

## Содержание

Введение	5
1 Нормативные ссылки	6
2 Постановка задачи	7
2.1 Общее описание приложения	7
2.2 Описание функций приложения	7
3 Проектирование форм приложения	8
4 Проектирование вариантов использования приложения	17
5 Проектирование структуры приложения	20
5.2 Проработка активностей редактора	21
5.3 Описание классов и методов	22
5.4 Выбор среды для дальнейшей разработки	25
Заключение	26
Список используемой источников	27
ПРИЛОЖЕНИЕ А – Отчет о проверке на заимствования	28

## Введение

В ходе данного курсового проекта будет разработан проект модели программного приложения «Конструктор игровых сценариев».

Для решения задачи проектирования программного обеспечения будет использовано UML (унифицированный язык моделирования). Который позволит решить такие кейсы в проектировании как визуализация, структурирование приложения, построить поведенческую модель.

Основным результатом проектирования системы будет визуальная и описательная части проекта, которая позволит на ее основе построить прототип приложения для дальнейшего принятия решений в развитии разрабатываемого проекта.

## 1 Нормативные ссылки

В данных методических указаниях использованы ссылки на следующие стандарты:

- ГОСТ Р 1.5-2012. Стандартизация в Российской Федерации. Стандарты национальные. Правила построения, изложения, оформления и обозначения;
- ГОСТ Р 7.0.5-2008 СИБИД. Библиографическая ссылка. Общие требования и правила составления;
- ГОСТ Р 7.0.12-2011 СИБИД. Библиографическая запись. Сокращение слов и словосочетаний на русском языке. Общие требования и правила;
- ГОСТ 2.301-68 ЕСКД. Форматы;
- ГОСТ 7.9-95 СИБИД. Реферат и аннотация. Общие требования;
- ГОСТ 7.82-2001 СИБИД. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления.

## 2 Постановка задачи

### 2.1 Общее описание приложения

Требуется разработать проект программного приложения «Конструктор игровых сценариев». Приложение предназначено для упрощения разработки игровых сценариев и взаимодействия сценаристов и разработчиков игр, где есть сюжет и диалоги. В особенности если это касается ветвления сюжета.

Приложение представляет из себя простое оконное приложение, где можно создавать сценарии при помощи графического взаимодействия путем перетаскивания элементов логики или объектов сценария соединяя их между собой. Узлами в этом представлении будут логические объекты, диалог, фраза, либо иное действие касаемое изменение/перехода переменных (завязаны на логике).

### 2.2 Описание функций приложения

В приложении должен быть реализован следующий перечень функций необходимых в приложении:

- создание проекта сценария;
- создание первого акта в сценарии;
- логические блоки для нелинейного ветвления сюжета;
- узел диалога (ведется последовательное построение фраз персонажей);
- создание логических переменные для ветвления;
- модуль создание персонажей;
- контрольные точки для перехода между актами.

### 3 Проектирование форм приложения

На рисунке 1 изображен макет первоначальной формы. На первой форме есть список уже существующих, либо по нажатию кнопки добавления нового сценария меняется как на рисунке 2, где заводится проект с именем и местом хранения.

Меню

Список последних сценариев

The image shows a wireframe of a menu form. It consists of a large outer rectangle. At the top, there is a horizontal bar labeled 'Меню'. Inside the main area of the rectangle, there is a smaller, centered rectangle labeled 'Список последних сценариев'.

Рисунок 1 – Прототип формы «Меню»

Создание сценария

Путь к проекту

Назваоние

Рисунок 2 – Прототип формы «Меню» при создании сценария

На рисунке 3 отражен макет формы редактирования сценария. Основными его элементами является список актов. Актам задается последовательность перехода в сценарии (сверху в низ), задается путем перетаскивания актов на форме.

Кнопка «Редактирование персонажей» выдет на одноименную форму, где создаются персонажи которые взаимодействуют в сюжетной линии.

«Редактирование переменных» тоже кнопка ведущая на отдельную форму, где создаются переменные.

Меню	
Акт 1	<div>Редактирование персонажей</div> <div>Редактирование переменных</div>
Акт 2	
Акт 3	
Акт 4	
Акт 5	

Рисунок 3 – Прототип формы «Сценарий»

На рисунке 4 отражена форма редактирования акта. Форма представляет из себя полотно, на котором расставляются и связываются порядке при помощи графических элементов таких как выбор, условие, переходный узел, узел диалога.

Каждая из нод выполняет собственные функции макеты и описание узлов ниже будет далее

