

МЕТОДЫ ОПТИМИЗАЦИИ ВИРТУАЛЬНОГО ОКРУЖЕНИЯ ПРИ РАЗРАБОТКЕ VR ПРИЛОЖЕНИЙ

Нефедов М.В.

Томский политехнический университет, студент гр. 8ПМ22, e-mail: mvn13@tpu.ru

Научный руководитель: В. В. Видман, старший преподаватель ИШИПР ТПУ

Введение

Оптимизация виртуального окружения является важным процессом при разработке программ на игровых движках. Оптимизация улучшает производительность программы и восприятие пользователем виртуального окружения.

Оптимизация виртуального окружения особенно актуальна при разработке программ под VR, так как программы, разрабатываемые под VR, требуют больших ресурсов от компьютера. На данный момент существует несколько методов оптимизации, каждый из которых влияет на производительность по-разному.

Анализ методов оптимизации виртуального окружения

1. Оптимизация 3D – моделей

– Редукция полигонов – это упрощение 3D-модели с помощью уменьшения количества полигонов.

– Удаление невидимых граней

– Levels Of Detail (LOD). Система LOD снижает количество визуализируемых полигонов по мере удаления объекта от камеры. Если объект находится далеко, то система LOD сменит текущую модель на модель с меньшим количеством полигонов.

– Использование батчинга (объединение объектов). Объединённые объекты при прорисовки прорисовываются вместе и тем самым используют меньше ресурсов при вычислении. Но объединять объекты можно только объекты с одинаковыми материалами.

2. Рендеринг объектов в поле зрения камеры

– Frustum Culling – это функция, которая позволяет не рендерить объекты находящиеся вне поле зрения камеры по краям.

– Occlusion Culling – это функция отключает рендеринг объектов, которые находятся вне поле зрения камеры включая объекты, которые находятся друг за другом.

3. Оптимизация текстур

Для оптимизации текстур их можно объединить в так называемые атласы, в которых будут все текстуры, используемые в программе. Для извлечения текстур можно указывать их координаты в атласе.

4. Оптимизация света и теней

Для оптимизации света и теней их можно заранее «запечь», то есть просчитать заранее, а не в реальном времени. Это подходит для статических объектов и достаточно сильно улучшает производительность.

Тестирование методов оптимизации в движке Unreal Engine 4

Протестируем метод оптимизации Levels Of Detail (LOD) и как он влияет на производительность. Создадим сцену из большого количества одинаковых моделей камней. И применим на модель камня систему LOD из 4 уровней. На первом уровне модель имеет оригинальное количество полигонов – 11334 шт., а на четвёртом – 1417 шт. (Рисунки 1-2).

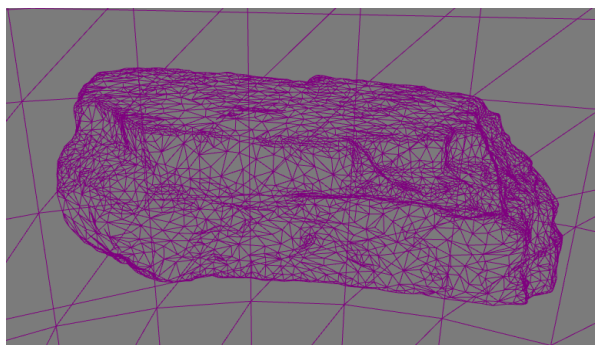


Рис. 1. Модель с первым уровнем

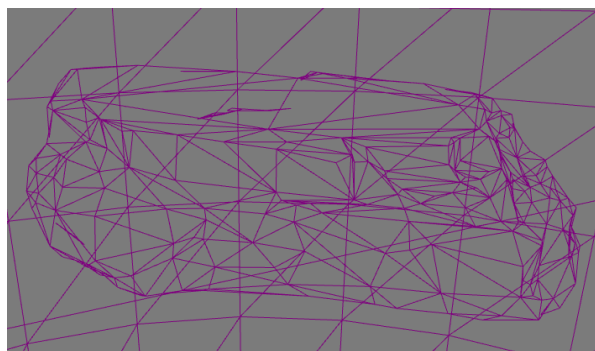


Рис. 2. Модель с четвертым уровнем

Численные значения времени подготовки кадра приведены на рисунке 3.

Frame: 14.78 ms	Frame: 18.73 ms
Game: 4.43 ms	Game: 5.84 ms
Draw: 14.50 ms	Draw: 18.56 ms
GPU: 14.85 ms	GPU: 18.86 ms

Рис. 3. С использованием уровней детализации (слева), в оригинальной детализации без уровней (справа)

Без использования LOD количество полигонов на сцене составило 34 490 336 шт.

С использованием LOD количество полигонов на сцене составило 816 308 шт.

Заключение

Из теста можно сделать вывод, что система уровней детализации (LOD), повышает производительность, так как эта система уменьшает количество полигонов на сцене в зависимости от расстояния до объекта. Но также надо заметить, что при использовании этой системы с большим количеством уровней, приходится жертвовать оперативной памятью и видеопамятью. Данный метод используется на всех актуальных игровых движках и является достаточно действенным.

Список использованных источников

1. Unreal Engine 4 Documentation. [Электронный ресурс]. – URL: <https://docs.unrealengine.com/4.27/en-US/>. (дата обращения 20.12.2022).
2. Денис Кузнецов. Оптимизация в UE4. Базовые советы. [Электронный ресурс]. – URL: <https://dtf.ru/u/3872-denis-kuznecov/225988-optimizaciya-v-ue4-bazovye-sovety>. – Дата публикации: 3 января 2021. (дата обращения 20.12.2022).
3. Paweł Skop. Comparison of performance of game engines across various platforms. [Электронный ресурс]. – URL: https://www.researchgate.net/publication/337172350_Comparison_of_performance_of_game_engines_across_various_platforms. – Дата публикации: сентябрь 2018. (дата обращения 20.12.2022).