



Высшая
Школа
Электроники и
Компьютерных
Наук

Разработка системы моделирования 5-осевой фрезерной обработки на Unity

Автор работы:

студент КЭ - 222

Захаров Д.И.

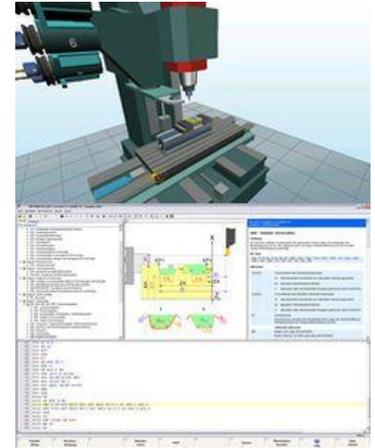
Руководитель работы:

к.т.н., доцент каф. ЭВМ

Парасич В.А.

Актуальность

- создавать симуляторы становится все проще;
- уменьшается риск потерь, связанных с поломками реального оборудования;
- возможно повторно проходить задания.



Цели и задачи

Цель:

Разработка программно-аппаратного комплекса эмуляции процесса обработки детали на фрезерном станке.

Задачи:

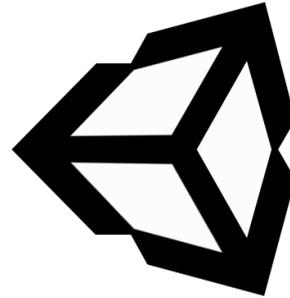
1. Провести обзор и анализ существующих аналогичных или похожих решений;
2. Провести анализ существующих алгоритмов компьютерной графики для редактирования 3D моделей путем генерирования точек;
3. Разработать парсер программ на G-кодах, используемых во фрезерных станках с ЧПУ;
4. Разработать модуль визуализации обработки 3D модели детали;
5. Провести тестирование и оценить работоспособность системы.

Обзор аналогов

| Необходимые требования: | LinuxCNC | CNC Simulator Pro | Autodesk ArtCAM Pro | SprutCAM |
|-----------------------------|----------|-------------------|---------------------|----------|
| Наглядность | - | + | + | + |
| Возможность изучения станка | - | + | + | + |
| Свободное распространение | + | - | - | - |
| Кроссплатформенность | - | - | - | + |

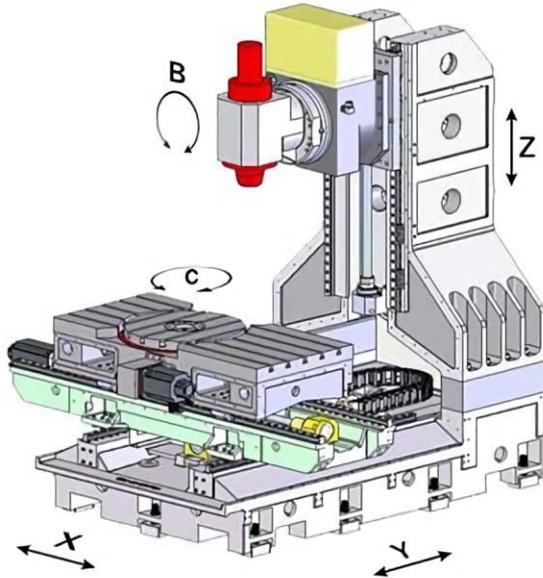
Среда разработки

Средой разработки был выбран игровой движок Unity 3D. Он позволяет разрабатывать наглядные кроссплатформенные приложения, имея при этом подходящие инструменты разработки. Программирование осуществляется на C#.

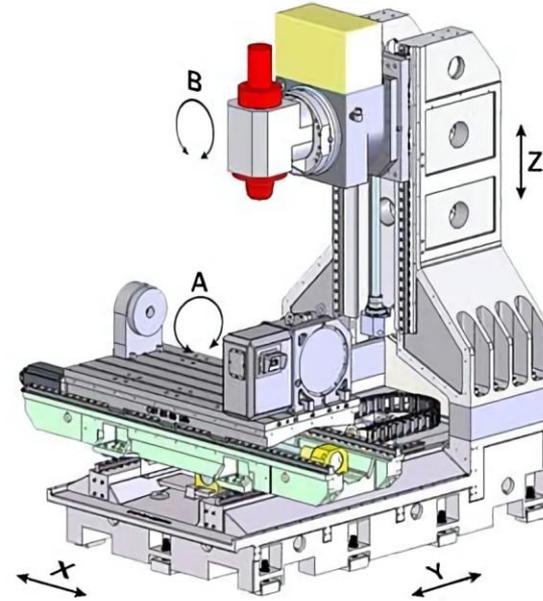


Unity 3D

5-осевая фрезерная обработка

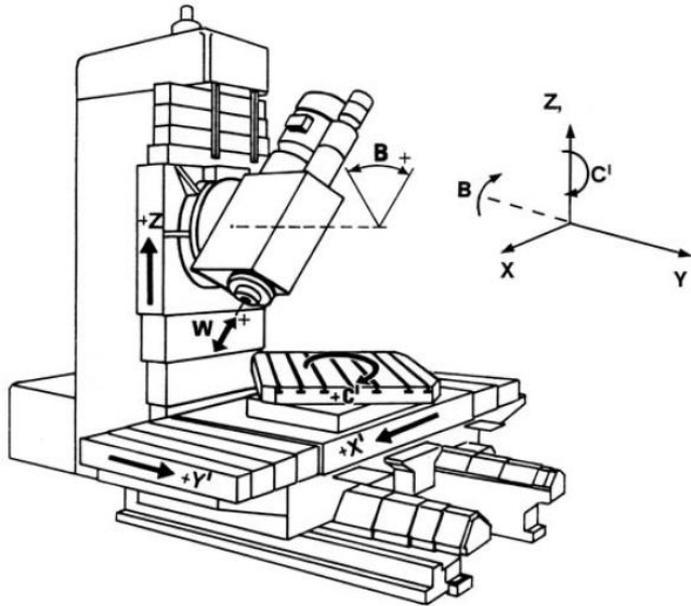


X-5 1250 / 630

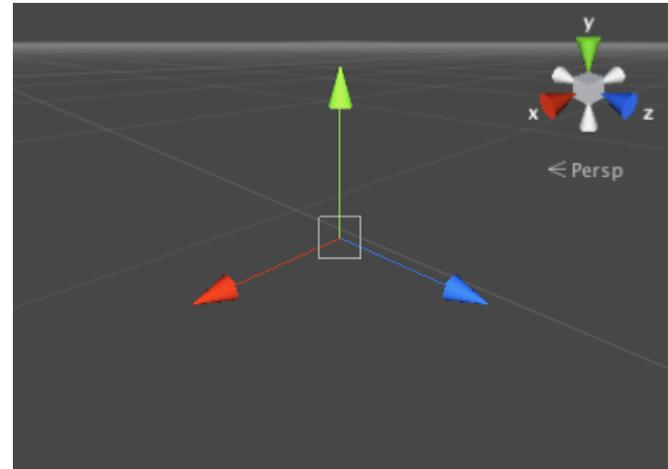


X-5 1300 / 400 *Blademaker*

Различия в системах координат



Правая координатная система (станки)

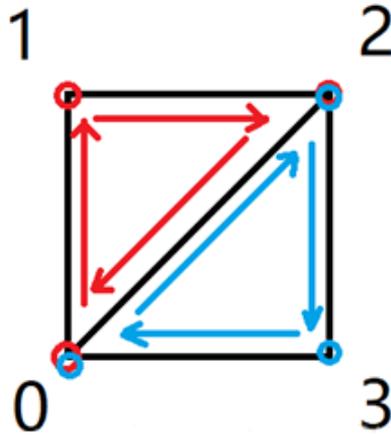
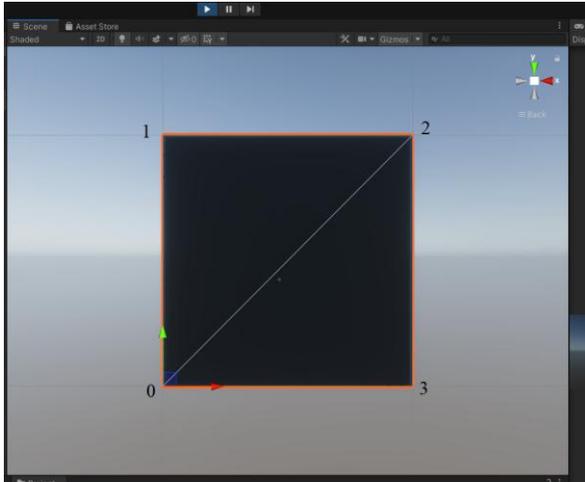


