

Компьютерная графика

Лабораторная работа №6.

Создание текстуры "Геометрический паттерн" с помощью процедурной генерации и вычислительного шейдера

Задание

Изучите проект программы и создайте шаблон геометрического паттерн согласно вашему варианту.

Ход выполнения работы

Шаг 1. Настройка проекта

Директория `CG/Practicum` содержит различные проекты. Чтобы выполнить текущее задание, откройте в отдельном окне проект, расположенный в директории `CG/Practicum/P6-Geometry_pattern`, с помощью настроенной ИСР.

Чтобы установить и настроить ИСР, можно воспользоваться инструкцией, расположенной в директории `CG/Practicum/P0-Getting_started`.

Шаг 2. Изучение проекта и необходимой литературы

Проект собирается из следующих основных файлов:

1. Файл конфигурации, предназначенный для сборки проекта, имеющий название `CMakeCache.txt` и расположенный в рабочей директории проекта.
2. Файлы программы, имеющие расширения `.h` (заголовочные файлы) и `.cpp` (файлы с исходным кодом), расположенные в директории `src`.
3. Файлы с различными типами шейдеров, имеющие расширение `.glsl` и расположенные в директории `shaders`.

Изучите указанную литературу и сделайте краткий конспект изученного материала, как минимум содержащий развернутые ответы на следующие контрольные вопросы:

1. Что такое вычислительный шейдер и чем он отличается от других шейдеров в OpenGL? Начиная с какой версии OpenGL поддерживаются вычислительные шейдеры?
2. Что такое вычислительное пространство (compute space), рабочие группы (work groups), вызовы (calls)? Как формируется процесс распределения вычислений на GPU?
3. Какие встроенные переменные поддерживают вычислительные шейдеры? Например, какова роль `gl_GlobalInvocationID` и как она используется?
4. Как определяется размер рабочей группы (`local_size_x`, `local_size_y`, `local_size_z`) в вычислительном шейдере?
5. Что такое work group и как она связана с `gl_WorkGroupID`?

6. Как настроить вычислительный конвейер (compute pipeline) в OpenGL для запуска вычислительного шейдера?
7. Для чего нужна функция `glDispatchCompute`? Что такое `barrier`? Для чего они нужны?
8. Какие происходят процесс чтения из текстуры и/или записи в текстуру данных с помощью вычислительных шейдеров?
9. Для создания чего могут быть использованы вычислительные шейдеры?

Список основной литературы:

1. OpenGL Wiki. Compute Shader
2. steps3D (А.В. Боресков). Вычислительные шейдеры в OpenGL
3. Learn OpenGL. Compute Shaders
4. Computer Graphics Shaders (M. Bailey). OpenGL Compute Shaders

Список дополнительной литературы:

1. GPU Particles с использованием Compute и Geometry шейдеров
2. GLSL. Language Specification

Шаг 3. *Построение геометрического паттерна согласно варианту.*

Согласно вашему варианту выберите геометрический паттерн (эти паттерны взяты с сайта VectorStock), который вам необходимо построить, из папки `CG/Practicum/P6-Geometry_pattern/task/tests`. Каждый геометрический паттерн пронумерован, т.е. это и есть вариант задания.

Создайте повторяющийся узор изображения с помощью выбранного геометрического паттерна. Поскольку части изображения повторяются, то необходимо научиться программно рисовать, как это сделано в примере.

Используя, полученные знания из предыдущих шагов и прошлого задания, модифицируйте файл с кодом вычислительного шейдера и, если необходимо, файлы с исходным кодом так, чтобы добиться желаемого результата.