

УДК 004.414.3

КОНЦЕПТ ИНСТРУМЕНТА АВТОМАТИЧЕСКОГО СОЗДАНИЯ СЦЕНАРНОГО ПРОТОТИПА КОМПЬЮТЕРНОЙ ИГРЫ

Г. Ф. Сахибгареева¹, В. В. Кугуракова²

^{1,2}Высшая школа информационных технологий и интеллектуальных систем
Казанского (Приволжского) федерального университета;

¹gulnara.sahibgareeva42@gmail.com, ²vlada.kugurakova@gmail.com

Аннотация

На основе существующих решений описано представление об архитектуре инструмента генерации сценарного прототипа из текста. Сформированы требования к разрабатываемому инструменту. Разработан прототип инструмента, иллюстрирующий основной принцип работы пользователя с приложением.

Ключевые слова: игровая сценаристика, нарративный дизайн, сценарный прототип, прототипирование, разработка игр, погружение, повествование

ВВЕДЕНИЕ

Повествование – это изображение действия во времени. Обычно оно ведется от имени автора или рассказчика. Другими словами, это упорядоченная последовательность связанных событий, в которых участвуют одна или несколько существ, и которая излагается в виде истории.

В кинематографе повествование наделено дополнительными свойствами: наглядность и звучание. На глазах у зрителя история оживает в виде конкретных образов.

В играх повествование – это не только то, какая история складывается в голове у игрока и то, какие эмоции вызывает увиденное и услышанное, но и то, что называется уникальным игровым опытом. Игрок «проживает» события игры, становится главным героем истории. Грамотно построенное повествование способно погрузить игрока в вымышленный мир, показать ему, какие механики ему

доступны, научить его ими пользоваться и достигать с их помощью внутриигровых целей. Однако реализация такого повествования – не такая простая задача, какой кажется на первый взгляд. В игровой индустрии решаются задачи, с которыми не сталкиваются сценаристы кино и театра.

Игровые сценаристы и нарративные дизайнеры в своей работе опираются на интерактивную природу игр, т. к. они знают, что хорошее повествование лежит в самой структуре игрового дизайна [1]. Поэтому игровой сценарий отличается тем, что нацелен на то, чтобы рассказать историю игрой.

На начальных этапах работы сценариста сложно оценить, будет ли игра захватывающей, выполнит ли повествование свою функцию, не будет ли в сюжете несостыковок. Только качественное повествование способствует погружению игрока в придуманный сценаристом мир, обеспечивает для него уникальный нарративный опыт [2].

Залог качественного повествования – раннее прототипирование, что также является особенностью разработки в игровой разработке [3]. Прототипирование – основа хорошего игрового дизайна. Прототип – это идея в том виде, в котором ее можно протестировать. Другими словами, это упрощенная модель будущей игры. Хорошая аналогия – это эскизы, которые художники набрасывают перед тем, как писать картину. Прототипирование помогает найти ошибки, устранить несостыковки и снизить риски. Нет нужды в том, чтобы беспокоиться о том, как будет выглядеть прототип, главное, как он будет работать [4].

Предлагаемый нами способ прототипирования повествования – создание и тестирование сценарного прототипа. Из-за отсутствия данного термина в научной литературе, дадим его определение.

Сценарный прототип – это модель игрового повествования, представленная в виде, доступном для демонстрации, анализа и тестирования. На основе сценарного прототипа можно оценить качество создаваемой истории, проверить, есть ли интерес к ней со стороны целевой аудитории. Создание сценарного прототипа минимизирует затраты на начальных этапах разработки, т. к. появляется возможность для тестирования.

ПОСТАНОВКА ПРОБЛЕМЫ

Для разработки сценария существуют инструменты, призванные облегчить структурирование сюжета. Но средств, способных автоматизировать процесс со-

здания структуры сценария, еще нет. Раннее прототипирование возможно только после создания структуры вручную в программах, которые представляют сюжет в виде связанных карточек. Прототипирование предполагает быстрый результат без лишних затрат.

