

ИССЛЕДОВАНИЕ И РАЗРАБОТКА СИСТЕМЫ ТРЕХМЕРНОЙ ГРАФИКИ

Десятков Владислав Владимирович

Тішин Петро Металлинович, к.т.н., доцент

Высокопроизводительная графика имеет множество областей применения. Однако для достижения хороших показателей производительности необходимо разделить вычисления на блоки, которые можно будет рассчитывать параллельно. Главным преимуществом компьютерной графики является отличная возможность распараллелить вычисления, таким образом сложные алгоритмы визуализации получается реализовывать за разумное время. На данный момент основными библиотеками для работы с компьютерной графикой считаются Vulkan и OpenGL.

Цель данной работы – попытка исследовать библиотеку, которая является следующим поколением в компьютерной графике.

Vulkan - кроссплатформенный API для 2D- и 3D-графики, изначально она создавалась как следующее поколение компьютерной графики после OpenGL API. Vulkan API имеет значительно более низкоуровневый доступ к ресурсам, чем его предшественник[1].

Android поддерживает Vulkan с версии 7.0, кроме того Vulkan обладает огромным объёмом базовой функциональности, что позволяет очень эффективно использовать эту библиотеку на мобильных устройствах.

Если сравнивать ее с OpenGL, то в среднем можно получить дополнительных 30% к производительности, что является довольно хорошим показателем. На Android эту библиотеку поддерживают такие устройства как Samsung S7 и выше. На официальном сайте [2] можно найти саму библиотеку и примеры ее использования. Если проводить сравнения с OpenGL ES[3], то Vulkan показал еще большую производительность[3].

Целью Vulkan было превзойти другие API, включая его предшественника OpenGL, в части снижения накладных расходов, повышения степени прямого контроля над GPU и уменьшения нагрузки на CPU. Vulkan имеет предполагаемые преимущества:

- OpenGL использует язык высокого уровня для написания шейдеров GLSL. Это вынуждает каждого производителя OpenGL драйвера реализовать свой собственный компилятор для GLSL, который работает во время выполнения графического приложения, компилируя шейдерные программы в исполняемый код целевой платформы. Vulkan вместо этого предлагает промежуточный двоичный формат SPIR-V (Standard Portable Intermediate Representation), аналогичный двоичному формату в который компилируются HLSL-шейдеры на платформе DirectX. Это снимает бремя с поставщиков драйверов, позволяя компилировать шейдеры на этапе разработки. Также позволяет разработчикам приложений писать шейдеры на других языках, кроме GLSL.

- Кроссплатформенный API поддерживается на мобильных устройствах и высокопроизводительных видеокартах.

- Улучшенная поддержка современных систем, использующих многопоточность.

- Снижение нагрузки на центральный процессор в ситуациях, когда он является недостаточно производительным, что позволяет достичь более высокой пропускной способности для GPU-вычислений и визуализации.

OpenGL ориентируется на следующие две задачи:

- Скрыть сложности адаптации различных 3D-ускорителей, предоставляя разработчику единый API.

- Скрыть различия в возможностях аппаратных платформ, требуя реализации недостающей функциональности с помощью программной эмуляции.

Основным принципом работы OpenGL является получение наборов векторных графических примитивов в виде точек, линий и треугольников с

последующей математической обработкой полученных данных и построением растровой картинка на экране и/или в памяти. Векторные трансформации и растеризация выполняются графическим конвейером (graphics pipeline), который по сути представляет собой дискретный автомат.

Абсолютное большинство команд OpenGL попадает в одну из двух групп: либо они добавляют графические примитивы на вход в конвейер, либо конфигурируют конвейер на различное исполнение трансформаций.

OpenGL является низкоуровневым процедурным API, что вынуждает программиста диктовать точную последовательность шагов, чтобы построить результирующую растровую графику (императивный подход). Это является основным отличием от дескрипторных подходов, когда вся сцена передается в виде структуры данных (чаще всего дерева), которое обрабатывается и строится на экране. С одной стороны, императивный подход требует от программиста глубокого знания законов трёхмерной графики и математических моделей, с другой стороны — даёт свободу внедрения различных инноваций.

Применение библиотеки Vulkan позволяет достичь лучших показателей производительности и качества компьютерной графики. Библиотека постоянно развивается и распространяется, благодаря чему уже сейчас она доступна и на мобильных платформах.