

3D визуализации 2D плана помещения методом бросания лучей

Выполнил: студент 05230 гр., Шорников А. Е.

Научный руководитель — к.ф.-м.н., ст. преп. **Трунин Дмитрий Олегович**

Научный консультант — вед. пр. ЛПС БГУ **Брагин Александр Фёдорович**

Бурятский государственный университет
Институт математики и информатики
Кафедра прикладной математики

Улан-Удэ
2017г.

Бурятский государственный университет

Институт математики и информатики

- **Улан-Удэ, Ранжурова, 5**
- У организации 22 филиала
- Сегодня 08:00–17:00, обед 12:00–13:00
Откроется завтра в 08:00

• **Сегодня 08:00–17:00** **Закрито**
обед 12:00–13:00

• **Улан-Удэ, Ранжурова, 5**

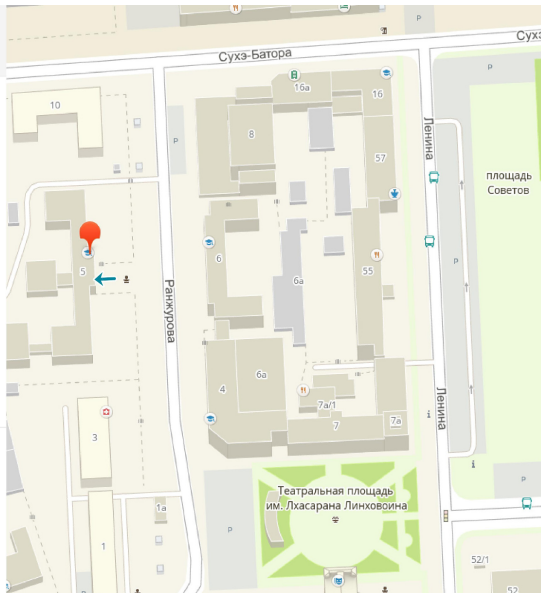
1 корпус; 1203, 1204 кабинет
670000

• 3 этажа

• пл. Советов — 350 м

• Найти парковки рядом

Университеты



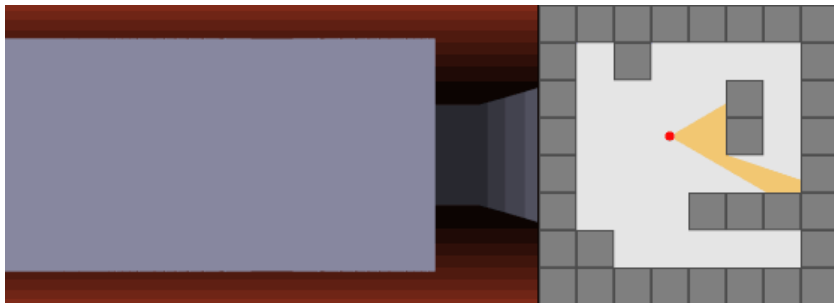
Цель работы

Создание кроссплатформенного псевдотрёхмерного движка (англ. engine) для 3D визуализации помещений и маршрутов в них по 2D плану.

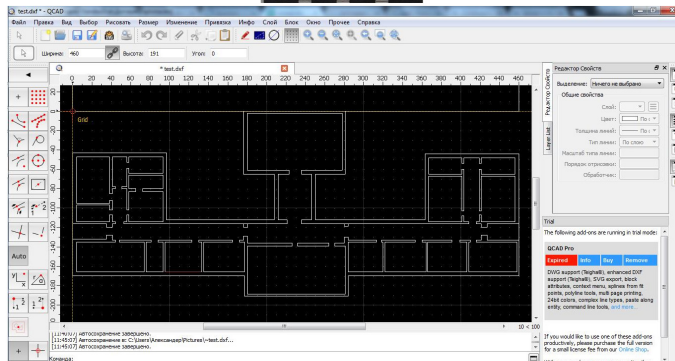
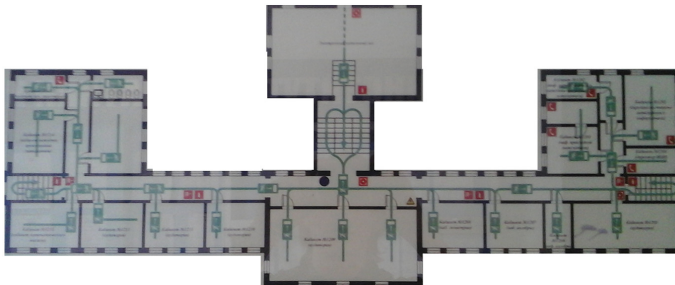
Задачи исследования

- исследование и непосредственная реализация эффективного алгоритма отрисовки проекции трёхмерной сцены
- модификация алгоритма рейкастинга для вещественных координат
- обеспечение интерактивности и поиск маршрутов
- платформа для возможной реализации дополнительных сервисов и кроссплатформенности

