

# Разработка компонента генерации визуализации сценарного прототипа видеоигр

Г.Ф. Сахибгареева<sup>1</sup>, О.А. Бедрин<sup>1</sup>, В.В. Кугуракова<sup>1</sup>

<sup>1</sup> *Высшая школа информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета*

**Аннотация.** На данный момент в индустрии разработки игр существует эффективный способ прототипирования повествования – сценарный прототип. В своей работе мы сформулировали основания для разработки инструмента, который генерировал бы сценарный прототип из текстовых документов. Для оценки реалистичности поставленной цели мы исследовали существующие решения, опыт которых мы можем перенять в разработке своего инструмента. Мы представляем текущие результаты работы и очередное прототипное решение – генератор раскадровки, а также планы дальнейшего развития.

**Ключевые слова:** сценарный прототип, IDN, прототипирование интерактивного повествования, интерактивная раскадровка, генерация раскадровки, формализованный текст, NLP, серьезные игры, игровая документация, игровой прототип, разработка игр, компьютерные игры, видеоигры

## Development of a video games scenario prototype visualization generating component

G.F. Sahibgareeva<sup>1</sup>, O.A. Bedrin<sup>1</sup>, V.V.Kugurakova<sup>1</sup>

<sup>1</sup> *Higher School of ITIS, Kazan Federal University*

**Abstract.** Currently, the game development industry has an effective way of prototyping the story – a scenario prototype. We formulated in our work the basis for the development of a tool that would generate a scenario prototype from text documents. To assess the realism of the goal we examined existing solutions, the experience of which we can adopt in the development of our tool. We present the current results of the work and the next prototype solution – the storyboard generator, as well as plans for further development.

**Keywords:** scenario prototype, IDN, interactive storytelling prototyping, interactive storyboard, storyboard generation, formalized text, NLP, serious games, game documentation, game prototype, game development, computer games, video games

## 1. Введение

В сфере разработки игр, как и в любом ИТ-проекте, важно прототипировать каждый компонент, будь то интерфейс, механики<sup>1</sup> или визуализация, а также любые дизайнерские решения, связанные в том числе и с повествованием. Прототипирование призвано снизить риски и стоимость конечного продукта. Поэтому оно является одной из лучших и эффективных практик в сфере разработки информационных и мультимедийных приложений [1].

Игры – сложная система компонентов, объединенных с целью создания для игроков незабываемого опыта [2]. В формировании опыта немаловажную роль играет история, которую транслирует игра. История в играх – это не только текст, диалоги и кат-сцены<sup>2</sup>, но и, собственно, действия игрока в вымышленном мире и все средства на создание вымышленного мира – визуализация, анимация, звук.

Однако в данной работе история подразумевается в единственной форме – в виде текста. На этапе, предшествующем разработке, создается сценарий игры, который имеет не тот классический вид, который сформировался в кинематографе. Сценарий игры – это скорее серия документов, представленных в разнообразных форматах, которые описывают историю игры с разных точек зрения: какие персонажи есть в игре, о чем они говорят; какие есть квесты<sup>3</sup>, что из них должен извлечь игрок; какую глобальную историю повествует игра; какую часть истории игрок получает через кат-сцены, какие через общение с персонажами, а какие через особенный визуальный язык окружения.

Для проектирования повествования в играх используют разные подходы. Один из них – сценарный прототип, которому посвящено пристальное внимание авторов. В рамках работы представлен прогресс по разработке прототипа компонента визуализации инструмента генерации сценарного прототипа видеоигр.

В результате был проведен анализ предметной области, обзор существующих решений, уточняются требования к инструменту, формулируются требования к компоненту визуализации, приводится результат прототипирования компонента визуализации, формулируются планы на будущее развитие.

В разделе 2 описаны практики документирования повествования в разработке видеоигр. В разделе 3 описана продвинутая практика – сценарный прототип. В разделе 4 аргументируется польза генерации

---

<sup>1</sup> Игровая механика – правило игры или способ взаимодействия с игровым миром. Например: бег, прыжок, изучение предмета, указание цели, общение с персонажами и т.д.

<sup>2</sup> Кат-сцена – видеоролик, который прерывает игровой процесс. Может воспроизводиться в любой момент игры.

<sup>3</sup> Квест – задание или серия заданий для игрока. Одноименное название носит также жанр игр. Квест включает в себя цель и этапы достижения цели.

сценарного прототипа. В разделе 5 приведен анализ функционала инструментов, опыт которых полезен в разработке генератора. В разделах 6-9 приведен прогресс в разработке инструмента генерации сценарного прототипа в целом и компонента визуализации в частности. В разделе 10 – результаты прототипирования компонента визуализации. В разделе 11 – планы на будущее.

## **2. Форматы технической документации сценариста игр**

Сценарий в форме технической документации создает игровой сценарист. Существует несколько форматов, в которых он может представить историю игры для заказчика и команды разработчиков: текстовый документ, таблица, wiki-разметка, бумажный прототип, раскадровка.

### **Текстовый документ**

В формате текстового документа может быть представлен сценарий основной сюжетной ветки игры, побочных веток и кат-сцен, а также такие тексты, как энциклопедии, письма, заметки, встречающиеся в игре. Однако текст требует много времени и усилий на изучение.

Суть работы сценариста заключается в том, чтобы точно донести историю сначала до участников команды разработки и всех лиц, принимающих важные решения, касательно игры, а уж потом до игрока. Без понимания истории, которую должна донести игра, невозможно создать гармоничную связь между ее компонентами.

Эффективность восприятия текста сравнительно не высока, в отличие от видео и игр. Было доказано, что эффективность визуализации [3] и, тем более, погружения (или иммерсии) [4] значительно выше, чем эффективность трансляции информации через текст.

После изучения текстовой документации в работу сценариста вносятся правки. Любая правка может повлечь лавинообразный эффект изменений в сценарии. Тяжело устранить последствия этого эффекта в случае, если сценарий не визуализирован хотя бы в виде графовой структуры.

Текстовый формат устоялся в кинематографе, но в случае разработки игр, представлять историю исключительно в текстовом виде – большой риск. При создании игр необходимы более эффективные методы создания, хранения и передачи документации. Лучше всего текст усиливает визуализация, а также возможность погружения в повествование.

### **Таблицы**

Большую часть того контента<sup>4</sup>, который в дальнейшем пойдет в разработку соответствующим специалистам, можно представить в табличном виде. На примере воображаемого квеста проведем анализ принципа работы таблиц (см. рис. 1).

