

## ГЕНЕРАТИВНЫЕ МЕТОДЫ ДЛЯ СОЗДАНИЯ АДАПТИВНЫХ ИГРАБЕЛЬНЫХ ПЕРСОНАЖЕЙ В ИГРАХ-СЕРВИСАХ

Т. Р. Арсланов<sup>[0009-0002-1110-8503]</sup>

Казанский (Приволжский) федеральный университет, г. Казань, 420008, Россия

timars-mail@yandex.ru

### **Аннотация**

В условиях роста популярности игр-сервисов, требующих постоянного обновления контента для удержания игроков, актуальной задачей становится автоматизация создания адаптивных играбельных персонажей. Нами рассмотрены существующие подходы к генерации персонажей.

Текущие решения не предусматривают долгосрочную адаптацию под стиль игрока и зависят от ручного проектирования. Для устранения этого недостатка предложена трёхкомпонентная система, сочетающая моделирование действий игрока на основе реплеев<sup>1</sup>, генерацию персонажей через комбинирование механик и балансировку параметров, а также автоматическую валидацию через симуляции для оценки баланса и соответствия игровому стилю конкретного человека.

Работа обобщает современные исследования, демонстрируя потенциал генеративных методов для снижения ресурсозатрат при разработке игр-сервисов. Результаты могут быть использованы для ускорения прототипирования и поддержки долгосрочной жизнеспособности игровых проектов.

**Ключевые слова:** *игры-сервисы, гейм-дизайн, игровые персонажи, видео-игры, процедурная генерация.*

---

<sup>1</sup> Реплей – запись игрового процесса, позволяющая осуществлять его дальнейшее воспроизведение. Может быть реализована как последовательная запись состояний игровых объектов, запись последовательности входящих команд с сохранением сида (от англ. seed) – начального значения генератора псевдослучайных чисел – для воспроизведения «случайных» операций или как запись экрана.

## ВВЕДЕНИЕ

В последнее время в игровой индустрии большую популярность получила модель монетизации игры как сервиса или игры-сервисы. К концу 2023 года 95% студий разработчиков видеоигр работали над игрой-сервисом либо планировали её выпуск [1, 2].

От обычной игры игра-сервис отличается длительным жизненным циклом. Для такой бизнес-модели критически важно обеспечить высокое удержание игроков, чтобы они продолжали постоянно заходить в игру.

Такие возврат и удержание достигаются регулярным выпуском нового контента, привлекательного для игроков. Многие игры-сервисы сфокусированы на добавлении новых персонажей, ставя перед разработчиками задачу создания игровых персонажей (т. е. таких, которыми может управлять игрок), сбалансированных в рамках существующей игровой экосистемы и при этом интересных людям с различными стилями игры. Дополнительную сложность такая задача представляет для сервисов разных жанров, которые уже имеют большое количество персонажей, в том числе: MOBA<sup>2</sup> (как пример, игра *League of Legends* – 168 персонажей), tower defense<sup>3</sup> (*Arknights* – 337 персонажей), hero-shooter<sup>4</sup> (*Rainbow 6: Siege* – 74 персонажа).

Стоит отметить, что классические подходы к созданию персонажей в основном опираются на ручные проектирование, прототипирование и тестирование, требуя значительных временных и ресурсных затрат. Разработка персонажа может занимать порядка 10 недель (гейм-дизайнеры работают параллельно с другими специалистами) [3].

---

<sup>2</sup> MOBA – Multiplayer Online Battle Arena, многопользовательская онлайн-боевая арена. Жанр, в котором две команды игроков, в которых каждый игрок управляет своим персонажем, стремятся уничтожить главное здание противника.

<sup>3</sup> Tower defense – башенная защита. Жанр, в котором игроки стремятся помешать компьютерному противнику дойти до цели путём размещения башен (персонажей) и использования способностей.

<sup>4</sup> Hero-shooter – геройская «стрелялка». Жанр, в котором основной игровой процесс представлен стрельбой от лица различных персонажей, часто обладающих уникальными способностями.

