ИПМ им.М.В.Келдыша РАН

Абрау-2022 • Труды конференции

Г.Ф. Сахибгареева, В.В. Кугуракова, Э.С. Большаков

Генерация и балансирование игровых механик видеоигр

Рекомендуемая форма библиографической ссылки

Сахибгареева Г.Ф., Кугуракова В.В., Большаков Э.С. Генерация и балансирование игровых механик видеоигр // Научный сервис в сети Интернет: труды XXIV Всероссийской научной конференции (19-22 сентября 2022 г., онлайн). — М.: ИПМ им. М.В.Келдыша, 2022. — С. 455-485.
https://doi.org/10.20948/abrau-2022-6
https://keldysh.ru/abrau/2022/theses/6.pdf

Видеозапись выступления

Презентация к докладу

Генерация и балансирование игровых механик видеоигр

Г.Ф. Сахибгареева[1], В.В. Кугуракова[1], Э.С. Большаков[1]

[1] Институт информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета

Аннотация. Статья является продолжением развития идеи о разработке генератора игровых прототипов из текста на естественном языке. Тема игрового баланса, которая упоминалась в ранних работах авторов, полноценно раскрывается с теоретической и практической точки зрения. Для понимания природы данного вопроса, были проанализированы определения и мнения экспертов. В результате было сформировано всестороннее понимание проблем, а также перечислены те вызовы, которые есть в данном направлении. Целью исследования является автоматизация рутины игровых дизайнеров на этапе прототипирования. Для этого решается задача генерации игрового баланса из текстовых документов, поэтому рассмотрен ряд научных работ, предлагающих алгоритмы и подходы для оптимизации и автоматизации тестирования и балансирования компьютерных игр. Подробно представлен функционал редактора динамического баланса Machinations, а также проиллюстрирован принцип его работы. Для доказательства уместности его подходов в общем конвейере работы генератора прототипов был проведен ряд экспериментов. Они доказывают, что работа с динамическими диаграммами эффективна и экономит важные ресурсы команды разработки. Кроме того была решена частная задача формализации и визуализации связи между геймплеем и сюжетом, что обосновано контекстной зависимостью игрового баланса. В заключении приведены планы дальнейшего развития полноценного инструмента для игровых сценаристов и дизайнеров. В качестве вывода утверждается тезис о том, что автоматическое корректирование игрового баланса возможно также, как и его генерация на основе текста.

Ключевые слова: игровой баланс, геймплей, игровые механики, автоматическое балансирование игровых механик, игровой дизайн, игровая сценаристика, сценарный прототип, генерация сценарных прототипов, Machinations.

Video Game's Mechanics Generation and Balancing

G.F. Sahibgareeva[1], V.V. Kugurakova[1], E.S. Bolshakov[2]

[1] *Institute of Information Technology and Intelligent Systems, Kazan Federal University*

Abstract. This article is a continuation of the idea about developing a gaming prototype generator from natural language text. Game balance theme, which mentioned in previous authors' papers, fully opens up from theoretical and practice standpoints. Many expert opinions and definitions were analyzed to understand the identity of that problem. As a result, a full understanding of the problem was formed and challenges were listed, which exists in this direction. The purpose of the research is automation of game-designers routine in the prototyping stage. For this, the problem of generation of game balance from text documents is solved, so a number of scientific papers are considered, which offer algorithms, optimizing and automating approaches and computer games balance. The functionality of the dynamic balance editor Machinations is presented in detail, and the principle of operations is illustrated. To check the availability of approaches in the overall collection of the prototype generator work, a number of experiments were provided. They prove effective work with exhaustive diagrams and saves important development team resources. In addition, a particular problem of formalization and visualizing connection between gameplay and plot was solved, which is justified by the context dependence of the game balance. In conclusion, plans for the further development of a full-featured tool for the game scriptwriters and designers are given. As a conclusion, we state that automatic correction of the game balance is possible as well as its generation based on text.

Keywords: game balance, gameplay, game mechanics, game mechanics automatic balancing, game design, game scenario writing, scenario prototype, scenario prototype generation, Machinations.

Введение

Процессы разработки игровых, а также интерактивных проектов, например, тренажеров и AR/VR-приложений, бывают не только творческими, но и рутинными. Это касается программирования, создания компьютерной графики, баланса игровых параметров, а также написания сценариев и различных дизайн-концептов. Причем направлений дизайна здесь множество: игровой и нарративный дизайн, дизайн уровней и окружения, дизайн интерфейсов и даже саунд-дизайн.

Ввиду того, что любой ИТ-проект так или иначе тяготеет к автоматизации повторяющегося или ручного труда, в сфере разработки игр также возникают решения, призванные оптимизировать ресурсы.

В серии статей авторов затронута тема разработки комплексного инструмента генерации игрового прототипа из текста на естественном языке. Ниже дана краткая суть проделанных этапов работ:

- извлечение информации о сущностях повествования и связях между ними из текста на естественном языке [1],
- визуализация сущностей повествования в контексте событий в форме раскадровки [2, 3, 4],
- настройка камеры и расстановка объектов в трехмерной сцене с помощью текстового запроса [5],
- обзор вопроса визуализации разветвленной структуры повествования [6],
- генератор игрового прототипа с разветвленным повествованием из текстового запроса [7, 8, 9].

До сих пор в работах упоминалась, но не раскрывалась тема игрового баланса – тонкого процесса по настройке правил игры, сложности, взаимосвязей и алгоритмов для достижения определенных целей: как сделать игру честной для всех игроков, как сделать игру сложной, но выигрышной и т.д.

Данная работа посвящается разбору современного представления разных авторов о балансировке и авторском видении того, как балансировка игры и, в частности, игровые механики должны быть встроены в сборку игрового прототипа, позволяющего прогнозировать успешность игры в целом. Дополнительно было скорректировано видение общей архитектуры генератора игрового прототипа, а также представлены три практические авторские реализации вопроса игрового баланса.

Основной вклад данной работы заключается в том, что в результате аналитической работы были представлены результаты экспериментов, а также планы дальнейшего развития инструмента, призванного решить проблемы оптимизации ресурсов при разработке игровых и интерактивных проектов.

В первой главе представлен обзор литературы на тему определения понятия "игрового баланса". Во второй главе рассматриваются разработанные инструменты балансировки игр. В третьей главе описан онлайн-редактора динамического математического баланса компьютерных игр Machinations, а также представлены результаты двух экспериментов, доказывающих его эффективность. В четвертой главе изложена суть авторского инструмента балансировки игр. В пятой главе представлено обновленное видение инструмента генерации сценарного прототипа и сформулированы перспективы развития. В заключении подведены итоги работы: такая задача, как математическое балансирование игрового прототипа, может быть автоматизирована, что качественно повысит артефакты этапа концептирования.

1. Определение игрового баланса

Хотя балансировка игр считается важной ключевой особенностью любой успешной игры [10], пока нет единого мнения о том, что на самом деле означает «балансировка игры».

В обзорной статье 2019 года на основе десятка определений проведен семантический анализ выражения «игровой баланс» [11]. Рассмотрим некоторые упомянутые в данной работе определения в хронологическом порядке, чтобы проследить развитие данного понятия.

Начнем с 2005 года. В книге ведущего дизайнера и сценариста компаний Ubisoft и Microsoft Studios Ричарда Роуза III «Game design. Theory and practice» поднимается тема игрового баланса в интервью с игровыми дизайнерами. Из контекста становится понятно, что речь идет о **математическом игровом балансе**. Однако, кроме этого, на примере дизайна уровней автор формулирует вывод о том, что интерактив, повествование, продвижение по уровню и другие аспекты игры связаны и должны поддерживать и не перебивать друг друга, быть в балансе [12].

В 2010 году сооснователь Global Game Jam и доцент интерактивных игр и медиа Рочестерского технологического института Ян Шрайбер в блоге «Game balance concepts» утверждает, что почти в каждой игре можно найти параметры, поддающиеся количественной оценке. Поэтому в этом источнике подробно и на примерах освещается тема математического игрового баланса [13]. Однако он добавляет, что любые данные имеют значение только **в контексте** и зависят от него. Помимо прочего, автор предлагает учитывать доступность информации для игроков, их способность обрабатывать информацию, их ожидания и даже внешние факторы.

Джонни Новак в книге «Game development essentials: an introduction» в 2012 году [14], как и Роллингс и Адамс в книге «Rollings and Ernest Adams on game design» в 2003 году [15], обозначают зависимость игрового баланса от **мастерства** игроков, но также разделяют **статический** и **динамический баланс**. Статический баланс включает *правила*, *числа*, *отношения* и *взаимодействия*. В свою очередь, динамический – иллюстрирует *изменения статического баланса* из-за игроков в режиме реального времени.

Интересно, что на Youtube-канале «Extra Credits» в видео 2012 года «Perfect imbalance – why unbalanced design creates balanced play» доказано, что **легкий игровой дисбаланс** мотивирует отдавать предпочтение новым стратегиям поведения, т.е. обучаться за счет ощущения легкого дискомфорта [16].

Основатель Ludeon Studios Тайнан Сильвестр в книге «Designing Games» в 2013 констатирует, что невозможно создать игру, в которой игроки с разным уровнем мастерства могли бы иметь равный шанс на успех [17]. Дизайнеры должны ориентироваться на целевую аудиторию,

чтобы создать то, с чем она справится. Таким образом, в основе игрового процесса – **честная игра**. При этом важно, чтобы в игре были доступны **разные стратегии**, чтобы игроки могли принимать взвешенное решение в пользу каждой из них. Однако любой баланс бесполезен, если он разрушает *повествование*, *плавность* и *темп игры*, *доступность* и *ясность правил*.

Профессор игрового дизайна Университета Карнеги-Меллона Джесси Шелл в удостоенной наград книге «Искусство игрового дизайна» в 2015 году аккумулирует большой объем знаний об игровом дизайне. В его работе можно найти противоречащие друг другу явления вызова и успеха, мастерства и удачи. Под балансированием здесь подразумевается ориентация на **целевую аудиторию** [18]. Формулировка расплывчатая и общая, но она ярко показывает, что игровой баланс в разных контекстах может подразумевать различные вызовы, что часто бывает в игровой индустрии, в которой сложно найти два проекта, которые разрабатывались бы одинаково.

