

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет  
(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук  
Кафедра системного программирования**

РАБОТА ПРОВЕРЕНА

Рецензент  
Заместитель директора по  
учебной работе (информатизация)  
ЛГ МАОУ «СОШ № 3»

\_\_\_\_\_ Г.Н. Калабина

“ \_\_\_\_ ” \_\_\_\_\_ 2019 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой,  
д.ф.-м.н., профессор

\_\_\_\_\_ Л.Б. Соколинский

“ \_\_\_\_ ” \_\_\_\_\_ 2019 г.

**РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ЭЛЕКТРОННОГО  
ОБУЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ  
EDUCATIONAL DATA MINING**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ЮУрГУ – 02.03.02.2019.115-095.ВКР

Научный руководитель,  
профессор кафедры СП,  
д.г.н., к.ф.-м.н.

\_\_\_\_\_ С.М. Абдуллаев

Автор работы,  
студент группы ВМИ-401

\_\_\_\_\_ Е.С. Подорожко

Ученый секретарь  
(нормоконтролер)

\_\_\_\_\_ О.Н. Иванова

“ \_\_\_\_ ” \_\_\_\_\_ 2019 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
**«Южно-Уральский государственный университет»**  
**(национальный исследовательский университет)**  
Высшая школа электроники и компьютерных наук  
Кафедра системного программирования

УТВЕРЖДАЮ

Зав. кафедрой СП

\_\_\_\_\_ Л.Б. Соколинский

10.02.2019

**ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы бакалавра**  
студенту группы КЭ-401 Подорожко Егору Сергеевичу,  
обучающемуся по направлению 02.03.02  
«Фундаментальная информатика и информационные технологии»

**1. Тема работы** (утверждена приказом ректора от 25.04.2019 № 899)

Разработка приложения для электронного обучения с использованием методов Educational Data Mining.

**2. Срок сдачи студентом законченной работы:** 04.06.2019.

**3. Исходные данные к работе:**

3.1. Romero C., Ventura S., Pechenizkiy M., Baker R.S. Handbook of Educational Data Mining. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Taylor and Francis Group, LLC. 2011. 503 p.

3.2. Интернет-портал Unreal Engine 4. Documentation // <https://docs.unrealengine.com/latest/INT> (дата посещения 16.11.2018).

**4. Перечень подлежащих разработке вопросов**

4.1. Осуществить постановку задачи по созданию приложения.

4.2. Провести сравнительный анализ существующих аналогов.

4.3. Изучить и провести обзор средств реализации приложения.

4.4. Провести анализ требований и выполнить проектирование приложения.

4.5. Выполнить реализацию приложения на выбранной платформе.

4.6. Провести функциональное тестирование полученного приложения.

5. **Дата выдачи задания:** 09.02.2019.

**Научный руководитель**

Профессор кафедры СП,

д.г.н., к.ф.-м.н.

С.М. Абдуллаев

**Задание принял к исполнению**

Е.С. Подорожко

## **ОГЛАВЛЕНИЕ**

ВВЕДЕНИЕ.....	5
1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ .....	8
1.1. Анализ предметной области .....	8
1.2. Анализ аналогичных проектов .....	12
1.3. Обзор средств реализации приложения.....	12
2. ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ .....	16
2.1. Техническое предложение .....	16
2.1.1. Общая информация.....	16
2.1.2. Основная концепция .....	16
2.1.3. Определение требований к проектируемому приложению.....	16
2.2. Разработка диаграммы вариантов использования.....	18
2.3. Диаграмма классов.....	19
2.4. Проектирование интерфейса приложения.....	20
3. РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ .....	23
3.1. Компоненты и структура проекта .....	23
3.2. Реализация базовой логики приложения средствами BluePrint.....	23
3.3. Реализация пользовательского интерфейса. Работа с widget-файлами	28
4. ТЕСТИРОВАНИЕ .....	30
4.1. Функциональное тестирование.....	30
4.2. Юзабилити-тестирование.....	32
ЗАКЛЮЧЕНИЕ .....	35
СПИСОК ЛИТЕРАТУРЫ .....	37

## **ВВЕДЕНИЕ**

### *Актуальность темы*

Увлечение компьютерными играми является значимым экономическим, социальным, культурным компонентом современной жизни. Популярность этого увлечения объясняется их практически неограниченной интерактивностью – игрок, в отличие от читателя или кинозрителя, перестает быть пассивным наблюдателем и активно управляет происходящими на экране компьютера событиями.

Спектр и интерактивность эмоций компьютерного игрока сравнимы с тем, что испытывают участники азартной игры, реальным участием в игровых видах спорта, либо интеллектуальными соревнованиями (типа «КВН» или «Что, Где, Когда?») и другими reality-show. Сообщества геймеров – это наглядный пример «чистой» киберсоциализации с полным набором всех элементов психофизического, культурно-лингвистического и экономического своеобразия, выделяющего геймеров в особую социально-культурную группу [2–6, 8–9]. В отличие от других социальных групп, например, футбольных фанатов, или последователей различных культов – сообщества геймеров менее консервативны: не предполагают раскрытия личных данных, гендерное равенство и др. Во многом, игровые привычки прививаются уже в раннем детстве. Они существенно меняют когнитивный профиль человека [2, 5, 9]. Поэтому игры становятся важным элементом среднего и профессионального образования – только в библиотеке [www.elibrary.ru](http://www.elibrary.ru) уже насчитывается более тысячи подобных исследований.

В то же время, быстрый рост популярности игровой индустрии подталкивает развитие технических средств и программного обеспечения. Ведь с каждым годом компьютерные игры занимают у молодёжи всё больше и больше времени. Именно поэтому внедрение компьютерных игр в образовательный процесс является неизбежным. Ни для кого не является секретом, что современной молодежи намного проще воспринимать всю информацию именно в игровой форме. Одной из актуальных проблем на данный момент

остаются адекватная и объективная подготовка в игровой форме учащихся к различным видам тестирований путём развития их когнитивных функций [21, 22].

В данной работе реализуется одно из возможных решений этой проблемы.

### ***Цель и задачи***

Целью работы является разработка образовательного приложения на платформе Unreal Engine 4 [1] с использованием методов Educational Data Mining [9, 24].

Для достижения поставленной цели необходимо выполнить следующие задачи:

- осуществить постановку задачи по созданию приложения;
- определить требования к системе, проведя сравнительный анализ существующих аналогов;
- изучить и провести обзор программных средств реализации игрового приложения;
- провести анализ требований и выполнить проектирование системы приложения средствами UML;
- реализовать игровое приложение на основе проанализированных требований;
- провести функциональное тестирование полученного приложения.

### ***Структура и объем работы***

Работа состоит из введения, четырех глав, заключения и библиографического списка. Объем работы составляет 39 страниц, объем библиографии – 25 источников.

В главе «Теоретическая часть» описана постановка задачи, произведен обзор существующих аналогов приложения и современных способов реализации. Также описано отличие методов educational data mining от data mining и их применение в образовательных приложениях.

Глава «Проектирование приложения» посвящена определению требований к разрабатываемому мобильному приложению. В этой же главе описываются: диаграмма вариантов использования, диаграмма классов, проектирование интерфейса приложения.

Третья глава посвящена «Реализации приложения», в ней описаны компоненты и структура реализованного приложения, реализация базовой логики приложения при помощи Blueprint, реализация интерфейса при помощи widget-файлов.

В главе «Тестирование» описаны результаты тестирования приложения. Представлены результаты функционального и юзабилити-тестирования.

В заключении сделаны выводы о проделанной работе и сформулированы перспективы дальнейшей разработки.

# 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1. Анализ предметной области

Перед дальнейшим проектированием приложения необходимо провести анализ предметной области, тем самым выделить основные моменты во избежание недочётов при составлении требований к проектируемому ПО.

Для начала необходимо дать определение термину data mining и определить его отличие от educational data mining.

Data mining – собирательное название, используемое для обозначения совокупности методов обнаружения в данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности [25].

Английское словосочетание «data mining» не имеет устоявшегося перевода на русский язык. Обычно при работе с данным словосочетанием в русском языке его характеризуют как добыча данных, извлечение данных, просев информации. Наиболее принятое обозначение – интеллектуальный анализ данных.

Основу методов data mining составляют различные методы моделирования, прогнозирования и классификации, основанные на применении деревьев решений, искусственных нейронных сетей, генетических алгоритмов, эволюционного программирования, ассоциативной памяти, нечёткой логики.

Также нередко к методам data mining относят статистические методы, в качестве примера рассмотрим некоторые из них.

