**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

**Лабораторная работа № 3**

Тема: Основы построения фотореалистичных изображений

Студент: Манташев Асадулла Уллубиевич

Группа: 08-305

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2022

1. Постановка задачи

Используя результаты Л.Р. №2, аппроксимировать заданное тело выпуклым многогранником. Точность аппроксимации задается пользователем. Обеспечить возможность вращения и масштабирования многогранника и удаление невидимых линий и поверхностей. Реализовать простую модель закраски для случая одного источника света.

Параметры освещения и отражающие свойства материала задаются пользователем в диалоговом режиме.

Вариант 12: Прямой круговой цилиндр.

1. Описание программы

Программа написана на JavaScript с использованием p5 и WebGL, состоит из 1 файла: main.js. Стадии инициализации приложения:

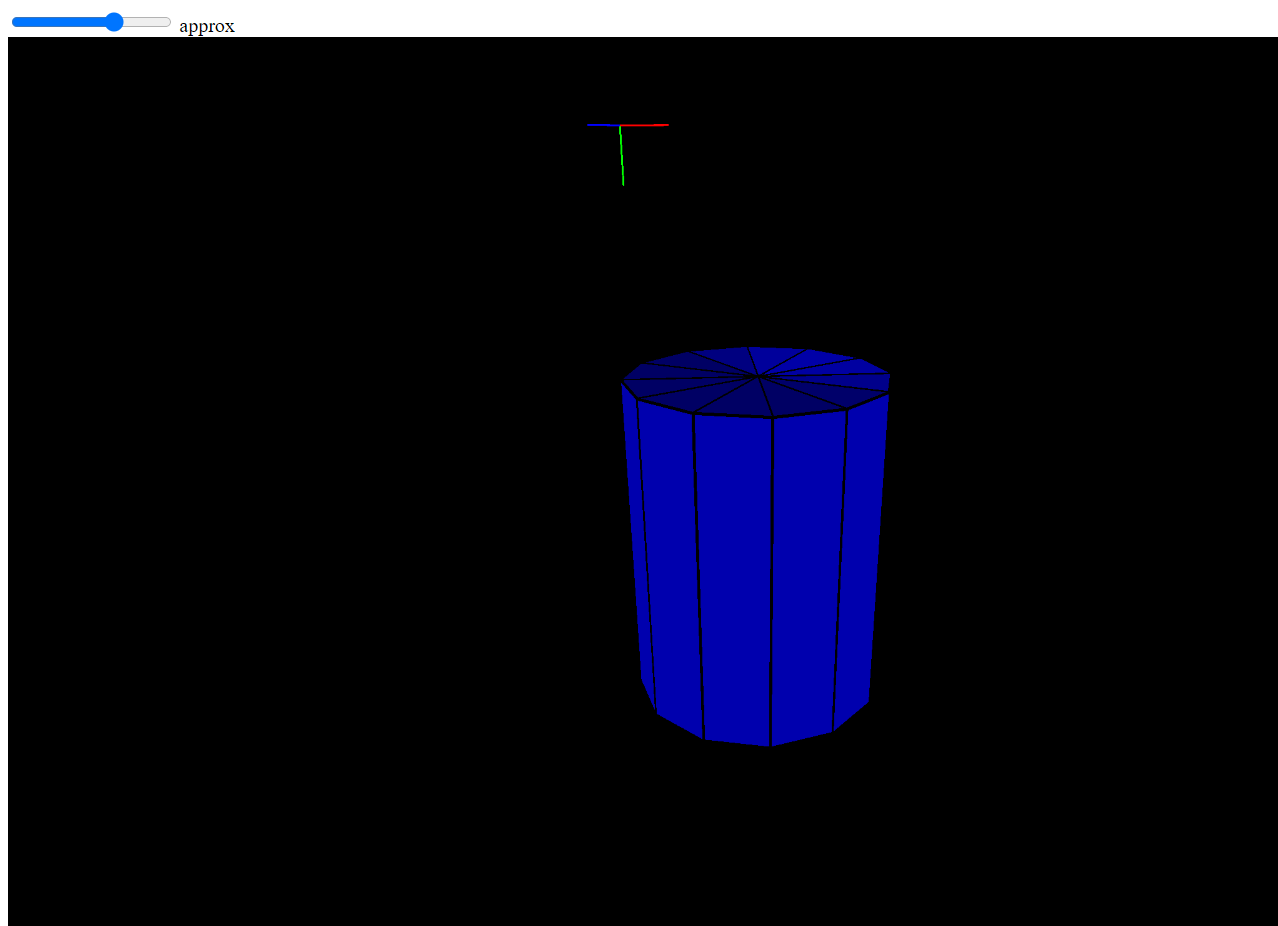
1. Цилиндр рисуется за два шага в функции drawCylinder(): создаются боковые стороны и основания. Отрисовка происходит полигонами по кругу. Основания составляются из треугольников, которые находятся на заданном расстоянии. Также для построения оснований используются окружности заданного радиуса.

