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

Алі-Шах І.Г.

**Перевірив:**

Демчишин А.А.

Київ 2023

**Постановка задачі**

**Варіант:** 1 – непарний, реалізувати масштабування.

**Мета:** навчитися працювати з текстурами в WebGL, відобразити текстуру поверхні з 2 практичного завдання та реалізувати масштабування текстурних координат.

**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.

**Звіт повинен містити:**

• титульну сторінку;

• розділ з постановкою задачі;

• розділ з теоретичними відомостями;

• розділ з описом деталей реалізації;

• розділ з інструкціями користувача зі скріншотами;

• зразок вихідного коду.

**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" \o "Англійська мова) *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)" \o "Тексел (графіка)) — 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 з комп’ютера завантажується зображення чи відео використовуючи функції JavaScript.

Налаштування параметрів текстури було встановлено на такі значення:

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

Обрахунок текстурних координат відповідно кожного вертекса реалізовано в основному циклі і здійснюється за допомогою нормалізації параметрів **u** та **v**, за допомогою яких будується поверхня.

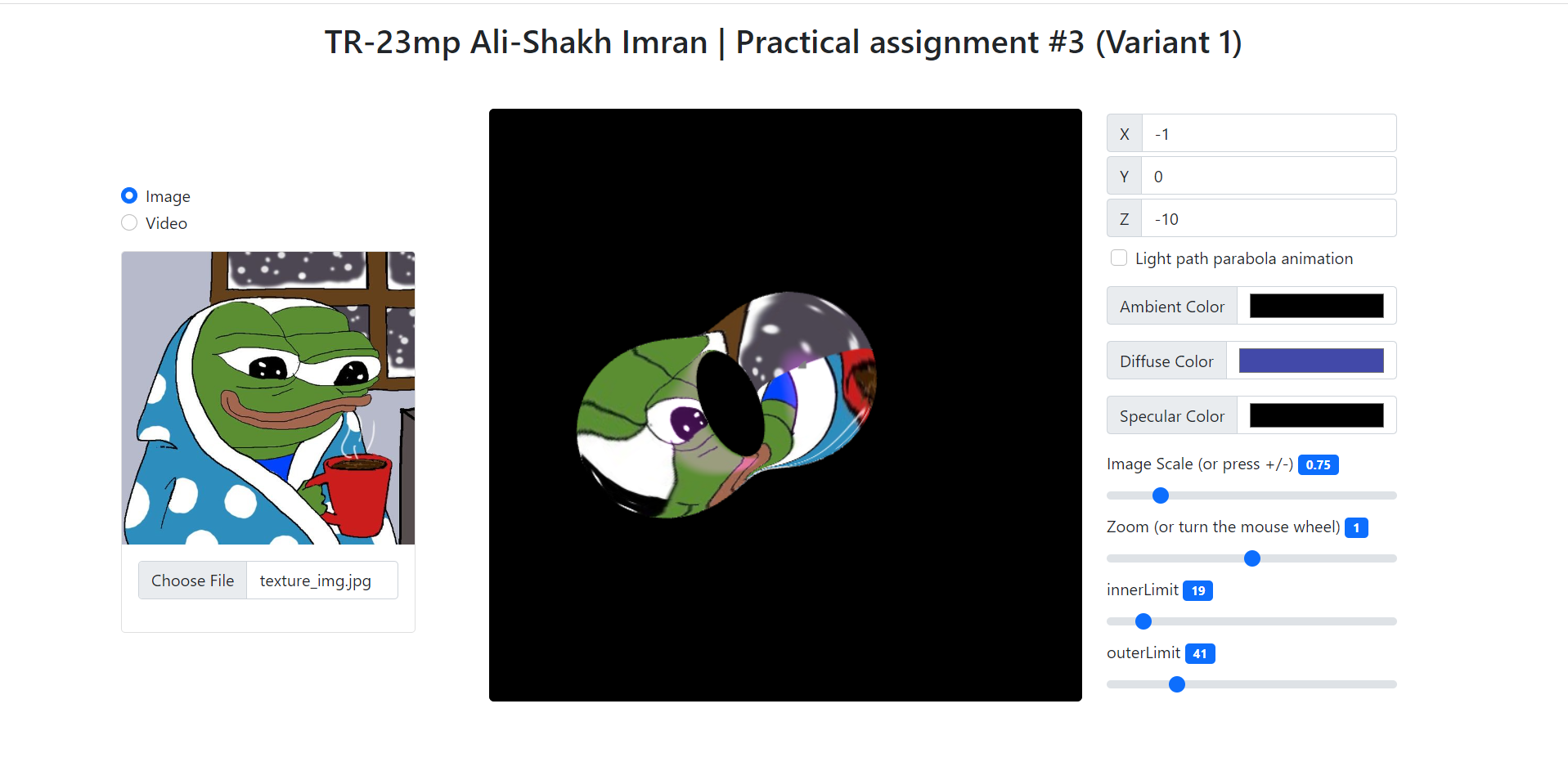
Також, були створені глобальні параметри, які зберігають координати точки масштабування та значення масштабування. Для взаємодії з веб-додатком реалізовано інтерфейс який включає в себе наступні клавіші та поля:

