

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

**СОГЛАСОВАНО**

Приглашенный преподаватель  
кафедры менеджмента инноваций

**УТВЕРЖДАЮ**

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

\_\_\_\_\_  
« 20 » \_\_\_\_\_ мая 2022 г. Е.А. Новиков

\_\_\_\_\_  
« 20 » \_\_\_\_\_ мая 2022 г. В.В. Шилов

**БЭКЕНД КОНСТРУКТОРА ВИЗУАЛЬНЫХ НОВЕЛЛ**

Программа и методика испытаний

**ЛИСТ УТВЕРЖДЕНИЯ**

RU.17701729.04.08-01 51 01-1-ЛУ

Име. № подл.	Подп.и дата	Взам. име. №	Име. № дубл.	Подп.и дата

Исполнитель

студент группы БПИ 202

\_\_\_\_\_  
« 20 » \_\_\_\_\_ мая 2022 г. /Д.Ю.Фёдоров/

Москва 2022

УТВЕРЖДЕН  
RU.17701729.04.08-01 51 01-1-ЛУ

## БЭКЕНД КОНСТРУКТОРА ВИЗУАЛЬНЫХ НОВЕЛЛ

Программа и методика испытаний

RU.17701729.04.08-01 51 01-1

Листов 17

Име. № подл.	Подп.и дата	Взам. име. №	Име. № дубл.	Подп.и дата

Москва 2022

## АННОТАЦИЯ

Программа и методика испытаний — это документ, в котором содержится информация о программном продукте, а также полное описание приемочных испытаний для данного программного продукта.

Настоящая Программа и методика испытаний для «Бэкенда конструктора визуальных новелл» содержит следующие разделы: «Объект испытаний», «Цель испытаний», «Требования к программе», «Требования к программной документации», «Средства и порядок испытаний», «Методы испытаний», «Приложения».

В разделе «Объект испытаний» указано наименование, краткая характеристика и назначение программы.

В разделе «Цель испытаний» указана цель проведения испытаний.

Раздел «Требования к программе» содержит основные требования к программе, которые подлежат проверке во время испытаний (требования к функционалу и интерфейсу).

Раздел «Требования к программным документам» содержит состав программной документации, которая представляется на испытания.

Раздел «Средства и порядок испытаний» содержит информацию о технических и программных средствах, которые следует использовать во время испытаний, а также порядок этих испытаний.

Раздел «Методы испытаний» содержит информацию об используемых методах испытаний.

## СОДЕРЖАНИЕ

<b>Глоссарий</b> . . . . .	5
<b>1. Объект испытаний</b> . . . . .	7
1.1. Наименование программы . . . . .	7
1.2. Краткая характеристика и область назначения . . . . .	7
<b>2. Цель испытаний</b> . . . . .	8
<b>3. Требования к программе</b> . . . . .	9
3.1. Требования к функциональным характеристиками . . . . .	9
3.2. Требования к входным и выходным данным . . . . .	11
3.2.1. Организация входных данных . . . . .	11
3.2.2. Организация выходных данных . . . . .	11
3.3. Требования к надёжности . . . . .	11
<b>4. Требования к программной документации</b> . . . . .	12
<b>5. Средства и порядок испытаний</b> . . . . .	13
5.1. Минимальный состав технических и программных средств . . . . .	13
5.2. Порядок проведения испытаний . . . . .	13
5.3. Требования к персоналу . . . . .	13
<b>6. Методы испытаний</b> . . . . .	14
6.1. Проверка требований к технической документации . . . . .	14
6.2. Проверка требований к функциональным характеристикам . . . . .	14
6.2.1. Проверка конструктора . . . . .	14
6.2.1.1. Проверка сохранения игры . . . . .	14
6.2.1.2. Проверка загрузки игры . . . . .	14
6.2.2. Проверка игры . . . . .	15
6.3. Проверка требований к надёжности . . . . .	15
Перечень использованных источников . . . . .	16

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 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению [7]

Изменения к данному документу 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 predetermined пользователем соединения.

