

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра інформатики та програмної інженерії

Звіт

З лабораторної роботи № 4 з дисципліни
«Комп'ютерна графіка та обробка зображень»

Тема: «Двовимірні текстури текстурні координати текстурна
матриця»

Виконав(ла)

ІП-15 Мешков Андрій

(шифр, прізвище, ім'я, по батькові)

Перевірів

Щебланін Ю. М.

(шифр, прізвище, ім'я, по батькові)

Київ 2024

ЗМІСТ

ВСТУП.....	3
ХІД РОБОТИ.....	4
Завдання 3.1	4
Завдання 3.2	4
Завдання 3.3	6
Завдання 3.4	9
ВИСНОВКИ.....	12
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	13

ВСТУП

У рамках даної лабораторної роботи було використано бібліотеку Three.js, яка є потужним інструментом для створення та відображення тривимірної графіки в веб-браузері. Метою цієї роботи було ознайомлення з методами роботи з двовимірними текстурами, використання текстурних координат, а також застосування текстурної матриці для трансформацій текстур.

Three.js дозволяє легко створювати і маніпулювати тривимірними об'єктами, використовуючи такі текстури. Використання текстур дозволяє значно покращити візуальне сприйняття об'єктів, додаючи деталізацію і реалізм. У цій лабораторній роботі ми створювали текстури програмно, що дозволяє генерувати унікальні візерунки без необхідності завантаження зовнішніх файлів.

Основні етапи роботи включали:

1. Створення сцени, камери і рендерера.
2. Створення геометрії об'єктів.
3. Генерація текстур з використанням HTML5 canvas.
4. Застосування текстур до об'єктів.
5. Анімування сцени для динамічного відображення результатів.

ХІД РОБОТИ

Завдання 3.1

Підготувати та зберегти в окремій папці зображення текстури у форматі BMP з розмірами 512x512 пікселів та глибиною кольору 24 біти. Деякі графічні редактори утворюють навколо зображення рамку, тому в них потрібно вказувати розміри на одиницю більші.

Підготовано текстуру трави.



Рисунок 3.1 — Текстура трави

Завдання 3.2

Доповнити процедуру командами для виведення текстури на верхній та нижній гранях куба.

Лістинг програмного коду:

Index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Lab 2</title>
    <style>
      * {
        box-sizing: border-box;
      }
      body {
        margin: 0;
        padding: 0;
      }
    </style>
  </head>
  <body>
    <script type="module" src="./task2.js"></script>
```

```
</body>
</html>
```

Task2.js

```
import * as THREE from 'three';
import { OrbitControls } from 'three/addons/controls/OrbitControls.js';
import { GLTFLoader } from 'three/addons/loaders/GLTFLoader.js';

const scene = new THREE.Scene();

const camera = new THREE.PerspectiveCamera(50, window.innerWidth / window.innerHeight, 0.1, 1000);
camera.position.z = 3;

const renderer = new THREE.WebGLRenderer();
renderer.setSize(window.innerWidth, window.innerHeight);
document.body.appendChild(renderer.domElement);

const controls = new OrbitControls(camera, renderer.domElement);

const loader = new GLTFLoader();
const geometry = new THREE.BoxGeometry();

const textureLoader = new THREE.TextureLoader();
const grassTexture = textureLoader.load('/grass.bmp');

const materials = [
  new THREE.MeshBasicMaterial({ color: 0x00ff00 }),
  new THREE.MeshBasicMaterial({ color: 0xffff00 }),
  new THREE.MeshBasicMaterial({ map: grassTexture }, // Apply grass texture
  new THREE.MeshBasicMaterial({ map: grassTexture }, // Apply grass texture
  new THREE.MeshBasicMaterial({ color: 0xff0000 }),
  new THREE.MeshBasicMaterial({ color: 0x0000ff })
];

const cube = new THREE.Mesh(geometry, materials);
scene.add(cube);

function animate() {
  requestAnimationFrame(animate);
  renderer.render(scene, camera);
}

