СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc104743887)

[Термины, определения и сокращения 5](#_Toc104743888)

[1 Анализ проектируемого Quake-подобного игрового движка 6](#_Toc104743889)

[1.1 Принцип разработки 2.5D и 3D пространства на примере игры Quake, и версий разработанных движков поколения Quake 6](#_Toc104743890)

[1.2 Эволюция разработки рендринга компьютерной графики 8](#_Toc104743891)

[2 Разбор и применение графических методов 11](#_Toc104743892)

[2.1 Выбор пакета аппаратно-программной прорисовки OpenGL 11](#_Toc104743893)

[2.2 Использование двоичного разбиения пространства или Binary space partitioning, дополнение к разбрасыванию лучей 11](#_Toc104743894)

[2.3 Техническое задание проекта 13](#_Toc104743895)

[3 Разработка проекта графической системы 14](#_Toc104743896)

[3.1 Изучение и использование свободных библиотек для конструкции уровня 14](#_Toc104743897)

[3.2 Доработка и внесение изменений к движку 14](#_Toc104743898)

[3.3 Создание и рендеринг мира 14](#_Toc104743899)

[3.4 Установка управления 14](#_Toc104743900)

[3.5 Применение текстур к объектам 14](#_Toc104743901)

[4 Тестирование игрового движка 15](#_Toc104743902)

ВВЕДЕНИЕ

Изучение дисциплины и самого курса компьютерной графики основывается на исследовании основных алгоритмов, на которых строится базовое построение машинной графики. Возникнув из потребностей рынка, развития информатики и вычислительной техники, компьютерная графика изучает методы построения изображений различных геометрических объектов и сцен.

Главными этапами построения изображения являются:

* моделирование математически описанного объекта и сцены;
* визуализация построения изображения объемного мира.

Целью данного курсового проекта является изучение такого понятия, как «игровой движок», и применить полученные знания для реализации quake-подобного игрового движка. Для достижения цели необходимо изучить методы обработки трехмерной графики в реальном времени, и применить основные технические характеристики, а так же главный алгоритм используемый для рендеринга сцены – рейкастинг или метод «бросания лучей».

Следовательно, задачи курсовой работы можно обозначить в следующих пунктах:

* изучить методы разработки quake engine для применения паттернов при разработке собственного движка;
* поэтапно выстроить сцену для реализации выбранного алгоритма:

1. применить первый этап: пробросить лучи для отображения препятствий;
2. применить второй этап: внести отображения «шейдеров» для препятствий;

* обозначить «нижнее» ограничение и «верхнее», прописать основные выполняемые действия;
* обозначить основное управления на сцене.

В рамках курсового проекта будет использоваться графическая библиотека OpenGL, согласно ранее выданному заданию. Разработка игрового движка будет осуществляться на языке программирования C.

Так же, дополнительно необходимо отметить, что при разработке сцены при использовании графики OpenGL, используется долго поддерживающее расширение FreeGlut, которое работает с С-подобным языком, как в данном случае Си.

Актуальность данной работы заключается в исследовании методов для разработки собственного движка, основываясь на конкретных паттернах ранее революционных идей, которые на текущий момент требуют доработки при условии развитии алгоритмических возможностей и компьютерной техники.

Термины, определения и сокращения

BSP, binary space partitioning – метод двоичного разбиения пространства

1 Анализ проектируемого Quake-подобного игрового движка

1.1 Принцип разработки 2.5D и 3D пространства на примере игры Quake, и версий разработанных движков поколения Quake

Перед объяснением феномена псевдотрехмерной графики и полноценной трехмерной, необходимо определить, на чем основываются данные законы. Главной основой является понятие евклидова пространства, которое подразумевает наличие размерности равное трем, то есть трехмерное.

В основе игрового движка Quake или Quake engine лежит основа как псевдотрехмерной графики, так и основа полноценной 3D графики. Данное утверждение верно в отношении первого поколения разработанных игровых движков для DOOM, который использовал полноценный ray casting и малую часть BSP-деревьев, основа которых лежит в разбиении основного пространства на две плоскости. Данный принцип тянется из математической модели евклидова пространства. Основными показателями, при которых считается что метод ray casting, или пробрасывание лучей, и BSP-деревья относятся к методам трехмерной компьютерной графики – это определение столкновения и сортировка визуальных объектов.

