ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО
Приглашенный преподаватель
кафедры менеджмента инноваций

Novikov Eugen VWPKI 31FE642550364FDC
Digital unterschrieben von Novikov Eugen VWPKI 31FE642550364FDC Datum: 2022.05.15 19:39:58 +02'00'

_____ Е.А. Новиков
«16» мая 2022 г.

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук

_____ В.В. Шилов
«16» мая 2022 г.

ФРОНТЕНД КОНСТРУКТОРА ВИЗУАЛЬНЫХ НОВЕЛЛ

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.08-01 81 01-1-ЛУ

Исполнитель
студент группы БПИ 206
_____/Д.О.Бутченко/
«16» мая 2022 г.

Москва 2022

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

УТВЕРЖДЕН
RU.17701729.04.08-01 81 01-1-ЛУ

ФРОНТЕНД КОНСТРУКТОРА ВИЗУАЛЬНЫХ НОВЕЛЛ

Пояснительная записка

RU.17701729.04.08-01 81 01-1

Листов 50

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Москва 2022

СОДЕРЖАНИЕ

Глоссарий . . . 5

1. Введение . . . 6

1.1. Название программы . . . 6

1.1.1. Название программы на русском языке . . . 6

1.1.2. Название программы на английском языке . . . 6

1.2. Документы, на основании которых ведется разработка . . . 6

2. Назначение и область применения . . . 7

2.1. Функциональное назначение . . . 7

2.2. Эксплуатационное назначение . . . 7

3. Технические характеристики . . . 8

3.1. Постановка задачи на разработку программы . . . 8

3.2. Описание алгоритма и функционирования программы с обоснованием выбора . . . 8

3.2.1. Описание функционирования . . . 8

3.2.2. Общие для окон проекта и сцены функции . . . 8

3.2.2.1. Панель инструментов . . . 9

3.2.2.2. Панель выбора переходов . . . 11

3.2.2.3. Панель координат . . . 11

3.2.2.4. Граф . . . 12

3.2.2.5. Узлы . . . 12

3.2.3. Окно проекта . . . 12

3.2.3.1. Узлы . . . 13

3.2.4. Окно сцены . . . 14

3.2.4.1. Узлы . . . 14

3.2.4.2. Переход к графу проекта . . . 18

3.3. Описание и обоснование выбора метода организации входных и выходных данных . . . 18

3.3.1. Входные данные . . . 18

3.3.1.1. Изображения . . . 18

3.3.1.2. Звуковые файлы . . . 19

3.3.1.3. Текстовые данные . . . 19

3.3.2. Выходные данные . . . 19

3.4. Описание и обоснование выбора состава технических и программных средств 19
3.4.1. Unity 19
3.4.2. C# 20
4. Ожидаемые технико-экономические показатели 21
4.1. Предполагаемая потребность 21
4.2. Ориентировочная экономическая эффективность 21
4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами 22
Перечень использованных источников 23
Приложение 1. Описание и функциональное назнаение классов 24
Приложение 2. Описание и функциональное назнаение полей, методов и свойств классов . 26

АННОТАЦИЯ

Данный документ является пояснительной запиской к программному проекту «Фронтенд Конструктора Визуальных новелл», реализующему клиентскую сторону пользовательского интерфейса приложения для создания игр в жанре "Визуальная новелла".

Раздел «Введение» содержит наименование программы, условное обозначение темы разработки, документ, на основании которого ведётся разработка, а также организация, утвердившая данный документ.

Раздел «Назначение и область применения» содержит функциональное и эксплуатационное назначение программы и краткая характеристика области её применения.

Раздел «Технические характеристики» содержит следующие подразделы: постановка задачи на разработку программы, описание алгоритма и функционирования программы с обоснованием выбора, описание и обоснование выбора метода организации входных и выходных данных, описание и обоснование выбора состава технических и программных средств.

Раздел «Ожидаемые технико-экономические показатели» содержит предполагаемую потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами.

Данный документ разработан в соответствии с требованиями:

1) ГОСТ 19.101-77 Виды программ и программных документов [1]
2) ГОСТ 19.102-77 Стадии разработки [2]
3) ГОСТ 19.103-77 Обозначения программ и программных документов [3]
4) ГОСТ 19.104-78 Основные надписи [4]
5) ГОСТ 19.105-78 Общие требования к программным документам [5]
6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6]
7) ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [7]

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

ГЛОССАРИЙ

Проект – итоговый продукт (игра), состоящий из сцен.

Сцена – созданная пользователем последовательность игровых событий.

Персонаж – объект сцены, обладающий именем и изображением.

Визуальная новелла – видеоигра в жанре «Визуальный роман».

Десктопное приложение – программа, которая устанавливается на ПК пользователя.

Граф проекта – граф, состоящий из входного узла и узлов сцен, связанных между собой через предопределенные пользователем соединения.

Граф сцены – граф, состоящий из узлов, указанных ниже, связанных между собой через предопределенные пользователем соединения.

1) **E-Узел (Entry)** – начальный узел сцены.
2) **CBG-Узел (Change Background)** – узел, при активации которого меняется фон текущей сцены.
3) **CD-Узел (Change Dialog)** – узел, при активации которого меняется текст выбранного персонажа.
4) **CN-Узел (Change Name)** – узел, при активации которого меняется имя персонажа, который говорит в настоящий момент.
5) **PS-Узел (Play Sound)** – узел, при активации которого воспроизводится выбранный звуковой файл.
6) **SC-Узел (Show Character)** – узел, при активации которого на сцене появляется выбранный персонаж.
7) **HC-Узел (Hide Character)** – узел, при активации которого на сцене пропадает выбранный персонаж.
8) **SS-Узел (Start Scene)** – узел, при активации которого воспроизводится выбранная сцена.

Область входа узла – выделенная область слева на графическом представении узла, при зажатии ЛКМ на которую можно создать переход, ведующий в данный узел.

Область выхода узла – выделенная область справа на графическом представении узла, при зажатии ЛКМ на которую можно создать переход, ведующий из данного узла.

Переход – один из двух типов соединения узлов графа, указанных ниже.

1) **SMPL-Переход (Simple)** – моментальный переход.
2) **CLK-Переход (Click)** – переход, если игрок нажал на левую клавишу мыши.

Json – текстовый формат обмена данными, используется для сохранения состояния программы.

Hotkey – Сочетание клавиш, вызывающее определенную функцию.

1. ВВЕДЕНИЕ

1.1. Название программы

1.1.1. Название программы на русском языке

«Фронтенд конструктора визуальных новелл»

1.1.2. Название программы на английском языке

«Front-end of the visual novel constructor»

1.2. Документы, на основании которых ведется разработка

Учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем программы тема курсового проекта.

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Функциональное назначение

Программа предоставляет специализированный инструментарий для создания, запуска и распространения игр в жанре «визуальная новелла».

2.2. Эксплуатационное назначение

Эксплуатационным назначением данного приложения является облегчение процесса разработки визуальных новелл. Для достижения этой цели пользователи могут использовать концепцию «visual programming».

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи на разработку программы

Программа была разработана в соответствии с техническим заданием «Фронтенд конструктора визуальных новелл».

