

# Прототипирование вариативности сюжета компьютерных игр

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

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

**Аннотация.** В работе рассматривается вопрос визуализации разветвленных структур повествования компьютерных игр, а также генерации продолжения развития событий. В связи с этим проанализированы существующие решения, а также представлены собственные наработки. В результате была сформирована концепция, а также был избран технологический стек, который позволил реализовать простое, но эффективное решение для сценаристов компьютерных игр. Основной посыл работы – оптимизировать труд, высвободить ресурсы на решении задач разработки благодаря автоматизации рутинных работ.

**Ключевые слова:** интерактивное повествование, компьютерные игры, сценарий игры, визуализация, разветвленные структуры, графики, автоматизация, тестирование, повествовательное прототипирование, прототип сценария

## Prototyping the variability of computer games plots

G.F. Sahibgarееva<sup>1</sup>, V.V. Kugurakova<sup>1</sup>

<sup>1</sup> *Institute of Information Technology and Intelligent Systems, Kazan Federal University*

**Abstract.** The paper presents a solution to the issue of visualizing the branched structures of the narrative of computer games, as well as generating the continuation of the development of events. In this regard, the existing solutions are analyzed, as well as our results are presented. As a result, a conception was formed, and a technology stack was chosen, which made it possible to implement a simple but effective solution for computer game script writers. The main message of the work is to optimize labor, to free up resources for solving development problems by automating the routine.

**Keywords:** interactive storytelling, computer games, game script, visualization, branched structures, graphs, automation, testing, narrative prototyping, script prototype

## **Введение**

Процесс разработки компьютерных игр – длительный и дорогостоящий процесс. По этой причине в индустрии актуально применение инструментов автоматизации рутинных процессов.

При создании компьютерных игр идет разработка в большом количестве направлений: 2D/3D визуализация, UI/UX дизайн, программирование и дизайн игрового процесса, программирование искусственного интеллекта, дизайн персонажей, уровня и окружения, создание сценария, нарративный дизайн, звуковое и музыкальное сопровождение.

В данной работе рассматривается автоматизация процесса реализации и тестирования сценария компьютерной игры. С этой целью были разработаны алгоритмы визуализации разветвленной структуры, построения и проверки графов, а также сборки проекта на основе одного файла JSON. Данные алгоритмы интегрированы в пилотное решение инструмента, который является реализацией небольшого перечня функциональности, которая реализуется в рамках разработки модульного инструмента работы над интерактивным повествованием компьютерных игр. Данные наработки представлены в ряде работ авторов (см., например, [1, 2]).

В данной работе представлено описание и результаты работы над пилотным решением инструмента. *Цель работы* – протестировать возможность реализации изолированного функционала целостного инструмента. В рамках цели были выполнены следующие *задачи*: анализ существующих референсных инструментов, поиск оптимальные технологий для решения цели, реализация пилотного решения, тестирование и проверка гипотез.

В *Разделе 1* представлен анализ существующих инструментов и научных статей со связанной тематикой. В *Разделе 2* описана концепция комплексного инструмента и пилотной реализации, а также представлена архитектура решения. В *Разделе 3* описаны некоторые детали реализации пилота, представлены примеры проекта и функциональности. В *Заключении* представлены выводы и планы на будущее.

### **1. Связанные работы и существующие инструменты**

В качестве референсных инструментов проанализирован следующий ряд приложений: Twine, Articy:Draft, Fungus, Storybricks Engine.

*Twine* – открытая платформа для написания интерактивной литературы, текстовых игр и визуальных новелл. Позволяет публиковать игры в виде html-страничек и не требует от автора познаний в программировании [3].

*Articy:Draft* – решение для работы с интерактивным повествованием и управлением игровым контентом. Инструмент предлагает широкий набор функций, облегчающих написание и тестирование игрового сценария [4].

*Fungus* – это бесплатный инструмент с открытым исходным кодом для создания игр-рассказов с акцентом на доступность для начинающих разработчиков игр [5].

*Storybricks Engine* – ИИ механизм рассказывания историй, который дает разработчикам возможность создавать и контролировать нарратив с крайне сложными, ветвящимися сюжетными арками [6].

Анализ данных инструментов показал, что возможно создание инструмента для сценаристов с низким порогом вхождения. Данный фактор важен, т.к. зачастую сценаристы компьютерных игр не имеют опыта программирования.

Кроме реализованных инструментов был выделен ряд интересных в рамках данной работы научных статей, которые сфокусированы на визуализации и построении разветвленных структур.

В *StoryFlow* [7] структура представлена в виде особой структуры Yarn (рис. 1). 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.

