

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____ Д.В. Топольский
« ___ » _____ 2022 г.

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ОБУЧЕНИЯ ГРАМОТЕ ДЕТЕЙ
МЛАДШЕГО ШКОЛЬНОГО ВОЗРАСТА С РАССТРОЙСТВАМИ ЧТЕНИЯ И
ПИСЬМА

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУРГУ-090301.2022.260 ПЗ ВКР

Руководитель работы,
к.п.н., доцент каф. ЭВМ
_____ Ю.Г. Плаксина
« ___ » _____ 2022 г.

Автор работы,
студент группы КЭ-406
_____ А.Ф. Якимов
« ___ » _____ 2022 г.

Нормоконтролёр,
к.п.н., доцент каф. ЭВМ
_____ М.А. Алтухова
« ___ » _____ 2022 г.

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

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

_____ Д.В. Топольский

«___» _____ 2022 г.

ЗАДАНИЕ

на выпускную квалификационную работу бакалавра
студенту группы КЭ-406
Якимову Алексею Феликсовичу,
обучающемуся по направлению
09.03.01 «Информатика и вычислительная техника»

1. **Тема работы:** «Разработка приложения для обучения грамоте детей младшего школьного возраста с расстройствами чтения и письма» утверждена приказом по университету от 12 декабря 2021 г. № 308/141

2. **Срок сдачи студентом законченной работы:** 3 июня 2022 г.

3. **Исходные данные к работе**

3.1. Бизнес-требования

3.1.1. Приложение для обучения грамоте обеспечивает возможности:

– младшим школьникам – получать знания в комфортных условиях;

– родителям – отслеживать динамику развития ребенка;

– специалистам в области образования и медицины – повысить эффективность занятий.

3.2. Бизнес-правила

3.2.1. Приложение должно быть разработано на основе следующих документов:

- № 273-ФЗ «Об образовании в Российской Федерации»;
- № 373 «Об утверждении и введении в действие федерального государственного образовательного стандарта начального общего образования».

3.3. Требования к эргономике и технической эстетике

3.3.1. Для представления текстовых и алфавитно-цифровых данных должны использоваться прямые шрифты без засечек, размером не менее 18 пт.

3.3.2. Все надписи должны быть выполнены на русском языке.

3.3.3. Коэффициент контрастности текста с фоном должен быть не ниже 4,5.

3.4. Ограничения

3.4.1. Приложение должно быть реализовано на языке программирования C# с использованием среды разработки Unity.

3.4.2. Для реализации алгоритмов манипулирования данными должен использоваться стандартный язык запроса к данным SQL и его процедурное расширение PL/pgSQL.

3.5. Функциональные требования

3.5.1. Упражнения должны быть основаны на методике учителя-логопеда А. Е. Игнатьева [16] и представлены в виде отдельных мини-игр, каждая из которых направлена на решение определенной коррекционной задачи.

3.5.2. Для уровня доступа «Ребенок» должен быть предусмотрен беспарольный вход в приложение.

3.5.3. Доступ к настройкам приложения без ввода пароля должен быть запрещен.

3.5.4. В настройках приложения должны быть реализованы основные операции со словарями (добавление, удаление, изменение).

3.6. Системные требования

3.6.1. Конфигурация рабочей станции:

– 32/64-разрядная операционная система Microsoft Windows версии 7 и выше, Linux версии 3.2 и выше;

– тактовая частота процессора не менее 200 МГц;

– объем оперативной памяти не менее 128 МБ;

– объем видеопамяти не менее 16 МБ;

– свободное пространство на жестком диске не менее 200 МБ;

3.6.2. Конфигурация сервера:

– 32/64-разрядная операционная система Microsoft Windows Server версии 2003 и выше, серверная сборка Linux версии 3.2 и выше;

– тактовая частота процессора не менее 500 МГц;

– объем оперативной памяти не менее 1 ГБ;

– свободное пространство на жестком диске не менее 10 ГБ.

4. Перечень подлежащих разработке вопросов

4.1. Анализ предметной области.

4.1.1. Обзор аналогов (выбор и обоснование критериев для сравнения, анализ существующих решений, выявление их особенностей, преимуществ и недостатков).

4.1.2. Анализ основных технологических решений (обзор современных инструментальных средств разработки, подходящих для решения поставленных задач).