## **СУЩЕСТВУЮЩИЕ ПОДХОДЫ**

Существует множество исследований, посвящённых как генерации игр, так и отдельных их составляющих. Например, в [4] описаны существующие классификации игровых механик, а также схема работы системы генерации игровых механик на основе классификации MGMT. В [5] представлена система Ludi, которая служит для описания и генерации новых настольных игр, а также для оценки их качества.

В [6] предложено полностью избавиться от различных персонажей. В разработанном прототипе жанра MOBA игроки получают персонажей с одинаковыми характеристиками и самостоятельно выбирают в начале игровой партии нужные способности персонажей из общего перечня. Таким образом, проблема создания персонажей заменяется проблемой разработки способностей.

Помимо самостоятельного выбора способностей, игроки получают персонализированный опыт благодаря системе адаптации персонажа (CAS – Character Adaptation System). В ходе игровой сессии CAS непрерывно отслеживает использование способностей и автоматически повышает соответствующие характеристики персонажа (например, интеллект при попадании магической атакой).

Такой подход действительно обеспечивает адаптацию игровых персонажей, однако адаптивность обеспечивается только внутри одной игровой сессии, что, с одной стороны, повышает гибкость в конкретном матче, с другой – не позволяет адаптировать персонажа под общий стиль игры конкретного человека. Как отмечено авторами [6], малое количество игроков в прототипе не позволяет сделать вывод о том, насколько предложенная система облегчает или усложняет разработку способностей и правку баланса по сравнению с существующими практиками создания персонажей.

На способностях персонажей также фокусируется работа [7], где предложено использовать эволюционные алгоритмы для создания новых способностей. Такая система была реализована в игре с боевой системой, аналогичной JRPG, – две партии из 4 персонажей, по 4 способности у каждого, противостоят друг другу в пошаговом режиме. Способности имеют два свойства – «польза» и «цена», которые применяются к цели и увеличивают/уменьшают какую-либо характери-

стику соответственно. Эти значения для каждой характеристики составляют генотип. Значение функции приспособленности зависит от количества использований способности игроками. Перед каждым боем игроки могут заменить свои способности на сгенерированные, а невыбранные способности уходят в архив, который используется для следующих генераций.

По результатам тестирования этой системы было обнаружено появление более разнообразных стратегий и сборок персонажей, чем можно было ожидать от классических архетипов RPG. Однако система генерации способностей была ограничена рамками одной игровой сессии 16 боёв в среднем и не сохраняла результаты в новой игре. Кроме того, было проведено всего 6 игровых сессий. Как отмечает автор указанной работы, создание постоянного сервера и общего пула способностей для генетического алгоритма, а также добавление к способностям новых механик должны привести к генерации способностей, более подходящих различным стилям игроков.

В [8] для решения проблемы малого количества игровых сессий для оценки сгенерированного контента использованы компьютерные симуляции. В разработанном файтинге Sol также были использованы эволюционные алгоритмы для создания персонажей. Схема работы этой системы представлена на рис. 1.