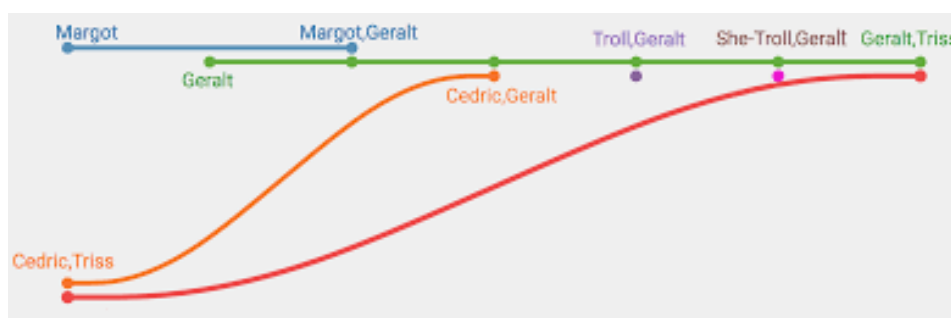


Рис. 1. Структура Yarn

В работе [8] был проанализирован 20-летний опыт разработки интеллектуальных систем для компьютерных игр. Один из выводов, который сделали авторы – интеллектуальные системы способствуют увеличению уровня вариативности сюжета игр.

Одна из интересных для проекта визуализаций – *диаграмма Санкей* [9] (рис. 2). Данное представление может помочь отразить специфические для проекта данные.

Кроме того была предпринята попытка собрать разнообразные структуры и представить их описание в работе одного из авторов – [10].

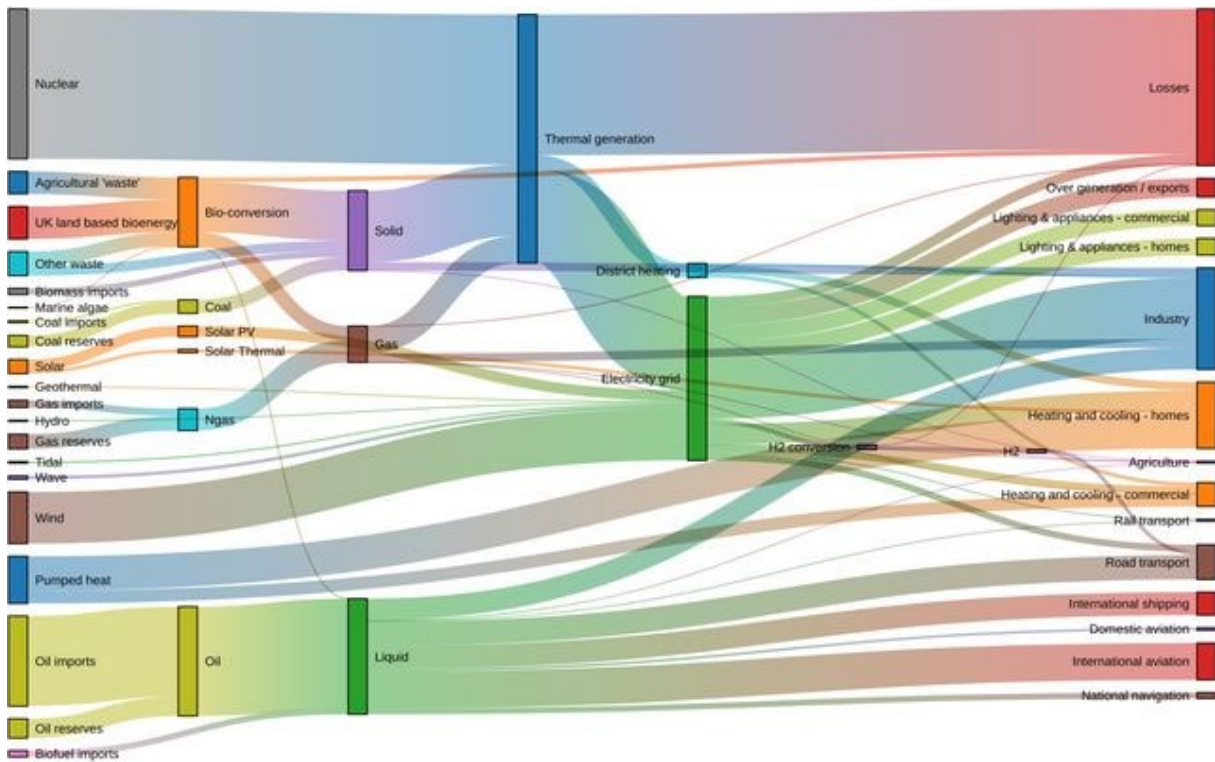


Рис. 2. Пример диаграммы Санкей

В качестве примера сложности интерактивных структур приложен скрин фрагмента графа сценария из рабочего проекта компании Quantic Dream над игрой Detroit: Become Human – (рис. 3) [11].

