

УДК 004.514

## **ДОСЛІДЖЕННЯ ТА ВИКОРИСТАННЯ МЕТОДІВ ПРОЦЕДУРНОЇ ГЕНЕРАЦІЇ ДЛЯ СТВОРЕННЯ РІВНІВ У 2D ПЛАТФОРМЕРАХ**

Ясир Д.А.

Науковий керівник – старший викладач Новіков Ю.С.

Харківський національний університет радіоелектроніки, каф. ПІ  
м. Харків, Україна

тел.: +380960215229, e-mail: danylo.iasyr@nure.ua

This work is dedicated to the issue of creating levels by using PSG algorithms. The work of 3 main methods of procedural generation that can be used in creating levels was described. The evaluation of advantages of each algorithm is analyzed and published. Also, the methods were compared with each other and conclusions were drawn about the use of each.

Рівні можна вважати однією з найголовніших складових гри, і для того щоб проходити рівень було цікаво декілька разів, можна використати 2 підходи, зробити дуже деталізований, та великий рівень, чи можна генерувати локації, які кожен раз будуть різними. Другий підхід гарно пасує іграм з невеликими та простими локаціями.

Головними методами процедурної генерації можна вважати: Шум Перліна, Клітинний автомат, Генетичний алгоритм. Шум Перліна у 2d іграх зазвичай використовують для генерація ландшафту, за допомогою алгоритма отримуємо у кожній точці значення шуму та за допомогою цього значення можна визначати висоту поверхні, чи наприклад вид тайлу який потрібно розмістити. Клітинний автомат є найпопулярнішим методом процедурної генерації для створення рівнів завдяки своєї гнучкості та легкості створення, цей алгоритм заснований на завданні початкового стану комірок на сітці та подальшим застосуванням правил для оновлення стану кожної комірки на наступному кроці. Генетичний алгоритм це метод використовує еволюційні процеси для пошуку найліпшого рішення, для створення рівнів цей алгоритм використовується для генерації випадкових рівнів та оптимізації їх під конкретні цілі.

Головною ціллю дослідження є визначення кращого метода для створення процедурно генеруемого рівня у 2D платформері. Головною задачею процедурно сгенерованого ігрового рівня є створення для гравця цікавого досвіду завдяки балансу між складністю, та зрозумілістю. Тому алгоритм створення рівня має бути доволі гнучким щоб мати змогу давати гравцю кожен раз новий ігровий досвід, та в той же момент бути інтуїтивно зрозумілим щоб використовувати досвід отриманий на попередніх рівнях.

Одним з найпопулярнішим рішенням цієї задачі є алгоритм шуму Перліна, він може генерувати досить різні рівні і давати гравцю кожен раз новий досвід, у 2D платформерах він дуже гарно підходить для генерації

неоднорідних поверхностей, та печер завдяки тому що значення шуму в кожній точці може відрізнятись і різні типи поверхні будуть розподілені досить випадково. Складність же цього алгоритму полягає в досить високій затраті ресурсів тому що для кожної точці на мапі треба вирахувати значення шумової функції, і часова складність алгоритму зазвичай буде становити  $O(2^n)$  де  $n$  це кількість обчислень, тобто кількість комірок. Звісно цей алгоритм можна оптимізувати, але впровадження його в гру є досить важким і затратним за часом.

Клітинний автомат є досить простим у впровадженні рішенням. Цей алгоритм досягає результату за допомогою збільшення кількості правил але чим більше правил тим більше буде часова складність алгоритму і тим більше ресурсів буде використано. Часова складність алгоритму зазвичай буде дорівнювати  $O((nm)^2)$  де  $n$  це розмір сітки, а  $m$  кількість правил. Його результат залежить від кількості правил, тому він не підходить для великих рівнів з багатою кількістю правил бо складність дерева рішень буде над високою для впровадження.

Генетичний алгоритм є найскладнішим з приведених алгоритмів у впровадженні для генерації рівнів, хоча і результат буде більш збалансованим за складністю ніж попередні. Його використання потребує багато часу на генерацію, оцінку результатів, та вибору з них “нащадків” для подальших ітерацій алгоритму. Алгоритм має бути правильно налаштований за параметрами, підборі ефективної цільової функції, та багатої кількості ресурсів для праці.

Було проаналізовано основні методи процедурної генерації які можна застосувати для створення рівня у 2D платформері, і наведено найпоширеніші алгоритми процедурної генерації, деякі з підходів було використано для розробки власного ігрового застосунку, та перевірено на практиці. Для створення рішення процедурної генерації рівнів потрібно враховувати складність, обсяг створюваного рівня, потреби цільової аудиторії, та кількість ресурсів, і виходячи з цих даних обирати напрямки для створення алгоритму, наприклад для великого рівня можна поєднати алгоритми шума Перліна, для створення основи рівня, і Клітинного автомата для розташування на цьому рівні об’єкти за певними правилами.

Отже можна зробити висновки що вибір методу для генерації рівня цілком залежить від потреб які має розробник. Тому покращення обізнаності в існуючих методах і алгоритмах може дуже сильно спростити такі великі і складні задачі як наприклад процедурна генерація рівня.

Список використаних джерел:

1. [gamedeveloper.com](http://gamedeveloper.com) Genetic Algorithms in Games : Genetic Algorithms in Games (Part 1) ([gamedeveloper.com](http://gamedeveloper.com))
2. 2D procedurally generated world building in Unity : 2D procedurally generated world building in Unity ([gamedeveloper.com](http://gamedeveloper.com))
3. Wikipedia Шум Перліна: Шум Перліна — Вікіпедія ([wikipedia.org](http://wikipedia.org))