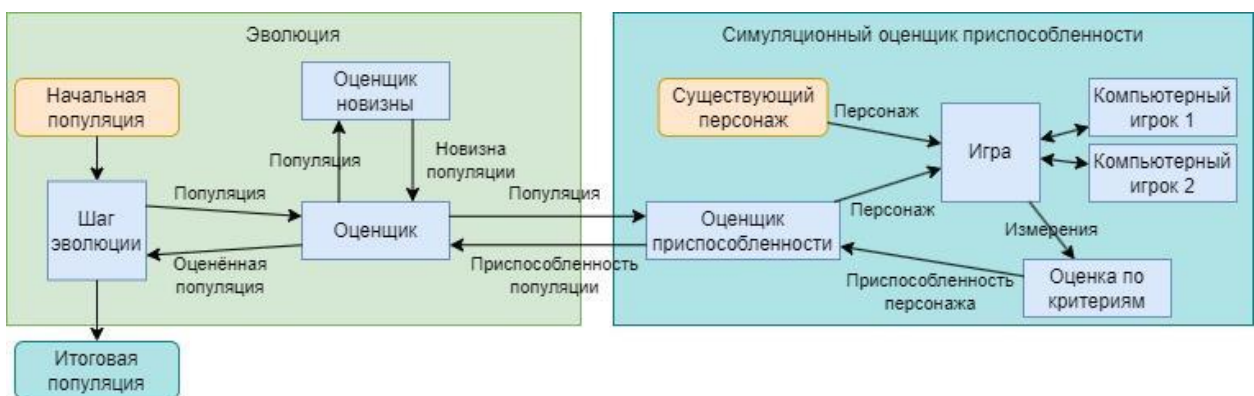


Рис. 1. Архитектура генератора [8]

Генотип составляют параметры персонажа и три способности. Создаваемые персонажи отправляются в симуляцию, где после серии игр оцениваются по 5 параметрам:

- сбалансированность матчей – процент побед персонажа;
- длительность матчей – средняя длительность матчей с персонажем;
- покрытие карты – среднее значение количества посещённых ячеек игровой карты (из 1400);
- сбалансированность персонажей – определяется средним соотношением количества использования способности персонажа к использованию всех трёх способностей того же персонажа за игру;
- смена лидерства – среднее число раз за игру, когда сменялся персонаж с лучшим состоянием.

Для реализации играющих агентов использовался подход, основанный на правилах. Для агентов было определено несколько правил: избегать пропасти, двигаться случайно, отступать от противника, подходить к противнику, избегать хитбоксов (от англ. hitbox – область поражения) – коллайдеров<sup>5</sup> противника или снарядов, наносящих урон, или атаковать способностью (в т. ч. случайно).

Для исключения генерации похожих персонажей, помимо результатов симуляции в эволюции, также учитывалось евклидово расстояние до нескольких из ближайших соседей (обычно до 5) в пространстве генотипов персонажей.

В заключительной стадии исследования 6 сгенерированных и 6 разработанных человеком персонажей тестировались реальными игроками в игре против компьютера. После этого игрокам было предложено сравнить персонажей между собой по тому, насколько интересно им было играть за них. В результате сгенерированные персонажи получили несколько более высокие оценки, что подтверждает начальную гипотезу о возможности генерации «интересных» персонажей при помощи формальной оценки.

---

<sup>5</sup> Коллайдер – область игрового пространства, пересечение объектов с которой фиксируется игрой как отдельное событие и вызывает соответствующий обработчик.

Генетические алгоритмы являются широко распространённым выбором для генерации игровых персонажей. В [9] показано их преимущество для генерации правдоподобных персонажей для футбольных менеджеров по сравнению со случайной генерацией и случайной генерацией с коррекциями, используемыми в рассматриваемом симуляторе.

Одной из существенных проблем персонализации чего-либо под конкретного игрока является отсутствие его индивидуальной модели. Ситуация усугубляется тем, что часто у разработчиков есть ограниченный набор реплеев игрока, недостаточный для обучения моделей. Решению этой проблемы на примере игры в реверси посвящена работа [10]. Алгоритм авторов проводит обучение модели в два этапа. С начала модель учится играть в игру в целом и не видит реплеев конкретного игрока. Затем подготовленная модель подстраивается на оставшихся записях (рис. 2).

Данная модель сравнивалась с нейронной сетью и случайным лесом. В ходе обучения модели схожесть с игроком не проверялась. Для проверки моделей использовались записи 6 ботов для игры реверси, обученных на разных алгоритмах.



Рис. 2. Схема имитационного моделирования отдельного игрока [10]

В качестве меры схожести бралась сумма совпадающих пар состояние – действие на тестовой выборке в 800 записей для каждого моделируемого бота. На обучающей выборке в 300 записей для каждого бота авторам удалось добиться от 33% до 59% схожести модели, что превышает результаты альтернативных моделей.

Несмотря на то что названный результат далёк от близкого копирования стратегии игрока, стоит отметить, что он достигнут на относительно малой обучающей выборке. Другим преимуществом этой модели является отсутствие эффекта

переобучения по сравнению с нейронными сетями при большом количестве итераций дообучения. В целом исследователи пришли к выводу, что универсальная обученная модель является хорошей точкой для создания модели игрока, т. к. стратегии разных игроков имеют общую составляющую.