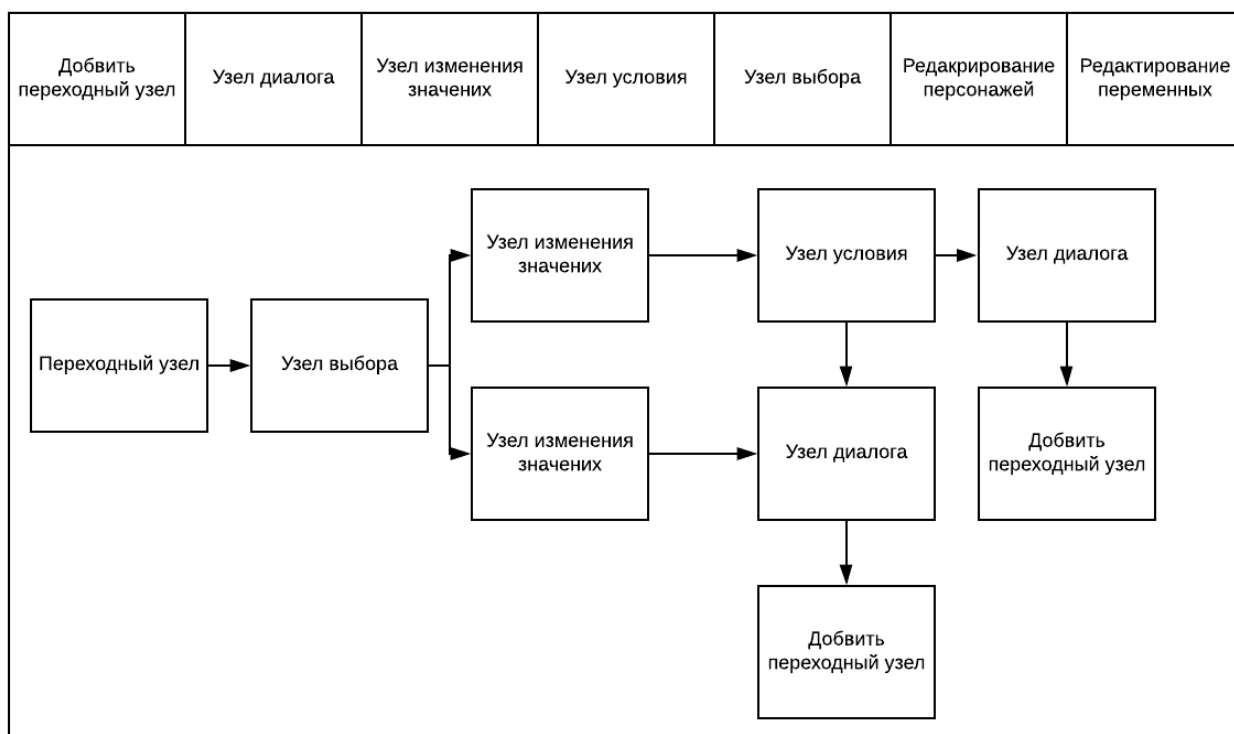


Рисунок 4 – Прототип формы «Акт»

Перходный узел представляет из себя элемет начала в акте. Ставится только в начале акта. Исполняет функцию направления сюжета при переходе из пердыдущих атов сценария. Имеет параметры для прохода по узлу и напраление на следующую ноду в дереве (рисунок 5).



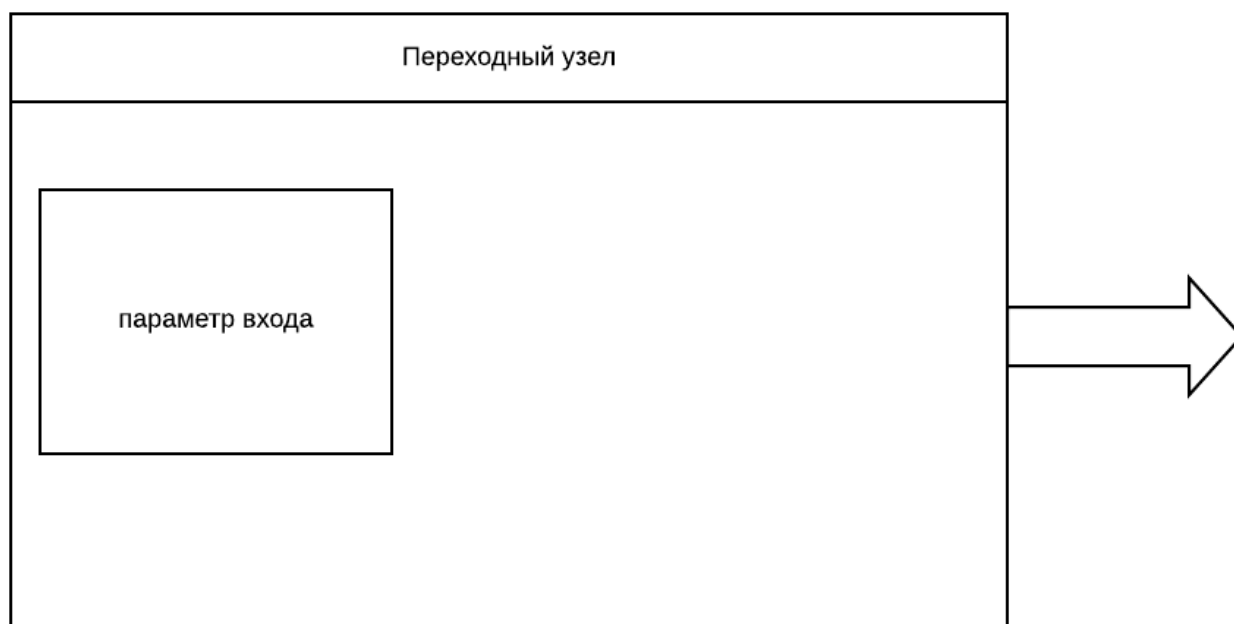


Рисунок 5 – Прототип формы «Переходный узел»

Узел диалога состоит из списка фраз и соответствующие персонажа и эмоции (рисунок 6). Пользователь на данном узле составляет диалог между двумя или группой персонажей. Сами эмоции привязаны к персонажам и выбирается из выпадающего списка.