---

<sup>4</sup> Контент – наполнение игры. Различают аудио-контент, визуальный и текстовый контент.

№	Название	Персонаж	Реплика персонажа	1 этап	2 этап	3 этап	4 этап	5 этап
1	Именинный торт	Инна	“Сегодня у Оли День рождения. Давай испечем ей торт.”	Купить продукты	Выбрать рецепт	Испечь коржи	Собрать торт	Подарить торт

Рис. 1. Пример таблицы для квеста

В данном примере мы видим одно из возможных представлений квеста для мобильной игры. Для каждого квеста в примере выделены следующие параметры: идентификационный номер, название, имя персонажа, который даст игроку квест, его реплика для дачи квеста, этапы реализации квеста действиями игрока.

Разберем необходимость данных элементов:

- номер квеста необходим для ускорения коммуникаций между участниками команды разработки;
- название квеста также необходимо для коммуникации, но помимо этого оно отображается в самой игре;
- игрок должен каким-то образом получить квест, в данном случае квест выдает конкретный персонаж, который с этой целью произносит определенную реплику;
- для завершения квеста игрок должен пройти конечное число этапов.

Табличное решение может быть расширено по количеству параметров и строк до такого объема, который будет необходим в разработке. Табличное решение эффективнее текстового, т.к. наглядно представляет информацию в структурированном виде. Заметить в таком представлении ошибки и исправить их значительно проще, чем в текстовом виде.

Таблицы подходят не только для квестов. Их можно создавать под любые задачи, указывать в них необходимые для работы тексты, анимации и звук. Таблицы – хороший технический инструмент, призванный структурировать информацию и организовать работу команды. Однако он не подходит для презентации истории игры и последующего тестирования и оценки, т.к. таблица не имеет ничего общего с игровым процессом.

### **Wiki-разметка**

Иерархия, поддерживаемая wiki-разметкой позволяет создать целую систему документов, в которой будет храниться целостное представление о внутриигровом мире. Принцип работы wiki-разметки в следующем: каждый текстовый документ содержит в себе понятия (слова, фразы), суть которых раскрывается в других документах, на которые можно попасть по ссылке из текущего документа.

Плюс wiki-разметки в том, что он позволяет поддерживать лор<sup>5</sup> в упорядоченном систематизированном виде.

Однако, это все те же текстовые документы, которые разбиты на серию веб-страниц. Команде разработки и всем заинтересованным лицам придется потратить время на изучение всей предоставленной информации.

### **Бумажный прототип**

Бумажный прототип – это отличное решение как для прототипирования игровых механик, так и для прототипирования игрового повествования [5]. Если сценарист смог провести слушателей и участников через историю с помощью подручных средств (бумажных карточек, фишек, кубиков, условных персонажей), и всем история понравилась, значит, теоретически, история должна понравиться в итоговом виде игры. Говоря о бумажном прототипировании истории, важно отметить, что эта та же игра, но лишенная реализованных правил и механик, упрощенная настолько, чтобы участники окунулись лишь в выдуманный мир, но не углублялись в процесс игры [6].

С одной стороны – это плохо с точки зрения нарративного дизайна<sup>6</sup>. С другой стороны, в том же нарративном дизайне есть серия оценочных средств, часть которых можно проверить с помощью простейшего бумажного прототипа. Эти понятия – погружение, темпоритм и эмпатия.

Погружение [7] – состояние, в котором игрок забывает о времени и реальности, проникается историей игры, осознает её важность для себя. Погружение – это знакомое состояние, в котором мы обнаруживаем себя, когда читаем книги и смотрим фильмы. Это чувство, из которого мы возвращаемся как ото сна.

Темпоритм [8] – частота и плотность подачи истории. Игрок должен знакомиться с миром игры постепенно. Это важно по двум причинам: первое – знакомство с чем-то новым должно происходить поэтапно, второе – должна сохраняться интрига, которая удержит игрока в процессе изучения истории. Темпоритм регулируется исключительно с помощью тестирования, и бумажный прототип позволяет это сделать.

Эмпатия [9] – это важное чувство, которое рождается у игрока по отношению к персонажу, за которого он играет, а также по отношению к другим персонажам. Эмпатию вызывают знакомые персонажи, которыми хотелось бы стать, которых игрок готов понять и сопереживать им. Каждый находит в персонажах историй что-то своё. Но бывают неудачные персонажи, которые либо сами по себе очень слабые, плохо продуманы, либо они не соответствуют ожиданиям целевой аудитории, текущему

---

<sup>5</sup> Lore (англ. знания, предания) — термин, обозначающий "предысторию" сюжета мира, а также знание о мире.

<sup>6</sup> Нарративный дизайн – это искусство рассказывать историю в видеоигре всеми доступными средствами.

времени. Данный параметр также проверяется только на практике в ходе тестирования.

Таким образом бумажный прототип является эффективным и дешевым инструментом тестирования игровой истории на ранних стадиях [10]. Для бумажного прототипа нужно присутствие автора или инструкция по проведению сессии тестирования, а также минимальные артефакты, созданные с помощью подручных средств, например, необходимые по сюжету предметы – меч, монетки, книги.

Минус бумажного прототипа в том, что каждый раз при необходимости провести тестирование необходимо организовать встречу и задействовать сценариста.

### **Раскадровка**

Один из способов визуализации сценария – это раскадровка. Раскадровка – это серия зарисовок, представляющих собой кадры будущего произведения. Их используют для создания фильмов, мультфильмов, рекламы, а также любого аудиовизуального произведения, в котором необходимо настроить положение камеры и расстановку объектов.

В нашем представлении раскадровку сценария игры можно назвать *интерактивной раскадровкой*. По своей форме интерактивная раскадровка является серией картинок, содержащих в себе визуализацию локации<sup>7</sup>, обязательных персонажей, их положение в пространстве, их имена и реплики. По сути, это снимки того, что будет видеть игрок в процессе игры. Данное представление очень напоминает существующий игровой жанр – визуальная новелла.

Визуальная новелла – это игра, в которой можно совершать выбор (как действия, так и выбор реплик в диалогах), представляющая собой набор картинок-фонов, картинок-персонажей и текста. Соответственно, сценарий игры, представленный в виде интерактивной раскадровки, это уже игра, минимальный продукт, который можно представить всем заинтересованным лицам и получить отзыв об истории игры.

