

ПРОЦЕДУРНАЯ ГЕНЕРАЦИЯ ЛАНДШАФТОВ

К.Б. Фёдоров

Научный руководитель: О.Б. Фофанов, к.т.н., доцент ОИТ
Томский политехнический университет

kbfl@tpu.ru

Введение

В настоящее время создание игр является очень популярным направлением. Современные технологии позволяют заниматься разработкой не только крупным компаниям, но и небольшим студиям. Процесс создания видеоигр состоит из многих этапов, одним из которых является создание «игрового мира». Это трудоёмкий процесс, занимающий большое количество времени. Следовательно, автоматизация данного процесса поможет существенно сэкономить время разработчиков.

Алгоритмическое моделирование природных и искусственных процессов, влияющих на внешний вид локаций, называют **процедурной генерацией ландшафта** [1]. Это популярное направление стало известно, в основном, благодаря компьютерным играм [2], однако в последние годы также используется и в кинематографе. Используя данный подход можно в короткие сроки получить уникальный ландшафт [3]. Также этот подход позволяет создавать «игровые миры» непосредственно перед запуском, что уменьшает размер самой игры и позволяет перепройти игру на новой карте.

Целью данного проекта является разработка специализированного набора средств для процедурной генерации ландшафта.

Требования к функциональности разрабатываемого ПО

При создании набора средств разработки были проанализированы основные требования к продуктам такого типа. На основании данного анализа были определены основные функциональные возможности разрабатываемого проекта «*Procedural World Generation SDK*».

- *Создание 3D- и 2D-ландшафтов.* Решение должно быть универсальным, вне зависимости от типа графики проекта.

- *Высокая степень кастомизации.* Мир, созданный совершенно случайно, может не оправдать ожиданий разработчиков. Задание размера мира, количества тех или иных объектов повысит контроль над генерацией и позволит избежать неприемлемых результатов.

- *Генерация естественных объектов.* Для реалистичного ландшафта необходимо наличие холмов, гор, равнин, рек, лесов и прочих объектов.

- *Генерация текстур.* Набор средств разработки должен позволять создавать реалистичные текстуры для ландшафта, а также карты нормалей к ним.

- *Разметка дорог и населенных пунктов.* Помимо естественных объектов в играх часто

существуют населенные пункты. На основании заданных пользователем данных об этих объектах генератор может устанавливать их автоматически, создавая необходимые структуры.

- *Экспорт данных в удобном для пользователя виде.* Экспорт должен быть представлен в виде 3D-моделей, текстур и дополнительной информации. Также он должен быть интегрирован в код игры.

- *Создание документации.* Для удобства разработчиков следует создать сайт, который позволит ознакомиться с проектом и предоставит полный доступ ко всем существующим на данный момент материалам.

На рисунке 1 представлена диаграмма последовательности действий при генерации нового ландшафта.



Рис. 1. Диаграмма последовательности работы Среда для разработки программного обеспечения

Для создания описываемого продукта была выбрана среда разработки Visual Studio 2017, инструменты генерации файлов решения CMake, а также язык программирования C++. Для сайта и документации был использован язык гипертекстовой разметки HTML, а также CSS3 и JavaScript.

В процессе разработки не было использовано сторонних библиотек. Работа с изображениями, 3D-моделями и файлами настроек была написана собственноручно на C++.

Реализация экспорта 3D-объектов

Алгоритм генерации ландшафта реализован с помощью двумерных массивов, заполненных градиентным шумом. Поэтому оптимальным решением будет экспорт полигональной сетки,

которая при виде сверху будет представлена множеством квадратов, а интенсивность шума в конкретной точке будет влиять на вертикальную составляющую каждой конкретной точки (рис. 2).

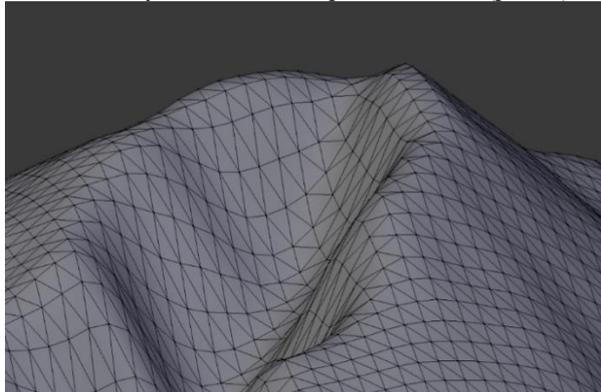


Рис. 2. Вид выбранной полигональной сетки

Важным этапом в создании 3D-моделей являются нормали. Эти вектора предназначены для хранения информации об отражении света от поверхности объекта. Нормали могут быть применены двумя способами: к полигонам (плоскостям) и к вершинам (рис. 3 и 4).