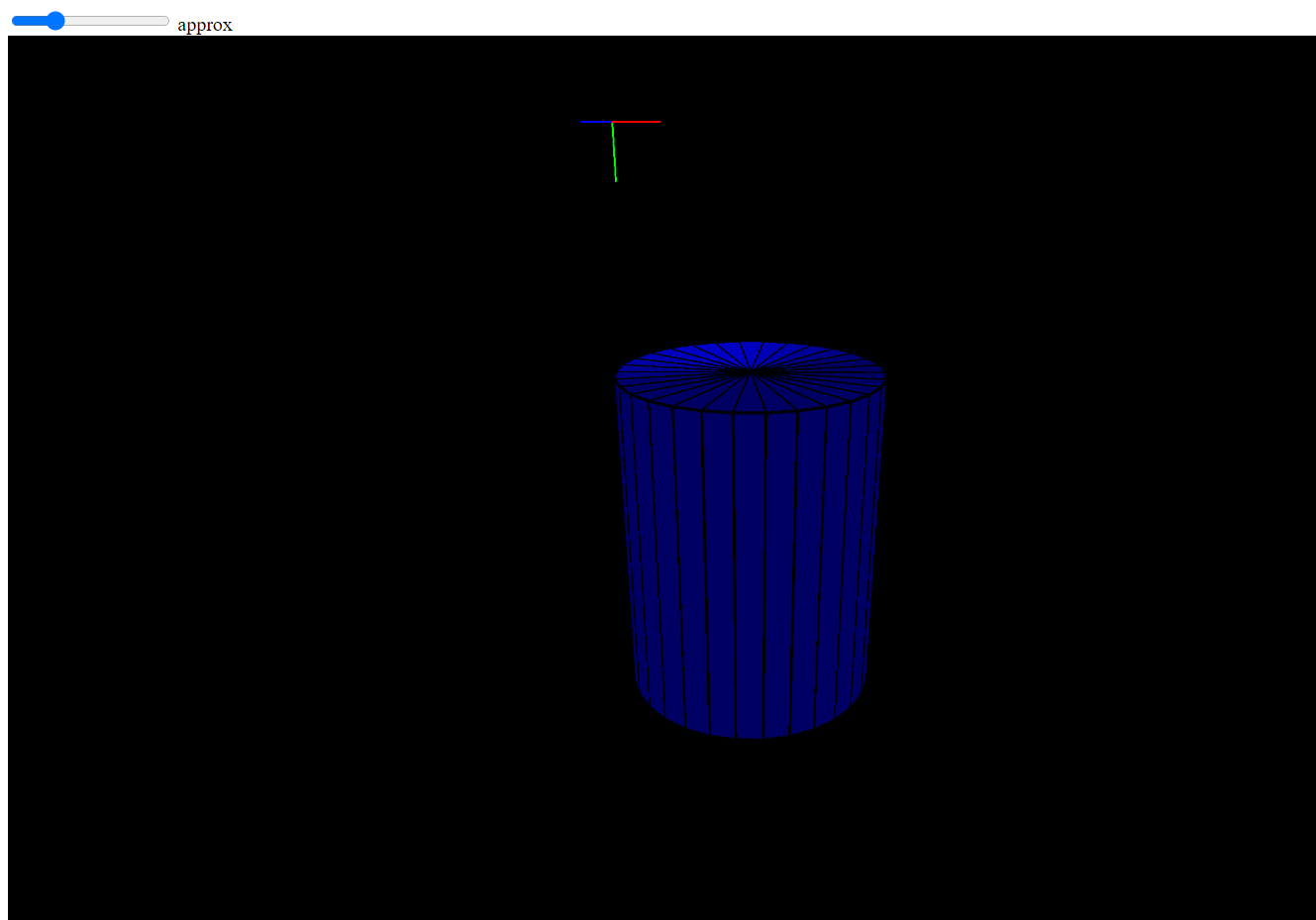
2. Метод beginShape() начинает записывать вершины фигуры, а endShape() – заканчивает. Далее следуют методы vertex() с тремя параметрами, которые задают позицию вершины в 3D-пространстве. Цвет контура каждой фигуры задается с помощью метода stroke().