Если некоторые игровые механики можно проверить, создав, при возможности, настольную версию игры или упрощенную компьютерную симуляцию, то в случае истории ее нужно рассказать, а лучше, показать команде разработчиков и целевой аудитории игры.

Итак, сформированная проблема заключается в том, что не существует инструмента, способного автоматизировать процесс прототипирования игровой истории.

Развивая идею о том, как может работать инструмент, призванный автоматизировать структурирование сценария, можно предположить, что хорошим выходом станет использование технологий автоматического распознавания текста.

На данный момент сильно прогрессируют в развитии сфера машинного обучения, а также сфера работы с нейросетями и алгоритмы анализа больших массивов текста. В сфере лингвистики такие технологии решают вопросы анализа стилистики текста, его смысловой нагрузки. Имеющиеся технологии позволили реализовать генерацию текста на заданную тему. С каждым разом улучшаются механизмы перевода текста с одного языка на другой.

Вопрос извлечения из текста необходимых сущностей стал обыденностью для анализа соцсетей и продвижения брендов [5]. Поэтому извлечение из текста сущностей и последующая их обработка считаются возможными с точностью, которую предлагают существующие инструменты.

Другой вопрос, связанный с автоматической сборкой сценарного прототипа, решается написанием собственных алгоритмов обработки полученных данных.

АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ

Классический пайплайн

Классический план разработки игры включает в себя несколько последовательных этапов: формирование идеи, предварительная работа, разработка, запуск игры, поддержка игры после запуска [6].

На интересующем нас этапе предварительных работ основное время посвящается решению следующих задач: создание истории, определение временных рамок, создание раскадровки, создание бумажного прототипа, дизайн уровней, реализация игровых механик, расчет бюджета разработки и создание технической документации.

Создание на данном этапе сценарного прототипа означает объединение нескольких видов задач (создание истории, определение временных рамок, создание раскадровки) и создание основы для тестирования. По окончании этого этапа должно стать ясно, приносит ли игра «фан», захватывает ли она игроков, интересная ли у игры история [7].

Выбор технологии анализа текста

Среди существующих инструментов анализа текста были выделены следующие разработки: Texterra, OpenNLP, Watson Natural Language Understanding, Google Cloud Natural Language. Тестирование данных инструментов (см. Табл. 1) показало, что инструмент от Google эффективнее всех.

Таблица 1. Результаты тестирования инструментов

№	Название инструмента	Типы сущностей	Результат
1	Texterra	ORGANIZATION	1
		TIME	1
2	OpenNLP	ORGANIZATION	1
		TIME	1
3	IBM Watson Natural Language Understanding	Facility	2
4	Google Cloud Natural Language	Person	7
		Location	4
		Organization	1
		Other	14
		CustomerGood	1

В рамках исследования мы проанализировали существующие решения, которые реализуют функционал в следующих направлениях: визуализация структуры, визуализация контента, автоматическая сборка контента, комплексные решения.

Когда мы говорим о визуализации структуры повествования, в голову приходит идея о представлении информации в виде графа. Но создатели инструмента StoryFlow пошли дальше и представили структуру в виде особой структуры Yarn [8]. В ней структуре события располагаются в хронологическом порядке: слева направо, причем каждая следующая сущность связана с предыдущей.

Существующие инструменты

У данной разработки есть версия, расширяющая функционал – Extended StoryFlow [9]. В данной версии учтены случаи, когда персонаж проходит определенное событие самостоятельно, и случаи, в которых возможен выбор. Этот инструмент полезен для красивой и понятной визуализации игровых событий, взаимодействия персонажей и определения временных промежутков.

Инструмент, решающий в некотором плане проблему прототипирования повествования, ScriptViz, позволяет визуализировать сценарий в виде сцены, которая генерируется в реальном времени [10]. Инструмент состоит из следующих модулей: модуля понимания текста, модуля планирования верхнего уровня и модуля генерирования сцены.