**Граф сцены** – граф, состоящий из узлов, указанных ниже, связанных между собой через predetermined 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 (красный квадрат) на графическом представлении узла, при зажатии ЛКМ на которую можно создать переход, ведущий в данный узел.

**Область выхода узла** – выделенная область справа (зеленый квадрат) на графическом представлении узла, при зажатии ЛКМ на которую можно создать переход, ведущий 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.

**Фронтенд** – клиентская сторона пользовательского интерфейса программы.

**Бэкенд** – часть программы, отвечающая за функционирование её внутренней части. Сюда входят функции перемещения данных между фронтендом и бэкендом; функции для взаимодействия графов, узлов, переходов; функции для сохранения/загрузки данных.

# 1. ОБЪЕКТ ИСПЫТАНИЙ

## 1.1. Наименование программы

Наименование темы разработки: «Бэкенд конструктора визуальных новелл».

## 1.2. Краткая характеристика и область назначения

«Бэкенд конструктора визуальных новелл» — это часть программы «Конструктор визуальных новелл», отвечающая за функционирование её внутренней части. Сюда входят функции перемещения данных между фронтендом и бэкендом; функции для взаимодействия графов, узлов, переходов; функции для сохранения/загрузки данных.

Конструктор визуальных новелл — это десктопное приложение для создания игр в жанре «Визуальный роман», не требующее знания языков программирования от пользователя. Процесс создания игры основан на концепции «visual programming», которое подразумевает обмен данных через предопределенные соединения между логическими узлами.



## **2. ЦЕЛЬ ИСПЫТАНИЙ**

Целью испытаний является проверка корректности выполнения программой функций, изложенных в п. 4 «Требования к программе» документа «Техническое задание» из комплекта документации в соответствии с ЕСПД (Единой системой программной документации).

### 3. ТРЕБОВАНИЯ К ПРОГРАММЕ

Программа должна соответствовать следующим функциональным требованиям, указанным в документе «Бэкенд конструктора визуальных новелл. Техническое задание»:

#### 3.1. Требования к функциональным характеристиками

##### 1. Графы

###### 1.1. Граф игры

1.1.1. Наличие поля для Е-Узла для возможности реализации отношений между узлами в виде двустороннего списка.

1.1.2. Возможность добавить SS-Узел в граф.

1.1.3. Реализация функции для запуска игры.

###### 1.2. Граф сцены

1.2.1. Наличие поля для Е-Узла для возможности реализации отношений между узлами в виде двустороннего списка.

1.2.2. Возможность добавить CBG-Узел в граф.

1.2.3. Возможность добавить CD-Узел в граф.

1.2.4. Возможность добавить PS-Узел в граф.

1.2.5. Возможность добавить SC-Узел в граф.

1.2.6. Возможность добавить HC-Узел в граф.

1.2.7. Возможность добавить CN-Узел в граф.

1.2.8. Реализация функции для запуска сцены.

##### 2. Узлы

###### 2.1. Е-Узел

2.1.1. Наличие поля для перехода-выхода из узла.

###### 2.2. CBG-Узел

2.2.1. Наличие поля для хранения изображения фона в формате PNG и функций для работы с ним.

2.2.2. Наличие поля для перехода-входа в узел.

2.2.3. Наличие поля для перехода-выхода из узла.

2.2.4. Наличие функции для появления фона при активации узла.

###### 2.3. CD-Узел

2.3.1. Наличие поля для хранения текста персонажа и функций для работы с ним.

2.3.2. Наличие поля для перехода-входа в узел.

2.3.3. Наличие поля для перехода-выхода из узла.

2.3.4. Наличие функции для изменения текста персонажа при активации узлы. Если текст пустой, то персонаж ничего не говорит.

###### 2.4. CN-Узел

2.4.1. Наличие поля для хранения имени персонажа и функций для работы с ним.

2.4.2. Наличие поля для перехода-входа в узел.

2.4.3. Наличие поля для перехода-выхода из узла.

2.4.4. Наличие функции для изменения имени персонажа при активации узла.

## **2.5. PS-Узел**

2.5.1. Наличие поля для хранения аудио в формате WAV и функций для работы с ним.

2.5.2. Наличие поля для перехода-входа в узел.