animate();
```

Результати виконання програми:

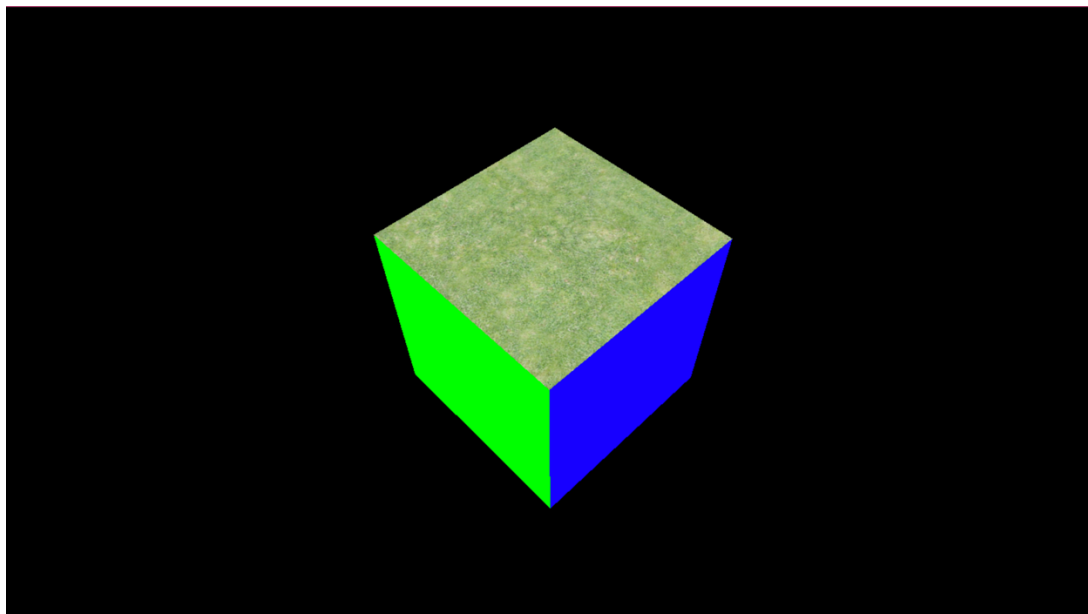


Рисунок 3.2 — Результати виконання програми

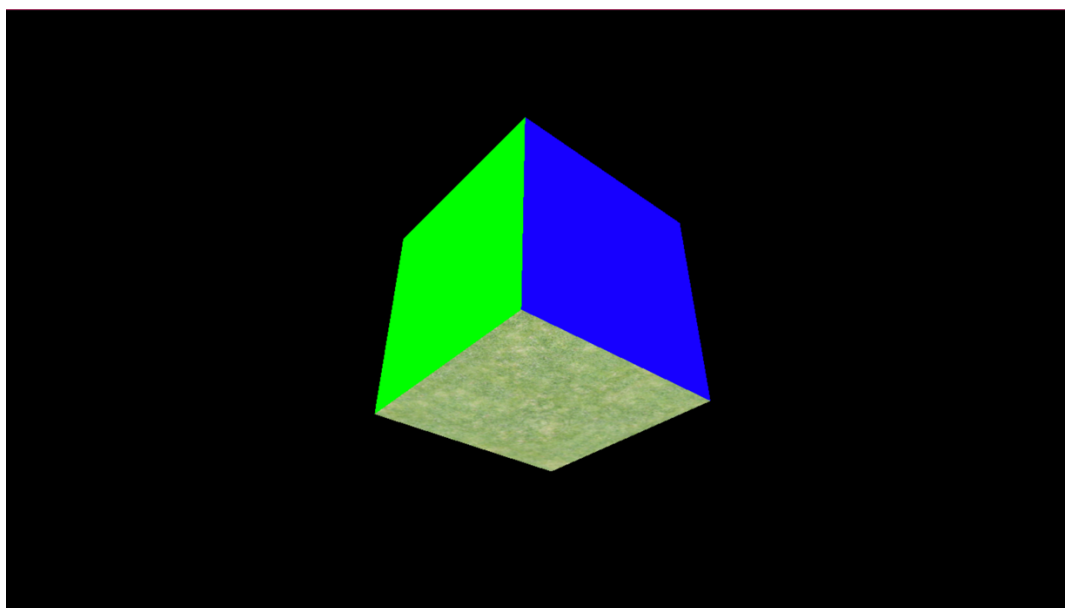


Рисунок 3.3 — Результати виконання програми

Завдання 3.3

Створити зображення шахових клітинок на гранях куба

Варіант 7 – Палісандр і Махагон



Рисунок 3.4 — Палісандр, 1.bmp

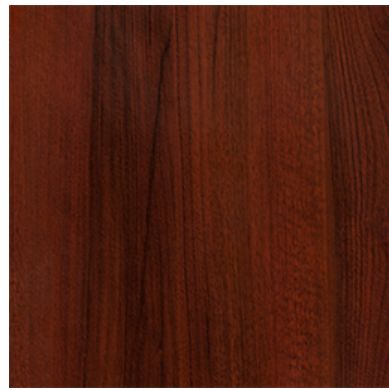


Рисунок 3.5 — Махагон, 2.bmp



Рисунок 3.6 — Шахівниця, 2x2.bmp

Лістинг програмного коду:

Task3.js