4.2. Проектирование.

4.2.1. Архитектура предлагаемого решения (описание структуры программного обеспечения, взаимодействия модулей и выполняемых ими функций).

4.2.2. Концепция игры (общие сведения об игре, описание игрового процесса, прототипирование игровых механик).

4.2.3. Описание данных (создание схемы базы данных и описание сущностей).

4.3. Реализация.

4.3.1. Средства разработки (обоснованный выбор инструментальных средств разработки применительно к поставленным задачам и требованиям).

4.3.2. Реализация модуля доступа к базе данных (написание кода для соединения с базой данных и выполнения SQL-запросов).

4.3.3. Реализация модуля контроля доступа (написание кода для авторизации пользователей).

4.3.4. Реализация модуля коррекционной работы (написание кода, отвечающего за логику работы мини-игр, измерение и оценку результатов).

4.3.5. Реализация модуля формирования отчетов (написание кода для графического отображения данных об успеваемости, полученных из базы данных).

4.3.6. Файловая структура (физическое представление проекта, описание классов и других используемых ресурсов).

4.4. Тестирование (проектирование и разработка тестов, проведение тестирования и интерпретация результатов).

5. **Дата выдачи задания:** 1 декабря 2021 г.

Руководитель работы _____ /Ю.Г. Плаксина/

Студент _____ /А.Ф. Якимов/

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	10.03.2022	
Проектирование архитектуры программного обеспечения	15.03.2022	
Разработка концепции игры	18.03.2022	
Проектирование и разработка базы данных	21.03.2022	
Реализация модуля доступа к базе данных	22.03.2022	
Реализация модуля контроля доступа	24.03.2022	
Реализация модуля коррекционной работы	01.04.2022	
Реализация модуля формирования отчетов	04.04.2022	
Проектирование и разработка тестов	11.04.2022	
Проведение тестирования и интерпретация результатов	25.04.2022	
Компоновка текста работы и сдача на нормоконтроль	16.05.2022	
Подготовка презентации и доклада	24.05.2022	

Руководитель работы _____ /Ю.Г. Плаксина/

Студент _____ /А.Ф. Якимов/

АННОТАЦИЯ

А.Ф. Якимов. Разработка приложения для обучения грамоте детей младшего школьного возраста с расстройствами чтения и письма. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2022, 81 с., 15 ил., библиогр. список – 41 наим.

В рамках выпускной квалификационной работы проводится детальный анализ наиболее популярных современных решений для формирования у детей навыков чтения и письма. Организуется разработка программного обеспечения для решения проблемы нарушений письменной речи у младших школьников. Обучение грамоте проходит в игровой форме и включает в себя различные мини-игры на поиск букв, слогов или слов. Вариативность содержания и сложности упражнений достигается за счет использования пользовательских словарей. Программное обеспечение осуществляет сбор данных об успеваемости учащихся и имеет возможность представления результатов в графическом виде.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	10
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	14
1.1 Обзор аналогов	15
1.1.1 SLOGY	17
1.1.2 Fast ForWord	19
1.1.3 GraphoGame	20
1.1.4 Дэльфа-142.1	21
1.1.5 Баба-Яга учится читать	22
1.1.6 Букварь	22
1.1.7 Игры для тигры	23
1.1.8 Мерсибо	24
1.1.9 Уроки мудрой совы	25
1.1.10 Играемся	25
1.1.11 Вывод	25
1.2 Анализ основных технологических решений	27
2 ПРОЕКТИРОВАНИЕ	30
2.1 Архитектура предлагаемого решения	30
2.2 Концепция, игровой процесс и сценарий	32
2.3 Описание данных	33
3 РЕАЛИЗАЦИЯ	35
3.1 Средства разработки	35
3.2 Реализация модуля доступа к базе данных	36
3.3 Реализация модуля контроля доступа	37
3.4 Реализация модуля коррекционной работы	38
3.5 Реализация модуля формирования отчетов	43

3.6 Файловая структура	44
4 ТЕСТИРОВАНИЕ	48
ЗАКЛЮЧЕНИЕ	51
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	53
ПРИЛОЖЕНИЕ А	58

