

Программное обеспечение для моделирования детского конструктора

Студент: Андреев А.А.

Группа: ИУ7-54Б

Руководитель: Погорелов Д.А.

Цели и задачи

Разработать программу моделирования склейки 3D-объектов детского конструктора Лего. Программа должна поддерживать следующие функции: вращение камеры вокруг сцены; добавление до двух источников света; добавление шаблонных заранее заданных, блоков конструктора, доступных в программе; перемещение выбранной фигуры конструктора из списка загруженных объектов в трех плоскостях; удаление любого блока. Каждый блок конструктора Лего должен обладать строгой жесткостью поверхности во избежание наложения объекта при перемещении. Размеры сцены задается пользователем и не может превышать 1920x1080. Размеры сцены пользователь задает при запуске программы.

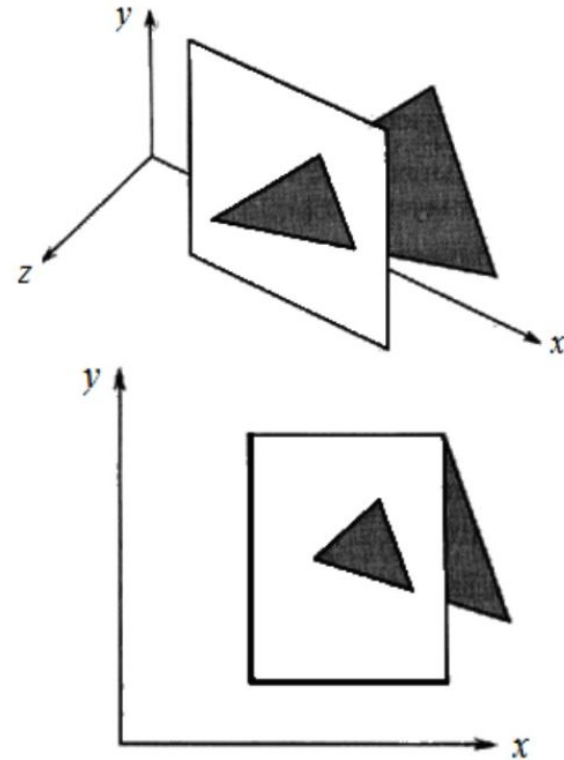
Графические алгоритмы

В ходе выполнения проект был проведен анализ потенциально применимых графических алгоритмов и выбраны более подходящие из них, а именно:

- Алгоритм z-буфер
- Метод тонирования Гуро
- Алгоритмы поворота и масштабирования при помощи матрицы поворота

Алгоритм z-буфера

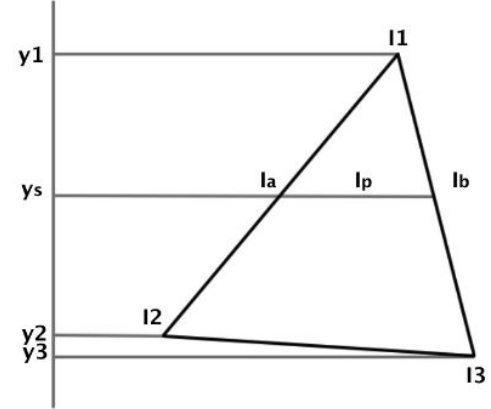
Принцип работы z-буфера заключается в том, что мы определяем глубину каждой отрисовываемой точки, сравниваем эту глубину с глубинами других точек, имеющие те же координаты X и Y , и в случае, если новая точка находится ближе к наблюдателю, тогда заносим координаты новой точки в z-буфер и цвет в буфер кадра.



Метод тонирования Гуро

Метод тонирования Гуро основан на интерполяции интенсивности, данный подход к закрашке объекта позволяет устранить дискретность изменения интенсивности.

Интерполяция интенсивности работает следующим образом: для всех ребер запоминается начальная интенсивность, изменение интенсивности при каждом шаге по координате y . Затем, заполнение видимого интервала производится путём интерполяции между значениями интенсивности на ребрах, ограничивающих интервал.