Левая координатная система (Unity)

Управляющие команды станка

| Код | Описание | Пример |
|----------|---|---------------------|
| G00 | Ускоренное перемещение инструмента (холостой ход) | G0 X0 Y0 Z100 |
| G01 | Линейная интерполяция | G01 X0 Y0 Z100 F200 |
| G02 | Круговая интерполяция по часовой стрелке | G02 X15 Y15 R5 F200 |
| G03 | Круговая интерполяция против часовой стрелки | G03 X15 Y15 R5 F200 |
| G20 | Режим работы в дюймовой системе | G90 G20 |
| G21 | Режим работы в метрической системе | G90 G21 |
| Tn Dn M6 | Смена инструмента | T1 D1 M6 |

Генерация 3D-моделей



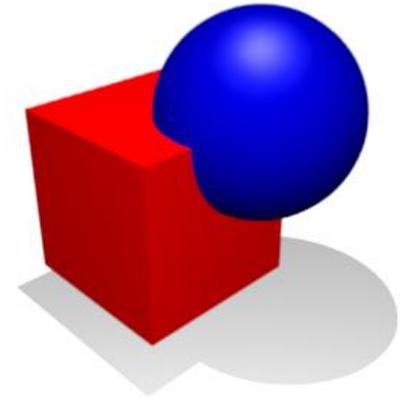
```
Vector3[] vertices = new
Vector3[4]
{
    new Vector3(0, 0, 0),
    new Vector3(1, 0, 0),
    new Vector3(1, 1, 0),
    new Vector3(0, 1, 0),
};

int[] triangles =
new int[6] {0, 2, 1, 0, 3, 2};

var createdMesh = new
Mesh();
createdMesh.vertices =
vertices;
createdMesh.triangles =
triangles;
```

Конструктивная сплошная геометрия (CSG)

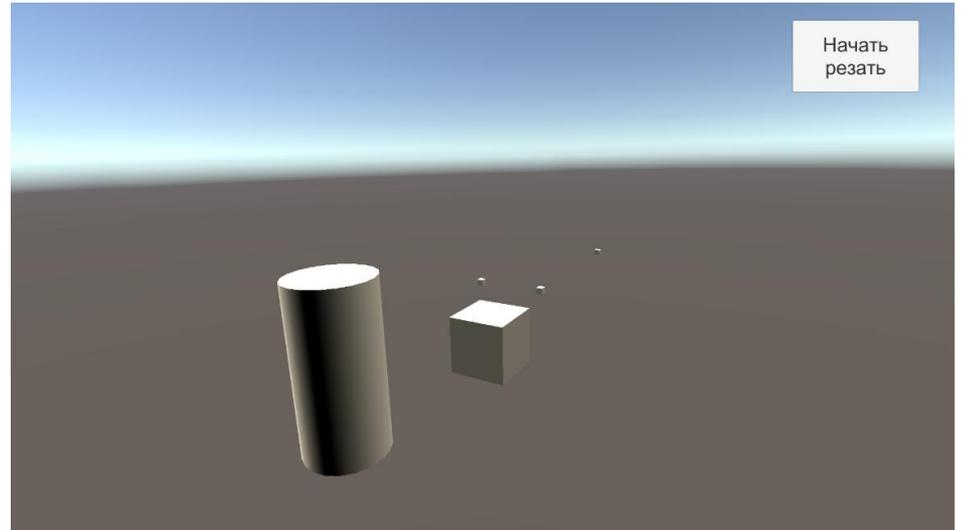
Технология, используемая в моделировании твёрдых тел. Конструктивная блочная геометрия зачастую является способом моделирования в трёхмерной графике и САПР. Позволяет создать сложную сцену или объект с помощью битовых операций для комбинирования нескольких иных объектов (объединение, пересечение, вычитание).



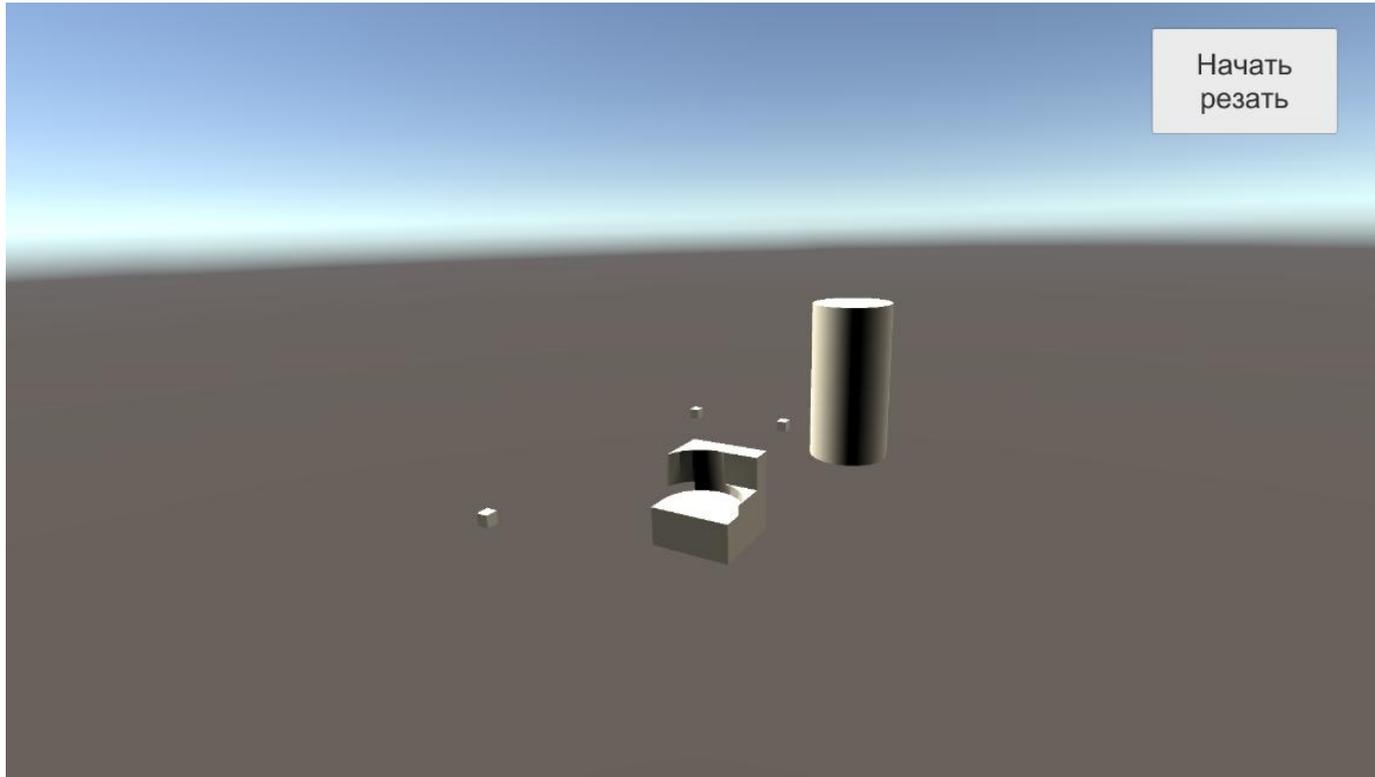
Обработка простой траектории

Цилиндр движется и режет большой куб.

