

МЕТОД АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ РІВНІВ КОМП'ЮТЕРНОЇ ГРИ НА ОСНОВІ ДИСПЕРСІЇ ВИПАДКОВОЇ ВЕЛИЧИНИ

Алмакаєва А.Є.

Науковий керівник – к.т.н., доцент Творошенко І.С.

Харківський національний університет радіоелектроніки
(61166, Харків, просп. Науки, 14, каф. Інформатики, тел. (057) 702-13-35)

e-mail: anastasiia.nechytailo@nure.ua

The work is devoted to the problem of automatic leveling in 2D video games. It is based on an already developed algorithm that independently places objects of various types on the game plane. The developed methods allow to generate an infinite number of levels and to increase their own weight without the intervention of the video game developer.

Thus, the algorithm developed for the implementation of the proposed methods is a standalone solution that ensures the infinity of the video game and does not require technical support after completion of all work on its creation. The principle of calculating the complexity of the game based on the analysis of the generated level is considered.

Робота присвячена створенню аркадної гри, де навмисно спрощений ігровий процес та короткий ігровий час, але підвищена інтенсивність, порівняно з іншими жанрами. До аркадних ігор відносяться платформери – це жанр відеоігор, ігровий процес в якому складається зі стрибків персонажа по різноманітних платформах та через перешкоди, збирання предметів, інколи необхідних для завершення рівня.

Складність розробки саме гри платформера полягає у тому, щоб створити велику кількість різноманітних зовні рівнів, що складаються із одного й того ж набору блоків. Суть роботи: спрощення розробки гри платформера та економія ресурсів для цього (часу, обладнання, грошових вкладень, залучення сторонніх спеціалістів у ігровій галузі) [1].

Розроблено алгоритм, що самостійно розміщує на ігровій площині об'єкти різних типів [1]: початок рівня, землю, блоки відновлення за умови смерті гравця, блок закінчення рівня. Розроблені методи дозволяють генерувати безкінечну кількість рівнів та самостійно підвищувати їх важкість без втручання розробника відеогри [2].

На основі цього алгоритму потрібно створити новий, що генерує пастки та персонажів «зłodіїв» на карті рівня.

Основними проблемами є:

- розміщення пасток у логічних місцях;
- розрахунок кількості пасток на карті, що була створена комп'ютером автоматично. Вона є повністю непередбачуваною за винятком кількості блоків на рівні та максимальної відстані між ними.

Пропонується використати для розрахунку аналіз дисперсії випадкової величини.

Дисперсія є мірою розсіяння значень випадкової величини відносно середнього значення розподілу. Більші значення дисперсії свідчать про більшій відхилення значень випадкової величини від центру розподілу.

Дисперсія дозволяє виміряти наскільки далеко випадкові значення розподілені від їх середнього значення. Дисперсія дорівнює квадрату стандартного відхилення, що є другим центральним моментом розподілу, і коваріації випадкової величини із самою собою

$$\sigma^2 \equiv D(X) = E[(X - \mu)^2] = \sum_x (x - \mu)^2 p(x),$$

де $\sigma = \sqrt{\sigma^2}$ і називається стандартним відхиленням величини X від її середнього значення μ ;

D – це оператор дисперсії випадкової величини.

Випадкова величина в нашому випадку буде належати області значень функції, що лежить в основі алгоритму розміщення блоків на рівні, та утворює множину $\{x = X\}$ елементарних подій, що позначають координати точок в просторі відносно початку координат, у якому знаходиться старт рівня. Значення середнього квадратичного відхилення розраховується в основному алгоритмі на основі фізичних ігрових можливостей моделі.

Таким чином, можливо використовувати значення дисперсії випадкової величини для розрахунку кількості персонажів «злочинців» та пасток на кожному рівні та рівномірного розміщення їх на карті. Логіка самого розміщення розробляється в окремому алгоритмі у вигляді команд та обмежень для кожного типу пасток, присутніх за сюжетом окремого рівня та окремої гри. Наприклад, персонажі не можуть знаходитися в повітрі, а мають стояти на землі, птахи не зможуть пролетіти крізь землю, елемент болота не має бути на вершині гори та інше.

Розроблений додаток може бути покращений за рахунок додавання в нього різноманітних блоків, пасток, неігрових персонажів, що ускладнюють рух за ігровою площиною. Його можна використовувати для розвитку уваги та швидкості реакції користувача шляхом занурення людини у непередбачувані умови, що постійно та непрогнозовано змінюються.

Список використаних джерел:

1. Tvoroshenko I.S., Kramarenko O.O. Software determination of the optimal route by geoinformation technologies. *Radio Electronics Computer Science Control*. 2019. Vol. 3. pp. 131–142.

2. Ahmad M. Ayaz, Tvoroshenko Irina, Baker Jalal Hasan and Lyashenko Vyacheslav Modeling the Structure of Intellectual Means of Decision-Making Using a System-Oriented NFO Approach. *International Journal of Emerging Trends in Engineering Research*. 2019. Vol. 7(11). pp. 460–465.