();

glPushMatrix();

glTranslated(0, 1, 1);

auxSolidBox(0.3, 0.1, 0.1);

glPopMatrix();

auxSwapBuffers();

}

int main()

{

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

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

float fogcolor[4] = { 0.25,0.25,0.25,1 }; // колір туману

auxInitPosition(50, 10, 400, 400);

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

auxInitWindowA("Snowman");

auxIdleFunc(display);

auxReshapeFunc(resize);

auxKeyFunc(AUX\_UP, Key\_UP); // встановлюємо обробник

auxKeyFunc(AUX\_DOWN, Key\_DOWN); // стрілок вгору / вниз

glEnable(GL\_FOG); // дозволяємо туман

glGetFloatv(GL\_FOG\_DENSITY, &density); // отримуємо значення щільності

glFogfv(GL\_FOG\_COLOR, fogcolor); // встановлюємо колір туману

glEnable(GL\_ALPHA\_TEST);

glEnable(GL\_DEPTH\_TEST);

glEnable(GL\_COLOR\_MATERIAL);

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glEnable(GL\_BLEND);

glBlendFunc(GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA);

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

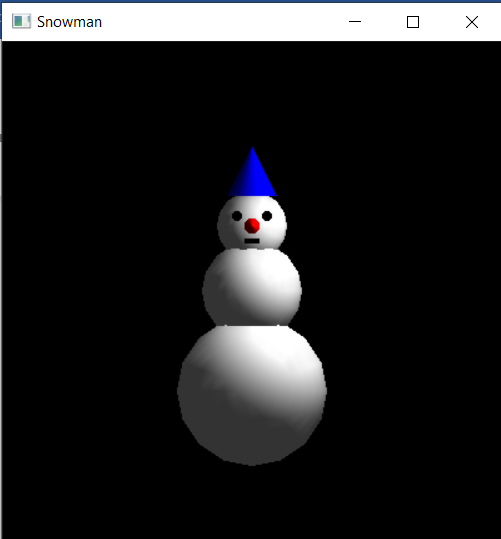
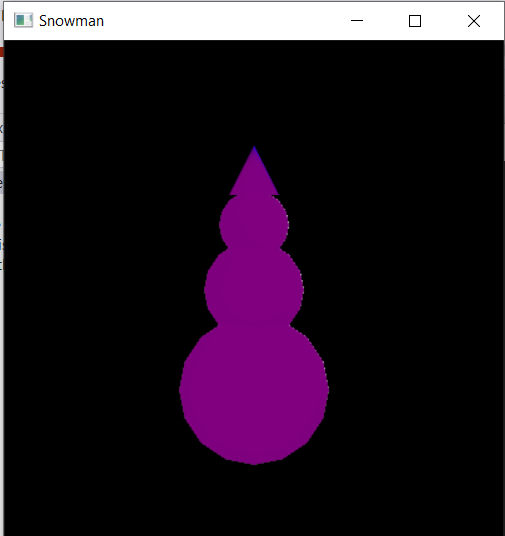
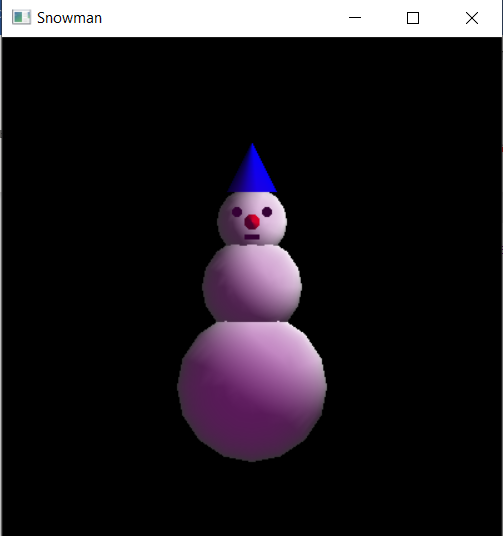
glLightfv(GL\_LIGHT0, GL\_SPOT\_DIRECTION, dir);

auxMainLoop(display);

return 0;

}

**Завадння 5 Вправа «Туман»**

int main()

{

…

float fogcolor[4] = { 0.5, 0, 0.5, 1 }; // колір туману

…

}