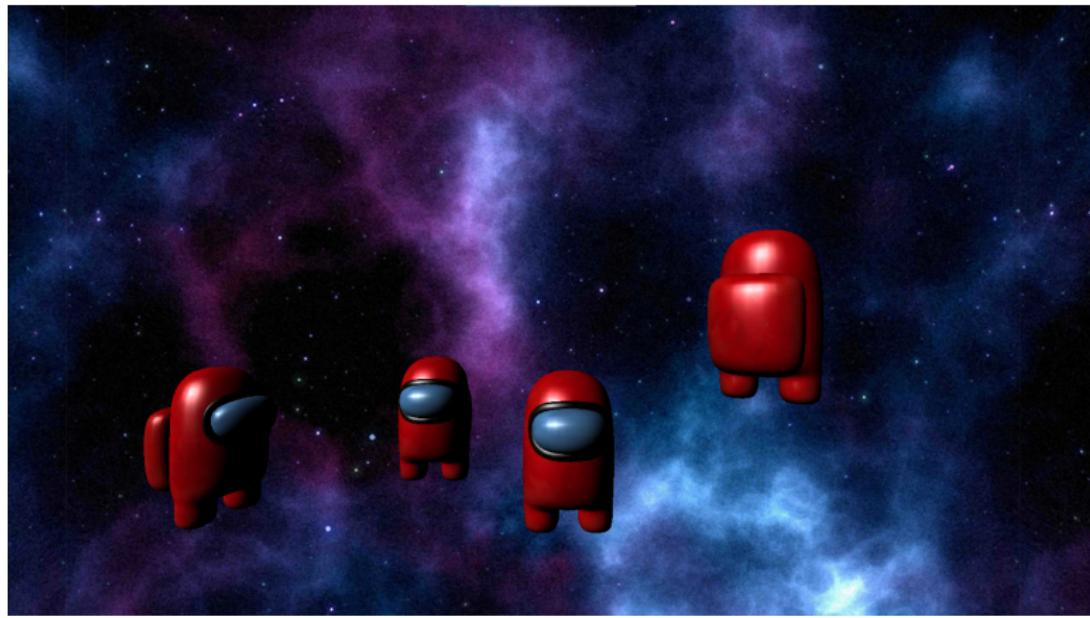


# 3D RENDERER С НУЛЯ

<https://github.com/dmitrysluch/3dRenderer/tree/develop>



# Задачи и реализованный функционал

Обязательные требования:

- Отрисовка 3d моделей
- Перспективная проекция
- Z-Buffer
- Клиппинг
- Интерактивное приложение

# Задачи и реализованный функционал

Обязательные требования:

- Отрисовка 3d моделей
- Перспективная проекция
- Z-Buffer
- Клиппинг
- Интерактивное приложение

Дополнительно реализовал:

- Текстурирование (Билинейная фильтрация)
- Освещение (Phong/Blinn-Phong)
- Интерфейс для пользовательских материалов

# Конвейер

- Переход между пространствами
  - Объекта
  - Мира
  - Камеры
  - Homogenous Cull Space
  - Экрана

# Конвейер

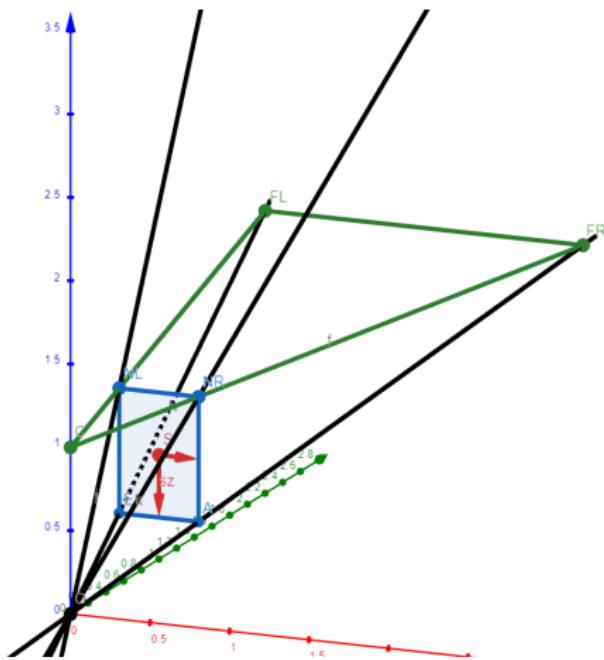
- Переход между пространствами
  - Объекта
  - Мира
  - Камеры
  - Homogenous Cull Space
  - Экрана
- Растеризация

# Конвейер

- Переход между пространствами
  - Объекта
  - Мира
  - Камеры
  - Homogenous Cull Space
  - Экрана
- Растеризация
- Пиксельный шейдер

- Переход между пространствами
  - Объекта
  - Мира
  - Камеры
  - Homogenous Cull Space
  - Экрана
- Растеризация
- Пиксельный шейдер
- Тесты (например глубины)

# Проекция



# Push vs Pull

## Pull

- Каждый кадр опрашивается клавиатура и в зависимости от этого изменяются модели
- Каждый кадр на экран отрисовывается текущее состояние моделей
- Цикл активного ожидания и отрисовки

## Push

- Observable и уведомления
- Контроллеры обновляют модели
- Обновление модели вызывает перерисовку
- Нет активного ожидания

# Моя реализация

- Renderer - низкоуровневый
- Kernel и модели - высокоуровневая библиотека

# Моя реализация

- Renderer - низкоуровневый
- Kernel и модели - высокоуровневая библиотека
- Ядро - владеет моделями - умеет отрисовать сцену
- Модели - ссылаются на ядро
- Контроллеры - меняют модели - `kernel.GetComponent<T>()`
- Меши и материалы - константны, `SceneTransform` - proxy