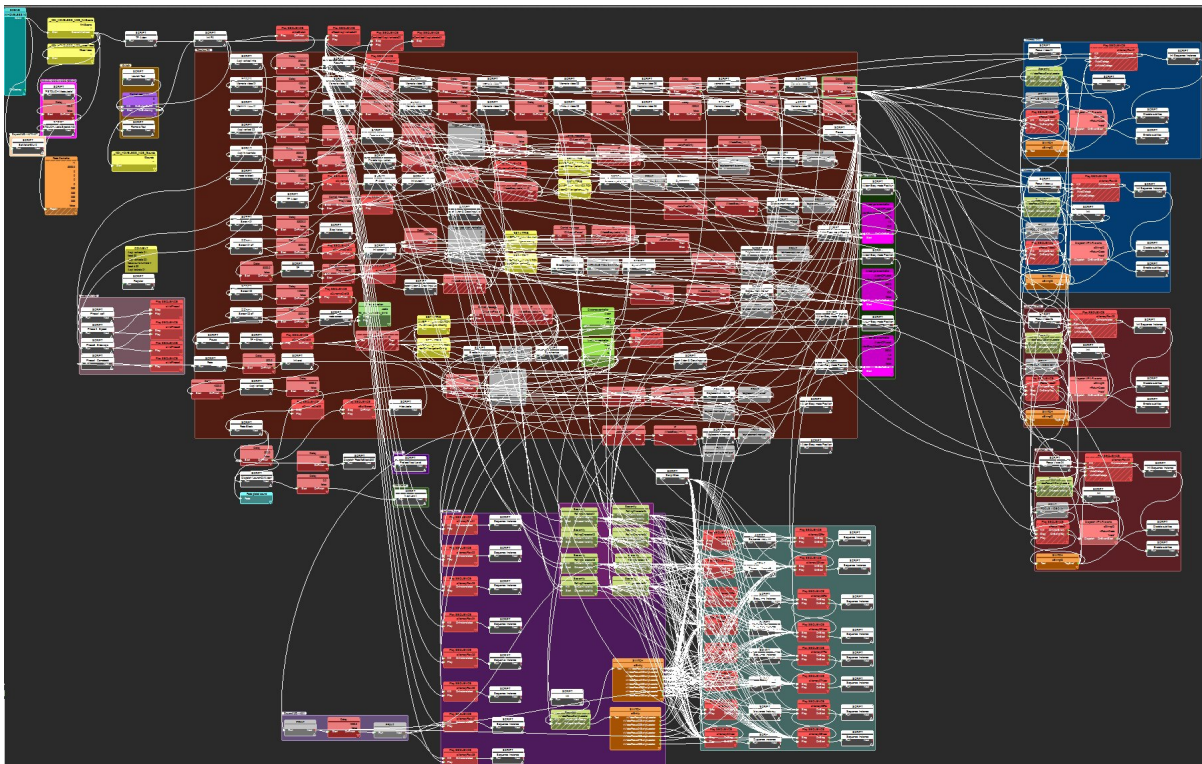


Рис. 3. Фрагмента графа сценария игры Detroit: Become Human

На текущем этапе работы было принято решение остановиться на представлении в виде графа. Однако это не означает отказ от других форм визуализации, которые могут оказаться уместными для отображения данных для частных задач.

## 2. Общая концепция и пилотное решение ветвления и продолжения сюжета

Для продолжения работы над инструментом, который был заявлен в предшествующих работах, был выделен изолированный функционал, который был реализован в виде пилотного решения, которое доступно для тестирования и проверки всевозможных гипотез.



Рис. 4. Архитектура инструмента генерации сценарного прототипа

Прежде чем представить концепцию локального функционала, представим концепцию целостного инструмента.

### 2.1. Работа конвейера в целом

Полный цикл работы инструмента генерации сценарного прототипа описан в предшествующих работах [1, 2, 12, 13]. Для удобства продублируем обобщенный конвейер:

1. Текст сценария на естественном языке анализируют алгоритмы, которые извлекают из них информацию о внутриигровых сущностях: имена и характеристики персонажей, их реплики, описание локации, основные события.
2. Информация о разветвленной структуре визуализируется в удобной форме, приводится статистика.

3. На основе полученной информации генерируется трехмерная сцена, автоматически подбираются трехмерные модели и анимации.
4. Генерируется программная возможность перехода между событиями.
5. Сборка проекта завершается формированием установочного файла, который является *сценарным прототипом*, иными словами, интерактивным проектом, которые игроки и все заинтересованные лица могут пройти или протестировать.

