**Делайте readme.txt!**

от Vladimir Frolov - Saturday, 21 March 2020, 14:47

Уважаемые коллеги! В описании задания не было это оговорено, но Вам необходимо сделать файл readme.txt, в котором описать все выполненные доп. задания, и сколько вы за них хотите -- по пунктам.

Смысл этого файла очень прост -- проверяющий не должен пропустить сделанные Вами доп. задания. Если вы что-то не укажете, велика вероятность того что он что-то пропустит и вы не получите дополнительные баллы.

Поэтому пождалуйста прикладывайте в архив readme.txt. Спасибо за понимание!

**Памятка про антиплагиат**

от Vladimir Frolov - Saturday, 28 March 2020, 14:51

Уважаемые, коллеги! В последнее время участились вопросы про плагиат, поэтому необходимо прояснить некоторые моменты.

1) Пользоваться открытыми источниками можно. Указывать их в ридми -- необходимо.

2) Копировать код напрямую из открытых источников, или тем более сдавать скачанный "реферат" из интернета не стоит.

 Используемая нами система антиплагиата MOSS безусловно покажет все такие случаи как плагиат, даже если вы сильно "поработаете" над кодом. Что будет дальше -- зависит от проверяющего. Он будет смотреть насколько много совпадений, чем ваши реализации отличаются и.т.д. Возможно будет что-то спрашивать Вас по почте.

 При желании конечно можно обмануть и MOSS, но зачем вам это надо не очень понятно. Первое задание в действительности очень простое. Его средний размер -- от 300 до 500 строк кода, и были случаи, когда оно делалось за одну лекцию. Однако же его плюс в том что за эти 500 строк вы действительно можете освоить существенную часть курса. Но это происходит только при условии самостоятельной реализации с чистого листа.

Конечно, если вы берётесь за компьютерную графику в первый раз, у Вас это займёт больше времени, но в любом случае объём задания небольшой, поэтому нет смысла сдирать существующие реализации из интернета. Если же вы всё-таки уже это сделали, советую сосредоточиться на реализации доп. заданий, которые позволят получить Вам уникальность для вашей работы. Срок сдачи я продлил.

**Памятка по поводу правил сдачи**

от Vladimir Frolov - Monday, 6 April 2020, 13:27

Уважаемые коллеги! Проверьте пожалуйста ваши архивы!

Ваши задания будут проверяться следующим образом:

1) Проверяющий скачивает и распаковывает архив на свою машину с каким-то дистрибутивом Linux, обычно это убунту.

2) Стандартный вариант сборки:

 mkdir build

 cd build

 cmake -DCMAKE\_BUILD\_TYPE=Release ..

 make -j 4

3) Запускает вашу программу из папки build (с аргументами командной строки которые указаны в задании). Убедитесь что пути к текстурам и другим файлам у Вас корректно указаны (либо при сборке файлы копируюся CMake-ом куда надо), и не нужно дополнительно куда-то их перемещать чтобы программа нашла необходимые внешние файлы.

4) Если не соберётся или не запустится, возможно, если Вам повезёт, он почитает ридми и поставит какие-то библиотеки которых нету у него на машине следуя вашей инструкции. Формально он может поставить вам 0 (если не получится собрать) или снизить баллы, поскольку в требованиях базы чёрным по белому написано что библиотеки должны быть частью проекта чтобы их не приходилось ставить.

5) Если вы не используете Cmake, то должны сделать сборку и запуск тривиальными для проверяющего.

**Пересдача осенью**

от Vladimir Frolov - Monday, 8 June 2020, 13:57

Зачёт осенью будет проходить следующим образом:

1) Выдаётся шаблон на OpenGL3 с шейдерами, который отрисовывает 1 треугольник или 1 квад.

2) В течении 4-6 часов (то есть во время отведённое для зачёта) студенты пишут пишут индивидуальные задания.

 Например, сделать скринсейвер из падающих и вращающихся треугольников.

 Или нарисовать вращающуюся пирамидку с разноцветными гранями.

Всем, кто не набрал 35 баллов и не получил зачёт, я советую за лето подготовиться к подобному сценарию.

На зачёте проверяется следующие основные знания/умения, которыми вы должны владеть:

1) Матрицы трансформации и их применение -- вращения, перенос, перспективная проекция.

 Некоторые варианты будут в 2D, некоторые в 3D.

 Необходимо уметь формировать эти матрицы (можно использовать библиотеки или брать код из интернета), передавать их в шейдер и применять к вершинам.

2) Работа с текстурами. Необходимо уметь загружать их из файла, передавать на GPU и применять в шейдере.

 Например, может быть следующее задание: дана программа отрисовки плоской сетки, карта высот и карта цвета.

 Необходимо нарисовать участок ландшафта, используя эти 2 карты.

3) Основы графического конвейера, взаимодействие вершинного и фрагментарного шейдера.

 Например, может быть задание -- отрисовать сложную геометрическую форму вроде листика дерева используя не более 2 треугольников.