Задание 4: Визуализация ланшавдта

Щербаков Александр, Максименко Денис, Фролов Владимир

28 ноября 2017 г.

Аннотация

Цель выполнения задания - изучение основ OpenGL3/4 в процессе выполнения задания с полезной нагрузкой. Материал подлежащий освоению:

- Основы представления геометрических объектов - полигональные сетки, карты высот (база).
- Графический конвейер, геометрические преобразования (база).
- Шейдерные программы (база).
- VBO/VAO. Объекты памяти OpenGL (база).
- Расчёт нормалей поверхности. Локальные модели освещения (база).
- Имитация микрорельефа (дополнительно).
- Карты теней (дополнительно).
- Рендеринг в текстуру (дополнительно).

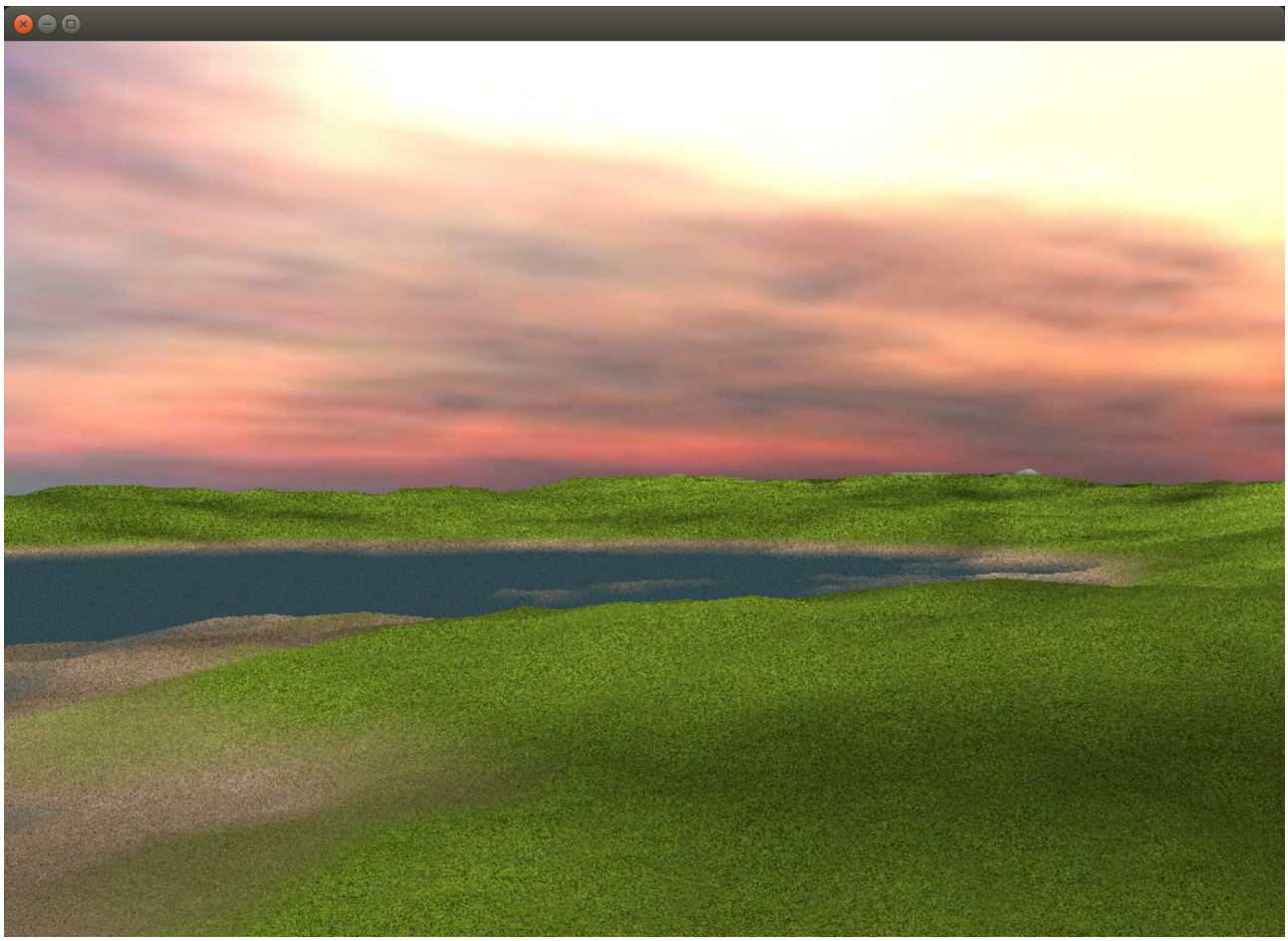


Рис. 1: Пример выполненного задания. 13 баллов (база + небо + прозрачная вода).