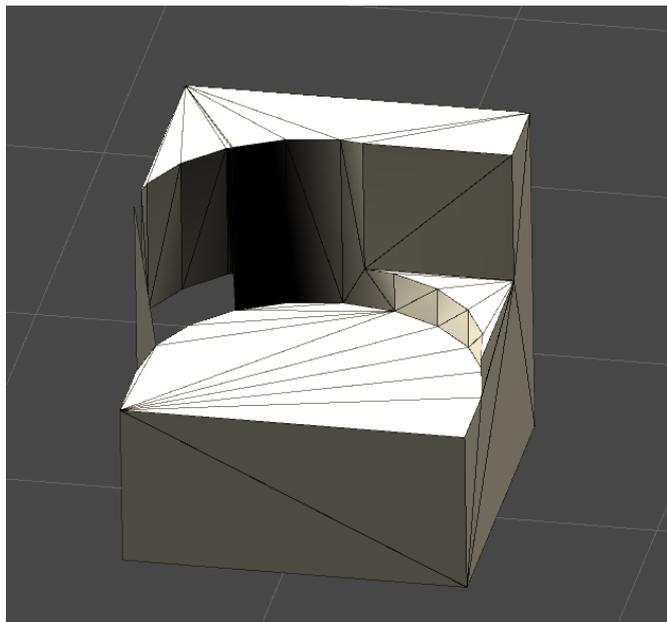
Маленькие кубики – точки траектории цилиндра.



После прохождения по траектории:



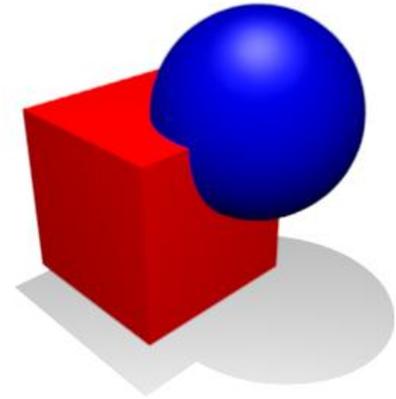
Результат обработки методом CSG



Количество точек в получившемся меше – 110.

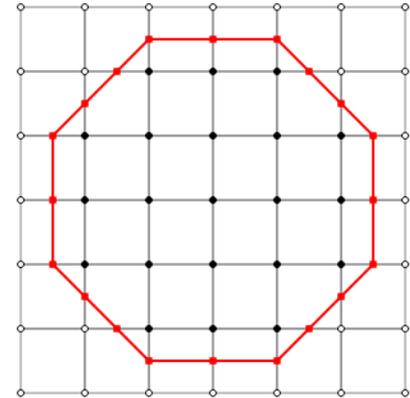
Недостатки метода CSG

- необходимо обрабатывать слишком большое количество точек;
- когда происходит обработка каждый кадр (примерно 30 раз в секунду), происходит проверка всех точек заготовки;
- иногда алгоритм генерирует лишние точки, а иногда некоторые пропускает.

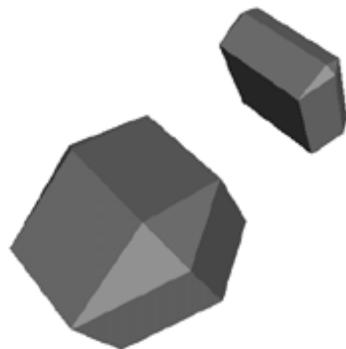


Реализация. Алгоритм «Marching Cubes»

Алгоритм пробегает скалярное поле, на каждой итерации просматривает 8 соседних позиций (вершины куба, параллельного осям координат) и определяет полигоны, необходимые для представления части изоповерхности, проходящей через данный куб. Далее на экран выводятся полигоны, образующие заданную изоповерхность.

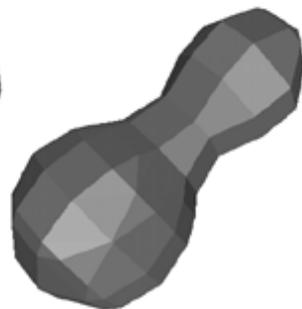


Детализация



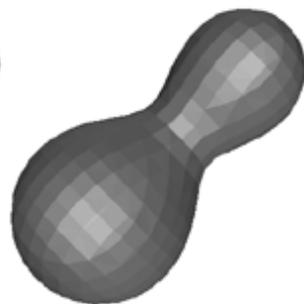
Grid size=10
70 Facets

Размер клетки = 10
70 полигонов



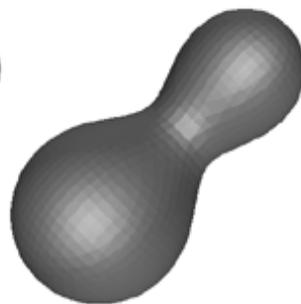
Grid size=5
220 Facets

Размер клетки = 5
220 полигонов



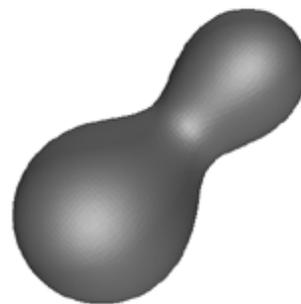
Grid size=2
1700 Facets

Размер клетки = 2
1700 полигонов



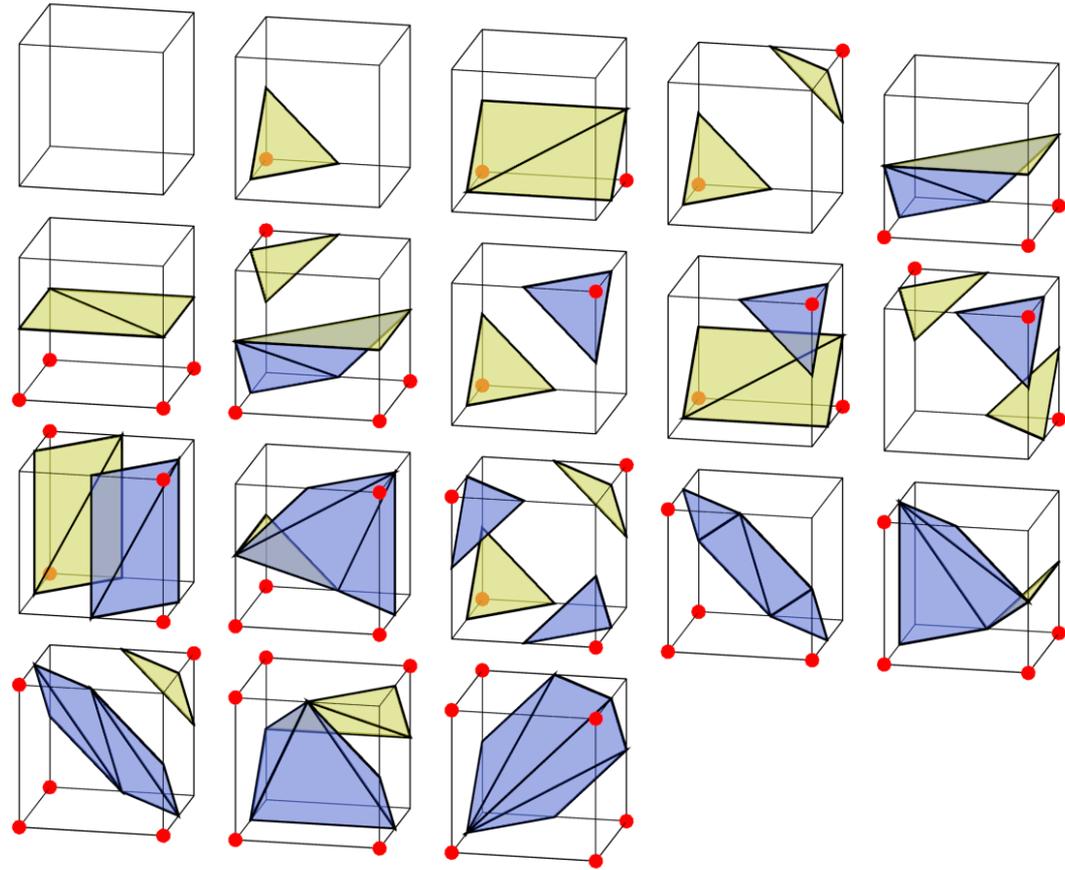
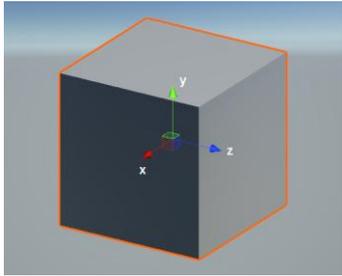
Grid size=1
6800 Facets