Пользователь вводит текст своего сценария. Условия для такого текста следующие: текст должен быть написан на английском, иметь понятные формулировки и хорошо структурирован.

Система интерпретирует текст в модуле понимания текста и отправляет результат в модуль планировщика верхнего уровня. Планировщик строит план действий для определенных скриптов. Скрипты выполняют план, а генератор сцен отображает полученный результат.

Данный инструмент рассматривается в исследовании, проведенном для реализации программного решения SceneMaker [11].

SceneMaker планируется разработать для решения следующей задачи – автоматизированное создание анимационного прототипа фильма. Проведенное для этого исследование имеет аналогичную с нашей цель: на основе прототипа иссле-

довать перспективы разрабатываемого продукта, отработать на нем различные подходы и методы.

Пайплайн работы SceneMaker следующий: из готового сценария извлекаются эмоции, настроение и жанр. Данные визуализируются в виде 3D-сцены, в которой эмоции отражаются через мимику, жесты и позу персонажей, через освещение, тайминг, постановку камеры и настройку аудиосопровождения в автоматическом режиме в зависимости от жанра. На основе визуализации можно сделать выводы о проделанной работе.

Для получения информации создатели SceneMaker берут за основу эмоции, извлекаемые из текста. На их основе в сцене формируется действие, определенное вплоть до мимики, жестов, позы и голоса персонажей, а также вплоть до постановки кадра, освещения и звука.

В качестве инструмента, способного на основе описательного текста подобрать подходящий анимационный материал, интересна нейросеть CRAFT. Принцип её работы состоит в следующем: на основе 25 184 фрагментов из мультсериала Flintstones нейросеть по текстовому описанию собирает мультфильм [12]. В собранных сценах, конечно, ещё встречаются ошибки, такие, как неправильное наложение объектов или неправильный выбор анимаций.

Интерес представляет инструмент Machination Tool [13], способный проанализировать интерактивную составляющую игры.

Machinations – это визуальный язык представления игровых механик в виде диаграмм, создающий игровую имитацию без написания кода. Так как игровая механика скрыта в системе и не доступна для анализа, кроме как из процесса игры, использование Machinations считается эффективным для анализа геймплея. Схемы обработки в диаграммах Machinations представляют собой потоки и структуры обратной связи, которые могут существовать в игровой системе. Именно эти структуры обратной связи в значительной степени дают представление о динамике игры.

Рассмотренные инструменты реализуют функционал, который представляет интерес в рамках нашего исследования, но в целом их недостаточно. Решением видится комплексный инструмент, который смог бы использовать существующие наработки.

НАШЕ ПРЕДСТАВЛЕНИЕ

На основе проведенного исследования был создан прототип инструмента, который иллюстрирует первый взгляд на то, какую архитектуру нужно реализовывать для достижения поставленной цели.

Были сформированы следующие требования: инструмент должен принимать любой, не обязательно формализованный, текст; извлекать максимальное количество полезных сущностей; иметь возможность хранить сущность с привязкой к предложениям; иметь возможность редактировать сущности, относящиеся к предложению; иметь возможность сборки сценарного прототипа с участием пользователя.

Инструмент работает по следующему алгоритму: пользователь вводит текст; затем он просматривает полученные сущности и редактирует их; пользователь выбирает режим сборки прототипа; затем он собирает сценарный прототип, который состоит из сцен, создающихся из сущностей, указанных пользователем предложений; пользователь просматривает полученный сценарный прототип.

Интерфейс инструмента состоит из следующих частей: обработка текстовых данных и сборка сценарного прототипа (Рис. 1).

Обработка текстовых данных

Автоматическое понимание текста включает в себя множество этапов анализа и сопоставления полученных результатов. Следовательно, необходимо использовать сторонние сервисы и разрабатывать собственные алгоритмы обработки.