3.2. Описание алгоритма и функционирования программы с обоснованием выбора

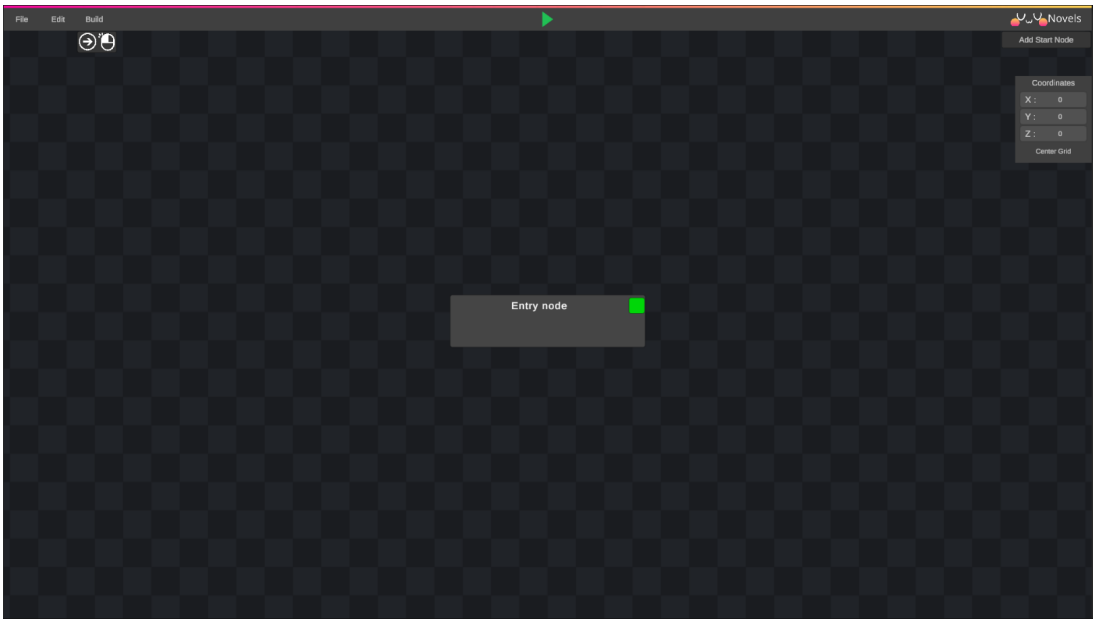
3.2.1. Описание функционирования

Констуктор работает на концепции визуального программирования с использованием узлов. Данная концепция позволяет полностью исключить написание кода конечным пользователем, упростив изучение и использование конструктора. За работу с инструментами конструктора и узлами отвечают Unity скрипты.

Ярким примером могут послужить Blueprint UnrealEngine 4 [10] и работа с материалами в Blender [11].

3.2.2. Общие для окон проекта и сцены функции

При входе в приложение пользователь попадает на страницу с проектом.



Окно редактирования проекта.

Рис. 1.

3.2.2.1. Панель инструментов

Сверху находится панель инструментов с вкладками "File", "Edit", "Build" и кнопкой запуска новеллы.



Панель инструментов

Рис. 2.

3.2.2.1.1. Меню File

По нажатию на вкладку "File" открывается меню с опциями "New Project", "Open", "Save as", "Exit".



Меню вкладки "File"

Рис. 3.

1) "New Project" создает новый проект, очищаяя графы проекта и сцен.
2) "Open" открывает сохраненный ранее json-файл, содержащий проект.
3) "Save as" сериализует проект в json-файл.
4) "Exit" завершает работу приложения.

3.2.2.1.2. Меню Edit

По нажатию на вкладку "Edit"открывается меню с опциями "Copy", "Paste", "Delete".



Меню вкладки "Edit"

Рис. 4.

Все узлы кроме E-Узла можно выделить. Если ни один узел еще не был выделен, то выделение происходит нажатием левой кнопки мыши по узлу, иначе нужно зажать левый Ctrl и уже затем выделить новый узел нажатием левой кнопки мыши.

Снимается выделение нажатием на пустое пространство графа.

1) "Copy" копирует выделенные узлы.
2) "Paste" вставляет ранее скопированные узлы на текущий граф.
3) "Delete" удаляет выделенные узлы.

Для всех опций реализованы HotKeys:

1) "Copy" - "Ctrl + C"
2) "Paste" - "Ctrl + V"
3) "Delete" - "Delete"

3.2.2.1.3. Меню Build

По нажатию на вкладку "Build"открывается меню с опцией "Build".



Меню вкладки "File"

Рис. 5.

"Build" Сериализует игру в json-файл и сохраняет его в корневую папку конструктора.

3.2.2.1.4. Запуск игры

По нажатию на кнопку "Play" в оконном режиме запускается созданная игра.



Кнопка "Play"

Рис. 6.

Из режима вопроизведения игры можно выйти по нажатию на кнопку "To Editor"в верхнем левом углу.

3.2.2.2. Панель выбора переходов

Снизу от панели инструментов находится панель выбора переходов.



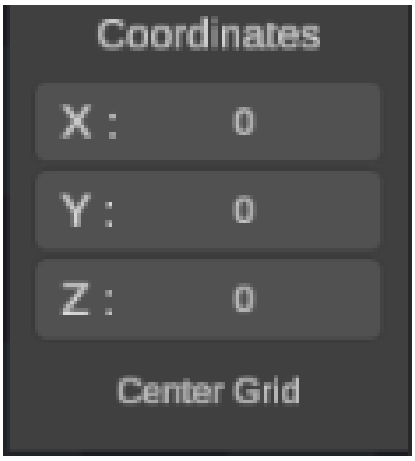
Панель выбора переходов

Рис. 7.

Левая кнопка отвечает за SMPL-переход, правая за CLK-переход.
SMPL-переход на графе выглядит как сплошная линия, соединяющая узлы.
CLK-переход на графе выглядит как прерывистая линия, соединяющая узлы.

3.2.2.3. Панель координат

В правом верхнем углу находится панель с координатами камеры, она помогает ориентироваться в графе, положительное значение X соответствует положению справа от центра координат, положительное значение Y соответствует положению сверху от центра координат. Координата Z принимает значение от 0 до 2700 и соответсвует масштабу графа, чем больше Z, тем дальше граф. Под значениями координат находится кнопка "Center Grid", которая возвращает камеру обратно в центр графа.



Панель координат

Рис. 8.