Аналогично авторы [11] использовали предсказания следующих действий, но с учётом метрик Precision и Recall: сравнивали методы случайного леса и рекуррентных нейронных сетей с LSTM для имитации игроков в Starcraft. В качестве источника данных брались состояния игры каждые 3 секунды в реплеях 562 и 762 игр первого и второго игроков.

Результат исследования показал преимущество рекуррентных нейронных сетей для непустых действий. При этом показатели точности (Accuracy) у RNN находились в диапазоне 45–78%, что авторы называют недостаточным для удовлетворительной имитации действий игрока. С другой стороны, в 95% случаев действие игрока оказывалось в топ-3 предсказанных рекуррентной нейронной сетью, и, как предполагают авторы, при доработке выбора ответа модель сможет серьёзно улучшить свой результат.

### **АРХИТЕКТУРА ПРЕДЛАГАЕМОЙ СИСТЕМЫ**

Несмотря на существующий прогресс в смежных вопросах, на данный момент отсутствует система, которая позволила бы генерировать интересные сбалансированные персонажи, подходящие для конкретного игрока в рамках не одной игровой сессии, а метаигры<sup>6</sup>.

В рамках разработки такой системы должны быть реализованы три подсистемы:

- подсистема, моделирующая действия игрока на основе записей его игрового процесса (реплеев);
- подсистема для автоматической генерации персонажей, подходящих игровому стилю игрока;

---

<sup>6</sup> Метаигра – игровая и околоигровая деятельность игроков за пределами основного игрового цикла. Например, для игр, которые чётко разделены на партии, матчи и т. п., метаигра может включать анализ предыдущего опыта, прокачку персонажей, выполнение кумулятивных достижений, в которых считается прогресс за несколько партий, и т. д.

– подсистема автоматической валидации баланса и эффективности сгенерированного контента с помощью симуляций поведения игроков.

Для описания игрового стиля игрока предлагаются следующие элементы:

- время использования способностей;
- паттерны расположения персонажей на карте;
- стиль управления ресурсами;
- классовые предпочтения персонажей;
- агрессивный и оборонительный стиль игры;
- частота выбора персонажей;
- процент выигрышей при использовании персонажей различных классов.

Для анализа игрового стиля могут быть использованы реплеи, которые представляют собой временные ряды с действиями игрока.

Для имитации действий игрока в [12] предложено использовать параметрические деревья принятия решений. Такой подход представляется недостаточно универсальным. В рамках будущего исследования планируется использовать RL-агентов, при обучении которых с большим весом использовались бы партии игрока. Ранее в [13] RL-агенты применялись для тестирования уровней для игры жанра tower defense. Конкретная реализация будет выбираться в зависимости от результатов тестов, однако существуют исследования (например, [14]), которые показывают превосходство архитектуры Decision Transforming в задачах долгосрочного планирования перед другими подходами к обучению с подкреплением. С учётом того что в играх для достижения цели часто надо отклониться от локального оптимума, данный подход видится перспективным.

Состояния персонажей и других игровых сущностей при этом, так же как и используемые механики, могут быть описаны в рамках единого подхода, такого как FAST-GM [15]. В соответствии с ним представления сущностей содержат текущие значения атрибутов, ограничения на значения атрибутов, метку времени и вероятностные характеристики перехода в другие состояния. Помимо этого FAST-GM позволяет описывать также взаимодействия между различными сущностями, в том числе способности персонажей.



Для агента пространство состояний будет состоять из состояния персонажей, состояния способностей и сущностей на игровом поле, пространство действий, соответственно, – из выбора требуемого персонажа, указания его положения, использования способности и выбора цели персонажа.

Для генерации персонажей будет взят набор инструментов, его описывающих: комбинирование примитивных техник для создания способностей, подстройка числовых значений параметров персонажа и способностей, взаимное усиление способностей через дополнительные статусы. Кроме того, в виде ограничения в описании сущности по методологии FAST-GM будет использован набор балансовых ограничителей: стоимости, урона/исцеления в единицу времени, продолжительности эффектов контроля.