Дискриминантный анализ, который используется для принятия решений о том, какие переменные разделяют возникающие наборы данных.

Анализ связей, который используется для оценки отношения между узлами, выступающими зачатую объектами системы. Отношения могут быть для различных типов узлов: людей, операций, организаций и т.д.



Анализ выживаемости, позволяющий оценить вероятность наступления события.

Факторный анализ, применяемый для изучения взаимосвязей между значениями переменных.

Использовать методы data mining обычно приходится при наличии одной из двух задач: наличие достаточно крупной базы данных, наличие «скрытых знаний».

Под «скрытыми знаниями» подразумевается один из типов или несколько типов знаний:

- нетривиальные, то есть знания, которые нельзя просто так увидеть при визуальном анализе данных;
- доступные для интерпретации – знания, которые легко представить в наглядной для пользователя форме;
- ранее неизвестные – новые знания, не подтверждающие ранее полученные сведения;
- практически полезные, то есть знания, представляющие определённую ценность для исследователя или потребителя.

Методы data mining применяются как для работы с большими данными, так и для малых объёмов данных, полученных по результатам анализа или отдельных экспериментов.

Методы educational data mining в основном преследуют все те же цели, что и методы data mining, но используются для принятия решений в сфере образования [9, 23–25].

Также помимо стандартных методов data mining, educational data mining может использовать специфичные методы, например, из области психометрии [23].

Ещё одной отличительной чертой educational data mining является вид данных, с которыми EDM работает. Это могут быть как лог-файлы с переходами и кликами, так и оценка знаний пользователя в ходе его работы с приложением.

Использование методов educational data mining в образовательных приложениях может быть полезным как для студентов и преподавателей, так и для разработчиков [23, 24].

Для студентов, это в первую очередь, адаптация образовательного процесса под их текущий уровень знаний.

Для преподавателей – разбиение обучающихся на обоснованные группы (пол, активность, возраст, успеваемость).

Для разработчиков – оценка эффективности обучения пользователя.

## **1.2. Анализ аналогичных проектов**

На данный момент рынок мобильных игр на голову обгоняет рынок компьютерных и консольных игровых приложений как по доходам, так и по частоте использования [10]. В наше время смартфоны есть почти что у каждого человека, и с каждым годом они становятся более доступнее и качественнее. Поэтому при рассмотрении аналогов было принято решение ориентироваться именно на данный сегмент индустрии.

В ходе выполнения дипломной работы были рассмотрены два самых популярных магазина приложений: Google Play и Apple Store. Из обилия проектов были выбраны самые популярные из жанров «головоломки» и «викторины».

### **Правда или Ложь [11].**

Игра в жанре «головоломки». В игре присутствуют вопросы на различные темы, которые разделены на уровни. При прохождении уровня, пользователю открывается следующий. В остальном интерфейс игры схож с её аналогами. Пользователю предоставляется вопрос, на который он должен дать ответ: «правда» это или «ложь». При ответе на вопрос, на экране пользователя появляется правильный ответ, при совпадении правильного ответа с ответом пользователя игрок увеличивает счётчик правильных ответов, при несовпадении теряет счётчик допустимого количества неправильных ответов. Проведение времени в игре положительно влияет на развитие памяти, логики и восприятия информации.

На данный момент приложение имеет среднюю оценку 4,8 и более 3 миллионов установок.

### **Верю или не верю [12].**

Игра в жанре «викторины». Развивающая игра, которая на основе различных фактов позволяет оценить и развить уровень эрудированности, а также существенно расширить кругозор игрока. Положительно влияет на развитие когнитивных функций. В игре пользователю каждый раз представляется новый факт, на который пользователь должен ответить: «верит» он в него или «не верит». Игра не имеет уровней, факты появляются один за другим, при ответе пользователя на экране появляется правильный ответ, иногда с пояснениями. При совпадении правильного ответа и ответа пользователя насчитывается один балл, при несовпадении игра продолжается без вычета баллов.

На данный момент в магазинах приложение по всем отзывам имеет среднюю оценку 4,7 и более 2 миллионов установок.

### **Да или Нет [13].**

Игра в жанре «викторины». Интеллектуальная игра, в которой приведены различные факты, игроку же надо определить какие из них являются правильными, а какие нет. В отличие от предыдущих аналогов игра сильно привязана к монетизации, изначально пользователю выдаётся 200 монет и снимает по 5 за каждый неправильный ответ, монеты можно получить только за реальные деньги, либо просмотр партнёрской рекламы.

На данный момент в магазинах приложение по всем отзывам имеет среднюю оценку 3,9 и более 1 миллиона установок.

По результатам проведённого анализа можно понять, что игры данного жанра популярны, имеют огромное количество установок и большое количество как положительных оценок, так и отзывов.

Проанализировав отзывы и оценки пользователей, стали понятны элементы приложения, которые нравятся людям, а какие, наоборот, неприятны и отталкивают пользователя от продукта.

Информация, полученная в результате анализа предметной области, позволяет адекватно спроектировать и реализовать приложение, исключить заведомо проигрышные стратегии реализации ещё на этапе проектирования и следовать в дальнейшем наиболее выигрышным вариантам реализации.

### **1.3. Обзор средств реализации приложения**

В настоящее время существует множество различных средств реализации приложений со своими плюсами и минусами. Для реализации приложения необходимо выбрать наиболее сбалансированную и оптимальную платформу с удобным интерфейсом, достаточным количеством документации и бесплатной лицензией.

В ходе выполнения работы был проведён анализ следующих средств реализации:

#### **LibGDX**

LibGDX [14] – кроссплатформенный фреймворк для создания игр и приложений. Работает на Windows, Linux, Mac OS X, Android, также на iOS и HTML 5, но с некоторыми ограничениями. Одинаково хорошо предназначен как для создания 2D игр, так и для 3D реализаций. Имеет поддержку огромного количества сторонних инструментов: поддерживает физический движок `box2d`, работает с редактором карт `Tiled`, прекрасно справляется с отрисовкой `ttf` шрифтов и т.д. Используя отдельные модули для сборки приложения под различные виды платформ, но при этом, имея независимый модуль с основным программным кодом, позволяет разработчику писать, тестировать и отлаживать код на личном компьютере и затем переносить его на другие операционные системы. Для написания игр и приложений используется язык Java, но также может подойти любой другой язык, работающий на Java-машине, например, `python`, `groovy`, `scala`.

Основные достоинства:

- 1) встроенная библиотека быстрой математики;
- 2) встроенный набор виджетов для рисования графического интерфейса;

- 3) высокая производительность;
- 4) кроссплатформенность;
- 5) открытый исходный код;
- 6) встроенная библиотека UI;
- 7) поддержка расширений;
- 8) легкость реализации логики поведения частиц.

Основные недостатки:

- 1) высокий порог вхождения;
- 2) относительно малый объём документации.

## **Unity**

Unity [15] – кроссплатформенная среда разработки игровых приложений от компании Unity Technologies. На Unity написано огромное количество игровых приложений, работающих под более чем 20 различными ОС. При создании приложений используется компонентно-ориентированный подход. Хорошо подходит для создания средних по сложности проектов. Основным языком разработки – C#. Выпуск Unity состоялся в 2005 году и активно развивается до сих пор.

Основные достоинства:

- 1) большое количество готовых ассетов, включающих в себя скрипты;
- 2) огромный список поддерживаемых платформ;
- 3) большое количество пользователей;
- 4) низкий порог вхождения;
- 5) наличие визуальной среды разработки;
- 6) модульная система компонентов;
- 7) бесплатная лицензия.

Основные недостатки:

- 1) сложность в работе с многокомпонентными схемами;
- 2) закрытый исходный код;
- 3) затрудненное подключение внешних библиотек;

4) медленная работа и большой вес приложений.

### **Unreal Engine 4**

Unreal Engine 4 [1] – кроссплатформенная среда разработки от компании Epic Games. Изначально предназначался для разработки игр в жанре «шутер от первого лица», но в дальнейшем был адаптирован под всевозможные виды жанров. Бесплатный для всех разработчиков, начиная с 2015 года.

В состав технологий Unreal Engine входят: движок обеспечения физики и графики, искусственный интеллект, управление сетевой и файловой системой, а также многофункциональный редактор UnrealEd.

Основной язык разработки – C++, также имеет встроенный редактор визуального программирования удобный для реализации базовой логики – Blueprint [1].

Основные достоинства:

- 1) современность, универсальность, кроссплатформенность;
- 2) высокая производительность;
- 3) бесплатная лицензия;
- 4) визуализация элементов в реальном времени;
- 5) огромное количество пользователей;
- 6) хорошая поддержка от разработчиков;
- 7) открытый исходный код.

