

## Процедурная генерация структуры прохождения в карточной компьютерной игре с элементами roguelike

*Шикин Илья Евгеньевич*

*Студент (магистр)*

Мордовский государственный университет им. Н.П. Огарёва, Факультет математики и информационных технологий, Саранск, Россия

*E-mail: ilya\_shikin\_2@mail.ru*

В условиях положительной динамики развития инди-сегмента [1] жанр карточных стратегий с элементами roguelike характеризуется высокой реиграбельностью, достигаемой за счёт вариативной генерации контента и нелинейного порядка прохождения. Однако существующие подходы к процедурной генерации либо требуют значительных вычислительных ресурсов, либо сложно поддаются балансировке [3, 4]. Возникает потребность в лёгком, контролируемом методе генерации, гарантирующем связность структуры и предсказуемое время выполнения.

В работе выбран конструктивный офлайн-метод процедурной генерации на базе ориентированного ациклического графа узлов, сочетающий параметризованную генерацию с жёсткими структурными ограничениями, что позволяет обеспечить контролируемость генерации. Программная реализация выполнена на языке C# в среде Unity [6]. Контентная база разделена на JSON-файлы [5]. Разделение данных и логики позволяет модифицировать параметры генерации без перекомпиляции проекта.

Разработан модуль с компонентной архитектурой (StoryMapGenerator, StoryMapManager, StoryContentLoader). Алгоритм послойной генерации определяет тип каждого узла (бой, событие, отдых, босс) на основе весовых коэффициентов из конфигурации главы. Вероятностная проверка связей (connectionChance) между слоями математически исключает «тушиковые» конфигурации и гарантирует проходимость маршрута [4]. Для усиления нарративной связности внедрена система тематических пулов, привязывающая врагов и события к ландшафту главы. Модульные события реализованы как пошаговые сценарии с поддержкой нелинейного ветвления через индексы переходов [2]. Визуализация графа построена на параметризуемом префабе с динамическим изменением четырёх состояний узлов. Система сохранений фиксирует топологию карты и мета-прогрессию игрока (здоровье, золото, колода) в отдельных JSON-файлах, обеспечивая корректное возобновление сессии [5, 6].

Таким образом, созданный модуль обеспечивает баланс между случайностью и контролируемостью, имеет низкую вычислительную сложность и простоту настройки через внешние конфигурации. В перспективе планируется интеграция адаптивной подстройки сложности на основе предыдущих прохождений.

### Источники и литература

- 1) Седых И.А. Индустрия компьютерных игр-2020. М.: Высшая школа экономики, Центр развития, 2020. 74 с.
- 2) Sahibgareeva G.F., Kugurakova V.V. Branched Structure Component for a Video Game Scenario Prototype Generator // CEUR Workshop Proceedings. 2021. Vol. 3066. P. 101–111.
- 3) Shaker N., Togelius J., Nelson M.J. Procedural Content Generation in Games: A Textbook and an Overview of Current Research. Springer, 2016.

- 4) Togelius J., Yannakakis G.N., Stanley K.O., Browne C. Search-based procedural content generation: A taxonomy and survey // IEEE Transactions on Computational Intelligence and AI in Games. 2011. Vol. 3, No. 3. P. 172–186.
- 5) JsonUtility. Unity Scripting API: <https://docs.unity3d.com/ScriptReference/JsonUtility.html>
- 6) Unity Technologies. Unity User Manual 2022.3: <https://docs.unity3d.com/Manual/>