В качестве схематичного представления представлены общая архитектура инструмента (рис. 4) и конвейер инструмента визуализации в форме раскадровки (рис. 5).



Рис. 5. Конвейер инструмента генерации раскадровки

## 2.2. Визуализация ветвления сюжета

В разных работах авторов описаны реализации изолированных модулей комплексного инструмента прототипирования компьютерных игр. В данной работе приведена реализация функции визуализации разветвленной структуры повествования, возможности редактировать ее, генерировать продолжение развития событий, проверять структуру на целостность и непротиворечивость, а также запускать игровой проект для проигрывания полученных событий. Функционал генерации продолжения был интегрирован благодаря опыту инструмента [6].

Жанр генерируемого сценарного прототипа – текстовая игра.

В целом комплексный инструмент должен представлять собой набор редакторов различных аспектов игрового проекта. Примечательно, что граф проходит проверку различными алгоритмами, которые освобождают пользователя от необходимости тщательно вычитывать проект самостоятельно, т.е. данные процессы автоматизированы.

Основная задача инструмента – визуализировать разветвленную структуру сценария компьютерной игры, предоставить возможность автоматической проверки структуры, а также возможность сборки и воспроизведения игрового проекта в виде текстовой игры.

### 2.3. Архитектура

Схема устройства пилота представлена на (рис. 6).

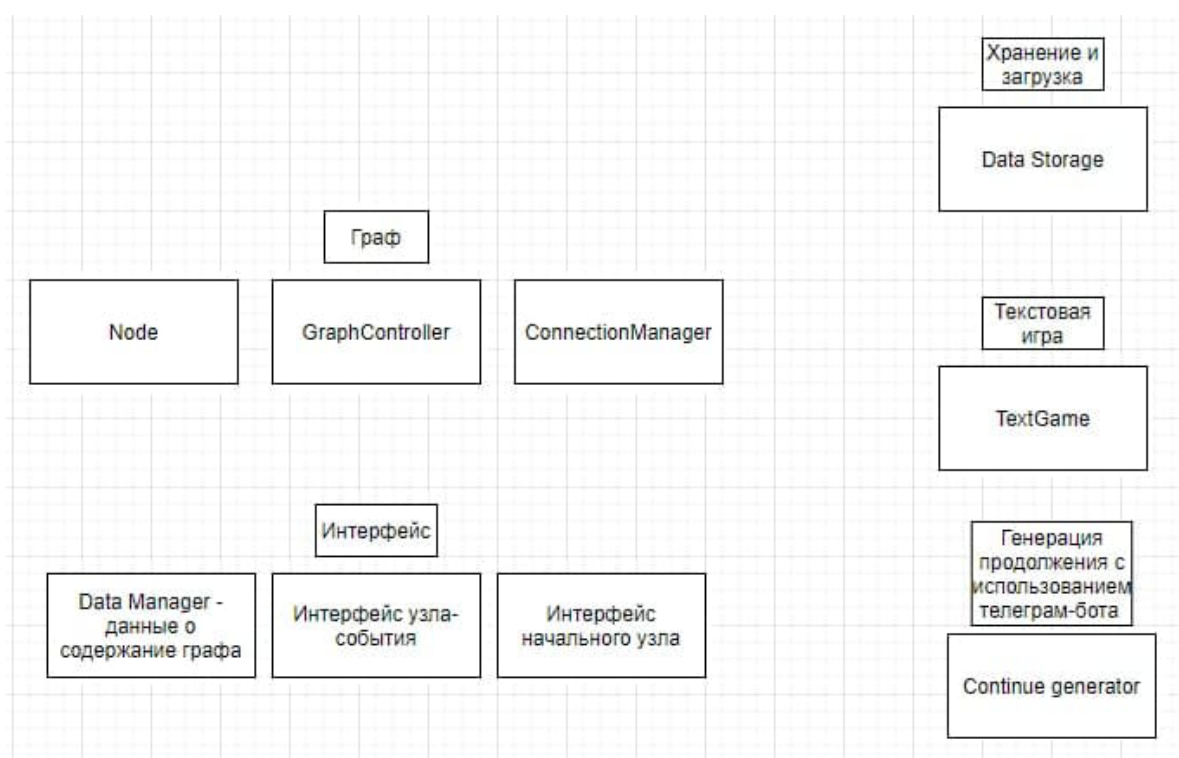


Рис. 6. Схема устройства инструмента

### 2.4. Графовое представление

Опишем уже реализованный пилотный интерфейс для визуализации сценария – части реализации детально представлены в ряде [14, 15, 16] выпускных работ студентов бакалавриата ИТИС КФУ, выполненных под научным руководством одного из авторов, а также непосредственном участии в постановке задачи, консультировании, выборе технологий обоих авторов. Практику совместной разработки пилотных реализаций отдельного функционала со студентами мы намерены продолжить и далее, т.к. она позволяет проверить большое количество подходов и тактик.