Считать рейкастинг полноценным методом рендеринга, как 2.5D пространства нельзя, поскольку существуют принципиальные различия. Такие как, пример передвижение игрока лишь по двум осям, но и рейкастинг не поддерживает перемещение вверх и вниз, тогда как полноценная трехмерная графика подразумевает и данное передвижение. Исходя из данного факта, можно выделить следующее - рейкастинг является псевдотрехмерной графикой от первого лица, в которой лучи пробрасываются для каждого вертикального среза экрана, это выглядит следующим образом, представлено на рисунке 1.

Дополнительно следует отметить, что метод с векторами работает в том плане, при котором обязательным является использование 2D карты, которая является полноценной плоскостью для реализации пространства.

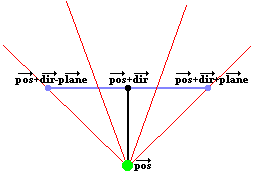


Рисунок 1 – Пробрасывание лучей по вертикали

Исходя из вышеописанных данных, следует рассмотреть маппинг, и где именно происходит передвижение в псевдопространстве, данное необходимо для рассмотрения следующего метода, который используется как в open source Quake engine, так и при проектировании курсового проекта.

В плане маппинга, в 2.5-мерном пространстве строится на 2D карте, которая и формирует полноценный уровень, пример на рисунке 2.

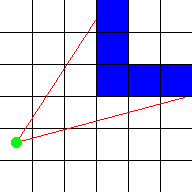


Рисунок 2 – Маппинг

Двумерная карта является как визуализацией уровня, так и формируемым пространством.

Далее необходимо вернуться, и полноценно дать определение рейкастингу, как основополагающему методу при разработке.

Основная идея рейкастинга заключается в следующем: карта представляет собой двухмерную квадратную сетку, и каждый квадрат может иметь либо 0 (= отсутствие стены), либо положительное значение (= стена с определенным цветом или текстурой). При разработке примера, в данном случае, шейдер использоваться не будет (добавление текстур для рендеринга).

Для каждого N экрана (то есть для каждой вертикальной полосы экрана) отправляется луч, который начинается в местоположении игрока и имеет направление, которое зависит как от направления взгляда игрока, так и от координаты N экрана. Следующим этапом, прописанными командами по 2D-карте позиция player или луч трассировки передвигается по плоскости, пока он не столкнется с квадратом карты, который является стеной. Если он ударился о стену, рассчитайте расстояние от этой точки попадания до игрока, то это расстояние используется для того, чтобы рассчитать, насколько высоко эта стена должна быть отрисована на экране: чем дальше стена, тем меньше она на экране и тем ближе, тем выше он кажется. Это все касается 2D расчетов, которые являются основными методом построения плоскости при управлении игроком.

Сама разработка игры Quake полноценно использует следующее поколение рендеринга, не основанного на разбиении двоичного бинарного дерева, то есть рассматривается пространство не как плоскость на 2D карте, а полноценное евклидово пространство.

1.2 Эволюция разработки рендринга компьютерной графики

Под эволюцией редринга используется основное понятие, как изменение графики из псевдотрехмерной в полноценную трехмерную графику. Данное стало возможным после применения рендеринга на 2D карте, как это рассматривалось ранее, в предыдущем пункте.

Рассматриваемая трехмерная игра, в которой происходит игра, называется картой или maps, то есть, сейчас она не является двумерной, как ранее. Программа редактора карт использует ряд простых выпуклых трехмерных геометрических объектов, известных как кисти или brushes, которые масштабируются и поворачиваются для создания среды. Кисти размещаются и ориентируются для создания замкнутого, пустого, объемного пространства, и когда дизайн завершен, карта проходит через препроцессор рендеринга. Препроцессор используется для обнаружения двух типов пустого пространства на карте: пустого пространства, окруженного кистями, где будет проходить игра, и другого пустого пространства за пределами кистей, которое игрок никогда не увидит. Затем препроцессор удаляет задние поверхности отдельных кистей, которые находятся за пределами игрового пространства, оставляя только несколько полигонов, определяющих внешний периметр закрытого игрового пространства.

Как правило после того, как карта была предварительно обработана, ее нельзя повторно отредактировать обычным образом, поскольку исходные кисти были разрезаны на мелкие кусочки. Вместо этого исходные данные редактора карт с кистями сохраняются и используются для создания новых версий карты. Но можно редактировать обработанную карту, открыв ее в специальном редакторе вершин и отредактировав исходные данные вершин, или добавить или удалить отдельные грани треугольников.

