МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря Сікорського»

Навчально-науковий інститут атомної та теплової енергетики

Кафедра ЦТЕ

ЗВІТ

з графічно-розрахункової роботи

з дисципліни: «Візуалізація графічної та геометричної інформації»

**Виконав:**

студент групи ТР-23мп

Фурманчук М.В.

**Перевірив:**

Демчишин А.А.

Київ 2022

**Постановка задачі**

**Варіант:** 25 – непарний, реалізувати масштабування.

**Мета**: навчитися працювати з текстурами в WebGL, відобразити текстуру на поверхні та реалізувати її масштабування.

**Вимоги:**

• Нанести текстуру на поверхню з практичного завдання №2.

• Реалізувати масштабування текстури (координати текстури) навколо визначеної користувачем точки.

• Реалізувати переміщення точки вздовж простору поверхні (u, v) за допомогою клавіатури. Клавіші a та d переміщують точку вздовж параметра u, а клавіші w та s переміщують точку вздовж параметра v.

**Теоретичні відомості**

**Текстура** ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *Texture mapping*) — це спосіб надання поверхні 3D деталі — [полігону](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%96%D0%B3%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F): кольору, фактури, блиску, матовості та інших фізичних властивостей (для імітації найчастіше якогось природного матеріалу, наприклад: паперу, дерева, каменю, металу тощо).

Якість поверхні текстури визначається [текселями](https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D0%B5%D0%BB_(%D0%B3%D1%80%D0%B0%D1%84%D1%96%D0%BA%D0%B0)) — кількістю пікселів на мінімальну одиницю текстури. Оскільки сама по собі текстура є зображенням, роздільність текстури і її формат відіграють велику роль, яка згодом позначається на загальному враженні від якості графіки у [3D](https://uk.wikipedia.org/wiki/3D)-додатку.

Карта текстури застосовується для утворення певного параметру візуального відображення на поверхні заданої форми. Цей процес нагадує застосування візерунчастого паперу на звичайній білій коробці. Кожній вершині в 3D моделі присвоюється координати текстури (яка у 2D відома, як *U\_V* координата). Місця відбору проб зображення згодом інтерполюється по поверхні моделі з отриманням візуального результату.

**UV mapping** — процес в [3D моделюванні](https://uk.wikipedia.org/wiki/3D_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), який полягає в накладанні двовимірного зображення на тривимірну модель. Літерами U і V позначають осі координат площини розгортки, оскільки літери X, Y і Z використовуються для позначення просторових координат.

UV mapping (UV-розгортка) — відповідність між координатами на поверхні тривимірного об'єкту (X, Y, Z) і координатами на текстурі (U, V). Значення U і V зазвичай змінюються від 0 до 1, тобто мають бути нормалізовані.

Для того, щоб реалізувати масштабування навколо довільної точки, потрібно створити матрицю переміщення (T), яка переміщує об’єкт з початкової точки в точку масштабування, та матрицю масштабування (R), яка виконує масштабування об’єкта навколо початкової точки. Тепер для масштабування навколо довільної точки спершу необхідно перемістити точку на місце початкової точки за допомогою інвертування матриці переміщення (T), записаної як T\_1. Потім, ми масштабуємо об’єкт відповідно до початкової точки, за допомогою матриці масштабування (R), а потім застосовуємо матрицю переміщення (T) для переміщення точки масштабування до свого вихідного положення.

**Опис деталей реалізації**

Для відображення текстури на поверхні було завантажено зображення за посиланням з інтернету використовуючи функції JavaScript. Для того щоб завантажити текстуру за посиланням, необхідно задати значення «anonymous» атрибуту crossOrigin.

Налаштування параметрів текстури було встановлено на такі значення:

* TEXTURE\_WRAP\_S:CLAMP\_TO\_EDGE;
* TEXTURE\_WRAP\_T:CLAMP\_TO\_EDGE;
* TEXTURE\_MIN\_FILTER — LINEAR;
* TEXTURE\_MAG\_FILTER — LINEAR.

Обрахунок текстурних координат відповідно кожної вершини реалізовано в основному циклі і здійснюється за допомогою нормалізації параметрів u та v, за допомогою яких будується поверхня. Точки текстурних координат зберігаються у масиві *texCoordList*.

Також, було створено функцію *getPointLocation()* яка обчислює координати точки для відображення її на поверхні.

Cтворенні додаткові змінні які містять значення масштабування (*ScaleValue*) та два параметри які зберігають координати точки відносно якої відбувається масштабування (*pointLocationI*, *pointLocationJ*).

У шейдері було розроблено три функції Scale() – отримує як аргумент значення масштабування, та повертає матрицю масштабування, Translate() – отримує на вхід значення координати точки масштабування, та повертає матрицю переміщення та функція GetTextCoord() що повертає значення масштабованих координат.