Основные недостатки:

- 1) мало документации на русском языке.

На основе данных, полученных в результате обзора средств реализации, была выбрана платформа для реализации игрового приложения – Unreal Engine 4. По итогу анализа он оказался наиболее оптимальным для реализации приложения, имел открытый исходный код, удобную систему создания интерфейса, активную поддержку как от разработчиков, так и от пользователей, также немалую роль сыграло наличие положительного опыта работы с этой платформой при написании курсовых и научных работ в течение всего периода обучения в ЮУрГУ.

## **Вывод**

Анализ предметной области позволил определиться с использованием методов EDM из области психометрии: разбиение пользователей на группы по восприятию информации, для поиска «скрытых знаний».

Таким образом, детальный анализ аналогичных приложений помог выявить наиболее благоприятные элементы для реализации нашего приложения. Это, в свою очередь, позволяет вычеркнуть заведомо проигрышные стратегии реализации ещё на этапе проектирования системы, экономя тем самым общее время реализации проекта.

На основе полученных данных был произведён обзор существующих средств реализации приложений, в результате которого была выбрана наиболее оптимальная платформа: **Unreal Engine 4**.

## **2. ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ**

### **2.1. Техническое предложение**

#### **2.1.1. Общая информация**

«Плюс или Минус» – игра в сочетании жанров «головоломка» и «викторина», основной задачей которой является подготовка пользователя к различным видам тестирований и развития его когнитивных функций [21, 22]: внимания, восприятия, памяти, интеллекта и т.д.

#### **2.1.2. Основная концепция**

Игра является приложением, в котором основная задача игрока заключается в прохождении уровней путём ответа на вопросы.

В каждом уровне вопросы имеют различную сложность, также уровни различаются количеством вопросов и количеством попыток на неправильные ответы.

Когда у пользователя кончаются попытки на неправильные ответы, уровень считается оконченным и пользователю даётся шанс пройти его ещё раз.

Также на этапе проектирования было решено, где именно будут применяться методы EDM. После прохождения первого уровня составляется «портрет» пользователя, на основе которого в дальнейшем подбирается сложность вопросов. Использование методов EDM из области психометрии позволяют разбивать пользователей на группы по восприятию информации, что в свою очередь позволяет полностью адаптировать образовательный процесс под пользователя, подстраиваясь под его текущий уровень [9, 21, 23, 24].

В качестве эксперимента в данной работе было выбрано стандартное разбиение на 4 типа: «двоечник», «троечник», «хорошист», «отличник» [21].

#### **2.1.3. Определение требований к проектируемому приложению**

Основная задача на данном этапе заключается в выполнении работы по планированию навигации и функционала, позволяя структурировать идеи в конкретные требования, предотвратить ошибки и избежать



выполнения лишних работ на стадии ранней разработки. В ходе выполнения работы по проектированию были определены функциональные и нефункциональные требования к системе [16].

### **Функциональные требования к проектируемой системе**

Функциональные требования в первую очередь определяют функциональность программного обеспечения, описывают пункты, которые необходимо реализовать в системе, какие возможности она должна предоставлять.

Таким образом, можно определить следующий набор функциональных требований к системе.

1. Система должна предоставлять пользователю выбор уровней и сохранение результатов.
2. Система должна предоставлять пользователю выбор: сохранения, либо сброса прогресса прохождения.
3. Система должна иметь удобный интерфейс выбора ответов.
4. Система должна обрабатывать данные ответов пользователя.
5. Система должна иметь список вопросов и ответов.
6. Система должна предоставлять интерфейс в случае проигрыша или выигрыша.
7. Система должна составлять психологический портрет игрока, по результатам прохождения первого уровня.
8. Система должна выводить на экран пользователя текущее количество оставшихся вопросов и попыток на неправильный ответ.

### **Нефункциональные требования к проектируемой системе**

Описание нефункциональных требований в свою очередь позволяет на этапе проектирования выявить и описать свойства и ограничения, накладываемые на систему. По итогу для реализации приложения были сформулированы следующие нефункциональные требования.

1. Система должна быть написана на платформе Unreal Engine 4, с использованием языка программирования C++.
2. Система должна использовать технологию Blueprint.

## 2.2. Разработка диаграммы вариантов использования

Для более точного моделирования функциональных требований к программному продукту при его проектировании, средствами UML [17] была построена диаграмма вариантов использования системы. В ходе проектирования взаимодействия с игровым приложением был выделен следующий актер.

**Пользователь** – пользователь приложения, которому доступна возможность использования всех его функций.

Диаграмма, созданная на основе функциональных требований к приложению, представлена на рис. 1.

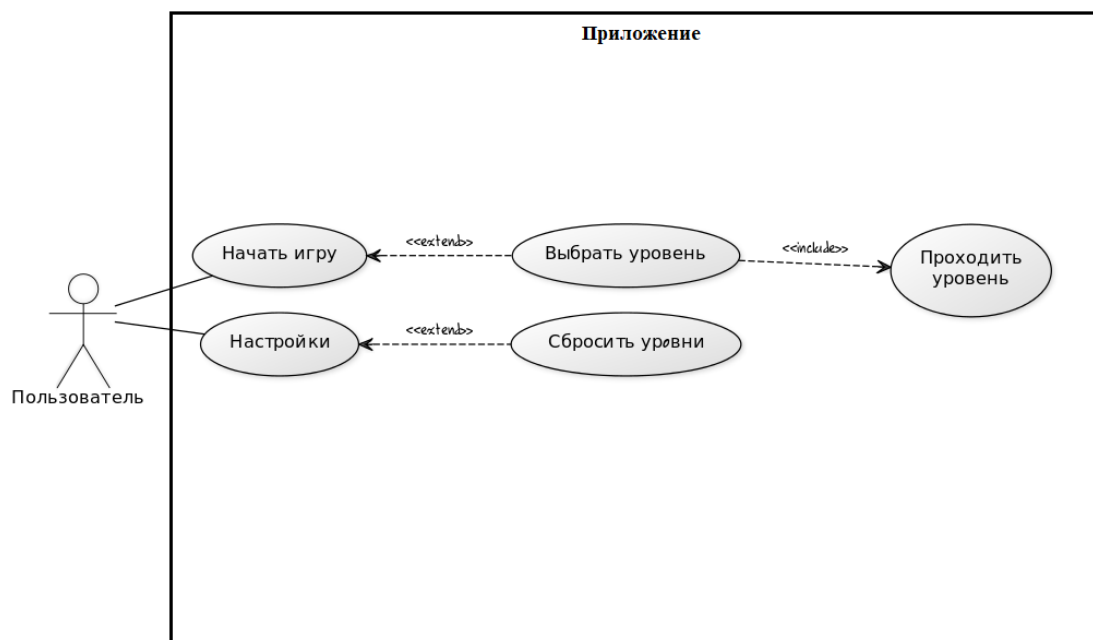


Рис. 1. Диаграмма вариантов использования

На представленной выше диаграмме изображены базовые варианты использования приложения пользователем.

*Начать игру* – перейти к выбору уровня, для дальнейшего его прохождения.

*Настройки* – перейти к возможности сброса уровней к начальному этапу.

*Выбрать уровень* – выбрать номер уровня для дальнейшего его прохождения.

*Сбросить уровни* – обнулить сохранение по текущим пройденным уровням и начать игру с нуля.

*Проходить уровень* – проходить текущий доступный уровень.

### 2.3. Диаграмма классов

Для более точного проектирования внутренней структуры проекта, используя средства UML, была построена диаграмма классов (рис. 2):

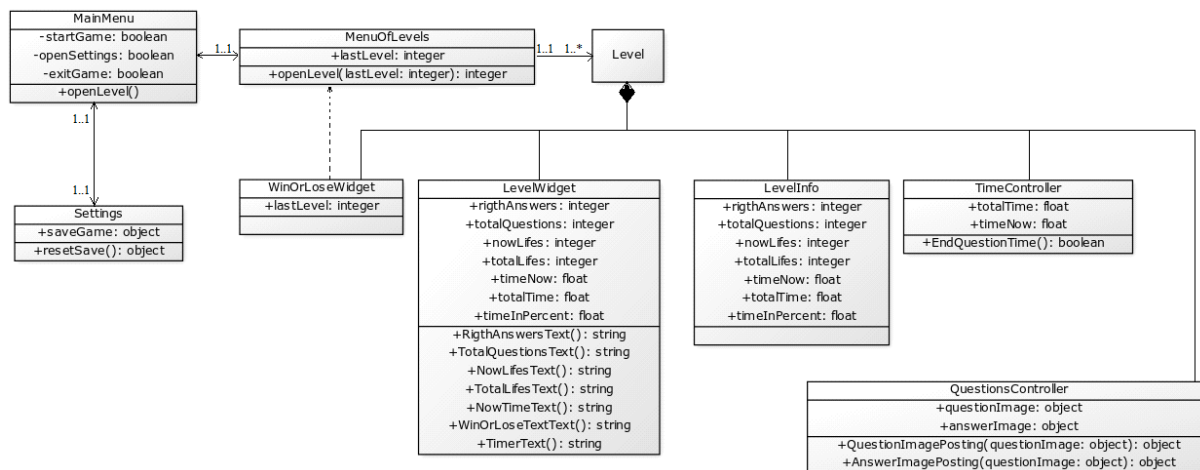


Рис. 2. Диаграмма классов

Диаграмма классов – диаграмма, которая отображает классы системы, их названия, атрибуты, методы и взаимосвязи между ними. Диаграмма классов служит для представления статической структуры модели системы в терминологии классов ООП.