Этот этап предварительной обработки не может работать, если есть какие-либо небольшие дыры или «утечки», которые соединяют внутреннее игровое пространство с внешним пустым пространством, и часто сложные проекты по построению карты отбрасывались, потому что дизайнер карты не мог найти утечку в их карта. Во избежание протечек щетки должны перекрываться и слегка проникать друг в друга; попытка идеально выровнять края кистей необычной формы на сетке может привести к очень маленьким зазорам, которые трудно обнаружить.

Открытое небо в картах Quake на самом деле не открытое, а покрыто и окружено большими кистями и текстурировано специальной текстурой скайбокса, которая запрограммирована на использование сферического отображения, и поэтому всегда выглядит одинаково с любой точки зрения, давая иллюзия далекого неба, пример для разрабатываемого проекта на рисунке 3.

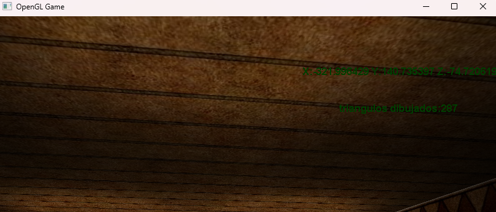


Рисунок 3 – Пример легкого использования «неба»

На рисунке 3 полноценно можно рассмотреть, что в качестве полноценного неба используется иллюзия, поскольку в рамках курсового проекта не предусмотрена работы с brushes, по этой причине будут использоваться стандартный набор свободно распространяемых текстур из одного из уровней игры Quake.

2 Разбор и применение графических методов

2.1 Выбор пакета аппаратно-программной прорисовки OpenGL

Проектирование игрового движка на свободно распространяемой библиотеки OpenGL была выбрана в следствии больших возможностей данной библиотеки, как в работе с освещением и рассеиванием света, поскольку основной задачей используемого алгоритма, как ray casting, так и его последующей эволюции в трехмерной графике – BSP, применяется работа со светом на генерируемой карте.

2.2 Использование двоичного разбиения пространства или Binary space partitioning, дополнение к разбрасыванию лучей

Основой двоичного разбиения послужила разработка предыдущей версий игр Doom и Quake – пробрасывание лучей. Данный метод модифицирован в стандартных библиотеках, в основу которых берется разработка рендеринга одного уровня (уровень в плоскости карты). Поскольку задача в курсовом проекте стоит иная, следует отметить, что в рамках разработки необходимо отдельно отметить вопрос об управлении, так как трехмерный движок, от разрабатываемого примера, управляется не только на клавиатуре, но и с использованием мыши – отличительная особенность поколения движков от предыдущих.

Если рейкастинг использует спрайты, то трехмерный мир использует полноценную модель и текстуры. Таким образом, необходимо определить пространство мира, и по какому принципу будет работать столкновение объекта (игрока) с другими объектами из пространства.

Так же перед проектированием отмечается следующая модель: по какому принципу игра будет формировать при загрузке мир.

На рисунке 4 представлена краткая схема отработки алгоритма при запуске движка:

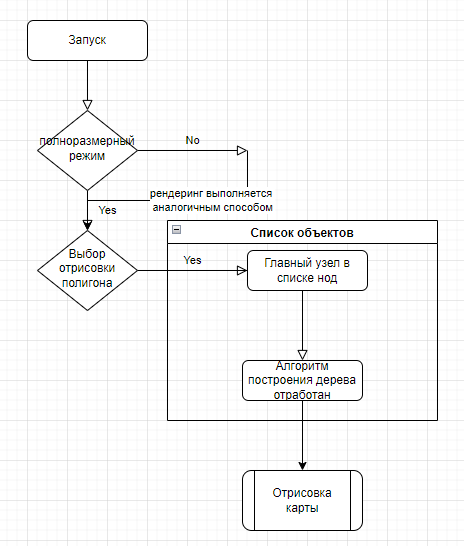


Рисунок 4 – Алгоритм отрисовки

Поскольку quake-подобный движок строится на алгоритме BSP-деревьев, как и для любого другого движка основой является построение бинарного дерева, то перед запуском в программе необходимо уточнение полно размерности. Связано это с количеством необходимых разбиений. Далее по алгоритму выбирается полигон, от которого будет проходить главный узел дерева при использовании гиперплоскостей. Таким образом, пространство заложенное в саму сцену будет разделено на два подпространства.

2.3 Техническое задание проекта

3 Разработка проекта графической системы

3.1 Изучение и использование свободных библиотек для конструкции уровня

3.2 Доработка и внесение изменений к движку

3.3 Создание и рендеринг мира

3.4 Установка управления

3.5 Применение текстур к объектам

4 Тестирование игрового движка