Міністерство освіти і науки України

Національний технічний університет України «КПІ ім. Ігоря Сікорського»

Кафедра цифрових технологій в енергетиці

Візуалізація графічної та геометричної інформації

**Розрахунково-графічна робота**

ВАРІАНТ №25

Виконав студент 1-го курсу магістратури

Кафедри ІАТЕ групи ТР-32мп

Тополюк Кирило Михайлович

Перевірив: Демчишин Анатолій Анатолійович

Київ – 2023

**Постановка задачі**

**Варіант:** 25 – непарний, реалізувати масштабування(scaling).

**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**

**WebGL** (Web Graphics Library) - це JavaScript API для взаємодії з графікою та 3D-графікою у веб-браузерах. Це стандарт, розроблений для створення візуально насичених та інтерактивних веб-сцен та додавання тривимірної графіки до веб-сайтів без використання плагінів. WebGL базується на OpenGL ES, стандарті для вбудованих систем, що використовуються в основному для мобільних пристроїв.

Основні риси WebGL включають в себе можливість використання апаратного прискорення графіки для відтворення тривимірної графіки у реальному часі, створення складних візуальних ефектів, таких як тіні та відбиття, та взаємодія з користувачем через мишу, клавіатуру та інші пристрої введення. WebGL широко використовується в інтернет-галузі для створення вражаючих веб-інтерфейсів та віртуальної реальності.

**Шейдери** у WebGL — це програми, написані мовою GLSL (OpenGL Shading Language), які використовуються для визначення обчислень, пов'язаних з обробкою графічних об'єктів у тривимірному просторі. У контексті WebGL шейдери відповідають за визначення положення вершин об'єктів, їх кольору, а також обчислення світлових ефектів та інших графічних атрибутів.

Шейдери розділяються на два типи: вершинні та фрагментні. Вершинні шейдери відповідають за обробку вершин об'єктів, визначаючи їхнє положення в просторі. Фрагментні шейдери визначають колір кожного пікселя в зображенні, а також виконують обчислення світлових та інших ефектів. Вони становлять ключовий елемент програмної частини WebGL, дозволяючи здійснювати гнучкий та реалістичний відображення графічних об'єктів у веб-браузері.

**Вершинні** **шейдери** виконують обчислення для кожної вершини об'єкта у тривимірному просторі. Основні завдання вершинного шейдера включають в себе трансформацію вершин з їхнього локального простору в глобальний, визначення їхнього кольору та інших атрибутів. Ці обчислення включають у себе матричні операції для обрання точного положення вершин у просторі, а також врахування камери та освітлення.

Вони можуть також використовуватися для передачі даних між вершинами, таких як нормалі чи текстурні координати. Ці шейдери генерують вихідні дані, які використовуються фрагментними шейдерами для обчислення кольору.

**Фрагментні** **шейдери** відповідають за обчислення кольору для кожного пікселя у зображенні. Вони працюють на рівні фрагментів, які генеруються після растеризації трикутників вершинним шейдером. Основна мета фрагментного шейдера - визначити кінцевий колір для кожного пікселя.

**Текстура** (*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.

Щоб виконати масштабування навколо будь-якої точки, потрібно здійснити кілька кроків з використанням матриць переміщення (T) і масштабування (R). Почнемо зі створення матриці переміщення (T), яка перенесе об'єкт з його початкового положення до точки масштабування. Далі, матриця масштабування (R) виконає зміну масштабу об'єкта навколо цієї точки. Для масштабування навколо обраної точки необхідно спочатку перемістити цю точку у позицію початкової точки шляхом використання оберненої матриці переміщення (T), яку ми позначимо як T\_1. Після цього ми можемо здійснити масштабування об'єкта відносно обраної точки, використовуючи матрицю масштабування (R), а потім застосувати матрицю переміщення (T), щоб повернути точку масштабування до її початкового положення.

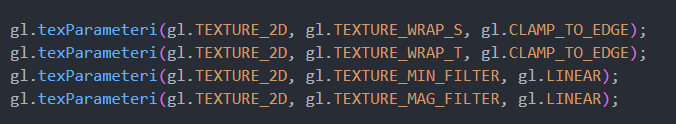
**Реалізація**

Для відображення на поверхні було завантажено зображення за посиланням з інтернету. Зображення має наступний вигляд:



*Рисунок 1. Обране зображення*

Налаштування параметрів текстури було встановлено на такі значення:



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. Додатково були створені змінні, що містять значення масштабу (**ScaleValue**) та два параметри, що зберігають координати точки, відносно якої відбувається масштабування (**pointLocationI**, **pointLocationJ**).

