**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**ІНСТИТУТ ПІСЛЯДИПЛОМНОЇ ОСВІТИ (II-га В.О.)**



**ЗВІТ**

**Про виконання лабораторної роботи №2**

*з предмету: «Засоби програмування комп’ютерної графіки»*

на тему:

**«Програмування тривимірної графіки за допомогою графічних бібліотек OpenGL»**

**Виконав:**

слухач групи ПЗС-21

Гринчук Т.А.

**Прийняла:**

доц. Левус Є.В.

ЛЬВІВ–2014

**Тема.** Програмування тривимірної графіки за допомогою графічних бібліотек OpenGL. **Мета.** Навчитись будувати тривимірні об‘єкти, покритих текстурою для подальшого їх обертання, та масштабуванням на основі застосування базових геометричних перетворень в OpenGL.

**Хід роботи**

**Умова завдання (варіант №1).** Намалювати «глобус України – кулю, покриту картою України та геодезичною сіткою. Реалізувати обертання глобуса навколо нахиленої вертикальної осі»

**Виконання роботи:**

Для того щоб ми зморли використати бібліотеки OpenGL, перш за все, потрібно підключити наступні заголовочні файли:

#include <gl\glut.h>

#include <gl\glu.h>

#include <gl\GLAux.h>

Ініціалізація бібліотек проводиться наступним чином:

//ініціалізація Open GL та створення вікна

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(Width, Height);

//параметрии початкової позиції

glutInitWindowPosition(100, 200);

//створення нового вікна

glutCreateWindow("Ukrainian Globe");

Задамо освітлення. Перший тип світла називається фоновим світлом. Фонове світло не має ніякого певного напрямку. Усі об'єкти у вашій сцені будуть освітлені фоновим світлом. Другий тип світла - дифузійне світло. Дифузійне світло створюється за допомогою вашого джерела світла й відбивається від поверхні об'єкта у вашій сцені. Будь-яка поверхня об'єкта, на яку падає прямо світло, буде дуже яскравої й області, куди світло падає під кутом, будуть темніше. Світло створюється так само як колір. Якщо перше число - 1.0f, а наступні два - 0.0f, ми одержимо яскраво червоне світло. Якщо третє число - 1.0f, і перші два - 0.0f, ми будемо мати яскраво синє світло. Останнє число - альфа значення. Ми залишимо його поки 1.0f.Тому в рядку нижче, ми задамо значення білого фонового світла половиною інтенсивності (0.5f). Оскільки всі числа - 0.5f, ми одержимо світло середньої яскравості між чорним (виключене світло) і білим (повна яскравість). Змішані в рівних значеннях червоний, синій і зелений дадуть відтінки від чорного (0.0f) до білого (1.0f). Без фонового світла плями, де немає дифузійного світла, будуть дуже темними.

//освітлення

// Масив фонового освітлення

GLfloat Amb[] = { 0.5f, 0.5f, 0.5f, 1.0f };

// Значення дифузорного освітлення

GLfloat Dif[] = { 1.0f, 1.0f, 1.0f, 1.0f };

//Фонове Світло

glLightfv(GL\_LIGHT1, GL\_AMBIENT, Amb);

//Дифузне світло

glLightfv(GL\_LIGHT1, GL\_DIFFUSE, Dif);

glLightf(GL\_LIGHT1, GL\_SPOT\_EXPONENT, 1);

glLightf(GL\_LIGHT1, GL\_SPOT\_CUTOFF, 180);

glEnable(GL\_LIGHT1);

glEnable(GL\_LIGHTING);

Завантажимо картинку з bmp-файлу:

unsigned texture;

AUX\_RGBImageRec \*texture1 = auxDIBImageLoadA("ukr128.bmp");

glGenTextures(1, &texture);

glBindTexture(GL\_TEXTURE\_2D, texture);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR\_MIPMAP\_NEAREST);

gluBuild2DMipmaps(GL\_TEXTURE\_2D, 3, texture1->sizeX, texture1->sizeY, GL\_RGB, GL\_UNSIGNED\_BYTE, texture1->data);

Намалюємо глобус (сферу) та зафарбуємо його текстурою:

glEnable(GL\_COLOR\_MATERIAL);

glEnable(GL\_NORMALIZE);

//включити розмальовування текстурою

glEnable(GL\_TEXTURE\_2D);

//розмальовування текстурою

gluQuadricTexture(quadricObj, GL\_TRUE);

//малюємо сферу з радіусом 200 пікселів

gluSphere(quadricObj, 200.0f, 100, 100);

//виключити розмальовування текстурою

gluQuadricTexture(quadricObj, GL\_FALSE);

glPopMatrix();

gluDeleteQuadric(quadricObj);

**Текст програми (С++)**

#include "stdafx.h"

#include <stdlib.h>

#include <gl\glut.h>

#include <gl\glu.h>

#include <gl\GLAux.h>

//константи Ширини та Висоти екрану

const float Width = 500.0;

const float Height = 500.0;

//кут повороту глобуса

float rotate = 0.0f;

void show()