Граф сценария любого проекта начинается со стартовой вершины. При взаимодействии пользователя с ней открывается редактор персонажей и их свойств. В любой момент пользователь может создавать вершины действия и связывать их между собой. Условия связывания: стартовая вершина должна быть связана с минимум одной вершиной действия; любая вершина действия должна иметь родительскую и дочернюю вершину, если это не последняя вершина в структуре.

При взаимодействии с вершиной действия становится доступен редактор внутриигровых событий.

Вершины можно дублировать или удалять.

Данный функционал работы с графом необходим только для данной реализации инструмента. В идеале, граф должен создаваться автоматически из текста на естественном языке. Данные алгоритмы были реализованы в прошлых отдельных реализациях инструмента. В будущем данный функционал будет объединен в рамках одного инструмента.

Упорядоченный файл сценария (представлен в формате *JSON* в текущей реализации, что не ограничивает сущности) содержит в себе следующую информацию: список персонажей, перечень и значения их свойств; разветвленную структуру в виде направленного графа; условия и производимые эффекты для каждой вершины графа; текстовый отрывок для вершины и выбора действия или реплики; условия перехода из вершины в вершину. Назовём такой упорядоченный файл – *структурированный сценарий*.

Инструмент визуализирует содержимое структурированного сценария в виде графа, а также воспроизводит игровую сессию, в которой игрок получает весь заложенный контент. Помимо этого внедренные алгоритмы позволяют проверить граф на целостность и непротиворечивость. Дополнительный модуль позволяет генерировать на основе существующих текстовых отрывков и доступных игроку действий дополнительные вершины, т.е. продолжение развития событий.

Инструмент разработан в игровом движке *Unity* [17], использован язык программирования *C#*.

В качестве генератора продолжения текста была использована модель *GPT-2* [18], которая описана ниже. Алгоритм всякий раз выдает уникальный текст даже при одинаковом запросе.

В рамках разработки компьютерных игр реализуется большой спектр разветвленных структур [10]. Данные структуры могут быть использованы в проектах, представленных в виде структурированных сценариев. Простейшие из этих структур могут быть считаны текущей версией инструмента. Главное ограничение на сегодняшний день – низкая степень вложенность и односторонняя направленность рёбер.

Приведем пример фрагмента структурированного файла:

```
{"Name": "Выбор класса"},
```



```
    {"IsRootNode":false,"Id":40,"Position":{"x":-29.970855712890626,"y":-4.958108425140381,"z":50.0},
    "Name":"Рыцарь"},
    {"IsRootNode":false,"Id":38,"Position":{"x":-30.200014114379884,"y":3.138796329498291,"z":50.0},
    "Name":"Лучник"},
    {"IsRootNode":false,"Id":39,"Position":{"x":-30.0472469329834,"y":-0.9096595048904419,"z":50.0}}
```

## 2.5. Поиск несоответствий

Отдельное внимание стоит уделить функционалу поиска несоответствий в построенном графе сценария. Это могут быть несоответствия свойств персонажей и переходов между вершинами.

Каждый персонаж – это его имя, а также значения составляющих описание данного персонажа свойств. Свойства могут представлять собой целочисленные, логические или текстовые значения.

Доступен автоматический переход в вершину при выполнении действия или самостоятельный переход игрока в зависимости от выбора.

В вершине возможна проверка значений свойств на выполнение условий. Если условие не выполнено – вершина для игрока недоступна.

*Пример:* персонаж имеет текстовое значение класса лучник. Если у персонажа другой класс, то ему доступен другой пул вариантов развития событий.

Кроме того в каждой вершине возможно применение функций изменения значений свойств в зависимости от происходящих событий.

*Пример:* лучник имеет целочисленное свойство “здоровье”. Если персонаж попал в неприятность, данное значение снижается, что производится за счет соответствующих вычислений в вершине.

При работе с балансом уместно применение интеграция функционала инструмента Machination [19].

## 2.6. Генерация продолжения текста

Для генерации продолжения развития сюжета в инструменте используется результат работы модели *GPT-2* (Generative Pre-trained Transformer) [15]. Генерация происходит на основе фрагмента текста в текущей вершине (рис. 7).

