

Кроссплатформенная реализация 3D визуализации 2D плана помещения

Бадагаров Дугар
Шорников Александр

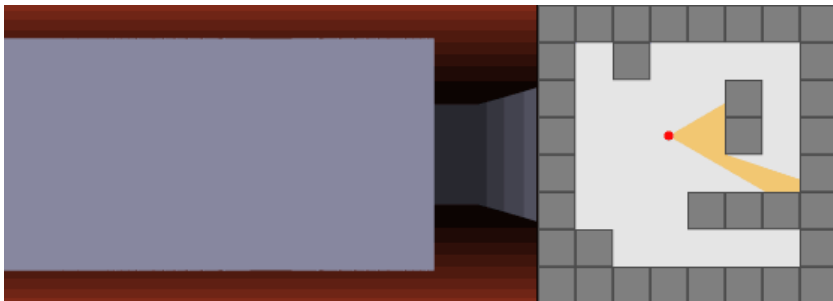
Бурятский государственный университет
Институт математики и информатики

Научный руководитель: ст.преп. Хабитуев Б.В.

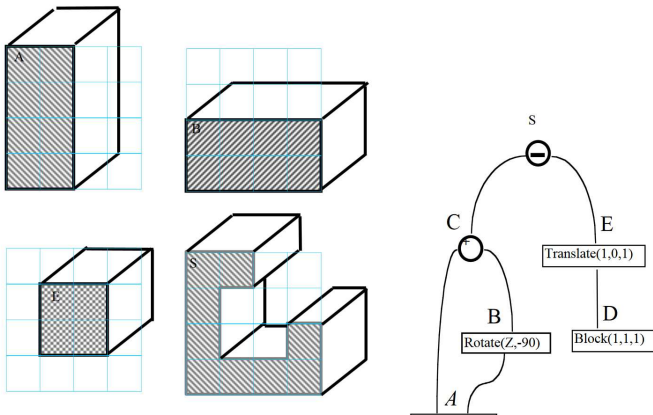
Улан-Удэ
2017г.

Метод бросания лучей

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.



Метод бросания лучей в публикациях по компьютерной графике впервые был применён в 1982 г. для отрисовки моделей конструктивной блочной геометрии



Roth, Scott D. (February 1982), "Ray Casting for Modeling Solids Computer Graphics and Image Processing T. 18: 109–144

Для ПК рейкастинг в 1992 г. популяризировала игра Wolfenstein 3D



Wolfenstein 3D реализует игровую сцену двумерной моделью, которая визуализируется в 3D



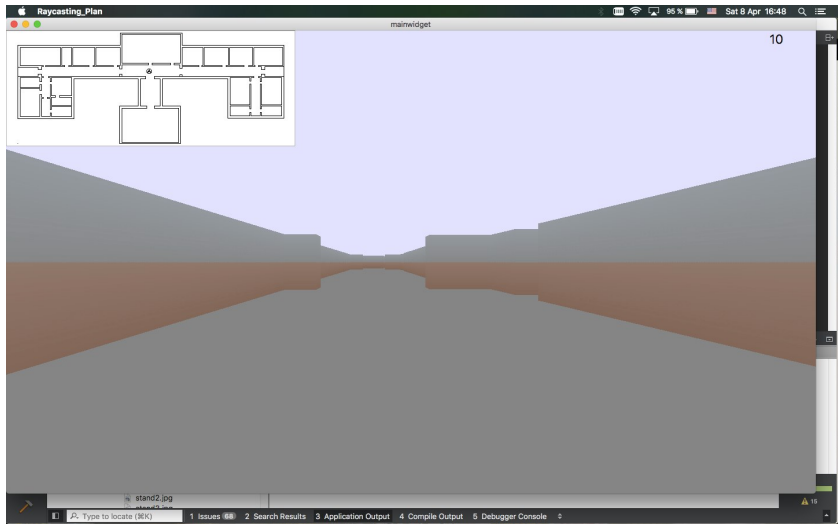
Цель данного проекта:

Создание кроссплатформенного псевдотрёхмерного движка для 3D визуализации помещения по 2D плану.