Помимо визуализации интерактивная раскадровка, как и визуальная новелла, может включать в себя текст, звуки и музыку, простейшие анимации.

Интерактивная раскадровка – это один из лучших способов представления повествования игры. Визуализация помогает представить атмосферу игры, пусть черновую, в виде серии скетчей<sup>8</sup>, или собранную из хаотичных картинок и фотографий из Интернета, но целостную и законченную. На разработку раскадровки уйдет время, однако она будет значительно эффективнее всех способов подачи истории, описанных выше. Более того,

---

<sup>7</sup> Локация – место, в котором происходят события игры. Чаще всего являются изолированными участками, которые разделяются между собой экранами загрузки. Примеры: лес, таверна, космический корабль и др.

<sup>8</sup> Скетч – зарисовка, которую делает художник в начале своей работы.

раскадровку можно разослать всем заинтересованным лицам, для её изучения не требуется личного присутствия сценариста. После изучения, а точнее, прохождения интерактивной раскадровки, можно предложить заполнить анкету и сформировать по итогу статистические данные.

### **3. Сценарный прототип**

Сценарный прототип – это интерактивный прототип повествования игры без игровых механик.

К слову, интерактивная раскадровка – это вариант сценарного прототипа. Существует несколько рекомендаций к созданию повествовательного прототипа: он должен соответствовать жанру разрабатываемой игры, иметь заглушки на геймплей и игровые механики, должен быть разработан на целевом движке<sup>9</sup>.

Соответствие сценарного прототипа жанру, а также его разработка на целевом движке – рекомендации, устраняющие возможные трудности с имплементацией сценарного прототипа в игру. К тому же, всегда проще нарастить функционал сценарного прототипа, чем разрабатывать игру с нуля. Иметь готовый сценарный прототип для каких-то игр означает иметь одну из итераций разработки. Таким образом, сценарный прототип сокращает объём работ в течении этапа разработки.

Заглушки на геймплей и механики подразумевают следующее: сценарный прототип создается для отработки гипотез и проверки качества повествования игры, но не её игровой составляющей. Отделять игровой процесс от истории – плохо, но на каком-то этапе это возможно и, на наш взгляд, эффективно. Понравившаяся аудитории история – лучше, чем история, которая не понравилась из-за сырых механик, которые перетягивают внимание на себя. Сценарный прототип – это попытка на время изолировать историю от игры, чтобы протестировать её на несколько иные характеристики, такие как погружение, темпоритм и эмпатия.

Сценарный прототип – это решение, позволяющее сэкономить проектные ресурсы, протестировать историю до начала разработки контентной составляющей игры.

Сценарный прототип применим как для небольших, так и для крупных проектов, как для разработки с нуля, так и для доработки дополнительных блоков существующей игры.

Разработка сценарного прототипа на основе работы сценариста, представленной в виде технической документации – работа нарративного дизайнера<sup>10</sup>.

---

<sup>9</sup> Игровой движок – программное обеспечение для создания, сборки и настройки компонентов игры: визуализации, звука, механик, текста.

<sup>10</sup> Нарративный дизайнер – специализация гейм-дизайнера; дизайнер по имплементации истории в игру.

#### **4. Генерация сценарного прототипа**

Генерация сценарного прототипа вместо ручной сборки, поможет решить две задачи: разгрузить нарративного дизайнера и снизить порог вхождения игровых сценаристов в индустрию.

Первая задача связана с тем, что на предварительном этапе разработки нарративный дизайнер занимается не только тем, что собирает сценарный прототип. Кроме этого, ему необходимо создать концепцию внедрения повествования в игру. Если снять с нарративного дизайнера одну задачу, он сможет сконцентрироваться на другой.

Вторая задача также частично связана с разгрузкой других специалистов. Например, если тот же сценарный прототип будет сгенерирован, игровой сценарист сможет самостоятельно находить какие-то нестыковки в сценарии и исправлять их раньше. Помимо этого у сценариста освободится время на изучение игрового движка и программирования, что является немаловажным критерием выбора кадров в игровой индустрии.

Таким образом мы формулируем ценность нашего инструмента: генератор сценарных прототипов может сэкономить все возможные ресурсы, которые есть на проекте по разработке игр. Более того, мы видим применение нашего инструмента не только в игровой индустрии, но и в индустрии разработки serious games<sup>11</sup>, тренажеров и симуляторов.

#### **5. Существующие решения**

На данный момент существует множество решений для генерации визуального контента на основе текста на естественном языке, которые отличаются целями и подходами. Так, например, в работе 2016 года [11], рассматриваются 26 реализованных инструментов. Инструменты в данной работе делятся на несколько типов: преобразование текста в серию картинок (Story Picturing Engine), преобразование текста в серию ассоциированных картинок (Text-to-Picture Synthesis System), преобразование текста в набор статичных сцен (TEXT-TO-SCENE), преобразование текста в анимацию, в том числе с учетом положения объектов в пространстве (TEXT-TO-ANIMATION).

Точность генерации визуализации во многом зависит от степени развития NLP и нейросетей, а также вычислительных мощностей компьютера. Однако уже сейчас есть приемлемые результаты генерации, с которыми можно работать в индустрии кинематографа [12] и мультипликации [13], [14].

В ходе проведенного исследования был выделен ряд инструментов для визуализации, опыт которых есть намерение перенять в собственной разработке.

---

<sup>11</sup> Serious games – интерактивное приложение, использующее геймификацию в своих целях: обучение, тренировка, исследование, экзаменация, маркетинг.



ScriptViz [15] – позволяет визуализировать сценарий в виде трехмерной сцены, содержимое которой генерируется в реальном времени. Пользователь вводит текст своего сценария, учитывая следующие ограничения: текст должен быть написан на английском, иметь понятные формулировки и хорошую структуру.

SceneMaker [16] – закрытая на данный момент разработка, цель которой заключалась в автоматизированной генерации ряда артефактов для кинопроизводства. Генерация сцен из сценария планировалась с точностью до мимики виртуальных актеров, освещения и аудиосопровождения.

CRAFT [13] – в качестве инструмента, способного на основе описательного текста подобрать подходящий анимационный материал, интересна нейросеть CRAFT. Принцип её работы состоит в следующем: на основе 25184 фрагментов из мультсериала Flintstones<sup>12</sup>, нейросеть собирает новые серии мультфильма по текстовому описанию. В собранных сценах, ещё встречаются ошибки, такие, как неправильное наложение объектов или неправильный выбор анимаций.