Ряд научных статей, которые основываются на мнении практиков, посвящаются теоретическому и практическому изучению и расширению понятия игрового баланса. Так, в статье 2006 года [10] приходят к выводу, что игровой баланс **заинтересовывает** игроков. Помимо этого, в этой статье приводятся требования к динамическому балансу, который комфортен игрокам: игра **адаптируется** к начальному уровню навыков игроков, к развитию мастерства и остается правдоподобной. В более поздней работе 2018 года [19] адаптивность игрового баланса связывают с эмоциями и утверждается, что игра понравится игрокам, если вызовет **удовольствие**, а не скуку или разочарование.

По итогу анализа упомянутых определений, можно составить следующую картину: игра сбалансирована, если её сложность зависит от навыков целевой аудитории. Лучше всего, чтобы она адаптировалась под игроков со временем. Однако важно, чтобы игра бросала вызов, чтобы игроки испытывали чувство достижения. Нельзя рассматривать математический баланс в отрыве от остальной части игры, все связано и формирует контекст. Соответственно, и все части игры должны находиться в балансе. Игра должна приносить удовольствие, быть честной, справедливой, не вызывать излишнюю фрустрацию и скуку. Игра должна предоставлять разные стратегии, а не призывать использовать привычные тактики.

Справедливо упомянуть, что игровой баланс – это также инструмент дизайна, и если разработчики преднамеренно хотят создать состояние дискомфорта, то можно, соответственно, чрезмерно снизить или повысить сложность игры.

В силу перечисленных выше тезисов, в данной работе будет разделяться **математический баланс**, который можно выразить в

параметрах и функциях, и **игровой баланс**, подразумевающий контекст и другие аспекты, с которыми связан математический баланс.

2. Инструменты игрового баланса

Тема игрового баланса неразрывно связана с тестированием игр. Обусловлено это отчасти тем, что математические характеристики, будучи зависимыми от алгоритмов и функций, раскрываются в полной мере лишь в контексте повествования и игрового процесса.

Поэтому работы, представленные далее, в разных пропорциях содержат в себе решение задач автоматического тестирования и балансирования.

Вопросы, которые встают перед авторами, можно сгруппировать в следующие категории:

- формализация игрового процесса с помощью систем функций и автоматов состояний, создание модели математического баланса,
- сбор статистических данных об игровых сессиях,
- обучение интеллектуальных агентов на данных,
- создание тестового окружения,
- анализ работы интеллектуальных агентов в текстовом окружении в режиме реального времени,
- создание алгоритма корректировки математического баланса,
- корректировка значений математического баланса по заданному алгоритму.

Работ, способных корректировать игровой баланс согласно контексту, не так много. Однако применение ИИ-технологии способно помочь в этой задаче.

Так, авторы одной из работ 2016 года прибегают к тому, чтобы формализовать игру в виде задачи [20]. Такой подход не обещает, что полученный алгоритм можно применить к другим играм или хотя бы к играм одного жанра.

В статье же 2022 года [21] утверждается, что большинство научных работ решает проблему эвристическими и обобщенными методами и оптимизируют данные о поведении игроков, создавая архетипичных нереалистичных агентов. Авторы противопоставляют этому подходу глубокое моделирование поведения игрока (DPBM[1]), основанное на создании у агентов индивидуальности, который имитирует реальных игроков. Для обучения авторы используют наборы данных игры Aion [22]. База состоит из множества единиц принятия решений 213 игроков периода в шесть месяцев.

[1] DPBM (сокр. Deep Player Behavior Modeling) – глубокое моделирование поведения игрока.

Кроме того, в работе приведен подробный анализ работ последних лет. Выделим некоторые интересные подходы и добавим некоторые неотмеченные работы:

- создание архетипов игроков благодаря генеративному моделированию и дальнейшее тестирование с помощью поиска по дереву методом Монте-Карло синтетическими ИИ-тестировщиками в играх Dungeon Crawl[2] [23];
- балансировка с помощью поиска по дереву методом Монте-Карло стохастических игр [24], а также игр жанра tower defense[3] [25];
- балансирование с помощью эволюционных алгоритмов ИИ аркадной игр Ms. Pac-Man и RTS[4] StarCraft [26];
- применение коэволюционного метода для поиска возможных игровых стратегий в игре “Захват флага” [27];
- предсказание сложности недетерминированных игр-головоломок «три в ряд» с помощью сверточных нейронных сетей [28];
- применение эволюционного алгоритма многокритериальной оптимизации для балансирования карточной игры Top Trumps [29];
- балансировка игры, основанная на моделировании взаимосвязи между игровой динамикой (балансом) и игровой механикой (параметрами). Оценка динамики игры основана на методах оператора Купмана [30];
- использование JSON для хранения и редактирования игрового баланса и создание генетического алгоритма для балансирования пошаговой стратегический игры ComPet [31],
- глубокое обучение и эволюционные вычисления для генерации сбалансированных уровней для шутера с заданной продолжительностью игры [32].

Проблемы в игровом балансе возникают из-за особенностей контекста – игра должна приносить удовольствие и не должна вызывать незапланированные негативные эмоции, а также фрустрацию. Для разработчиков стоит задача удержать игрока как можно дольше, и если причиной ухода будет дисбаланс, то это проблема, которую надо решать. Более того, данная тема, пусть и получила популярность в научных кругах недавно, но уже имеет ряд продуктивных работ, которые помогают составить образ о проблемах и направлениях их решения.

Исключительно числовое представление игрового баланса безусловно склоняется к высшей математике и комбинаторике [33, 34].

---

[2] Dungeon Crawl (с англ. – «ползание по подземелью») – жанр ролевых настольных и компьютерных игр, основной игровой процесс которых заключается в путешествии по подземельям.

[3] Tower defense (с англ. – «башенная защита») – жанр видеоигр об уничтожении противников с помощью боевых башен.

[4] RTS (англ. «real-time strategy») – жанр видеоигр, стратегия в реальном времени.

Авторы подобных работ находят данные, которые можно использовать для дальнейшей обработки и различного рода выравнивания. Однако те же работы доказывают, что нельзя манипулировать исключительно числами. Важны и смыслы, которые несут те или иные изменения в значениях, и контекст, который образуется в сюжете.

С другой стороны, к этой теме подходят разработчики IDN[5], для которых одним из ключевых направлений является генерация повествования. Для этих целей берутся массивы текстовых данных и на выходе получаются генераторы, способные взаимодействовать с компонентами визуализации, что порождает различные генерации цифрового визуального контента [35, 36, 37].

Еще одно направление работ, которое ведется в IDN, – это работа с моделями эмоций. Так был создан API, который собирает информацию о биосигналах игрока и балансирует события так, чтобы удовлетворить ожидания [38]. Модель находится в тестовой форме, но авторы надеются, что ее получится интегрировать в сложные игры. А пока это многообещающий проект, способный регулировать сложность игры, а значит, и игровой баланс согласно эмоциям игрока.

Стоит подчеркнуть, что зачастую работы ориентированы на жанр, т.к., например, система баланса для пошаговой RPG[6] может быть малопригодна для платформеров или RTS.

Тенденция, которая прослеживается в работах, это внедрение новейших возможностей ИИ в игровую разработку. Кроме того, видно тяготение к автоматизации рутины.

Однако на сегодняшний день отсутствует решение, призванное объединить математическую и графовую составляющую решения вопроса, а также семантический анализатор, дабы генерировать игровой баланс в связи с контекстом, а не в информационном вакууме.

Таким образом, учитывая обозначенную выше позицию авторов, мы представляем собственную интерпретацию игрового баланса в рамках работы над инструментом генерации игровых прототипов.

3. Machinations

Machinations – визуальный онлайн редактор динамического математического баланса игры [39]. Инструмент позволяет проектировать, балансировать и моделировать числовое представление игрового процесса благодаря диаграммам, которые показывают изменения системы в режиме реального времени.

---

[5] IDN (англ. «interactive digital narrative») – научное направление о цифровом интерактивном повествовании.

[6] RPG (англ. «role-playing game») – жанр видеоигр, прообразом которых являются настольные ролевые игры.

Хоть Machinations сфокусирован на работе над игровыми механиками вне контекста, он позволяет работать с балансом без программирования и на ранних стадиях разработки. Гибкие возможности имитации игрового процесса позволяют не заострять внимание на жанре, решая всевозможные задачи создания математического представления игры.

Полученные данные возможно экспортировать для дальнейшей работы над проектом.

Рассмотрим функционал инструмента.

Ресурсы – это значения внутриигровых переменных, участвующих в игре и циркулирующих в потоках между пулами.

Пул – это узлы, позволяющие накапливать или распространять ресурсы.

Источник – это узел, который генерирует бесконечное количество ресурсов и передает их в пулы. Сток – это узел, который уничтожает бесконечное количество попадающих в него ресурсов.

Для активации пула, источника или стока доступен выбор одного из режимов:

1. Автоматический режим – одновременный запуск работы вместе с другими пулами.
2. Интерактивный режим – имитирует взаимодействие с игроком (для активации необходим клик).
3. Режим инициирующего действия – запускается один раз вместе с запуском диаграммы.
4. Пассивный режим – запускается благодаря триггеру от других элементов.

Пропускная способность канала между пулами настраивается согласно условиям игры.

После сборки диаграммы пользователь может воспроизвести ее: шаг за шагом диаграмма будет отражать изменения игровой системы.

Иллюстрация работы Machinations

Чтобы показать принцип работы Machinations, рассмотрим конкретный пример, который был разработан в рамках текущей исследовательской работы [40].

За основу была взята игра The Elder Scroll V: Skyrim [41]. Длительность игры составляет около 300 часов, исследование и прокачка персонажа потенциально может длиться годы.

В игре было проанализировано поведение NPC[7] в локациях Винтерхолд, самой маленькой и незаселенной, и Виндхельм, самой большой и густо населенной.

[7] NPC (англ. Non-Player Character) – неигровой персонаж, им управляет не игрок, а ИИ-алгоритм.

Для сравнения эффективности работы с инструментом был проведен эксперимент, суть которого заключалась в том, чтобы сопоставить два способа работы: документирование требований к поведению NPC и сборка динамических диаграмм в Machinations.