Главной особенностью *GPT-2* является то, что нейросеть не нужно дообучать под конкретную задачу, чтобы та показывала нужные пользователю результаты. Нейросеть приспосабливается к стилю и содержанию текста, что позволяет ей генерировать реалистичные отрывки, продолжающие исходные фразы. Сразу после обучения нейросеть уже готова генерировать текст со всеми логическими вставками: повторное упоминание имен героев, цитаты, отсылки, выдержка одного стиля на протяжении всего текста, связанное повествование.

На данный момент полная версия *GPT-2* не доступна, однако уменьшенная версия находится в открытом доступе. Также есть возможность обучить существующую модель на своем собственном массиве текстов, с чем мы начали работы.

Вы начинающий искатель приключений.  
Вы только что покинули город и стоите на развилке.  
Перед вами открывается замечательный вид на равнины и дорогу, уходящую в даль.  
Пока вы стоите и любуетесь видом, к вам подходит дружелюбный с виду гном.  
- Привет, я Эрих, держу путь в Эрисгорн. Смотрю ты тоже из искателей приключений. Не хочешь отправиться вместе? Вдвоем и веселей будет, и если что помогу тебе.

Вы начали путь по тракту.  
Путешествие проходило хорошо. Эрих рассказывал много интересных историй из своей жизни и давал полезные советы о том, как быть искателем приключений.  
Однако через некоторое время из кустов выпрыгнула группа разбойников.  
- Отдавайте нам все ваши деньги, ценности и оружие. Может быть тогда мы даже отпустим вас живыми.  
Эрих сразу понял, что в живых их не оставят. Он сообщил об этом ГГ. Что же решит делать ГГ?  
**А ГГ недолго думал. "Чего же я буду убегать от них, - подумал он. - Я уже давно хочу испытать себя в качестве искателя приключений, а для этого есть отличный способ. Давай-ка..."**

Рис. 7. Сгенерированное продолжение текста выделено жирным

## 2.7. Запуск игрового проекта

После того, как пользователь загрузил и настроил граф (рис. 8), он может запустить процесс сборки проекта.