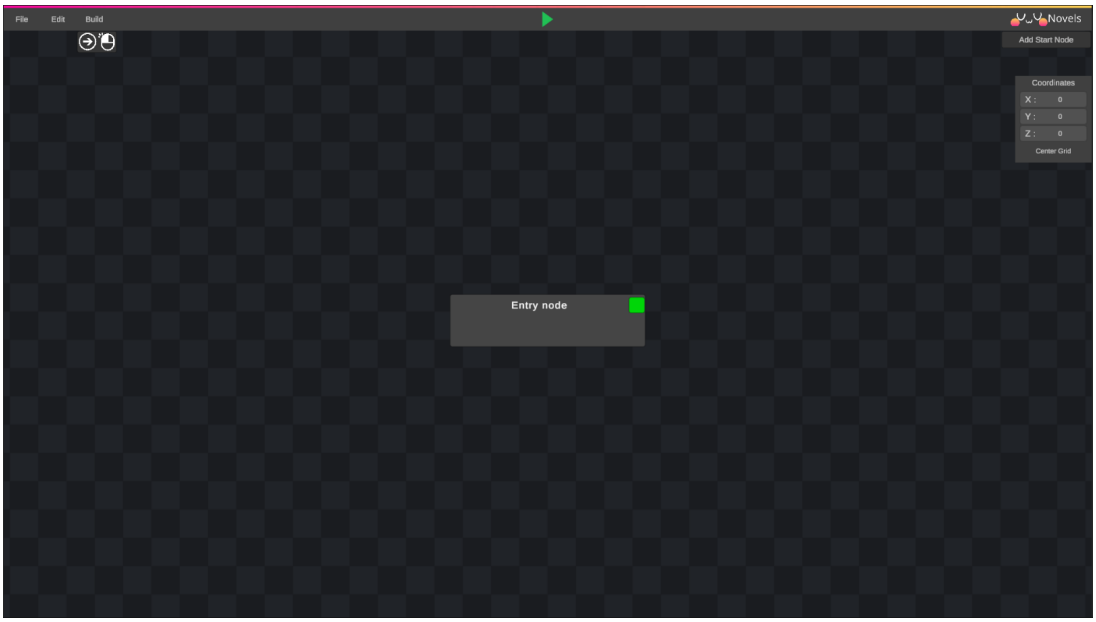
3.2.2.4. Граф

Пространство графа можно перемещать, двигая мышью с зажатым колесиком. Также можно увеличивать и уменьшать масштаб графа, зажав левый Ctrl и покрутив колесиком мыши.

3.2.2.5. Узлы

Все узлы соединяются визуально через области выхода (зеленый прямоугольник в верхнем правом углу узла) и области входа (красный прямоугольник в верхнем левом углу узла).

3.2.3. Окно проекта



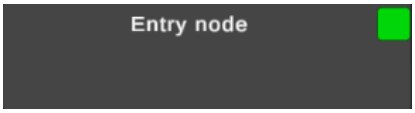
Окно редактирования проекта.

Рис. 9.

3.2.3.1. Узлы

3.2.3.1.1. E-узел

По умолчанию в графе проекта находится E-Узел, который невозможно удалить.



E-Узел

Рис. 10.

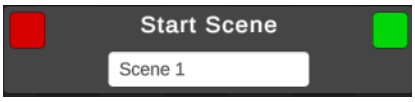
3.2.3.1.2. SS-узел

В графе проекта можно добавить SS-Узел, который создает новую сцену. В узле указывается название сцены для простоты ориентирования в графе проекта. На граф этой сцены можно перейти по двойному нажатию мыши на узел.



Кнопка добавления SS-Узла

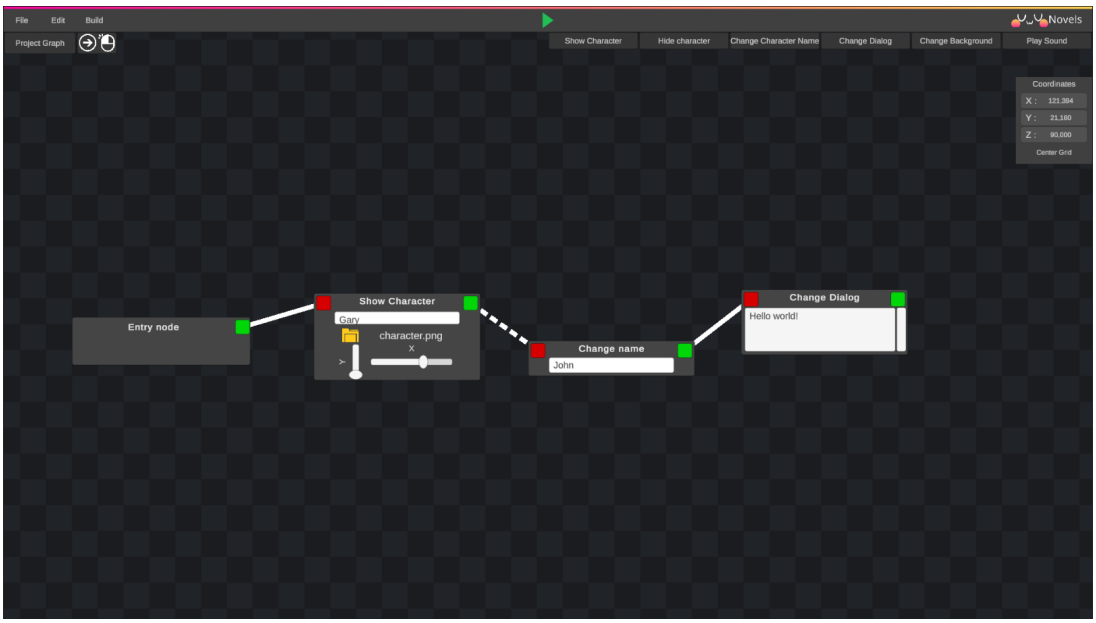
Рис. 11.



SS-Узел

Рис. 12.

3.2.4. Окно сцены



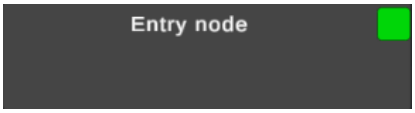
Окно редактирования сцены

Рис. 13.

3.2.4.1. Узлы

3.2.4.1.1. E-узел

По умолчанию в графе сцены находится Е-Узел, который невозможно удалить.

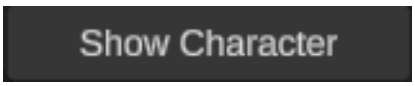


Е-Узел

Рис. 14.

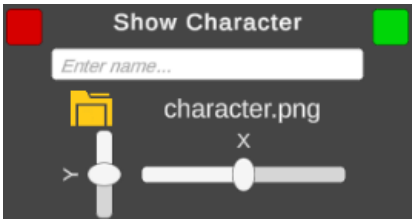
3.2.4.1.2. SC-узел

SC-Узел добавляет персонажа на сцену. Сперва вводится имя персонажа в пустое поле с подписью "Enter name..." , затем выбирается изображение персонажа по нажатию на кнопку с иконкой папки, которая открывает проводник, и пользователь выбирает файл. Также позицию персонажа по горизонтали и вертикали можно изменить горизонтальным и вертикальным ползунком соответственно.



Кнопка добавления SC-Узла

Рис. 15.



SC-Узел

Рис. 16.

3.2.4.1.3. HC-узел

HC-Узел убирает персонажа из сцены. Пользователю требуется только ввести

имя персонажа, которого нужно убрать. Для этого в узле есть поле для ввода текста с подписью "Enter name".



Кнопка добавления HC-Узла

Рис. 17.



HC-Узел

Рис. 18.