Эти инструменты подтверждают нашу гипотезу о том, что техническую документацию игровых сценаристов возможно визуализировать программным путем.

Однако ключевое отличие прототипирования повествования игр заключается в реализации возможности интерактивного взаимодействия, а также вариативности сюжета, чего не подразумевают вышеупомянутые инструменты для кинематографа и мультипликации.

Мы нашли инструменты, применимые и к играм.

StoryFlow [17] – 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. Данная структура имеет возможность визуализировать вариативность в происходящих событиях, что крайне полезно для красивой и понятной визуализации игровых событий, взаимодействия персонажей и определения временных промежутков.

Machination Tool [18] – веб-платформа для проектирования, балансирования и моделирования игровых систем. Данный инструмент позволяет визуализировать и симулировать игровые системы в виде балансных диаграмм. Данное решение мы видим одним из элементов уже не сценарного, а скорее игрового прототипа.

Еще один инструмент интересен с точки зрения возможностей визуализации и агрегатора различных программных решений.

---

<sup>12</sup> Флинтстоуны (англ. The Flintstones) — американский комедийный мультсериал о жизни в каменном веке, транслировавшийся на телеканале ABC с 1960 по 1966 год.

Orange3 [19] – это программный пакет визуального программирования на основе компонентов для визуализации данных, машинного обучения и интеллектуального анализа данных. В программу включены сотни готовых нод, каждая из которых отвечает за свою часть работы: различные визуализаторы (схемы, графики, таблицы), алгоритмы обработки и препроцессинга текста, готовые к обучению и работе нейросети. Помимо этого разработчики предоставляют пользователям возможность реализовать собственный функционал в виде программ на Python, для которой есть отдельный нод.

Инструмент, который вдохновил нас на очередную разработку – это Storyboarder.

Storyboarder – инструмент для создания раскадровок от компании Wonder Unit [14]. В данном инструменте есть функционал генерации кадров на основе введенных текстовых запросов. Однако, стоит отметить, что Storyboarder понимает (пока) лишь ограниченный словарь слов и некоторые альтернативы для слов из этого списка, которые находятся в документе `alternativeValue`.

## **6. Видение инструмента**

Несмотря на то, что проделано всеобъемлющее исследование по анализу подходов к генерации визуализации, мы не нашли предложения по генерации полноценного сценарных прототипов игр в какой бы то ни было форме.

Наша основная цель – это разработка инструмента генерации сценарного прототипа игры на основе технической документации игрового сценариста. Процесс работы с повествовательной составляющей игры с применением нашего инструмента видится следующим:

- 1) написание сценария игры в виде обычного текста на естественном языке в форме технической документации (текст, таблицы, wiki-разметка);
- 2) генерация сценарного прототипа на основе документации;
- 3) презентация и обсуждение сценарного прототипа внутри команды, с заказчиком и тестирование на фокус-группе;
- 4) редактирование сценарного прототипа.

Был составлен следующий план разработки такого инструмента:

- 1) визуализация структуры игрового сценария, представленного в виде текста на естественном языке (далее называемый ТЕХТ) в виде графов или yarn-представлений;

- 2) автоматическая сборка интерактивной раскадровки (для ограниченного количества жанров, в виде визуальной новеллы или point-and-click<sup>13</sup> игры) сценария игры из текста;
- 3) создание алгоритма для генерации интерактивного сценарного прототипа из текста в игровом движке Unity;
- 4) автоматическая генерация диаграммы баланса Machination на основе текста;
- 5) автоматическая сборка игрового прототипа игры из текста.

Весь перечисленный функционал мы видим в рамках целостного приложения, на вход которому поступает серия текстовых документов, а на выходе получается проект, который объединяет в себе сгенерированный сценарный и игровой прототипы игры.

На картинке (см. рис. 2) представлена схема инструмента, которую мы разработали для последующей реализации.

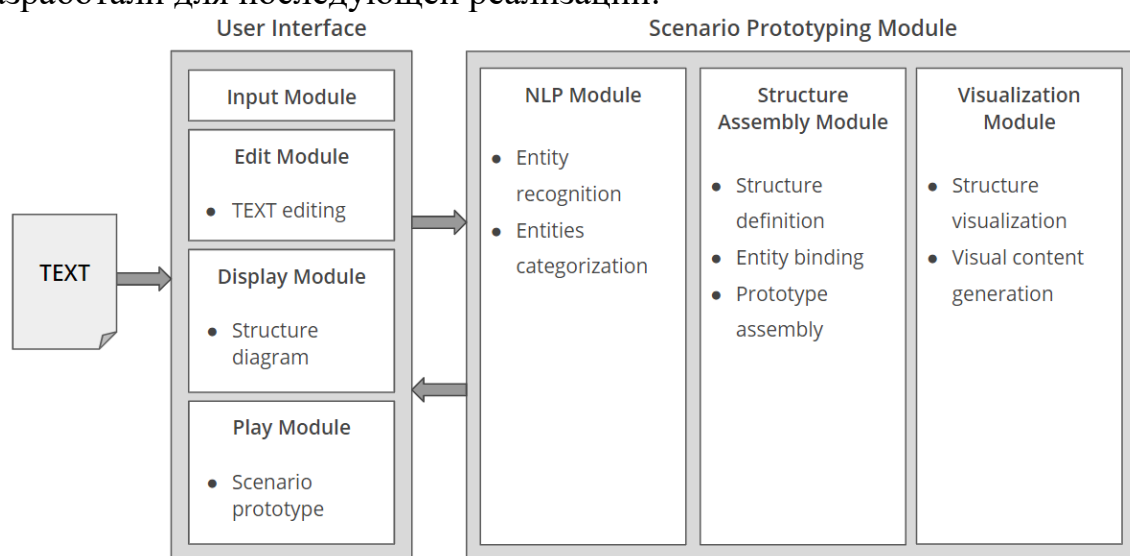


Рис. 2. Схема работы инструмента

На вход принимается сценарий (ТЕХТ), который написан на естественном языке. Мы допускаем, что данный документ будет удовлетворять некоторым правилам форматирования для лучшего распознавания сущностей. Edit Module позволяет редактировать текстовый документ.

В режиме Display Module пользователь знакомится с результатом визуализации структуры.

В режиме Play Module пользователь запускает сценарный прототип, в который можно “сыграть”.