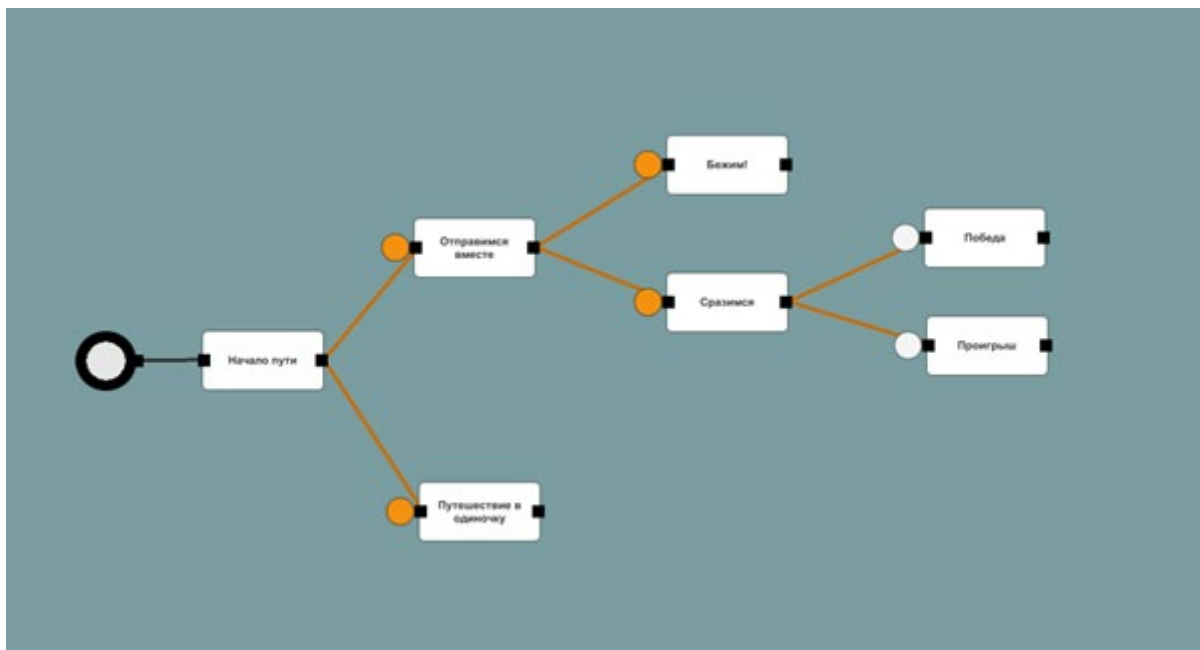
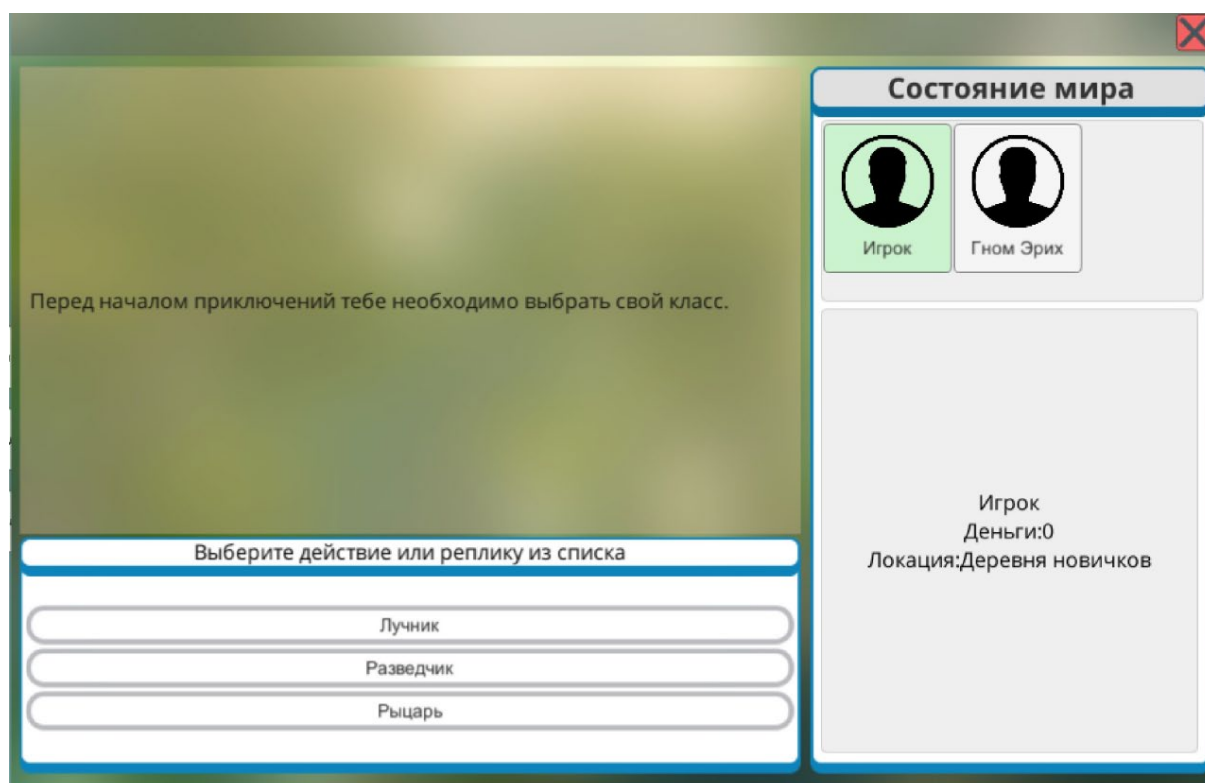


Рис. 8. Визуализация графа

Данный этап включает в себя анализ графа с помощью алгоритма проверки на целостность и непротиворечивость. Данный алгоритм проверяет все условия и воздействия вершин на значения свойств персонажа, просчитывая все линии развития сюжета игры. Если была обнаружена ошибка, вершины с несоответствиями подсвечиваются, а при просмотре их содержимого доступна информация о том, где была допущена неточность.



## Рис. 9. Запуск игры

Если ошибок в структуре нет, то запускается *текстовая игра* (с англ. *text-based game* – устоявшийся термин для жанра игр, состоящих из описаний и диалогов, оформленных статичными или анимированными картинками, с переходами по выбору), которую игрок может пройти (рис. 9).

### Заключение

Представленное пилотное решение для работы над сценарием компьютерной игры показало, что возможно эффективное решение вопроса визуализации разветвленной структуры, а также реализуемы варианты автоматической генерации продолжений.

Данное решение станет очередной частью одного большого инструмента, призванного упростить работу игровых сценаристов и повысить качество игрового повествование.

Перспективы развития данного решения и будущего инструмента в целом заключается в последующей реализации и тестировании, а также в объединении функционала в общий конвейер обработки и визуализации информации, а также в генерировании текстового, визуального, программного и проектного контента.

В частности, в текущую разработку будет интегрирована возможность создания и отслеживания квестов, а также реализована возможность для поддержки более сложным и массивных разветвленных структур.

### 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.
3. Twine – <https://twinery.org/>.
4. Articy:draft – <https://www.articy.com/en/>.
5. Fungus – <https://fungusgames.com/>.
6. Storybricks Engine – [https://www.youtube.com/watch?v=id-3sUo\\_DFU&ab\\_channel=Storybricks](https://www.youtube.com/watch?v=id-3sUo_DFU&ab_channel=Storybricks).
7. Padiá K., Bandara K., Healey C. A system for generating storyline visualizations using hierarchical task network planning // Computers & Graphics, 2019. – Elsevier. – P. 64–75.

