

Рендеринг псевдотрёхмерного пространства методом бросания лучей

Шорников Александр Евгеньевич, группа 05230

Бурятский государственный университет
Институт математики и информатики
Кафедра прикладной математики

Научный руководитель — асс. каф. ИТ **Брагин Александр Фёдорович**

Улан-Удэ
2016г.

Цель дипломного проекта:

Разработать инструментарий для создания интерактивных планов помещений для интернет-сайтов организаций:

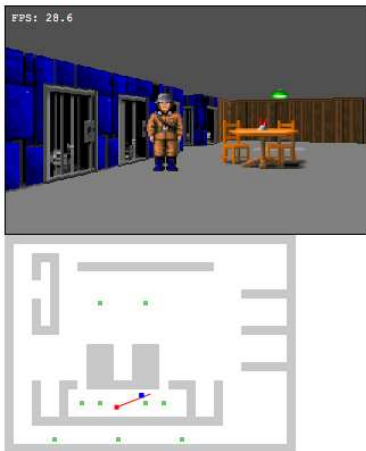
- редактор плана помещения;
- интерактивная визуализация;
- поиск оптимального маршрута на плане и визуализация его прохождения;
- создание интерактивной карты корпусов БГУ для сайта ИМИ.

Цель данной курсовой работы:

- выбор и изучение технологии рейкастинга;
- убедиться в возможности реализации;
- убедиться применимости для решения задачи проекта;
- реализовать прототип.

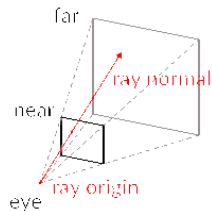
Технология рейкастинга

Метод бросания лучей(англ. Raycasting, "Рейкастинг") - это технология получения изображения по модели с помощью компьютерной программы, позволяющая создавать 3D перспективу в 2D картах. По сути, это метод преобразования ограниченной формы данных (очень простая карта этажа) в трёхмерную проекцию с помощью трассировки лучей из точки обзора в объём обзора.

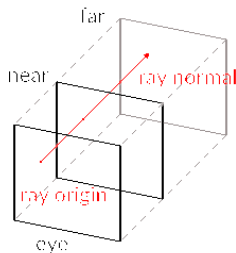


Основное преимущество технологии рейкастинга перед WebGL - поддержка любыми браузерами

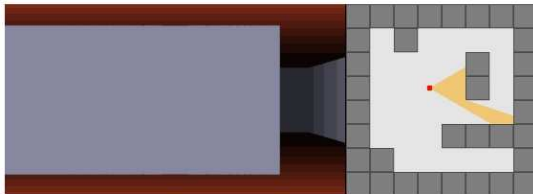
Perspective



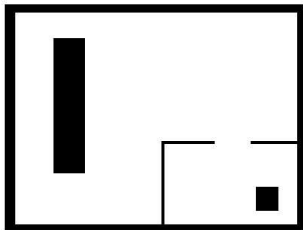
Orthogonal



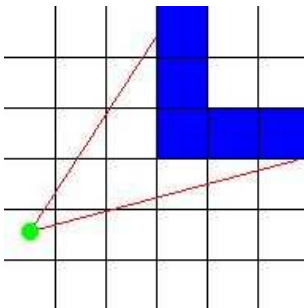
Каждому значению x на экране (для каждой вертикальной линии на экране) соответствует луч, который исходит из местонахождения игрока и направление которого зависит от двух критериев: направление взгляда игрока и координата x на экране



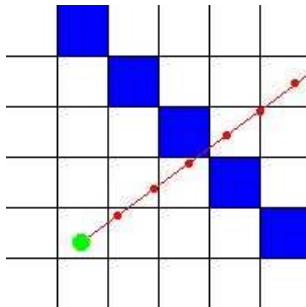
Карта уровня представляет собой 2D-решетку с квадратными ячейками (двухмерный массив), где значением каждой ячейки может быть равно 0, что означает отсутствие стены, либо положительное число, означающее стену определенного цвета или текстуры.

[illegible]

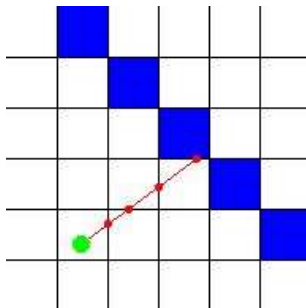
На рисунке в ракурсе «сверху вниз» представлены два луча (выделены красным), которые исходят от игрока (зеленая точка) и упираются в синюю стену.



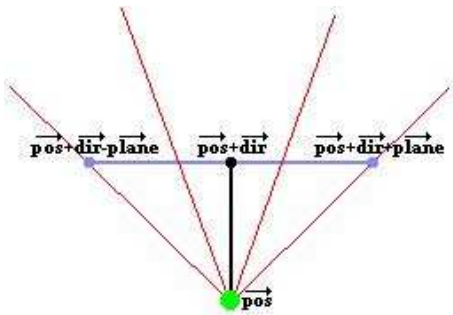
Чтобы обнаружить первую стену, которую луч встречает на своем пути, необходимо, чтобы он исходил из точки местоположения игрока, а затем нужно все время проверять, не находится ли луч внутри стены



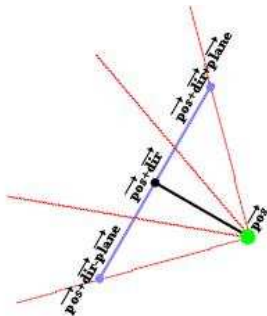
Цифровой Дифференциальный Анализ(DDA) - скоростной алгоритм, позволяющий определить, какие ячейки задевает луч