3.2.4.1.4. CD-узел

CD-Узел меняет текст в текущей сцене игры на указанный в узле. Для этого в узле есть поле для ввода текста с подписью "Enter speech".



Кнопка добавления CD-Узла

Рис. 19.



CD-Узел

Рис. 20.

3.2.4.1.5. CN-узел

CN-Узел меняет имя персонажа, который говорит в настоящий момент. Для этого в узле есть поле для ввода текста с подписью "Enter name".



Кнопка добавления CN-Узла

Рис. 21.



CN-Узел

Рис. 22.

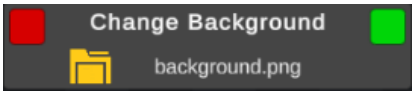
3.2.4.1.6. CBG-узел

CBG-Узел меняет фон в текущей сцене игры на указанный в узле. Для этого в узле есть кнопка с иконкой папки, по нажатию на которую открывается проводник, и пользователь выбирает файл.



Кнопка добавления CBG-Узла

Рис. 23.



CBG-Узел

Рис. 24.

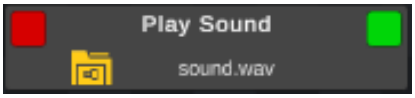
3.2.4.1.7. PS-узел

PS-Узел воспроизводит звуковой файл. Для этого в узле есть кнопка в виде папки с иконкой динамика, по нажатию на которую открывается проводник, и пользователь выбирает файл.



Кнопка добавления PS-Узла

Рис. 25.



PS-Узел

Рис. 26.

3.2.4.2. Переход к графу проекта

В левом верхнем углу находится кнопка "Project Graph", по нажатию на нее пользователь переходит к графу проекта.

3.3. Описание и обоснование выбора метода организации входных и выходных данных

3.3.1. Входные данные

3.3.1.1. Изображения

Изображения используются в двух узлах:

1) CBG-Узел
2) SC-Узел

Для CBG-Узла изображение используется для создания фона в сцене.

Для SC-Узла изображение используется для создания образа персонажа.

Все изображения подаются в формате "PNG". Данный формат подходит для создания слоев изображений, так как обладает встроенной прозрачностью. Например, изображение персонажа ставится поверх изображения фона. Также данный формат поддерживает большое количество цветов и высокое разрешение.

3.3.1.2. Звуковые файлы

Все звуковые файлы подаются в формате "WAV". Данный формат обладает высоким качеством звука.

Используется в PS-Узле для воспроизведения звуков в сцене.

3.3.1.3. Текстовые данные

Текстовые данные используются в следуюших узлах:

1) SS-Узел
2) CD-Узел
3) SC-Узел
4) HC-Узел
5) CN-Узел

Для SS-Узла пользователь вводит название сцены.

Для CD-Узла пользователь вводит новый диалог.

Для SC-Узла пользователь вводит имя персонажа.

Для HC-Узла пользователь вводит имя персонажа.

Для CN-Узла пользователь вводит имя персонажа.

3.3.2. Выходные данные

1) Отображение (запуск) созданной игры
2) Json-файл, содержащий игру
3) Json-файл, содержащий проект

3.4. Описание и обоснование выбора состава технических и программных средств

3.4.1. Unity

Основная причина выбора платформы Unity заключается в том, что он кросс-платформенн
что позовляет запускать конструктор и созданные на нем игры на таких операционных системах как macOS, Windows, Linux. Также Unity имеет большой инструментарий и свою библиотеку готовых ассетов.

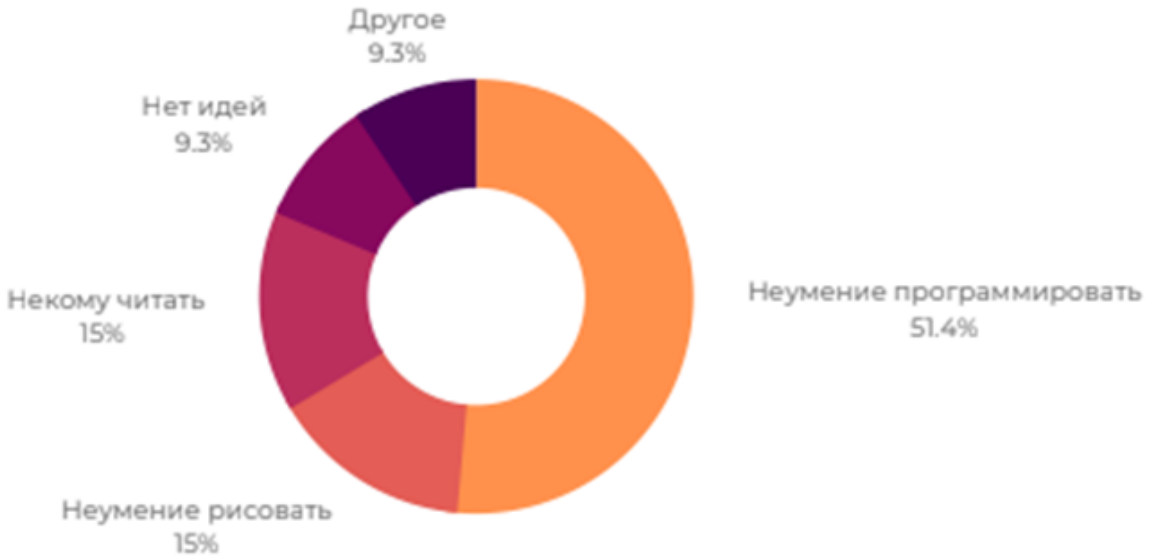
3.4.2. C#

Unity использует C# как основной язык программирования. Язык прост в освоении и обладает широким функционалом.

4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1. Предполагаемая потребность

Данный продукт предоставляет возможность создать и запустить новеллу людям, увлекающимся данным жанром игр, чтением книг, писательством или рисованием, но по тем или иным причинам, не желающим заниматься программирования. По результатам Customer Development были получены следующие ответы на вопрос о том, почему заинтересованные в создании визуальных новеллах люди отказываются их создавать.



4.2. Ориентировочная экономическая эффективность

Использование данного продукта позволит разработать новеллу, не используя языки программирования, а пользуясь интуитивно-понятной системой логических узлов, тем самым исключая потребность в деятельности программистов. Это предоставляет гейм-дизайнеру полную свободу действий, так как не требуется ждать завершения работы со стороны программистов.

4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

Прямыми конкурентами являются «Episode» [12], «Twine» [13], «RENPY» [14].
Преимущества по сравнению с Episode:

1) Возможность загружать свои изображения.
2) Наличие десктопной версии.

Преимущества по сравнению с Twine:

1) Наличие графической составляющей, а не только текстовые квесты.
2) Не требуется писать скрипты внутри текста.

Преимущества по сравнению с RENPY:

1) Не требует навыков программирования.

