**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

**Лабораторная работа № 4**

Тема: Ознакомление с технологией OpenGL

Студент: Манташев Асадулла Уллубиевич

Группа: 08-305

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2022

1. Постановка задачи

Создать графическое приложение с использованием OpenGL.

Используя результаты Л.Р.№3, изобразить заданное тело (то же, что и в л.р.№3) с использованием средств OpenGL 2.1. (или выше) или WebGL.

Вариант: Прямой круговой цилиндр.

1. Описание программы

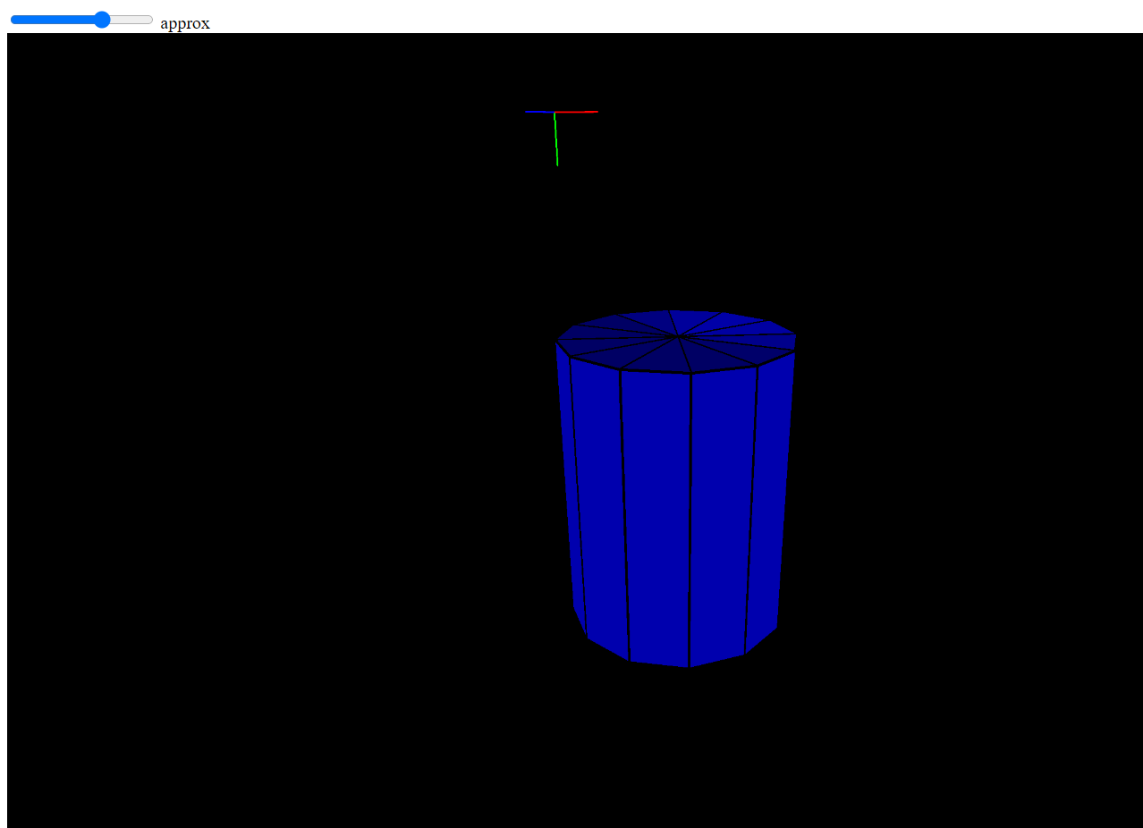
Фигура строится в два этапа: отрисовка боковых сторон и двух

