Техническое задание к выпускной квалификационной работе

1. Общие сведения

Я не уверен, что именно здесь нужно писать.

В рамках выпускной квалификационный работы бакалавра разрабатывается кроссплатформенная библиотека под кодовым названием *vulkalc*, дающая разработчикам возможность проводить вычисления на графических процессорах, абстрагировавшись от целевой платформы. В основе разрабатываемой библиотеки лежит библиотека *Vulkan*, разработанная группой компаний *Khronos Group* на замену устаревшему стандарту *OpenGL*. Основное назначение библиотеки *Vulkan* – разработка приложений *2D* и *3D* графики для устройств на основе операционных систем *Windows*, *macOS*, *Linux* и *Android*. Многие из задач в *3D* графике требуют быстрых вычислений, в том числе и вычислительно-сложных. Для этих задач графические библиотеки вроде *DirectX*, *OpenGL* и *Vulkan* предоставляют возможность написания так называемых вычислительных шейдеров. Именно вычислительные шейдеры и будет использовать разрабатываемая библиотека, не используя графические возможности *Vulkan*.

Так как разработчиком заявлены лучшая по сравнению с *OpenGL* поддержка многопоточности, уменьшенная нагрузка на центральный процессор и низкоуровневый контроль над графическим процессором, то я ожидаю добиться производительности вычислений не меньше, чем при использовании технологии *CUDA* и библиотеки *CUDA SDK*. (возможно лучше будет сравнивать с *OpenCL*)

1. Назначение и цели создания системы

Назначение разрабатываемой библиотеки – высокопроизводительные вычисления на графических процессорах без привязки к конкретной платформе.

1. Требования к разрабатываемой системе
   1. Общие требования

Разрабатываемая библиотека *vulkalc* должна:

* Иметь открытый исходный код
* Иметь открытую лицензию
* Иметь соглашение по оформлению исходного кода
* Иметь инструкции по сборке
* Иметь инструкцию по
* Иметь демо-проект для демонстрации примеров использования
* Иметь документацию на английском языке по публичным классам, методам, полям, функциям
* Компилироваться под операционные системы *Windows* и *Linux*
* Иметь юнит-тесты и интеграционные тесты
* Проходить все существующие тесты
* Использовать систему контроля версий
* Иметь комментарии в исходном коде
* Компилироваться и запускать тесты на целевых операционных системах при каждом изменении
* Иметь как можно меньше зависимостей
* Быть потокобезопасной (не знаю, надо ли, не добавит ли лишних сложностей)
  1. Функциональные требования

Разрабатываемая библиотека *vulkalc* должна предоставлять следующие возможности:

* Статическая линковка к программам на операционных системах *Windows* и *Linux*
* Проверка наличия поддерживаемых графических процессоров и перечисление установленных в системе поддерживаемых графических процессоров
* Получение технических характеристик установленных поддерживаемых графических процессоров – частота графического процессора, размер видеопамяти, разрядность шины памяти, пропускная способность памяти, количество видеочипов (не уверен, что *Vulkan* даёт такую возможность, не хотелось бы тянуть *Windows API*)
* Проверка наличия в системе установленного *Vulkan Runtime*
* Выбор графического процессора из списка доступных для запуска на нём вычислений
* Запуск разработанных пользователем вычислительных шейдеров
* Использование предоставленных разработчиком библиотеки *vulkalc* встроенных функций математических вычислений и связанных с ними типами данных
* Замер времени выполнения вычислений и количестве операций, выполненных на графическом процессоре (возможно это лишнее)
* Подготовка установленных поддерживаемых графических процессоров к вычислениям
* Освобождение графического процессора
* Запуск и остановка вычислений на поддерживаемых установленных графических процессорах
* Копирование данных в видеопамять
* Копирование данных из видеопамяти
* Выделение видеопамяти для собственных нужд определённого размера
* Очищение выделенной видеопамяти
* Сообщать код и информацию об ошибке в случае её возникновения

Библиотека *vulkalc* должна предоставлять следующие функции для математических операций на графическом процессоре:

1. Функции работы с векторами:
   * Сложение векторов
   * Векторизованное сложение массивов векторов
   * Вычитание векторов
   * Векторизованное вычитание массивов векторов
   * Векторизованное умножение массива векторов на массив скаляров
   * Скалярное произведение векторов
   * Векторизованное скалярное умножение массивов векторов
   * Векторизованное нахождение длины векторов в массиве
   * Поиск точки пересечения векторов (это вообще нужно?)
2. Функции работы с матрицами:
   * Проверка равенства матриц
   * Векторизованная проверка равенства массивов матриц
   * Сложение матриц
   * Векторизованное сложение массивов матриц
   * Вычитание матриц
   * Векторизованное вычитание массивов матриц
   * Произведение матриц
   * Векторизованное произведение массивов матриц
   * Умножение матрицы на число
   * Векторизованное умножение массива матриц на массив чисел
   * Возведение матрицы в степень
   * Векторизованное транспонирование массива матриц
3. Функции работы с графами(не шарю в графах, не знаю, на сколько параллелизуемые алгоритмы)

Библиотека *vulkalc* должна предоставлять следующие вспомогательные функции для вычисления на графическом процессоре:

1. Сортировка массива чисел
2. Линейный поиск в неотсортированном массиве чисел
3. Векторизованное сложение, вычитание, произведение и деление массивов чисел
4. Векторизованное возведение в степень массивов чисел
5. Интерполяция(обычная интерполяция вроде не сильно сложная, не знаю есть ли смысл в векторизованной интерполяции массива)