МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

КАФЕДРА ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

Варіант - 10

**Розрахунково-графічна робота**

з дисципліни «Візуалізація графічної та геометричної інформації»

**Виконав:**

Студент 5-го курсу

ІАТЕ

групи ТР-22мп

Кедрун Володимир Олександрович

**Перевірив:**

Демчишин Анатолій Анатолійович

Київ-2023**Завдання**

1. Нанести текстуру на поверхню з практичного завдання №2.
2. Реалізувати обертання текстури навколо вказаної користувачем точки.
3. Реалізувати можливість переміщати цю точку по поверхні за допомогою клавіш W, A, S, D (W, S відповідають за пересування вздовж параметра v, а A, D вздовж параметра u).
4. Створити gif зображення роботи розробленої програми.
5. Створити звіт з виконання розрахунково графічної роботи.
6. Завантажити код, gif та звіт в окрему гілку в репозиторії GitHub.

**Мета роботи**: засвоїти навички роботи з текстурами та бібліотеки WebGL.

**Основне завдання**: відобразити текстуру на поверхні та реалізувати обертання текстури навколо вказаної точки на поверхні.

**Теоретичні відомості**

**WebGL** (скорочення від Web Graphics Library) — це API JavaScript для відтворення інтерактивної 2D і 3D графіки в будь-якому сумісному веб-браузері без використання плагінів. WebGL повністю інтегровано з іншими веб-стандартами, що дозволяє використовувати фізику, обробку зображень і ефекти з прискореним графічним процесором як частину полотна веб-сторінки. Елементи WebGL можна змішувати з іншими елементами HTML і поєднувати з іншими частинами чи фоном сторінки.

Програми WebGL складаються з керуючого коду, написаного на JavaScript, і коду шейдера, написаного мовою OpenGL ES Shading Language (GLSL ES), мовою, схожою на C або C++, і виконується на графічному процесорі (GPU) комп’ютера. WebGL розроблено та підтримується неприбутковою компанією Khronos Group.

WebGL 1.0 базується на OpenGL ES 2.0 і надає API для 3D-графіки. Він використовує елемент canvas HTML5 і доступ до нього здійснюється за допомогою інтерфейсів Document Object Model (DOM).

WebGL 2.0 базується на OpenGL ES 3.0 і забезпечує гарантовану доступність багатьох додаткових розширень WebGL 1.0 і надає нові API. Автоматичне керування пам'яттю неявно забезпечується JavaScript.

Як і OpenGL ES 2.0, WebGL не має фіксованих функцій API, представлених у OpenGL 1.0 і застарілих у OpenGL 3.0. Цю функцію, якщо вона потрібна, має реалізувати кінцевий розробник, надавши код шейдера та налаштувавши прив’язки даних у JavaScript.

Шейдери в WebGL виражаються безпосередньо в GLSL і передаються в API WebGL як текстові рядки. Реалізація WebGL компілює ці інструкції шейдерів у код GPU. Цей код виконується для кожної вершини, надісланої через API, і для кожного пікселя, растеризованого на екрані.

**Текстура** (англ. Texture mapping) — це спосіб надання поверхні 3D деталей — полігону: кольору, фактури, блиску, матовості та інших фізичних властивостей (для імітації найчастіше якогось природного матеріалу, наприклад: паперу, дерева, каменю, металу тощо). Першим цю техніку запровадив Едвін Кетмул у 1974 році.

Відображення текстур спочатку називалося дифузним відображенням. Це був методо, який просто зображав пікселі текстури на 3D-поверхні («обгортаючи» об’єкт зображенням). В останні десятиліття поява багатопрохідного рендерингу, мультитекстурування, mipmaps і більш складних відображень, таких як height mapping, bump mapping, normal mapping, displacement mapping, reflection mapping, specular mapping, occlusion mapping та багато інших варіацій зробили можливим імітацію майже фото-реалістичних об’єктів в режимі реального часу.

Досягти цього вдалось шляхом значного зменшення кількості багатокутників, а також розрахунку освітлення, необхідного для створення реалістичної та функціональної 3D-сцени.

**Карта текстури** — це зображення, нанесене на поверхню фігури або багатокутника. Це може бути растрове зображення або процедурна текстура.

