1. Надо продумать загрузки моделей
2. Нужно научиться грузить текстурки

То есть если это в glut правда легко, юзаем его. Если это запарно, то просто на opengl быстро все напишем

И вообще внешний класс Application, который можно сделать синглтоном

1. Система юнитов

Точно нужен класс сцены

1. Загрузиться из файла, сохраниться в файл (?)
2. Добавить юнита, удалить юнита
3. Нарисоваться
4. Обновляться
5. Загрузиться – ну просто продумать формат.

Сцена должна рекурсивно вызывать для каждого юнита. Тогда у каждого юнита должна быть функция сереализации. Тоже получается в базовый интерфес. Но так он должен уметь и дереалсиерилизироваться.

1. Добавлять

std::map<std::string, std::shared\_ptr<Unit>> - структура данных

1. Просто рекурсивно рисует всех юнитов

Нужен класс юнитов – он будет полиморфным (с вирт функциями)

1. Констуктор/деструктор (деструктор вирутальный!!!)
2. Обновляться (Update) – виртуальная
3. Рисоваться (Draw) – виртуальная
4. (Де)сереиалиация

Конкреткные юниты:

class Person

Есть поле position.

Можно взять его позицию. *vec2 get\_position()*.

Можно заставить говорить (оповестить об этом) *void Talk(Unit \*talker)*.

Если мы стоим рядом с ним, он выбрасывает фразу (какую именно он решает сам).

1. Рендер
2. Юниты, сцены, и продумать все что может понадобиться для итоговой игры