* (**a** та **d**) – переміщення точки масштабування вздовж параметра u;
* (**w** та **s**) – переміщення точки масштабування вздовж параметра v;
* (+ та -) – зміна значення масштабування;
* Поля вводу типу *file* для завантаження фото чи відео;
* Поля вводу типу *text* для переміщення в просторі;
* Поля вводу типу *checkbox* для запуску анімації освітлення за параболою;
* Поля вводу типу *color* для задання кольорів;
* Поля вводу типу *range* для масштабування текстури та поверхні і зміни внутрішньої та зовнішньої межі освітлення.

**Інструкція користувача**

Для масштабування вздовж параметра u використовується (a та d). Для масштабування вздовж параметра v використовується (w та s). (+ та -) використовуються для зміни значення масштабування. Поля вводу типу *file* використовуються для завантаження фото чи відео. Поля вводу типу *text* використовуються для переміщення в просторі. Поле вводу типу *checkbox* використовуються для запуску анімації освітлення за параболою. Поля вводу типу *color* використовуються для задання кольорів. Поля вводу типу *range* використовуються для масштабування текстури та поверхні і зміни внутрішньої та зовнішньої межі освітлення.

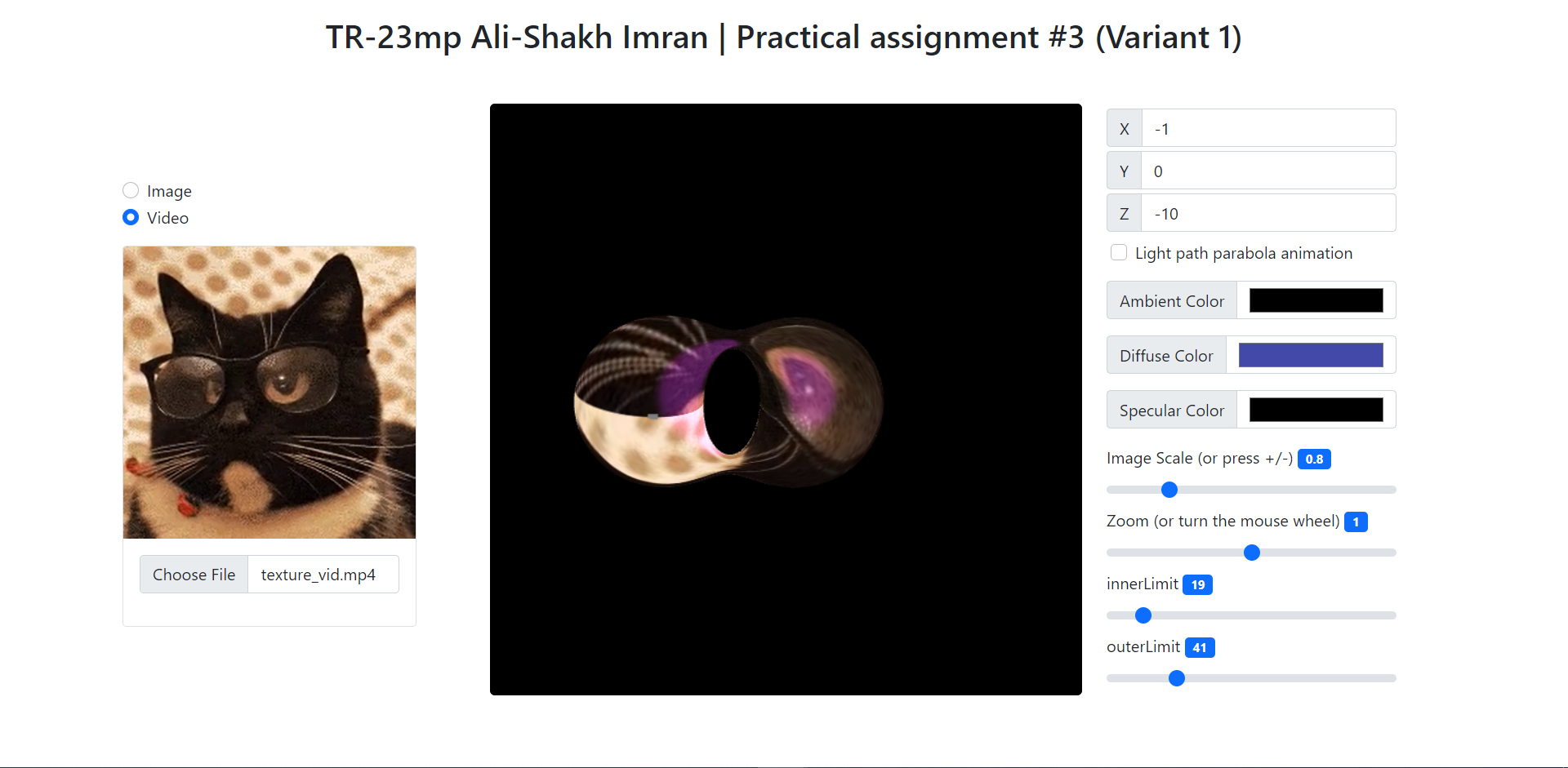
При завантаженні сторінки, за замовчуванням задано фото, відео та наступні параметри (рисунок 1):



