



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.94

## ПРОЦЕСС СОЗДАНИЯ ПРОЦЕДУРНО–СГЕНЕРИРОВАННЫХ ЛОКАЦИЙ

Абуев А.Р., <sup>1</sup>Голикова Л.С., Поначугин А.В.

ФГБОУ ВО "НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ КОЗЬМЫ МИНИНА", Нижний Новгород, Россия, (603000, Нижегородская область, город Нижний Новгород, ул. Ульянова, д.1), e-mail: <sup>1</sup>[lola\\_mironova\\_2014@mail.ru](mailto:lola_mironova_2014@mail.ru)

В современном мире игры набирают все большую популярность и индустрия их разработки набирает обороты. Игры имеют множество жанров и видов, от визуальных новелл до приключенческих экшен проектов. Существует жанр игр, где требуется автоматизировать процесс создания мира, сделать так, чтобы этот самый мир генерировался с помощью кода и каждый такой мир был уникальным и неповторимым. Такие миры называют процедурно сгенерированные. Цель данной статьи рассказать о том, какие методы и математические алгоритмы используют разработчики при создании подобных игр.

Ключевые слова: Генерация, алгоритм, видеоигра, ландшафт, псевдослучайное число.

## THE PROCESS OF CREATING PROCEDURALLY GENERATED LOCATIONS

Abuev A.R., <sup>1</sup>Golikova L.S., Ponachugin A.V.

«NIZHNY NOVGOROD STATE PEDAGOGICAL UNIVERSITY NAMED AFTER KOZMA MININ», Nizhny Novgorod, Russia, (603000, Nizhny Novgorod region, Nizhny Novgorod, uyanova str., 1), e-mail: <sup>1</sup>[lola\\_mironova\\_2014@mail.ru](mailto:lola_mironova_2014@mail.ru)

In the modern world, games are becoming increasingly popular and the industry of their development is gaining momentum. Games have many genres and types, from visual novels to adventure action projects. There is a genre of games where you need to automate the process of creating a world, make sure that this world is generated using code and each such world is unique and inimitable. Such worlds are called procedurally generated. The purpose of this article is to talk about what methods and mathematical algorithms developers use when creating such games.

Keywords: Generation, algorithm, game, landscape, pseudo-random number.

В современных играх очень большой диапазон жанров, а количество интересных механик, которые они в себя включают сложно уместить в голове. В данной статье рассматривается такой интересный с точки зрения программирования и алгоритмов элемент, механика игр, как процедурная генерация локаций, ландшафта, уровней.

Для начала стоит определиться с термином. Процедурная генерация – это процесс, когда условная локация игры не заранее проработана создателями игры, а была сгенерирована с помощью внутренних алгоритмов игры. Сам процесс написания таких алгоритмов зависит от жанра и сеттинга игры, а нужен он для разнообразия игрового процесса, так как с процедурно сгенерированными уровнями, каждое новое прохождение игры не похоже на предыдущее. [1]

Общий алгоритм процедурной генерации завязан на side или переводя термин на русский – зерне генерации. Это случайное число, которое влияет на всю генерацию. Алгоритм основывается на данном числе и создает ландшафт. Для примера возьмем популярную игру

“Minecraft”, многим известно, что каждый мир данной игры генерируется. Если объяснить просто, то игра создает мир по уровням вертикально, начиная с 0 координаты, сперва создается уровень камней, высота которых зависит от нашего «зерна», далее земля и так далее. [2] За счет случайных чисел и определенных алгоритмов по типу «между деревьями есть определенное расстояние» создается уникальный ландшафт.

В целом алгоритм примерно одинаковый и лишь немного отличается от игры к игре. Просто в зависимости от жанра добавляются или наоборот упрощаются/удаляются некие элементы. Если говорить более научными терминами, то на входе у нас есть случайное число, которое в дальнейшем влияет на выходные данные:

1. Геометрия уровня – расположение материалов;
2. Текстуры – определяет какой именно материал;
3. Музыка – какая музыка на уровне;
4. Сюжетные элементы – условные важные предметы или игровые персонажи.

