

СОЗДАНИЕ АЛГОРИТМА РАЗДЕЛЕНИЯ УЗЛОВ РЕНДЕР-ГРАФА МЕЖДУ ВИДЕОКАРТАМИ

Козлов Н.С (ИТМО)

Научный руководитель – инженер ИРВ по сов-ву, Богданов М.К.
(ИТМО)

Введение. В современном мире, компьютерная графика является важнейшей частью любой компьютерной игры. Из года в год компании-производители стараются внедрить в свои игры всё больше и больше графических технологий, которые приблизили бы внешний облик игры к фотореалистичной картинке. Вместе с тем, цена фотореализма в компьютерных играх достаточно высока, что выражается в постоянно растущих системных требованиях для конечного пользователя и внедрении огромного количества дополнительных зависимостей для самих разработчиков, что усложняет процесс работы. По этой причине, разработчики всё чаще прибегают к внедрению рендер-графа [1] в графическое ядро своих игровых движков, что позволяет в разы проще оперировать зависимостями, а так же визуализировать конвейер отрисовки в виде графа для дальнейшего анализа и проведения автоматических оптимизаций. Иными словами, введение рендер-графа позволяет свести задачу оптимизации конвейера отрисовки к задаче оптимизации структуры самого рендер-графа.

Как было сказано ранее, компьютерная графика сильно полагается на вычислительные мощности системы. Но несмотря на это, один из подходов к реализации графического ядра остаётся наименее популярным за счёт его сложности – Multi-GPU рендеринг. Однако, применение рендер-графа помогло бы проще и нагляднее балансировать нагрузку и распределять ресурсы между устройствами, а вместе с тем и добавить вычислительных мощностей. Статистика «Steam» показывает, что встроенная видеокарта имеется почти у каждого пользователя [2], а значит её можно использовать для Multi-GPU вычислений.

Целью данной работы является исследование возможности использования двух видеокарт для разделения задач отрисовки между ними в пределах одного рендер-графа.

Основная часть. Для решения задачи разделения узлов рендер-графа между устройствами предлагается использовать эвристический алгоритм, основывающийся на динамическом анализе узлов рендер-графа и производительности устройств в режиме мультиадаптера [3]. Основная идея заключается в том, чтобы учитывать размеры ресурсов, необходимых для прохода отрисовки, и обрабатывать проходы параллельно, на втором устройстве, если его характеристики позволяют это сделать достаточно эффективно.

Для этого алгоритм должен включать следующие этапы:

1. Анализ рендер-графа: определение зависимостей между узлами, оценка вычислительной сложности каждого узла и размера его ресурсов.
2. Анализ устройств: анализ производительности каждого устройства.
3. Динамическое распределение: использование эвристических методов для определения оптимального распределения узлов между устройствами.
4. Адаптация выполнения: мониторинг ресурсов и перераспределение узлов в случае дисбаланса нагрузки.

Выводы. Проведён анализ существующих решений для оптимизации рендер-графа. На основе проведённого анализа разработана методика оптимизации с применением Multi-GPU подхода, позволяющая задействовать несколько устройств для просчёта проходов отрисовки.

Список использованных источников:

1. FrameGraph: Extensible Rendering Architecture in Frostbite // gdcvault URL: <https://www.gdcvault.com/play/1024612/FrameGraph-Extensible-Rendering-Architecture-in/>
2. Steam Hardware & Software Survey: January 2025 // store.steampowered.com URL: <https://store.steampowered.com/hwsurvey/videocard/>
3. Explicit Multi GPU Programming with DirectX 12 // gdcvault URL: https://media.gdcvault.com/gdc2016/Presentations/Juha_Sjoholm_DX12_Explicit_Multi_GPU.pdf