1 База (10 баллов)

Необходимо реализовать (1) генерацию карты высот и карты цвета ландшафта и (2) рендеринг ландшафта используя OpenGL3+.

Обязательными элементами задания являются:

- Визуализация ландшафта с текстурой (по умолчаию и по кнопке 1).
- Визуализация нормалей цветом (по кнопке 2).
- Ландшафт должен реалистично выглядеть. Освещение ландшафта может рассчитываться в шейдере либо быть запечено в текстуре. За отсутствие освещения баллы за базу могут быть снижены до половины.

2 Дополнительно

- Визуализация неба.
  - Кубические или сферические карты окружений - (1 балл).

  - Смена времени суток и пересчёт освещения на ландшафте и объектах при помощи сферических гармоник (до +6 баллов в зависимости от реалистичности).

- Тени (по кнопке «z» должна включаться визуализация карты глубины для объектов порендеренных их позиции источника - в противном случае баллы могут быть не засчитаны!):
  - Тени на ландшафт и другие объекты (простой Shadow Map) - (+3 балла).
  - Простой Shadow Map + PCF - (4 балла).
  - Более сложные и реалистичные методы теней (до +6)

- Визуализация водной поверхности:
  - Простая прозрачная вода - (1 балл).
  - Отражения неба в воде - (1 балл).
  - Отражения других объектов - (4 балла).
  - Имитация волн на воде - (1 балл за простую имитацию текстурой, до 4 баллов за реалистичную симуляцию волн).
  - Преломления - (2 балла).

- Визуализация дополнительных объектов.
  - Деревья - (2-4 балла в зависимости от реалистичности).
  - Трава/кусты/цветочки - (2-4 балла в зависимости от реалистичности).
  - Камни - (1 балл).
  - Дома, машины - (1-4 балла в зависимости от реалистичности).
  - Животные и люди - (1-4 балла в зависимости от реалистичности).

- Имитация микро-рельефа:
  - Простой нормалмаппинг (1 балл).
  - Простой нормалмаппинг + фильтрация по расстоянию чтобы не рябило в глазах в далеке (2 балла).
  - Parrallax Occlusion Mapping - (4 балла).
  - Использование тесселяции на GPU - (2-6 баллов).

- Визуализация масштабной поверхности:
  - Зацикленный или бесконечный ландшафт (до + 4 баллов).
  - Туман (+1 балл).

  - Использование уровней детализации для отдельных участков ландшафта и отображение каркасной модели (+2 балл).

- Использование профессиональных программ для генерации ландшафта и экспорт карты высот из них (не отменяет обязательную часть генерации карты высот!!!) (+1-2 балла).

- Визуализация нормалей стрелочками (+2)

3 Алгоритм генерации карты высот

Вам предлагается реализовать алгоритм diamond-square [1, 2]. Высоты генерируются на сетке размером $2^n + 1$. Самым распространенным алгоритмом генерации карты высот, дающим одни из самых реалистичных результатов, является алгоритм diamond-square (или square-diamond), расширение одномерного алгоритма midpoint displacement на двумерную плоскость.