3. Функция setup(). Метод createCanvas() создаёт поле для изображения фигуры, заданы размеры поля. debugMode() помогает визуализировать 3D-пространство, добавляя сетку, добавляющую к эскизу значок осей, указывающих направления X, Y и Z.

4. Функция draw(). ambientLight(100) – создаёт свет. directionalLight – создаёт направленный свет с цветом и направлением. orbitControl – позволяет перемещаться по 3D эскизу с помощью мыши..

1. Результат работы программы





1. Листинг программы

/\*

Манташев А.У. М8О-305Б-20

Лабораторная работа №3.

Задание:

Используя результаты Л.Р. №2, аппроксимировать заданное тело выпуклым многогранником. Точность

аппроксимации задается пользователем. Обеспечить возможность вращения и масштабирования многогранника и

удаление невидимых линий и поверхностей. Реализовать простую модель закраски для случая одного источника света.

Параметры освещения и отражающие свойства материала задаются пользователем в диалоговом режиме.

Вариант 12:

Прямой круговой цилиндр.

\*/

'use strict'

let WIDTH = 1200, HEIGHT = 800;

function drawCylinder(h, r, x, y, z) {

let step = int(document.getElementById('approx').value);

for (var fi = 0; fi < 360; fi += step) { // array of vertices

beginShape();

vertex(x, y + h / 2, z);

vertex(x + cos(fi / 360 \* 2 \* PI) \* r,

y + h / 2,

z + sin(fi / 360 \* 2 \* PI) \* r);

vertex(x + cos((fi + step) / 360 \* 2 \* PI) \* r,

y + h / 2,

z + sin((fi + step) / 360 \* 2 \* PI) \* r);

endShape(); // соединение вершин в одну плоскость

beginShape();

vertex(x, y - h / 2, z);

vertex(x + cos(fi / 360 \* 2 \* PI) \* r,

y - h / 2,

z + sin(fi / 360 \* 2 \* PI) \* r);

vertex(x + cos((fi + step) / 360 \* 2 \* PI) \* r,

y - h / 2,

z + sin((fi + step) / 360 \* 2 \* PI) \* r);

endShape();

beginShape();

vertex(x + cos(fi / 360 \* 2 \* PI) \* r,

y - h / 2,

z + sin(fi / 360 \* 2 \* PI) \* r);

vertex(x + cos(fi / 360 \* 2 \* PI) \* r,

y + h / 2,

z + sin(fi / 360 \* 2 \* PI) \* r);

vertex(x + cos((fi + step) / 360 \* 2 \* PI) \* r,

y + h / 2,

z + sin((fi + step) / 360 \* 2 \* PI) \* r);

vertex(x + cos((fi + step) / 360 \* 2 \* PI) \* r,

y - h / 2,

z + sin((fi + step) / 360 \* 2 \* PI) \* r);

endShape();

}

}

function setup() { //создаем поле для изображения фигуры

createCanvas(WIDTH, HEIGHT, WEBGL);

background(200);

debugMode(AXES);

}

function draw() {

stroke(0, 0, 0);

strokeWeight(2);

background('black');

ambientLight(100); //создает свет

let x1 = map(mouseX, 0, WIDTH, -200, 200);

let x2 = map(mouseY, 0, HEIGHT, 200, -200);

directionalLight(100, 100, 100, x1, x2, 300); //создает направленный свет

ambientMaterial('blue');

orbitControl();

drawCylinder(260, 100, 0, 0, 0);

}

ЛИТЕРАТУРА

1. Трехмерная графика в вебе [Электронный ресурс]. URL- [https://habr.com/ru/post/325646/](https://habr.com/ru/post/325646/%20) (дата обращения 22.11.2022)
2. Документация p5 [Электронный ресурс]. URL - [https://p5js.org/reference/](https://p5js.org/reference/%20) (дата обращения 22.11.2022)
3. Цилиндр [Электронный ресурс]. URL- [https://ru.wikipedia.org/wiki/Цилиндр](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D0%BB%D0%B8%D0%BD%D0%B4%D1%80) (дата обращения 22.11.2022)