

**ПРИМЕНЕНИЕ АЛГОРИТМА MIDPOINT-DISPLACEMENT В
ПРОЦЕДУРНОЙ ГЕНЕРАЦИИ ЛАНДШАФТОВ**
USING ALGORITHMS «MIDPOINT-DISPLACEMENT» IN PROCEDURAL
GENERATION LANDSCAPES

УДК 004.921

Барсуков Алексей Валерьевич,

студент,

Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
ИТМО», г. Санкт-Петербург,

Barsukov A.V. alexbarsukov@hotmail.com

Аннотация

В этой статье рассматривается способ создания фрактальных ландшафтов и рек, а также предлагается частичное решение, которое включает в себя реку и ландшафт, построенный по алгоритму «midpoint-displacement». С прогрессом вычислительной мощности персональных устройств стало возможно генерировать те самые ландшафты в реальном времени и без участия человека, переложив эту задачу на алгоритмы. В этой статье предоставлен быстрый способ генерации ландшафта, внешний вид которого приближен к реальности. Также предлагаются правки для увеличения скорости генерации. И обсуждаются некоторые вопросы, которые касаются доведения ландшафта до высокого уровня.

Summary

This article discusses the way to create fractal landscapes and rivers, and also offers a partial solution that includes a river and a landscape built using the midpoint-displacement algorithm. With the progress of the computing power of personal devices, it became possible to generate the same landscapes in real time and without human intervention, shifting this task to algorithms. This article provides a quick way to generate a landscape whose appearance is close to reality. Edits are also offered to increase the speed of generation. And some issues are discussed that relate to bringing the landscape to a high level.

Ключевые слова: процедурная генерация ландшафта, midpoint-displacement, фракталы, процедурная генерация рек.

Keywords: procedural generation landscape, midpoint-displacement, fractals landscape, procedural generation rivers.

Процедурная генерация — это способ создания контента «без участия человека». То есть контент создается автоматически с помощью вычислительных алгоритмов. Эти алгоритмы создают модели домов, ландшафтов или других необходимых объектов на основе заранее заданных параметров.

Проблемой является большая доля кропотливого ручного труда при генерации ландшафтов и другого контента. Поэтому возникает необходимость увеличить долю процедурной генерации контента.

Целью настоящей исследовательской работы является исследование и изучение алгоритмов процедурной генерации контента (ландшафтов), а также способов программной реализации этих алгоритмов визуализации и выявление сильных и слабых сторон этих алгоритмов.

В исследованиях алгоритмов ПГК особенный интерес представляет проблема процедурной генерации правил, которые лежат в основе всех задач. Решение данной проблемы обладает практической значимостью, поскольку процедурная генерация правил позволит не только существенно разнообразить контент за счет динамического изменения правил. Кроме того, системы генерации правил могут быть полезны исследователям алгоритмов искусственного интеллекта в сфере универсальных игровых программ, а также игровым дизайнерам для быстрого создания и тестирования различных игровых прототипов.

Процедурная генерация контента становится очень популярной. В настоящее время это применяется в кинематографе при генерации сцен второго плана, а также в игровой индустрии. Именно процедурная генерация позволяет создавать изображения со специальными характеристиками динамическим образом. Таким образом можно получить большое количество моделей в короткие сроки без кропотливого ручного труда.

Так как весь контент генерируется, такой подход позволяет во много раз сократить объем требуемого пространства на накопителе. Более того, с такой технологией не требуется никаких загрузок во время игры — все процедуры загружаются во время запуска приложения. Несмотря на то, что все генерируется процедурно, каждый объект, каждый кустик и кирпич имеют свой собственный серийный код, что открывает возможности кастомизации.

Раньше художникам приходилось либо использовать реальные фотографии и на их основе делать модель, либо создавать текстуру с нуля. У этих подходов

есть существенные недостатки, например такие как отсутствие быстрого редактирования этих моделей, а также невозможность вносить дополнительные параметры по ходу. Эти два подхода являются представителями рутинного труда.