2.5.3. Наличие поля для перехода-выхода из узла.

2.5.4. Наличие функции для воспроизведения аудио при активации узла.

## **2.6. SC-Узел**

2.6.1. Наличие поля для хранения имени персонажа и функций для работы с ним.

2.6.2. Наличие поля для хранения текста персонажа и функций для работы с ним.

2.6.3. Наличие поля для хранения изображения персонажа в формате PNG и функций для работы с ним.

2.6.4. Наличие поля для перехода-входа в узел.

2.6.5. Наличие поля для перехода-выхода из узла.

2.6.6. Наличие функции для появления персонажа при активации узла.

## **2.7. HC-Узел**

2.7.1. Наличие поля для хранения имени персонажа и функций для работы с ним.

2.7.2. Наличие поля для перехода-входа в узел.

2.7.3. Наличие поля для перехода-выхода из узла.

2.7.4. Наличие функции для сокрытия персонажа при активации узла.

## **2.8. SS-Узел**

2.8.1. Наличие поля для хранения графа сцены и функций для работы с ним.

2.8.2. Наличие поля для перехода-входа в узел.

2.8.3. Наличие поля для перехода-выхода из узла.

2.8.4. Наличие функции для запуска сцены при активации узла.

# **3. Переходы**

## **3.1. SMPL-Переход**

3.1.1. Наличие поля для узла, в который ведёт переход, и функций для работы с ним.

3.1.2. Наличие поля для узла, из которого ведёт переход, и функций для работы с ним.

3.1.3. Наличие функции для моментального перехода на следующий узел.

## **3.2. CLK-Переход**

3.2.1. Наличие поля для узла, в который ведёт переход, и функций для

работы с ним.

3.2.2. Наличие поля для узла, из которого ведёт переход, и функций для работы с ним.

3.2.3. Наличие функции для перехода на следующий узел при вводе игрока.

## **3.2. Требования к входным и выходным данным**

### **3.2.1. Организация входных данных**

Входные данные поступают через фронтенд с помощью графического интерфейса приложения – текстовые поля, выпадающие списки, кнопки, переключатели.

Для графических изображений требуются файлы формата PNG.

Для звуковых файлов требуются файлы форматы WAV.

Для десериализации файла игры требуется файл формата JSON. Причём файл должен содержать данные, пригодные для десериализации класса Novel.

### **3.2.2. Организация выходных данных**

Выходные данные отображаются на стороне фронтенда.

Сериализованный файл игры имеет формат JSON.

## **3.3. Требования к надёжности**

Для устойчивой работы программы необходимо соблюдать ряд организационно технических мер:

1. Обеспечить бесперебойное питание технических устройств.
2. Обеспечить регулярную проверку оборудования и программного обеспечения на наличие сбоев и неполадок.

В случае некорректных входных данных пользователя, программа не должна аварийно завершаться.

## **4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ**

На испытание должна быть представлена документация в следующем составе:

1. «Бэкенд конструктора визуальных новелл». Техническое задание (ГОСТ 19.201-78);
2. «Бэкенд конструктора визуальных новелл». Программа и методика испытаний (ГОСТ 19.301-78);
3. «Бэкенд конструктора визуальных новелл». Пояснительная записка (ГОСТ 19.404-79);
4. «Бэкенд конструктора визуальных новелл». Руководство оператора (ГОСТ 19.505-79);
5. «Бэкенд конструктора визуальных новелл». Текст программы (ГОСТ 19.401-78);

## **5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ**

### **5.1. Минимальный состав технических и программных средств**

Для работы десктопного приложения необходим компьютер, характеристики которого позволяют производить стабильную работу с операционной системой Windows 10 любой редакции и всеми ее компонентами:

1. Процессор с тактовой частотой не ниже 2 ГГц;
2. Оперативная память не меньше 4 Гб;
3. Свободное место на жестком диске 512 Мб или больше;
4. Наличие мышки или сенсорной панели, клавиатуры, монитора.

Вышеописанные условия обусловлены функциональностью программы, примерным размером архивов и исполняемых файлов, а также минимальными системными требованиями целевых операционных систем.

