

Слайд 1 [Inspiration *stars*]

Большую часть информации человек воспринимает глазами, именно поэтому одним из самых популярных видов контента на сегодняшний день является визуальный контент.

Красочная, яркая и пёстрая или серая, драматичная. За множество лет человечество успело отобразить реальность в графическом формате множество раз в виде картин, фильмов, фотографии.

Компьютеры стали незаменимыми помощниками в создании графического контента. Вычислительные мощности компьютеров, растущие ежегодно, уже позволяют создавать картинку, неотличимую от реальности. Это стало возможно благодаря развитию графических процессоров - отдельному устройству ПК.

Слайд 2 [Хуль мы сделали]

Целью данного проекта стала разработка графического движка, использующего техники реалистичного рендера и видеокарту.

Предметами исследования стали библиотеки для работы с видеокартами от AMD HIP и HIP RT, техники реалистичного рендера: трассировка лучей и физически корректный рендеринг.

Для достижения цели необходимо изучить техники реалистичного рендера и документацию для работы с видеокартами, выбрать подходящий формат хранения сцены и реализовать на основе полученных данных программу.

Слайд 3 [рейтрейсинг]

В основе трассировки лучей лежит довольно простая идея. Предположим, нам нужно нарисовать картину, но всё что мы можем сделать - это ставить точки и безошибочно определять цвет, куда мы смотрим. Можно разбить холст на квадраты и методично просматривать каждый из них, определяя цвет и ставя точку соответствующего цвета. Таким образом можно получить картину.

Трассировка лучей работает схожим образом. Из точки наблюдения мы будем испускать луч в соответствующем направлении и определять, в какой цвет окрашивать текущий пиксель.

Точка наблюдения - это координаты камеры, а направление луча определяется поворотом камеры и вертикальным углом обзора. Будем отрисовывать

цвет ближайшего объекта, в который мы попали.

Выглядит неплохо, однако чего-то не хватает. Давайте введём свет. В движении доступно три источника света: солнечный, точечный и лампа. Солнечный свет равномерен и падает на любой объект под одинаковым углом. Точечный свет испускается из одной точки во все другие, свет лампы отличается от точечного направлением, радиусом внутреннего конуса (в нём свет будет максимальный) и внешнего конуса (вне его света не будет). Освещённость будем высчитывать из параметров света, его цвета, затухания, угла падения. Получим следующую картину

Стало лучше, но не хватает теней. В случае света от лампы и точечного света будем определять, находится ли другой объект между точкой света и текущим рассматриваемым объектом. Объект есть - есть тень. Нет - тени нет. В случае солнечного света будем испускать луч наоборот из рассматриваемой точки и проверять, пересекается ли он с другими объектами.

Получили следующую картину:

Слайд 4 [PBR] До сих пор мы рассматривали взаимодействие света и объекта упрощённо, однако в реальности такого не бывает.

Сборник техник Physically Based Rendering (в дальнейшем физически корректный рендеринг) позволяет определить, как свет будет взаимодействовать с объектом в зависимости от его физических свойств - шероховатости и металличности.

Согласно физически корректному рендерингу, чтобы модель освещения могла быть реалистичной, она должна удовлетворять трём условиям: Основываться на модели отражающих микрограней, Подчиняться закону сохранения энергии, Использовать двулучевую функцию отражательной способности.

Декомпозируем формулу освещения и сразу приведём её к удобному виду:

Здесь f_r - BRDF, L_i - интенсивность света, w_i - вектор к источнику света, N - нормаль поверхности, p - рассматриваемая точка объекта, w_0 - вектор к наблюдателю, n - количество источников света. Условие три выполнено - мы использовали BRDF.

BRDF в свою очередь состоит из двух частей: диффузного света и блика.

k_d и k_f - коэффициенты, которые влияют на то, насколько сильным будет тот или иной вид освещения, при этом их сумма всегда меньше или равна 1 - здесь выполняется второе условие.

Для получения диффузного света будем использоваться метод Ламберта. Перед погружением в бликовую часть света, необходимо сначала разобраться с отражающими микрогранями.

Поверхность любого объекта, какой бы она ни казалась идеальной, на самом деле таковой не является. Она состоит из маленьких микроподповерхностей, которые отражают свет в другом направлении, заводят его в "ловушку" из других граней. Модель Кука-Торренса учитывает эти особенности микрограней:

D - функция нормальной дистрибуции. Она описывает, как много микрограней будут повернуты к наблюдателю. Для её вычисления будем использовать модель GGX/Trowbridge-Reitz. G - функция геометрической затенённости, описывает, как много поверхности не самозатенено и не скрыто другими микрогранями. Для её вычисления будем использовать модель Schlick-GGX. F - функция Френеля, описывает, какая часть света была отражена, а какая - преломлена (поглощена самим объектом).

В итоге получим картинку:

Слайд 5 [Как работает видеокарта]

Графическому процессору, как и любому другому процессору, для выполнения задачи необходимо указать, с какими данными нужно работать и каким образом. Библиотека HIP RT предоставляет специальные акселерационные структуры данных и классы, ускоряющие нахождение пересечения луча и примитива. Эти структуры данных содержат информацию о геометрии сцены, о трансформации объектов и времени трансформации. Дополнительно видеокарте будем передавать информацию о нормалях вершин, материалы.

Особенностью видеокарты является то, что она может выполнять один процесс многократно и параллельно, такой процесс описывается кернелом - функцией, которая будет запускаться на видеокарте. После того, как мы загрузили сцену в акселирирующие структуры данных, можно приступить к запуску кернела. Процессы в видеокарте выполняются в 3-мерном пространстве, каждому процессу присваивается координата x, y и z. Так как наша

задача - получить двухмерную картинку, вычисления будут двухмерные, а процессу будут присвоены координаты x и y , соответствующие координате отрисовываемого пикселя на холсте.

В результате получим изображение сцены:

Слайд 6 [Скрины и выводы]

В ходе курсовой работы были изучены основы техник генерации реалистичной компьютерной графики, инструменты для добавления аппаратной поддержки в видеокарты. Полученный движок способен отрисовывать простые сцены, содержащие примитивы, свет. Полученный движок реализует базовые техники реалистичного рендера и для дальнейшего применения в определённой сфере требует соответствующих доработок.