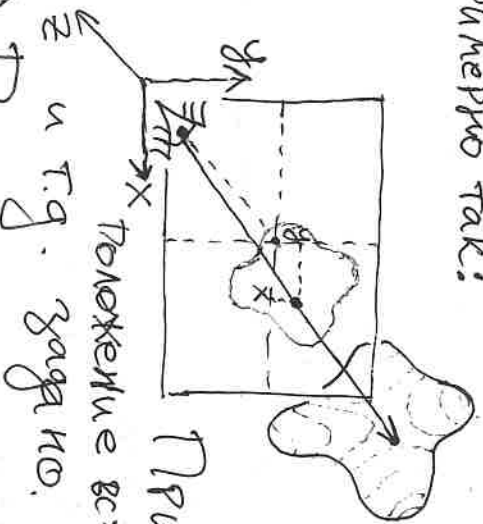


# Что это такое?

Этот проект призван визуализировать объекты движущиеся со скоростью близики и скорости света.

## Как он будет?

За основу взят алгоритм траектории лучей. Работает он примерно так:



1) В какой точке  $x', y', z'$  находится объект?  $x, y, z$  - координаты объекта, а  $x', y', z'$  - координаты объекта в системе координат наблюдателя.

При этом в мировом пространстве координаты объекта, направление нашего взгляда

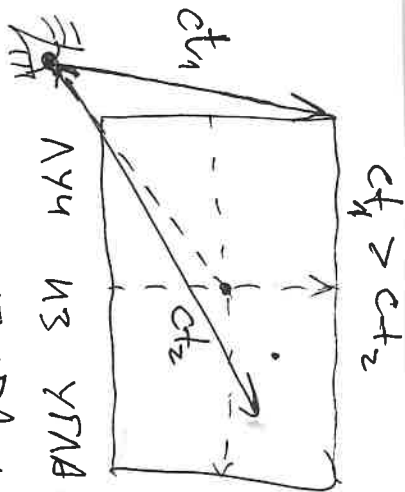
2) Далее мы пересечем наш луч с объектами сцены и либо выйдем что за цвет мы будем, либо дальше продолжим движение луча, как и раньше. 3) Далее так как мы хотим увидеть, что происходит на карте с объектами, то мы сделаем, что при этом мы будем видеть, что происходит на карте.

Перед тем как начать, что объект, который мы хотим нарисовать, будет перед нами со скоростью, близкой к скорости света. Скорость пушки мы знаем, что этот объект должен двигаться со скоростью света. Это уже подразумевает, что мы должны увидеть объект, без необходимости ждать.

Чтобы отобразить этот объект мы можем ~~пересечь~~ перейти из системы отсчета лучей в систему отсчета объекта, в ней объект, очевидно, неподвижен, поэтому трансформирование лучей можно просто пересечь с объектом и заглянуть в его движущийся, отсюда какой мы будем видеть и так далее. В этом и содержится

Вы можете спросить, как же я буду переходить в эту систему отсчета? К счастью, специальная теория относительности дает нам на него ответ.

Нужно добавить к вектору луча координаты времени;



Перед каждым лучу соответствует не 3 числа, а 4. координаты, а 4:

добавляется время, за которое луч долетает от экрана до датчика ЗРАЧКА. Очевидно, что

луч из угла датера будет лететь дольше, чем из центра, и.к. будет пролетать большее расстояние. Поэтому есть четверка чисел:  $(x, y, z, t)$ .

Эту четверку мы преобразуем в одну, а именно преобразованная Лоренца, в которую она ~~преобразуется~~ создается функцией boost matrix, можете её использовать, но в источниках будет ссылка на работу софтверщиков её в исходном виде. Теперь посчитать пересечение можно, используя предельный алгоритм.

Что я сделать не хочу / не захотел / не смог?

- Текстуры
- Освещение
- ~~• Звук~~ ~~• Интерактивность~~ ~~• Материалы~~

Можно было добавить зеркала, прозрачные и дырки материалы. По первому пункту Джинтейна все законы физики действуют одинаково во всех инерциальных системах отсчета, поэтому трюхиение через прозрачные среды, отражение и т.д. всё можно сделать классическими методами компьютерной графики.

- Допневоскум эффект для света: есть ссылка
- Промитивы: хочу ещё что добавить, вроде бесконечного цилиндра, бочонок, твёрдотельные, тогда можно было бы использовать любые моделику сестрицы из теоретиков.