

УДК: 004.9

*Біємська А.С., магістрант,
Свінцицька О.М., к.е.н., доцент
Державний університет «Житомирська політехніка»*

ЗВАЖЕНИЙ ВИПАДКОВИЙ ВИБІР У ПРОЦЕДУРНІЙ ГЕНЕРАЦІЇ ІГРОВИХ СУТНОСТЕЙ

Процедурна генерація контенту (Procedural Content Generation, PCG) у сучасній розробці програмного забезпечення розважального характеру є фундаментальним підходом, що дозволяє автоматизувати створення ігрових ресурсів за допомогою алгоритмічних засобів замість ручного моделювання кожного елемента [1]. Використання стохастичних алгоритмів дозволяє вирішити проблему обмеженості контенту, забезпечуючи високу варіативність ігрового процесу та реігровальність.

Однак, застосування рівномірного розподілу ймовірностей (де всі події мають рівний шанс виникнення) часто є неприйнятним для задач ігрового балансу, оскільки це призводить до неконтрольованого хаосу та порушення кривої складності. Для забезпечення керованості генерації доцільним є використання методів, що дозволяють задавати ймовірність появи кожної сутності окремо. У даній роботі розглядається реалізація алгоритму зваженого випадкового вибору (Weighted Random Selection) як ефективного інструменту для динамічного спавну ігрових об'єктів.

Суть запропонованого підходу полягає у відмові від жорстко заданих послідовностей появи об'єктів на користь імовірнісної моделі, де кожному типу сутності (наприклад, класу ворога або типу ресурсу) присвоюється певний числовий коефіцієнт — вага (w). Вага є невід'ємним дійсним числом, що відображає відносну частоту появи даного елемента у загальній вибірці. Математично ймовірність $P(e_i)$ появи сутності e_i з множини доступних сутностей $E = \{e_1, e_2, \dots, e_n\}$ визначається як відношення ваги цієї сутності до суми ваг усіх елементів множини. Формула розрахунку виглядає наступним чином:

$$P(e_i) = \frac{w_i}{\sum_{j=1}^n w_j}$$

де w_i — вага i -го елемента, а знаменник дробу являє собою загальну вагу системи (W_{total}). Такий підхід дозволяє гнучко налаштовувати баланс: збільшення ваги одного елемента автоматично зменшує ймовірності появи інших, зберігаючи цілісність системи розподілу.

Алгоритмічна реалізація даного методу базується на концепції кумулятивного розподілу. Процес вибору можна представити як

розміщення відрізків довжиною w_i на одній числовій прямій загальною довжиною W_{total} . Випадкове число R , згенероване в діапазоні $[0, W_{total}]$ неминуче потрапить в один із цих відрізків, що і визначить обраний елемент.

Програмна логіка алгоритму може бути описана наступним псевдокодом:

```
Function GetWeightedRandomItem(ItemList):
    TotalWeight = 0
    For each Item in ItemList:
        TotalWeight += Item.Weight

    RandomValue = Random(0, TotalWeight)
    CumulativeWeight = 0

    For each Item in ItemList:
        CumulativeWeight += Item.Weight
        If RandomValue <= CumulativeWeight:
            Return Item

    Return Null (or DefaultItem)
```

Рис. 1 – Псевдокод процедурної генерації ігрових сутностей на основі методу зваженого випадкового вибору

У наведеному алгоритмі спочатку відбувається прохід по колекції для обчислення загальної суми ваг. На другому етапі генерується випадкове число. Третій етап — це лінійний пошук, де на кожній ітерації до акумулятора додається вага поточного елемента. Умова $RandomValue \leq CumulativeWeight$ спрацьовує саме тоді, коли кумулятивна сума "покриває" випадкове значення, що відповідає потраплянню у відповідний імовірнісний інтервал. Обчислювальна складність даного алгоритму становить $O(N)$, де N — кількість типів сутностей, що є прийнятним для систем реального часу з невеликою кількістю категорій об'єктів.

Таким чином, використання процедурної генерації на основі зваженого випадкового вибору забезпечує необхідний рівень абстракції між логікою прийняття рішень (яку складність встановити) та логікою виконання (якого саме ворога створити). Це дозволяє створювати гнучкі, масштабовані системи, де додавання нового контенту зводиться до розширення списку конфігурацій без втручання в ядро алгоритму, а варіативність ігрових ситуацій досягається шляхом маніпуляції числовими коефіцієнтами розподілу ймовірностей.

Список використаних джерел:

1. Shaker N., Togelius J., Nelson M. J. *Procedural Content Generation in Games*. Springer, 2016. URL: <https://link.springer.com/book/10.1007/978-3-319-42716-4> (дата звернення: 22.11.2025).