Один из примеров диаграммы иллюстрирует поведение NPC, действия которого не зависят от времени суток (рис. 1).

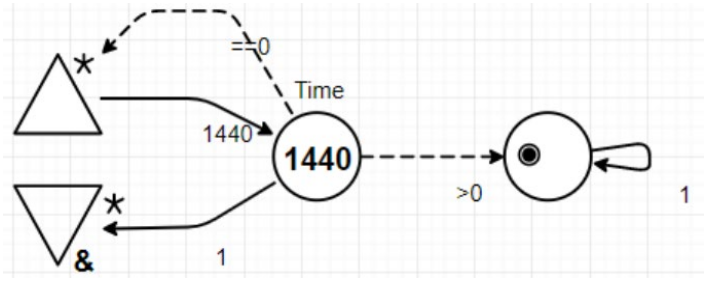

Рис. 1. Пример простой диаграммы баланса

Каждый день содержит 1440 минут, именно столько ресурса поступает из истока в пул в начале дня. Каждую минуту уничтожается одна единица ресурса времени. Ввиду того, что любое число больше нуля, единственный ресурс во втором пуле будет постоянно пребывать в одном состоянии. Так поведению NPC присваивается постоянное действие в течении дня.

В фрагменте сложного поведения NPC показано, как ресурс изо дня в день движется по одним и тем же шагам сценария всегда в один и тот же период времени (рис. 2). Так проиллюстрировано последовательное следование шагам NPC в течении дня.

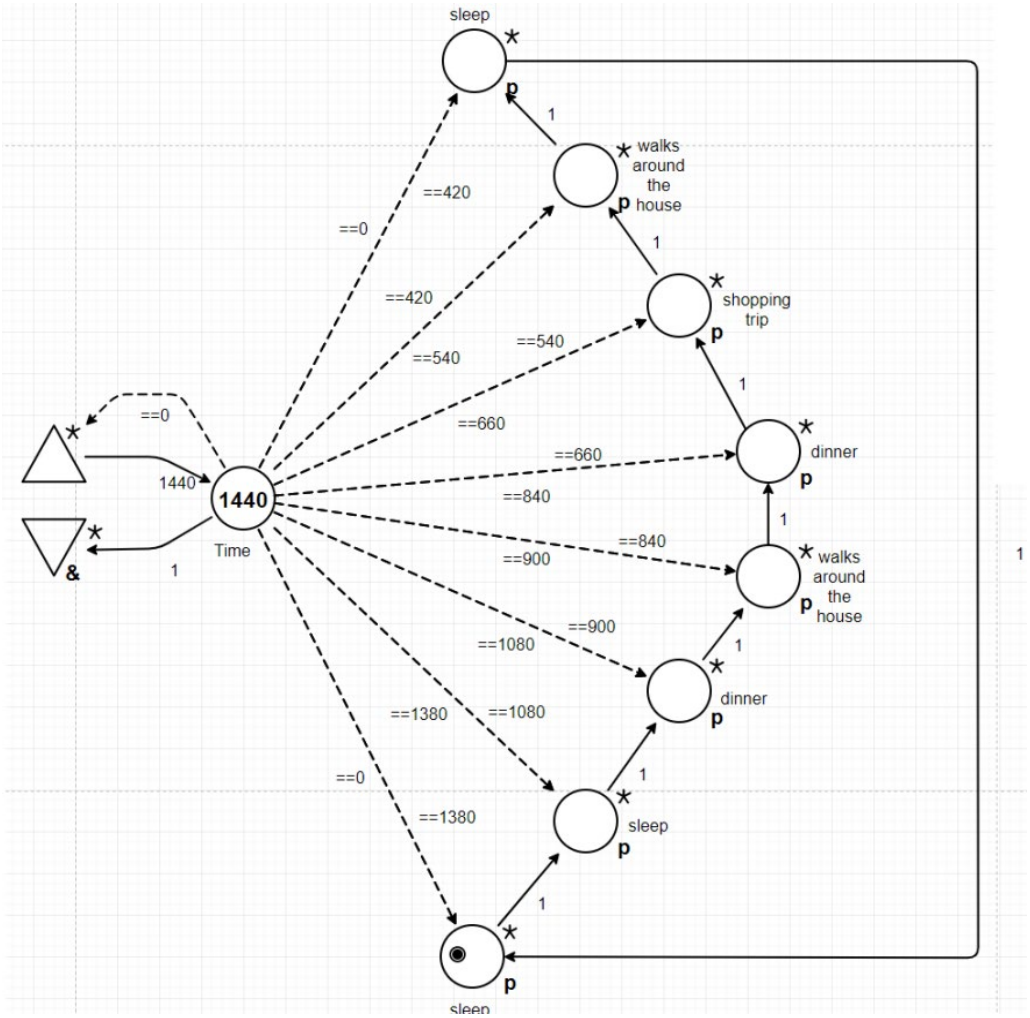

Рис. 2. Пример сложной диаграммы баланса

В результате сравнения времени, затраченного на обе части эксперимента, был получен вывод: суммарная работа над обеими локациями может сократиться как минимум на час, а также возможно избавление от необходимости заполнять объемные текстовые документы, т.к. было потрачено 62 страницы. Учитывая максимальную репрезентативность диаграммы, данный подход можно считать значительно более эффективным, чем ручной труд.

Machinations в автоматическом балансировании

Для работы над автоматическим балансированием полученной диаграммы был проведен другой эксперимент [42]. За основу была взята настольная ролевая игра Dungeons and Dragons [43], а именно механика кровотечения, которую персонаж получает в результате ранения во время боя.

Состав данной механики следующий: урон от кровотечения, шанс ослабить или остановить кровотечение, урон от атак противника и шанс попадания атаки противника.

Сбалансированность диаграммы оценивается по количеству ходов, которое можно нанести персонажу. Соответствующим образом

построенная диаграмма может быть доведена до желаемого состояния в результате многократной симуляции игрового процесса и корректировки.

В качестве иллюстрации работы диаграммы рассмотрим компонент, отвечающий за кровотечение (рис. 3).

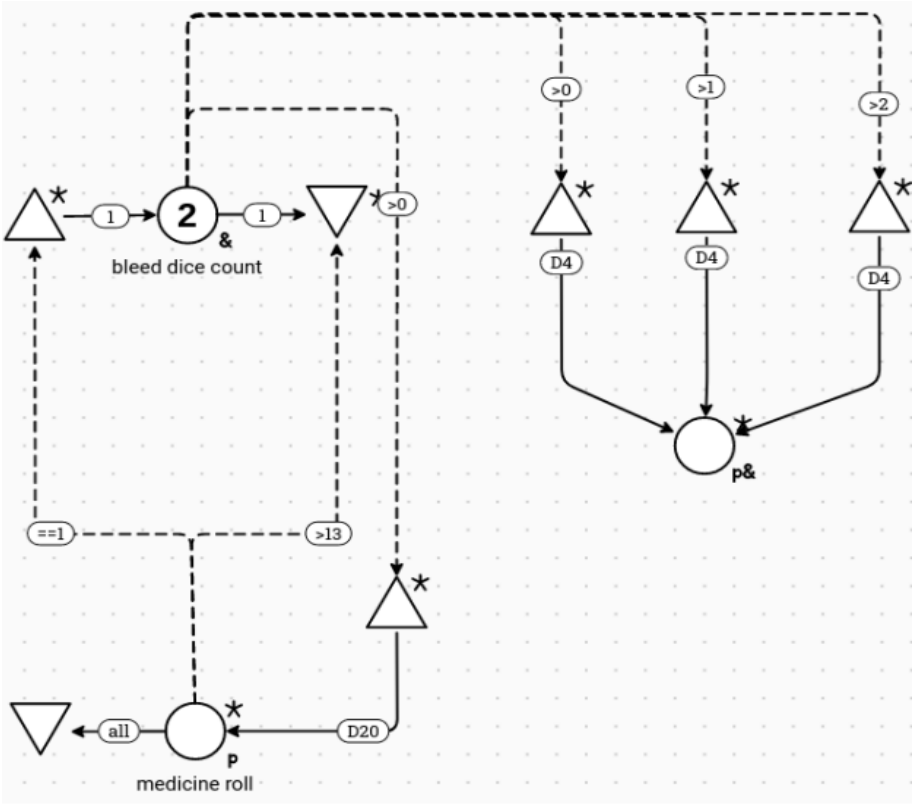

Рис. 3. Диаграмма баланса механики кровотечения

Пул «medicine roll» проверяет медицинские навыки персонажа с кровотечением. Если проверка покажет наличие таких навыков, то кровотечение уменьшится. В противном случае, кровотечение усилится.

Пул «bleed dice count» отвечает за то, сколько конкретно крови потеряет игрок.

Данный компонент отвечает за то, чтобы передавать информацию о том, сколько урона получает игрок от кровотечения.

В полученную итоговую диаграмму возможна отправка необходимых параметров, которые и будут составлять баланс игры (рис. 4). В эксперименте были использованы следующие значения: уровень здоровья персонажа – 44, ожидаемая продолжительность битвы – 6-10 ходов. В результате автоматической балансировки были получены следующие значения, которые удовлетворяют ожиданиям: уровень кровотечения – 2, количество граней кости урона – 12.

Чтобы подтвердить качество сгенерированных значений, проводится проверка. Так, в описанном эксперименте в результате имитации 40 боевых ситуаций средняя длительность боев составила семь.