Модуль генерации сценарного прототипа включает в себя три части, каждая из которых последовательно обрабатывает результат работы предыдущих модулей:

<sup>13</sup> Point-and-click – жанр игр, суть которых заключается в поиске активных областей и клике по ним мышкой. Активные области представляют собой возможности для взаимодействия с внутриигровыми объектами.

- 1) модуль NLP распознает необходимые сущности и связи между ними;
- 2) модуль построения структуры устанавливает целостную структуру сценария, связывает сущности между собой и строит структуру сценарного прототипа;
- 3) модуль визуализации подбирает необходимые трехмерные модели и распределяет их по структуре сценарного прототипа. Далее модуль отправляет полученную визуализацию сценарного прототипа, а также визуализированную структуру сценария, например, в виде Yarn, в Play Module.

## 7. Прделанная работа

Решению задачи создания генератора сценарных и игровых прототипов посвящена уже целая “серия” наших разработок. В работе [20] был описан собственный подход к созданию генератора сценарного прототипа и пилотная реализация инструмента для создания сцен на основе извлеченных из текста сущностей. В [21] представлена (см. рис. 3) программа для Orange 3, которая принимает на вход формализованный текст, а на выходе показывает различную визуализацию структуры: геометрическую связанность локаций, в каких локациях появляется персонаж, какие реплики произносятся в этих сценах и т.д..

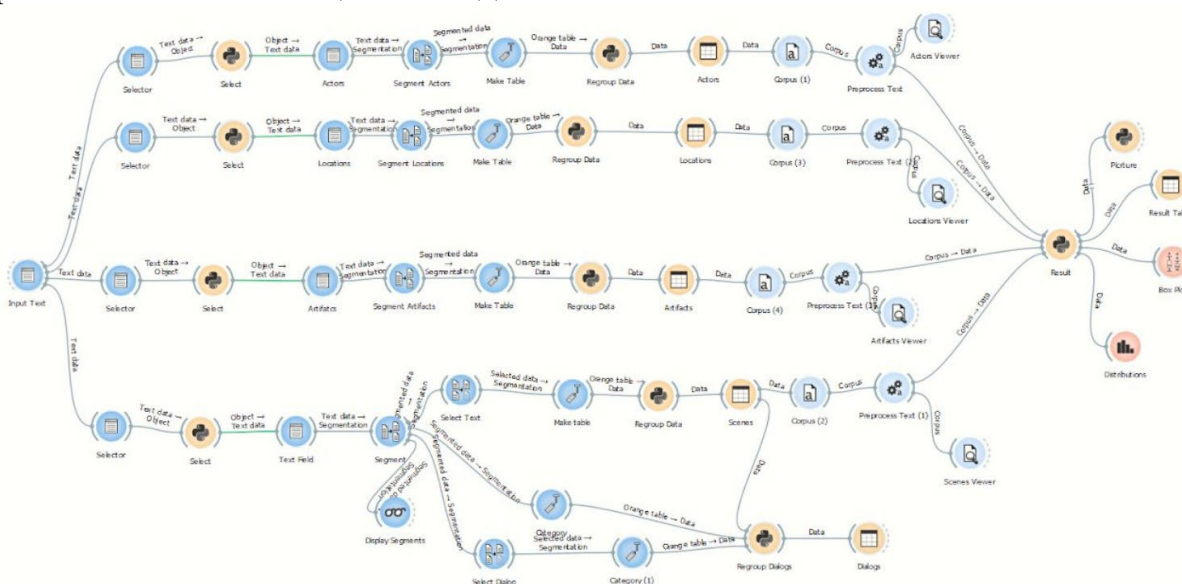


Рис. 3. Нодовая система генерации визуализации сценария

Другая работа [22] связана с решением задачи извлечения информации о постановке кадра из текста. Для этого создана система анализа текста и вычисления необходимых настроек камеры из контекста. Работа находится на раннем этапе: воссоздается функционал текстового распознавания кадра, аналогичный Storyboarder (см. рис. 4).



Рис. 4. Извлечение информации о кадре из текста

## 8. Представление текущего этапа работ

В данной статье представлен прогресс по второму этапу разработки – автоматической сборке раскадровки сценария игры из текста.

Как говорилось ранее, мы рассматриваем интерактивную раскадровку в качестве одной из форм сценарного прототипа. Интерактивная раскадровка имеет достаточно сложную структуру, которую необходимо извлечь из аналогично структурированного текста. Пока ещё не решён вопрос создания и извлечения вариативности из текста, но задача генерации не интерактивной раскадровки реализована.

Вышеописанный инструмент Storyboarder использован в качестве платформы для экспериментов. Принцип работы состоит в следующем: на вход подаётся текст на английском языке, предложения которого представляют собой описание каждой из сцен. В каждом предложении есть информация о том, кто присутствует в сцене и что он делает. Затем извлекаются из предложений сущности (персонаж и его действие). Результат обрабатывается и отправляется на вход Storyboarder, где генерируются кадры, которые собираются в раскадровку для оценки результата.

Рассмотрим подробности технической реализации и результаты эксперимента.

## 9. Техническая реализация текущего этапа

Возможность генерации элементов раскадровки по заданному вектору слов присутствует только в слепке коммита `abbf5f24` открытого официального репозитория `Storyboarder` [23]. Для того чтобы была возможность интегрировать `Storyboarder` в автоматический конвейер, где вход – это текст, находящийся в управлении другого программного компонента, а выход – это установленные в `Storyboarder` сгенерированные согласно тексту элементы раскадровки, был изолирован функционал продукта, который позволяет, вызывая процедуру генерации с аргументом в виде исходного преобразованного текста, инициировать все необходимые механизмы для получения кадров раскадровки. Также был внедрен микро веб-сервер `express`, который принимает на `loopback` интерфейсе запрос с параметром в виде входного текста, передаёт данный параметр в изолированный функционал и, в качестве ответа, отправляет тот текст, что получил, для того чтобы отправитель удостоверился в успешности операции генерации.

В рамках описания разработки вышеупомянутого функционала была использована терминология, содержащая элементы лингвистики и машинного обучения.

Исходный текст преобразуется в корпус [24] – это подобранная и обработанная по определенным правилам совокупность текстов, используемых в качестве базы для исследования языка. Они используются для статистического анализа и проверки статистических гипотез, подтверждения лингвистических правил в данном языке.