У ноды диалога нет ветвлений и выбора, является как факт обработки логики и отражает уже выбранный путь в нелнейном сюжете игры, либо повествовании графической новеллы.

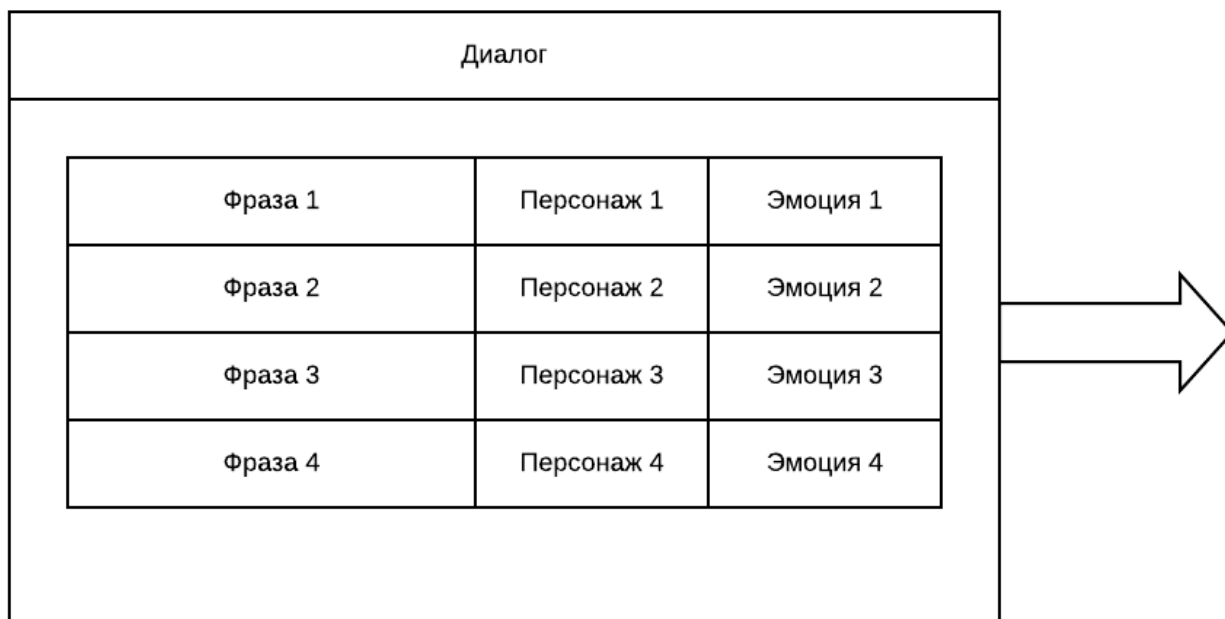


Рисунок 6 – Прототип формы «Диалог»

Узел условие представляет собой графический блок в котором есть список условий которые уходят в связанный маршрут (рисунок 7). Маршрут указывает на следующую ноду, которая может быть как диалогом, так и выбором.

Условия в ноде указывается как обычное сравнение переменных внесенных в соответствующей форме

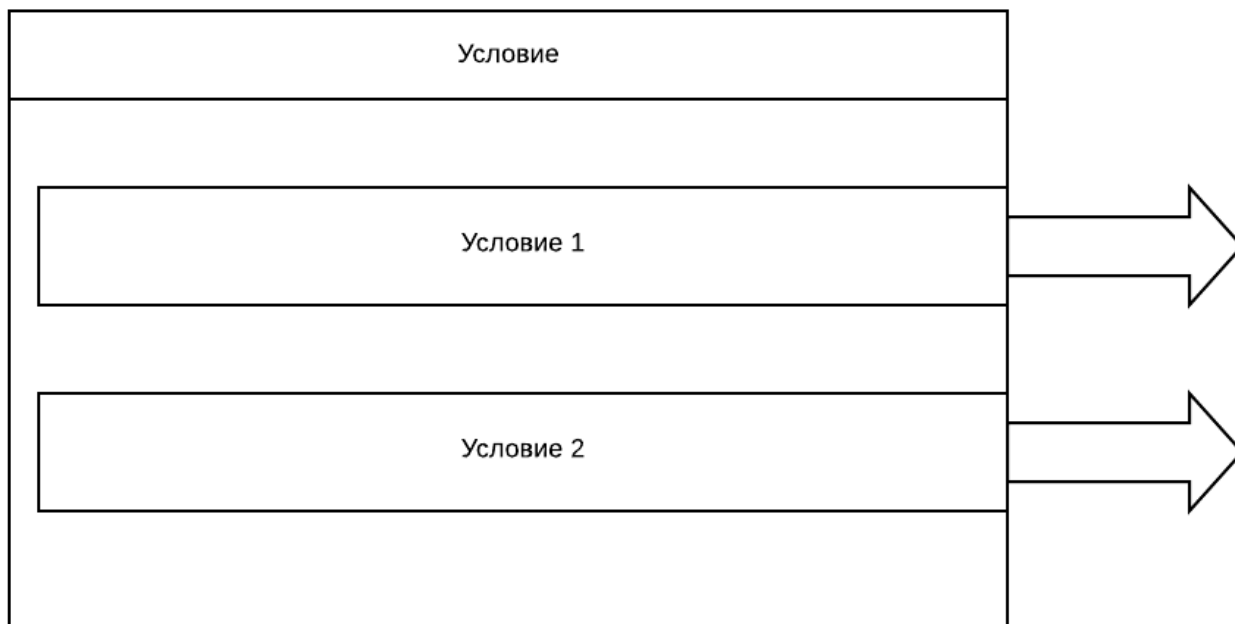


Рисунок 7 – Прототип формы «Условие»