**Інструкція користувача**

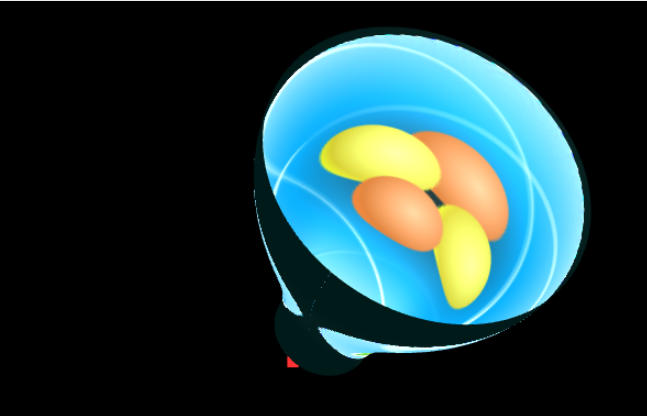
Для взаємодії з веб-додатком реалізовано інтерфейс який включає в себе наступні клавіші та поля:

• (a та d) – переміщення точки масштабування вздовж параметра u;

• (w та s) – переміщення точки масштабування вздовж параметра v;

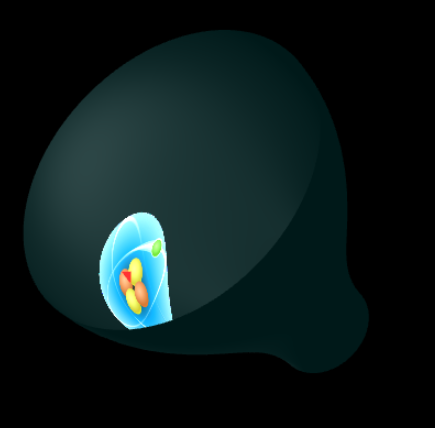
• (+ та -) – зміна значення масштабування.

На рисунку 1 показано початкове положення точки та текстури



*Рисунок 1. Початкове положення*

На рисунку 2 відображено зміну координат точки за допомогою визначених клавіш, та масштабування текстури.



*Рисунок 2. Зміна параметрів*

**Вихідний код**

Функція що відповідає за завантаження текстури :

function createTexture(){

    let texture = gl.createTexture();

    gl.bindTexture(gl.TEXTURE\_2D, texture);

    gl.texParameteri(gl.TEXTURE\_2D, gl.TEXTURE\_WRAP\_S, gl.CLAMP\_TO\_EDGE);

    gl.texParameteri(gl.TEXTURE\_2D, gl.TEXTURE\_WRAP\_T, gl.CLAMP\_TO\_EDGE);

    gl.texParameteri(gl.TEXTURE\_2D, gl.TEXTURE\_MIN\_FILTER, gl.LINEAR);

    gl.texParameteri(gl.TEXTURE\_2D, gl.TEXTURE\_MAG\_FILTER, gl.LINEAR);

    gl.texImage2D(gl.TEXTURE\_2D, 0, gl.RGBA,1,1,0,gl.RGBA,gl.UNSIGNED\_BYTE,

                  new Uint8Array([255,255,255,255]));

    let img = new Image();

    img.crossOrigin = "Anonymous";

    img.src = 'https://upload.wikimedia.org/wikipedia/commons/thumb/3/37/Sciences\_exactes.svg/256px-Sciences\_exactes.svg.png';

    img.addEventListener('load', function() {

        gl.bindTexture(gl.TEXTURE\_2D, texture);

        gl.texImage2D(gl.TEXTURE\_2D, 0, gl.RGBA,gl.RGBA,gl.UNSIGNED\_BYTE, img);

        draw();

    });

}

Функція що обчислює координати точки для відображення на поверхні:

function getPointLocation(){

    let pointList = [];

    let x,y,z;

    x = getX(pointLocationI,pointLocationJ);

    y = getY(pointLocationI,pointLocationJ);

    z = getZ(pointLocationI);

    pointList.push(x,y,z);

    return pointList;

}

Функції що обчислюють масштабування текстури навколо заданої точки

mat3 Scale(float fScaleValue) {

    mat3 sc = mat3(vec3(fScaleValue, 0.0, 0.0),

                   vec3(0.0, fScaleValue, 0.0),

                   vec3(0.0, 0.0,  1.0));

    return sc;

}

mat3 Translate(vec2 UserPointLocation){

    mat3 tr = mat3(vec3(1.0, 0.0, 0.0),

                   vec3(0.0, 1.0, 0.0),

                   vec3(UserPointLocation.x, UserPointLocation.y,1.0));

    return tr;

}

vec2 GetTextCoord(float fScaleValue, vec2 UserPointLocation, vec2 texCoordLocation ){

        vec3 TextCoords = vec3(texCoordLocation, 1);

        mat3 sc = Scale(fScaleValue);

        mat3 tr = Translate(-UserPointLocation);

        mat3 trBack = Translate(UserPointLocation);

        TextCoords = tr \* TextCoords;

        TextCoords = sc \* TextCoords;

        TextCoords = trBack \* TextCoords;

        return TextCoords.xy;

}