**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут **ІКНІ**

Кафедра **ПЗ**



**ЗВІТ**

До лабораторної роботи №5

**На тему:** *«Програмування тривимірної графіки за допомогою графічних бібліотек OpenGL.»*

**З дисципліни:** *«Програмування мультимедійних систем»*

**Виконав:**

студ. гр. ПІ-31

*Портах Б.В.*

**Прийняв:**

Гасько Р.Т.

Львів – 2017

**Тема роботи:** Програмування тривимірної графіки за допомогою графічних бібліотек OpenGL.

**Мета роботи:** Навчитися будувати тривимірні об’єкти, покриті текстурою, з подальшим їхнім обертанням та масштабуванням на основі застосування базових геометричних перетворень OpenGL.

**Теоретичні відомості**

Для задання різних перетворень об’єктів сцени в OpenGL використовують операції над матрицями, при цьому розрізняють кілька типів матриць, зокрема вигляду і проекції. Перша матриця визначає перетворення об’єкта у світових координатах, такі як паралельний перенос, зміна масштабу і поворот. Матриця проекцій задає, як будуть проектуватися тривимірні об’єкти на площину екрана (у віконні координати). Для того, щоб вибрати матрицю для опрацювання, використовують команду **glMatrixMode(mode),** виклик якої зі значенням параметра**GL\_MODELVIEW** чи **GL\_PROJECTION**вмикає режими роботи з матрицею виду чи проекцій відповідно.

**Робота з матрицями проекції та вигляду**

Отож, щоб працювати з матрицею проекції, треба активувати відповідний режим командою**glMatrixMode(GL\_PROJECTION)**. Параметри ортогональної проекції вибирають за допомогою функції **glOrtho(…)**, для отримання перспективної проекції сцени використовують функцію**gluPerspective(…)** або **glFrustum(…)**. Якщо команда проектування не задана явно, OpenGL виконує ортогональну проекцію сцени за замовчуванням. Параметри спостереження задають за допомогою функції **gluLookAt(…)**.

Перед роботою з об’єктами необхідно задати режим роботи з матрицею вигляду, виконавши команду **glMatrixMode(GL\_MODELVIEW)**. Прямокутну область виведення результатів визначають функцією **glViewport(…)**.

**Завдання до лабораторної роботи**

Завдання повинні бути реалізовані у вікні з наявними елементами керування для запуску виконання окремих дій або етапів роботи. Для оформлення зовнішнього вигляду тривимірних об’єктів (окрім однотонного зафарбування) необхідно застосувати накладання текстури. Для формування текстур матеріалів (граніту, мармуру тощо) можна скористатися текстурним заповненням фігур у програмі Word. За замовчуванням сцена має бути освітлена одним джерелом світла. У всіх важливих місцях програми необхідно давати коментарі з описом призначення поданого далі фрагмента коду.

*Варіант 7:*

Намалювати вежу, основа якої – циліндр, покритий текстурою мармуру, а дах – конус із бічною поверхнею зеленого кольору. Реалізувати обертання об’єкта навколо горизонтальної осі, що проходить через центр основи. Додати ефект туману.

**Код програми**

Назва файлу: **app.js**

init();

function init(){

const width = window.innerWidth;

const height = window.innerHeight;

const canvasEl = document.querySelector('#canvas');

canvasEl.setAttribute('width', width);

canvasEl.setAttribute('height', height);

const renderer = new THREE.WebGLRenderer({

canvas: canvasEl

});

renderer.setClearColor(0xdddddd);

const scene = new THREE.Scene();

scene.fog = new THREE.Fog(0xffffff, 400, 600);

const camera = new THREE.PerspectiveCamera(45, width / height, 0.1, 800);

camera.position.set(0, 0, 500);

const light = new THREE.AmbientLight(0xffffff);

scene.add(light);

var meshes = [];

const CylinderGeometry = new THREE.CylinderGeometry( 30, 30, 210, 30 );

const CylinderMaterial = new THREE.MeshPhongMaterial({ transparent: false, map: THREE.ImageUtils.loadTexture('images/marmur.jpg') });

const cylinder = new THREE.Mesh( CylinderGeometry, CylinderMaterial );

meshes.push(cylinder);

var ConeGeometry = new THREE.ConeGeometry( 32, 80, 50 );

var ConeMaterial = new THREE.MeshBasicMaterial( {color: 0x019c01, vertexColors: THREE.FaceColors } );

var cone = new THREE.Mesh( ConeGeometry, ConeMaterial );

meshes.push(cone);

cone.position.set(0, 145, 0);

let group = new THREE.Object3D();

group.add (cylinder);

group.add (cone);

scene.add(group);

loop();

function loop() {

group.rotation.x += Math.PI / 100;

renderer.render(scene, camera);

requestAnimationFrame(() => { loop(); })