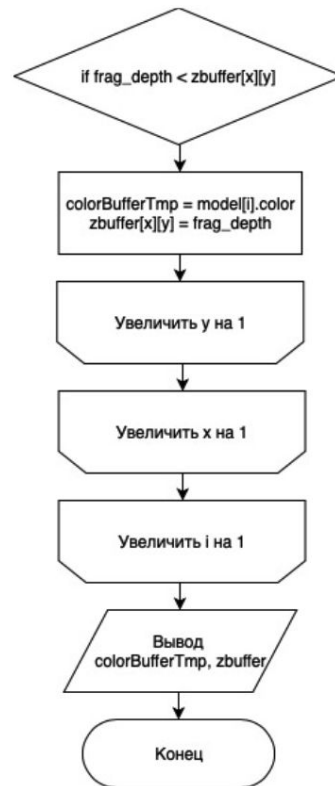
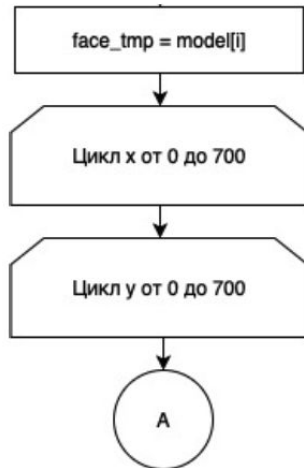
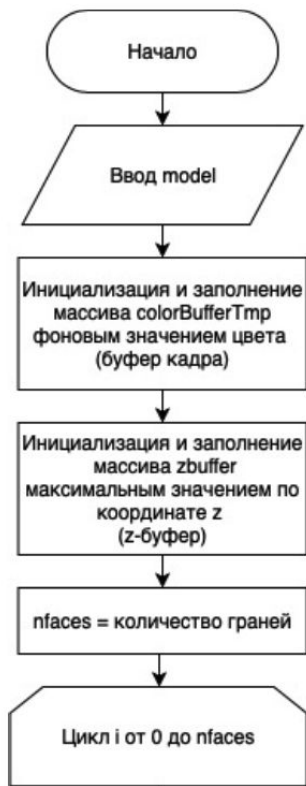


$$I_a = I_1 \frac{y_s - y_2}{y_1 - y_2} + I_2 \frac{y_1 - y_s}{y_1 - y_2}$$

$$I_b = I_1 \frac{y_s - y_3}{y_1 - y_3} + I_3 \frac{y_1 - y_s}{y_1 - y_3}$$

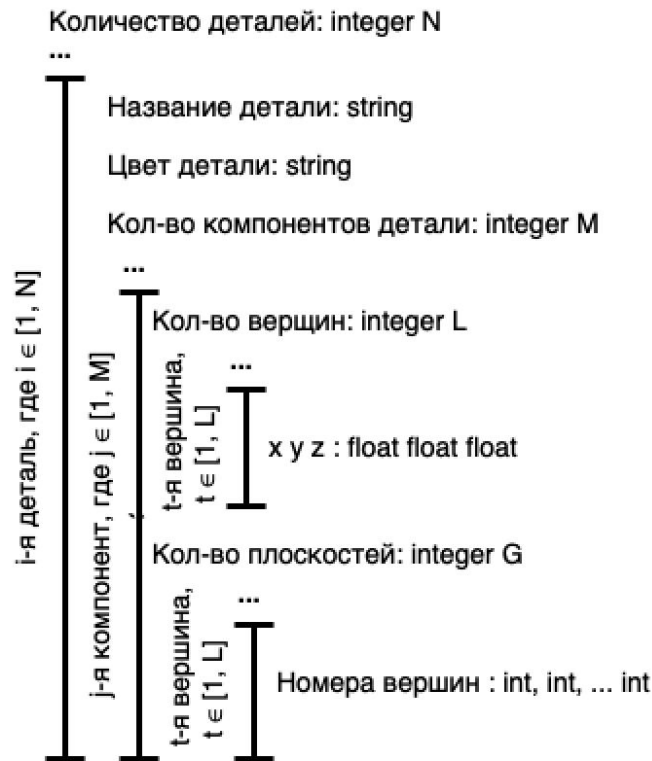
$$I_p = I_a \frac{x_b - x_p}{x_b - x_a} + I_b \frac{x_p - x_a}{x_b - x_a}$$