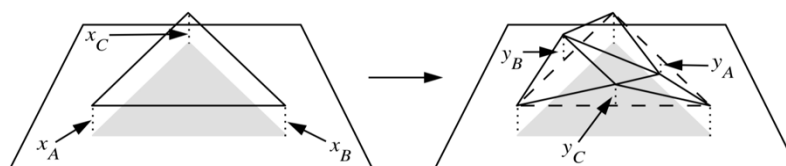
Преимуществами процедурной генерации моделей являются:

- Внесение изменений в модель на любом этапе работы.
- Изменение размеров модели без потерь качества.
- Скорость генерации контента
- Небольшой размер данных. Так как этот размер не зависит от разрешения изображения. В файле хранится только алгоритм.

Создание контента содержит в себе большое количество проблем. Например, проблема большого количества необходимых работ для создания какой-либо модели, проблема статичности моделей (они тяжело поддаются изменениям), а ведь в процессе разработки требования к модели могут изменяться.

Мы начнем с обзора базовой конструкции смещения средней точки. Описание конструкции кривой кривой приведено ниже. Затем эти две конструкции соотносятся как разные грани одного и того же контекстно-зависимого процесса подчинения треугольника и объединяются в модель ландшафта, в котором присутствуют горы и реки. Статья завершается списком тем, открытых для дальнейшего исследования.

Рисунок 11 –генерация фрактальных гор с использованием метода



«midpoint-displacment»

Рассмотрение данной темы хотелось бы начать с пересмотренного алгоритма «Midpoint-displacment» В простейшей версии метода «Midpoint displacment» начальный горизонтальный треугольник подразделяется на четыре меньших треугольника, а вновь созданные вершины смещаются по вертикали случайными значениями. Аналогичный процесс повторяется для каждого из меньших треугольников, затем для каждого из их потомков, пока не будет достигнут заданный предел рекурсии. Информация, внутренняя для исходного треугольника, никогда не передается соседним треугольникам. Хотя эта точка зрения была поддержана в литературе [10], [12, С. 244], она игнорирует связь, которая имеет место между соседними треугольниками.

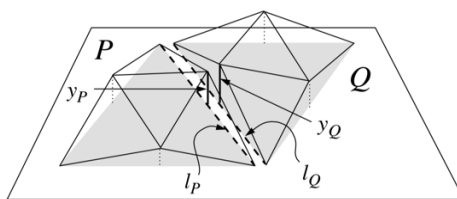


Рисунок 2 – алгоритм «midpoint-displacement»

Метод смещения средней точки является контекстно-зависимым. После разбиения треугольников P и Q средние точки совпадающих ребер l_p и l_q смещаются векторами равной длины y_p y_q . В практических реализациях линии l_p и l_q сворачиваются в одно ребро l , разделяемое треугольниками P и Q .

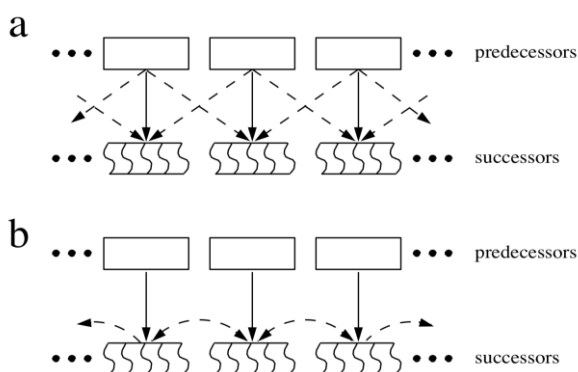


Рисунок 3 – метод «midpoint-displacement»

Как показано на рисунке 3, мы не можем применить производство рисунка 2 независимо к треугольникам P и Q , разделяющим общий край l , поскольку смещение средней точки l должно быть одинаковым как для P , так и для Q [3]. Таким образом, производственный экземпляр, примененный к треугольнику P , зависит от того, который применяется к треугольнику Q , и наоборот. Это подразумевает передачу информации между P и Q , хотя ее природа отличается от той, которая обычно рассматривается в теории формального языка. Традиционно, набор применимых производств ограничен соседями строгого предшественника (заменяемый символ). С другой стороны, в «midpoint-displacement» каждое производство ограничено преемниками производств, примененных одновременно к соседним треугольникам (рис.4). Тем не менее, в обоих случаях результат генерации зависит от его соседей, и в этом смысле оба типа генерации зависят от контекста.