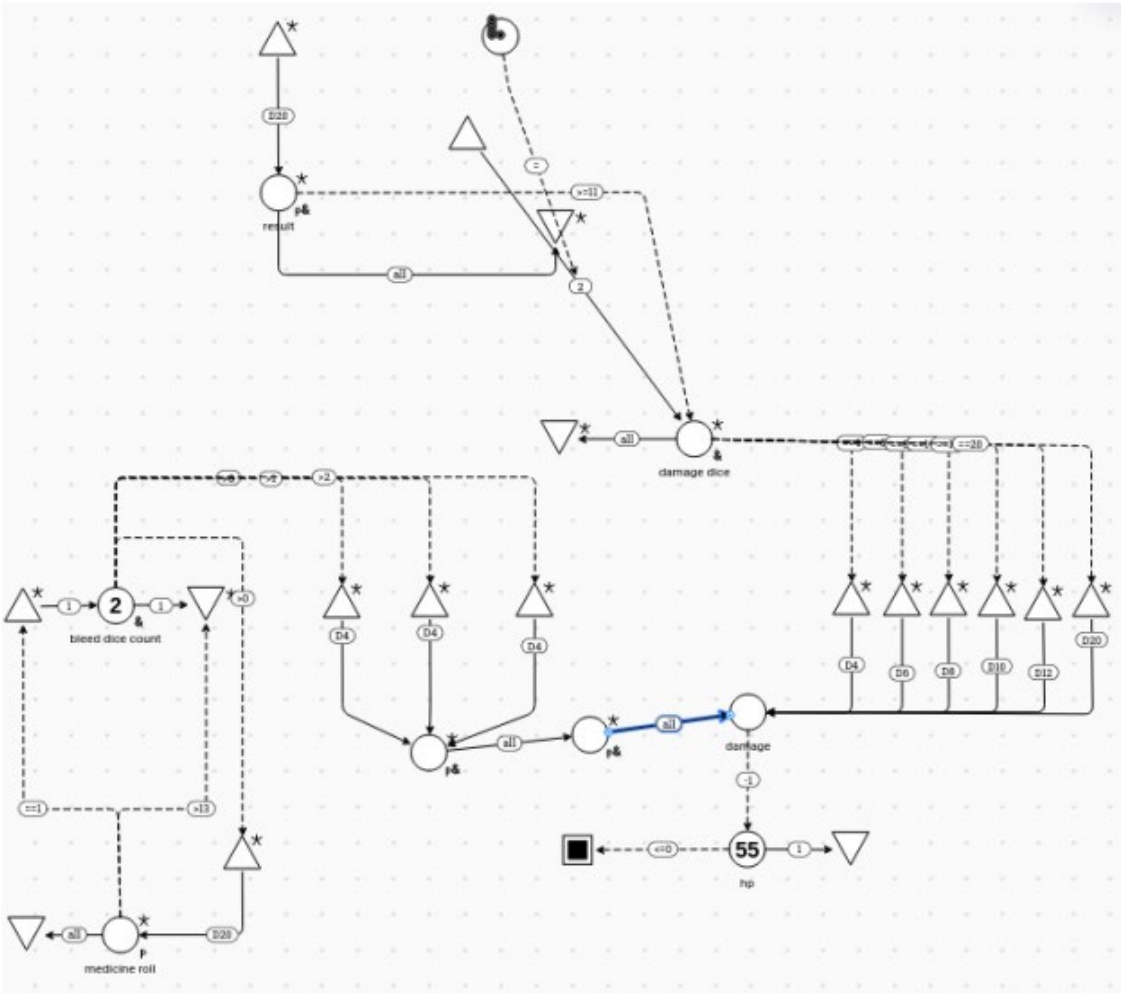

Рис. 4. Механика кровотечения в общей диаграмме

Эксперимент позволил прийти к выводу, что автоматический баланс значительно сокращает время на разработку и повышает качество результата. Так, автоматическая генерация и проверка баланса составила чуть более трех часов. В то время, как ручная проверка может затянуться на дни и недели.

На основании двух экспериментов напрашивается вывод о том, что динамические диаграммы математического баланса крайне эффективны и как сокращают время на разработку, так и не требуют глубоких технических знаний и навыков.

Благодаря оценке Machinations удалось прийти к видению того, как необходимо автоматизировать балансирование игровых механик в рамках работы генератора игровых прототипов на основе текста.

4. Корреляция сюжета и геймплея

На основе всех определений создается впечатление, что игровой баланс в классическом узком смысле – это то, что называют игровой экономикой или набором функций, параметрами которых являются

различные числовые значения, которые задают сложность игры, а также погружают игрока с различные состояния: нужда или достаток, слабость или сила, провал или удача.

Так или иначе, в каждой игре баланс в таком смысле будет свой, вне зависимости от того, как сильно желание определить его. В RPG[8] математическая модель – это основа, без которой невозможен игровой процесс. В это время в текстовых квестах баланса в классическом смысле может не быть, однако могут присутствовать такие функции, как определение вероятности события, что не сравнимо по простоте решения с тем, с чем вынуждены сталкиваться разработчики RPG-систем.

В широком же смысле слова, игровой баланс скорее про дизайн и про эмоции игрока и его личное состояние. Нарративному и игровому дизайнеру и даже дизайнеру уровней важно, чтобы те ситуации, к которым приводит игра в виду числовых значений или просто во взаимодействии игровых компонентов (правил, графики, UI, сюжета и других систем) должны вызывать нужные состояния. Это может быть качественный баланс компонентов, и тогда игроки будут чувствовать страх там, где надо, и воодушевление там, где это задано разработчиками. В то же время, способ заставить игрока испытывать фрустрацию – это сломать заданные правила баланса, не соблюдать их. Однако эта ситуация не исключение, а работа команды дизайнеров.

В рамках данной работы значение имеют все смыслы игрового баланса. Для создания опыта, соответствующего задумке дизайнеров, необходимо, чтобы формирующийся геймплей соответствовал заданным переменам в настроении игрока, а также не создавал конфликт с повествованием. Целостный игровой баланс – это следствие целостной игровой концепции.

В рамках генерации игрового прототипа важно, чтобы из текста была получена вся информация о том, какой математический баланс может существовать в геймплее, и, в то же время, важно, чтобы алгоритмы обработки “понимали”, какие эмоции рождает та или иная ситуация в сюжете, и является ли это балансом или ошибкой драматургии.

Решение задачи корректировки игрового баланса кажется достаточно сложной. В обсуждении подходов игрового балансирования интересна работа [44], где была предпринята попытка формализовать зависимость игровых механик от сюжета и был создан уникальный функционал.

[8] RPG (сокр. Role-playing Game) – жанр видеоигр, основанный на элементах игрового процесса традиционных настольных ролевых игр, где игрок управляет одним или несколькими персонажами, каждый из которых описан набором численных характеристик, списком способностей и умений.

Формализация и визуализация зависимости сюжета и геймплея

Система создана для игр с единственным протагонистом[9], соответствующим персонажу игрока. Фиксируются только те события сюжета, которые влияют на протагониста, а также на доступные ему объекты окружения.

Еще одно требование – наличие у протагониста цели, что означает, что история конечная. Ожидается, что сюжет развивается в связи с изменением какого-либо аспекта жизни персонажа, что и позволяет ему достичь цели.

Принцип работы заключается в том, что система следит за изменением состояний следующих **«объектов истории»**:

- развитие сюжетных линий,
- достижение протагонистом цели,
- динамика эмоциональных и физических состояний протагониста.

Последний аспект непосредственным образом влияет на геймплей.

Сюжетные линии определяются локацией, событиями и участвующими в них персонажами. Сюжет влияет на состояние протагониста.

В течении сюжета протагонист может приближаться и отдаляться от цели. Серьезное изменение относительно цели влияет на состояние протагониста.

Для решения поставленной задачи в заданных условиях было принято решение формализовать параметры так, чтобы представить геймплейную и сюжетную составляющую игры в виде чисел и зависимостей.

Так, сюжет состоит из набора событий. Каждое событие имеет свою координату.

Первая компонента координаты – это время. Условная единица времени называется фазой. Пока длится фаза, длится и событие. Изменение сюжета, как и состояний объектов, участвующих в событии, происходит между фазами, то есть в результате события.

Вторая компонента – сюжетная линия. Получается, что в единицу времени может быть одно или несколько событий, или их может не быть вовсе. Каждое событие относится к своей сюжетной линии.

За основу визуализации временных потоков была взята структура «yurn» [45], которая позволяет отслеживать переключение между сюжетными линиями. Однако идея была унаследована лишь частично. Задача, которую решает авторская версия визуализации заключается в том, чтобы показать, из каких сюжетных веток формируется развитие и путь героя. Тем временем, сюжетные ветки не могу существовать без

[9] Протагонист – главный герой повествования, центральное действующее лицо.

протагониста, поэтому вне посещения текущего события другие события возможны, только если они подразумевают варианты изменения состояний персонажа.

Так появляется возможность показать параллельные ветви развития сюжета: разные игроки будут проходить историю по-своему.

Наличие двух линий сюжета и только одного возможного состояния протагониста формирует график, в котором видно, как путь протагониста может составлять разные события в зависимости от выбранной ветки сюжета (рис. 5).

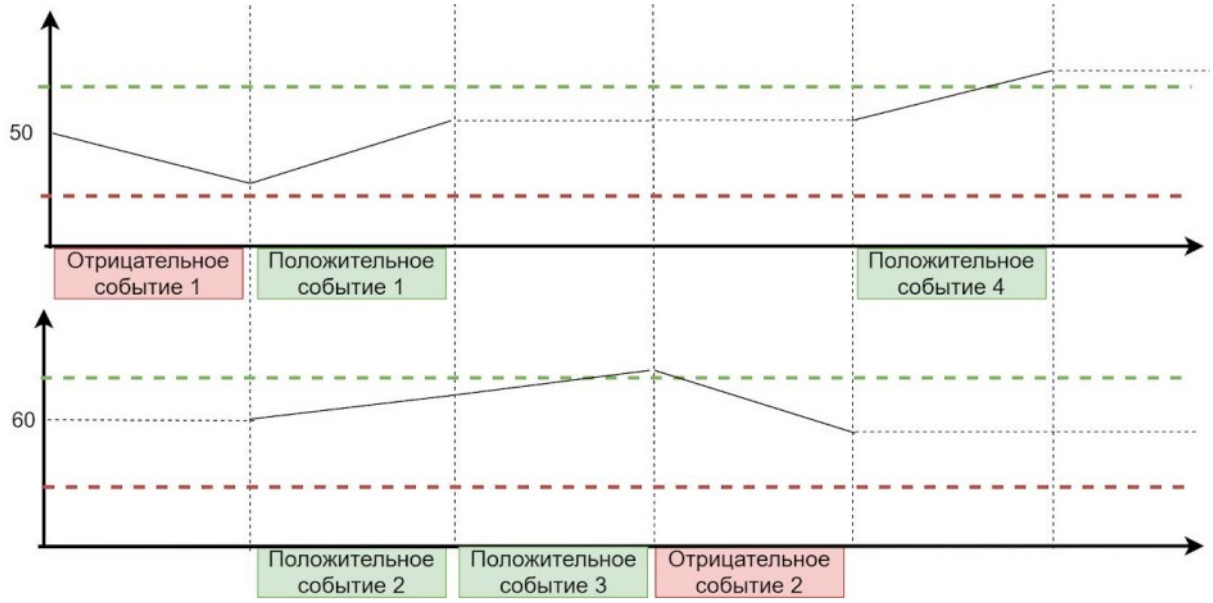

Рис. 5. Диаграмма сюжетных линий

«Объекты истории» имеют одно из трех состояний: негативное, нейтральное, положительное.