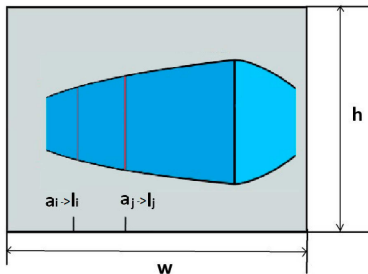
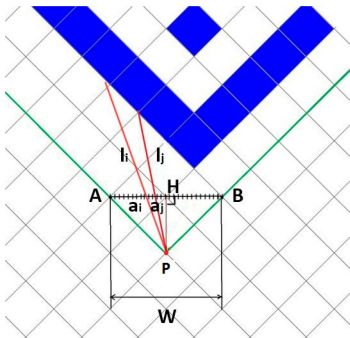
Метод бросания лучей (англ. raycasting, рейкастинг) - один из методов рендеринга в компьютерной графике, при котором сцена строится на основе замеров пересечения лучей с визуализируемой поверхностью.



2D вид карты



Описание метода бросания лучей



$$\begin{aligned}\vec{\Delta a} &= \frac{|\vec{AB}|}{w} \\ \vec{a}_{i+1} &= \vec{a}_i + \vec{\Delta a} \\ \vec{a}_1 &= \vec{A} \\ \vec{a}_n &= \vec{B} \\ \vec{r}_i &= \vec{a}_i - \vec{P}\end{aligned}$$

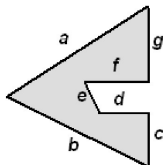
Алгоритм рейкастинга

для каждой $i \in [1, n]$:

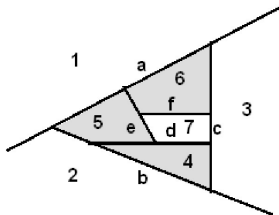
$l \leftarrow \text{расстояние-до-стены}(\vec{P}, \vec{r}_i)$

$h \leftarrow \text{высота-отрезка}(l)$

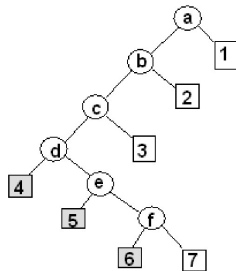
отобразить-отрезок(i, h)



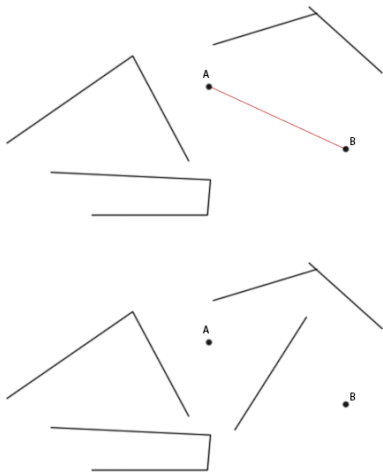
Плоское
пространство



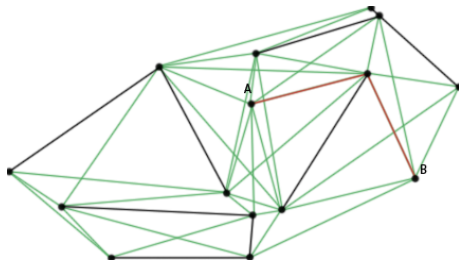
Двоичное разбиение
пространства



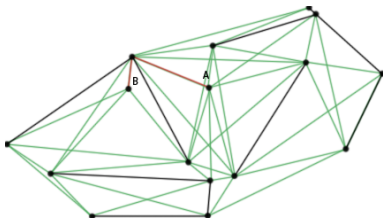
Двоичное дерево



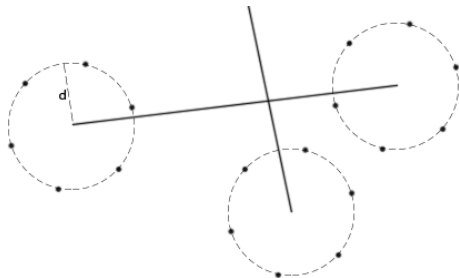
Соединение точек A и B



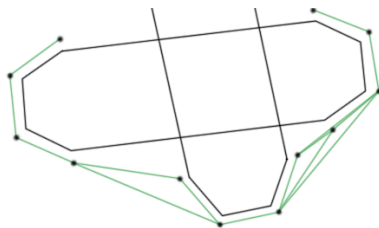
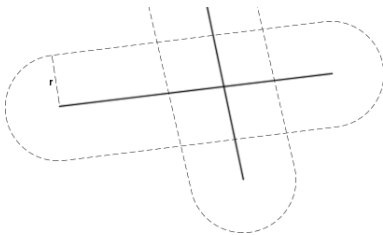
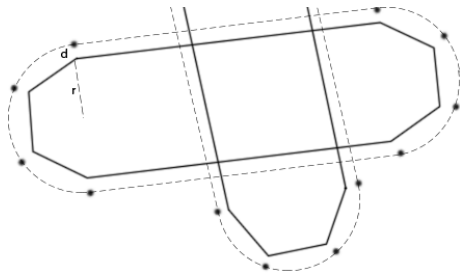
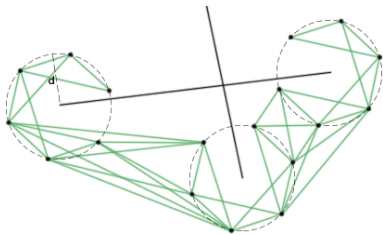
Существование рёбер