ВВЕДЕНИЕ

В последнее время значительно возросло количество детей, сталкивающихся с различными трудностями обучения в начальной школе. Проблема нарушений письменной речи – одна из самых важных, поскольку письмо и чтение не только являются базовыми средствами для дальнейшего получения знаний, но и стимулируют психическое развитие, влияют на формирование личности и самооценку ребенка. Отсутствие своевременной коррекции приводит к снижению работоспособности, повышению утомляемости, утрате интереса к учебе и замкнутости. По данным Международной Ассоциации Дислексии [26], 15-20% людей в мире имеют нарушения чтения (дислексия) и письма (дисграфия) и связанные с ними трудности, препятствующие социализации и устройству в жизни.

Традиционные методы коррекции дислексии и дисграфии основаны на индивидуальных и групповых логопедических занятиях с использованием разнообразного речевого и дидактического материала. Комплекс упражнений подбирается в зависимости от типа и степени тяжести нарушения. Выделяют упражнения для развития зрительного и слухового восприятия, пространственной ориентации, речевого аппарата и голоса, тренировки мелкой моторики рук.

Как правило, процесс формирования навыков чтения и письма занимает длительное время и проходит с большими трудностями. Применение информационных технологий позволяет повысить заинтересованность и мотивацию учащихся, а также использовать различные формы подачи материала. Движение, звук, мультипликация – все это привлекает внимание ребенка. Помощь игровому персонажу в поиске решения той или иной проблемы является стимулом познавательной активности детей. А наличие

различных способов поощрения за правильные ответы и возможность исправления ошибок помогают ребенку обрести уверенность в себе. Кроме того, внедрение информационных технологий во все сферы человеческой деятельности позволяет упростить и ускорить решение как личностных, так и производственных задач.

В настоящее время современная педагогическая сфера наиболее подвержена влиянию информационных технологий. В связи с распространением коронавирусной инфекции (COVID-19) в 2020 году многие образовательные учреждения были вынуждены организовывать учебный процесс посредством применения электронного обучения и дистанционных образовательных технологий. Неготовность к резкой смене формата обучения вызвала ряд проблем [1]. Во-первых, многие учащиеся и педагоги столкнулись с такими препятствиями, как нехватка оборудования, отсутствие стабильного интернет-соединения и недостаточная «цифровая» компетенция. Во-вторых, перед учебными заведениями возникла необходимость реорганизации процессов проверки и оценки знаний, создания центра психологической и финансовой поддержки. По данным исследования Европейского пространства высшего образования [27], учащиеся испытывали трудности в усвоении материала, увеличился объем учебной нагрузки.

Из-за низкой вовлеченности и недостаточного качества дистанционного обучения рекомендуется разнообразить занятия наличием красочных презентаций, интерактивных заданий, обучающих видеоматериалов. Кроме того, согласно ряду исследований [2, 3, 4], некоторые компьютерные и мобильные игры также могут быть использованы для повышения эффективности образовательного процесса. Они не только помогают расслабиться и снять психическое напряжение, но и способствуют развитию зрительного внимания, пространственного мышления, памяти и других важных

навыков. Особенно полезны приложения, которые носят развивающий и обучающий характер. Применение компьютерных и мобильных игр в образовании имеет множество преимуществ, таких как высокая мотивация, наглядность, активное участие и др. Данная форма обучения также является легкодоступной, поскольку проведенные MOMRI опросы [2] показали, что сегодня почти у каждого ребенка есть компьютер, мобильный телефон или планшет, а количество разрабатываемых игр растет с каждым годом. В этом можно легко убедиться, изучив отчеты исследований [28, 29], проведенных компаниями Google и Newzoo. По данным аналитиков, мировой игровой рынок в 2020 году составил 177,8 миллиардов долларов, что на 23,1 % больше по сравнению с предыдущим годом. Более того, эксперты прогнозируют дальнейший рост продаж и преодоление порога в 200 миллиардов долларов в 2023 году.

Целью настоящей работы является разработка обучающего приложения для детей младшего школьного возраста с расстройствами чтения и письма. Данное приложение может использоваться как в домашних условиях, так и в образовательных и медицинских учреждениях. Для достижения поставленной цели необходимо решить следующие задачи:

- 1) проанализировать аналогичные решения;
- 2) спроектировать архитектуру программного обеспечения;
- 3) выполнить программную реализацию в соответствии с заданными требованиями;
- 4) протестировать разработанное программное обеспечение.

Выпускная квалификационная работа имеет следующую структуру:

- 1) анализ предметной области – обзор аналогов и технологических решений;

2) проектирование – разработка архитектуры программного обеспечения, схемы базы данных и концепции игры;

3) реализация – выбор средств разработки, программирование модулей, описание файловой структуры проекта;

4) тестирование – проверка соответствия программного обеспечения требованиям.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

По мнению С. А. Филенковой [13], дисграфия сегодня имеет широкое распространение и в том или ином виде присутствует у 40 % детей, обучающихся в начальных классах. В статье автор приводит классификацию нарушений письменной речи в соответствии со стойкими специфическими ошибками, допускаемыми младшими школьниками в работах, и предлагает некоторые методы и приемы коррекции. Филенкова также обращает внимание на то, что нарушения письма и чтения тесно связаны между собой, т. е. дисграфия в большинстве случаев возникает вместе с дислексией.

В следующих работах [5, 6, 7] рассматриваются преимущества информатизации образования. Авторы сходятся во мнении, что информационные технологии повышают интерес учащихся к овладению учебным материалом, упрощают его восприятие и обеспечивают возможность персонализации обучения. Также не менее важной является способность компьютера выступать в роли учителя в ходе самостоятельной домашней работы, когда весь необходимый для изучения материал уже был представлен на занятии, поскольку она избавляет педагогов от излишней рутинной работы, с которой ученик в состоянии справиться сам. Для этой цели существуют специальные обучающие программы-тренажеры, содержащие тренировочные упражнения для отработки некоторого правила или развития навыка. В одной из приведенных работ особое внимание уделяется использованию игровых элементов и механик при разработке обучающих приложений. Авторы объясняют это тем, что основная задача учителя начальных классов, согласно требованиям федерального государственного образовательного стандарта начального общего образования (ФГОС НОО) [25], состоит в формировании

учебной мотивации у младших школьников, а игра в данном случае является лучшим решением.

1.1 Обзор аналогов

Обязательным требованием для внедрения цифровых образовательных ресурсов в учебный процесс является их соответствие ФГОС. Поэтому данный критерий является решающим при выборе подобных продуктов.

В результате анализа исследований [30] по применению цифровых образовательных ресурсов в процессе обучения детей, испытывающих трудности в освоении учебных программ, авторами был выделен ряд критериев, в соответствии с которыми коррекционно-развивающее программное обеспечение должно:

- 1) использовать в качестве учебной стратегии повторение, т. е. отработку одного и того же навыка снова и снова;
- 2) обеспечивать наличие различных уровней сложности в зависимости от индивидуальных способностей ученика;
- 3) включать последовательные и лаконичные инструкции, которые учащиеся могут повторить;
- 4) предоставлять положительную обратную связь, актуальную для действий, предпринятых учащимися;
- 5) автоматически регистрировать ответы учащихся;
- 6) обеспечивать возможность вывода отчетов об успеваемости учащихся;
- 7) иметь незагроможденный интерфейс с последовательными и понятными элементами управления;
- 8) требовать от пользователя минимальных навыков работы с клавиатурой.

Стратегия повторения является наиболее простой, т. к. она не требует больших затрат энергии и глубокой проработки учебного материала, а задействует ресурсы, которые находятся на поверхности. Кроме того, формирование любого навыка осуществляется именно путем многократного повторения какого-либо действия и доведения его до автоматизма.

Наличие различных уровней сложности позволяет охватить большую аудиторию и организовать учебный процесс, адаптированный к способностям ученика. Это способствует повышению эффективности усвоения материала и формированию адекватной самооценки учащихся.

Для того чтобы ребенок мог самостоятельно выполнять задания без помощи взрослого, инструкции должны быть представлены на понятном для него языке и содержать минимальный объем информации.

Обратная связь оказывает положительное влияние на результативность учащихся и способствует формированию у них уверенности в себе, повышению мотивации к обучению. При своевременной подаче обратной связи ученик положительно реагирует на отзыв и связывает его с последним совершенным действием, закрепляя учебный материал в памяти.

Программное обеспечение должно давать мгновенный автоматический отклик на ответы пользователя, чтобы у ребенка не возникало сомнений в правильности выбранного ответа.

Важно также отслеживать динамику эффективности коррекционных занятий и своевременно вносить необходимые изменения в программу. Для этого программное обеспечение должно иметь возможность предоставления информации об успеваемости учащихся.

Загроможденный интерфейс отвлекает и рассеивает внимание ребенка, перегружает его излишней информацией. Это ведет к появлению ошибок и негативно сказывается на учебном процессе.

Дети младшего школьного возраста с нарушениями чтения испытывают трудности при наборе текста с клавиатуры, также у многих из них недостаточно развита мелкая моторика рук. Поэтому рекомендуется преимущественно использовать компьютерную мышь для управления.

Перед началом проектирования был проведен поиск и анализ уже существующих и зарекомендовавших себя на рынке развивающих игр для детей, испытывающих трудности в овладении письменной речью.

1.1.1 SLOGY

SLOGY [17] – это онлайн-платформа, образовательный процесс которой реализован в виде регулярных ежедневных занятий. Урок представляет собой комплекс упражнений по методике профессора А. Н. Корнева [18], направленной на развитие различных навыков, необходимых для успешного овладения чтением. Сложность заданий адаптируется под способности ребенка, а наградой за выполнение служит изменение одежды у игрового персонажа. Интерфейс программы представлен на рисунке 1.

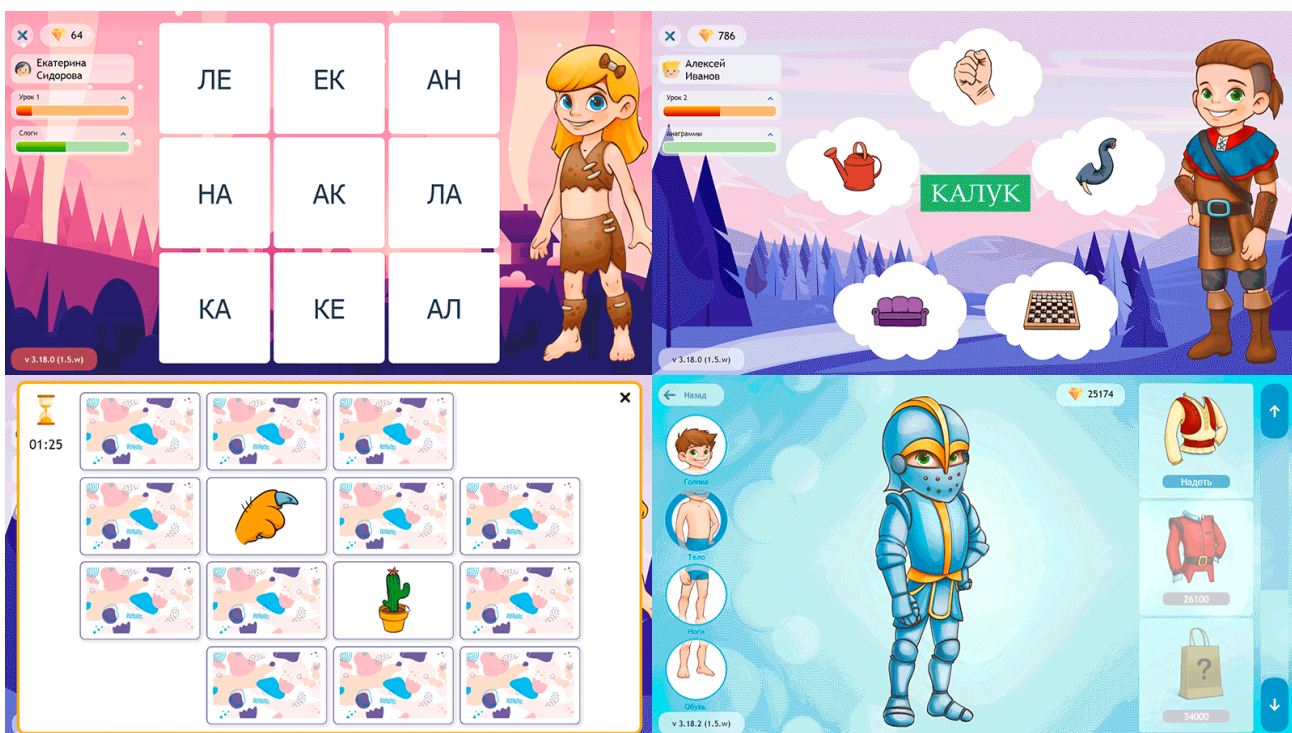


Рисунок 1 – Интерфейс программы SLOGY [18]