Форма редактора персонажей (рисунок 8). На данной форме заносятся информация о субъектах участвующих в сценарии. Вход на форму есть из сценария и редактора акта.

У персонажей есть основные характеристики для их отличия это в повествования сюжета:

- имя;
- цвет;
- список эмоций;
- дополнительные параметры.

Редактирование персонажей	
<div>Имя</div>	<div>Дополнительные параметры</div>
<div>Цвет</div>	
<div>Эмоции</div>	

Рисунок 8 – Прототип формы «Персонажи»

Форма заведения переменных. Используются в действиях выборах, то что используется в проверка.

Основные параметры для переменной это ее название, тип и начальное значение. Три вида типов: счетчики, значения, флаги.

Переменные	
Тип переменной	начальное значение

Рисунок 9 – Прототип формы «Переменные»

#### 4 Проектирование вариантов использования приложения

Конструкция или стандартный элемент языка UML вариант использования применяется для спецификации общих особенностей поведения системы без рассмотрения внутренней структуры этой сущности.

Каждый вариант использования определяет последовательность действий, которые должны быть выполнены проектируемой системой при взаимодействии ее с соответствующим актером.

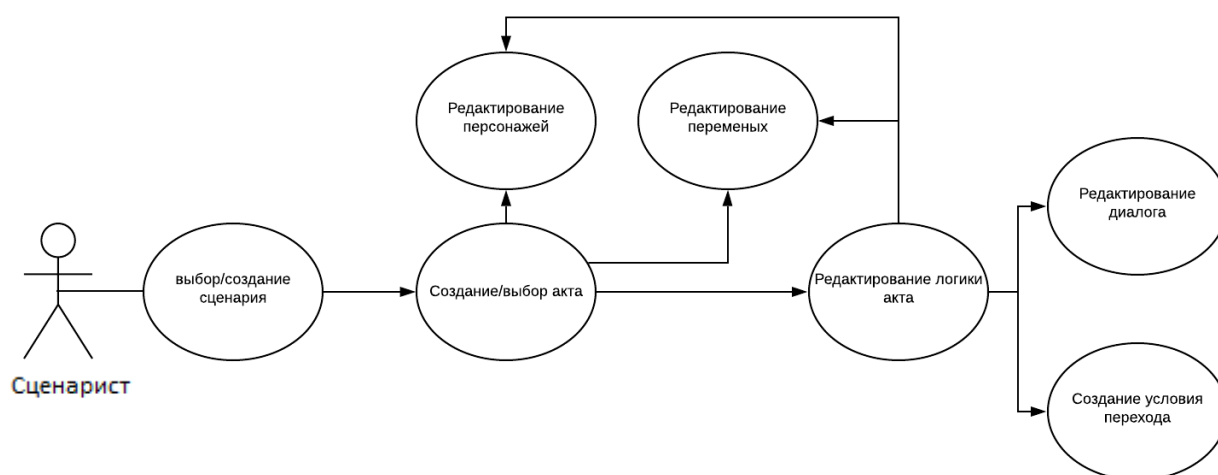


Рисунок 10 - Диаграмма вариантов использования

Поведение и использование форм пользователем практически линейно единственное доступ к редактору персонажей и переменных есть как с формы сценария, так и с редактора акта.

Переходы продемонстрированы на рисунке 11.