}

};

**Результати виконання програми**

****

*Рис.1 Вежа у вигляді циліндра з дахом конусоподібної форми*

**Контрольні питання**

1. Які спільні та відмінні параметри відображення мають плоскі та об’ємні графічні фігури?

Вся різниця у об’ємі та осі Z.

1. Поясніть загальне призначення основних геометричних перетворень графічних об’єктів.

Вони необхідні для забезпечення динаміки зображенню.

1. Як використовують в OpenGL функції **glTranslate**, **glRotate** і **glScale**?

Їх використовують для базових перетворень об’єктів.

1. Опишіть загальну схему малювання фігури в OpenGL не в початку координат.

Для зображення об’єктів в певній точці, центр координат переміщують в цю точку, тоді відмальовують об’єкт.

1. Як повернути об’ємну фігуру навколо довільної точки у просторі?

Слід перемістити фігуру у дану точку, повернути її, та перенести у відповідну точку.

1. Як вибрати необхідну матрицю для роботи зі сценою чи її об’єктами?

Все залежить від вимог розробника, куди, як і під яким кутом має дивитися камера.

1. Які типи проекцій можна використовувати в OpenGL?

Projection, model view.

1. Опишіть загальну схему визначення параметрів проекції та спостереження.

Параметри ортогональної проекції вибирають за допомогою функції **glOrtho(…)**, для отримання перспективної проекції сцени використовують функцію**gluPerspective(…)** або **glFrustum(…)**.

1. Запишіть загальний вигляд функції відображення сцени у випадку зміни розмірів вікна.

**void resize(int width,int height)**

**{**

**glViewport(0,0,width,height);**

**glMatrixMode(GL\_PROJECTION);**

**glLoadIdentity();**

**glOrtho(…);**

**gluLookAt(…);**

**glMatrixMode(GL\_MODELVIEW);**

**}**

1. Які тривимірні об’єкти можна створити в OpenGL за допомогою функцій додаткових бібліотек GLU, GLUT і GLAUX?

Сфера, циліндр, диск, куб тощо.

1. Що таке текстура і для чого її використовують?

Текстура це зовнішня обгортка об’єкта.

1. Опишіть загальну схему накладання текстури на тривимірний об’єкт в OpenGL.

Завантажується текстура, згодом вона задається для малювання, визначається її розміщення на об’єкті.

1. Перелічіть функції, необхідні для реалізації процесу накладання текстури в OpenGL.

glGenTextures, glBindTexture, glTexImage2D, glTexParameter, glTexEnv.

1. Як пов’язати координати текстури з об’єктом і навіщо це потрібно?

За допомгою функцій **glTexCoord2d** або **glTexGen**.

1. Як пов’язати координати текстури з об’єктом і навіщо це потрібно?

За допомгою функцій glTexCoord2d або glTexGen.

1. Яку роль у тривимірних сценах відіграє освітлення?

Воно додає ефект реалістичності до зображення.

1. Опишіть процес увімкнення освітлення та визначення його глобальних параметрів.

* Увімкнення освітлення.
* Увімкнення джерела світла.
* Задання параметрів джерела світла.

1. Що необхідно зробити для надання тривимірним об’єктам ефекту реалістичності?

Потрібно увімкнути перевірку на об’єм.

1. Як увімкнути кілька джерел освітлення з різними властивостями?

Для цього використовуються різні джерела освітлення (LIGH0, LIGHT1..7).

**Висновок**

У процесі виконання цієї лабораторної роботи я навчився будувати тривимірні об’єкти, покриті текстурою, з подальшим їхнім обертанням та масштабуванням на основі застосування базових геометричних перетворень WebGL.

**Посилання на GitHub**

https://github.com/bportakh/PMS