Размер клетки = 1
6800 полигонов

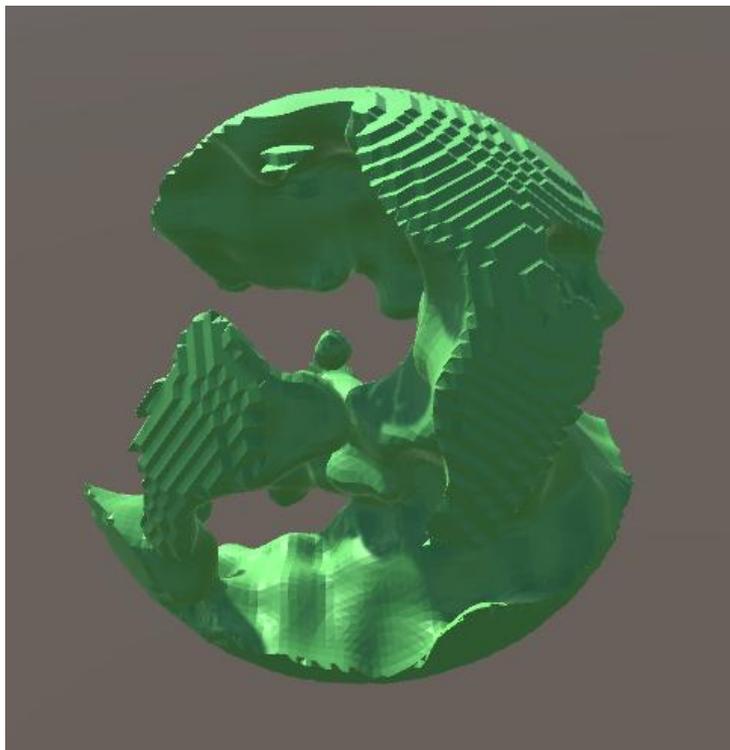


Grid size=0.5
27000 Facets

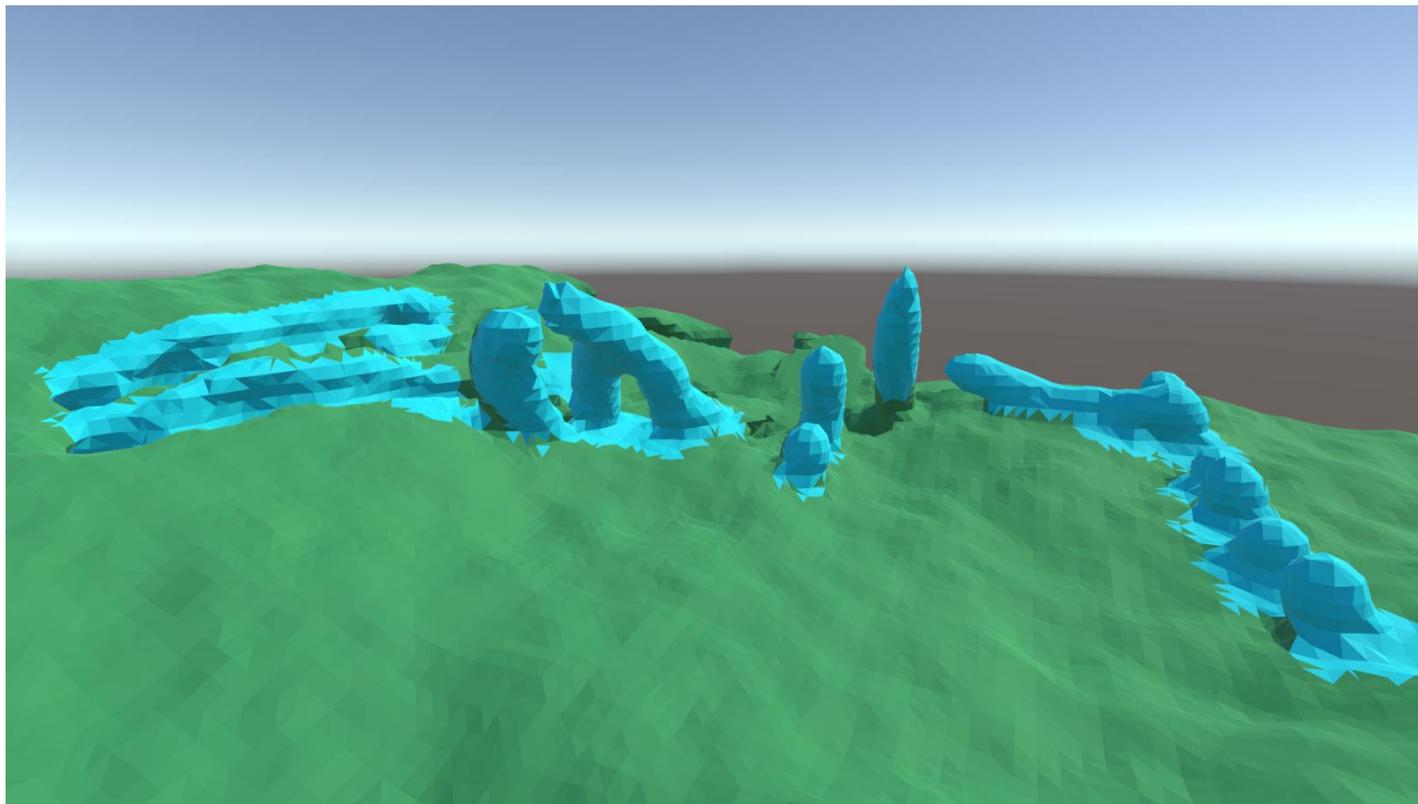
Размер клетки = 0.5
27000 полигонов



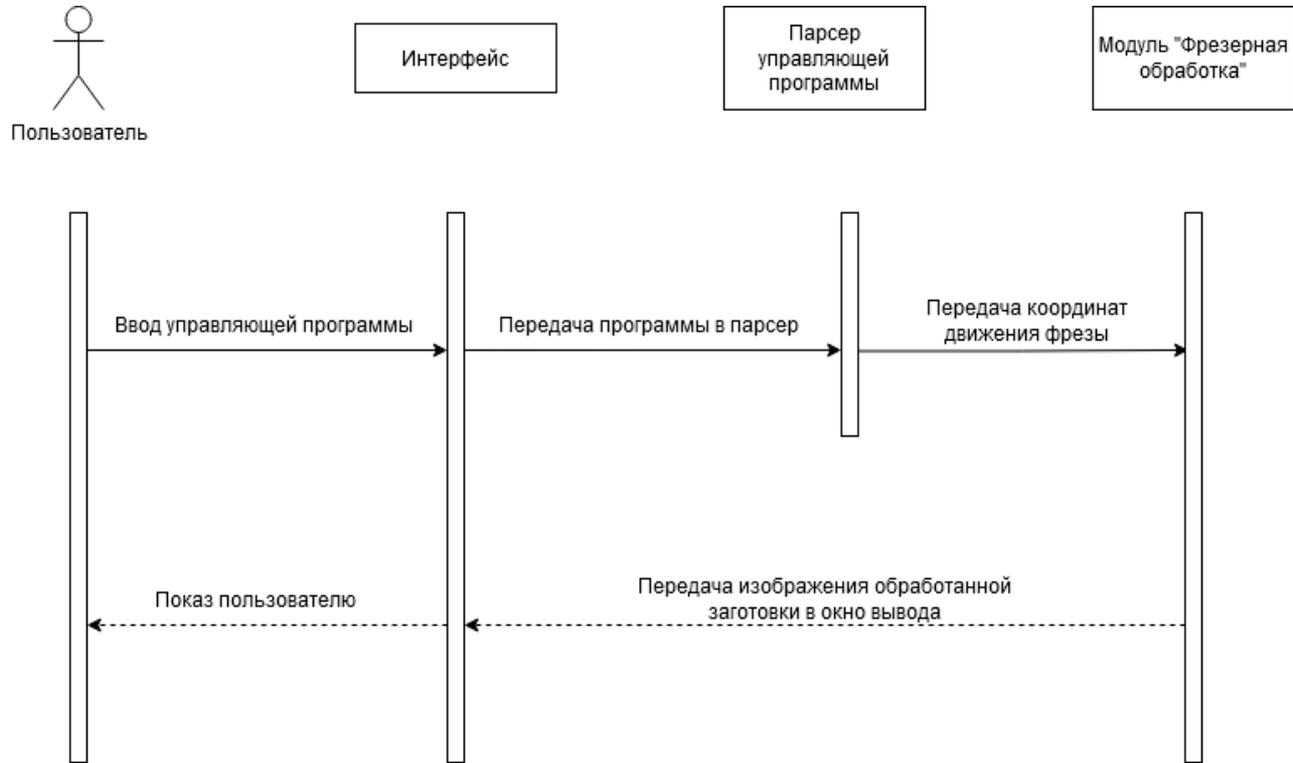
Алгоритм «Marching Cubes» от Eldemarkki



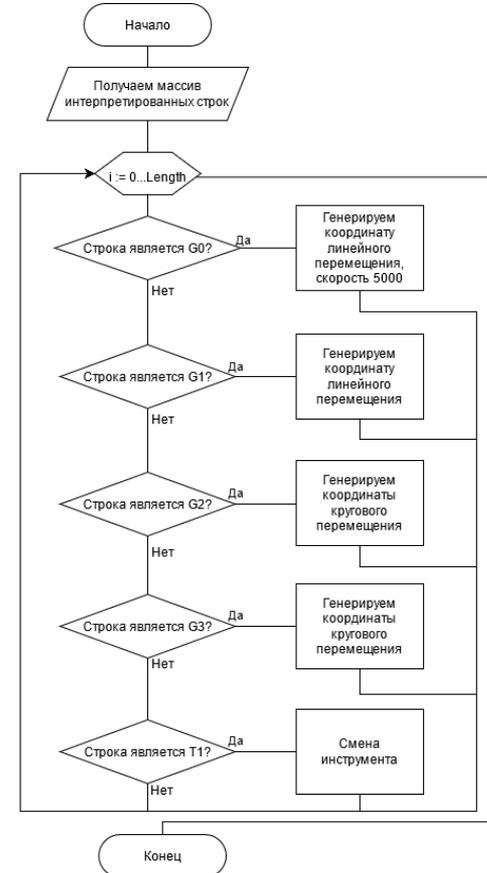
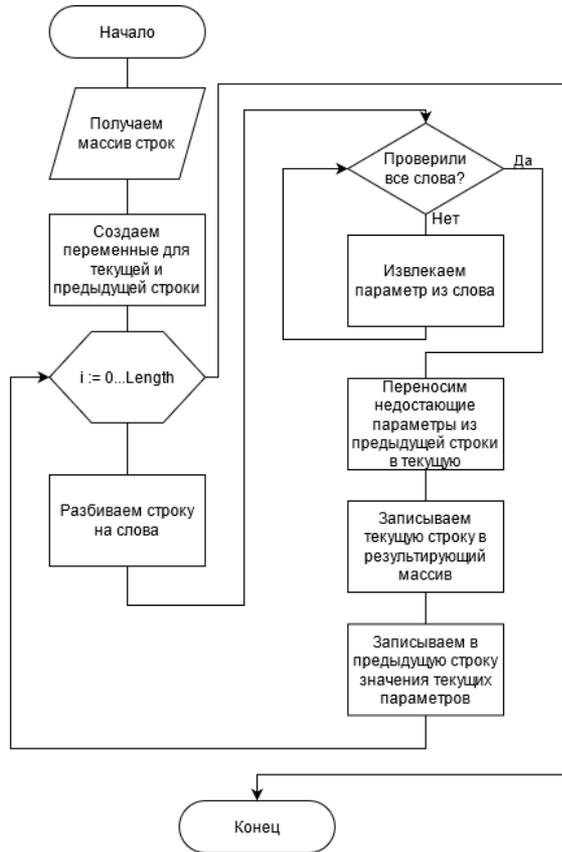
Алгоритм «Marching Cubes» от Eldemarkki



