

ОПТИМИЗАЦИЯ 3D МОДЕЛИ

Иванов А.В., Мальчева Р.В.

Донецкий национальный технический университет г.Донецк

Факультет компьютерных наук и технологий

Кафедра компьютерной инженерии

E-mail: tahyku@lisl.ru

Введение

iPhone — линейка четырёхдиапазонных мультимедийных смартфонов, разработанная корпорацией Apple. Смартфоны совмещают в себе функциональность плеера iPod, коммуникатора и интернет-планшета. Работают под управлением операционной системы Apple iOS, представляющей собой упрощённую и оптимизированную для функционирования на мобильном устройстве версию Mac OS X. Впервые iPhone был анонсирован Стивом Джобсом на конференции MacWorld Expo 9 января 2007 года. В продажу поступил 29 июня 2007 года и быстро завоевал существенную часть рынка смартфонов в США. 10 июня 2008 года на конференции WWDC 2008 была представлена новая модель iPhone 3G — более совершенная и лишённая многих аппаратных и программных (с новой версией iPhone OS 2.0) недостатков предшественника, по более низкой цене с контрактом оператора в США. 8 июня 2009 года была представлена следующая, третья по счёту модель телефона — iPhone 3GS, которая, по сути, является усовершенствованной версией iPhone 3G. Всего по состоянию на март 2010 года было продано около 50 млн iPhone. Несмотря на большую популярность iPhone 3GS и на ажиотажный спрос на новинку iPhone 4G, iPhone 3G по прежнему занимает значительный сегмент среди уже проданных iPhone. Информация о продажах iPhone достаточно закрыта, но по данным группы аналитиков из AFB и Investor Villages AAPL Sanity, на 30 августа 2008 года в мире продано 5 млн 649 тыс. iPhone 3G. В связи с этим разработка игр для iPhone 3G по прежнему остаётся востребованной и высокодоходной.

Обработка графики в iPhone 3G

Графическая подсистема представлена разработкой корпорации Imagination Technologies PowerVR MBX-Lite с поддержкой OpenGL ES 1.1. OpenGL ES 1.1 предназначен для работы на функционально ограниченных устройствах. Он позволяет добиться неплохой скорости работы и качества картинки. OpenGL ES 1.1 обеспечивает улучшенное качество изображения, кроме того, он предназначен для использования на устройствах с ограниченными ресурсами (память, электропитание). Официально iPhone 3G способен обрабатывать до 1 млн. полигонов в секунду, но необходимо учесть, что основным объектом в игре является одна или несколько 3D моделей. По

утверждениям разработчиков игр целесообразным количеством полигонов для модели будет 5 – 7 тыс. Ведущей средой для разработки 3D моделей является 3DS MAX, который предоставляет широкий инструментарий создания, изменения и экспорта моделей. Экспортным форматом, который чаще всего используют разработчики приложений для iPhone, является obj.

Приведение исходной модели к низкополигональному эквиваленту

При построении модели необходимо учитывать её дальнейшее спользование. Если это будет основной компонент сцены и он будет отображаться крупно, то необходима его детальная проработка. Если же это будет небольшая деталь, то её детальная проработка будет неуместна. Выполним сравнение подходов моделирования колеса автомобиля, примерное соотношение размеров выполняется на основе игры Need For Speed, адаптированной для iPhone. На рисунке 1 представлены 2 варианта покрышки автомобиля: а) – получен путём сеточного моделирования, б) – полученная с использованием примитива, который подвергся дополнительной обработке.



Рисунок 1 – Примеры двух способов создания объекта

В случае а) только на моделирование покрышки было затрачено 129 859 полигонов, но бола достигнута фотографическая точность реального протектора. В случае б) на моделирование покрышки и диска было затрачено 19 727 полигонов. Поскольку модель предполагается использовать на мобильной платформе, то следует знать примерное соотношение размеров колеса и сцены. На рис.2 а) представлен кадр из игры Need For Speed, адаптированной для iPhone и фрагмент этого кадра б).