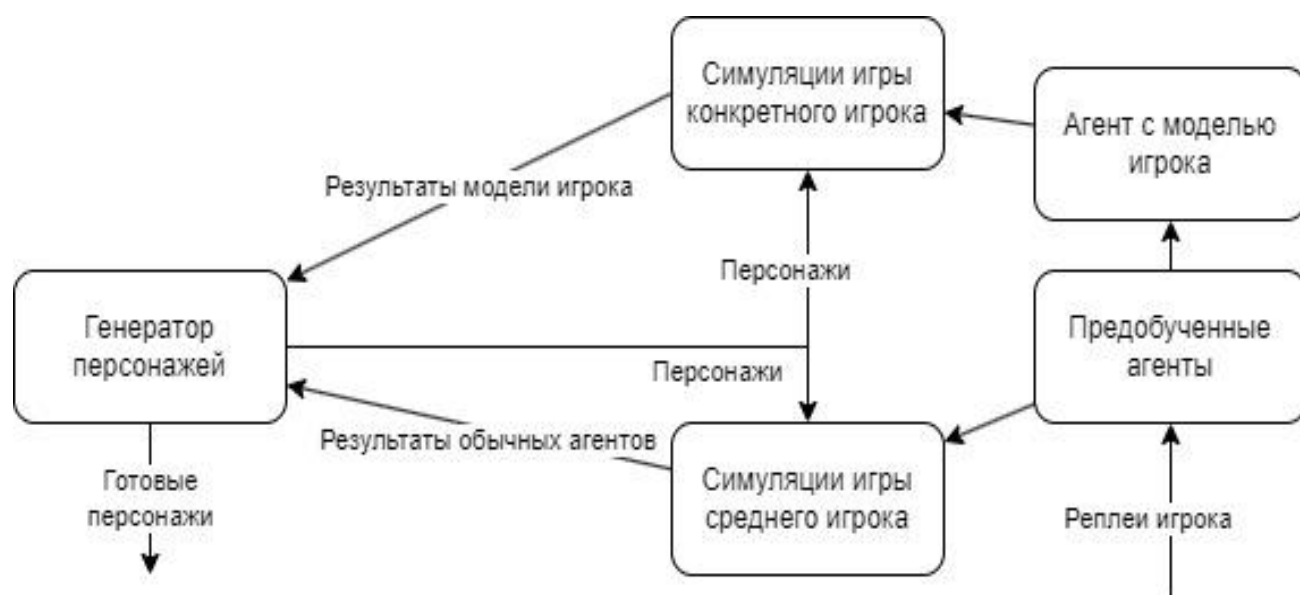


Рис. 3. Схема работы предлагаемой системы

После генерации следуют тестирование и оценка качества получившихся персонажей путём многократной игры обученных RL-агентов (рис. 3). Предполагается, что агенты, обученные на реплеях конкретного игрока, должны показывать результат не хуже, чем обычные агенты, чтобы игрок мог успешно использовать их в дальнейшем.

Помимо стандартных метрик, таких как процент побед с использованием персонажа, также будут реализованы метрики оценки схожести персонажей

со стилем игрока. В [12] для вычисления метрики схожести использованы матрица подобия для ресурсов игрока и её изменения с течением времени. В будущей работе предполагается учитывать соотношение урона, исцеления, наложения эффектов контроля и их изменение с течением времени.

## ЗАКЛЮЧЕНИЕ

Рассмотрено текущее состояние исследований, связанных с процессами генерации персонажей и моделирования игроков для автоматического анализа контента.

Предложена система, учитывающая существующие результаты. Предполагается, что разрабатываемая система докажет возможность реализации автоматической генерации интересных адаптированных персонажей и будет обладать следующими параметрами:

- моделирование игроков с более чем 75% точностью для выбора персонажей, процента выигрыша персонажей, паттернов расположения;
- устойчивое определение игрового стиля игрока не более чем на 100 реплеях;
- устойчивая генерация персонажей, которые проходят оценку качества.