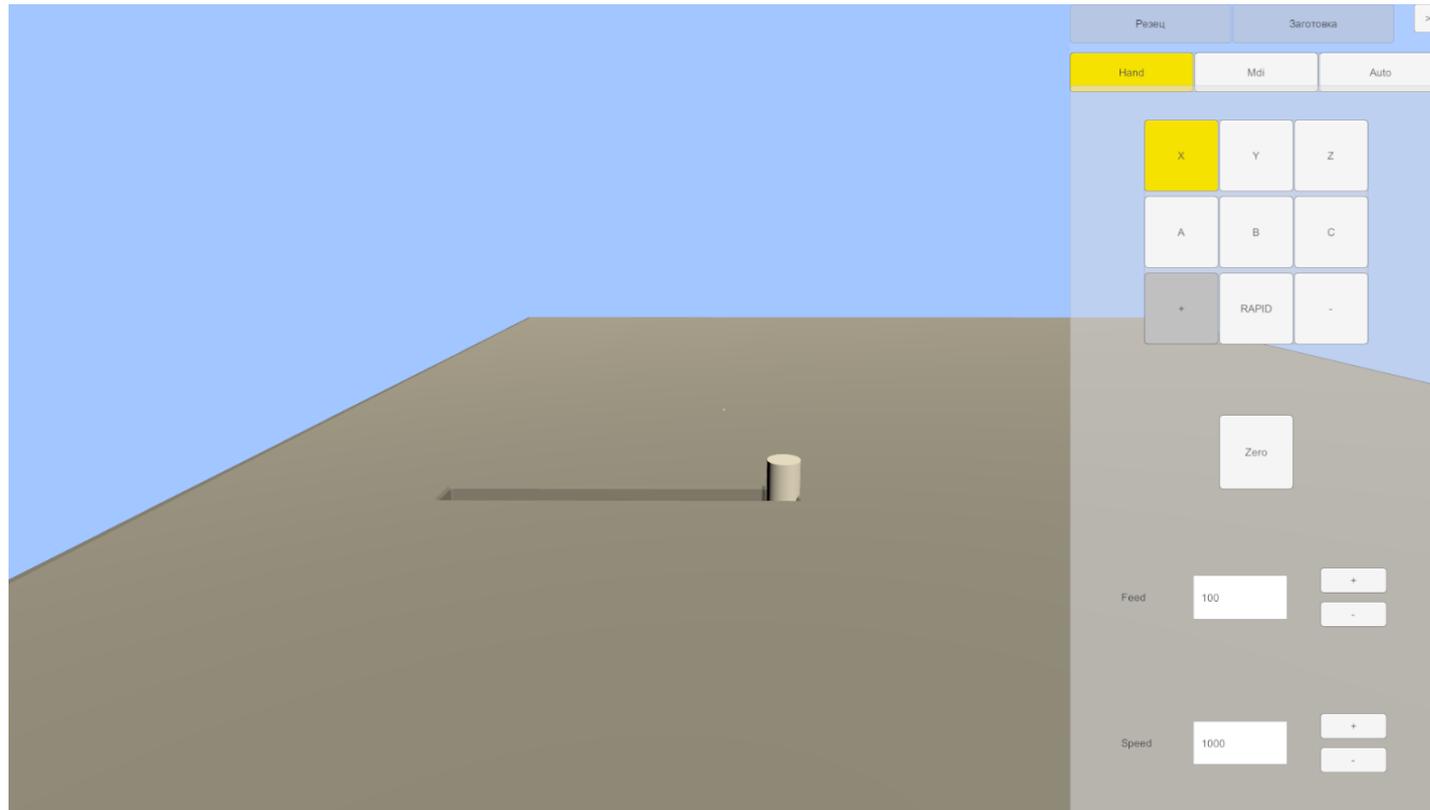
Архитектура приложения



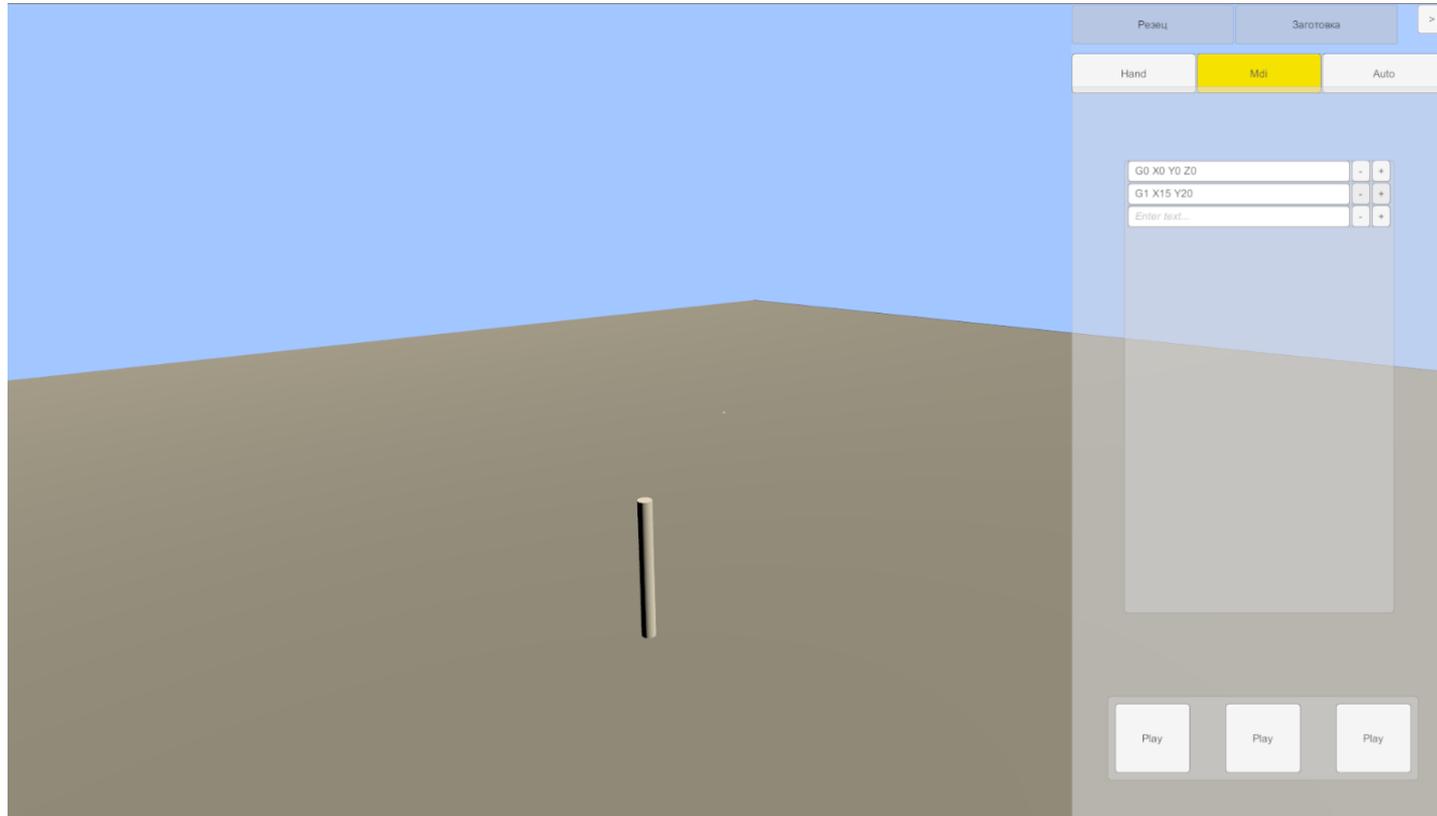
Парсер управляющих команд



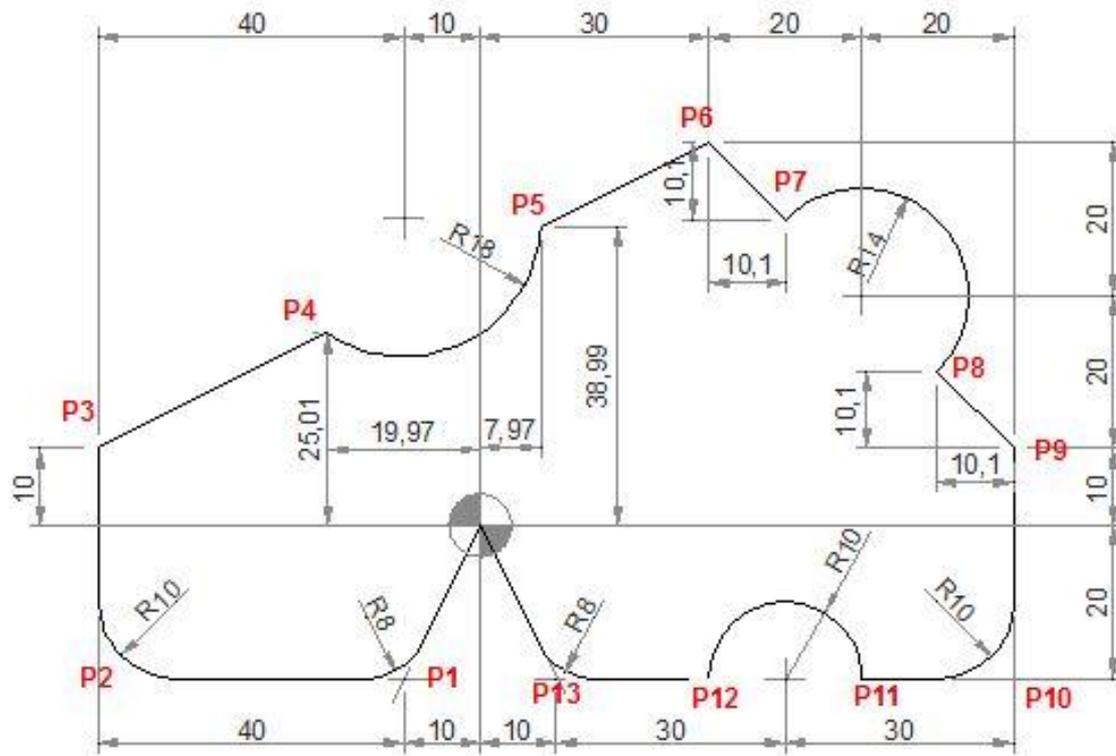
Реализация. Ручной режим



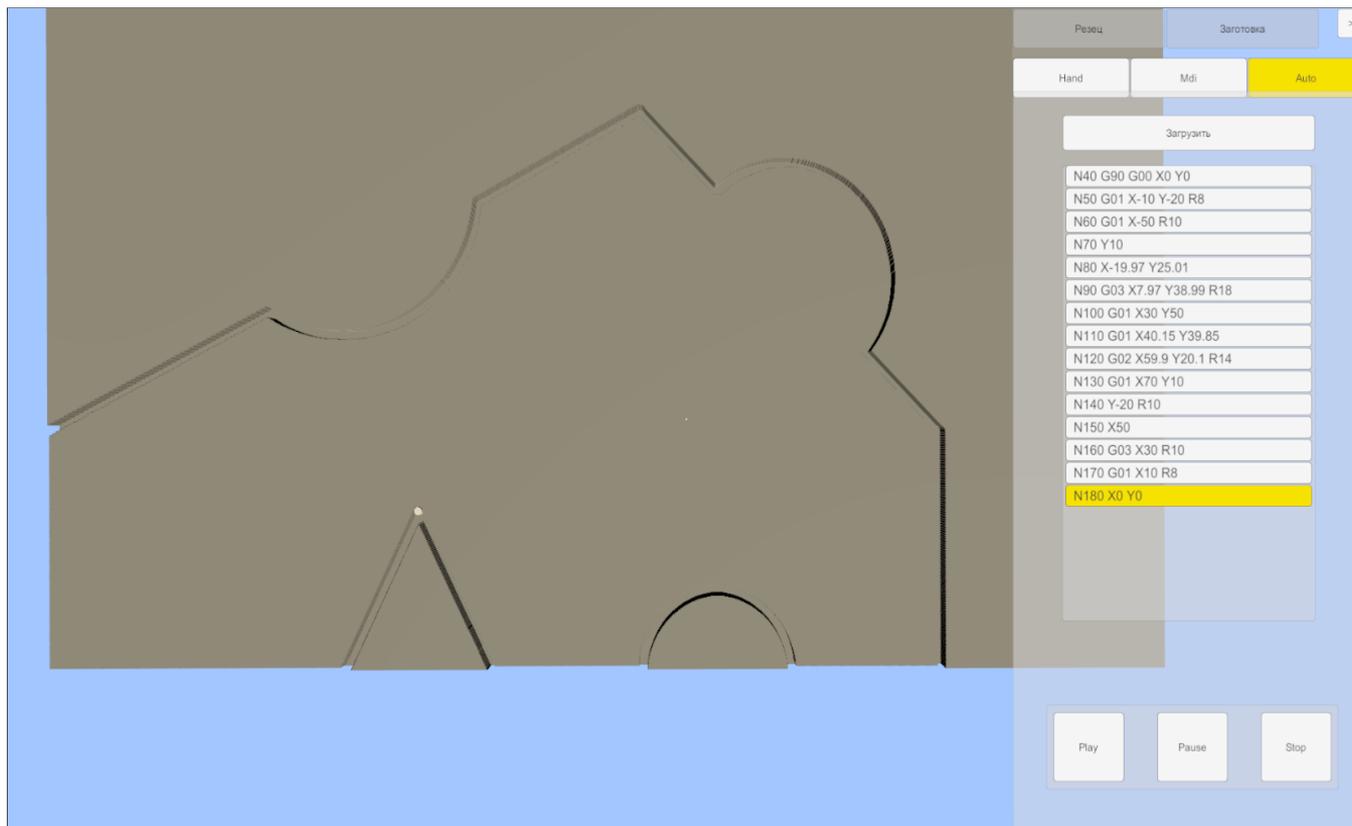
Реализация. Режим ручного ввода данных



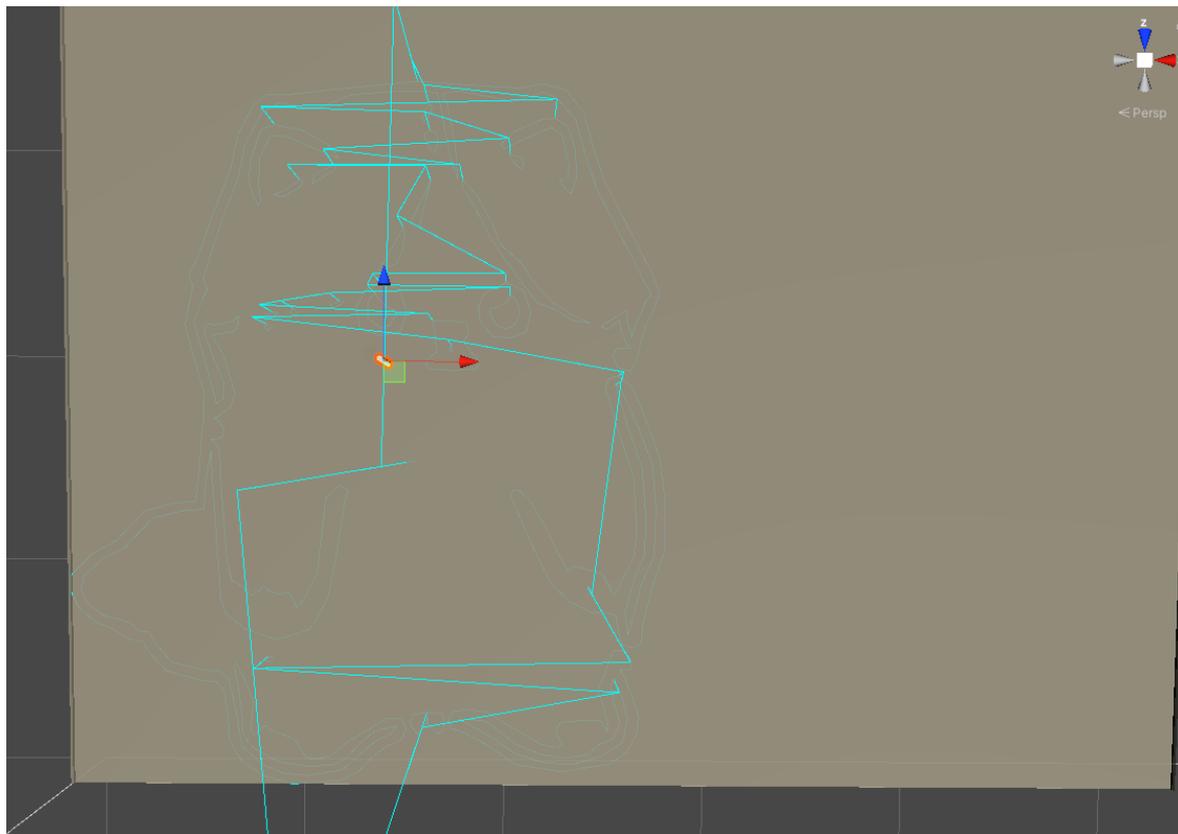