ПЕРЕЧЕНЬ ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19.101-77. ЕСПД. Виды программ и программных документов. — Москва : Стандартинформ, 2005.
2. ГОСТ 19.102-77. ЕСПД. Стадии разработки. — Москва : Стандартинформ, 2005.
3. ГОСТ 19.103-77. ЕСПД. Обозначение программ и программных документов. — Москва : Стандартинформ, 2005.
4. ГОСТ 19.104-78. ЕСПД. Основные надписи. — Москва : Стандартинформ, 2005.
5. ГОСТ 19.105-78. ЕСПД. Общие требования к программным документам. — Москва : Стандартинформ, 2005.
6. ГОСТ 19.106-78. ЕСПД. Требования к программным документам, выполненным печатным способом. — Москва : Стандартинформ, 2005.
7. ГОСТ 19.404-79. ЕСПД. Пояснительная записка. Требования к содержанию и оформлению. — Москва : Стандартинформ, 2005.
8. ГОСТ 19.603-78. ЕСПД. Общие правила внесения изменений. — Москва : Стандартинформ, 2005.
9. ГОСТ 19.604-78. ЕСПД. Правила внесения изменений в программные документаы, выполненные печатным способом. — Москва : Стандартинформ, 2005.
10. Цикл статей о инструменте визуального программирования Blueprint в движке Unreal Engine 4:
https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints.
11. Статья о работе с материалами в Blender:
https://render.ru/ru/XYZ/post/18769.
12. Платформа для создания визуальных новелл Episode:
https://www.episodeinteractive.com/.
13. Инструмент для создания текстовых квестов, визуальных новелл Twine:
https://twinery.org/.
14. Движок для создания визуальных новелл RenPy:
https://www.renpy.org/.

ПРИЛОЖЕНИЕ 1

Описание и функциональное назнаение классов

Таблица 1.

Класс	Функция
DragGrid	Отвечает за перемещение графа по осям X и Y. Перемещение происходит движенем мыши с зажатым колесиком мыши.
MovableGrid	Оптимизирует перемещение графа, перемещая удаленные от камеры ячейки сетки графа на новое пустое пространство.
ZoomGrid	Отвечает за перемещение графа по оси Z. Перемещение происходит прокруткой колесика мыши с зажатым левым Ctrl.
NewProject	Отвечает за полное очищение текущего графа проекта.
OpenProject	Отвечает за поиск в корневой папке конструктора и открытие сохранненого ранее json-файла.
SaveProject	Отвечает за сохранение проекта в json-файл в корневую папку конструктора.
ExitProject	Отвечает за закрытие констуктора.
CopySelectedNodes	Отвечает за копирование выделенных узлов.
PasteCopiedNodes	Отвечает за вставвку скопированных ранее узлов.
DeleteSelectedNodes	Отвечает за удаление выделенных ранее узлов.
KeyListener	Слушает нажатие Hotkeys для функций "Copy", "Paste", "Delete".
BuildNovel	Отвечает за сериализацию игры в json-файл и помещение данного файла в корневую папку конструктора.
SelectNode	Отвечает за выделение узлов, для первого узла пользователь должен кликнуть левой кнопкой мыши на узел, для двух и более он должен зажать левый Ctrl и уже затем кликнуть на узел.
ResetSelection	Снимает выделение со всех узлов, если пользователь кликнул левой кнопкой мыши на пустое пространство в графе.
SpawnEditorNode	Создает новый узел указанного типа на текущем графе.
SwitchToScene	Переходит на граф сцены, при двойном нажатии ЛКМ по SS-Узлу.
DragAndDrop	Инициализирует начало движения узла, меняет курсор.
DragAndDropMove	Перемещает узел в пространстве графа при движении мыщи и зажатой ЛКМ на узле.
FileOptionsManager	Отвечает за вызов указанной пользователем функции во вкадке "File".
EditOptionsManager	Отвечает за вызов указанной пользователем функции во вкадке "Edit".

BuildOptions-Manager	Отвечает за вызов указанной пользователем функции во вкадке "Build".
EditorNode	Является общей реализацией узлов, от него наследуется все остальные.
ActionEditorNode	Является общей реализацией узлов, которые выполняют действия в игре, не зависящие от входных данных игрока (кроме переходов), от него наследуется все остальные.
EntryEditorNode	Реализация E-Узла.
ChangeBackground-EditorNode	Реализация CBG-Узла.
ShowCharacter-EditorNode	Реализация SC-Узла.
HideCharacter-EditorNode	Реализация HC-Узла.
ChangeName-DialogueEditorNode	Реализация CN-Узла.
ChangeSpeech-DialogueEditorNode	Реализация CD-Узла.
StartScene-EditorNode	Реализация SS-Узла.
PlaySound-EditorNode	Реализация PS-Узла.
Commutative-Connection	Программная реализация соединения между узлами.
Editor-TransitionPoint	Программная реализация входной и выходной области.
Editor-TransitionVisual	Визуально создает или удаляет новый переход.
GraphTransition-VisualManager	Управляет визуельными переходами между всеми узлами.
LineConnector	Визуальная реализация перехода между двумя узлами.
GraphEditor-Visualizer	Отображает и скрывает графы, в зависимости от того, на каком графе пользователь в текущий момент.

ПРИЛОЖЕНИЕ 2

Описание и функциональное назнаение полей, методов и свойств классов

Таблица 2.

DragGrid				
Поля				
Имя	Модификатор	Тип	Функция	
grid	private	Transform	Хранит положение сетки графа.	
xCoordinateLabel	private	GameObject	Хранит значение координаты X графа.	
yCoordinateLabel	private	GameObject	Хранит значение координаты Y графа.	
dragCursor	private	Texture2D	Хранит курсор перемещения графа.	
panSpeed	private	float	Хранит скорость перемещения графа.	
_oldOrigin	private	Vector3	Хранит старое положение графа.	
_newOrigin	private	Vector3	Хранит новое положение графа.	
_clicked	private	Vector3	Хранит состояние колесика мыши, True - колесико нажато, False - нет.	
Свойства				
Имя	Модификатор	Тип	Функция	
Graph	public	Transform	Getter и Setter для положения графа в пространстве.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
Update	private	void	none	При нажатии пользователем колесика мыши и движении этой мыши меняет положение камеры в графе по осям X и Y.
ResetPosition	public	void	none	Сбрасывает положение камеры, возвращая ее в центр координат.

Таблица 3.

MovableGrid				
Поля				
Имя	Модификатор	Тип	Функция	
grid	private	Transform	Хранит положение сетки графа. Граф состоит из 9 квадратных ячеек.	
upperLeftPlane	private	Transform	Хранит положение левой верхней ячейки.	
upperPlane	private	Transform	Хранит положение верхней ячейки.	
upperRightPlane	private	Transform	Хранит положение правой верхней ячейки.	
leftPlane	private	Transform	Хранит положение левой ячейки.	
middlePlane	private	Transform	Хранит положение средней ячейки.	
rightPlane	private	Transform	Хранит положение правой ячейки.	
lowerLeftPlane	private	Transform	Хранит положение левой нижней ячейки.	
lowerPlane	private	Transform	Хранит положение нижней ячейки.	
lowerRightPlane	private	Transform	Хранит положение правой нижней ячейки.	
Свойства				
Имя	Модификатор	Тип	Функция	
Planes	public	Transform[,]	Getter и Setter для положения ячеек графа.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
Start	private	void	none	Первоначальная инициализация переменных класса.