```
import * as THREE from 'three';  
import { OrbitControls } from 'three/addons/controls/OrbitControls.js';  
import { GLTFLoader } from 'three/addons/loaders/GLTFLoader.js';  
  
const scene = new THREE.Scene();
```

```

const camera = new THREE.PerspectiveCamera(50, window.innerWidth / window.innerHeight, 0.1,
1000);
camera.position.z = 3;

const renderer = new THREE.WebGLRenderer();
renderer.setSize(window.innerWidth, window.innerHeight);
document.body.appendChild(renderer.domElement);

const loader = new GLTFLoader();
const geometry = new THREE.BoxGeometry();

const textureLoader = new THREE.TextureLoader();
const texture = textureLoader.load('/2x2.bmp');

texture.wrapS = THREE.RepeatWrapping;
texture.wrapT = THREE.RepeatWrapping;

const repeatFactor = 4;
texture.repeat.set(repeatFactor, repeatFactor);

const material = new THREE.MeshBasicMaterial({ map: texture })

const materials = [
    material,
    material,
    material,
    material,
    material,
    material,
];

const cube = new THREE.Mesh(geometry, materials);
scene.add(cube);

const controls = new OrbitControls( camera, renderer.domElement );
controls.update();

function animate() {
    requestAnimationFrame(animate);
    renderer.render(scene, camera);
}

animate();

```


Результати виконання програми:

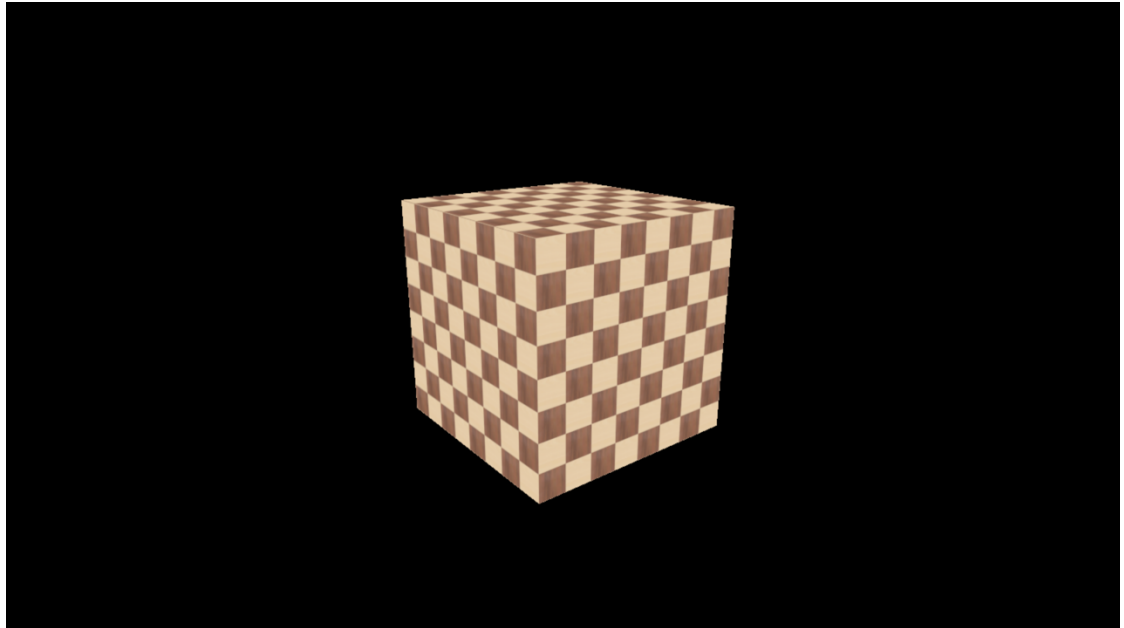


Рисунок 3.7 — Результати виконання програми, repeatFactor = 4

Завдання 3.4

Створити власний варіант зображення з візерунком.

Лістинг програмного коду:

Task4.js

```
import * as THREE from 'three';
import { OrbitControls } from 'three/addons/controls/OrbitControls.js';
import { GLTFLoader } from 'three/addons/loaders/GLTFLoader.js';

const scene = new THREE.Scene();

const camera = new THREE.PerspectiveCamera(50, window.innerWidth / window.innerHeight, 0.1, 1000);
camera.position.z = 3;

const renderer = new THREE.WebGLRenderer();
renderer.setSize(window.innerWidth, window.innerHeight);
document.body.appendChild(renderer.domElement);

const geometry = new THREE.BoxGeometry();

// Create a canvas and draw a pattern on it
const canvas = document.createElement('canvas');
canvas.width = 256;
canvas.height = 256;
const context = canvas.getContext('2d');

// Clear the canvas
```

```

context.clearRect(0, 0, canvas.width, canvas.height);