Реализация. Автоматический режим

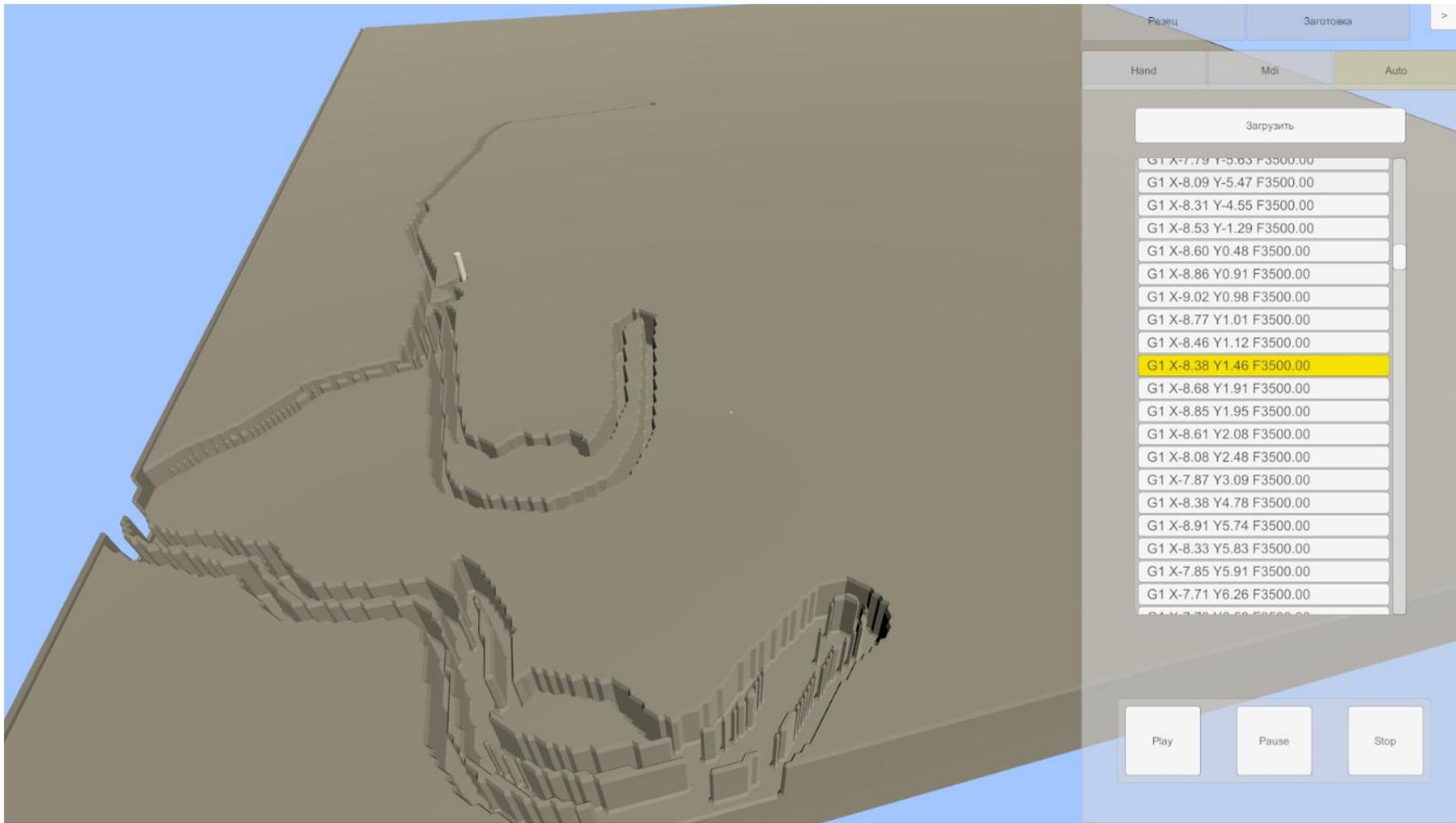


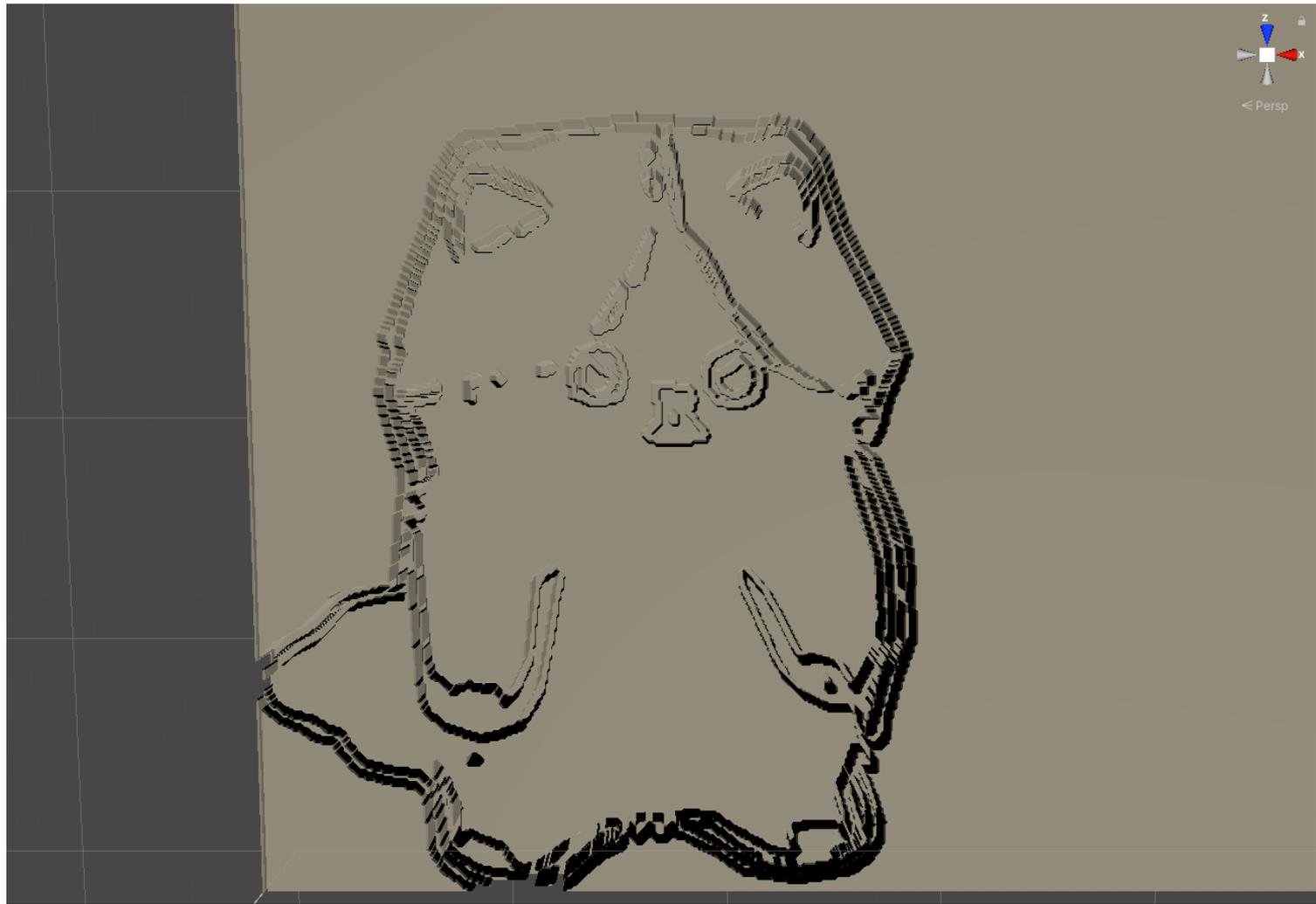
Реализация. Автоматический режим



Сложные файлы управляющих команд

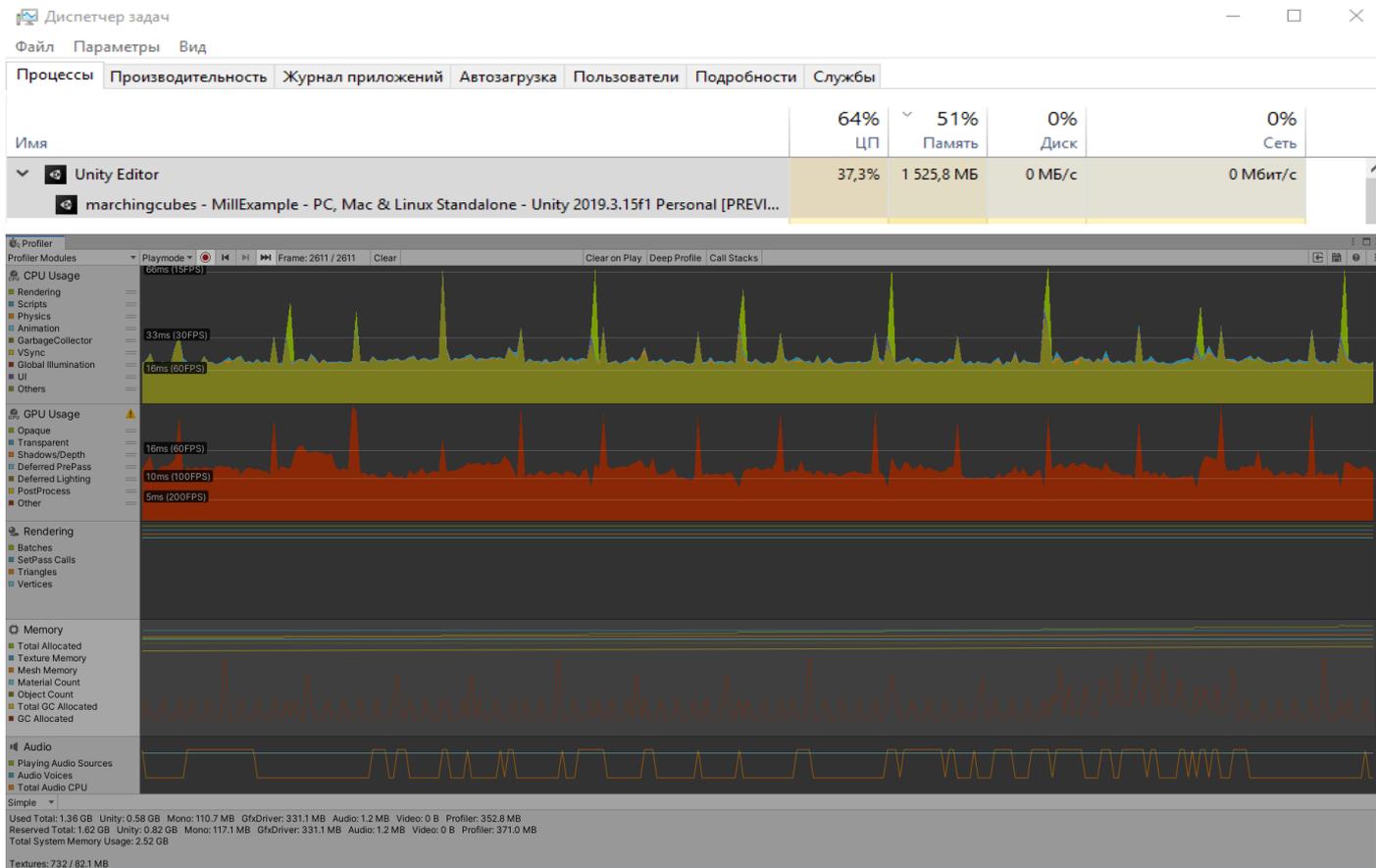








Производительность



Заключение

В ходе выполнения выпускной квалификационной работы были решены следующие задачи:

1. Проведен анализ аналогичных или похожих решений;
2. Проведен анализ существующих алгоритмов компьютерной графики для редактирования 3D моделей;
3. Разработан парсер управляющих программ фрезерных станков с ЧПУ;
4. Разработан модуль визуализации обработки 3D модели детали;
5. Проведено тестирование системы.

В результате, данный модуль эмуляции фрезерной обработки готов к внедрению в разрабатываемые эмуляторы фрезерных станков.



Разработка системы моделирования 5-осевой фрезерной обработки на Unity

Автор работы:

студент КЭ - 222

Захаров Д.И.

Руководитель работы:

к.т.н., доцент каф. ЭВМ

Парасич В.А.

Июнь, 2021 год