Алгоритм z-буфера



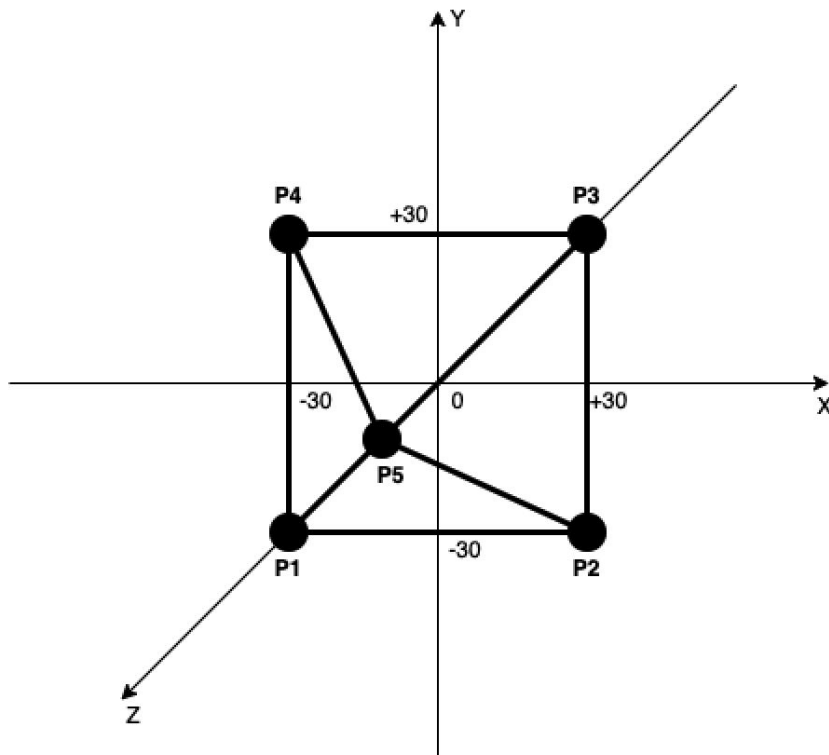
Формат файла

Был выбран собственный формат файла описания геометрии, потому что задача моделирования детского конструктора может требовать внесения дополнительных, неподдерживаемых стандартными общепринятыми форматами, данных, которые текущий формат позволит сделать. Основная причина выбора - доступность к модифицированию под специфическую задачу моделирования детского конструктора.



Формат файла

```
1
Пирамида
green
1
Пирамида
5
-30 -30 0
+30 -30 0
+30 +30 0
-30 +30 0
+0 +0 +30
6
4 1 0
4 2 1
3 4 0
3 4 2
3 2 1
3 1 0
```



Из листинга 2.5.1:

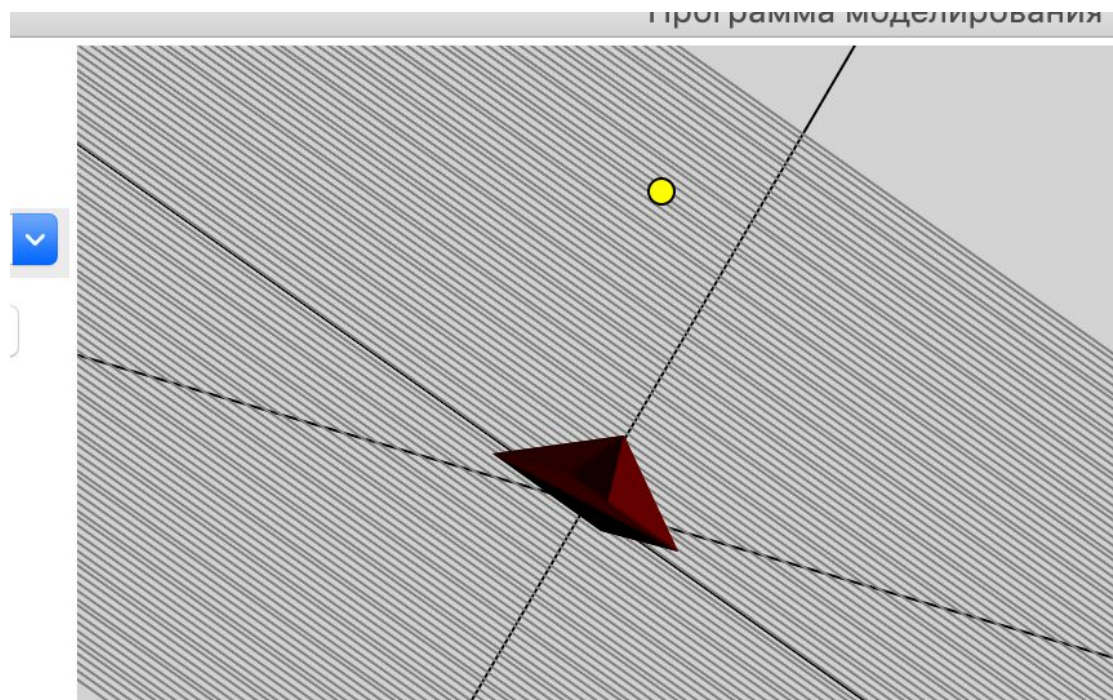
P1-P5 - Набор вершин

P1(-30, 30, 0)
P2(+30, 30, 0)
P3(+30, 30, 0)
P4(-30, 30, 0)
P5(0, 0, +30)

Набор связей, как плоскости
по вершинам:

P5 P2 P1
P5 P3 P2
P4 P5 P1
P4 P5 P3
P4 P3 P2
P4 P2 P1

Формат файла



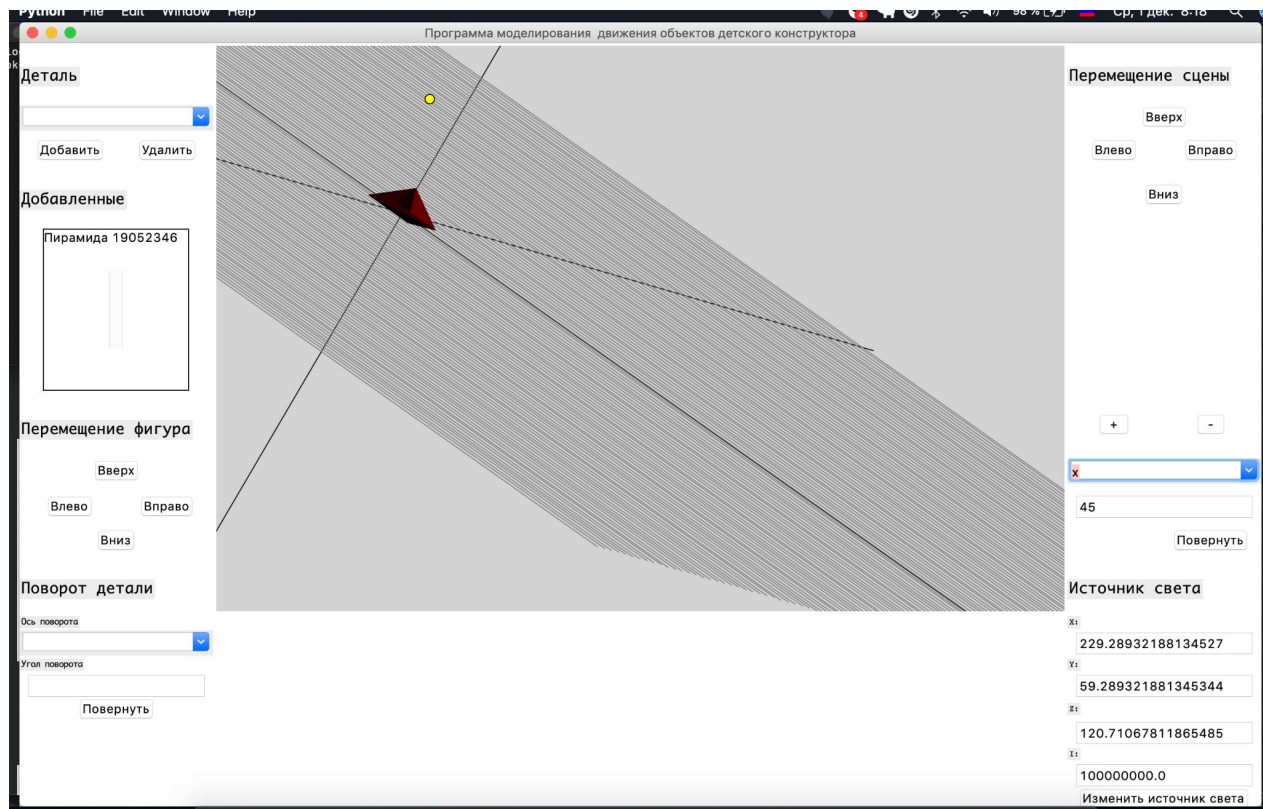
Пользовательский интерфейс

Программа позволяет пользователю:

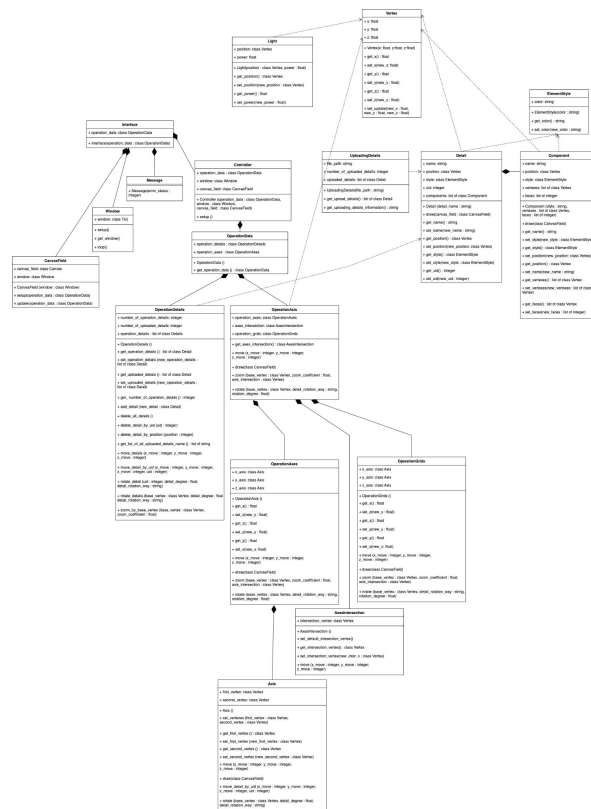
- выбирать и загружать/удалять деталь из списка загруженных деталей
- вращать, перемещать выбранную деталь.
- вращать и перемещать сцену.
- менять позицию и интенсивность источника света.

В папку *input_details* можно поместить файл с любой комбинацией деталей и компонентов, соответствующей установленному стандарту.

Пользовательский интерфейс



Парадигма ООП в реализации



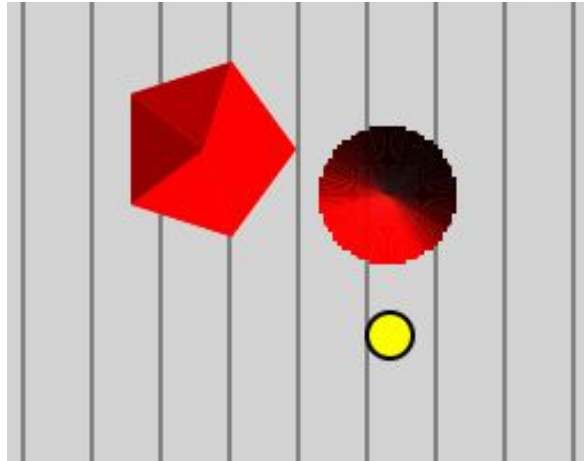
Парадигма ООП в реализации

Для удобства представление и обработки данных в рамках поставленной задачи были выделены специально разработанные собственные объекты, каждый из которых наделен определенным функционал в соответствии с парадигмой ООП.

Такая модель позволяет достаточно быстро наращивать дополнительный функционал у программы.

Представление деталей

Каждый компонент детали представляется, как набор примитивов - треугольников.



Пример отрисовки цилиндров (сверху) с разной детализацией.

Результаты проекта

Была разработана программа моделирования детского конструктора. Хранение объектов было реализовано в .txt файлах. Отображение каждого объекта на сцене было реализовано при помощи алгоритма z-буфера.

Реализованные задачи:

- процедура чтения из файлов формата .txt;
- создание объектов сцены;
- алгоритм z-буфера;
- метод тонирования Гуро для закраски объекта;
- поворот и перемещение выбранных деталей, сцены целиком;
- камера и источник света;
- пользовательский интерфейс.
- изменение данных источника света;

Результаты проекта

