

УДК 004.946

ОПТИМИЗАЦИЯ 3D-МОДЕЛЕЙ КАК СПОСОБ УМЕНЬШЕНИЯ НАГРУЗКИ НА ГРАФИЧЕСКИЙ ПРОЦЕССОР

Ковалёва Д.Ф., магистр, кафедра МСТ, ХНУРЭ

Дейнеко Ж.В., доцент, кафедра МСТ, ХНУРЭ

***Аннотация.** В данной работе рассмотрены способы уменьшения нагрузки на графический процессор мобильных устройств за счет оптимизации используемых 3D-моделей. Описаны причины падения скорости рендеринга мобильных игр.*

***Ключевые слова.** ОПТИМИЗАЦИЯ, 3D-МОДЕЛЬ, РЕНДЕРИНГ, ГРАФИЧЕСКИЙ ПРОЦЕССОР, ПРОИЗВОДИТЕЛЬНОСТЬ, ПОЛИГОНЫ, ВЕРШИНА МОДЕЛИ.*

За последние несколько лет индустрия мобильных игр пребывает на пике своей активности и прибыльности. Стремительное развитие смартфонов, флагманские линейки которых по мощности уже наступают на пятки современным ноутбукам, привело к увеличению количества и ассортимента виртуальных игр к ним – надо же людям чем-то заниматься в метро, в маршрутках, в очередях, и при этом тестировать растущее количество ядер процессоров на своих мобильных телефонах. Аппаратная «начинка» современных смартфонов состоит обычно из четырех ядер. На смартфон можно установить дополнительные приложения, которые позволяют играть в видеоигры, работать с графическими и веб-приложениями. Уникальные гаджеты оборудованы профессиональными камерами и порой становятся полноценной заменой ноутбука или компьютера. Современные мобильные устройства стали мощнее и производительнее, но и требования к качеству мобильных игр также возрастают. Пользователь ожидает от потенциальной игры не только красивой картинки, но и быстрого реагирования приложения.

Рынок мобильных игр развивается настолько быстро, что по прогнозам к 2021 году превратится в индустрию с доходом в сто миллиардов долларов, в то время как прибыль от обычных видеоигр едва превысит тридцать два миллиарда [1]. При всем разнообразии рынка мобильных игр, далеко не каждая может похвастаться хорошей оптимизацией и поддержкой самых распространённых мобильных устройств. Известно, что бережное отношение к ресурсам мобильного устройства при создании игр – это важнейший фактор, влияющий на качество конечного продукта. Особенно это касается мобильных платформ, капризных к активному использованию оперативной памяти. Сокращение количества полигонов позволяет более эффективно заполнять пространство текстурных атласов и немного снижает вычислительную нагрузку [2].

Для того, чтобы достичь наилучшей производительности необходимо оптимизировать игры на всех этапах разработки – выбирать тот вариант, который будет сохранять наибольшую привлекательность с наименьшими затратами ресурсов

устройства. Целью данной работы является исследование процесса оптимизации трехмерных моделей и базовое понимание того, как графическая карта (graphics processing unit, GPU) отрисовывает создаваемые при трехмерном моделировании меши.

Одним из самых важных процессов работы мобильной игры является рендеринг. Рендеринг (дословно, прорисовка, визуализация) – это преобразование трехмерной сцены в статическую картинку или секвенцию кадров. В компьютерной графике и 3д-художники, и программисты под рендерингом понимают создание плоской картинки – цифрового растрового изображения из 3д сцены, процесс получения изображения с помощью компьютерных программ [3].

В случае с 3D-играми первым этапом рендеринга является сбор информации о вершинах 3D-объекта (Vertex) шейдером – программой, которая выполняется на видеокарте. Атрибуты вершин собираются в пакет, который обязательно включает в себя позицию, а также может содержать информацию о цвете и нормали. Поступившие данные обрабатываются в соответствии с инструкциями вершинного шейдера и выводят новые значения позиции вершин и другую информацию. Она попадает в растеризатор, который переносит положение вершин модели в координаты экрана устройства. Данные передаются во фрагментную или, как ее еще называют – пиксельную часть, где задаются такие значения как цвет и прозрачность. Это очень ресурсоёмкий процесс, особенно сцен с большим количеством моделей поэтому 3D-художники максимально уменьшают количество полигонов и по возможности запекают детализацию в карты нормалей. Однако, этого часто бывает недостаточно.

Чтобы достичь наибольшей эффективности необходимо понимать, как именно графический процессор обрабатывает полигоны. Как правило полигон задается массивом данных координат x , y , z для каждой вершины, таким образом, имеем 12 значений (4 вершины с тремя координатами каждая). Второй массив описывает ссылки на сами вершины в виде двух треугольников, поскольку графические процессоры способны отрисовывать только их, несмотря на то, что в более привычном представлении полигон имеет прямоугольную форму, перед импортом модели всегда триангулируются – превращаются в треугольники, чтобы избежать артефактов на дальнейших этапах.

Таким образом, для того, чтобы отрисовать четырехугольный полигон, необходимо разбить его на 2 треугольных полигона. Шейдер соберет и обработает информацию о вершинах, после чего соотнесет каждую с пикселями на экране конечного устройства.

В случае, прямоугольного полигона для ускорения процесса используется кэш вершин, поскольку 2 вершины треугольника общие, повторно их можно не пересчитывать (рис. 1). Таким образом, отрисовка двух треугольников пройдет почти так же быстро, как и одного. Поэтому при моделировании необходимо обращать внимание именно на количество треугольников, а не полигонов, в целом.