В нашей реализации предполагается ручное редактирование, поэтому лучше, если сторонний сервис сможет распознать максимальное число полезных сущностей. Под полезными подразумеваются сущности, которые потенциально рассматриваются для сборки сценарного прототипа.

Работа инструмента предполагает следование следующему алгоритму: принять введенный текст, отправить его на обработку на сторонний сервис или в используемую библиотеку, принять и распределить сущности по категориям, сохранить введенные изменения.

Под категориями подразумеваются следующие типы сущностей: действующие персонажи, место действия и артефакты (сущности, присутствие в сцене которых обязательно).



Рис. 1. Архитектура разработанного инструмента

Модуль сборки сценария

Получив на выходе необходимые сущности, можно собрать сценарный прототип по подготовленному шаблону. В качестве шаблона выберем следующую структуру: персонажи, место действия, артефакты. В такой структуре хранятся сущности выбранных для сцены предложений.

Представление сценарного прототипа

Реализовать сценарный прототип можно с разной степенью абстракции. Предлагаемый вариант прост в реализации и скорее является достаточным для прототипа, нежели реальным решением поставленной задачи. Это представление сценарного прототипа в виде сцен, каждая из которых перечисляет необходимые для нее сущности: локацию, персонажей и артефакты.

РЕАЛИЗАЦИЯ ПРОТОТИПА

Рассмотрим принцип работы полученного прототипа. Прототип реализован на языке C#, поэтому используются соответствующие термины.

Обработка текстовых данных

Обработка текста осуществляется по принципу, который был сформирован в ходе разработки. Информация по каждому предложению хранится в виде отдельного объекта, который содержит в себе следующие данные: текст предложения и структура, хранящая в себе сущности, которые найдены в этом предложении. В качестве структуры для хранения этих сущностей выбрана структура Dictionary<TKey, TValue>, т. к. данные, которые возвращаются с Google Cloud Language, включают в себя дубликаты.

Для работы с полученными сущностями выделяется несколько категорий: Person, Location и Atreifact. Категории Google Cloud Language отображаются в соответствии с перечисленными категориями: сущности типа Person отображаются в категории Person, сущности типа Location и Organization отображаются в категории Location и все остальные сущности хранятся в категории Atreifact, чтобы не упустить из виду проанализированные сущности.

Редактирование сущностей осуществляется относительно каждого предложения текста. Если Google Cloud Language не определил какие-то сущности или сделал это неправильно, пользователь может это исправить.

Модуль сборки сценария

Для создания сценарного прототипа пользователь должен выбрать несколько предложений, подтвердить свой выбор, и программа выведет значения сущностей, которые будут задействованы в данной сцене.

Теперь пользователю доступен сценарный прототип, который представляет собой определенное количество сцен, каждая из которых включает в себя следующие категории: персонажи, локации и артефакты. В дальнейшем данный прототип может использоваться для планирования этапов разработки, проверки плотности событий в сцене и для отслеживания развития событий.

Наша реализация – первый шаг на пути создания полноценного программного обеспечения, которое будет полезно при разработке крупных проектов.