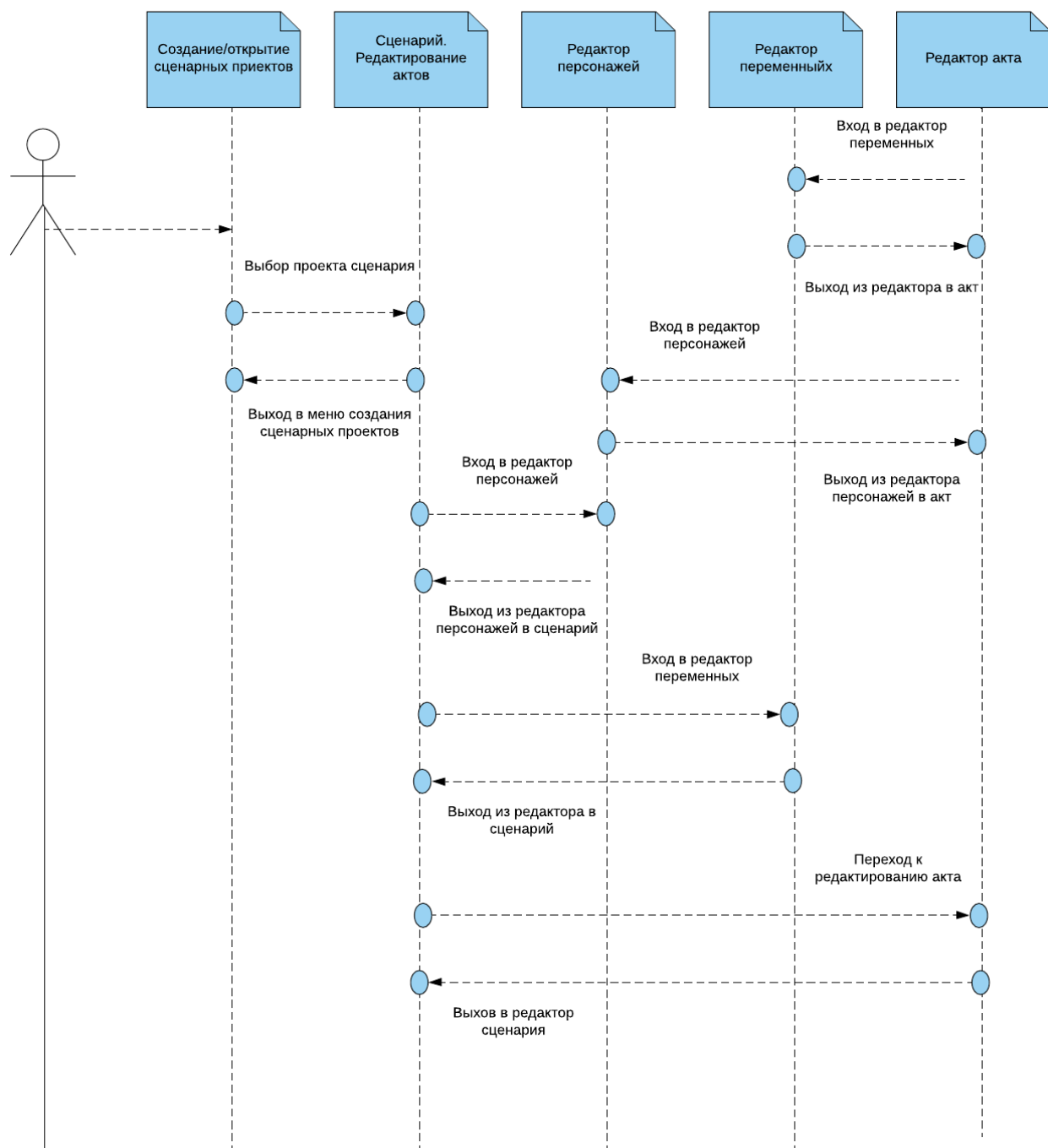


Рисунок 11 – Последовательность действий перехода по основным формам

Большая часть действий пользователя происходит на форме редактора акт. Пользователь выставляет узлы в зависимости от поведения сюжета и условий протекания тех или иных сюжетных поворотов.

Узлы соединяются между собой линиями которая показывает на последовательность действий сценария.

В разных типах узлах есть различное количество входящих выходящих связей. Типы узлов:

- начало акта;
- условие;
- выбор;
- диалог.



## 5 Проектирование структуры приложения

### 5.1 Составление классов и состояний в приложении

На рисунке 12 – Структура классов и методов, представлены описание классов и методов, которые необходимо реализовать в приложении.