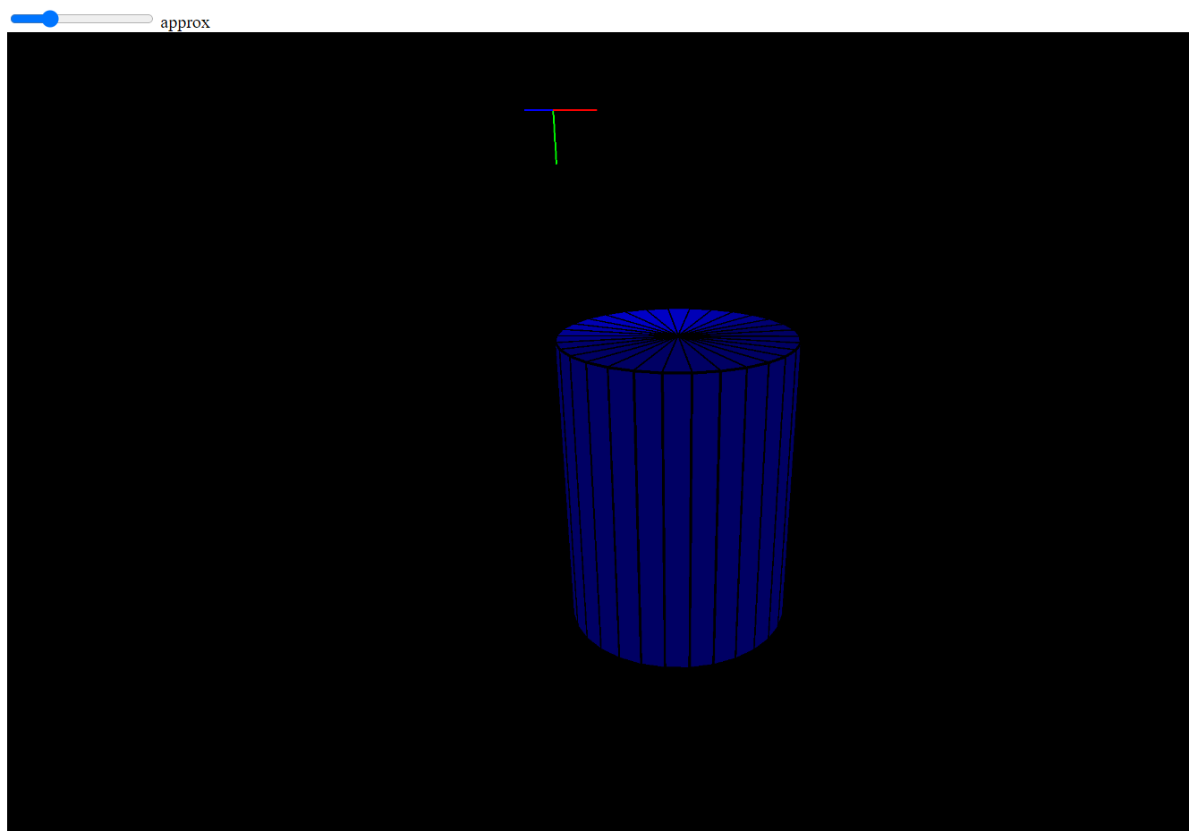
оснований. Из полигонов-треугольников составляются основания, а стенки

строятся как прямоугольники, соединяющие попарно соответственно треугольники верхнего и нижнего оснований. Ползунком n регулируется количество полигонов-треугольников в основании, то есть по сути качество модели цилиндра.

Вместо технологии OpenGl была использована технология WebGl, которая является полной копией первой, только используется для отображения веб-графики.

1. Результат работы программы





1. Листинг программы

/\*

Манташев А.У. М8О-305Б-20

Лабораторная работа №4.

Задание:

Создать графическое приложение

с использованием OpenGL. Используя результаты Л.Р.№3,

изобразить заданное тело (то же, что и в л.р.№3)

с использованием средств OpenGL 2.1. (или выше) или WebGL

Вариант:

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(0, 0, 0);

debugMode(AXES);

}

function draw() {

stroke(0, 0, 0);

strokeWeight(2);

background(0, 0, 0);

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(0, 0, 255);

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 27.11.2022)
2. Документация p5 [Электронный ресурс]. URL - [https://p5js.org/reference/](https://p5js.org/reference/%20) (дата обращения 27.11.2022)
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 27.11.2022)