

3D-визуализация двумерного плана помещения методом бросания лучей

Шорников Александр Евгеньевич, группа 05230

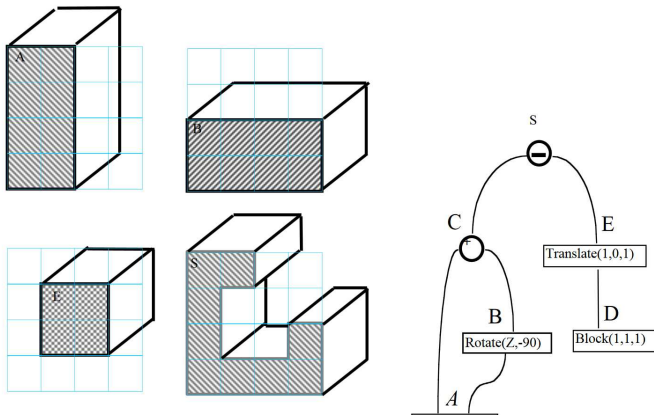
Бурятский государственный университет
Институт математики и информатики
Кафедра прикладной математики

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

Улан-Удэ
2016г.

Метод бросания лучей (или 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

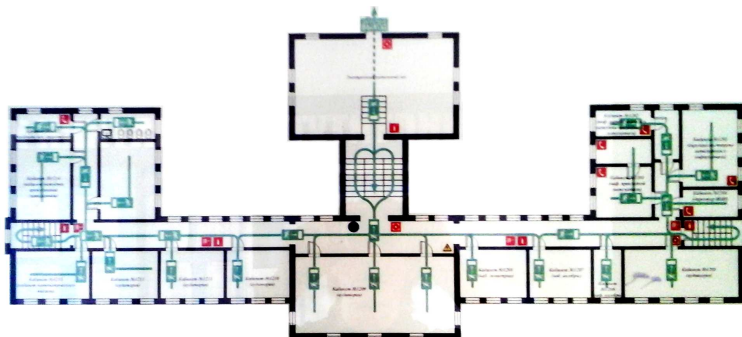


В Wolfenstein 3D реализует игровую сцену двумерной моделью, которая визуализируется в 3D

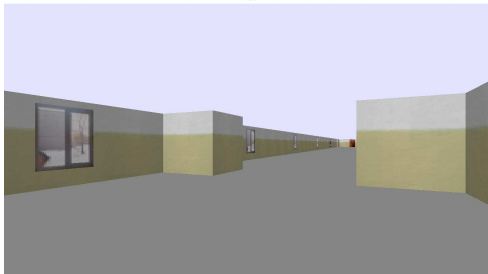
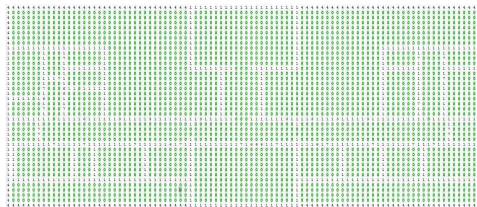


ПЛАН ЭВАКУАЦИИ

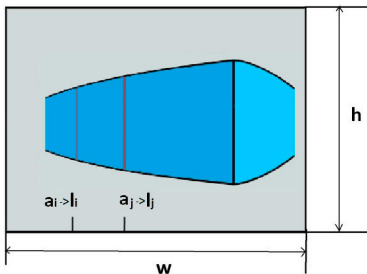
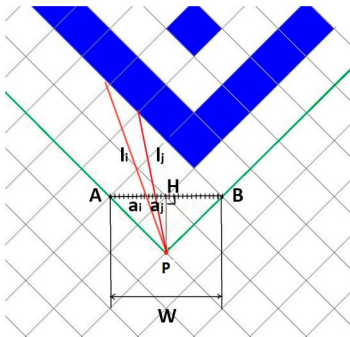
людей и материальных ценностей при возникновении пожара и
других чрезвычайных ситуаций
из здания 2 этажа учебного корпуса №1 Бурятского
государственного университета (г. Улан-Удэ, ул Ранжурова, 5)



3D визуализации плана этажа методом рейкастинга



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

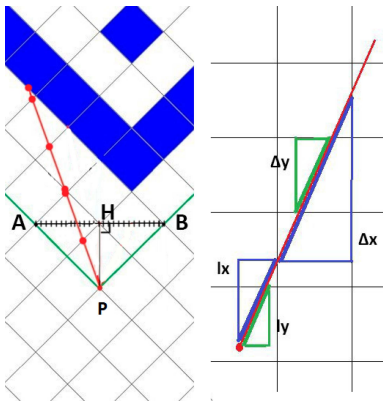
для каждой $i \in [1, n]$:

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

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

отобразить-отрезок(i, h)

Расстояние до стены рассчитываем алгоритмом **цифрового дифференциального анализатора** (digital differential analyzer)



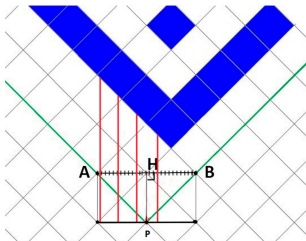
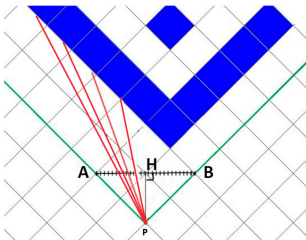
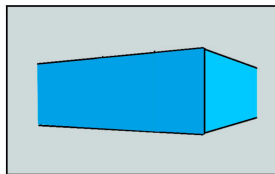
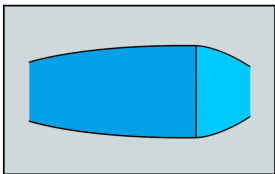
\vec{p} - нормированный вектор луча
 $M_{i,j}$ - число стоящее в матрице уровня в i строке в j столбце

Алгоритм DDA

цикл:

если $l_x < l_y$
 то $l_x \leftarrow l_x + \Delta x$, $l_{cur} \leftarrow l_x$
 иначе $l_y \leftarrow l_y + \Delta y$, $l_{cur} \leftarrow l_y$
 $\vec{r} = \vec{P} + l_{cur} \vec{p}$
если $M_{[r_x], [r_y]} > 0$
 вернуть l_{cur}

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

- ▶ Представление карты геометрическими примитивами
- ▶ Задание и визуализация путей на карте
- ▶ Автоматический поиск путей на карте
- ▶ Реализация метода на JavaScript для встраивания на web-страницы
- ▶ User-friendly редактор планов помещений

Проект разрабатывается открыто, исходные коды доступны по ссылке:

<https://github.com/chetca/Raycasting>



Спасибо за внимание!