Задачи:

- ▶ модификация алгоритма рейкастинга для вещественных координат
- ▶ разработка математической модели для рейкастового рендерера
- ▶ изучение и освоение технологии кросс-компиляции Cheerp

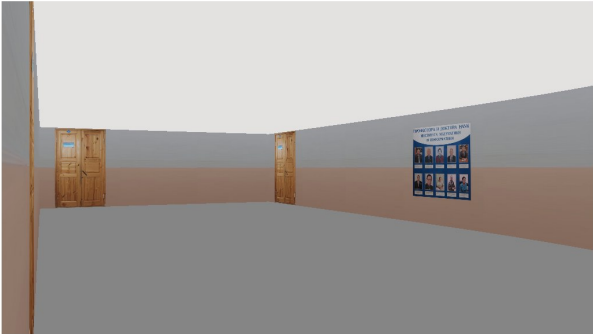
План этажа: Desktop



План этажа: Web

Chrome File Edit View History Bookmarks People Window Help

imi.bsu.ru/lps/projects/raycasting/



Движение:

WASD, стрелки - движение

Shift - движение

М - скрыть/показать карту

C++ Project



Algo Raycasting
View Map
Abstract Canvas
Abstract Image Buffer

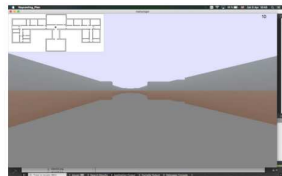
Cheerp Web

CheerpCanvas(On Web)
CheerpImageBuffer(On Web)

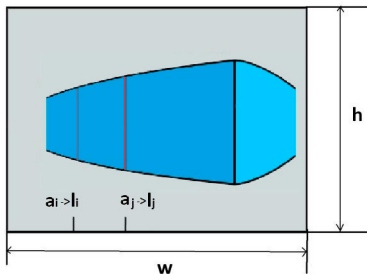
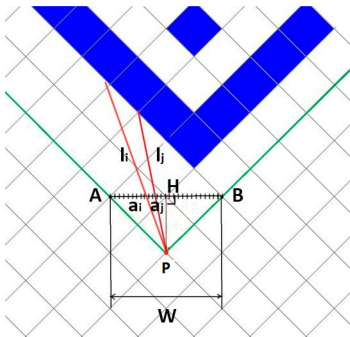


QtCanvas(On Desktop)
QtImageBuffer(On Desktop)

Qt Desktop



Описание метода: основная идея



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

Алгоритм рейкастинга

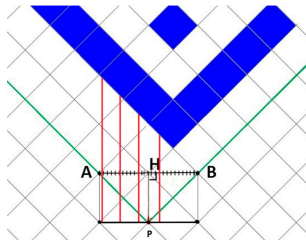
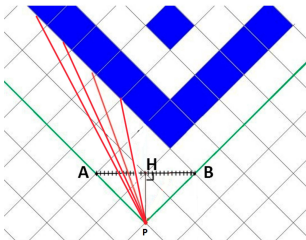
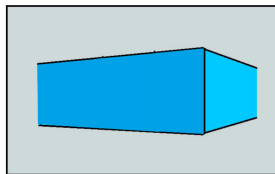
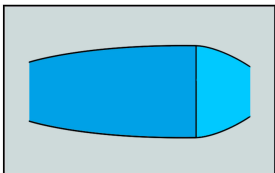
для каждой $i \in [1, n]$:

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

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

отобразить-отрезок(i, h)

Описание метода: эффект «рыбьего глаза»



для каждой $i \in [1, n]$:

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

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

отобразить-отрезок(i, h)

для каждой $i \in [1, n]$:

$l \leftarrow \text{расст-до-стены}(\vec{a}_i - \vec{PH}, \vec{PH})$

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

отобразить-отрезок(i, h)

Дальнейшее развитие проекта

- ▶ Задание и визуализация путей на карте
- ▶ Автоматический визуальный поиск путей на карте
- ▶ User-friendly редактор планов помещений

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!