Классы *MainMenu* и *Settings* связаны двунаправленной ассоциацией, так как существует переход от первого ко второму и наоборот. Такая же связь наблюдается между *MainMenu* и *MenuOfLevels*.

Класс *MenuOfLevels* связан ассоциацией с классом *Level*, так как этот класс использует его для запроса данных об уровне, который на данный момент необходимо вывести пользователю на экран.

Класс *WinOrLoseWidget* связан с классом *MenuOfLevels* при помощи зависимости, так как после успешного или неуспешного прохождения уровня пользователем зависит дальнейший доступный на данный момент для него уровень.

Классы *WinOrLoseWidget*, *LevelWidget*, *LevelInfo*, *TimeController*, *QuestionsController* связаны с классом *Level* композицией. Уничтожение объектов *Level* приводит к уничтожению объектов *WinOrLoseWidget*, *LevelWidget*, *LevelInfo*, *TimeController*, *QuestionsController*.

Класс *ChannelItem* связан с классами *ChannelActivity*, *ChannelsAdapter* и *SearchActivity* композицией.

## 2.4. Проектирование интерфейса приложения

На этапе проектирования интерфейса необходимо не просто его скомпоновать и организовать, но и в первую очередь сделать его понятным для пользователя.

Самое главное, чтобы интерфейс помогал пользователям добиваться своих целей: его назначение должно быть очевидным для пользователя, человек должен понимать с чем он взаимодействует через интерфейс.

Проектирование интерфейса осуществлялось на основе анализа, проведённого в пункте 1.1 «Анализ аналогичных проектов». В интерфейс [18] были включены элементы с положительной оценкой от пользователей.

Ниже представлены интерфейс: Главное меню (рис. 3), Меню выбора уровней (рис. 4), Игровое поле (рис. 5).



Рис. 3. Интерфейс «Главное меню»

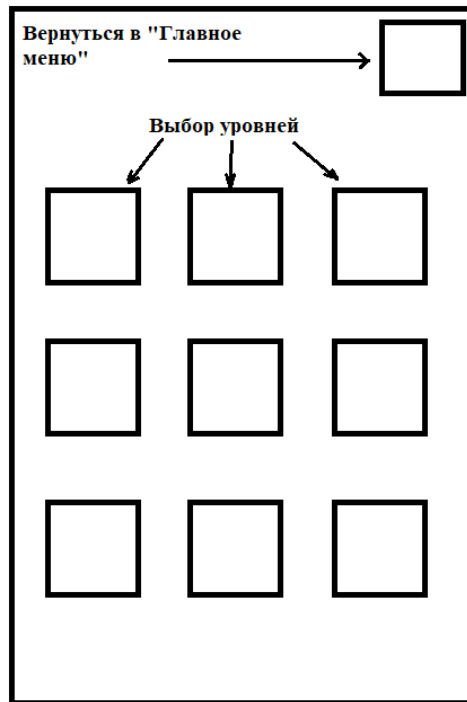


Рис. 4. Интерфейс «Выбор уровней»



Рис. 5. Интерфейс «Игровое Поле»

Ясность и понятность интерфейса зарождает в пользователях уверенность в себе и способствует тому, что он продолжит работать с нашим приложением. Не продуманный интерфейс наоборот отпугивает клиента.

## **Вывод**

На основании теоретической части в ходе проектирования были определены функциональные и нефункциональные требования к разрабатываемому приложению.

Для наглядного отображения функциональных требований была разработана диаграмма вариантов использования, после чего для детального представления внутренней части проекта была построена диаграмма классов.

Окончательным этапом проектирования стала разработка интерфейса приложения.

### 3. РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ

#### 3.1. Компоненты и структура проекта

Для создания приложения использовался пустой шаблон.

Приложение было решено разделить на четыре экрана: главное меню, меню уровней, сам уровень и окно проигрыша или победы. Для каждого экрана был создан свой класс. На рис. 6 отображена структура проекта.

В папке *Content* содержатся все основные файлы приложения.

Папка *BP* содержит все необходимые классы в виде BluePrint файлов, а именно классы уровней, главного меню, меню уровней и окон победы и проигрыша.

Папка *Maps* содержит полностью соединенные widget файлы и классы, а также свой игровой GameMode для каждого из окон.

Папка *Pictures* содержит элементы дизайна приложения, разработанные вне Unreal Engine.

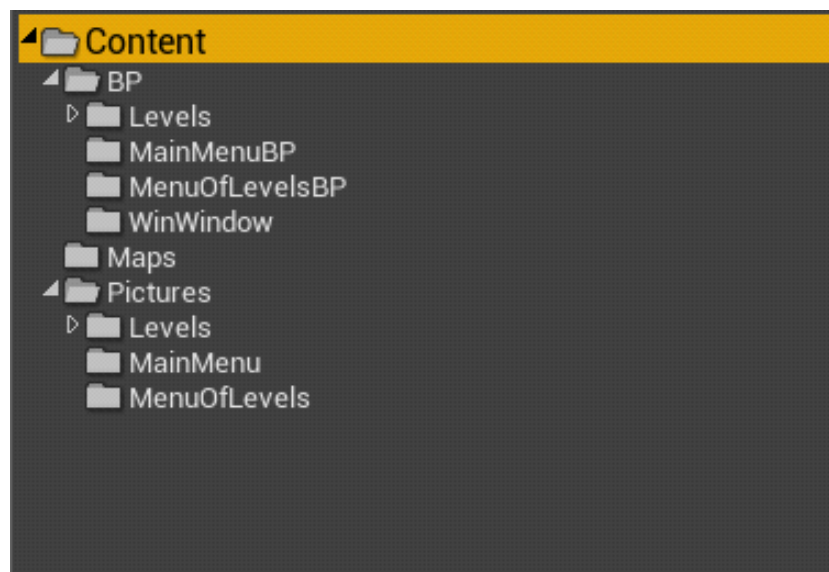


Рис. 6. Компоненты и структура проекта

#### 3.2. Реализация базовой логики приложения средствами **BluePrint**

Для реализации базовой логики приложения были использованы BluePrints, которые являются достаточно эффективными, если система не требует изменения исходного кода платформы. BluePrints хоть и является относительно новой системой, но активно развивается разработчиком.

Первым было реализовано *Главное меню* (рис. 7) и соответствующие для него классы, описанные в п. 2.2 «Разработка диаграммы вариантов использования» главы «Проектирование приложения».