*Рисунок 1. Сторінка веб-додатку*

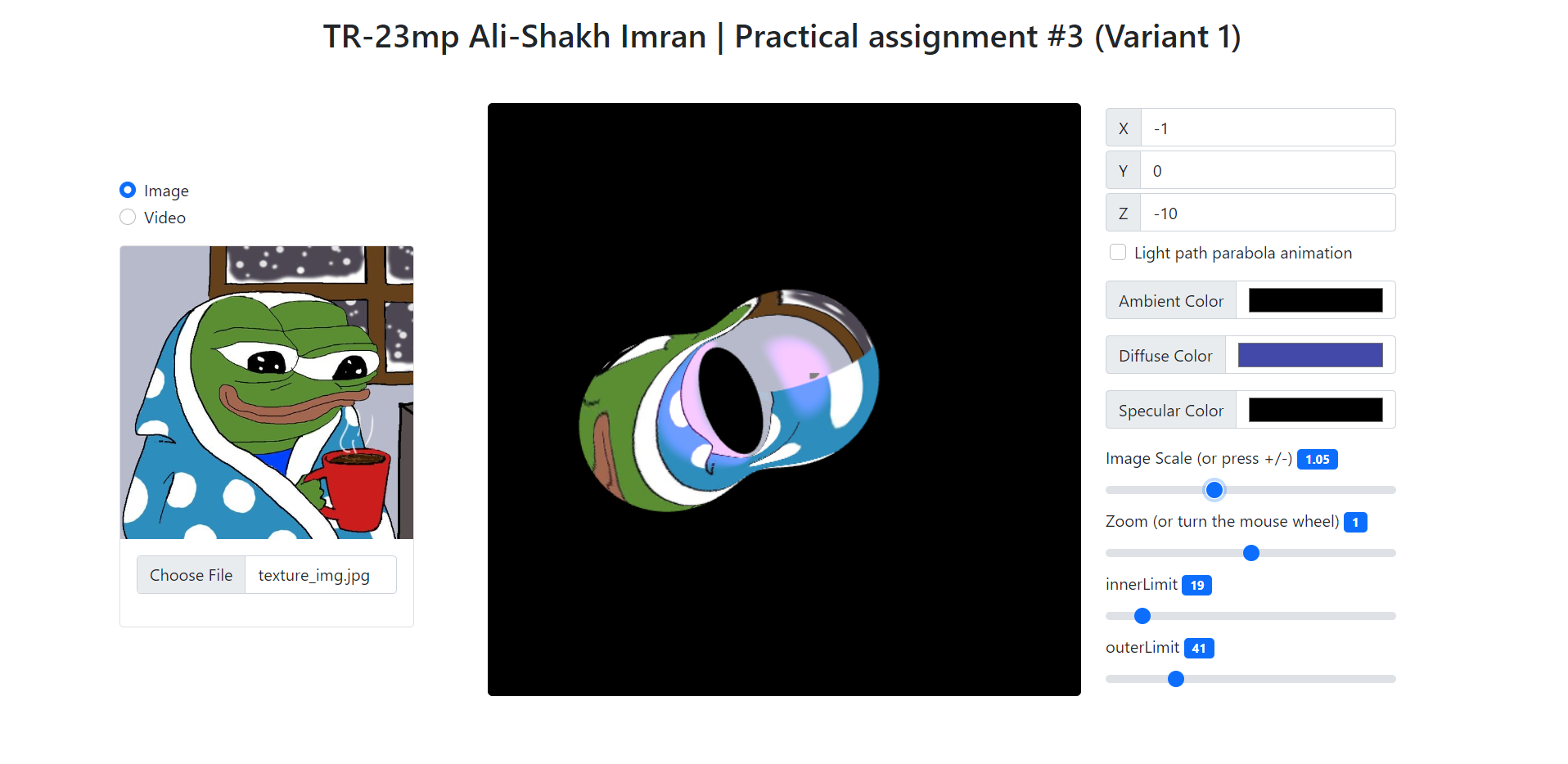
Натиснувши на відповідні кнопки, користувач має змогу завантажити зі свого комп’ютера фотографію чи відео яке він хотів би нанести на поверхню.

На рисунку 2, зображено відрисовку відео на поверхні:



*Рисунок 2. Демонстрація накладання відео на поверхню*

На наступному рисунку, зображено масштабування текстури.



*Рисунок 3. Демонстрація масштабування текстури*

**Зразок вихідного коду**

Код, що відповідає за завантаження зображення, його створення та ініціалізацію, для подальшого відображення на поверхні:

function loadTexture(gl, url) {

    var texture = gl.createTexture();

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

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

        new Uint8Array([0, 0, 255, 255]));

    var image = new Image();

    image.crossOrigin = "anonymous"

    image.onload = () => {

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

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

        if (isPowerOf2(image.width) && isPowerOf2(image.height)) {

            gl.generateMipmap(gl.TEXTURE\_2D);

        } else {

            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);

        }

        draw();

    };

    image.src = url;

    return texture;

}

function isPowerOf2(value) {

    return (value & (value - 1)) === 0;

}

Код, що відповідає за завантаження відео, його створення та ініціалізацію, для подальшого відображення на поверхні:

function initTexture(gl) {

    var texture = gl.createTexture();

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

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

        new Uint8Array([0, 0, 255, 255]));

    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);

    return texture;

}

function updateTexture(gl, texture, video) {

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

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

    draw();

}

function setupVideo(url) {

    var video = document.createElement("video");

    video.crossOrigin = "anonymous"

    let playing = false;

    let timeupdate = false;

    video.playsInline = true;

    video.muted = true;

    video.loop = true;

    video.addEventListener(

        "playing",

        () => {

            playing = true;

            checkReady();

        },

        true

    );

    video.addEventListener(

        "timeupdate",

        () => {

            timeupdate = true;

            checkReady();

        },

        true

    );

    video.src = url;

    video.play();

    function checkReady() {

        if (playing && timeupdate) {

            copyVideo = true;

        }

    }

    return video;

}