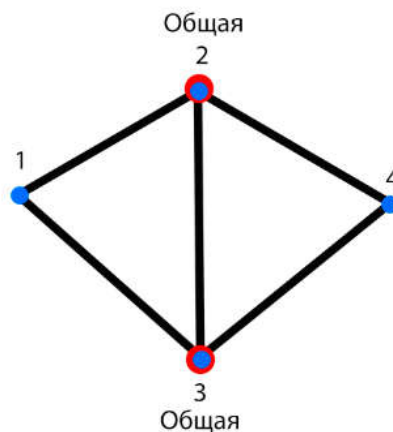


Рисунок 1 – Общие вершины треугольных полигонов

Графический процессор заполняет квадраты, состоящие из пикселей, до тех пор, пока не будут заполнены все. Почти всегда некоторые вершины объектов оказываются снаружи этих каратов, а некоторые внутри. Квадраты, которые затрагивают вершины, просчитываются полностью, даже если объект не заполняет их полностью, а квадраты, которые не задействованы вообще – отсекаются (рис. 2).

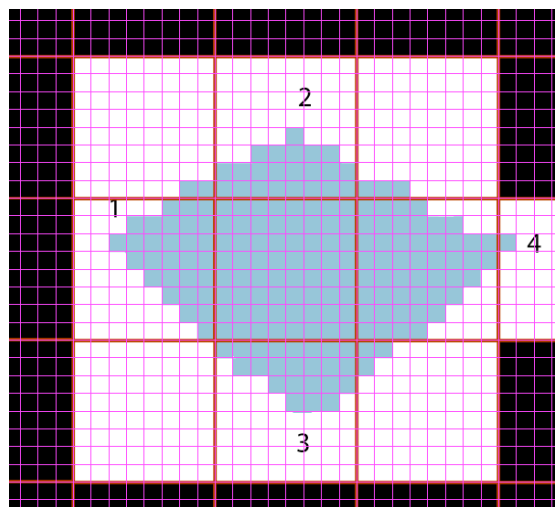


Рисунок 2 – Процесс отрисовки четырехугольного полигона

Скорости рендеринга напрямую зависит от количества квадратов, которые отрисовываются. Вытянутый объект с таким же количеством точек будет рендериться дольше, поскольку он займет большее количество квадратов. Потому 3D-модели необходимо триангулировать избегая длинных полигонов, где это возможно. Этот процесс чаще всего делается вручную или частично автоматизировано, когда художник обрабатывает наиболее проблемные на его взгляд места, а простые разделяются программно.

При рендеринге игр часто возникает такое понятие, как перерисовка (рис. 3). Оно заключается в том, что пиксели могут перерисовываться несколько раз. Например, в случае, когда шар спрятан за кубом. Шар отрисовывается полностью,

несмотря на то, что пользователь его не увидит. Хорошим способом решения этой проблемы является удаление еще на этапе моделирования невидимых частей модели, которые конечный пользователь никогда не увидит. Часто в компаниях создают специальные алгоритмы, которые определяют какая часть модели никогда не попадает в камеру, это очень распространенная практика в играх с частично подвижной камерой. При этом стоит обращать внимание на сколько оптимальным будет решение для конкретного случая.

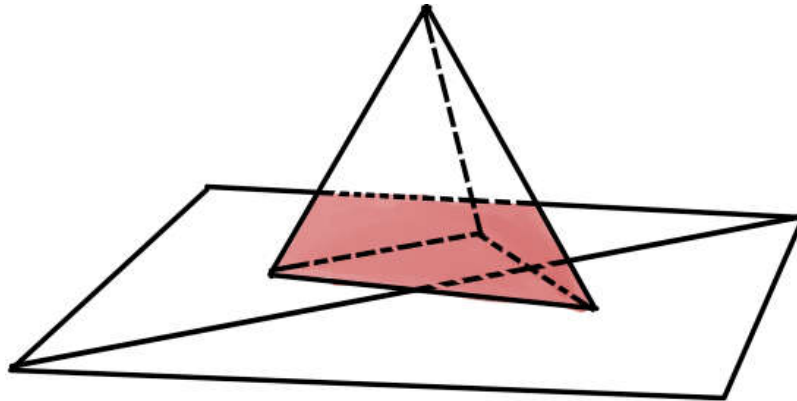


Рисунок 3 – Перерисовка прямоугольника

Таким образом, оптимизация 3D-моделей требует общего понимания процесса рендеринга для достижения максимальной эффективности. Отрисовка точки занимает очень маленькое количество времени, но когда графический процессор сталкивается с десятками тысяч таких точек, необходимо не только минимизировать общее количество полигонов, но и отсекал невидимые части модели, чтобы избежать перерисовки, избегать длинных полигонов. Это позволяет экономить миллисекунды рендеринга и напрямую влияет на производительность игры.

Литература.

1. Игровая индустрия растет за счет мобильных игр. URL: <http://security.mosmetod.ru/internet-zavisimosti/153-igrovaya-industriya-rastet-za-schet-mobilnykh-igr>.
2. Удалить то, что скрыто: оптимизация 3D-сцен в мобильной игре. Советы сотрудников Plarium. URL: <https://habr.com/ru/company/plarium/blog/348494/>.
3. Оптимизация 3D-моделей для игровой сцены. URL: <https://habr.com/ru/company/plarium/blog/484792/>.
4. Оптимизация рендера под Mobile. URL: <https://habr.com/ru/company/playrix/blog/492874/>.