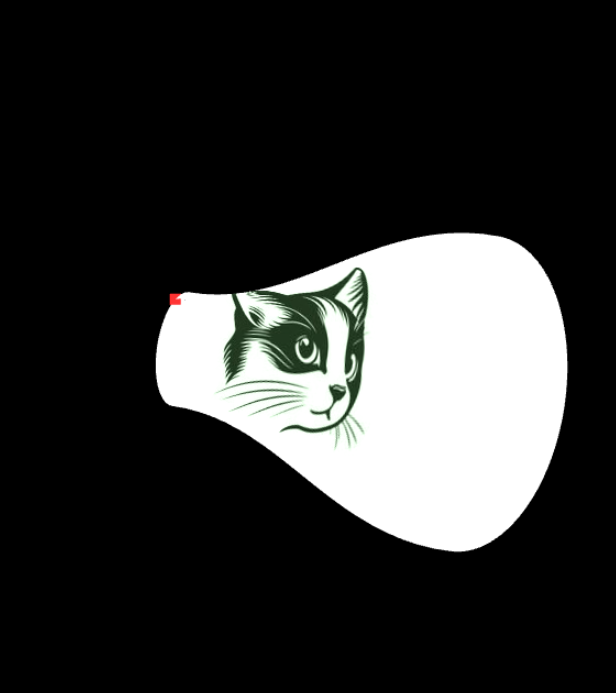
У шейдері реалізовано три функції: **Scale**(), яка приймає значення масштабу як аргумент і повертає матрицю масштабу; **Translate**(), яка приймає значення координат точки, що масштабується, і повертає матрицю зміщення; і функція **GetTextCoord**(), яка повертає значення масштабованих координат.

**Інструкція користувача**

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

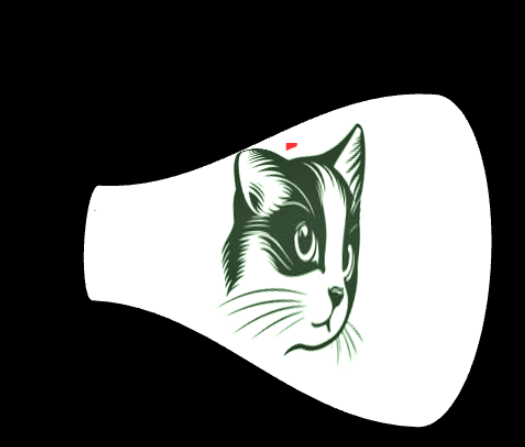
* (a та d) – переміщення точки масштабування вздовж параметра u;
* (w та s) – переміщення точки масштабування вздовж параметра v;
* (+ та -) – зміна значення масштабування.

На рисунку 2 показано вигляд точки та текстури за площині



*Рисунок 2. Приклад відображення*

* За допомогою (d) перемістимо вздовж u:



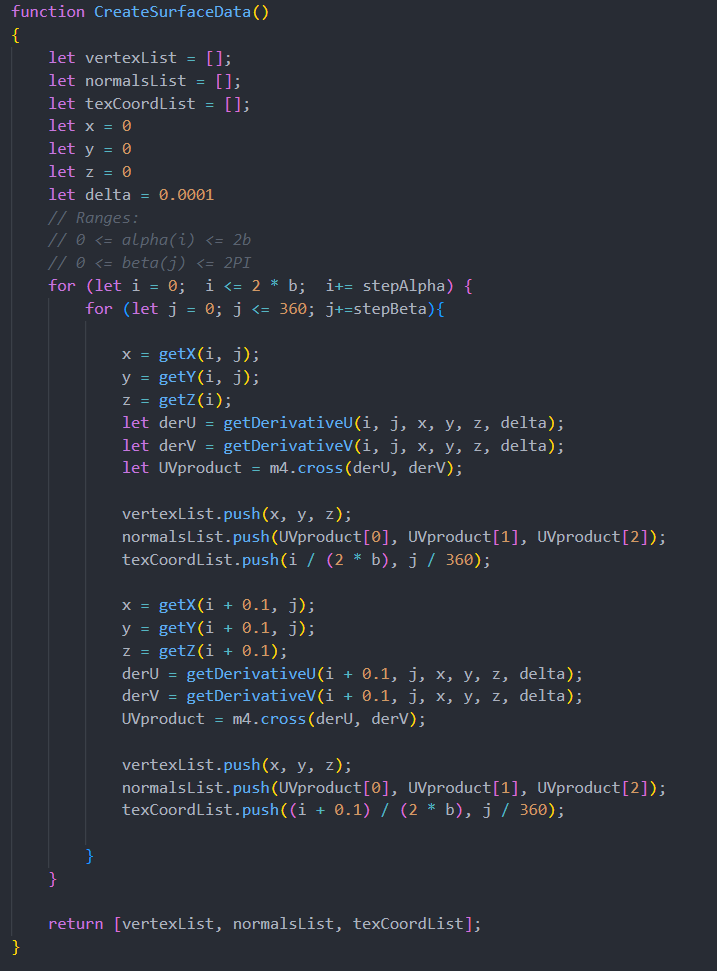
*Рисунок 3. Зміна параметрів*

**Програмний код**

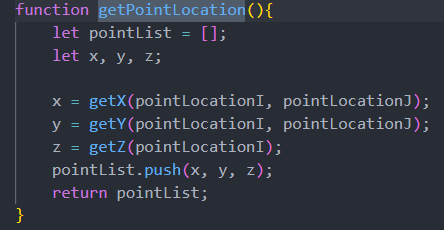
Функція що відповідає за завантаження текстури :



Функція обчислення площини та вершин:



Функція що обчислює координати точки для відображення на поверхні:



Функції що обчислюють масштабування текстури в шейдері:

