Д. С. Шек, Д. С. Папакуль

АНАЛИЗ СПОСОБОВ ОПТИМИЗАЦИИ ОТРИСОВКИ КАДРОВ В РЕАЛЬНОМ ВРЕМЕНИ

Введение

Начинающие специалисты трёхмерного моделирования и разработчики игр регулярно сталкиваются с задачами по оптимизации трёхмерных моделей с целью уменьшения “стоимости” отрисовки кадров. К сожалению, не всегда специалисты разбираются в том, как же отрисовывается кадр, из-за чего специалисты допускают грубые ошибки. [1].

I. Как устроена отрисовка кадра. Вызов отрисовки

Отрисовки кадра - комплексный процесс, направленный на поэтапный расчёт и отрисовку кадра в реальном времени, который происходит внутри игрового движка. Отрисовка кадра имеет стоимость, которая измеряется необходимым временем и ресурсами. Для оценки времени рендера недостаточно знать только количество отрисованных кадров в секунду. Вызов отрисовки (draw call) — это одна графическая команда, которая должна что то отрисовать. Сложность кадра можно оценивать именно с помощью вызовов отрисовки. Чем их больше, тем дольше они обсчитываются, тем медленнее выполняется 1 кадр, тем меньше кадров в одной секунде и тем меньше итоговая частота кадров. [2–3].

II. Батчинг (Batching)

Динамический батчинг работает следующим образом: движок сам анализирует геометрию и сшивает те объекты, которые подходят под лимиты. Обычно у динамического батчинга очень жёсткие лимиты; он не позволяет сшивать тяжёлую геометрию и занимается в основном мелкими деталями в кадре. Например, если от стены динамически отделились мелкие осколки, то они они, скорее всего, будут сшиты динамическим батчингом в каждом отдельном кадре. Статический же батчинг происходит «заранее». Для этого нужно указать что этот объект статичен, и никогда не будет изменен или сдвинут. Отличие от динамического батчинга в том, что он позволяет перерабатывать огромное количество объектов. За один батч Unity может сшить объектов общим количеством до 64 000 треугольников. Кроме того, динамический батчинг будет пытаться пересобрать геометрию почти в каждом новом кадре, — а значит будет постоянно увеличивать время расчёта. [4–5].

III. Уровень детализации (LOD)

Level of Detail (LOD, или уровень детализации) — простая техника, которая позволяет уместить огромное количество объектов в кадре, не отнимая время на обработку лишней геометрии. [6].

IV. Перерасчёт прозрачности (Alpha Overdraw)

Рендер-пайплайн должен оптимизировано рассортировать и нарисовать сцену — обработать шейдеры, геометрию, текстуры и цветовые фильтры. К тому же он должен учитывать, что объекты могут быть прозрачными. Чем больше площадь прозрачности в кадре, и чем больше прозрачные объекты наслаиваются друг на друга — тем дольше рендер-пайплайн будет рисовать итоговый пиксель, ведь ему придётся каждый раз его перерисовывать. Это называется Alpha overdraw. [7].

Итог

Как мы выяснили, практически нет беспроигрышного варианта в случае с отрисовкой кадра: технологии, которые позволяют что-то оптимизировать, не работают по нажатию нужной галочки. Всегда нужно понимать, для чего мы используем тот или иной инструмент. Нынешнее железо — даже мобильное, — может обрабатывать огромное количество треугольников. Объёмы памяти и скорость растут. Чипы на мобильных устройствах и их батареи, которые перегреваются при высоком FPS, тоже становятся всё совершеннее.

Список литературы

1. https://simonschreibt.de/gat/renderhell/
2. Куданков Д. Оптимизация: почему время важнее полигонов
3. Гладилин Д. Проблемы и решения при создании графики для мобильного шутера Guns of Boom

Шек Даниил Сергеевич, студент 3 курса ФИТиУ, grinder8705@gmail.com.
Папакуль Денис Сергеевич, студент 3 курса ФИТиУ, denis.papakul.dp@gmail.com.
Научный руководитель: Кукин Дмитрий Петрович, к.т.н, заведующий кафедрой ВМиП, kukin@bsuir.by