Успешная реализация предлагаемой системы откроет новые возможности для оптимизации процессов разработки и сопровождения игр-сервисов, значительно сократив время между концептуализацией и внедрением персонажей.

## ЛУДОГРАФИЯ

Riot Games. (2009) *League of Legends*. [Видеоигра] [Загрузка] [Windows, MacOS]. Riot Games.

Hypergryph. (2020) [2019] *Arknights*. [Видеоигра] [Загрузка] [Android, iOS]. Yostar Games.

Ubisoft Montreal. (2015) *Tom Clancy's Rainbow Six Siege*. [Видеоигра] [Загрузка] [Windows, Xbox One, Xbox Series, Playstation 4, Playstation 5]. Ubisoft Entertainment.

Skjærseth E. и Vinje H. (2020) [2016] *Sol*. [Видеоигра] [Загрузка] [Windows]

Blizzard Entertainment. (1998) *Starcraft: Brood War*. [Видеоигра] [CD, Загрузка] [Windows, MacOS]. Blizzard Entertainment.

---

## СПИСОК ЛИТЕРАТУРЫ

1. *Rousseau J.* Report: 95% of studios are working on or aim to release a live service game // GamesIndustry.biz. 2024. URL: <https://www.gamesindustry.biz/report-95-of-studios-are-working-on-or-aim-to-release-a-live-service-game>.
2. 2023 Game Development Report / Rendered Venture Capital, Griffin Gaming Partners. 2024. URL: <https://griffingp.com/2023-game-development-research/>.
3. *Canheti C, Andalo F., Vieira M.L.H.* Case Study: Game Character Creation Process // Advances in Human Factors in Wearable Technologies and Game Design: Advances in Intelligent Systems and Computing / T.Z. Ahram (Ed.). Cham: Springer International Publishing, 2019. Vol. 795. P. 343–354. [https://doi.org/10.1007/978-3-319-94619-1\\_34](https://doi.org/10.1007/978-3-319-94619-1_34).
4. *Шубин А.В., Кугуракова В.В.* Проектирование инструментария для создания игрового процесса через систематизацию игровых механик // Электронные библиотеки. 2024. Т. 27. № 5. С. 774–795. <https://doi.org/10.26907/1562-5419-2024-27-5-774-795>.
5. *Browne C, Maire F.* Evolutionary Game Design // IEEE Transactions on Computational Intelligence and AI in Games. 2010. Vol. 2, No. 1. P. 1–16. <https://doi.org/10.1109/TCIAIG.2010.2041928>.
6. *Cannizzo A., Ramírez E.* Towards Procedural Map and Character Generation for the MOBA Game Genre // Ingeniería y Ciencia. 2015. Vol. 11, No. 22. P. 95–119. <https://doi.org/10.17230/ingciencia.11.22.5>.
7. *Pantaleev A.* In Search of Patterns: Disrupting RPG Classes through Procedural Content Generation // Proceedings of the third workshop on Procedural Content Generation in Games FDG'12: International Conference on the Foundations of Digital Games. 2012. P. 1–5. <https://doi.org/10.1145/2538528.2538532>.
8. *Skjærseth E.H., Vinje H.* Evolutionary algorithms for generating interesting fighting game character mechanics: Master thesis. Trondheim: Norwegian University of Science and Technology, 2020. URL: <https://hdl.handle.net/11250/2777860>.
9. *No H.-S., Rhee D.-W.* A Study on The Game Character Creation Using Genetic Algorithm in Football Simulation Games // Journal of Korea Game Society. 2017. Vol. 17, No. 6. P. 129–137. <https://doi.org/10.7583/JKGS.2017.17.6.129>.