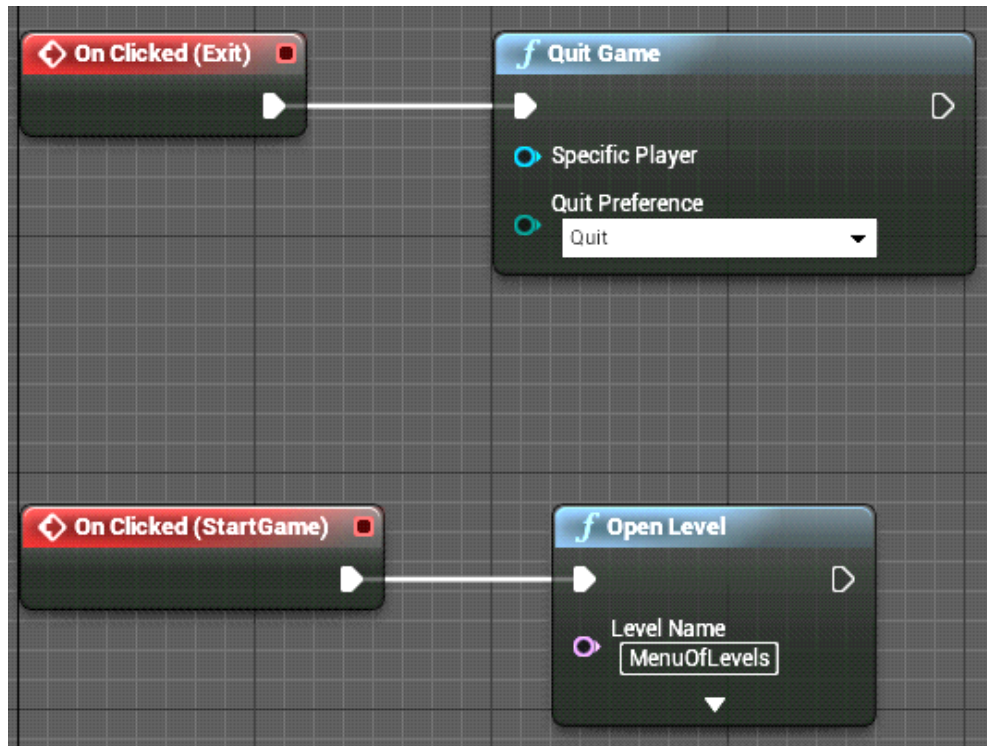


Рис. 7. Реализация класса *MainMenu*

Далее был реализован класс *Выбора уровней* (рис. 8), позволяющий непосредственно пользователю выбирать необходимый уровень, и сохраняющий данные о пройденных уровнях.

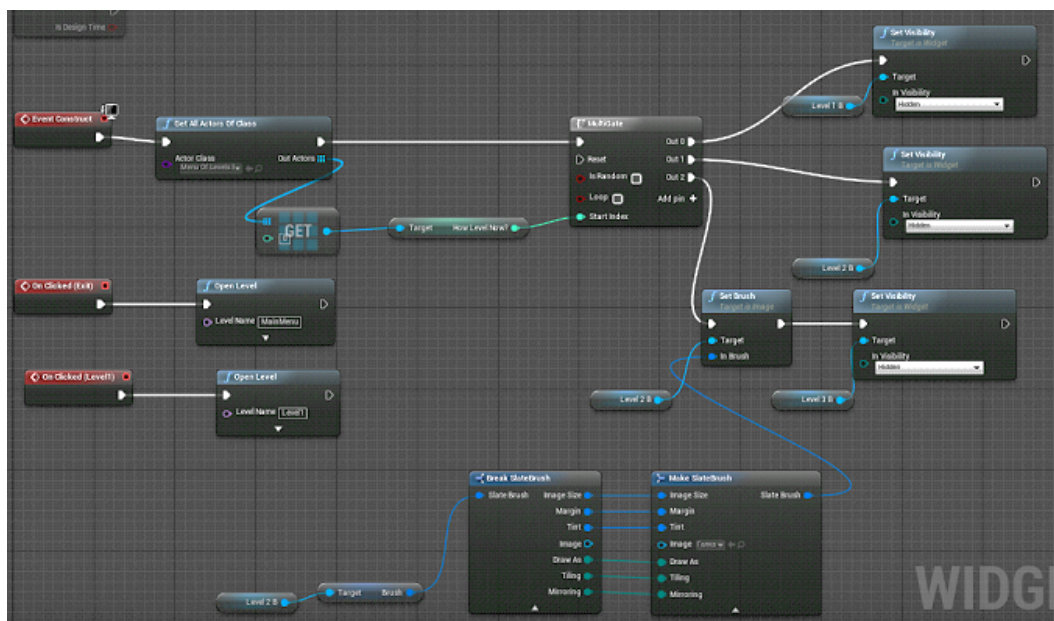


Рис. 8. Реализация класса *MenuOfLevels*



Затем был реализован самый обширный класс самого *Уровня* (рис. 9, рис. 10), в котором происходит обработка ответов пользователя и взаимосвязь между тремя widget файлами: окно уровня, окно правильного ответа и окно проигрыша\победы.

Алгоритм работы уровней игры представлен в виде блок-схемы на рис. 11.

Изначально программа проверяет равенство текущего количества правильных ответов пользователя с количеством необходимых для прохождения. В случае, если они равны, то пользователю выводится widget-файл «окно победы», а также открывается следующий уровень в меню уровней.

Если же количество ответов не совпадает, программа сравнивает количество оставшихся попыток на ошибки, и если оно не равно нулю, то цикл продолжается.

Далее на экран пользователя выводится текущий вопрос и запускается таймер. Таймер каждый раз уменьшается на стандартную величину равную одной секунде. До тех пор, пока не станет равен нулю, либо пользователь не ответит на вопрос.

В случае достижения таймером значения равного нулю, у пользователя отнимается одна попытка на ошибку, выводится правильный ответ, и программа ожидает нажатия кнопки «продолжить», для начала цикла с начала.

Если же пользователь даёт ответ на вопрос, то программа проверяет: являлся ли вопрос изначально ложным, то есть таким, у которого ответ «нет» является правдивым. Исходя из этих данных происходит анализ ответа пользователя.

В случае правильного ответа, количество правильных ответов пользователя увеличивается на один, в ином же варианте отнимается одна попытка на неправильный ответ. Независимо от правильности ответа пользователю на экран выводится правильный ответ, и программа ожидает нажатия кнопки «продолжить» для начала цикла с нуля.

Если количество попыток на ошибки становится равным нулю, то программа выводит widget-файл «окно проигрыша», и при нажатии кнопки «продолжить», перенаправляет пользователя в меню уровней, где он может заново выбрать и начать прохождение уровня.

Также важным фактором является разбиение пользователей на профили: «двоечник», «троечник», «хорошист», «отличник».

В дальнейшем это влияет на набор вопросов, выдаваемых пользователю в качестве заданий. Например, пользователь с профилем «отличник», будет получать более сложные вопросы, чем пользователь с профилем «двоечник». Профиль присваивается пользователю после успешного прохождения первого уровня, на основе данных, полученных из модуля анализа производительности игрока, более подробно о котором написано в пункте 4.2. «Юзабилити-тестирование».