### **5.2. Порядок проведения испытаний**

Испытания должны проводиться в следующем порядке:

1. Скачать и разархивировать файл с проектом;
2. Провести испытания для конструктора, описанные ниже в разделе «Методы испытаний».
3. Выйти из конструктора
4. Провести испытания для игры, описанные ниже в разделе «Методы испытаний».
5. Выйти из игры

### **5.3. Требования к персоналу**

Для корректной работы программы достаточно одного человека. Специальные знания оператору не требуются.

## **6. МЕТОДЫ ИСПЫТАНИЙ**

### **6.1. Проверка требований к технической документации**

Состав программной документации проверяется наличием всех подписей и программной документации в системе SmartLMS. Также проверяется соответствие документации требованиям ГОСТ.

Все документы удовлетворяют представленным требованиям.

### **6.2. Проверка требований к функциональным характеристикам**

Проверка проводится в ручном режиме.

#### **6.2.1. Проверка конструктора**

1. Запустить конструктор.
2. Создать пустой проект
3. Создать игру, которая будет использовать все предоставленные в программе узлы и переходы.
4. Запустить режим игры из режима конструктора.
5. Пройти игру. Во время прохождения игры требуется проверить, чтобы все узлы и переходы работали так, как описано в техническом задании.

##### **6.2.1.1. Проверка сохранения игры**

1. Запустить конструктор.
2. Создать пустой проект.
3. Создать игру, которая будет использовать все предоставленные в программе узлы и переходы.
4. Сохранить игру.
5. Открыть сгенерированный JSON файл игры в любом текстовом редакторе и посмотреть, чтобы все требуемые сущности сериализовались в формате, который можно будет позже десериализовать.

##### **6.2.1.2. Проверка загрузки игры**

1. Запустить конструктор.
2. Создать пустой проект.
3. Создать игру, которая будет использовать все предоставленные в программе узлы и переходы.
4. Сохранить игру.
5. Закрыть конструктор.

6. Запустить конструктор.
7. Загрузить игру.
8. Проверить, чтобы загруженные данные не отличались от данных, созданных в прошлой сессии работы с конструктором.

### **6.2.2. Проверка игры**

1. Запустить конструктор.
2. Создать пустой проект.
3. Создать игру, которая будет использовать все предоставленные в программе узлы и переходы.
4. Скомпилировать игру.
5. Скопировать сгенерированный json файл, который находится рядом с исполняемым файлом конструктора, в буфер обмена.
6. Вставить сгенерированный json файл из буфера обмена рядом с исполняемым файлом игры.
7. Запустить игру.
8. Пройти игру. Во время прохождения игры требуется проверить, чтобы все узлы и переходы работали так, как описано в техническом задании.

### **6.3. Проверка требований к надежности**

В соответствии с техническим заданием программа не должна аварийно завершаться при исключительных ситуациях. При любых действиях пользователя программа не завершается аварийно.



## ПЕРЕЧЕНЬ ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19.101-77. ЕСПД. Виды программ и программных документов. — Москва : Стандартиформ, 2005.
2. ГОСТ 19.102-77. ЕСПД. Стадии разработки. — Москва : Стандартиформ, 2005.
3. ГОСТ 19.103-77. ЕСПД. Обозначение программ и программных документов. — Москва : Стандартиформ, 2005.
4. ГОСТ 19.104-78. ЕСПД. Основные надписи. — Москва : Стандартиформ, 2005.
5. ГОСТ 19.105-78. ЕСПД. Общие требования к программным документам. — Москва : Стандартиформ, 2005.
6. ГОСТ 19.106-78. ЕСПД. Требования к программным документам, выполненным печатным способом. — Москва : Стандартиформ, 2005.
7. ГОСТ 19.301-79. ЕСПД. Программа и методика испытаний. Требования к содержанию и оформлению. — Москва : Стандартиформ, 2005.
8. ГОСТ 19.603-78. ЕСПД. Общие правила внесения изменений. — Москва : Стандартиформ, 2005.
9. ГОСТ 19.604-78. ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом. — Москва : Стандартиформ, 2005.

[illegible]