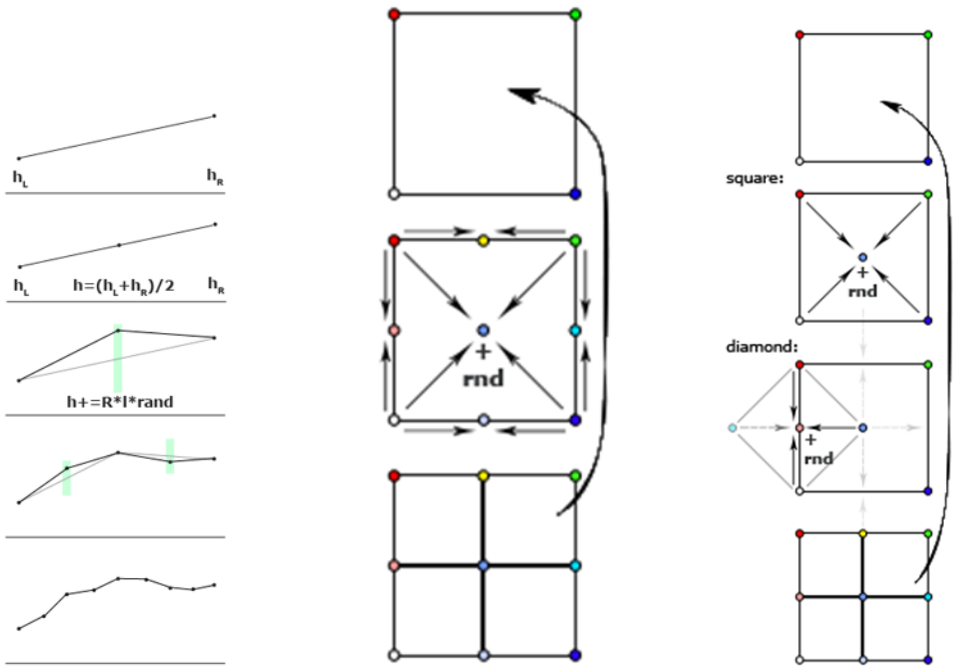
Алгоритм «midpoint displacement» (работает на одномерном отрезке, рис. 2): Изначально мы любым образом задаем высоту на концах отрезка и разбиваем его точкой посередине на два под-отрезка. Эту точку мы смещаем на случайную величину и повторяем разбиение и смещение для каждого из полученных под-отрезков. И так далее — пока отрезки не станут длиной в один пиксель. (см. рисунок справа). Важное замечание: случайные смещения должны быть пропорциональны длинам отрезков, на которых производятся разбиения. Например, мы разбиваем отрезок длиной l — тогда точка посередине него должна иметь высоту:

$$h = \frac{hL + hR}{2} + random(-R * l, R * l)$$

где hL и hR — высоты на левом и правом конце отрезка, а константа R определяет «шероховатость» (roughness) получающейся ломаной и является главным параметром в данном алгоритме.

Обобщение для двумерной карты высот: присваиваем случайные высоты четырем углам всей карты целиком и разобьем её на четыре равных квадрата. В каждом из них известно значение в одном из углов. Точка в центре получается усреднением высот всех 4 угловых точек, а каждая серединная точка на стороне большого квадрата — усреднением пары точек, лежащих на концах соответствующей стороны. Осталось привнести немного шума — сдвинуть случайным образом центральную точку вверх или вниз (в пределах, пропорциональных стороне квадрата). Повторить действия рекурсивно для полученных под-квадратов.

Алгоритм diamond-square — отличается от двумерного midpoint displacement тем, что состоит из двух шагов. Первый шаг — т. н. «square» — точно так же определяет центральную точку в квадрате путем усреднения угловых и добавлением собственно displacement'а



midpoint displacement midpoint displacement(2D) diamond square

Рис. 2: Иллюстрации к рассматриваемым алгоритмам.

— случайного отклонения. Второй шаг — «diamond» — призван определить высоту точек, лежащих на серединах сторон. Здесь усредняются не две точки — «сверху» и «снизу» (если говорить о точках на вертикальной стороне), но и пара точек «слева» и «справа» — то есть еще две полученных на шаге «square» центральных точки. Важно заметить, что эти две высоты, которые достались нам на предыдущем шаге, должны быть уже посчитаны — поэтому обсчет нужно вести «слоями», сначала для всех квадратов выполнить шаг «square» — затем для всех ромбов выполнить «diamon» — и перейти к меньшим квадратам (рисунок 2 справа)

Кстати, даже используя только один diamond-square, полученные значения (предварительно нормализованные, то есть в диапазоне от 0.0 до 1.0) полезно возвести в квадрат — это сделает равнины более пологими, а склоны гор более крутыми (помним про эрозию).

Список литературы

[1] *Diamond-Square.* // Wikipedia. url = https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_Diamond-Square

[2] Steve Losh. *Terrain generation with diamond square.* // posted at june 27, 2016. url = http://stevelosh.com/blog/2016/06/diamond-square/