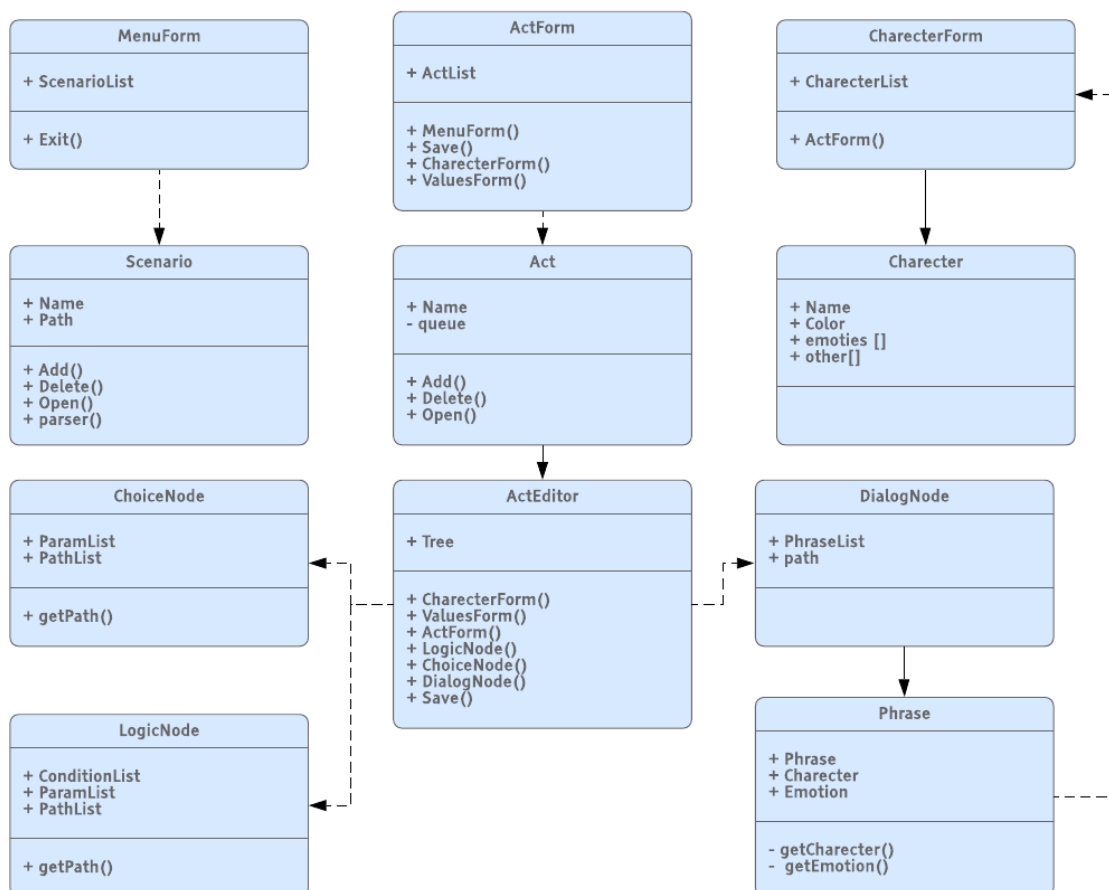


Рисунок 12 – Структура классов и методов

Есть два основных вида классов в структуре. Классы, отвечающие за формы и классы, которые отражают сущности. Класс меню связан с сущностью сценарий так как из меню создаются сценарии и создается проект.

Также уже из класса формы сценария порождаются акты, в которых в свою очередь выстраивается граф.

Изменение состояний проекта сценария отражены на рисунке 13.

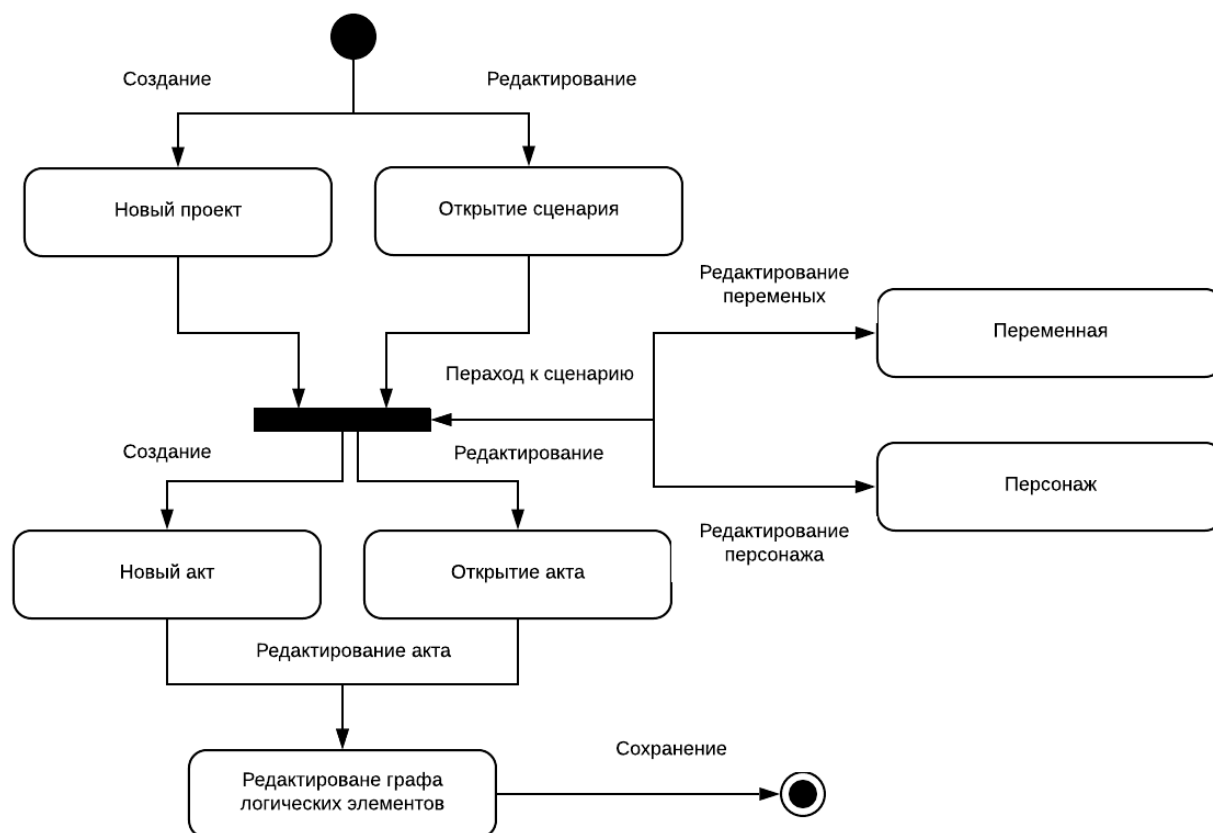


Рисунок 13 – Изменений состояния проекта сценария

## 5.2 Проработка активностей редактора

Редактора актов основная рабочая зона сценариста, именно здесь составляются диалоги персонажей логическое, составление логики перехода сюжета в виде древовидного графа.