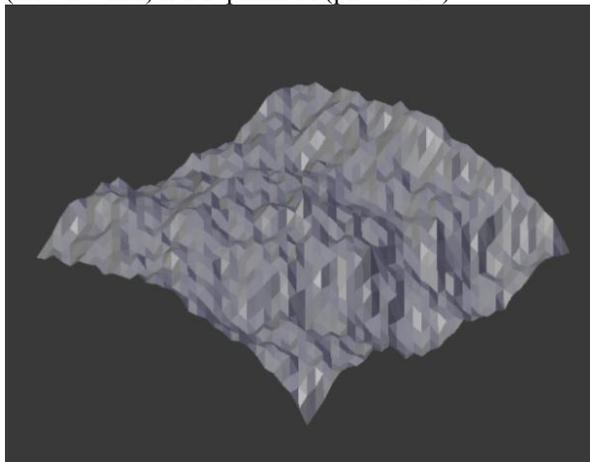


Рис. 3. Модель с нормальными, привязанными к полигонам

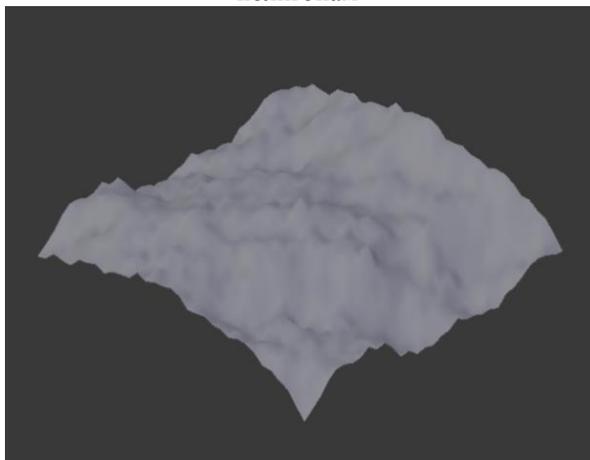


Рис. 4. Модель с нормальными, привязанными к вершинам

На рисунке 4 можно заметить, что четкие границы ребер исчезли.

Чтобы привязать нормали к вершинам, стоит рассчитать их для каждой точки. В этом заключается основная проблема, так как нормаль можно рассчитать только по трём точкам. Для решения данной задачи было решено рассчитывать нормали для всех прилегающих к точке плоскостей, а затем находить их среднее значение.

Информация о текстурных координатах также крайне важна при работе с 3D-моделями. Эти данные привязаны к полигонам. Благодаря формированию полигональной сетки в виде квадратных клеток, текстурные координаты будут равномерно распределены по всей плоскости фигуры.

После экспорта моделей их можно использовать в любом приложении, поддерживающем работу с 3D-объектами. Стоит отметить, что не все программные средства способны правильно обрабатывать такие данные. Например, часть продуктов просмотра 3D-объектов не способна считывать нормали, а высчитывает их сама. Для таких случаев, предусмотрена опция отключения карт нормалей, чтобы не занимать лишнее дисковое пространство. Также такие продукты накладывают ограничения на порядок обхода вершин в полигонах – от этого зависит, какое из двух направлений примет вектор нормали.

Заключение

В результате выполнения проекта был создан набор средств разработки для процедурной генерации ландшафтов. Полученное решение способно создавать различные, заданные с помощью настроек, местности. К проекту прилагается документация, примеры и руководство пользователя.

На текущее время проводится тестирование уже существующего функционала и добавление нового. В дальнейшем планируется создать дополнительные инструменты для упрощения работы с генератором, а также интеграция с пакетом Unreal Engine 4.

Список использованных источников

1. Процедурная генерация планетарных карт [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/313420/> (дата обращения: 20.11.2018).

2. Генерация ландшафта как в Minecraft [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/128368/> (дата обращения: 20.11.2018).

3. Спецификация формата OBJ [Электронный ресурс]. – Режим доступа: <http://paulbourke.net/dataformats/obj> (дата обращения: 21.11.2018).