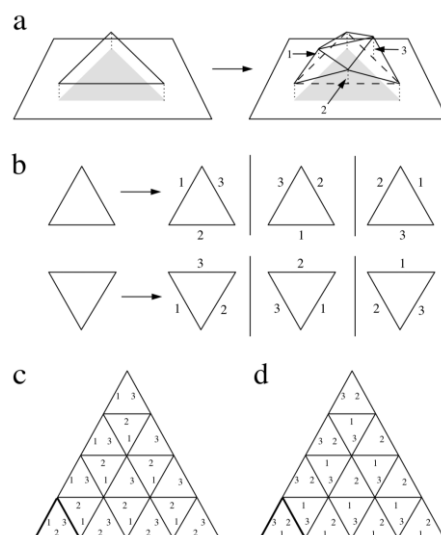


Рисунок 4 – влияние треугольников друг на друга

Нужно заметить, что конечные элементы «с» и «d» (рис. 4) можно воспринимать как мозаику. На рисунке 4 заметно то, что стороны которые совпадают друг с другом имеют одинаковые значения. Это обязательный признак.

При рассмотрении данной темы нельзя не затронуть такой метод, как сквиг-кривые. Простейшей сквиг-кривой является случайная фрактальная кривая. Эта модель русла реки, созданная по образу и подобию известных картинок из учебников по географии и геологии, на которых изображены последовательные этапы развития реки, промывающей себе путь через долину; с каждым этапом будущее русло приобретает все более четкие очертания.

Перед началом i -го этапа река течет в «предсквиг – долине», составленной из ячеек правильной треугольной решетки со стороной 2^i . Разумеется, ни в одну ячейку нельзя наведываться более чем однажды, к тому же каждое звено в решетке должно касаться сторонами двух соседних звеньев, оставляя третью сторону «свободной».

На i -м этапе эта предсквиг – кривая заменяется другой, более точной, построенной на интерполированной решетке со стороной 2^{i-1} . Очевидно, что предсквиг – кривая $(i+1)$ -го порядка обязательно содержит половину каждой стороны, общей для двух соседних звеньев i -го порядка. Верно также строгое обратное утверждение, а именно: положение общих (несвободных) половин сторон однозначно определяет вид предсквиг – кривой i -го порядка.

Далее перейдем к рассмотрению аналогичного метода под названием симметрично-случайные сквиг-кривые. Будем выбирать половину стороны, которую следует оставить свободной, случайным образом, полагая, что каждый из вариантов равновероятен. Тогда число звеньев $(i+1)$ -го порядка

внутри звена i -го порядка равно 1 с вероятностью $\frac{1}{4}$ или 3 с вероятностью $\frac{3}{4}$. Среднее значение составит 2,5.

С каждым этапом долина сужается и в пределе асимптотически сходится в некую фрактальную кривую. Я, естественно, предположил, что размерность этой предельной кривой равна $D = \frac{\ln 2,5}{\ln 2} = 1,321$

Следующий рассматриваемый метод в данной статье носит название ассиметрично-случайные сквиг-кривые. Предположим, что вероятность того, что после разделения стороны треугольника на две половины поддолина выберет, скажем, «левую», не равна $\frac{1}{2}$. Понятия «правый» и «левый» можно определять либо с позиции наблюдателя, смотрящего в направлении вниз по реке, либо с позиции наблюдателя, находящегося в центре разделяемого треугольника.

В первом случае $D = \frac{\ln[3-p^2-(1-p^2)]}{\ln 2}$ и может принимать значения от 1 до $\frac{\ln 2,5}{\ln 2}$. Во втором случае $D = \frac{\ln[3-2p-(1-p)]}{\ln 2}$ и может принимать значения от $\frac{\ln 2,5}{\ln 2}$ до $\frac{\ln 3}{\ln 2}$. В общей сложности допустимы все значения D от 1 до $\frac{\ln 3}{\ln 2}$.