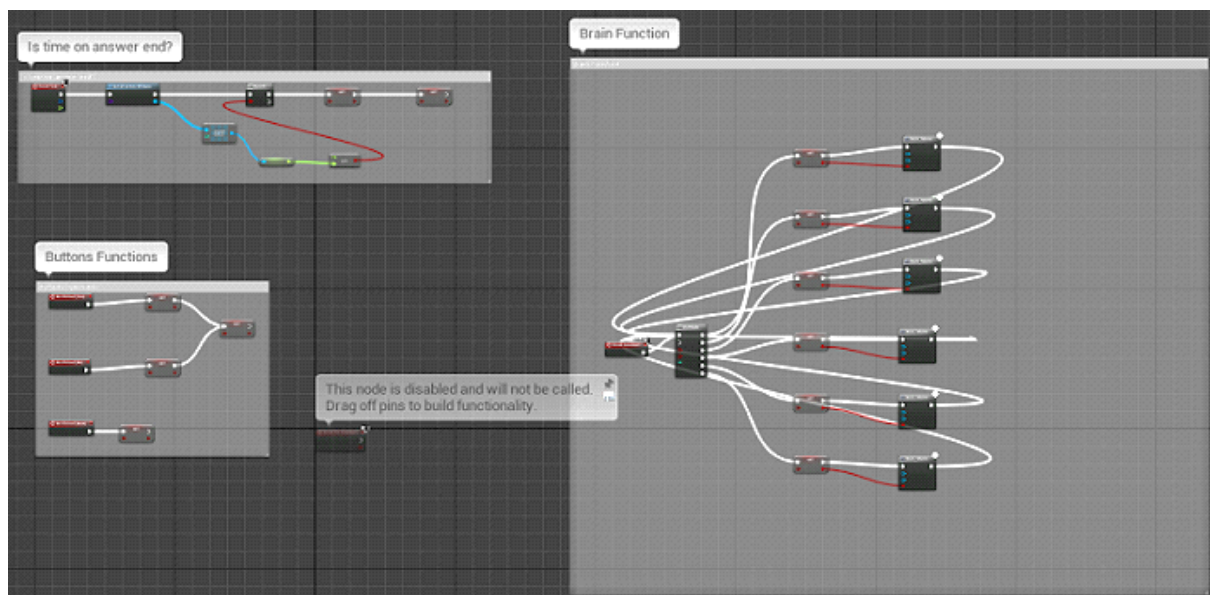


Рис. 9. Реализация класса *Level*. Общие сведения

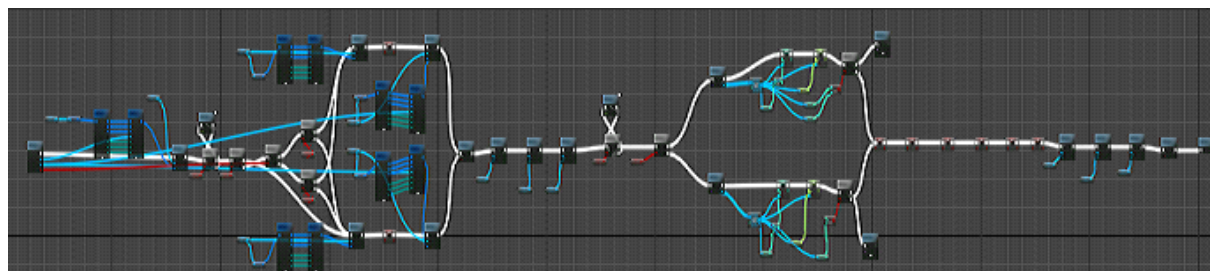


Рис. 10. Реализация класса *Level*. Обработка ответов пользователя

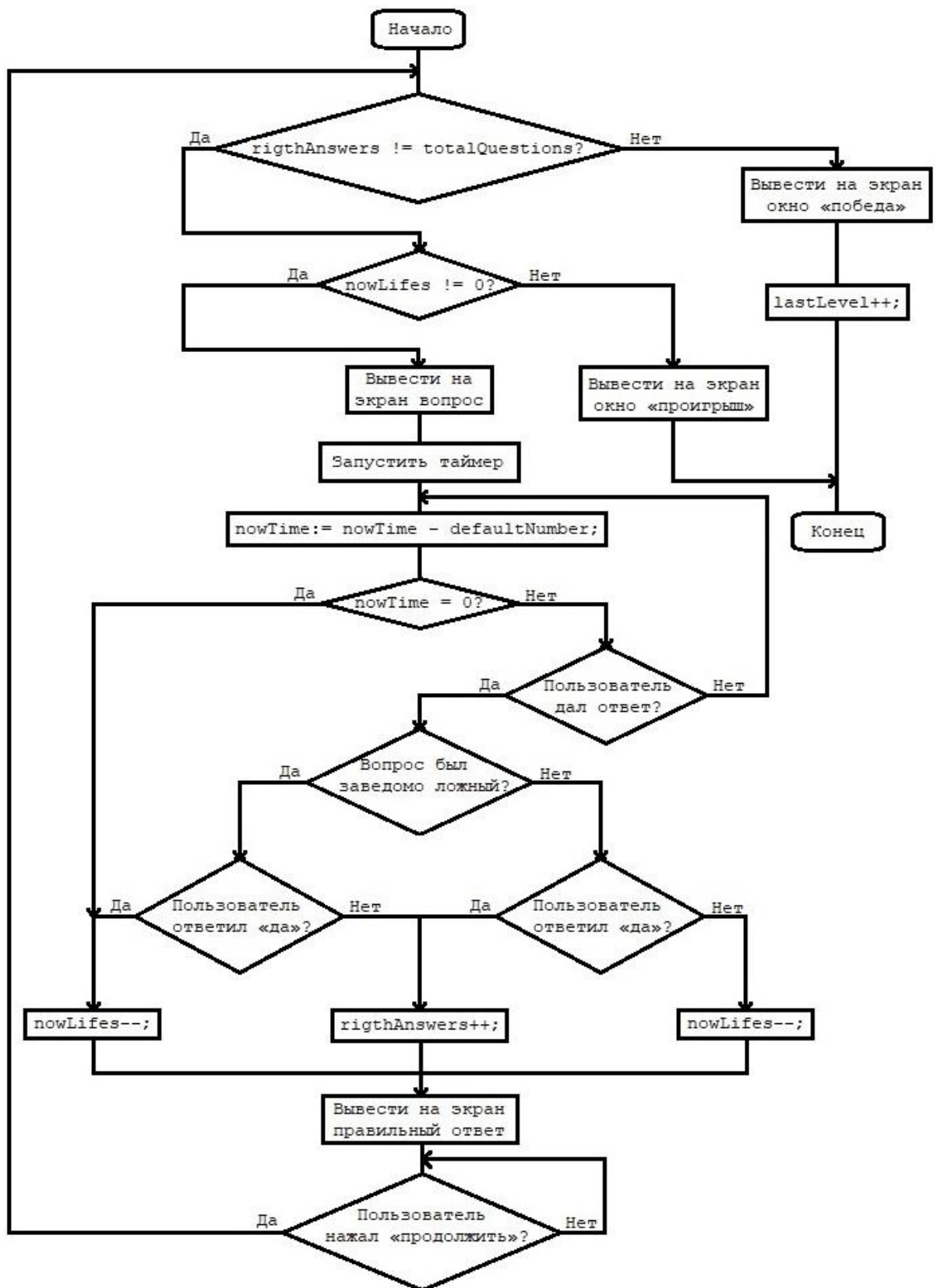


Рис. 11. Блок-схема алгоритма работы уровня игры

После успешной реализации всех основных компонентов приложения, переходим к этапу реализации пользовательского интерфейса.

### 3.3. Реализация пользовательского интерфейса. Работа с widget-файлами

В Unreal Engine реализация интерфейса происходит при помощи Widget-файлов [1]. Данная система имеет удобный интерфейс по типу Drag&Drop, что позволяет быстро создавать все необходимые элементы и сразу же назначать их параметры, задавая при необходимости макросы.

Опираясь на пункт 2.4. главы 2 «Проектирование приложения», был реализован интерфейс для каждого класса (рис. 12, рис. 13).

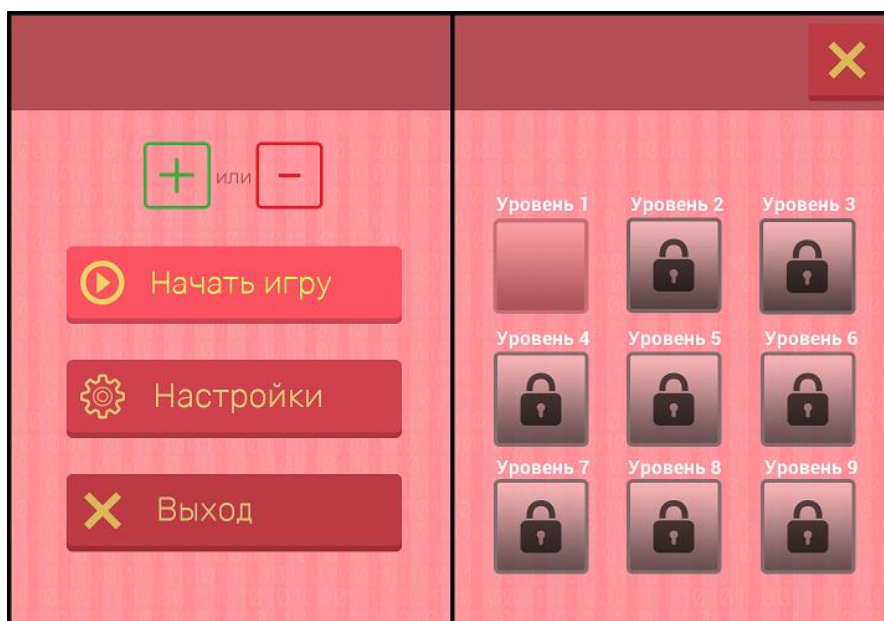


Рис. 12. Вид интерфейса *Главного меню* и *Меню выбора уровня*

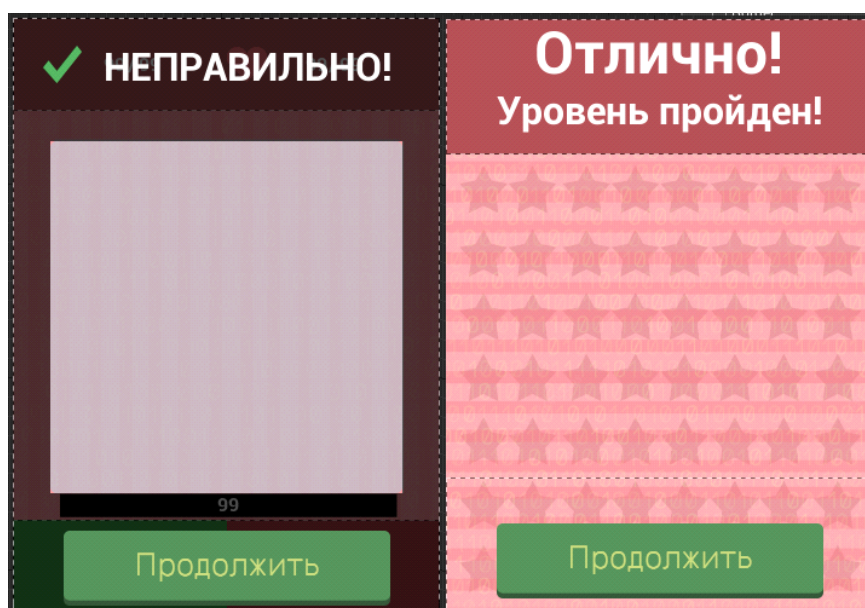


Рис. 13. Вид интерфейса *Уровня* и *Окна «Победы»*

## **Вывод**

В результате проектирования было создано и описано приложение со всеми его компонентами.