В форме редактора актов производится добавление узлов графа, которые имеют различное назначение. Узлы делятся на:

- узел диалога;
- узел условия;

– узел выбора.

В связи с такой вариативностью нод необходимо проработать их состояния более детально. Составленная диаграмма активности (рисунок 14) покажет необходимую для реализации информацию в плане управления различными узлами графа.

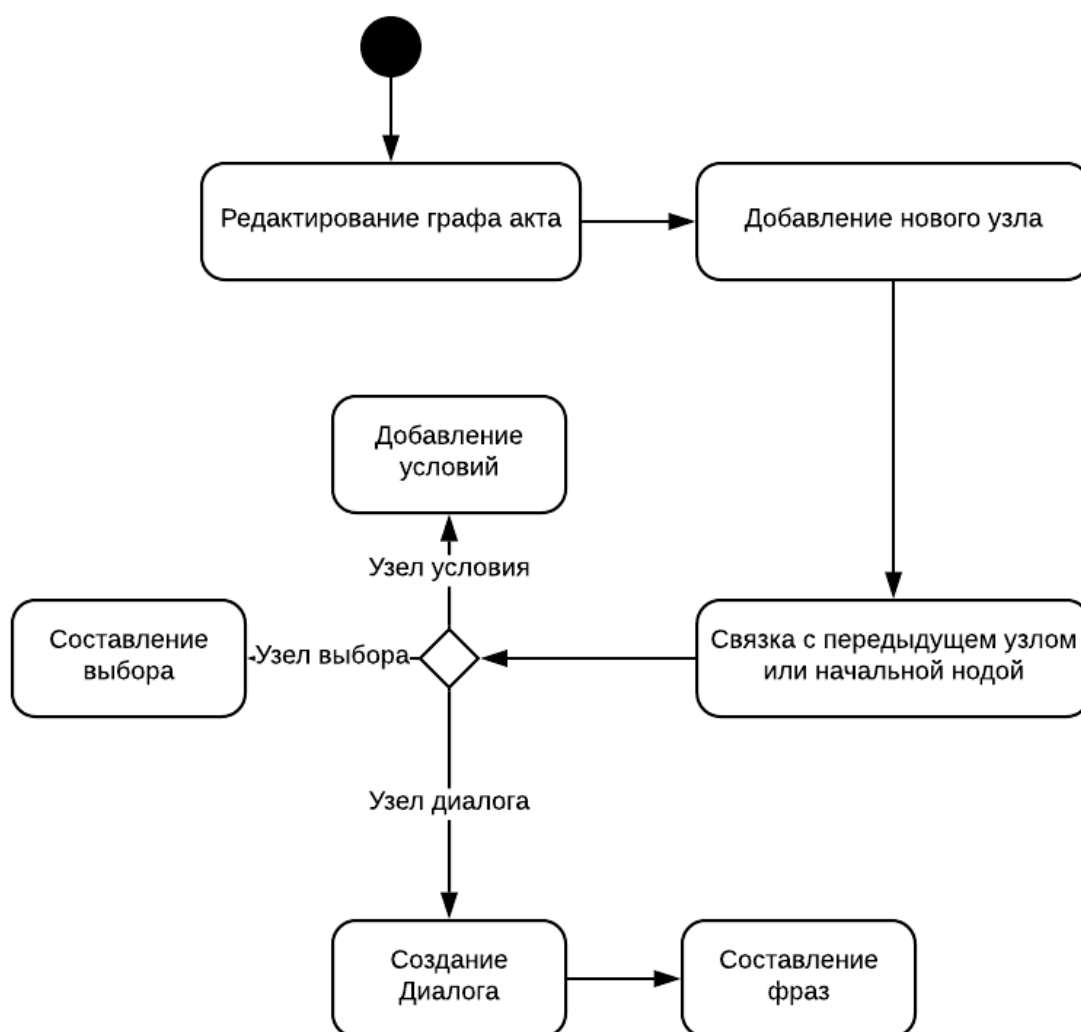


Рисунок 14 – Активности составления графа акта

### 5.3 Описание классов и методов

MenuForm – Форма меню. Основные действия — это создание, удаление, выбор проектов сценария.

ScenarioList – список сценариев.

Exit() – выход из приложения.

Scenario – класс олицетворяющий сущность класса.

Name – название сценария.

Path – путь хранения проекта.

Delete() - удаление объекта.

Open() – открытие сценария в форме сценариев ActForm

parser() – метод сохранения проекта для переноса данных в игровой проект.

ActForm – форма сценария.

ActList – список актов в сценарии.

MenuForm() - переход на форму меню.

Save() - сохранение проекта.

CharecterForm() – переход на форму редактор персонажей.

ValuesForm() – переход на форму редактор перчёных.

Act – Класс описывающий сущность акта.

Name – Название акта.

Queue – номер в очередь.

Delete() - удаление объекта.

Open() – открытие сценария в форме сценариев ActEditor.

ActEditor

CharecterForm() – переход на форму редактор персонажей.

ValuesForm() – переход на форму редактор перчёных.

ActForm() – переход на форму редактор сценария.

LogicNode() – обработка логического блока.