10. *Pan C.-F, Min X.-Y. et al.* Behavior imitation of individual board game players // *Applied Intelligence*. 2023. Vol. 53, No. 10. P. 11571–11585.  
<https://doi.org/10.1007/s10489-022-04050-w>.
11. *Partla N. et al.* Player Imitation for Build Actions in a Real-Time Strategy Game // *AIIDE workshop on Artificial Intelligence for Strategy Games*. 2019.
12. *Kozik A., Machalewski T. et al.* Mimicking Playstyle by Adapting Parameterized Behavior Trees in RTS Games / *arXiv:2111.12144 [cs]*. 2021.  
URL: <http://arxiv.org/abs/2111.12144>.
13. *Liu S., Chaoran L. et al.* Automatic generation of tower defense levels using PCG // *Proceedings of the 14th International Conference on the Foundations of Digital Games FDG'19: The Fourteenth International Conference on the Foundations of Digital Games*. San Luis Obispo California USA: ACM, 2019. P. 1–9.  
<https://doi.org/10.1145/3337722.3337723>.
14. *Chen L., Lu K et al.* Decision Transformer: Reinforcement Learning via Sequence Modeling. *Decision Transformer* / *arXiv:2106.01345 [cs]*. 2021.  
URL: <http://arxiv.org/abs/2106.01345>.
15. *Кузурасова В.В.* Формальный подход к пространственно-временному моделированию игровых систем // *Ученые записки Казанского университета. Серия Физико-математические науки*. 2024. Т. 166, №. 4. С. 532–554.  
<https://doi.org/10.26907/2541-7746.2024.4.532-554>.

## GENERATIVE METHODS FOR CREATING ADAPTIVE PLAYABLE CHARACTERS IN SERVICE GAMES

T. R. Arslanov<sup>[0009-0002-1110-8503]</sup>

Kazan Federal University, Kazan, 420008, Russia

timars-mail@yandex.ru

### **Abstract**

With the growing popularity of game services that require constant content updates to retain players, automating the generation of adaptive playable characters has become an urgent task. This article examines existing approaches to character generation, including evolutionary algorithms, and in-session adaptation systems. Current solutions are limited by their inability to provide sufficient long-term adaptation to individual player styles and their reliance on manual design.

To address these limitations, we propose a three-component system that integrates: player action modeling based on gameplay replays using reinforcement learning (RL) agents, character generation through combinatorial mechanics and parameter balancing, automatic validation via simulations to assess balance and alignment with a player's individual style.

This work synthesizes contemporary research, highlighting the potential of generative methods to reduce development costs for game services. The results could accelerate prototyping and enhance the long-term viability of live-service projects.

**Keywords:** *service games, game design, game characters, video game, procedural content generation.*

### **LUDOGRAPHY**

Riot Games. (2009). *League of Legends* [Windows, MacOS]. Riot Games.  
<https://www.leagueoflegends.com/>

Hypergryph. (2020). *Arknights* (Version Global) [Android, iOS]. Yostar.  
<https://arknights.global>

Ubisoft Montreal. (2015). *Tom Clancy's Rainbow Six Siege* [Windows, Xbox One, Xbox Series, Playstation 4, Playstation 5]. Ubisoft Entertainment.  
<https://www.ubisoft.com/en-us/game/rainbow-six/siege>

Skjærseth, E. H., & Vinje, H. (2020). *Sol* [Windows]. [https://github.com/sol-ai/sol-ai\\_project](https://github.com/sol-ai/sol-ai_project)

Blizzard Entertainment, & Saffire. (1998). *Starcraft: Brood War* [Windows, MacOS]. Blizzard Entertainment.