Базовая логика приложения была реализована посредством BluePrint.

Также был создан полноценный пользовательский интерфейс.

Для проверки корректности работы приложения и соответствия его предъявляемым ранее требованиям, необходимо провести функциональное тестирование, которое описано в следующей главе.

## 4. ТЕСТИРОВАНИЕ

Для проверки соответствия реализованного программного продукта ожиданию необходимо тестирование ПО.

Тестирование производилось при помощи конечного набора тестов. И проходило в два этапа.

1. Функциональное тестирование реализованных классов на соответствие требованиям, составленным в пункте проектирования;

2. Юзабилити-тестирование: тестирование приложения в учебном заведении в классах с учащимися различных возрастов с последующим получением рецензии от заместителя директора по учебной работе.

### 4.1. Функциональное тестирование

Функциональное тестирование – это тестирование программного обеспечения в целях проверки реализуемости функциональных требований [19], другими словами – способность ПО решать поставленные задачи.

Ниже будут представлены результаты заранее отобранных тестов, соответствующих функциональным требованиям.

**Тест № 1.** Отображение экрана главного меню приложения.

*Входные данные:* -

Цель: Проверить корректность отображения Главного меню.

*Результат:* тест пройден.

**Тест № 2.** Корректная работоспособность выбора уровня.

*Входные данные:* Приложение открыто на экране Меню Уровней.

Цель: Проверить запуск уровня после выбора его на экране.

*Результат:* тест пройден.

**Тест № 3.** Корректность работы Игрового поля.

*Входные данные:* Приложение открыто на экране игрового поля.

Цель: Проверить корректность работы игрового поля

Ход проведения:

1. Проверить работоспособность таймера, правильно ли отображаются цифры, точный ли подсчёт времени в секундах;

2. Дать один правильный ответ, получить информацию о верном ответе;

3. Дать один неправильный ответ, получить информацию о неверном ответе, проверить счётчик «жизней».

*Результат:* тест пройден.

**Тест № 4.** Корректное отображение экрана «Победы\Проигрыша».

*Входные данные:* Приложение открыто на экране игрового поля.

Цель: Проверить корректность перехода на экран победы\проигрыша в случае окончания прохождения уровня.

*Ход проведения:*

1. Отвечать на вопросы до окончания уровня;
2. Получить информации об успешном прохождении уровня\получить информацию о неуспешном прохождении уровня.

*Результат:* тест пройден.

**Тест № 5.** Корректность работы счётчика прохождения уровней.

*Входные данные:* Приложение открыто на экране «Победа».

Цель: Проверить работоспособность счётчика уровней.

*Ход проведения:*

1. Нажать кнопку «Продолжить»;
2. Убедиться, что пройденный уровень засчитан и доступен новый для прохождения.

*Результат:* тест пройден.

**Тест № 6.** Корректный сброс пройденных уровней.

*Входные данные:* Приложение открыто на экране Главное Меню.

Цель: Проверить корректность сброса уровней к начальному этапу.

*Ход проведения:*

1. Перейти в Настройки;
2. Убедиться, что уровни обнулились.

*Результат:* тест пройден.

Результаты некоторых тестов представлены на рис. 14.

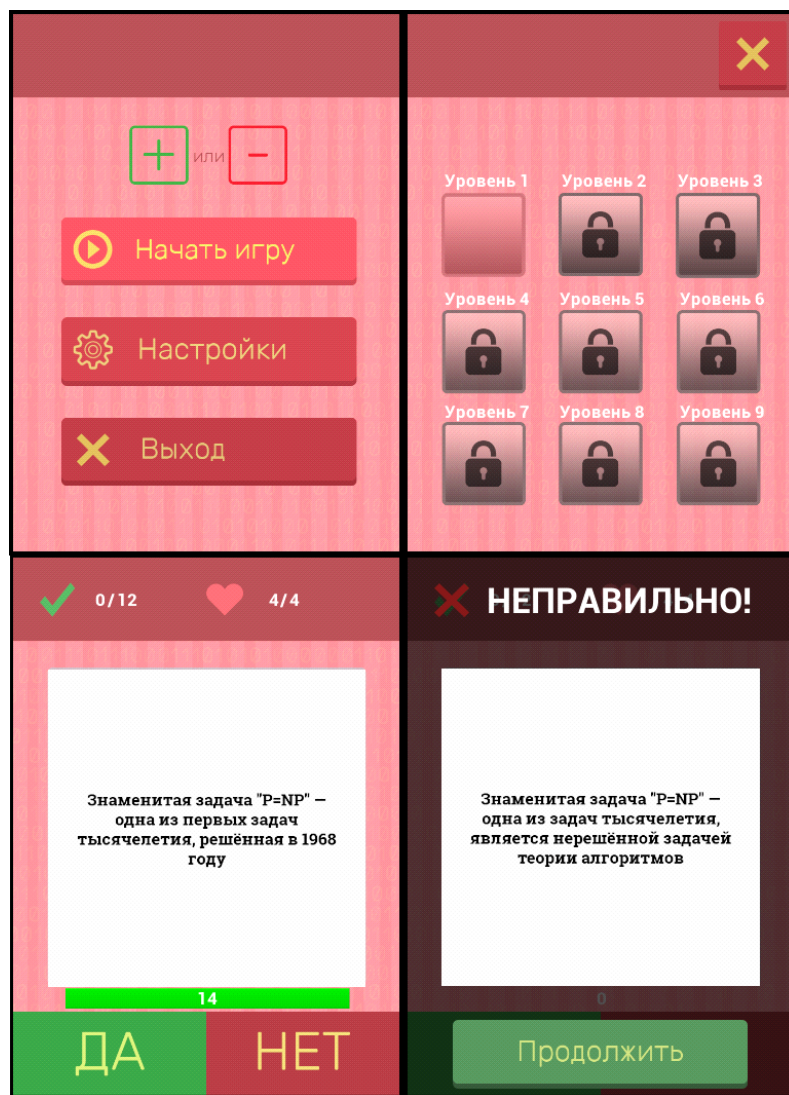


Рис. 14. Функциональное тестирование приложения

## 4.2. Юзабилити-тестирование

Юзабилити-тестирование – проверка удобства продукта в использовании [20]. Выполняется при привлечении пользователей для тестирования приложения.

Данный шаг тестирования приложения был выполнен в средней общеобразовательной школе № 3 г. Лангепас, ХМАО-Югры (далее ЛГ МАОУ «СОШ № 3»).

Проведя тестирование, была получена рецензия от заместителя директора по учебной работе (информатизация), учителя информатики высшей категории, которая рекомендовала внести небольшие коррективы в окончательный вариант приложения.



В итоге, учитывая рекомендации, указанные в рецензии, в приложении был доработан блок анализа производительности игрока.

Для оценки прохождения теста используется стандартная система: процент соотношения правильных ответов к количеству заданных вопросов (так называемый, процент прохождения теста учеником).

Итоговая оценка выставляется по следующим критериям:

1. «отлично» – 85–100 %;
2. «хорошо» – 70–84,99 %;
3. «удовлетворительно» – 50–69,99 %;
4. «неудовлетворительно» – 0–49,99 %.

Кроме этого, было учтено, что ученик может «схитрить» и проходить тест несколько раз, что искажает реальную оценку его знаний, так как существует вероятность, что ему попадутся одинаковые или схожие вопросы и он будет отвечать по памяти, не задумываясь над содержанием вопроса.

Для решения этой проблемы было принято решение, чтобы модуль оценки запоминал предыдущие прохождения теста, но не выводил итоговую оценку, пока тестируемый не закончит тест. В итоге, модуль оценки учитывает все прохождения теста и считает среднее значение.