8. Riedl M.O., Bulitko V. Interactive Narrative: An Intelligent Systems Approach // AI Magazine, 2013. – V. 34. – №1. – P. 67.
9. Sankey Diagram – <https://observablehq.com/@d3/sankey-diagram>
10. Сахибгареева Г.Ф. Применимость разветвленных структур для генерации сценарных прототипов видеоигр // 65-я Международная научная конференция Астраханского государственного технического университета, 2021.
11. Detroit: Become Human – [https://twitter.com/David\\_\\_Cage/status/1034374760392794112](https://twitter.com/David__Cage/status/1034374760392794112).
12. 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.
13. Sahibgareeva G.F., Bedrin O.A., Kugurakova V.V. Visualization Component for the Scenario Prototype Generator as a Video Game Development Tool // CEUR. Proceedings of the 22nd Conference on Scientific Services & Internet (SSI-2020), 2020. – P.267-282.
14. Вакатов С.А. Разработка инструмента вариативности сюжета с запуском прототипа в виде текстовой игры [рукопись] – 2021. – 36 с.
15. Вакатова Э.С. Разработка функционала генерации продолжения сюжета для инструмента прототипирования сюжета в компьютерных играх [рукопись] – 2021. – 33 с.
16. Каюмов Б.И. Проблемы визуализации разветвленных сюжетов компьютерных игр [рукопись] – 2021. – 79 с.
17. Unity – <https://unity.com/ru>.
18. GPT-2 – <https://openai.com/blog/better-language-models/>.
19. 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).

## References

1. 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.
2. Sahibgareeva G.F., Bedrin O.A., Kugurakova V.V. Razrabotka komponenta generatsii vizualizatsii stsenarnogo prototipa videoigr // Scientific service & Internet: proceedings of the 22nd All-Russian Scientific Conference (September 21-25, 2020, online). – M.: IPM im. M.V.Keldysha, 2020. – P. 581-603.
3. Twine – <https://twinery.org/>.
4. Articy:draft – <https://www.articy.com/en/>.
5. Fungus – <https://fungusgames.com/>.

6. Storybricks Engine – [https://www.youtube.com/watch?v=id-3sUo\\_DFU&ab\\_channel=Storybricks](https://www.youtube.com/watch?v=id-3sUo_DFU&ab_channel=Storybricks)
7. Padia K., Bandara K., Healey C. A system for generating storyline visualizations using hierarchical task network planning // *Computers & Graphics*, 2019. – Elsevier. – P. 64–75.
8. Riedl M. O., Bulitko V. Interactive Narrative: An Intelligent Systems Approach // *AI Magazine*, 2013. – V. 34. – №1. – P. 67.
9. Sankey Diagram – <https://observablehq.com/@d3/sankey-diagram>.
10. Sahibgareeva G.F. Primenimost razvetvlennykh struktur dlia generatsii stsenarnykh prototipov videoigr // 65th International Scientific Conference of Astrakhan State Technical University, 2021.
11. Detroit: Become Human – [https://twitter.com/David\\_\\_Cage/status/1034374760392794112](https://twitter.com/David__Cage/status/1034374760392794112).
12. Sahibgareeva G.F., Bedrin O.A., Kugurakova V.V. Raskadrovka kak odno iz predstavlenii stsenarnogo prototipa kompiuternykh igr // *Jelektronnye biblioteki*, 2021. – V. 24. – No 2. – P. 408-444.
13. Sahibgareeva G.F., Bedrin O.A., Kugurakova V.V. Visualization Component for the Scenario Prototype Generator as a Video Game Development Tool // *CEUR. Proceedings of the 22nd Conference on Scientific Services & Internet (SSI-2020)*, 2020. – P. 267-282.
14. Vakatov S.A. Razrabotka instrumenta variativnosti syuzheta s zapuskom prototipa v vide tekstovoj igry [rukopis'] – 2021. – 36 s.
15. Vakatova E.S. Razrabotka funkcionala generacii prodolzheniya syuzheta dlya instrumenta prototipirovaniya syuzheta v komp'yuternykh igrakh [rukopis'] – 2021. – 33 s.
16. Kayumov B.I. Problemy vizualizacii razvetvlennykh syuzhetov komp'yuternykh igr [rukopis'] – 2021. – 79 s.
17. Unity – <https://unity.com/ru>.
18. GPT-2 – <https://openai.com/blog/better-language-models/>.
19. 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).