**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

**Лабораторная работа № 2**

Тема: Каркасная визуализация выпуклого многогранника. Удаление невидимых линий

Студент: Кондратьев Егор Алексеевич

Группа: 80-306

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2021

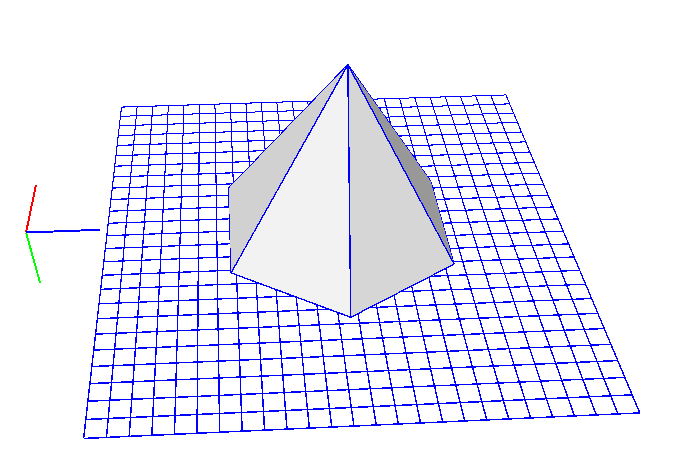
1. Постановка задачи

Вариант №13: многогранная прямая правильная пирамида.

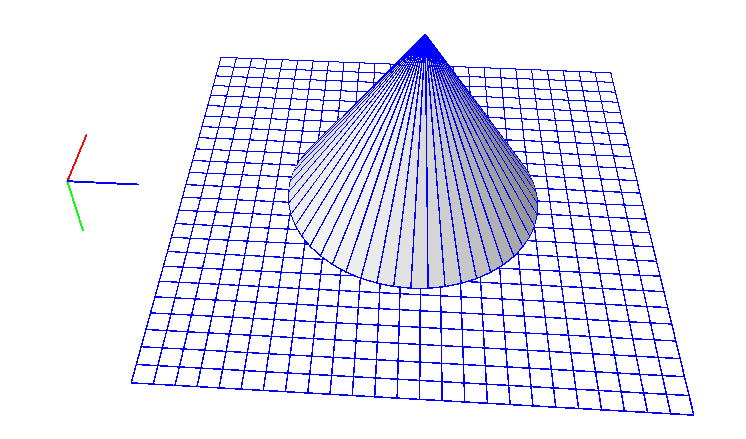
1. Описание программы

Данная фигура строится в два этапа: Для начала берется некий круг в основании, где в цикле многогранником приближенно отображается основание пирамиды. Далее, для каждой стороны этого многогранника строится треугольник, который соединяет основание с вершиной пирамиды, которая задается расстоянием от основания, также как и радиус основания, он передается в функцию отображения. Ползунком n регулируется количество сторон (от 3 до 50)

1. Набор тестов



Приведена пирамида для 6 боковых сторон.



Приведена пирамида для 50 боковых сторон.

1. Листинг программы

<!DOCTYPE html>

<html>

<head>

<title>

Лаба 2 by Dukend

</title>

<style>

.visualy\_hidden {

visibility: hidden

}

</style>

</head>

<body>

<div>

Пирамида <br>

Количество сторон пирамиды <input id="n\_number\_of\_sides" type="range" min="3" max="50" step="1" value="3">

</div>

<script src="p5.js"></script>

<script src="lab02.js"></script>

</body>

</html>

'By Dukend'

new p5();

let WIDTH = 1500, HEIGHT = 1000;

function drawPiramid(h, R, x, y, z) {

let n = document.getElementById('n\_number\_of\_sides').value;

if (n < 3) {

return;

}

let arr = []

for (let i = 0; i < n; i++) {

arr.push({ x: cos(i \* 2 \* PI / n) \* R + (x || 0), y: 0 + (y || 0), z: -(sin(i \* 2 \* PI / n) \* R) + (z || 0) })

}

let top\_point = { x: 0 + (x || 0), y: - h + (y || 0), z: 0 + (z || 0) };

beginShape(TRIANGLES);

vertex(top\_point.x, top\_point.y, top\_point.z);

vertex(arr[arr.length - 1].x, arr[arr.length - 1].y, arr[arr.length - 1].z);

vertex(arr[0].x, arr[0].y, arr[0].z);

endShape();

for (let i = 1; i < n; i++) {

beginShape(TRIANGLES);

vertex(top\_point.x, top\_point.y, top\_point.z);

vertex(arr[i - 1].x, arr[i - 1].y, arr[i - 1].z);

vertex(arr[i].x, arr[i].y, arr[i].z);

endShape();

}

beginShape();

for (let i = 0; i < n; i++) {

vertex(arr[i].x, arr[i].y, arr[i].z);

}

endShape(CLOSE);

}

function setup() {

createCanvas(WIDTH, HEIGHT, WEBGL);

background(255);

debugMode();

}

function draw() {

background(255);

ambientLight(128, 128, 128);

directionalLight(128, 128, 128, 100, -100, 0);

ambientMaterial(255);

orbitControl();

push();

drawPiramid(300, 200, 0, -20, 0);

pop();

}

1. Вывод

Использование графической библиотеки p5.js для визуализации алгоритмических процессов, таких как представление кривых, объектов и структур очень удобно. Вебмодель реализует кроссплатформенность.

ЛИТЕРАТУРА

1. p5.js [Электронный ресурс]. URL: <https://p5js.org/>

(дата обращения: 05.10.2021).