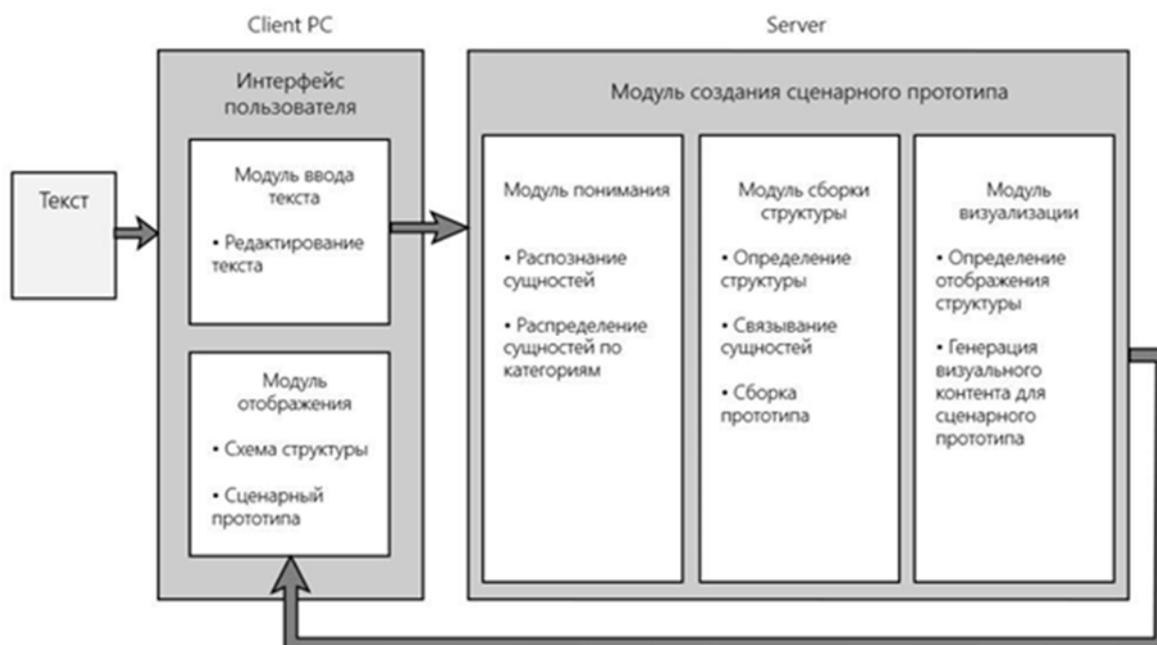


Рис. 2. Предлагаемая архитектура

ДАЛЬНЕЙШЕЕ РАЗВИТИЕ

На основе проведенного исследования была сформирована архитектура, которая обозначает векторы развития инструмента (Рис. 2). Инструмент должен будет иметь следующий функционал: принимать формализованный или неформализованный текст (из поля ввода или в виде файла); максимально точно определять и распределять сущности по категориям; определять структуру сценария (в том числе нелинейную) и отображать ее в визуальном виде (граф или Yarn); автоматически собирать сценарный прототип; добавлять в сценарный прототип визуальный контент.

Ветвление сюжета

Текущая версия прототипа поддерживает только линейный сюжет. В перспективе данный инструмент должен будет иметь возможность собирать разветвленную структуру. Данный подход может быть визуализирован в виде Yarn [9].

Генерация визуального контента

На данный момент предложено несколько вариантов подвязывания и генерации визуального контента. Инструмент SceneMaker [11] предлагает свое решение. Также полезен опыт разработки нейросети CRAFT [12].

Если внедрить данный функционал в создаваемый инструмент, то это позволит с наименьшими усилиями и временными затратами создавать визуализацию сценарного прототипа, что благоприятно скажется на тестировании повествования.

Автоматическая сборка сценарного прототипа

На данный момент работает полуавтоматическая генерация сценарного прототипа, т. е. генерация происходит с участием пользователя. В идеале сценарный прототип должен генерироваться автоматически, исходя из определенных шаблонов.

Данные для генерации уровня

Извлекаемые из текста сущности можно интерпретировать по-разному. В зависимости от контекста их можно разделить на разные категории. Одна из таких категорий – описание уровня. Пусть текст дает неполную информацию, но может задать основные правила, будь то относительное расположение улиц в городе или наличие особенного вида растений вдоль тропинок. Поэтому можно связать генерирование сценарного прототипа и создания на его основе правил для генерации уровня [14].

ЗАКЛЮЧЕНИЕ

Прототипирование и тестирование – важные этапы создания игр. Пренебрежение этими шагами может повысить риски и увеличить затраты, запланированные на проект.