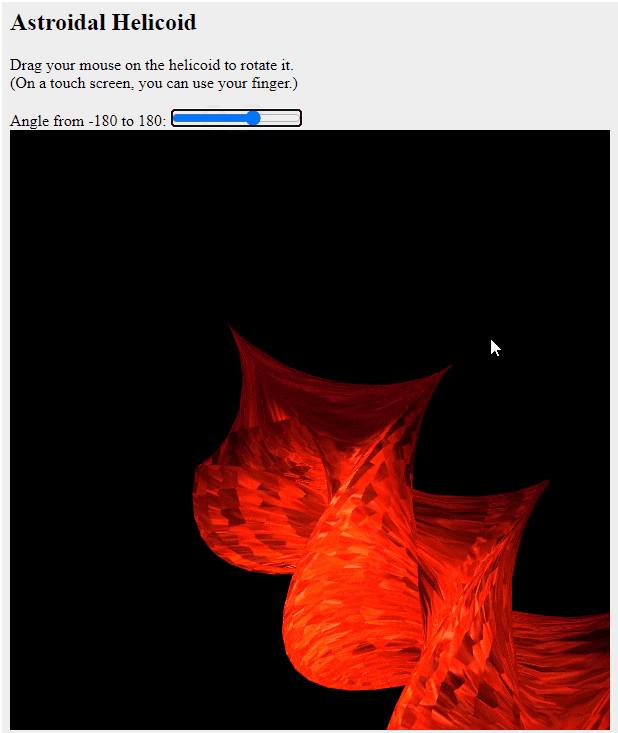
Вони можуть бути одно, двох, або тривимірними. Для використання з сучасним апаратним забезпеченням дані карти текстури можуть зберігатися в розрізненому або мозаїчному порядку для покращення когерентності кешу. API візуалізації зазвичай керують ресурсами карти текстури (які можуть бути розташовані в пам’яті пристрою) як буферами або поверхнями.

Зазвичай вони містять дані кольору **RGB** (збережені як прямі кольори, стислі формати або індексовані кольори), а іноді й додатковий канал для альфа-змішування (**RGBA**). Можна використовувати альфа-канал (який може бути зручним для зберігання у форматах, аналізованих апаратним забезпеченням) для інших цілей, наприклад для відображення.

Кілька текстурних карт (або каналів) можна комбінувати для контролю дзеркальності, нормалей, зміщення або підповерхневого розсіювання, наприклад для візуалізації шкіри.

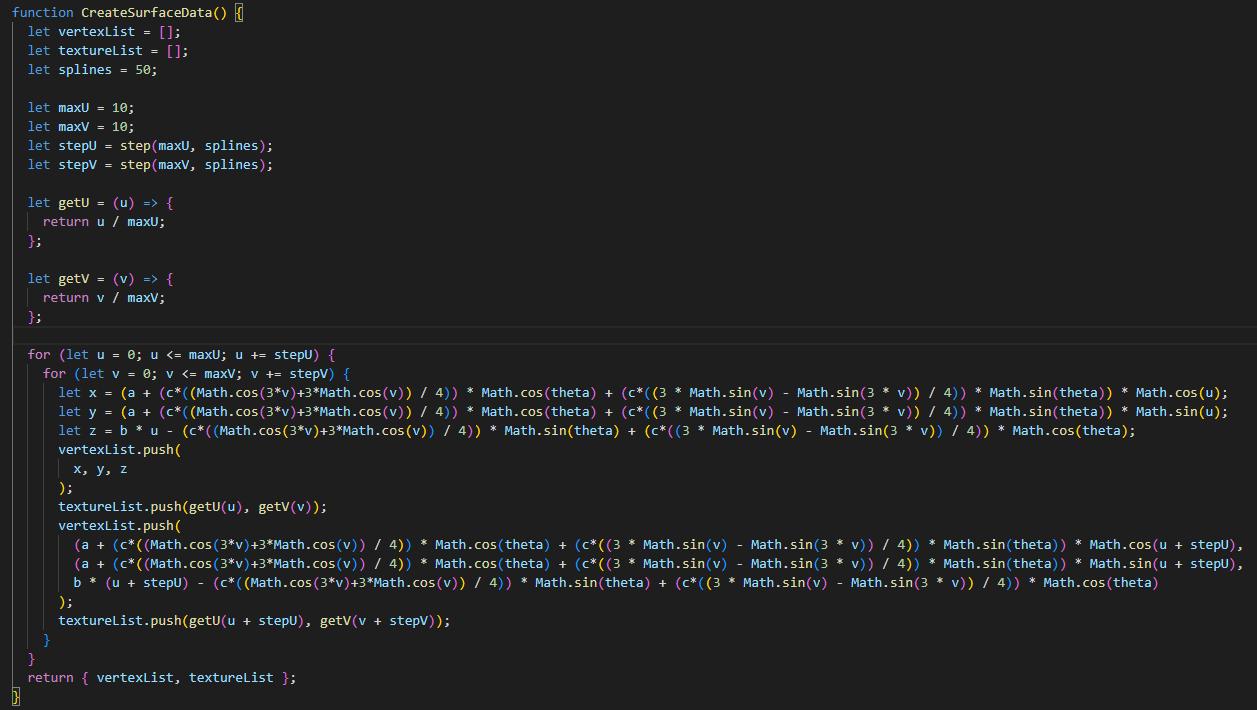
# Інструкція користувача

* W S – переміщення точки масштабування вздовж параметра v;
* A D – переміщення точки масштабування вздовж параметра u;
* Для змінення куту створено “бігунок” зі значеннями від -180 до 180
* Можливість обертання фігури за допомогою миші або сенсорного екрану;



# Опис вихідного коду

Функція для створення фігури



Код вертексного та фрагментного шейдера