Update	private	void	none	При перемещении камеры двигает удаленные от камеры ячейки сетки на приближающиеся пустые места.
movingRight	private	void	none	При перемещении камеры вправо передвигает левые ячейки сетки на новые места справа от сетки.
movingLeft	private	void	none	При перемещении камеры влево передвигает правые ячейки сетки на новые места слева от сетки.
movingUp	private	void	none	При перемещении камеры вправо передвигает нижние ячейки сетки на новые места сверху от сетки.
movingDown	private	void	none	При перемещении камеры вниз передвигает верхние ячейки сетки на новые места снизу от сетки.

Таблица 4.

ZoomGrid			
Поля			
Имя	Модификатор	Тип	Функция
grid	private	Transform	Хранит положение сетки графа.
zCoordinateLabel	private	GameObject	Хранит значение координаты Z графа.
_zoomSpeed	private	float	Хранит скорость масштабирования графа.
maxDistance	private	int	Является коснтантой, хранит максимальный масштаб сетки.

minDistance	private	int	Является коснтантой, хранит минимальный масштаб сетки.	
Свойства				
Имя	Модификатор	Тип	Функция	
Graph	public	Transform	Getter и Setter для положения графа в пространстве.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
Update	private	void	none	Масштабирует граф, если пользователь зажал левый Ctrl и подвигал колесиком мыши.
UpdateNewGraph	private	void	none	Устанавливает старый масштаб при переходе на другой граф.

Таблица 5.

NewProject				
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
CreateNewProject	public	void	none	Очищает граф проекта, создавая пустой проект.

Таблица 6.

OpenProject				
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
CreateNewProject	public	void	none	Очищает граф проекта, создавая пустой.

Таблица 7.

SaveProject				
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
SaveDebugProject	public	void	none	Сериализует проект в json-файл.

Таблица 8.

ExitProject				
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
Exit	public	void	none	Закрывает приложение конструктора.

Таблица 9.

CopySelectedNodes			
Свойства			
Имя	Модификатор	Тип	Функция
CopiedNodes	public get;private set	List-<GameObject>	Getter и Setter для скопированных узлов графа.
Методы			
Имя	Модификатор	Тип	Аргументы	Функция
CopyNodes	public	void	none	Добавляет выделенные пользователем узлы в список скопированных.

Таблица 10.

PasteCopiedNodes			
Поля			
Имя	Модификатор	Тип	Функция

parent	private	GameObject	Граф, в который вставляются скопированные узлы.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
PasteNodes	public	void	none	Создает новый префаб для каждого скопированного узла в текущем графе.

Таблица 11.

DeleteSelectedNodes				
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
DeleteNodes	public	void	none	Удаляет все выделенные узлы и очищает список выделенных узлов.

Таблица 12.

KeyListener			
Поля			
Имя	Модификатор	Тип	Функция
editCopyScript	private	CopySelected-Nodes	Скрипт, вызываемый для копирования узлов.
editPasteScript	private	PasteCopied-Nodes	Скрипт, вызываемый для вставки узлов.
editDeleteScript	private	DeleteSelected-Nodes	Скрипт, вызываемый для удаления узлов.
Методы			
Имя	Модификатор	Тип	Аргументы	Функция

Update	private	void	none	Слушает нажатие Hotkeys для функций копирования, вставки и удаления узлов.

Таблица 13.

BuildNovel				
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
BuildAndSaveNovel	public	void	none	Сериализует игру в json-файл и кладет его в корневую папку игры.

Таблица 14.

SelectNode			
Поля			
Имя	Модификатор	Тип	Функция
node	private	GameObject	Узел, на который вешается скрипт.
Свойства			
Имя	Модификатор	Тип	Функция
selectedNodes	public get;private set	List-<GameObject>	Getter и Setter для выделенных узлов графа.
Методы			
Имя	Модификатор	Тип	Аргументы	Функция
RemoveOutline	public	void	none	Снимает визуальное выделение (белую рамку) с выделенных узлов.

OnPointerClick	public	void	Pointer-EventData	Реализует метод интерфейса IPointerClickHandler, после клика пользователя на узел вызывается данный метод. Он добавляет выделенные ноды в список и визуально выделяет их (создает белую рамку по краям узла).

Таблица 15.

ResetSelection				
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
OnPointerClick	public	void	Pointer-EventData	Реализует метод интерфейса IPointerClickHandler, после клика пользователя на пустое место графа, снимает выделение со всех узлов.

Таблица 16.

SpawnEditorNode				
Поля				
Имя	Модификатор	Тип	Функция	
prefab	private	GameObject	Префаб узла, который будет создан.	
parent	private	GameObject	Граф, в котором будет создан префаб выбранного узла.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция

CreateNode	public	void	none	Создает указанный пользователем префаб узла в текущем графе.

Таблица 17.

SwitchToScene				
Поля				
Имя	Модификатор	Тип	Функция	
_sceneEditorNode	private	StartScene-EditorNode	Сцена, в которую осуществляется переход.	
_clickNumber	private	int	Число нажатий ЛКМ.	
_clicktime	private	flaot	Время нажатия ЛКМ в первый раз.	
_clickdelay	private	flaot	Допустимая задержка по времени для нажатия ЛКМ во второй раз.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
Awake	private	void	none	Инициализирует поле со сценой.
OnPointerDown	public	void	Pointer-EventData	Реализует интерфейс IPointerDownHandler, вызывается, если пользователь кликнул ЛКМ по SS-Узлу, при двойном нажатии запускает граф выбранной сцены.
LoadSceneUI	private	void	none	Запускает граф выбранной сцены.

Таблица 18.

DragAndDrop			
Поля			
Имя	Модификатор	Тип	Функция

beginDragEvent	private	UnityEvent-<Pointer-EventData>	Хранит функции, запускаемые при начале перемещения узла.	
dragTickEvent	private	UnityEvent-<Pointer-EventData>	Хранит функции, запускаемые при перемещении узла.	
endDragEvent	private	UnityEvent-<Pointer-EventData>	Хранит функции, запускаемые при окончании перемещения узла.	
dragCursor	private	Texture2D	Курсор для перемещения узлов.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
OnBeginDrag	public	void	Pointer-EventData	Меняет курсор на курсор перемещения узла, вызывает функции, запускаемые при начале перемещения узла.
OnDrag	public	void	Pointer-EventData	Вызывает функции, запускаемые при начале перемещения узла.
OnEndDrag	public	void	Pointer-EventData	Меняет курсор на стандартный, вызывает функции, запускаемые при начале перемещения узла.

Таблица 19.

DragAndDropMove			
Поля			
Имя	Модификатор	Тип	Функция
isWorldCoordinates	private	bool	Хранит систему координат, в которой перемещаются узлы, True - глобальная, False - локальная.

shouldCenterObject	private	bool	Хранит состояние центрирования узла, True - при начале перемещения центр узла нужно переместить к курсору, False - нет.	
_positionDelta	private	Vector3	Расстояние от центра узла до курсора.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
StartMove	public	void	Pointer-EventData	Сохраняет старое положение узла в графе.
Move	public	void	Pointer-EventData	Перемещает узел на расстояние, равное разнице между старым и новым.