Стоит обратить внимание, что сам процесс генерации гораздо сложнее, чем он описан выше. Порой может использоваться простой алгоритм, а в иных случаях используются более сложные и продвинутое математические алгоритмы, которые позволяют создавать реалистичные локации.[3]

Чаще всего процедурная генерация встречается в играх с жанром рогалик и песочница. К первому лагерю можно отнести практически любого представителя данного жанра, например Hades, Dead Cells. А ко второму лагерю можно отнести, ранее упомянутый, minecraft, а также terraria и No Man Sky.

Последняя упомянутая игра – отличный пример процедурной генерации. Сама игра позиционирует себя как симулятор и РПГ, в игре пользователь может исследовать практически бесконечную вселенную, где каждый новый мир не похож на предыдущий. Такого эффекта можно добиться за счет процедурной генерации и сложной системы графов. Фактически сервер хранит лишь общий принцип генерации миров и информацию об исследованных мирах, так как хранить всю информацию о  $2^{64}$  в 64 степени системах, что есть в каждой из заявленных 256 галактик физически невозможно. Предполагается, что для генерации ландшафта на каждой планете используется «Шум Перлина» – это математический алгоритм, который позволяет создать уникальный геометрический объект, а точнее уникальную поверхность. [4]

Представим поверхность планеты в виде решетки, затем случайным образом определим высоту для каждого квадрата этой высота, далее нужно повторять данный процесс, накладывая сетку на сетку, пока не получится довольно реалистичная модель карты. Если говорить точнее, то задается не высота, а градиент, то есть направление изменения высоты, а затем идет интерполяция для разных уровней сеток, то есть находится промежуточный результат. Такой подход дает плавную и реалистичную генерацию, которая используется во многих играх с процедурной генерацией.

Есть и иной алгоритм, который позволяет делать процедурную генерацию ландшафта. Триангуляция Делоне – способ разбиения пространства на треугольники для создания сетки ландшафта. [5] Общий принцип алгоритма можно описать, как разбиение пространства на сетку, на сетке есть множество точек, которые в дальнейшем соединяются отрезками и в следствии превращаются в треугольники. Главное условие, что в описанной окружности любого такого треугольника не должно быть вершины другого треугольника, а значит этим

исключается неравномерность треугольников, исключается появление узких, длинных или чрезмерно широких треугольников. Данный процесс позволяет заполнить ландшафт объектами, например деревьями или строениями. Такой подход создает реалистичную картину ландшафта. Соединяя ранее описанный «Шум Перлина» с «Триангуляцией Делоне» можно получать уникальные по геометрии и разнообразию ландшафты. Сам процесс написания таких алгоритмов для игры может показаться довольно сложным,

Однако, не всегда разработчикам нужно столь усложнять процесс генерации уровней, локации или как в последнем примере, целых планет. В инди играх, то есть в играх, где команда разработчиков не очень большая, а требования к игре не сильно высокие, используется псевдогенерация или же упрощенная генерация. Например, можно основываться только на случайных числах, дополняя алгоритм неким правилом.

Допустим существует игра в жанре рогалик и разработчикам нужно создать процедурную генерацию уровней. У разработчиков есть набор текстур, врагов и наград. В данном случае, для генерации уровней может использовать, как раз таки именно упрощенный алгоритм. В своей сути алгоритм может генерировать локацию, начиная со случайного размера в заданных диапазонах. Далее, может выбираться форма арены из заготовленных, потом идет разметка территории случайным набором текстур и, под конец, добавление случайного количества определённых врагов из набора под текущий уровень.

Затронем тему, упомянутую ранее, а именно «зерно» или же «seed». Это случайное число, что влияет на генерацию мира. Зерно представляет собой определенную последовательность символов. [6] Чем длиннее зерно, тем больше вариантов генерации потенциально существует. Само же зерно влияет на генератор псевдослучайных чисел (далее ГПСЧ). ГПСЧ создает разнообразные последовательности чисел, однако все они определены этим самым зерном.