Сама идея получения игрового прототипа на основе текстовых описаний, кажется трудной для реализации, но привлекательной в перспективе. Такой инструмент мог бы в комплексе решать вопрос прототипирования игр. Реализация такого инструмента также может означать повышение качества создаваемых игр, т. к. многие ошибки сценария можно будет заметить на ранних стадиях разработки.

СПИСОК ЛИТЕРАТУРЫ

1. Lee T. Designing game narrative. URL: <http://hitboxteam.com/designing-game-narrative>

2. *Ramadhan R. and Hendradjaya B.* Development of Game Testing Method for Measuring Game Quality // Proc. of Int. Conf. on Data and Software Engineering. 2014. № 7062694.
3. *Пшеничный И.* Чем отличается разработка игр от разработки «обычных» ИТ-проектов. URL: <https://vc.ru/4897-gamedev-it>
4. *Fullerton T., Swain C. and Hoffman S.S.* Game design workshop: a play-centric approach to creating innovative games. Burlington: Elsevier Inc., 2008. 470 p.
5. *Khatoon M., Aisha Banu W., Zohra A.A. and Chinthamani S.* Sentiment Analysis on Tweets // Advances in Intelligent Systems and Computing. 2019. V. 731. P. 717–724.
6. *Mignano M.* Game Development Pipeline: From Concept to Store. URL: <http://gamedevelopertips.com/game-development-pipeline/>
7. *Garneau P.-A.* Fourteen Forms of Fun. URL: https://www.gamasutra.com/view/feature/227531/fourteen_forms_of_fun/
8. *Liu S., Wu Y., Wei E., Liu M. and Liu Y.* Storyflow: Tracking the Evolution of Stories // IEEE Transactions on Visualization and Computer Graphics. 2013. No. 12. P. 2436–2445.
9. *Padia K., Bandara K.H. and Healey C.G.* Yarn: Generating Storyline Visualizations Using HTN Planning // Graphics Interface. 2018. P. 17–24.
10. *Liu Z.-Q. and Leung K.-M.* Script Visualization (ScriptViz): A Smart System that Makes Writing Fun // Soft Computing. 2006. V. 10. P. 34–40.
11. *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. and Roman L.* SceneMaker: Creative Technology for Digital Storytelling // Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering. 2016. P. 29–38.
12. *Gupta T., Schwenk D., Farhadi A., Hoiem D. and Kembhavi A.* Imagine This! Scripts to Compositions to Videos // Computer Vision and Pattern Recognition. 2018.
13. *Adams E.* The Designer's Notebook: Machinations, A New Way to Design Game Mechanics. URL: https://www.gamasutra.com/view/feature/176033/the_designers_notebook

14. Баранов В.С., Сергеев А.С., Кузуракова В.В., Ситдииков А.Г., Хафизов М.Р. Археологические объекты болгарского городища X–XV вв., как материал для создания виртуальной культурно-исторической реконструкции // Электронные библиотеки. 2015. Т. 18. №5. С. 269–282.

THE CONCEPT OF AUTOMATIC CREATION TOOL FOR COMPUTER GAME SCENARIO PROTOTYPE

G. F. Sahibgareeva¹, V. V. Kugurakova²

^{1,2}Higher School of Information Technologies and Intelligent Systems.
Kazan Federal University

¹gulnara.sahibgareeva@gmail.com, ²vlada.kugurakova@gmail.com

Abstract

The description of the architecture of the tool for generating a scenario prototype from the text outlined in this paper is described based on the existing solutions.

We formed software requirements and developed prototype of the tool that illustrating the basic principle of the user's work with the application.