{

//освітлення

// Масив фонового освітлення

GLfloat Amb[] = { 0.5f, 0.5f, 0.5f, 1.0f };

// Значення дифузорного освітлення

GLfloat Dif[] = { 1.0f, 1.0f, 1.0f, 1.0f };

//Фонове Світло

glLightfv(GL\_LIGHT1, GL\_AMBIENT, Amb);

//Дифузне світло

glLightfv(GL\_LIGHT1, GL\_DIFFUSE, Dif);

glLightf(GL\_LIGHT1, GL\_SPOT\_EXPONENT, 1);

glLightf(GL\_LIGHT1, GL\_SPOT\_CUTOFF, 180);

glEnable(GL\_LIGHT1);

glEnable(GL\_LIGHTING);

//Очистка екрана

glClear(GL\_COLOR\_BUFFER\_BIT);

//Встановлення кольору малювання - білий з сірим відтінком

glColor3f(0.9, 0.9, 0.9);

//Створення об'єкта GLUquadricObj,

//потрібного для запам'ятовування координат, щоб їх наповнити текстурою

GLUquadricObj \*quadricObj;

quadricObj = gluNewQuadric();

if (quadricObj)

{

glPushMatrix();

//поворот навколо осі

glRotatef(rotate, 1.0, 1.0, 0.0);

glEnable(GL\_COLOR\_MATERIAL);

glEnable(GL\_NORMALIZE);

//включити розмальовування текстурою

glEnable(GL\_TEXTURE\_2D);

//розмальовування текстурою

gluQuadricTexture(quadricObj, GL\_TRUE);

//малюємо сферу з радіусом 200 пікселів

gluSphere(quadricObj, 200.0f, 100, 100);

//виключити розмальовування текстурою

gluQuadricTexture(quadricObj, GL\_FALSE);

glPopMatrix();

gluDeleteQuadric(quadricObj);

}

//заміна буферів - фактичний вивід зображення на екран

glutSwapBuffers();

}

void Timer(int)

{

rotate += 1.0; //Збільшуємо кут повороту

//Перемальовування поточного вікна

glutPostRedisplay();

//встановлення фунції таймера з інтервалом 200 мілісекунк

glutTimerFunc(200, Timer, 0);

}

//Ініціалізація OpenGL

void initOpenGL()

{

//Колір екрану

glClearColor(0.5f, 0.5f, 0.5f, 1.0f);

//Завантаження текстури з файлу

unsigned texture;

AUX\_RGBImageRec \*texture1 = auxDIBImageLoadA("ukr128.bmp");

glGenTextures(1, &texture);

glBindTexture(GL\_TEXTURE\_2D, texture);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR\_MIPMAP\_NEAREST);

gluBuild2DMipmaps(GL\_TEXTURE\_2D, 3, texture1->sizeX, texture1->sizeY, GL\_RGB, GL\_UNSIGNED\_BYTE, texture1->data);

glEnable(GL\_TEXTURE\_2D);

glClearColor(1.0, 1.0, 1.0, 1.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

//отримуємо ортогональні перетворення свого простору на площину екрана

glOrtho(-Width / 2, Width / 2, -Height / 2, Height / 2, -Width / 2, Width / 2);

glMatrixMode(GL\_MODELVIEW);

}

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

{

//ініціалізація Open GL та створення вікна

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(Width, Height);

//параметрии початкової позиції

glutInitWindowPosition(100, 200);

//створення нового вікна

glutCreateWindow("Ukrainian Globe");

//Визначення функції малювання

glutDisplayFunc(show);

//Визначення функції таймера

glutTimerFunc(200, Timer, 0);

//ініціалізація Open G

initOpenGL();

//запуск процедури опрацювання подій glut

//входження в безкінечний цикл перемальовування

glutMainLoop();

return 0;

}

**Результат виконання програми**

Запустимо програму на виконання (рис. 1):

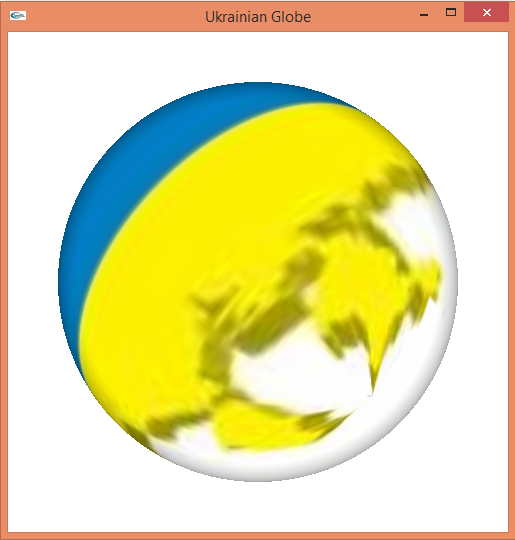


Рис. 1. Вигляд головного вікна

**Висновки**

Виконавши дану лабораторну роботу я навчився будувати 3-вимірні об‘єкти, розміщувати їх в канві, розфарбовувати, накладати текстури. Навчився запрограмовавати анімацію об‘єктів.