// Draw a circular pattern
context.fillStyle = 'green';
context.fillRect(0, 0, canvas.width, canvas.height);

context.strokeStyle = 'yellow';
context.lineWidth = 2;

for (let i = 0; i < 50; i++) {
  const radius = i * 4;
  context.beginPath();
  for (let angle = 0; angle < 2 * Math.PI; angle += Math.PI / 180) {
    const x = canvas.width / 2 + radius * Math.cos(angle);
    const y = canvas.height / 2 + radius * Math.sin(angle);
    context.lineTo(x, y);
  }
  context.closePath();
  context.stroke();
}

// Create a texture from the canvas
const texture = new THREE.CanvasTexture(canvas);

const material = new THREE.MeshBasicMaterial({ map: texture });

const materials = [
  material,
  material,
  material,
  material,
  material,
  material,
];

const cube = new THREE.Mesh(geometry, materials);
scene.add(cube);

const controls = new OrbitControls(camera, renderer.domElement);
controls.update();

function animate() {
  requestAnimationFrame(animate);
  renderer.render(scene, camera);
}

animate();

```

Результати виконання програми:

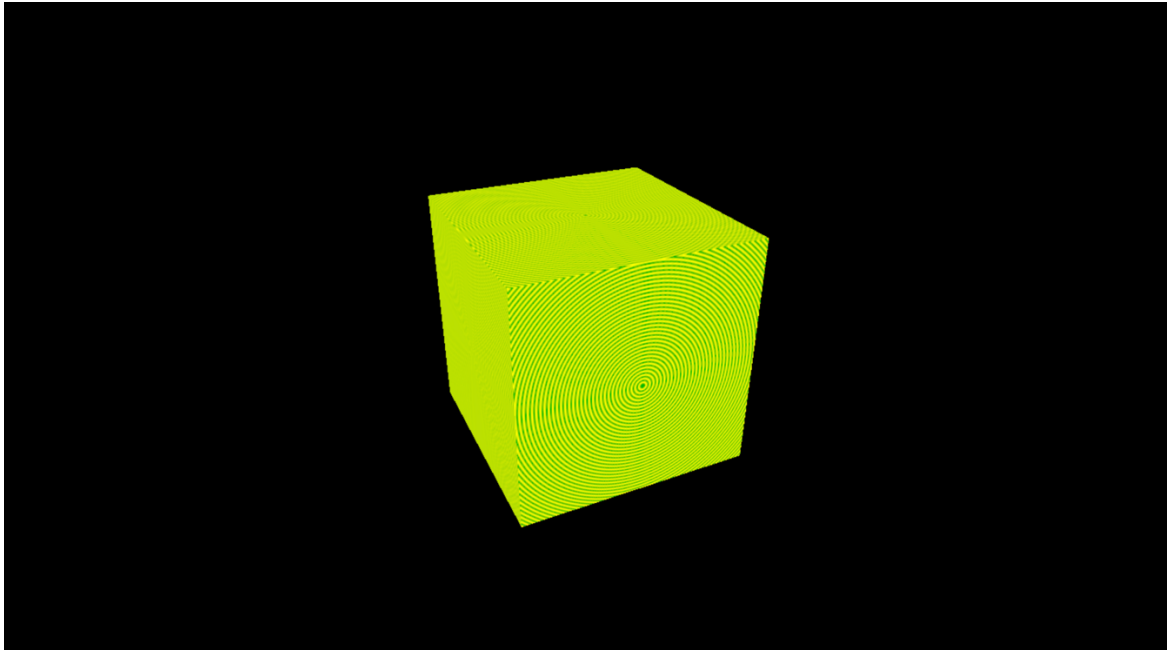


Рисунок 3.7 — Результати виконання програми

ВИСНОВКИ

У результаті виконання лабораторної роботи було досягнуто наступних результатів:

1. Ознайомлення з основними концепціями та методами роботи з текстурами у бібліотеці Three.js.
2. Практичне застосування текстурних координат для правильного відображення текстур на тривимірних об'єктах.
3. Використання HTML5 canvas для програмного створення унікальних текстурних візерунків.
4. Застосування текстурної матриці для трансформацій текстур, що дозволяє гнучко маніпулювати їх відображенням.

Використання Three.js для роботи з текстурами значно полегшує процес створення візуально привабливих тривимірних сцен, роблячи їх більш реалістичними та деталізованими. Отримані знання та навички можуть бути застосовані в майбутніх проектах з комп'ютерної графіки та обробки зображень, а також у розробці веб-додатків, що вимагають інтерактивної 3D-графіки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Three.js [Електронний ресурс] — Режим доступу: <https://threejs.org/docs/>