Keywords: *game scripts, narrative design, scenario prototype, prototyping, game development, immersion, narrative*

REFERENCES

1. Lee T. Designing game narrative. URL: <http://hitboxteam.com/designing-game-narrative>
 2. Ramadhan R. and Hendradjaya B. Development of Game Testing Method for Measuring Game Quality // Proc. of Int. Conf. on Data and Software Engineering. 2014. № 7062694.
 3. Pshenichnyj I. Сhem otlichaetsya razrabotka igr ot razrabotki «obychnyh» IT-proektov. URL: <https://vc.ru/4897-gamedev-it>
 4. Fullerton T., Swain C. and Hoffman S.S. Game design workshop: a play-centric approach to creating innovative games. Burlington: Elsevier Inc., 2008. 470 p.
-

5. *Khatoon M., Aisha Banu W., Zohra A.A. and Chinthamani S.* Sentiment Analysis on Tweets // *Advances in Intelligent Systems and Computing*. 2019. V. 731. P. 717–724.
6. *Mignano M.* Game Development Pipeline: From Concept to Store. URL: <http://gamedevelopertips.com/game-development-pipeline/>
7. *Garneau P.-A.* Fourteen Forms of Fun. URL: https://www.gamasutra.com/view/feature/227531/fourteen_forms_of_fun/
8. *Liu S., Wu Y., Wei E., Liu M. and Liu Y.* Storyflow: Tracking the Evolution of Stories // *IEEE Transactions on Visualization and Computer Graphics*. 2013. No. 12. P. 2436–2445.
9. *Padia K., Bandara K.H. and Healey C.G.* Yarn: Generating Storyline Visualizations Using HTN Planning // *Graphics Interface*. 2018. P. 17–24.
10. *Liu Z.-Q. and Leung K.-M.* Script Visualization (ScriptViz): A Smart System that Makes Writing Fun // *Soft Computing*. 2006. V. 10. P. 34–40.
11. *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. and Roman L.* SceneMaker: Creative Technology for Digital Storytelling // *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*. 2016. P. 29–38.
12. *Gupta T., Schwenk D., Farhadi A., Hoiem D. and Kembhavi A.* Imagine This! Scripts to Compositions to Videos // *Computer Vision and Pattern Recognition*. 2018.
13. *Adams E.* The Designer's Notebook: Machinations, A New Way to Design Game Mechanics. URL: https://www.gamasutra.com/view/feature/176033/the_designers_notebook
14. *Baranov V.S., Sergeev A.S., Kugurakova V.V., Sitdikov A.G., Khafizov M.R.* Arheologicheskie ob"ekty bolgarskogo gorodishcha X–XV vv., kak material dlya sozdaniya virtual'noj kul'turno-istoricheskoy rekonstrukcii // *Elektronnye biblioteki*. 2015. T. 18. №5. S. 269–282.

СВЕДЕНИЯ ОБ АВТОРАХ



САХИБГАРЕЕВА Гульнара Фаритовна – бакалавр Высшей школы информационных технологий и интеллектуальных систем (ВШ ИТИС) Казанского (Приволжского) федерального университета (КФУ). Сфера интересов: игровая сценаристика, нарративный дизайн, изучение вопроса эффективности создания сценарного прототипа и возможности автоматизировать данный процесс.

Gulnara Faritovna SAHIBGAREEVA – Bachelor of Higher School ITIS. Sphere of interests: game scripts, narrative design, studying the effectiveness of creating a scenario prototype and the ability to automate this process.

email: gulnara.sahibgareeva42@gmail.com



КУГУРАКОВА Влада Владимировна – старший преподаватель ВШ ИТИС КФУ, руководитель лаборатории «SIM». Сфера интересов: разработка игр, иммерсивность виртуальных сред, интерпретация биосигналов человека, погруженного в виртуальную среду, методы трансформации нарративной идентичности.

Vlada Vladimirovna KUGURAKOVA – Senior Lecturer of Higher School of Information Technology and Intelligent Systems, Head of Laboratory «SIM». Sphere of interests: game development, immersivity of virtual environment, narrative identity.

email: vlada.kugurakova@gmail.com

Материал поступил в редакцию 1 июля 2018 года.