Ситуация “проскакивания через углы стен”



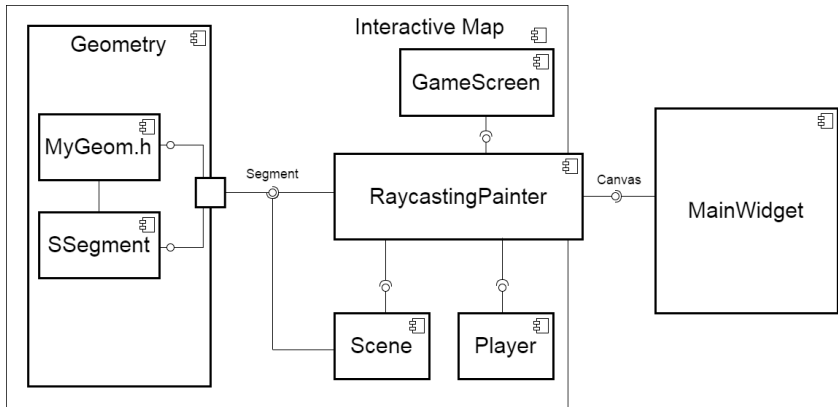
Включение окружностей на концы отрезков



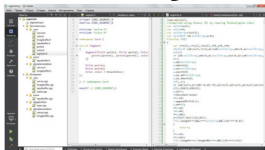
Построение графа на концах сегментов

Алгоритм построения графа для поисков маршрута на карте

- 1) Для каждой точки, являющейся концом отрезка из S найти k точек равномерно расположенных на окружности радиуса $r + d$ с в этой точке. Добавить полученные точки в V .
- 2) Для каждого отрезка для расстояния r найти эквидистанту, приблизить её многоугольником. Удалить из S начальные отрезки и добавить отрезки, являющиеся сторонами полученных многоугольников.
- 3) Удалить из V вершины, лежащие внутри полученных многоугольников.
- 4) Для каждой пары вершин из V , рассмотреть отрезок их соединяющий. Если этот отрезок не пересекает никакой отрезок из S , то добавить ребро между этими вершинами в E .



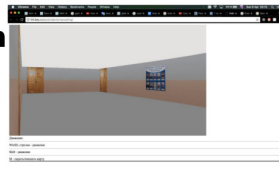
C++ Project



Algo Raycasting
View Map
Abstract Canvas
Abstract Image Buffer

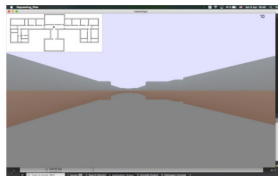
Emscripten Web

EmscriptenCanvas(On Web)
EmscriptenImageBuffer(On Web)



QtCanvas(On Desktop)
QtImageBuffer(On Desktop)

Qt Desktop



Результаты работы

- Модифицирован алгоритм рейкастинга для работы в вещественных координатах
- Создан движок на основе модифицированного алгоритма рейкастинга
- На основе движка сделан интерактивный план помещений корпуса ИМИ БГУ
- Проект реализован на многих платформах

Проект разрабатывается открыто, исходные коды доступны по ссылке:

https://github.com/chetca/Raycasting_Plan



Ссылка на web-реализацию:

<http://imi.bsu.ru/lps/projects/raycasting/>

Спасибо за внимание!

Шорников Александр Евгеньевич

3D визуализация 2D плана помещения методом бросания лучей

Таблица: Замеры производительности на ПК

OS	Memory	CPU	FPS
<i>Windows 10</i>	25 Mbyte	10%	27-41
<i>Ubuntu 16.04.2 LTS</i>	17.4 Mbyte	8%	34-44
<i>FreeBSD 11.0</i>	18 Mbyte	8%	32-47

Таблица: Замеры производительности на мобильных устройствах

Model	Antutu Test	FPS
<i>Samsung Galaxy A5</i>	59834	24-30
<i>Highscreen Power Ice Evo</i>	31672	20-26
<i>Samsung Galaxy Star Plus</i>	8012	12-20

Таблица: Замеры производительности в различных браузерах

Browser	Memory	FPS
<i>Mozilla Firefox</i>	25.4 Mbyte	42-60
<i>Chromium</i>	28 Mbyte	39-56
<i>Safari</i>	28 Mbyte	39-56
<i>Microsoft Edge</i>	30.6 Mbyte	34-51