Так как машинное обучение в большинстве своём имеет дело с функционалами (любыми функциями, где образ является числом или множеством чисел), для обработки текста, его приходится векторизировать, а именно преобразовывать в векторы чисел по определенным правилам, таким чтобы потеря информации, заложенной в тексте, была минимальной. Также одним из этапов обработки текста, является выделение стоп-слов – таких слов, которые в рамках генеральной совокупности играют малую роль в исследовании свойств языка. Например, в английском языке это будут слова: `a`, `an`, `an`, `is`, `are`, `etc`. А в русском языке таковыми будут являться: `же`, `то`, `бы` и т.д.

Для того, чтобы `Storyboarder` корректно сгенерировал раскадровку, входной текст необходимо обработать. Первым этапом будет являться токенизация, а именно преобразование текста в корпус в виде массива предложений, также каждое предложение необходимо разбить по словам и убрать пунктуацию. Во второй этап входит фильтрация всех слов в корпусе, в результате которого корпус будет очищен от стоп-слов. Далее стоит задача распознавания имён собственных и преобразование их в обезличенную

женскую или мужскую сущность. Это необходимо сделать, так как Storyboarder не может различать имен, но может различать пол сущностей, преобразуя его в кадр в модель мужчины или женщины.

После проведения токенизации, во время этапа фильтрации, были использованы стоп-слова, собранные в приложении к работе [25]. Процесс фильтрации происходил итеративно, слова сравнивались посимвольно полно. На следующем, более интересном этапе алгоритма, необходимо было построить классификатор, который бы распознавал в тексте имена собственные и определял бы их пол.

В процессе разработки в качестве модели был выбран мультиклассовый наивный байесовский классификатор, так как закономерность выбора имен в обществе отсутствует и имена даются независимо друг от друга. Имена для обучения классификатора были взяты из публичных данных Службы социального обеспечения США [26]. Для каждого из имени в обучающей выборке происходила векторизация в символьные биграммы, а также происходил частотный анализ, исходя из которого был виден процент использования имени в качестве женского и мужского. Так как обучение наивного байесовского классификатора, это лишь вычисление независимых вероятностей, произведение которых лежит в знаменателе формулы по теореме Байеса [27], трата времени на собственно сам процесс, была минимальной.

Итак, обученный байесовский классификатор отвечал нулём, если имя, пришедшее на вход, было женским и единицей, если мужским. Используя операцию классификации, каждый раз, когда в тексте находилось имя собственное, оно заменялось на слово man или woman в зависимости от пола, который присваивал ему обученный классификатор. Данные обезличенные слова были выбраны как наиболее понятные для парсера предложений Storyboarder.

## 10. Результат работы инструмента

Стоит отметить, что текст на естественном языке был написан с оговоркой, что Storyboarder понимает ограниченный список слов. Для нас было важно, чтобы в сцене появлялся персонаж с возможной в рамках Storyboarder анимацией.

Положение камеры задаются вручную последовательностью параметров, которые понимает Storyboarder.

Итак, на вход мы подаем следующий текст:

Bob is walking.

Alice says hi.

Bob is walking and looking back.

Bob is walking.

Alice hangs one arm.

Alice crossing hands.

Далее, в зависимости от контекста предложения, мы самостоятельно формируем необходимые настройки камеры. Настройки для каждого предложения хранятся в отдельном файле:

1) looking forward, medium long, single person, centered, left angle, eye level, long lens, light, frontrightlit, outside;

2) looking forward, medium long, single person, centered, right angle, eye level, long lens, light, frontrightlit, outside.

Список процессов преобразования, которые осуществляются над текстом:

1. Токенизация с очищением от пунктуации и стоп-слов.

2. Распознавание и обезличивание имен собственных путем замены на слова man/woman в зависимости от пола имени с помощью наивного байесовского классификатора.

Полученный после преобразования текст дополняется настройками камер с помощью конкатенации строк.

Таким образом формируется результат, подающийся на вход в Storyboarder:

1) 'man walking, looking forward, medium long, single person, centered, left angle, eye level, long lens, light, frontrightlit, outside';

2) 'woman say hi, looking forward, medium long, single person, centered, right angle, eye level, long lens, light, frontrightlit, outside';

3) 'man walk look back, looking forward, medium long, single person, centered, left angle, eye level, long lens, light, frontrightlit, outside';

4) 'man walking, looking forward, medium long, single person, centered, left angle, eye level, long lens, light, frontrightlit, outside';

5) 'woman hang one arm, looking forward, medium long, single person, centered, left angle, eye level, long lens, light, frontrightlit, outside';

6) 'woman crossing hands, looking forward, medium long, single person, centered, left angle, eye level, long lens, light, frontrightlit, outside'.

Storyboarder автоматически генерирует кадры для раскадровки на основании полученных данных. Для того, чтобы кадры стали частью раскадровки их нужно добавить в сторилайн вручную (см. рис. 5).





Рис. 5. Раскадровка, полученная в результате

Как видим, возможности для генерации комикса в Storyboarder позволяют использовать его лишь в случаях с подготовленным текстом. Расширение функционала и корпуса Storyboarder позволит эффективнее работать с визуализацией натурального текста, однако мы видим дальнейшую разработку в рамках собственного приложения.

## 11. Будущие планы

В ходе обсуждения результатов разработки мы пришли к некоторым выводам, которые сформировали наши будущие планы и задачи:

1. Извлечение информации о постановке кадра из текста.
2. Разметка и форматирование технической документации.
3. Создание интерактивной раскадровки.
4. Генерация визуализации.

Теперь хотелось бы описать каждую задачу подробнее.

Нереализованной осталась задача постановки кадра в зависимости от контекста повествования. Мы считаем этот вопрос важным с точки зрения автоматизации генерации визуализации сценария, и уже начали разработку в этом направлении.

Мы не касались темы классической разметки сценарных документов для не интерактивных произведений и особенностей сценариев для интерактивных произведений. Данному вопросу мы планируем посвятить время в дальнейшем. Решение вопроса разметки и форматирования документов с нелинейным повествованием поможет решить следующую задачу, связанную с интерактивностью раскадровки.

Раскадровки хороши в кинематографе и анимации, а также в играх, например, в создании кат-сцен. Главное изменение, которое мы хотим внести в покадровое представление игрового повествования, это

возможность переключаться с помощью взаимодействия с игровой действительностью. Интерактивная раскадровка, отдаленно напоминает нам визуальную новеллу или игры жанра point-and-click. В игре повествование движется за счет действий игрока. Исключая из прототипа влияние геймплея и выбор действий, мы лишаем сценаристов возможности тестирования повествования в условиях реальной разработки. В сценарном прототипе интерактив может быть заменен геймплейными заглушками, которые однозначно трактуют одну из веток развития событий. С таким инструментом игровой сценарист и нарративный дизайнер смогут заранее “отыграть” сюжет игры, показать его разработчикам, заказчику и пользователям. Возможность оценки и тестирования сценарного прототипа поможет оценить как классические для линейного повествования, так и характерные для интерактивного повествования игр черты: погружение, темпоритм, формирование эмпатии. Таким образом, следующей задачей мы ставим приведение раскадровки к интерактивному виду.