Состояние сюжетной линии оценивается с точки зрения протагониста, т.е. приближает результат события к цели или отдаляет.

Исходя из этого можно сделать вывод, что состояние события влияет на состояние цели, что в свою очередь влияет на состояния протагониста. Простыми словами, маленькие победы радуют игрока, а неудачи вызывают огорчение.

Вместе с тем, каждое событие имеет измеримый результат, который представляет собой число в пределе [-100; 100]. Отрицательное значение имеют события с негативной окраской, а положительное – с позитивной. Величина значения зависит от того, насколько большое у события влияние на остальные компоненты системы.

Изменение состояний объектов должно влиять на игровые механики и характеристики. Поэтому любое изменение в системе сопровождается активацией одной механики. Вследствие этого появляется полярность: при изменении состояния в положительное активизируется одна механика, а

при изменении в негативную сторону – другая. В нейтральном состоянии доступна механика по умолчанию.

Негативному и положительному состоянию каждого «объекта истории» присваивается по одной дополнительной игровой механике. Следовательно, в разных ситуациях игроку будут доступны усиления и ослабления, соответствующие контексту.

Механики присваиваются по следующим принципам:

- когда сюжет влияет на внешнюю среду, механики тоже влияют вовне: ослабляют или усиливают других персонажей и артефакты;
- когда сюжет влияет на состояние протагониста и его цели, механики воздействуют на его возможности и способности: ограничивают или расширяют их.

Еще «объектам истории» присваивается зависимая характеристика. Ее реализация также привязана к контексту и имеет числовое выражение.

Приведем пример: цель протагониста – «стать героем» (рис. 6). Игровым дизайнером задается ассоциация данной цели с ощущением у протагониста собственной самоотверженности.

Какая механика может иллюстрировать самоотверженность протагониста в положительном ключе? В данном случае, это зависимость урона от уровня здоровья: чем ближе герой к смерти, тем больше урона наносят его удары.

В противоположность этому, в негативном ключе активизируется другая зависимость: при нанесении урона герой повышает свой уровень здоровья.

Игрок, в свою очередь, может иметь свое представление о герое, как о неуязвимом персонаже. Тем самым, зависимой характеристикой становится способность к регенерации.

Обобщенно получаются следующие механики:

1. положительная механика соответствует самоотверженности: ее наличие мотивирует игрока не бояться урона;
2. зависимая характеристика позволяет игроку испытывать чувство безопасности, не опасаться гибели персонажа;
3. негативная механика вынуждает испытывать чувства жестокости, конфликта с собственными ощущениями и взглядами его персонажа.

Механики коррелируют между собой следующим образом:

1. положительная механика и высокое значение зависимой характеристики вызывают у игрока ощущение героизма, т.к. здоровье можно пускать в расход, регенерация будет поддерживать игрока на грани;
2. негативная механика и низкое значение зависимой характеристики вызывают у игрока страх и дискомфорт, а также чувство эгоизма ради выживания, что противоположно образу героя.

Рис. 6. Корреляция цели героя и механик

Система реализована в виде библиотеки классов, написанной на языке C#. Данные поступают в виде json-файла. Привязки к жанру нет, что делает инструмент универсальным в применении.

На основании всех введенных пользователем данных система выводит визуализацию: графики, таблицы (рис. 7). В последующем можно продолжить корректировку данных.

	Значение	Характеристика	Критическая механика	Результат
Грань: злость	20	Высокий бонус к урону	Наносимый урон дает очки действия на следующий ход	
Грань: страх	40	Шанс промахнуться чуть завышен	--	- 20
Цель: герой	50	Стандартная регенерация	--	- 20
Линия: дракон	50	Стандартный урон дракона	--	- 20
Линия: принцесса	Не активна	--	--	
Линия: родной город	Не активна	--	--	

Рис. 7. Визуализация события в виде таблицы

Надо отметить, что система также способна подсвечивать ошибки разработчика, связанные с потоком повествования.

Подобная возможность реализована по большей части благодаря тому, что все введенные сущности, которыми оперирует система, представлены в числах. Готовые функции последовательно проверяют значения на логичность и соответствии контексту, заданному разработчиком.

В качестве примера можно выделить ситуацию с частой переменчивостью в сюжете (рис. 8). Алгоритм видит резкое изменение числовых значений и указывает на это.

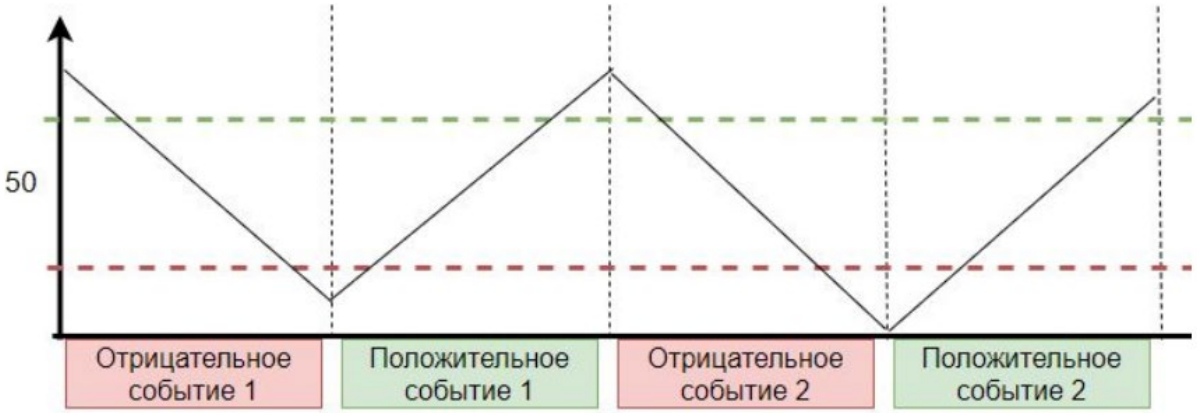

Рис. 8. Ошибочный график: резкая смена событий

Полученный инструмент имеет уникальный функционал. При всем объеме ограничений он является универсальным редактором, который может стать частью общей экосистемы генератора игровых прототипов.

Функционал может выглядеть нагромождением ручных настроек, но в будущем предвидится автоматизация ряда процессов для ускорения работы с сюжетом и геймплеем, а также с игровым балансом.

5. Место игрового баланса в генераторе игровых прототипов

Задача автоматического балансирования завязана на контексте. В рамках работы над инструментом генерации сценарного прототипа [1, 3, 5, 8] игровой баланс занимает свое место в конвейере преобразования данных.

Так, извлечение информации о контексте реализуема методами семантического анализа текстовой документации, а автобалансировка может проводиться благодаря машинному обучению.

Идея о генерации игровых прототипов на основе текстовой документации призвана сэкономить ресурсы, которые могут быть потрачены на более продуктивные задачи, а также на то, чтобы повысить качественный уровень разрабатываемой игры.

Работы, приведенные в данном исследовании, доказывают, что автоматизировать рутину возможно без потери авторского замысла.

Проанализируем обновленную структуру инструмента генерации сценарного прототипа с точки зрения игрового баланса (рис. 9).

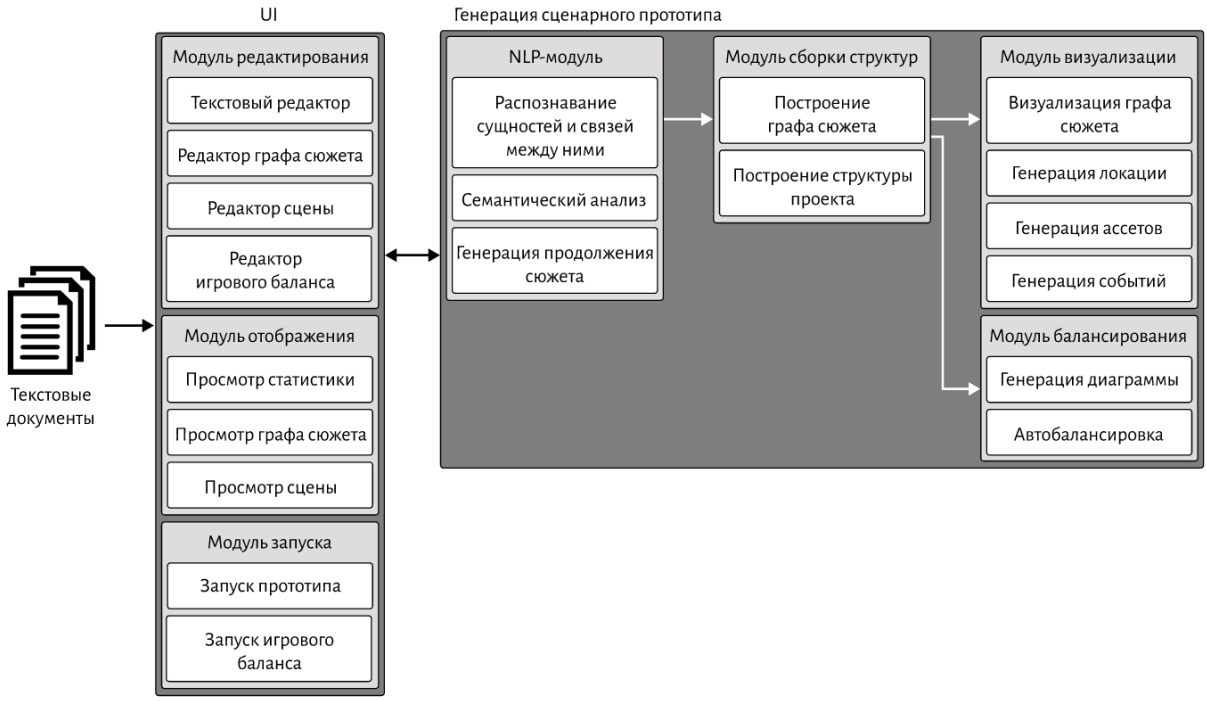

Рис. 9. Структура инструмента генерации сценарного прототипа

Инструмент принимает на вход набор текстовых документов.