## REFERENCES

1. *Rousseau J.* Report: 95% of studios are working on or aim to release a live service game // GamesIndustry.biz. 2024. URL: <https://www.gamesindustry.biz/report-95-of-studios-are-working-on-or-aim-to-release-a-live-service-game>.
2. 2023 Game Development Report / Rendered Venture Capital, Griffin Gaming Partners. 2024. URL: <https://griffingp.com/2023-game-development-research/>.
3. *Canheti C, Andalo F., Vieira M.L.H.* Case Study: Game Character Creation Process // Advances in Human Factors in Wearable Technologies and Game Design: Advances in Intelligent Systems and Computing / T.Z. Ahram (Ed.). Cham: Springer International Publishing, 2019. Vol. 795. P. 343–354. [https://doi.org/10.1007/978-3-319-94619-1\\_34](https://doi.org/10.1007/978-3-319-94619-1_34).
4. *Shubin A.V., Kugurakova V.V.* Designing a tool for creating gameplay through the systematization of game mechanics // Russian Digital Libraries Journal. Vol. 27, No. 5. P. 774–795. <https://doi.org/10.26907/1562-5419-2024-27-5-774-795>. <https://doi.org/10.26907/1562-5419-2024-27-5-774-795>.
5. *Browne C, Maire F.* Evolutionary Game Design // IEEE Transactions on Computational Intelligence and AI in Games. 2010. Vol. 2, No. 1. P. 1–16. <https://doi.org/10.1109/TCIAIG.2010.2041928>.
6. *Cannizzo A., Ramírez E.* Towards Procedural Map and Character Generation for the MOBA Game Genre // Ingeniería y Ciencia. 2015. Vol. 11, No. 22. P. 95–119. <https://doi.org/10.17230/ingciencia.11.22.5>.
7. *Pantaleev A.* In Search of Patterns: Disrupting RPG Classes through Procedural Content Generation // Proceedings of the third workshop on Procedural Content Generation in Games FDG'12: International Conference on the Foundations of Digital Games. 2012. P. 1–5. <https://doi.org/10.1145/2538528.2538532>.
8. *Skjærseth E.H., Vinje H.* Evolutionary algorithms for generating interesting fighting game character mechanics: Master thesis. Trondheim: Norwegian University of Science and Technology, 2020. URL: <https://hdl.handle.net/11250/2777860>.

9. *No H.-S., Rhee D.-W.* A Study on The Game Character Creation Using Genetic Algorithm in Football Simulation Games // *Journal of Korea Game Society*. 2017. Vol. 17, No. 6. P. 129–137. <https://doi.org/10.7583/JKGS.2017.17.6.129>.
  10. *Pan C.-F, Min X.-Y. et al.* Behavior imitation of individual board game players // *Applied Intelligence*. 2023. Vol. 53, No. 10. P. 11571–11585. <https://doi.org/10.1007/s10489-022-04050-w>.
  11. *Partla N. et al.* Player Imitation for Build Actions in a Real-Time Strategy Game // *AIIDE workshop on Artificial Intelligence for Strategy Games*. 2019.
  12. *Kozik A., Machalewski T. et al.* Mimicking Playstyle by Adapting Parameterized Behavior Trees in RTS Games / *arXiv:2111.12144 [cs]*. 2021. URL: <http://arxiv.org/abs/2111.12144>.
  13. *Liu S., Chaoran L. et al.* Automatic generation of tower defense levels using PCG // *Proceedings of the 14th International Conference on the Foundations of Digital Games FDG'19: The Fourteenth International Conference on the Foundations of Digital Games*. San Luis Obispo California USA: ACM, 2019. P. 1–9. <https://doi.org/10.1145/3337722.3337723>.
  14. *Chen L., Lu K et al.* Decision Transformer: Reinforcement Learning via Sequence Modeling. *Decision Transformer* / *arXiv:2106.01345 [cs]*. 2021. URL: <http://arxiv.org/abs/2106.01345>.
  15. *Kugurakova V.V.* A formal approach to spatio-temporal modeling of game systems // *Uchenye Zapiski Kazanskogo Universiteta. Seriya Fiziko-Matematicheskie Nauki*. 2024. Vol. 166, No. 4. P. 532–554. <https://doi.org/10.26907/2541-7746.2024.4.532-554>.
-

## СВЕДЕНИЯ ОБ АВТОРЕ



***АРСЛАНОВ Тимур Рузелевич*** – аспирант Института информационных технологий и интеллектуальных систем Казанского федерального университета. Направление исследований: фундаментальные аспекты создания видеоигр с высокой реиграбельностью.

***Timur Ruzelevich ARSLANOV*** – Postgraduate student at the Institute of Information Technologies and Intelligent Systems of Kazan Federal University. Research area: fundamental aspects of creating video games with high replayability.

email: timars-mail@yandex.ru

ORCID: 0009-0002-1110-8503

*Материал поступил в редакцию 25 января 2025 года*