Таблица 20.

FileOptionsManager				
Поля				
Имя	Модификатор	Тип	Функция	
filePage	private	GameObject	Хранит вкладку "File".	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
HandleValue	public	void	none	Вызывает выбранную пользователем функцию из доступных функций "File".
OnPointerClick	public	void	Pointer-EventData	Реализует метод интерфейса IPointerClickHandler, срабатывает при нажатии на одну из опций владки "File".

Таблица 21.

EditOptionsManager
Поля

Имя	Модификатор	Тип	Функция	
editPage	private	GameObject	Хранит вкладку "Edit".	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
HandleValue	public	void	none	Вызывает выбранную пользователем функцию из доступных функций "Edit".
OnPointerClick	public	void	Pointer-EventData	Реализует метод интерфейса IPointerClickHandler, срабатывает при нажатии на одну из опций владки "Edit".

Таблица 22.

BuildOptionsManager				
Поля				
Имя	Модификатор	Тип	Функция	
buildPage	private	GameObject	Хранит вкладку "Build".	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
HandleValue	public	void	none	Вызывает выбранную пользователем функцию из доступных функций "Build".
OnPointerClick	public	void	Pointer-EventData	Реализует метод интерфейса IPointerClickHandler, срабатывает при нажатии на одну из опций владки "Build".

Таблица 23.

EditorNode
Свойства

Имя	Модификатор	Тип	Функция	
TransitionPointIn	public	Editor-TransitionPoint	Getter и Setter для входной области.	
TransitionPointOut	public	Editor-TransitionPoint	Getter и Setter для выходной области.	
Node	public	Node	Getter для текущего узла.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
InitNode	public	void	Node	Загружает информацию об узле из абстрактного класса, используется при сериализации.
SetPosition	кзшмфеу	void	Vector2D	Устанавливает положение узла в пространстве на основе данных из абстрактного узла.

Таблица 24.

EntryEditorNode			
Свойства			
Имя	Модификатор	Тип	Функция
Node	public	Node	Getter для текущего узла.

Таблица 25.

ChangeBackgroundEditorNode				
Поля				
Имя	Модификатор	Тип	Функция	
backgroundPath	private	string	Хранит путь к фону.	
spriteText	private	TextMesh-ProUGUI	Подпись в узле, которая меняется на путь к фону.	
Свойства				
Node	public	Node	Getter для текущего узла.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция

ChooseSprite-Dialogue	public	void	Node	Вызывает проводник, пользователь выбирает файл для загрузки на фон.
InitNode	public	void	Node	Загружает информацию об узле из абстрактного класса, используется при сериализации.

Таблица 26.

ShowCharacterEditorNode				
Поля				
Имя	Модификатор	Тип	Функция	
characterName	private	string	Хранит имя персонажа.	
spritePath	private	string	Хранит путь к изображению персонажа.	
color	private	Color	Хранит цвет имени персонажа в диалоговом окне	
nameInputField	private	TMP_-InputField	Поле для ввода имени персонажа.	
spriteText	private	TextMesh-ProUGUI	Поле с текстом, содержащим путь к изображению персонажа.	
sliderPosX	private	Slider	Горизонтальный слайдер, указывает положение героя по оси X.	
sliderPosY	private	Slider	Вертикальный слайдер, указывает положение героя по оси Y.	
pos	private	Vector3	Хранит положение персонажа в координатах X и Y.	
Свойства				
Node	public	Node	Getter для текущего узла.	
GetCharacter	public	Character	Getter для персонажа.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
SetName	public	void	string	Устанавливает имя персонажу.

SetPosXSlider	public	void	none	Устанавливает положение X персонажу.
SetPosYSlider	public	void	none	Устанавливает положение Y персонажу.
ChooseSprite-Character	public	void	none	Вызывает проводник, пользователь выбирает файл для загрузки на изображение персонажа.
InitNode	public	void	Node	Загружает информацию об узле из абстрактного класса, используется при сериализации.

Таблица 27.

HideCharacterEditorNode				
Поля				
Имя	Модификатор	Тип	Функция	
characterName	private	string	Хранит имя персонажа.	
nameInputField	private	TMP_-InputField	Поле для ввода имени персонажа.	
Свойства				
Node	public	Node	Getter для текущего узла.	
GetCharacter	public	Character	Getter для персонажа.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
SetName	public	void	string	Устанавливает имя персонажа, которого надо скрыть.
InitNode	public	void	Node	Загружает информацию об узле из абстрактного класса, используется при сериализации.

Таблица 28.

ChangeSpeechDialogueEditorNode				
Поля				
Имя	Модификатор	Тип	Функция	
dialogueSpeech	private	string	Хранит текущую речь.	
inputField	private	TMP_-InputField	Поле для ввода речи.	
Свойства				
Node	public	Node	Getter для текущего узла.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
SetSpeech	public	void	string	Устанавливает текущую речь.
InitNode	public	void	Node	Загружает информацию об узле из абстрактного класса, используется при сериализации.

Таблица 29.

ChangeNameDialogueEditorNode				
Поля				
Имя	Модификатор	Тип	Функция	
dialogueName	private	string	Хранит имя того персонажа, который говорит в данный момент.	
inputField	private	TMP_-InputField	Поле для ввода имени.	
Свойства				
Node	public	Node	Getter для текущего узла.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
SetName	public	void	string	Устанавливает имя персонажа, который произносит речь.

InitNode	public	void	Node	Загружает информацию об узле из абстрактного класса, используется при сериализации.

Таблица 30.

StartSceneEditorNode				
Поля				
Имя	Модификатор	Тип	Функция	
sceneNameInputField	private	TMP_InputField	Поле для ввода названия сцены.	
_scene	private	NovelScene	Хранит объект сцены.	
Свойства				
SceneName	public	string	Getter и Setter для названия сцены.	
Scene	public	NovelScene	Getter и Setter для объекта сцены.	
Node	public	Node	Getter для текущего узла.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
Awake	private	void	none	Инициализирует название сцены значением поля для ввода названия сцены.
OpenSceneInEditor	public	void	string	Открывает текущую сцену.
InitNode	public	void	Node	Загружает информацию об узле из абстрактного класса, используется при сериализации.

Таблица 31.

PlaySoundEditorNode			
Поля			
Имя	Модификатор	Тип	Функция
soundPath	private	string	Хранит путь к звуковому файлу.

soundPathText	private	TextMesh-ProUGUI	Поле с путем к файлу.	
Свойства				
GetSound	public	Audio	Getter для звукового файла.	
Node	public	Node	Getter для текущего узла.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
ChooseSound-Dialogue	public	void	none	Открывает проводник для выбора файла.
InitNode	public	void	Node	Загружает информацию об узле из абстрактного класса, используется при сериализации.

Таблица 32.

CommutativeConnection				
Свойства				
Object1	public	Audio	Getter и Setter для первого элемента соединения.	
Object2	public	Node	Getter и Setter для второго элемента соединения.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
DoConnectThese-Objects	public	bool	T, T	Проверяет, образует ли оба аргумента метода данный соединения.
Equals	protected	bool	Commutative-Connection<T>	Проверяет равенство объектов в соединении.
Equals	public	bool	object	Проверяет равенство соединений.
GetHashCode	public	int	none	Возвращает хеш-код.