Алгоритм гарантирует, что у любого игрока один и тот же side вызовет идентичные генерации локаций. Это может показаться бесполезно, ведь суть процедурной генерации заключается в том, чтобы создавать уникальные и неотличимые миры, но, на самом деле, вероятность того, что у двух игроков будут сгенерированы одинаковые миры крайне мала, а возможность делиться «зернами генерации» позволяет развить сообщество игры. Также, это позволяет экономить место, ведь за счет хранения не всего мира, а лишь ключа к его генерации можно сократить расходы на содержание серверов, если это мультиплеерная игра, или же просто оптимизировать игру, если это одиночная игра с возможностью кооператива.

В заключении можно сказать, что на данный момент существует множество вариантов, как именно можно создавать процедурную генерацию в играх. Выбор того или иного алгоритма всецело зависит от личных предпочтений команды разработки, а также от жанра и типа игры. Правильная и корректная разработка алгоритма гарантирует уникальные и неповторимые миры. Сама же процедурная генерация гораздо более масштабная вещь, нежели, чем просто инструмент для разработки игр. В своей сути, она используется также в геологии для симуляций ландшафта, в астрофизике - для моделирования вселенной, в архитектуре - для создания вариантов дизайна городов или зданий. Одним словом, процедурная генерация может найти применение в разных научных областях.

## Список литературы

1. Волкова О.Н., Кривдин В.А., Рулин И.А. Анализ применения методов геймификации в образовании // В сборнике: Образование в цифровую эпоху: опыт, проблемы и перспективы. Нижний Новгород, 2024. С. 68-72.
2. Груздева М.Л., Волков С.М., Толчин Д.С. Развитие инженерного мышления у школьников на занятиях по 3d-моделированию // В сборнике: Интеграция информационных технологий в систему профессионального и дополнительного образования. Материалы региональной научно-практической конференции. Нижний Новгород, 2023. С. 16-20.
3. Джозеф Х. Unity в действии. Мультиплатформенная разработка на C#. Спб: Питер, 2016. 336 с.
4. Жучков Е.А., Жучкова Т.Г. Разработка и реализация алгоритма процедурной генерации ландшафта на основе шума Перлина. // В сборнике: Актуальные проблемы модернизации математического и естественно–научного образования. Сборник научных трудов по материалам Всероссийской научно-методической конференции. 2020. С. 112–114.
5. Качурин А. В., Ильичев В. Ю. Основы формирования визуальной картины шума Перлина при программировании на языке Python // E–Scio [Электронный ресурс]: Электронное периодическое издание «E–Scio.ru» Режим доступа: <http://e-scio.ru/wp-content/uploads/2021/08/Качурин-А.-В.-Ильичев-В.-Ю.pdf>.
6. Распределение вероятностей // Большая российская энциклопедия: [в 35 т.] / гл. ред. Ю. С. Осипов. — М.: Большая российская энциклопедия, 2004—2017.

## References

1. Volkova O.N., Krivdin V.A., Rulin I.A. Analysis of the use of gamification methods in education // in the collection: education in the digital age: experience, problems and prospects. Nizhny Novgorod, 2024. pp. 68-72.
  2. M. Gruzdeva.L., Volkov S.M., Tolchin D.S. The development of engineering thinking among schoolchildren in 3D modeling classes // in the collection: integration of information technologies into the system of professional and additional education. Materials of the regional scientific and practical conference. Nizhny Novgorod, 2023. pp. 16-20.
  3. Joseph H. Unity in action. Multiplatform development in C#. St. Petersburg: Peter, 2016. p.336
  4. E. Zhuchkova.A., Zhuchkova T.G. Development and implementation of an algorithm for procedural landscape generation based on Perlin noise. // In the collection: actual problems of modernization of mathematical and natural science education. Collection of scientific papers based on the materials of the All-Russian Scientific and Methodological Conference. 2020. pp. 112-114.
  5. Kachurin A.V., Ilyichev V. Y. Fundamentals of forming a visual picture of Perlin noise in Python programming // e–Scio [Electronic resource]: Electronic periodical "E–Scio.ru " Access mode: <http://e-scio.ru/wp-content/uploads/2021/08/Качурин-А.-В.-Ильичев-В.-вып.pdf> .
  6. Probability distribution // Great Russian Encyclopedia: [in 35 volumes]/ / chief editor Yu. S. Osipov. — М.: Great Russian Encyclopedia, 2004-2017.
-