ChoiceNode() – обработка блока выбора.

DialogNode() – обработка ноды диалога.

Save() – Сохранение акта.

CharecterForm – Форма редактирования персонажей.

CharecterList – список персонажей.

ActForm() – переход на форму сценария.

Charecter – класс описывающий объект персонажа.

Name – имя персонажа.

Color – цвет текста персонажа.

emoties [] – эмоции персонажа.

other[] – дополнительные параметры.

ValueForm – форма редактирования переменных.

ValueList – список значений.

ActForm() – переход на форму сценария.

Value класс описывающий объект переменных.

Name – имя переменной.

Value – первоначальное значение переменной.

Type – тип переменной (флаг, значение, счетчик).

ChoiceNode – Элемент условия.

ParamList – список параметров.

PathList – соответствующие параметрам лики до следующих нод.

DialogNode – Элемент диалога.

PhraseList – список фраз составляющих диалог.

Phrase – класс описывающий объект фразы.

Phrase – текст, фраза одного из персонажей.

Charecter – персонаж кому принадлежит фраза.

Emotion – эмоция персонажа в момент фразы.

getCharecter() – подбирает список персонажей.

getEmotion() - подбирает список эмоций соответствующих выбранному персонажу персонажей.

LogicNode

ConditionList – список условий

PathList – соответствующие условиям лики до следующих нод.

getPath() – метод выбора пути.

#### 5.4 Выбор среды для дальнейшей разработки

В качестве среды для разработки выбран игровой движок Godot Engine. Он подходит не только для создания 2D и 3D игр, но и оконных приложений. У Godot есть очень проработанная система создания элементов управления и написание обработчиков событий.

Основными причинами выбора такой среды разработки в том, что:

- приложение разрабатывается для дальнейшей обработки созданных сценарных проектов в игровых движках;
- удобного создания графических форм;
- из коробки заложена система связывания графических узлов в граф/дерево.

## Заключение

В ходе выполнения курсового проекта, был разработан проект модели программного приложения «Конструктор игровых сценариев». Была описана общая концепция использования приложения.

Определены основные пользовательские сценарии. Разработана черновая схема работы классов и их взаимодействия, которая будет улучшаться и дополняться в процессе разработки прототипа по данному проекту.

Данных матерьялов достаточно для разработки прототипа приложения и дальнейшего изучения слабых мест в проекте, а также их последующего улучшения.

## Список используемой источников

- 1 Мартин Фаулер, «UML. Основы. Краткое руководство по стандартному языку объектного моделирования», 2018г - Символ-Плюс, 192 стр.
- 2 Крэг Ларман, «Применение UML и шаблонов проектирования. Третье издание», 2018г - Вильямс, 736 стр.
- 3 Робрет Максимчук, «UML для простых смертных», 2016г - Лори, 400 стр.
- 4 Curtis НК, «Object-oriented Technology From Diagram to Code With Visual Paradigm for UML», 2016г - McGraw-Hill Education, 578 стр.
- 5 Джеймс Рамбо «Введение в UML от создателей языка», 2015г – ДМК Пресс, 496 стр.



# ПРИЛОЖЕНИЕ А – Отчет о проверке на заимствования



## Отчет о проверке на заимствования №1



Автор: [mihail2771@gmail.com](mailto:mihail2771@gmail.com) / ID: 6317881  
Проверяющий: [mihail2771@gmail.com](mailto:mihail2771@gmail.com) / ID: 6317881  
Отчет предоставлен сервисом «Антиплагиат»- <http://users.antiplagiat.ru>

### ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 30  
Начало загрузки: 29.05.2020 01:52:52  
Длительность загрузки: 00:00:01  
Имя исходного файла: КП ПиАПС Бабич М.М.  
17-ЗКБс-003.pdf  
Название документа: КП ПиАПС Бабич М.М.  
17-ЗКБс-003  
Размер текста: 1 кБ  
Символов в тексте: 12341  
Слов в тексте: 1482  
Число предложений: 129

### ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)  
Начало проверки: 29.05.2020 01:52:53  
Длительность проверки: 00:00:01  
Комментарии: не указано  
Модули поиска: Модуль поиска Интернет



ЗАИМСТВОВАНИЯ  
10,53%

САМОЦИТИРОВАНИЯ  
0%

ЦИТИРОВАНИЯ  
0%

ОРИГИНАЛЬНОСТЬ  
89,47%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.  
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.  
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общепотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.  
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.  
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.  
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.  
Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.  
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Доля в тексте	Источник	Ссылка	Актуален на	Модуль поиска	Блоков в отчете	Блоков в тексте
[01]	4,53%	4,74%	Нормативные ссылки	<a href="http://studopedia.org">http://studopedia.org</a>	20 Apr 2016	Модуль поиска Интернет	2	3
[02]	0%	4,74%	Перечень тем для проектирования программных систем	<a href="http://lektsi.org">http://lektsi.org</a>	24 Окт 2017	Модуль поиска Интернет	0	4
[03]	1,05%	4,08%	Выпускная квалификационная работа. Требования к стр...	<a href="https://vogu35.ru">https://vogu35.ru</a>	02 Мая 2020	Модуль поиска Интернет	2	6

Еще источников: 14  
Еще заимствований: 4,97%