Таблица 33.

EditorTransitionPoint
Поля

Имя	Модификатор	Тип	Функция	
isPointIn	private	bool	True - входная область, False - выходная область.	
_transitionVisualManager	private	GraphTransitionVisualManager	Ссылка на менеджер переходов.	
_transition	private	Transition	Переход, связанный с этой областью.	
Свойства				
ConnectedPoint	public	EditorTransitionPoint	Getter и Setter для смежной области.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
Start	private	void	none	Инициализация менеджера визуальных переходов.
GetEditorNode	public	EditorNode	none	Возвращает узел, соответствующий данной области.
GetNode	public	Node	none	Возвращает абстракцию узла.
GetTransition	public	Transition	none	Создает переход между текущим и соединенным с ним узлом.
UpdateEndPoint	public	void	PointerEventData	Пытается найти область под курсором, при нахождении образует переход.
GetPossiblePointPointer	private	EditorTransitionPoint	PointerEventData	Возвращает область входа или выхода, находящуюся под курсором.
ConnectToPoint	public	void	EditorTransitionPoint, bool, Transition	Возвращает область входа или выхода, находящуюся под курсором.
ConnectToPoint	public	void	EditorTransitionPoint, Transition	Возвращает область входа или выхода, находящуюся под курсором.

ResetConnection	public	void	none	Удаляет соединение, смежное с этой областью.

Таблица 34.

EditorTransitionVisual				
Поля				
Имя	Модификатор	Тип	Функция	
_transition-VisualManager	private	Graph-Transition-VisualManager	Ссылка на менеджер визуальных соединений.	
Свойства				
ConnectedPoint	public	Editor-TransitionPoint	Getter и Setter для смежной области.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
Start	private	void	none	Инициализация менеджера визуальных переходов.
StartPreview-EndPoint	public	Transition	none	Отображает визуально переход, который заканчивается на курсоре.
EndPreview-EndPoint	public	void	Pointer-EventData	Удаляет переход, если пользователь отпустил ЛКМ.

Таблица 35.

GraphTransitionVisualManager			
Поля			
Имя	Модификатор	Тип	Функция
connectionPrefabs	private	List-<GameObject>	Префабы переходов.
connectionsScene-Spawn	private	Transform	Граф сцены, в которой появляются новые переходы.

connectionsNovel-Spawn	private	Transform	Граф проекта, в котором появляются новые переходы.	
selectedTransition-Index	public	int	Тип текущего перехода, 0 - SMPL, 1 - CLK.	
_transitions	private	List <Transition>	Список возможных переходов.	
_connections	private	Dictionary< Commutative-Connection <Transform>, GameObject>	Отображения соединение - узел.	
_connectionTo-Mouse	private	GameObject	True - соединен с курсором, False - соединен с узлом.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
Awake	private	void	none	Инициализация менеджера визуальных переходов.
GetSelected-Transition	public	Transition	none	Возвращает текущий выбранный переход.
GetSelected-TransitionPrefab	public	GameObject	none	Возвращает префаб текущего выбранного перехода.
GetSelected-TransitionPrefab	public	GameObject	none	Возвращает префаб текущего выбранного перехода.
GetTransition-Prefab	public	GameObject	Transition	Возвращает префаб по переданному переходу.
GetCurrent-Transitions- Parent	public	Transform	none	Возвращает граф, в который добавляются переходы в данный момент.
GetTransitions-Parent	public	Transform	bool	Возвращает граф, в который добавляются переходы в данный момент исходя из аргумента функции.

ConnectNodes-AndTheirPrefabs	public	void	Dictionary <Node, EditorNode>, bool	Отрисовывает все существующие переходы между узлами.
ConnectNodes-AndTheirPrefabs	public	void	Dictionary <Node, EditorNode>, bool	Отрисовывает все существующие переходы между узлами.
AddConnection	public	void	Transform, Transform, Transition	Соединяет два узла указанным переходом.
AddConnection	public	void	Transform, Transform, Transition, bool	Соединяет два узла указанным переходом на указанном графе.
ClearConnection	public	void	Transform, Transform	Удаляет переход между двумя узлами.
ClearAllConnections	public	void	Transform, Transform	Удаляет все переходы, которые связаны с данным узлом.
ClearAllScene-Connections	public	void	none	Удаляет все переходы на графе сцены.
ClearAllProject-Connections	public	void	none	Удаляет все переходы на графе проекта.
ClearDeleted-Connections	private	void	none	Очищает словарь, удаляет все значения null.
AddMouse-Connection	public	void	none	Добавляет переход от узла до курсора.
ClearMouse-Connection	private	void	none	Удаляет переход от узла до курсора.
SpawnConnection-Object	private	GameObject	Transform, Transform, Transition	Создает префаб перехода между двумя указанными узлами.
SpawnConnection-Object	private	GameObject	Transform, Transform, Transition, Transform	Создает префаб перехода между двумя указанными узлами в указанном графе.

SpawnConnection-Object	private	GameObject	Transform	Создает префаб перехода между указанным узлом и курсором.

Таблица 36.

LineConnector				
Поля				
Имя	Модификатор	Тип	Функция	
object1	private	Transfrom	Область входа или выхода.	
object2	private	Transfrom	Область входа или выхода.	
connectToMouse	private	bool	True - переход от узла к курсору, False - переход от узла к узлу.	
transition	private	Transition	Переход.	
_oldObject1Position	private	Vector3	Старое положение 1 области.	
_oldObject1Position _lineRenderer	private private	Vector3 LineRenderer	Старое положение 2 области. Визуальное представление перехода.	
Свойства				
Object1	public	Transform	Getter и Setter для первой области.	
Object2	public	Transform	Getter и Setter для второй области.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
Awake	private	void	none	Инициализация _lineRenderer.
Update	private	void	none	Отрисовывает визуально переход в реальном времени.
GetVectorFixZ	private	Vector3	Vector3	Устанавливает координату Z, соответствующую первой области.

Таблица 37.

GraphEditorVisualizer			
Поля			
Имя	Модификатор	Тип	Функция
dragGrid	private	DragGrid	Скрипт перемещения камеры по графу.

zoomGrid	private	ZoomFrid	Скрипт масштабирования графа.	
sceneUI	private	GameObject	UI сцены.	
graphUI	private	GameObject	UI графа.	
sceneGraph	private	GameObject	Узлы и переходы сцены.	
projectGraph	private	GameObject	Узлы и переходы графа.	
Свойства				
Object1	public	Transform	Getter и Setter для первой области.	
Object2	public	Transform	Getter и Setter для второй области.	
Методы				
Имя	Модификатор	Тип	Аргументы	Функция
ShowSceneGraph	public	void	none	Меняет UI графа на UI сцены.
ShowProjectGraph	public	void	none	Меняет UI сцены на UI графа.

Лист регистрации изменений									
Номера листов (страниц)					Всего листов (страниц) в докум.	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
Изм.	измененных	замененных	новых	аннулированных					