Рейкастинг работает с векторами для задания угла обзора и камеры: положение игрока всегда является вектором



При вращении игрока камера также должна вращаться, следовательно, и вектор направления, и вектор плоскости камеры также должны поворачиваться вместе с ними. Далее все остальные лучи будут вращаться автоматически

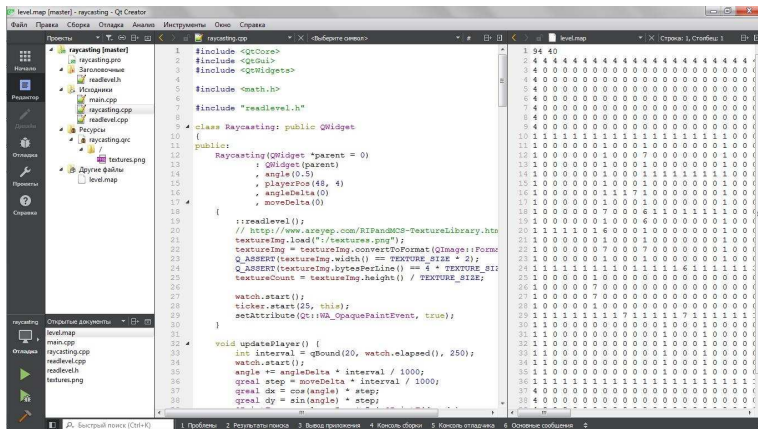


Чтобы повернуть вектор, необходимо рассчитать его по следующему шаблону вращения:

$$\begin{vmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\beta) & \cos(\beta) \end{vmatrix}$$

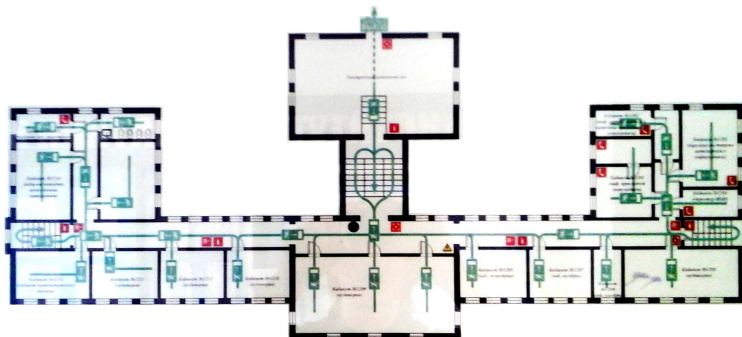
Реализация прототипа

Для реализации прототипа использовался фреймворк Qt для C++.
Работоспособность была проверена под операционными системами *Android*,
Linux Mint 18 и *Windows 10*.



Исходный код доступен всем желающим по ссылке:
<https://github.com/chetca/Raycasting>

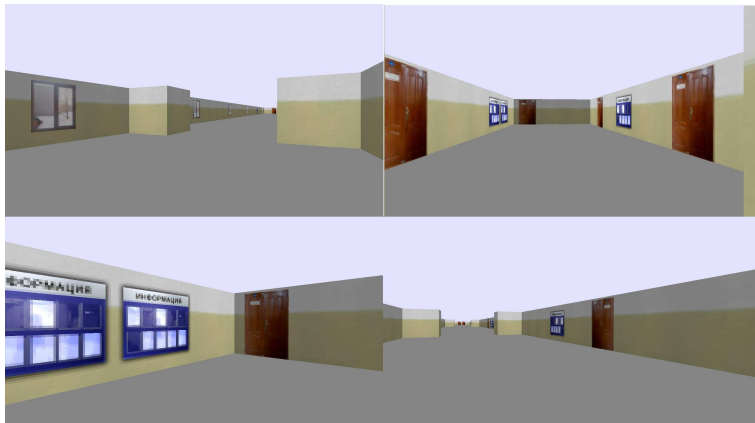
В качестве примера был сконструирован план уровня 2 этажа 1 корпуса БГУ:



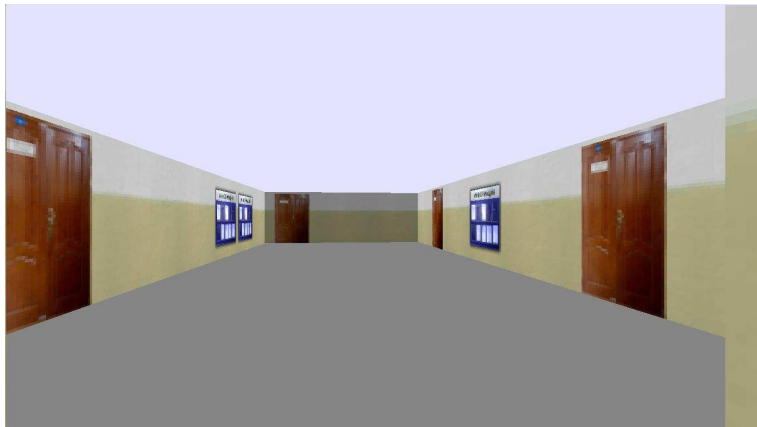
План уровня 2 этажа 1 корпуса БГУ в виде уровня

[illegible]

Скриншоты работающего прототипа:



Скриншоты работающего прототипа:



Разработать инструментарий для создания интерактивных планов помещений для интернет-сайтов организаций:

- переписать прототип на языке JavaScript;
- написать редактор плана помещения;
- поиск оптимального маршрута на плане и визуализация его прохождения;
- внедрение интерактивной карты корпусов БГУ на сайт ИМИ.