*Пример*, если первый раз тестируемый ответил правильно на 1 вопрос и допустил 4 ошибки (что максимально для первого уровня), то система запоминает первое прохождение теста учеником, которое составило 1/5 или 20 % и соответствует оценке «Неудовлетворительно». Далее, если ученик пройдет тест еще раз, ответив на все 12 вопросов, совершив 0 ошибок (12/12 или 100 % – «Отлично»), то он получит среднее между первой и второй попыткой: суммарно 13 правильных ответов система делит на суммарно 17 заданных вопросов, 13/17 или 76 %, что соответствует оценке «Хорошо».

По описанным выше критериям был разработан модуль оценки игрока. Завершив реализацию этого модуля, мы приступили к его тестированию. Результаты некоторых, проведённых тестов, представлены на рис. 15 и рис. 16.

Уровень 1  
Прохождений: 1  
Правильные ответы: 12  
Неправильные ответы: 2  
Правильных ответов(%):85,714  
Ваша оценка: Отлично

Рис. 15. Прохождение тестирования с одного раза

Уровень 1	1	Уровень 1	2
Прохождений: 0		Прохождений: 1	
Правильные ответы: 1		Правильные ответы: 13	
Неправильные ответы: 4		Неправильные ответы: 5	
Правильных ответов(%):20		Правильных ответов(%):72,222	
Ваша оценка: Неуд.		Ваша оценка: Хорошо	

Рис. 16. Прохождение тестирования с нескольких попыток

Из рис.15 видно, что прохождение тестирования пользователем с первого раза возможно на оценку «отлично» (85 %).

Из рис.16 видно, что прохождение тестирования пользователем за несколько попыток даст следующий результат: в первый раз на оценку «неудовлетворительно», а во второй раз на оценку «отлично». Итоговая оценка за тест – «хорошо».

### **Вывод**

Готовое приложение прошло два вида тестирования: функциональное и юзабилити-тестирование.

Все тесты, из заданных наборов, были пройдены успешно.

Приложение было доработано по рекомендации, указанной в рецензии на компьютерное приложение «Правда или ложь», которое тестировалось на учениках разных классов в средней общеобразовательной школе № 3 г.Лангепаса, Ханты-Мансийского автономного округа.

## **ЗАКЛЮЧЕНИЕ**

С каждым годом увлечение играми только увеличивается среди молодёжи. Ни один вид досуга не способен дать ту интерактивность, что дают нам именно игры. Поэтому внедрение игр в образовательный процесс – вопрос времени.

Целью работы являлась разработка образовательного приложения на платформе Unreal Engine 4, с использованием методов Educational Data Mining.

В процессе работы был проведён анализ предметной области, который показал актуальность и важность выбранной тематики. Анализ аналогов и технологий реализации подобных приложений позволил сформулировать наиболее оптимальные требования, а также выбрать наиболее универсальную и современную платформу для реализации проекта.

На этапе проектирования были подробно сформулированы функциональные и нефункциональные требования, что позволило избежать ошибок при разработке на ранних этапах и выбрать наиболее выигрышную стратегию реализации.

В ходе реализации проекта были получены навыки работы с новыми методами, а также приобретен опыт работы с документацией, указанной в спецификации к Unreal Engine.

На первом этапе тестирования были успешно проверены функциональные требования, сформулированные на этапе проектирования. Второй этап тестирования в общеобразовательном учреждении показал недоработки приложения, что позволило в краткие сроки их исправить.

Созданный модуль анализа производительности игрока (обучающегося) также был успешно протестирован, а нелинейная система оценки знаний пользователя позволила адекватно оценивать знания.

Таким образом, завершив все поставленные задачи, было создано рабочее компьютерное приложение, которое можно использовать не только

для оценки знаний обучающихся, подготовки к различным тестам, но и для саморазвития тестируемого, чему способствует адаптивность модуля EDM.

В дальнейшем, после соответствующей модификации: разработки банка вопросов теста по программам курса начального общего (НОО), основного общего (ООО) и среднего общего образования (СОО), приложение может быть использовано в том числе и в общеобразовательных учреждениях.

Таким образом, дипломная работа имеет практическую значимость при проектировании и создании приложений для электронного обучения при помощи Unreal Engine.

## СПИСОК ЛИТЕРАТУРЫ

1. Интернет-портал Unreal Engine 4. Documentation, [Электронный ресурс] URL: <https://docs.unrealengine.com/latest/INT> (дата обращения: 20.02.2019).
2. Богачева Н.В. Войскунский А.Е. Спецификация когнитивных стилей и функций контроля у геймеров. // Психологические исследования. 2014. Т. 7, №38. С. 1. [Электронный ресурс] URL: <http://psystudy.ru> (дата обращения: 24.04.2019).
3. ГОСТ 19.701-90 (ИСО 5807-85). Единая система программной документации. 1992 – 24 с.
4. Денисова А.И. Компьютерные игры как феномен современной культуры: УДК аспирантки ТГУ им. Державина. / А.И. Денисова. – Тюмень: изд-во ТГУ, 2014 – 3 с.
5. Дубровина О.В. Психологические особенности личности с виртуальной аддикцией. // Сибирский педагогический журнал, 2009.– № 1. – С. 333–341.
6. Каткова А.Л. Категория цели в педагогической теории компьютерных игр. // Вестник Вятского государственного университета, 2009. – Т. 3. № 3. – С. 163–167.
7. Доусон М. Изучаем C++ через программирование игр. / М. Доусон. – М: АСТ: Астрель, 2016. – 353 с.
8. Плешаков В.А., Наместников В.В. Компьютерные игры как фактор киберсоциализации человека в XXI веке. // Среднее профессиональное образование, 2013. – № 8. – С. 36–37.
9. Romero C. Handbook of Educational Data Mining. / C.Romero, S.Ventura, Pechenizkiy, R.S. Baker. // Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Taylor and Francis Group, LLC, 2011.– 503 p.
10. The State of Mobile in 2019 – The Most Important Trends to Know. [Электронный ресурс] URL: <https://appannie.com/ru/insights/market-data/the-state-of-mobile-2019> (дата обращения: 18.03.2019).

11. Официальный сайт Google Play. [Электронный ресурс] URL: [https://play.google.com/store/apps/details?id=com.alexapp-company.True\\_FalseNew&hl=ru](https://play.google.com/store/apps/details?id=com.alexapp-company.True_FalseNew&hl=ru) (дата обращения: 20.03.2019).
12. Официальный сайт Google Play. [Электронный ресурс] URL: <https://play.google.com/store/apps/details?id=com.alpopstudio.truefalse&hl=ru> (дата обращения: 20.03.2019).
13. Официальный сайт Google Play. [Электронный ресурс] URL: <https://play.google.com/store/apps/details?id=ru.onyanov.yesorno&hl=ru> (дата обращения: 20.03.2019).
14. Официальный сайт libGDX. [Электронный ресурс] URL: <https://libgdx.badlogicgames.com> (дата обращения: 22.03.2019).
15. Официальный сайт Unity. [Электронный ресурс] URL: <https://unity.com/ru> (дата обращения: 22.03.2019).
16. Дин Л., Дон У. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. – М.: Издательский дом, «Вильямс», 2002. – 448 с.
17. Арлоу Д., Нейштадт И. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2007. – 624 с.
18. Saunders K., Novak J. Game Development Essentials: Game Interface Design. – USA: Cengage Learning, 2012. – 320 p.
19. Липаев В. Тестирование компонентов и комплексов программ. – М.: Директ-Медиа, 2015. – 528 с.
20. Kaner C., Falk J., Nguyen H. Q. Testing Computer Software. – USA: Wiley Computer Publishing, 1999. – 479 p.
21. Абдуллаев С.М., Ленская О.Ю., Салал Я.К. Компьютерные системы индивидуального обучения: особенности модели студента // Университет XXI века в системе непрерывного образования. Материалы IV международной научно-практической конференции, 11-12 октября 2018, Челябинск, – С. 7–14.

22. Виленская Г.А. Исполнительные функции: природа и развитие // Психологический журнал 2016. – Т. 37, – №4. – С. 21–31.

23. Абдуллаев С.М., Ленская О.Ю., Салал Я.К. Компьютерные системы индивидуального обучения: предпосылки и перспективы // Вестник ЮУрГУ. Серия «Образование. Педагогические науки», 2018. – Т. 10, №4. – С. 64–71.

24. Slater S., Joksimovic S., Kovanovic V., et al. Tools for Educational Data Mining: A review // Journal of Educational and Dehavioral Statistics, 2017. – Vol. 42, no. 1. – pp. 85–106.

25. Salal Y.K., Abdullaev S.M. Using of Data Mining Techniques to Predict of Student`s Performance in Industrial Institute of Al-Diwaniyah, Iraq. // Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronic, 2019. – Vol. 19, no. 1. – pp. 121–130.