Вопрос о генерации визуализации по текстовому описанию связан с тем, что необходимо заранее создать необходимые трехмерные объекты, анимации для них, логику сборки сцены, непротиворечивые правила взаимодействия между объектами. При этом каждая из перечисленных единиц сборки должна иметь свойство, схожее со свойством вектора в базисе линейного пространства. Другими словами, количество созданных единиц должно быть минимальным при условии, что вариативность генерируемых сцен должна быть максимальной. Данный момент необходимо учитывать в последующих разработках.

После усовершенствования прототипа планируется продолжить тестирование, а также же сформулировать метрики, которые покажут, насколько генерация сценарного прототипа эффективнее ручного прототипирования.

## **12. Заключение**

В работе введен термин сценарного прототипа и описано авторское представление о нём. Помимо этого была сформулирована актуальность инструмента генерации сценарного прототипа.

В работе приведен анализ инструментов, которые вдохновили авторов. Существование этих инструментов говорит о том, что разработка инструмента генерации сценарного прототипа возможна.

Данный факт подтверждают примеры инструментов коллег, которые были разработаны с участием авторов статьи.

В качестве эксперимента был разработан прототип компонента визуализации инструмента генерации сценарного прототипа. Прототип генерирует раскадровку на основе текста. Компонент призван сэкономить время на визуализацию игрового повествования.

В результате проделанной работы были детализированы требования к инструменту генерации. Размышления на тему развития инструмента привели к формированию ряда задач для следующей итерации прототипирования.

Данная разработка интересна в рамках использования для разработки не только игр, но и serious games, симуляторов и виртуальных обучающих тренажеров, которым также посвящено наше пристальное внимание (см. например, [28], [29]).

### **Благодарности**

Работа выполнена в соответствии с государственной программой повышения конкурентоспособности Казанского федерального университета.

The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

### **Литература**

1. Al-Husseini Kh., Obaid A. Usage of prototyping in software testing // Multi-Knowledge Electronic Comprehensive Journal For Education And Science Publications. — 2018. — V. 14. — P. 1–15.
2. Schell J. The Art of Game Design: A Deck of Lenses, Second Edition. — CRC Press. — 2014. — 594 p.
3. Davies D., Bathurst D., Bathurst R. The Telling Image: The Changing Balance between Pictures and Words in a Technological Age // Technology and Culture. — The Johns Hopkins University Press and the Society for the History of Technology Stable. — 1992. — V. 33. — No 4. — P. 845–846.
4. Raja D., Bowman D., Lucas J., North C. Exploring the benefits of immersion in abstract information visualization // Proc. of the 8th Int'l Immersive Projection Technology Workshop. — 2004.
5. Koenitz H., Dubbelman T., Knoller N., Roth C., Haahr M., Sezen D., Sezen T. Card-Based Methods in Interactive Narrative Prototyping // ICIDS 2018, Proceedings International Conference on Interactive Digital Storytelling. — Springer Nature Switzerland AG. — 2018. — P. 552–555.
6. Antonisse J. Paper Tales: A Guide to Narrative Prototyping [Video] // Game Developer Conference. — 2014. — URL: <https://www.gdcvault.com/play/1020509/Paper-Tales-A-Guide-to> (дата обращения: 09/04/2020).
7. Cairns P., Cox A., Nordin I. Immersion in Digital Games: Review of Gaming Experience Research // Handbook of Digital Games. — Wiley-IEEE Press. — 2014. — P. 337–361.
8. Newman J. Videogames // Routledge. — 2014. — P. 198.
9. Blanchard L. Creating empathy in video games. [MSC Thesis] // The University of Dublin. — 2016. — 42 p.

10. Koivisto E., Suomela R. Using prototypes in early pervasive game development // Sandbox '07: Proc. of the 2007 ACM SIGGRAPH symposium on Video games. — Association for Computing Machinery. — 2007. — P. 149–156.
11. Hassani K., Lee W.-S. Visualizing Natural Language Descriptions: A Survey // ACM Computing Surveys — Association for Computing Machinery — 2016. — V. 49. — No 1.
12. RivetAI. — URL: <https://www.rivetai.com/> (дата обращения: 09/04/2020).
13. Gupta T., Schwenk D., Farhadi A., Hoiem D., Kembhavi A. Imagine This! Scripts to Compositions to Videos // Cornell University. — 2018.
14. Storyboarder. // Wonder Unit. — URL: <https://wonderunit.com/storyboarder/> (дата обращения: 09/04/2020).
15. 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. — P. 34–40.
16. 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 // Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering. — Springer Nature Switzerland AG. — 2016. — P. 29–38.
17. 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.
18. Adams E., Joris D. The Designer's Notebook: Machinations, A New Way to Design Game Mechanics — URL: [https://www.gamasutra.com/view/feature/176033/the\\_designers\\_notebook](https://www.gamasutra.com/view/feature/176033/the_designers_notebook) (дата обращения: 09/04/2020).
19. Orange 3. — URL: <https://orange.biolab.si/> (дата обращения: 09/04/2020).
20. Сахибгареева Г.Ф., Кугуракова В.В. Концепт инструмента автоматического создания сценарного прототипа компьютерной игры // Электронные библиотеки. — 2018. — Т. 21. — № 3-4. — С. 235–249.
21. 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.
22. Астафьев А. Разработка инструмента для сборки сцен по тегам [рукопись] – 2019. – 31 с.
23. Storyboarder / Исходный код проекта // GitHub. — URL: <https://github.com/wonderunit/storyboarder> (дата обращения: 09/04/2020).