а)



б)

Рисунок 2 – Кадры игры Need For Speed для iPhone

При сравнении видно, что колесо автомобиля имеет малый размер, следовательно, его высокая детализация не уместна. Выше было указано, что желательная детализация объекта должна быть порядка 5 – 7 тыс. полигонов. Выбрав модель колеса на основе примитива и смоделированный отдельно сеточным способом диск, попробуем оптимизировать нашу модель до необходимого количества полигонов. Критерием для определения достаточного уровня детализации будем использовать сравнение восприятия обработанной и неоптимизированной модели, представленной на рисунке 3. Первое, что можно сделать для улучшения модели, это вручную убрать некоторые ненужные полигоны. При этом следует руководствоваться рядом принципов:

- 1) Следует избегать удаления граней и вершин, которые могут существенно ухудшить внешний вид модели;
- 2) Следует удалять вершины не несущие функциональной загрузки, т.е. никак не влияющие на внешний вид модели;
- 3) Следует удалять полигоны, которые ни при каких обстоятельствах ни будут видны.



Рисунок 3 – Неоптимизированная модель колеса

На рисунке 4 представлены изображение оптимизированного колеса: а) его вершины и каркас, б) его детализация при использовании для iPhone.



а)

б)

Рисунок 4 – Каркас и отображение колеса при ручной оптимизации

Получили модель, состоящую из 6 840 полигонов. Данная модель почти не отличается от неоптимизированной. Для уменьшения количества полигонов можно использовать стандартные средства - это модификатор Optimize, который входит в 3DS MAX. Предназначен для оптимизации поверхности объекта путем уменьшения количества граней и вершин с сохранением исходной формы модели. Optimize (Оптимизировать) не позволяет задать процент вершин, которые будут содержаться в обработанном объекте. Однако использование данного модификатора не даёт ожидаемого результата, так как колесо имеет много закруглений, которые этот модификатор преобразовывает к углам. На рисунке 5 изображён результат применения модификатора optimize.

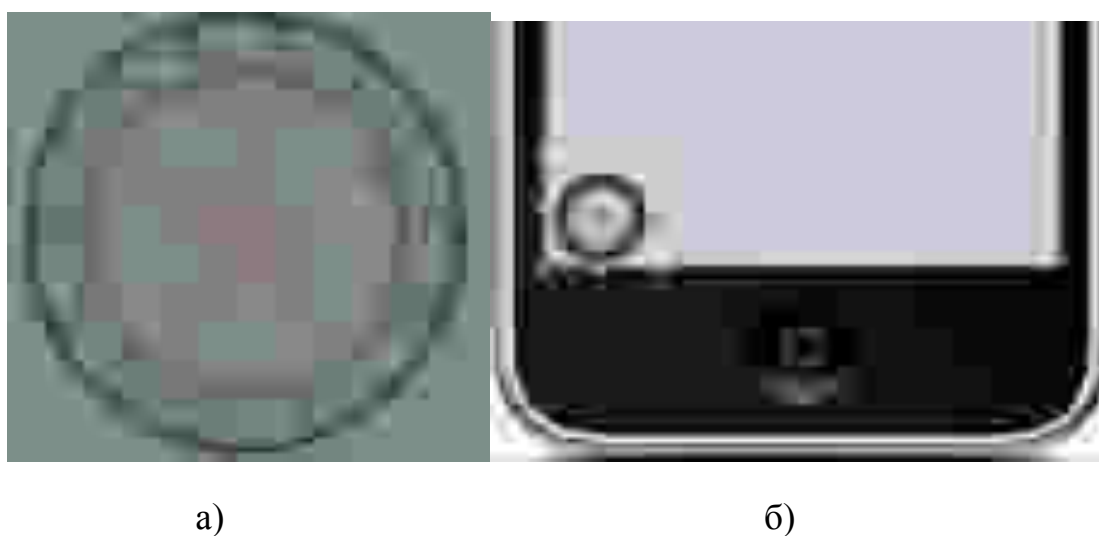


Рисунок 5 – Каркас и отображение модели, оптимизированной при помощи модификатора optimize.