Минуя внешнюю оболочку приложения (UI), рассмотрим внутреннюю систему обработки данных.

Первый этап, который проходит текст, это распознавание сущностей и связей между ними. В более ранних работах выделялись локации, персонажей и артефакты сценария, всевозможные предметы, без которых повествование невозможно [1, 46, 47]. Данная задача решается классическими методами NLP.

Однако возможно расширение анализируемых параметрах. Так, в документации могут скрываться требования к игровому балансу, описанные в текстовой форме. Это может быть художественное описание боя, однако благодаря использованию таких характерных эпитетов, как “мощная атака”, “быстрый удар”, “уворот” и т.п., возможно формализовать параметры усилений и способностей, доступных игроку в связи с контекстом.

Таким образом, предполагается, что зачастую в документации содержится вся необходимая для анализа информация. Перевод этой информации в язык параметров и функций значительно облегчает программирование и реализацию игры.

На первых порах решение о присвоении той или иной схемы игрового баланса для текущего контекста может решаться присвоением

наиболее подходящего шаблона из готового набора. В дальнейшем данный этап может быть полностью автоматизирован.

Извлеченные для игрового баланса параметры можно заполнить автоматически в случайном порядке. Этого может оказаться достаточно для тестирования. В дальнейшем, в результате исследования большого количества существующих систем возможно более осознанное присвоение значений. В данном вопросе уместно применение алгоритмов машинного обучения.

И наконец полученную модель игрового баланса можно воплотить в виде логики и кода, используя возможности визуального программирования и готовые библиотеки.

При всей амбициозности идеи генерации игр из текста необходимо помнить, что честный взгляд на автоматическую генерацию цифрового контекста – это, в первую очередь, лояльность по отношению к артефактам генерации и неточностям. Задача автоматизации может не решать проблему производства качественного контента, но может ускорять этап прототипирования, а также может позволить задействовать минимальное количество специалистов.

Необходимость в делегировании рутиной работы имеется в любых компаниях производства интерактивных проектов. Данный факт подтверждает опыт работы авторов статьи. Любой творческий проект затормаживает этап, когда существует неопределенность в том, какое решение для реализации задачи будет правильным, выигрышным.

Попытка манипулировать параметрами уже была предпринята в опубликованных работах [8, 46, 47, 48, 49]. Успех в реализации малых задач говорит о том, что в дальнейшем локальные задачи можно будет объединить под эгидой объемного многокомпонентного инструмента, способного заменить, а где-то и предложить новый способ решения задач разработки и прототипирования.

Заключение

В работе рассмотрены проблемы балансирования игры, рассмотрены исторические прочтения балансировки, дано собственное определение, рассмотрены публикации с конкретными примерами балансировки.

Полученные результаты экспериментов скорректировали видение итогового инструмента. Сформулированное представление об интеграции этапа балансирования игры видится реалистичным.

Поставлена амбициозная задача создать универсальное и в перспективе самообучающееся решение, которое динамически связывает изменяющиеся характеристики персонажей и локаций, автоматически ищет уязвимые места баланса и их решение с математической точки зрения.

Итак, можно сформулировать будущие задачи реализации модуля игрового баланса в рамках комплексного инструмента генерации сценарного прототипа:

1. создание более качественной обработки данных для излечения математических и параметрических значений из текста на естественном языке;
2. обработка извлеченных данных для генерации игровой модели;
3. создание алгоритмов автоматического балансирования игровой модели;
4. имплементация полученного игрового баланса в виде игровой логики.

Каждый пункт является сложным техническим решением, поэтому будет рассматриваться изолировано. В дальнейшем полученные результаты будут объединены в единый компонент инструмента.

Потребность в подобном инструменте также обусловлена тем, что в сфере дизайна требуются методы поиска оптимальных и качественных решений, что возможно при наличии соответствующих баз знаний и алгоритмов.

Связка с другими элементами конвейера прототипирования видеоигры, о которых упоминалось в более ранних работах авторов, позволяет получить мастерский нарративный дизайн и сбалансированную систему переменных.

Благодарности

Работа выполнена за счет средств Программы стратегического академического лидерства Казанского (Приволжского) федерального университета («ПРИОРИТЕТ-2030»).

Литература

1. Сахибгареева Г.Ф., Кугуракова В.В. Концепт инструмента автоматического создания сценарного прототипа компьютерной игры // Электронные библиотеки, 2018. — Т. 21. — № 3–4. — С. 235–249.
2. Сахибгареева Г.Ф., Бедрин О.А., Кугуракова В.В. Разработка компонента генерации визуализации сценарного прототипа видеоигр // Научный сервис в сети Интернет: труды XXII Всероссийской научной конференции (21-25 сентября 2020 г., онлайн). — М.: ИПМ им. М.В.Келдыша, 2020. — С. 581–603. — https://doi.org/10.20948/abrau-2020-4
3. Сахибгареева Г.Ф., Бедрин О.А., Кугуракова В.В. Раскадровка как одно из представлений сценарного прототипа компьютерных игр // Электронные библиотеки, 2021. — Т. 24. — № 2. — С. 408–444.

4. Sahibgareeva G.F., Bedrin O.A., Kugurakova V.V. Visualization Component for the Scenario Prototype Generator as a Video Game Development Tool // Proceedings of the 22nd Conference on Scientific Services & Internet (September 21-25, 2020, online). — CEUR Workshop Proceedings, 2020. — V. 2784. — P. 267–282.
5. Кугуракова В.В., Сахибгареева Г.Ф., Нгуен А.З., Астафьев А.М. Пространственная ориентация объектов на основе обработки текстов на естественном языке для генерации раскадровок // Электронные библиотеки, 2020. — Т. 23. — № 6. — С. 1213–1238.
6. Сахибгареева Г.Ф. Применимость разветвленных структур для генерации сценарных прототипов видеоигр // 65-я Международная научная конференция Астраханского государственного технического университета, 2021.
7. Сахибгареева Г.Ф., Кугуракова В.В. Прототипирование вариативности сюжета компьютерных игр // Научный сервис в сети Интернет: труды XXIII Всероссийской научной конференции (20–23 сентября 2021 г., онлайн). — М.: ИПМ им. М.В.Келдыша, 2021. — С. 347–360. — https://doi.org/10.20948/abrau-2021-11
8. Сахибгареева Г. Ф., Кугуракова В. В. Редактор интерактивной структуры для инструмента генерации сценарных прототипов // Электронные библиотеки, 2022. — Т. 24. — № 6. — С. 1184–1202.
9. Sahibgareeva G.F, Kugurakova V.V. Branched Structure Component for a Video Game Scenario Prototype Generator // Proceedings of the 23nd Conference on Scientific Services & Internet (September 20–23, 2021, online). — CEUR Workshop Proceedings, 2021. — V. 3066. — P. 101–111. — https://doi.org/10.20948/abrau-2021-10-ceur
10. Andrade G., Ramalho G., Gomes A.S., Corruble V. Dynamic game balancing: An evaluation of user satisfaction // Proceedings of the 2nd Conference on Artificial Intelligence and Interactive Digital Entertainment (June 20–23, 2006, Marina del Rey, California). — AAAI Digital Library, 2006. — V. 2. — № 1. — P. 3-8.
11. Becker A., Görlich D. Game balancing — A semantical analysis // First International Workshop on Video Games, Gamification and Educational Innovation (November 6–9, 2019, Madrid, Spain). — CEUR Workshop Proceedings, 2019. — V. 2486. — P. 344–359.
12. Rouse R. Game design: Theory and practice // Plano: Jones & Bartlett Learning, 2005. — 704 p.
13. Game balance concepts. — http://gamebalanceconcepts.wordpress.com/
14. Novak J. Game development essentials: an introduction // Melbourne: Cengage Learning, 2012. — 510 p.
15. Rollings A., Adams E. Andrew Rollings and Ernest Adams on game design. Indianapolis: New Riders Publishing, 2003. — 621 p.

16. Perfect imbalance — why unbalanced design creates balanced play. — https://youtu.be/e31OSVZF77w
17. Sylvester T. Designing games: A guide to engineering experiences // Sebastopol: O'Reilly Media, 2013. — 413 p.
18. Schell J. The Art of Game Design: A Book of Lenses // Boca Raton: A K Peters/CRC Press, 2015. — 600 p.
19. Tijs T. J. V, Brokken D., IJsselsteijn W. A. Dynamic game balancing by recognizing affect // Fun and Games: Second International Conference (October 20–21, 2008, Eindhoven, Netherlands). — Heidelberg: Springer-Verlag GmbH, 2008. — V. 5294. — P. 88–93. — https://doi.org/10.1007/978-3-540-88322-7_9
20. V. Volz, G. Rudolph, B. Naujoks. Demonstrating the feasibility of automatic game balancing // Genetic and Evolutionary Computation Conference. — New York: Association for Computing Machinery, 2016. — P. 269–276. — https://doi.org/10.1145/2908812.2908913
21. Pfau J., Liapis A., Yannakakis G.N., Malaka R. Dungeons & Replicants II: Automated Game Balancing Across Multiple Difficulty Dimensions via Deep Player Behavior Modeling // IEEE Transactions on Games, 2022. — https://doi.org/10.1109/TG.2022.3167728
22. The Tower of Aion. — https://www.ncsoft.jp/aion/
23. Holmgard C., Green M., Liapis A., Togelius J. Automated playtesting with procedural personas through MCTS with evolved heuristics // IEEE Transactions on Games, 2018. — V. 11. — № 4. — P. 352–362. — https://doi.org/10.1109/TG.2018.2808198
24. O. Keehl, A. M. Smith. Monster carlo 2: Integrating learning and tree search for machine playtesting // IEEE Conference on Games, 2019. — P. 1–8.
25. Beau P., Bakkes S. Automated game balancing of asymmetric video games // IEEE Conference on Computational Intelligence and Games (September 20–23, 2016, Santorini, Greece). — IEEE, 2016. — P. 333–340. — https://doi.org/10.1109/CIG.2016.7860432
26. Moroşan M., Poli R. Automated Game Balancing in Ms PacMan and StarCraft Using Evolutionary Algorithms // 20th European Conference on the Applications of Evolutionary Computation (April 19–21, 2017, Amsterdam, Netherlands). — Springer, 2017. — V. 10199. — P. 377–392. — https://doi.org/10.1007/978-3-319-55849-3_25
27. Leigh R., Schonfeld J., Louis S. J. Using coevolution to understand and validate game balance in continuous games // 10th annual conference on Genetic and Evolutionary Computation (July 12–16, 2008, Atlanta, USA). — New York: ACM, 2008. — P. 1563–1570. — https://doi.org/10.1145/1389095.1389394
28. Gudmundsson S. F., Eisen P., Poromaa E., Nodet A., Purmonen S., Kozakowski B., Meurling R., Cao L. Human-like playtesting with deep

