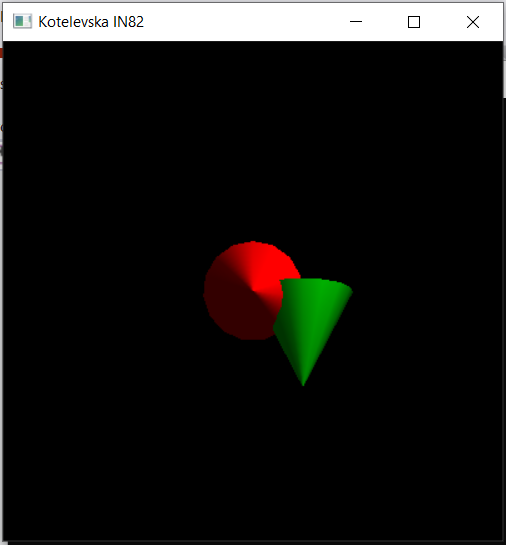
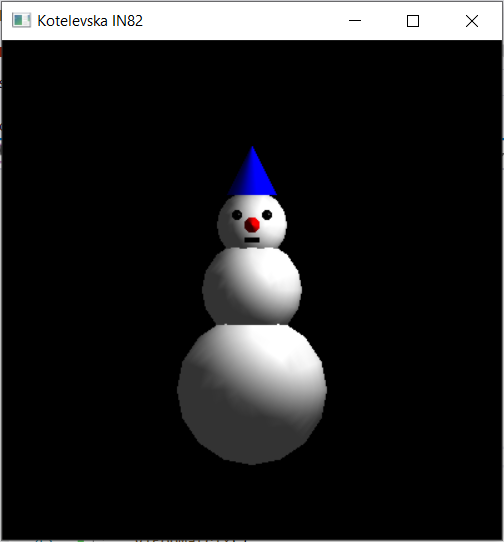
Лабораторна №2 Котелевська В ІН-82І



***Завдання 2. Сніговик (самостійно)***



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();

}

3 Завдання Анімація

void CALLBACK display(void)

{

static int time = 0;// мой код

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

glPushMatrix();

glTranslated(((double)time) / 1000.0, 0, 0);

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();

glPopMatrix();

time++;

auxSwapBuffers();

}

4 Завдання

#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)

{

static int time = 0;// мой код

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

glPushMatrix(); // зберігаємо поточні координати, тому що при виході

// з функції нам потрібно повернутися до абсолютних координатах

glRotated(time / 20, 0.0, 1.0, 0.0); // повертаємо координати

glLineWidth(5); // встановлюємо товщину лінії - п'ять

glColor3f(1, 0, 0); // встановлюємо поточний колір - червоний

glBegin(GL\_LINES); // малюємо червону вісь, тому що ми повернули координати,

glVertex3d(-0.3, 0, 0); // вісь буде повернута в абсолютних координатах,

glVertex3d(1.5, 0, 0); // і тому функція display викликається в циклі, і змінна

glEnd(); // time збільшується на одиницю при кожному виклику display, ми

// отримуємо анімацію - обертову вісь

glPushMatrix(); // тепер щодо повернених координат

// ми переходимо до нових координат

glRotated(2 \* time, 1, 0, 0); // повертаємо їх на кут 2 \* time навколо червоної осі

glTranslated(-0.3, 0, 0); // і зрушуємо на край осі

glColor3f(0, 0, 1); // встановлюємо синій колір

glPushMatrix(); // тепер ще раз переходимо до нових координат,

glRotated(90, 0, 1, 0); // щоб розгорнути тор на 90 градусів

glLineWidth(1);

auxWireTorus(0.2, 0.7);

glPopMatrix(); // намалювавши тор, ми повертаємося до попередніх координат

// початок цих координат лежить на кінці червоної осі

glLineWidth(7);

glColor3f(0, 1, 0);

glBegin(GL\_LINES); // тепер малюємо три зелених лінії

glVertex3d(0, 0, 0);

glVertex3d(0, 1, 0);

glVertex3d(0, 0, 0);

glVertex3d(0, -0.5, 1);

glVertex3d(0, 0, 0);

glVertex3d(0, -0.5, -1);

glEnd();

glPopMatrix(); // тепер повертаємося до координат, поверненим на кут time / 2

glPopMatrix(); // і переходимо до абсолютних координат

time++; // збільшуємо лічильник кадрів або час на одиницю

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);

}