Полученная таким способом модель также мало отличается от неоптимизированной, но при этом имеет 6 052 полигонов. Несомненно, что оптимизация при помощи optimize в данном случае предпочтительней, но при необходимости увеличения модели мы можем наблюдать её значительное ухудшение. На рисунке 6 представлены модели при некотором их увеличении, что нередко встречается в современных играх:

- а) неоптимизированная модель;
- б) оптимизированная вручную;
- в) оптимизированная при помощи optimize.

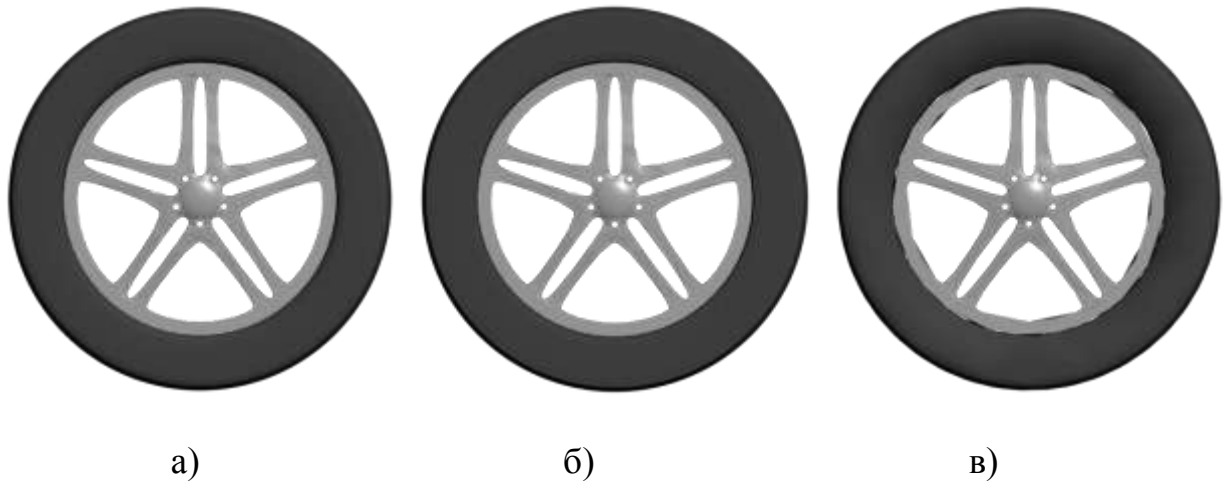


Рисунок 6 – Увеличенные модели колёс

Из рисунка 6 видно, что модель б) практически не уступает модели а), но модель в), которая была оптимизированная стандартно, имеет явные недостатки.

Выводы

Проведя данный опыт по приведению среднеполигональной модели к её упрощённому эквиваленту можно сделать вывод, что такая обработка является неотъемлемой частью процесса моделирования. Данные задачи могут решаться непосредственно создателями модели, т.к. оптимизация - достаточно специфичный процесс и может отнять много времени у программиста, который данную модель будет реализовывать.

Список литературы

1. Статистика продаж iPhone 3G –[электронный ресурс]-<http://www.iliberty.ru/novosti/statistika-prodazh-iphone-3g-v-rossii>.
2. Alessi J. - iPhone 3D Game Programming – 2010
3. Сведения о требованиях к 3d модели –[электронный ресурс]-<http://iphoner.org.ua/blog/opengles/>
4. Высокополигональные модели –[электронный ресурс]-<http://3d-uroki.org/tutorial/3d-max/220>