learning // IEEE Conference on Computational Intelligence and Games (August 14–17, 2018, Maastricht, Netherlands). — IEEE, 2018. — https://doi.org/10.1109/CIG.2018.8490442
29. Volz V., Rudolph G., Naujoks B. Demonstrating the feasibility of automatic game balancing // Genetic and Evolutionary Computation Conference (July 20–24, 2016, Denver, USA). — New York: ACM, 2016. — P. 269–276. — https://doi.org/10.1145/2908812.2908913
30. Avila A.M., Fonoberova M., Hespanha J.P., Mezic I., Clymer D., Goldstein J., Pravia M.A., Javorsek D. Game Balancing using Koopman-based Learning // American Control Conference (May 25–28, 2021, New Orleans, USA). — IEEE, 2021. — P. 710. — https://doi.org/10.23919/ACC50511.2021.9483027
31. M. Morosan, Poli P. Lessons from Testing an Evolutionary Automated Game Balancer in Industry // IEEE Games, Entertainment, Media Conference (August 15–17, 2018, Galway, Ireland). — IEEE, 2018. — P. 263-270. — https://doi.org/10.1109/GEM.2018.8516447
32. Karavolos D., Liapis A., Yannakakis G.N. Using a Surrogate Model of Gameplay for Automated Level Design // IEEE Conference on Computational Intelligence and Games (August 14–17, 2018, Maastricht, Netherlands). — IEEE, 2018. — https://doi.org/10.1109/CIG.2018.8490425
33. Game Balance Concepts. — https://gamebalanceconcepts.wordpress.com/
34. Wang W., Zhang R. Improved Game Units Balancing In Game Design Through Combinatorial Optimization // IEEE International Conference on e-Business Engineering (April 11, 2022, Guangzhou, China). — IEEE, 2022. — https://doi.org/10.1109/ICEBE52470.2021.00022
35. Liu Z.-Q., Leung K.-M. Script visualization (ScriptViz): A smart system that makes writing fun // Soft Computing. — Springer Nature Switzerland AG, 2006. — V. 10. — № 1. — P. 34–40. — https://doi.org/10.1007/s00500-005-0461-4
36. Gupta T., Schwenk D., Farhadi A., Hoiem D., Kembhavi A. Imagine This! Scripts to Compositions to Videos // 15th European Conference of Computer Vision (September 8–14, 2018, Munich, Germany). — Springer, 2018. — V. 11212. — https://doi.org/10.48550/arXiv.1804.03608
37. Akser M., Bridges B., Campo G., Cheddad A., Curran K., Fitzpatrick L., Hamilton L., Harding J., Leath T., Lunney T., Lyons F., Ma M., Macrae J., Maguire T., McCaughey A., McClory E., McCollum V., Mc Kevitt P., Melvin A., Moore P., Mulholland E., Muñoz K., O'Hanlon G., Roman L. SceneMaker: Creative technology for digital storytelling // 5th EAI International Conference: ArtsIT, Interactivity & Game Creation (May 2–3, 2016, Esbjerg, Denmark). — Springer Nature Switzerland AG, 2016. — V. 196. — P. 29–38. — https://doi.org/10.1007/978-3-319-55834-9_4
38. Dworak W., Filgueiras E., Valente J. Automatic Emotional Balancing in Game Design: Use of Emotional Response to Increase Player Immersion //

9th International Conference on Design, User Experience, and Usability (July 19-24, 2020, Copenhagen, Denmark). — Springer, 2020. — V. 12201. — P. 426–438. — https://doi.org/10.1007/978-3-030-49760-6_30
39. Machinations. — https://machinations.io/
40. Черечукина А.Н. Содержание GDD как требований к разработке программного обеспечения // Казанский федеральный университет, 2019. — 47 с. — https://kpfu.ru/student_diplom/10.160.178.20_236517_F_Cherechukina_1_.pdf
41. The Elder Scrolls V: Skyrim. — https://elderscrolls.bethesda.net/ru/skyrim
42. Галимзянов Г.Р. Разработка инструмента автоматической корректировки внутриигровых параметров // Казанский федеральный университет, 2021. — 35 с. — https://kpfu.ru/student_diplom/10.160.178.20_TXB9250VCS6S6OVSLZOCXQDP4J7WFCRV__J7FXN80EEZNIXS6Q_Galimzyanov.pdf
43. Dungeons & Dragons. — https://dnd.wizards.com/
44. Липатов В.В. Автоматическая генерация игровых механик в соответствии с нарративом // Казанский федеральный университет, 2019. — 78 с. — https://kpfu.ru/student_diplom/10.160.178.20_5065127_F_Lipatov.pdf
45. Padia K., Bandara K., Healey C. A system for generating storyline visualizations using hierarchical task network planning // Computers & Graphics. — Elsevier, 2019. — V. 78. — P. 64–75. — https://doi.org/10.1016/j.cag.2018.11.004
46. Доброквашина А.С., Газизова Э.А. Автоматизация проектирования игрового прототипа на основании обработки формализованного игрового дизайн-документа // Ученые записки ИСГЗ. — Казань: Институт социальных и гуманитарных знаний, 2019. — Т. 17. — № 1. — С. 583–589.
47. Вакатов С.А. Разработка инструмента вариативности сюжета с запуском прототипа в виде текстовой игры // Казанский федеральный университет, 2021. — 36 с. — https://kpfu.ru/student_diplom/10.160.178.20_TTKKD9XW59RG5L7TVLTB73YPTISE59Y16W5D1U435WOXWI10US_Vakatov.pdf
48. Вакатова Э.С. Разработка функционала генерации продолжения сюжета для инструмента прототипирования сюжета в компьютерных играх // Казанский федеральный университет, 2021. — 34 с. — https://kpfu.ru/student_diplom/10.160.178.20_PQK51KDGAPZ5Z82IKYY69MV84PCLTPERV0NNYJ33B7P5T7NJFP_F_Vakatova.pdf

References

1. Sahibgareeva G.F., Kugurakova V.V. Koncept instrumenta avtomaticheskogo sozdaniya scenarnogo prototipa komp'yuternoj igry // Elektronnye biblioteki, 2018. — T. 21. — № 3–4. — P. 235–249.
2. Sahibgareeva G.F., Bedrin O.A., Kugurakova V.V. Razrabotka komponenta generacii vizualizacii scenarnogo prototipa videoigr // Nauchnyj servis v seti Internet: trudy XXII Vserossijskoj nauchnoj konferencii (21-25 sentyabrya 2020 g., onlajn). — M.: IPM im. M.V.Keldysha, 2020. — P. 581–603. — https://doi.org/10.20948/abrau-2020-4
3. Sahibgareeva G.F., Bedrin O.A., Kugurakova V.V. Raskadrovka kak odno iz predstavlenij scenarnogo prototipa komp'yuternyh igr // Elektronnye biblioteki, 2021. — T. 24. — № 2. — P. 408–444.
4. Sahibgareeva G.F., Bedrin O.A., Kugurakova V.V. Visualization Component for the Scenario Prototype Generator as a Video Game Development Tool // Proceedings of the 22nd Conference on Scientific Services & Internet (September 21-25, 2020, online). — CEUR Workshop Proceedings, 2020. — V. 2784. — P. 267–282.
5. Kugurakova V.V., Sahibgareeva G.F., Nguen A.Z., Astaf'ev A.M. Prostranstvennaya orientaciya ob"ektov na osnove obrabotki tekstov na estestvennom yazyke dlya generacii raskadrovok // Elektronnye biblioteki, 2020. — T. 23. — № 6. — P. 1213–1238.
6. Sahibgareeva G.F. Primenimost' razvetvlennyh struktur dlya generacii scenarnyh prototipov videoigr // 65-ya Mezhdunarodnaya nauchnaya konferenciya Astrahanskogo gosudarstvennogo tekhnicheskogo universiteta, 2021.
7. Sahibgareeva G.F., Kugurakova V.V. Prototipirovanie variativnosti syuzheta komp'yuternyh igr // Nauchnyj servis v seti Internet: trudy XXIII Vserossijskoj nauchnoj konferencii (20–23 sentyabrya 2021 g., onlajn). — M.: IPM im. M.V.Keldysha, 2021. — P. 347–360. — https://doi.org/10.20948/abrau-2021-11
8. Sahibgareeva G. F., Kugurakova V. V. Redaktor interaktivnoj struktury dlya instrumenta generacii scenarnyh prototipov // Elektronnye biblioteki, 2022. — T. 24. — № 6. — P. 1184–1202.
9. Sahibgareeva G.F, Kugurakova V.V. Branched Structure Component for a Video Game Scenario Prototype Generator // Proceedings of the 23nd Conference on Scientific Services & Internet (September 20–23, 2021, online). — CEUR Workshop Proceedings, 2021. — V. 3066. — P. 101–111. — https://doi.org/10.20948/abrau-2021-10-ceur
10. Andrade G., Ramalho G., Gomes A.S., Corruble V. Dynamic game balancing: An evaluation of user satisfaction // Proceedings of the 2nd Conference on Artificial Intelligence and Interactive Digital Entertainment (June 20–23, 2006, Marina del Rey, California). — AAAI Digital Library, 2006. — V. 2. — № 1. — P. 3-8.