24. Николаев И.С., Митренина О.В., Ландо Т.М. Прикладная и компьютерная лингвистика // М.URSS. — 2016. — 320 с.
25. Nothman J., Qin H., Yurchak R. Stop Word Lists in Free Open-source Software Packages. // Proc. Workshop for NLP Open Source Software. — Association for Computational Linguistics. — 2018. — P. 712.
26. Baby Names from Social Security Card Applications – National Data. — URL: <https://catalog.data.gov/dataset/baby-names-from-social-security-card-applications-national-level-data> (дата обращения: 09/04/2020).
27. McCreery C. First-year Statistics for Psychology Students Through Worked Examples. 1. Probability and Bayes' Theorem // Oxford Forum. — 2018. — 29 p.
28. Abramov V., Kugurakova V., Rizvanov A., Abramskiy M., Manakhov N., Evstafiev M., Ivanov D. Virtual Biotechnological Lab Development // BioNanoScience. — 2017. — V. 7. — No 2. — P. 363–365.
29. Kugurakova V., Abramov V., Sultanova R., Tsivilskiy I., Talanov M. Virtual Reality-Based Immersive Simulation for Invasive Surgery Training // European journal of clinical investigation. — 2018. — V. 48. — P. 224–225.

## References

1. Al-Husseini Kh., Obaid A. Usage of prototyping in software testing // Multi-Knowledge Electronic Comprehensive Journal For Education And Science Publications. — 2018. — V. 14. — P. 1–15.
2. Schell J. The Art of Game Design: A Deck of Lenses, Second Edition. — CRC Press. — 2014. — 594 p.
3. Davies D., Bathurst D., Bathurst R. The Telling Image: The Changing Balance between Pictures and Words in a Technological Age // Technology and Culture. — The Johns Hopkins University Press and the Society for the History of Technology Stable. — 1992. — V. 33. — No 4. — P. 845–846.
4. Raja D., Bowman D., Lucas J., North C. Exploring the benefits of immersion in abstract information visualization // Proc. of the 8th Int'l Immersive Projection Technology Workshop. — 2004.
5. Koenitz H., Dubbelman T., Knoller N., Roth C., Haahr M., Sezen D., Sezen T. Card-Based Methods in Interactive Narrative Prototyping // ICIDS 2018, Proceedings International Conference on Interactive Digital Storytelling. — Springer Nature Switzerland AG. — 2018. — P. 552–555.
6. Antonisse J. Paper Tales: A Guide to Narrative Prototyping [Video] // Game Developer Conference. — 2014. — URL: <https://www.gdcvault.com/play/1020509/Paper-Tales-A-Guide-to> (data obrashhenija: 09/04/2020).
7. Cairns P., Cox A., Nordin I. Immersion in Digital Games: Review of Gaming Experience Research // Handbook of Digital Games. — Wiley-IEEE Press. — 2014. — P. 337–361.
8. Newman J. Videogames // Routledge. — 2014. — P. 198.

9. Blanchard L. Creating empathy in video games. [MSC Thesis] // The University of Dublin. — 2016. — 42 p.
10. Koivisto E., Suomela R. Using prototypes in early pervasive game development // Sandbox '07: Proc. of the 2007 ACM SIGGRAPH symposium on Video games. — Association for Computing Machinery. — 2007. — P. 149–156.
11. Hassani K., Lee W.-S. Visualizing Natural Language Descriptions: A Survey // ACM Computing Surveys — Association for Computing Machinery — 2016. — V. 49. — No 1.
12. RivetAI. — URL: <https://www.rivetai.com/> (data obrashhenija: 09/04/2020).
13. Gupta T., Schwenk D., Farhadi A., Hoiem D., Kembhavi A. Imagine This! Scripts to Compositions to Videos // Cornell University. — 2018.
14. Storyboarder. // Wonder Unit. — URL: <https://wonderunit.com/storyboarder/> (data obrashhenija: 09/04/2020).
15. 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. — P. 34–40.
16. 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 // Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering. — Springer Nature Switzerland AG. — 2016. — P. 29–38.
17. 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.
18. Adams E., Joris D. The Designer's Notebook: Machinations, A New Way to Design Game Mechanics — URL: [https://www.gamasutra.com/view/feature/176033/the\\_designers\\_notebook](https://www.gamasutra.com/view/feature/176033/the_designers_notebook) (data obrashhenija: 09/04/2020).
19. Orange 3. — URL: <https://orange.biolab.si/> (data obrashhenija: 09/04/2020).
20. Sahibgareeva G.F., Kugurakova V.V. Koncept instrumenta avtomaticheskogo sozdaniya scenarnogo prototipa komp'yuternoj igry (The concept of automatic creation tool for computer game scenario prototype) // Jelektronnye biblioteki. — 2018. — V. 21. — No 3-4. — P. 235–249.
21. Dobrokvashina A.S., Gazizova Je.A. Avtomatizacija proektirovanija igrovogo prototipa na osnovanii obrabotki formalizovannogo igrovogo dizajndokumenta (Automatization of a gaming prototype development based on the result of processing of a formalized game design document) // Uchenye zapiski ISGZ. — Kazan': Institut social'nyh i gumanitarnyh znaniy. — 2019. — V. 17. — No 1. — P. 583–589.

22. Astaf'ev A. Razrabotka instrumenta dlja sborki scen po tegam (Development of a tool for assembling scenes by tags) [rukopis'] – 2019. – 31 p.
23. Storyboarder. / Source code // GitHub. — URL: <https://github.com/wonderunit/storyboarder> (data obrashhenija: 09/04/2020).
24. Nikolaev I.S., Mitrenina O.V., Lando T.M. Prikladnaja i komp'juternaja lingvistika (Applied and Computational Linguistics) // M.URSS. — 2016. — 320 p.
25. Nothman J., Qin H., Yurchak R. Stop Word Lists in Free Open-source Software Packages. // Proc. Workshop for NLP Open Source Software. — Association for Computational Linguistics. — 2018. — P. 712.
26. Baby Names from Social Security Card Applications – National Data. — URL: <https://catalog.data.gov/dataset/baby-names-from-social-security-card-applications-national-level-data> (data obrashhenija: 09/04/2020).
27. McCreery C. First-year Statistics for Psychology Students Through Worked Examples. 1. Probability and Bayes' Theorem // Oxford Forum. — 2018. — 29 p.
28. Abramov V., Kugurakova V., Rizvanov A., Abramskiy M., Manakhov N., Evstaf'ev M., Ivanov D. Virtual Biotechnological Lab Development // BioNanoScience. — 2017. — V. 7. — No 2. — P. 363–365.
29. Kugurakova V., Abramov V., Sultanova R., Tsivilskiy I., Talanov M. Virtual Reality-Based Immersive Simulation for Invasive Surgery Training // European journal of clinical investigation. — 2018. — V. 48. — P. 224–225.