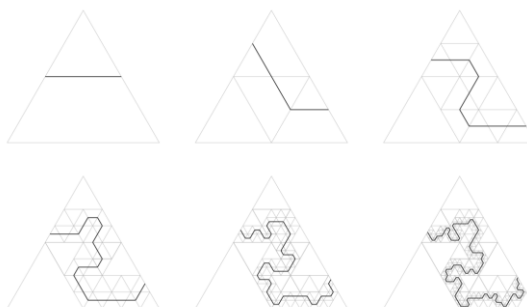


Рисунок 6 – построение сквиг-кривых

Ветвящиеся сквиг-кривые – еще один из методов, которые необходимо осветить в рамках данного исследования. Вернемся к построению речного русла. Вот мы заменили треугольный интервал долины участком поддолины, состоящим из одного или трех подтреугольников; представьте теперь, что оставшиеся три (или один) подтреугольника вдруг решают отвести от основного русла собственную поддолину. Построение нового русла полностью определяется уже известным процессом. Точки, в которых подреки пересекают границы между треугольниками, выбираются с помощью той же системы, что используется в главной реке. В пределе конструкция сходится к древовидной кривой, которая заполняет треугольник случайным образом, как можно видеть на рисунке:

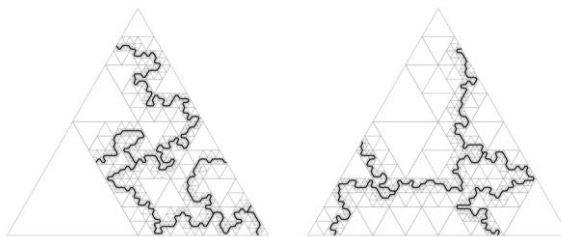


Рисунок 12 – ветвление сквиг-кривых

Подводя итоги, можно сделать вывод, что в статье была рассмотрена генерация фрактальных ландшафтов с помощью алгоритма «midpoint-displacement» и алгоритма генерации кривых Сквига для генерации кривой линии рек. Используя эти техники, можно добиться высокой степени реализма при генерации ландшафта, но две проблемы остаются открытыми для дальнейших исследований в этой области:

- Река течет на постоянной высоте, однако в горных ландшафтах не стоит пренебрегать уклонами, чтобы допустить реализацию водопадов.
- Река не имеет притоков. Можно дополнить кривые притоками, для этого нужно разработать способ генерации точек стыковок.

Литература

1. More than 1.2 billion people are playing games [Электронный ресурс] / Dean Takahashi // GamesBeat. – Режим доступа: <http://venturebeat.com/2013/11/25/more-than-1-2-billion-people-are-playing-games/> – яз. англ.
2. Supreme Court sees video games as art [Электронный ресурс] / John D. Sutter // CNN.com. – Режим доступа: <http://edition.cnn.com/2011/TECH/gaming.gadgets/06/27/supreme.court.video.game.art/> – яз. англ.
3. Map representations [Электронный ресурс] // Amit's Thoughts on Pathfinding. – Режим доступа: <http://theory.stanford.edu/~amitp/GameProgramming/MapRepresentations.html> – яз. англ.
4. Basic BSP Dungeon generation [Электронный ресурс] // RogueBasin. – Режим доступа: http://www.roguebasin.com/index.php?title=Basic_BSP_Dungeon_generation – яз.англ.
5. Random level generation in Wasteland Kings [Электронный ресурс] / Jan Willem Nijman // Vlambeer . – Режим доступа: <http://www.vlambeer.com/2013/04/02/random-level-generation-in-wasteland-kings/> – яз. англ.
6. Generate Random Cave Levels Using Cellular Automata [Электронный ресурс] / Michael Cook // EnvatoTuts+. – Режим доступа:

- <http://gamedevelopment.tutsplus.com/tutorials/generate-random-cave-levels-using-cellular-automata--gamedev-9664> – яз. англ.
7. Пишем игровой движок [Электронный ресурс] // Хабрахабр. – Режим доступа: <https://habrahabr.ru/post/102930/> – яз. рус.
 8. Unreal Engine FAQ [Электронный ресурс] // Epic Games : Часто задаваемые вопросы и ответы по движку Unreal Engine 4. – Режим доступа: <https://www.unrealengine.com/faq> – яз. англ.
 9. Unreal Engine Features [Электронный ресурс] // Epic Games : Возможности движка Unreal Engine 4. – Режим доступа: <https://www.unrealengine.com/unreal-engine-4> – яз. англ.
 10. IntelliSense [Электронный ресурс] // Википедия, свободная энциклопедия – Режим доступа: <https://ru.wikipedia.org/wiki/IntelliSense> – яз. рус.
 11. Microsoft software license terms | Visual Studio Community 2013 [Электронный ресурс] // Microsoft : Лицензионное соглашение Microsoft – Режим доступа: <https://www.visualstudio.com/en-us/dn877550.aspx> – яз. англ.
 12. C++ [Электронный ресурс] // Энциклопедия языков программирования – Режим доступа: <http://progopedia.ru/language/c-plus-plus/> – яз. рус.
 13. Current Status : Standard C++ [Электронный ресурс] // Standard C++ Foundation : Статус стандарта языка C++ – Режим доступа: <https://isocpp.org/std/status> – яз. англ.
 14. Unreal Engine API Reference [Электронный ресурс] // Epic Games : Документация Unreal Engine 4 – Режим доступа: <https://docs.unrealengine.com/latest/INT/API/index.html> – яз. англ.
 15. В.В. Галицкий, Е.В. Мироненко “Мозаика Вороного на плоскости”
 16. Static Meshes | Unreal Engine [Электронный ресурс] // Epic Games. – Режим доступа: <https://docs.unrealengine.com/latest/INT/Engine/Content/Types/StaticMeshes/> – яз. англ.

Literature

1. More than 1.2 billion people are playing games [Electronic resource] / Dean Takahashi // GamesBeat. - Access mode: <http://venturebeat.com/2013/11/25/more-than-1-2-billion-people-are-playing-games/> - language English
2. Supreme Court sees video games as art [Electronic resource] / John D. Sutter // CNN.com. - Access mode: <http://edition.cnn.com/2011/TECH/gaming.gadgets/06/27/supreme.court.video.game.art/> - language English

3. Map representations [Electronic resource] // Amit's Thoughts on Pathfinding.
- Access mode:
<http://theory.stanford.edu/~amitp/GameProgramming/MapRepresentations.html> - language English
4. Basic BSP Dungeon generation [Electronic resource] // RogueBasin. - Access mode:
http://www.roguebasin.com/index.php?title=Basic_BSP_Dungeon_generation – English
5. Random level generation in Wasteland Kings [Electronic resource] / Jan Willem Nijman // Vlambeer. - Access mode:
<http://www.vlambeer.com/2013/04/02/random-level-generation-in-wasteland-kings/> - language English
6. Generate Random Cave Levels Using Cellular Automata [Electronic resource] / Michael Cook // EnvatoTuts +. - Access mode:
<http://gamedevelopment.tutsplus.com/tutorials/generate-random-cave-levels-using-cellular-automata--gamedev-9664> - language English
7. We write a game engine [Electronic resource] // Habrahabr. - Access mode:
<https://habrahabr.ru/post/102930/> - language Russian
8. Unreal Engine FAQ [Electronic resource] // Epic Games: Frequently asked questions and answers on the Unreal Engine 4. - Access mode:
<https://www.unrealengine.com/faq> - language. English
9. Unreal Engine Features [Electronic resource] // Epic Games: Features of the Unreal Engine 4. - Access mode: <https://www.unrealengine.com/unreal-engine-4> - language. English
10. IntelliSense [Electronic resource] // Wikipedia, the free encyclopedia - Access mode: <https://ru.wikipedia.org/wiki/IntelliSense> - language. Russian
11. Microsoft software license terms | Visual Studio Community 2013 [Electronic resource] // Microsoft: Microsoft License Agreement - Access mode:
<https://www.visualstudio.com/en-us/dn877550.aspx> - language. English
12. C ++ [Electronic resource] // Encyclopedia of programming languages - Access mode: <http://progopedia.ru/language/c-plus-plus/> - language. Russian
13. Current Status: Standard C ++ [Electronic resource] // Standard C ++ Foundation: Status of the C ++ language standard - Access mode:
<https://isocpp.org/std/status> - language English
14. Unreal Engine API Reference [Electronic resource] // Epic Games: Documentation Unreal Engine 4 - Access mode:
<https://docs.unrealengine.com/latest/INT/API/index.html> - language English
15. V.V. Galitsky, E.V. Mironenko “Voronoi mosaic on the plane”

16. Static Meshes | Unreal Engine [Electronic resource] // Epic Games. - Access mode:
<https://docs.unrealengine.com/latest/INT/Engine/Content/Types/StaticMeshes>
/ - language English