

Компьютерная графика

Домашнее задание 2: Визуализатор сцен с освещением и тенями

2021

- Сделать визуализатор сцен в формате OBJ с текстурами и освещением по Фонгу

Задание

- Сделать визуализатор сцен в формате OBJ с текстурами и освещением по Фонгу
- Путь до сцены задаётся, например, параметром командной строки

Задание

- Сделать визуализатор сцен в формате OBJ с текстурами и освещением по Фонгу
- Путь до сцены задаётся, например, параметром командной строки
- Два источника света: направленный ('солнце') и точечный, оба как-то двигаются со временем

- Сделать визуализатор сцен в формате OBJ с текстурами и освещением по Фонгу
- Путь до сцены задаётся, например, параметром командной строки
- Два источника света: направленный ('солнце') и точечный, оба как-то двигаются со временем
- От обоих источников света есть тени, построенные алгоритмом shadow mapping (+ PCF или VSM, с размытием)

Задание

- Сделать визуализатор сцен в формате OBJ с текстурами и освещением по Фонгу
- Путь до сцены задаётся, например, параметром командной строки
- Два источника света: направленный ('солнце') и точечный, оба как-то двигаются со временем
- От обоих источников света есть тени, построенные алгоритмом shadow mapping (+ PCF или VSM, с размытием)
- Камера должна управляться пользователем (любым способом, главное – чтобы можно было всё разглядеть)

Хорошая сцена:

- Порядка 100k-1kk треугольников в сумме
- Порядка 100-1000 различных объектов
- У большинства объектов есть текстура альбедо

Рекомендую тестировать на *Crytek Sponza* и *Rungholt*, обе сцены можно найти здесь: casual-effects.com/data/index.html





- Одна из самых популярных тестовых сцен в 3D графике
- Модель атриума дворца в Дубровнике (Хорватия)



- Город, построенный в Minecraft, переведённый в формат OBJ
- Достаточно большая сцена (6 миллионов треугольников)
- Также содержит небольшую модель дома (house.obj), на которой хорошо тестировать

Wavefront OBJ – один из общепринятых форматов сцен

- Текстовый, достаточно простой для чтения
- Содержит координаты вершин, нормали и текстурные координаты
- Содержит индексы вершин, образующих треугольники
- Может содержать много объектов
- Может ссылаться на MTL-файл (Material Template Library), содержащий описания материалов
- MTL может в свою очередь ссылаться на текстуры (альбедо, нормали, и т.п.)

Что нужно из описания материалов:

- Текстура альбедо `map_Ka`
- Текстура прозрачности `map_d` (делаем discard, если в текстуре значение меньше 0.5; нужно использовать и в шейдере для shadow map)
- Коэффициент отражения (glossiness) `Ks`
- Показатель отражения (power) `Ns`

- Можно написать загрузчик руками (это несложно)
- Можно использовать какую-нибудь библиотеку, например Assimp или TinyOBJ
- Для загрузки текстур можно использовать какую-нибудь библиотеку, например SDL_image, stb_image или Boost.GIL
- **N.B.** Есть два соглашения о том, как идут текстурные координаты: снизу вверх, или сверху вниз. Если текстуры будут выглядеть странно, попробуйте отразить текстурные координаты по оси Y: $y = 1 - y$

- Можно иметь один общий набор VAO + VBO + EBO, и для каждого объекта хранить только диапазон индексов
- Можно иметь по VAO + VBO + EBO на каждый объект
- Скорее всего, у вас будет 1 draw call (`glDrawElements` или т.п.) на один объект
- Напоминание: для рендеринга VBO и EBO не нужны, только `glBindVertexArray` и `glDraw*`!

- Для теней от точечного источника света можно использовать одну subemap текстуру, а можно 6 обычных 2D текстур
- В любом случае, в сумме у вас будет 7 FBO: один для теней от солнца и 6 для теней от точечного источника
- Размер shadow map лучше взять побольше, если не будет тормозить (4096x4096)
- Размывать тени можно прямо в результирующем шейдере, при чтении из shadow map, а можно – в два прохода, сепарабельным фильтром за $O(N)$ (см. *separable gaussian blur*), за это будет больше баллов
- Радиус размытия – как можно больше, пока не начинает тормозить :)

- При загрузке полезно посчитать bounding box сцены, чтобы потом по нему вычислять матрицы для теней
- Лучше вычислять проекцию тени для направленного источника света используя видимую область камеры (чтобы ничего лишнего не попало в shadow map), за это будет больше баллов (см. конец лекции 4)
- Сцены могут быть разного размера \Rightarrow полезно делать скорость движения камеры и движение источника света пропорциональными размерам сцены

Баллы

- **2 балла:** геометрия сцены загружается и рисуется, движется камера
- **2 балла:** есть текстуры альбедо и прозрачности
- **1 балл:** есть ambient освещение + два источника света
- **2 балла:** есть тени от направленного источника света
 - **+1 балл:** PCF + blur
 - либо **+1 балл:** VSM + blur
 - либо **+2 балла:** VSM + separable blur
- **1 балл:** проекция тени для направленного источника вычисляется на основе камеры
- **3 балла:** есть тени от точечного источника света
 - **+1 балл:** PCF + blur
 - либо **+1 балл:** VSM + blur
 - либо **+2 балла:** VSM + separable blur

Всего: **15 баллов**

Защита заданий на практике **19 ноября**