11. Becker A., Görlich D. Game balancing — A semantical analysis // First International Workshop on Video Games, Gamification and Educational Innovation (November 6–9, 2019, Madrid, Spain). — CEUR Workshop Proceedings, 2019. — V. 2486. — P. 344–359.
12. Rouse R. Game design: Theory and practice // Plano: Jones & Bartlett Learning, 2005. — 704 p.
13. Game balance concepts. — http://gamebalanceconcepts.wordpress.com/
14. Novak J. Game development essentials: an introduction // Melbourne: Cengage Learning, 2012. — 510 p.
15. Rollings A., Adams E. Andrew Rollings and Ernest Adams on game design. Indianapolis: New Riders Publishing, 2003. — 621 p.
16. Perfect imbalance — why unbalanced design creates balanced play. — https://youtu.be/e31OSVZF77w
17. Sylvester T. Designing games: A guide to engineering experiences // Sebastopol: O'Reilly Media, 2013. — 413 p.
18. Schell J. The Art of Game Design: A Book of Lenses // Boca Raton: A K Peters/CRC Press, 2015. — 600 p.
19. Tijs T. J. V, Brokken D., IJsselsteijn W. A. Dynamic game balancing by recognizing affect // Fun and Games: Second International Conference (October 20–21, 2008, Eindhoven, Netherlands). — Heidelberg: Springer-Verlag GmbH, 2008. — V. 5294. — P. 88–93. — https://doi.org/10.1007/978-3-540-88322-7_9
20. V. Volz, G. Rudolph, B. Naujoks. Demonstrating the feasibility of automatic game balancing // Genetic and Evolutionary Computation Conference. — New York: Association for Computing Machinery, 2016. — P. 269–276. — https://doi.org/10.1145/2908812.2908913
21. Pfau J., Liapis A., Yannakakis G.N., Malaka R. Dungeons & Replicants II: Automated Game Balancing Across Multiple Difficulty Dimensions via Deep Player Behavior Modeling // IEEE Transactions on Games, 2022. — https://doi.org/10.1109/TG.2022.3167728
22. The Tower of Aion. — https://www.ncsoft.jp/aion/
23. Holmgard C., Green M., Liapis A., Togelius J. Automated playtesting with procedural personas through MCTS with evolved heuristics // IEEE Transactions on Games, 2018. — V. 11. — № 4. — P. 352–362. — https://doi.org/10.1109/TG.2018.2808198
24. O. Keehl, A. M. Smith. Monster carlo 2: Integrating learning and tree search for machine playtesting // IEEE Conference on Games, 2019. — P. 1–8.
25. Beau P., Bakkes S. Automated game balancing of asymmetric video games // IEEE Conference on Computational Intelligence and Games (September 20–23, 2016, Santorini, Greece). — IEEE, 2016. — P. 333–340. — https://doi.org/10.1109/CIG.2016.7860432

26. Moroşan M., Poli R. Automated Game Balancing in Ms PacMan and StarCraft Using Evolutionary Algorithms // 20th European Conference on the Applications of Evolutionary Computation (April 19–21, 2017, Amsterdam, Netherlands). — Springer, 2017. — V. 10199. — P. 377–392. — https://doi.org/10.1007/978-3-319-55849-3_25
27. Leigh R., Schonfeld J., Louis S. J. Using coevolution to understand and validate game balance in continuous games // 10th annual conference on Genetic and Evolutionary Computation (July 12–16, 2008, Atlanta, USA). — New York: ACM, 2008. — P. 1563–1570. — https://doi.org/10.1145/1389095.1389394
28. Gudmundsson S. F., Eisen P., Poromaa E., Nodet A., Purmonen S., Kozakowski B., Meurling R., Cao L. Human-like playtesting with deep learning // IEEE Conference on Computational Intelligence and Games (August 14–17, 2018, Maastricht, Netherlands). — IEEE, 2018. — https://doi.org/10.1109/CIG.2018.8490442
29. Volz V., Rudolph G., Naujoks B. Demonstrating the feasibility of automatic game balancing // Genetic and Evolutionary Computation Conference (July 20–24, 2016, Denver, USA). — New York: ACM, 2016. — P. 269–276. — https://doi.org/10.1145/2908812.2908913
30. Avila A.M., Fonoberova M., Hespanha J.P., Mezic I., Clymer D., Goldstein J., Pravia M.A., Javorsek D. Game Balancing using Koopman-based Learning // American Control Conference (May 25–28, 2021, New Orleans, USA). — IEEE, 2021. — P. 710. — https://doi.org/10.23919/ACC50511.2021.9483027
31. M. Morosan, Poli P. Lessons from Testing an Evolutionary Automated Game Balancer in Industry // IEEE Games, Entertainment, Media Conference (August 15–17, 2018, Galway, Ireland). — IEEE, 2018. — P. 263-270. — https://doi.org/10.1109/GEM.2018.8516447
32. Karavolos D., Liapis A., Yannakakis G.N. Using a Surrogate Model of Gameplay for Automated Level Design // IEEE Conference on Computational Intelligence and Games (August 14–17, 2018, Maastricht, Netherlands). — IEEE, 2018. — https://doi.org/10.1109/CIG.2018.8490425
33. Game Balance Concepts. — https://gamebalanceconcepts.wordpress.com/
34. Wang W., Zhang R. Improved Game Units Balancing In Game Design Through Combinatorial Optimization // IEEE International Conference on e-Business Engineering (April 11, 2022, Guangzhou, China). — IEEE, 2022. — https://doi.org/10.1109/ICEBE52470.2021.00022
35. Liu Z.-Q., Leung K.-M. Script visualization (ScriptViz): A smart system that makes writing fun // Soft Computing. — Springer Nature Switzerland AG, 2006. — V. 10. — № 1. — P. 34–40. — https://doi.org/10.1007/s00500-005-0461-4
36. Gupta T., Schwenk D., Farhadi A., Hoiem D., Kembhavi A. Imagine This! Scripts to Compositions to Videos // 15th European Conference of

Computer Vision (September 8–14, 2018, Munich, Germany). — Springer, 2018. — V. 11212. — https://doi.org/10.48550/arXiv.1804.03608
37. Akser M., Bridges B., Campo G., Cheddad A., Curran K., Fitzpatrick L., Hamilton L., Harding J., Leath T., Lunney T., Lyons F., Ma M., Macrae J., Maguire T., McCaughey A., McClory E., McCollum V., Mc Kevitt P., Melvin A., Moore P., Mulholland E., Muñoz K., O'Hanlon G., Roman L. SceneMaker: Creative technology for digital storytelling // 5th EAI International Conference: ArtsIT, Interactivity & Game Creation (May 2–3, 2016, Esbjerg, Denmark). — Springer Nature Switzerland AG, 2016. — V. 196. — P. 29–38. — https://doi.org/10.1007/978-3-319-55834-9_4
38. Dworak W., Filgueiras E., Valente J. Automatic Emotional Balancing in Game Design: Use of Emotional Response to Increase Player Immersion // 9th International Conference on Design, User Experience, and Usability (July 19-24, 2020, Copenhagen, Denmark). — Springer, 2020. — V. 12201. — P. 426–438. — https://doi.org/10.1007/978-3-030-49760-6_30
39. Machinations. — https://machinations.io/
40. CHerechukina A.N. Soderzhanie GDD kak trebovaniĭ k razrabotke programmnogo obespecheniya // Kazanskij federal'nyj universitet, 2019. — 47 p. — https://kpfu.ru/student_diplom/10.160.178.20_236517_F_Cherechukina_1_.pdf
41. The Elder Scrolls V: Skyrim. — https://elderscrolls.bethesda.net/ru/skyrim
42. Galimzyanov G.R. Razrabotka instrumenta avtomaticheskoĭ korrektirovki vnutriigrovyh parametrov // Kazanskij federal'nyj universitet, 2021. — 35 p. — https://kpfu.ru/student_diplom/10.160.178.20_TXB9250VCS6S6OVSLZOCXQDP4J7WFCRV__J7FXN80EEZNIXS6Q_Galimzyanov.pdf
43. Dungeons & Dragons. — https://dnd.wizards.com/
44. Lipatov V.V. Avtomaticheskaya generaciya igrovyh mekhanik v sootvetstvii s narrativom // Kazanskij federal'nyj universitet, 2019. — 78 p. — https://kpfu.ru/student_diplom/10.160.178.20_5065127_F_Lipatov.pdf
45. Padia K., Bandara K., Healey C. A system for generating storyline visualizations using hierarchical task network planning // Computers & Graphics. — Elsevier, 2019. — V. 78. — P. 64–75. — https://doi.org/10.1016/j.cag.2018.11.004
46. Dobrokvashina A.S., Gazizova E.A. Avtomatizaciya proektirovaniya igrovogo prototipa na osnovanii obrabotki formalizovannogo igrovogo dizajn-dokumenta // Uchenye zapiski ISGZ. — Kazan': Institut social'nyh i gumanitarnyh znanij, 2019. — T. 17. — № 1. — P. 583–589.
47. Vakatov S.A. Razrabotka instrumenta variativnosti syuzheta s zapuskom prototipa v vide tekstovoj igry // Kazanskij federal'nyj universitet, 2021. — 36 p. —

https://kpfu.ru/student_diplom/10.160.178.20_TTKKD9XW59RG5L7TVLTB73YPTISE59Y16W5D1U435WOXWI10US_Vakatov.pdf
48. Vakatova E.S. Razrabotka funkcionala generacii prodolzheniya syuzheta dlya instrumenta prototipirovaniya syuzheta v komp'yuternyh igrah // Kazanskij federal'nyj universitet, 2021. — 34 p. — https://kpfu.ru/student_diplom/10.160.178.20_PQK51KDGAPZ5Z82IKYY69MV84PCLTPERV0NNYJ33B7P5T7NJFP_F_Vakatova.pdf