Программа может быть запущена на любом устройстве из браузера при наличии подключения к Интернету. Также для большей эффективности рекомендуется проводить занятия в тихом помещении с использованием наушников или динамиков, обеспечивающих высокое качество звука, поскольку задание озвучивается диктором только один раз. Минимальная стоимость подписки составляет 1666 руб. в месяц при годовой оплате, при этом пользоваться продуктом можно не более 10 минут в день (длительность одного занятия). Общая продолжительность обучения по данной методике может достигать 2 лет и более, а способы мотивации, предлагаемые программой, являются недостаточно эффективными для долгосрочных занятий.

1.1.2 Fast ForWord

Компьютерная программа «Fast ForWord» [38] предназначена для развития базовых речевых навыков, тренировки памяти и внимания, способности обработки информации. Обучение проходит по схожему принципу: 5 раз в неделю по 30-40 минут в день. Продолжительность курса при такой интенсивности нагрузки, по словам авторов программы, составляет 2-3 месяца. Программа разделена на несколько серий, каждая из которых содержит по 5-7 игр и направлена на развитие того или иного навыка. Сложность заданий регулируется автоматически исходя из скорости и качества ответов обучающегося. К недостаткам приложения можно отнести высокую стоимость (пакет «Базовый» – 20 624 руб. в месяц) и отсутствие русскоязычного интерфейса. Интерфейс программы представлен на рисунке 2.



Рисунок 2 – Интерфейс программы Fast ForWord [38]

1.1.3 GraphoGame

Мобильное развивающее приложение GraphoGame: Kids Learn to Read [39] (рисунок 3) обучает детей основным навыкам чтения. В игре применяется подход постепенного увеличения сложности упражнений, начиная с букв и звуков и заканчивая словами. В качестве награды за прохождение уровня игрок получает внутриигровую валюту, которую можно потратить на приобретение предметов одежды для своего персонажа. Взрослые также могут быть вовлечены в процесс обучения ребенка, поскольку в GraphoGame организован сбор информации как общей (дата последнего входа, игровое время и т. д.), так

и результатов упражнений (по каждому слову: количество повторов, правильных и неправильных ответов), позволяющих выявить проблемы в успеваемости и оказать дополнительную поддержку. Эффективность занятий рассчитывается как отношение количества правильных ответов к числу вопросов. К сожалению, игра не адаптирована для русских пользователей.



Рисунок 3 – Мобильное приложение GraphoGame [39]

1.1.4 Дэльфа-142.1

Логопедический тренажер «Дэльфа-142.1» [19] представляет собой программно-аппаратный комплекс для работы над устной и письменной речью. В комплект входит микрофон, блок обработки сигнала, специализированное программное обеспечение и текстовое практическое руководство с рекомендациями для педагогов и психологов. В приложении имеется большое количество разнообразных упражнений, разделенных на группы в зависимости от используемого стимульного материала (звук, буква, слог, слово, предложение, текст). Индивидуальный подход к каждому ученику обеспечивается за счет уровневой системы и набора словарей как базовых, так и пользовательских.

1.1.5 Баба-Яга учится читать

Несмотря на то, что компьютерная программа «Баба-Яга учится читать» изначально разрабатывалась для детей, которые только начинают учить буквы, она пользуется популярностью и у тех, кто уже начал читать свою первую книгу. Это связано с использованием в игровых упражнениях и головоломках методики опережающего обучения. Программа способствует развитию у детей навыков звукового анализа и синтеза, помогает формированию слухового восприятия.

В приложении присутствует красочная мультипликация и анимация, качественное звуковое сопровождение, проработанный сценарий и интересные персонажи с яркими характерами. За прохождение игры ребенок награждается шуточной грамотой и получает возможность послушать веселые частушки про каждую букву в исполнении Бабы-Яги. Для более способных детей игра может показаться слишком легкой и скучной, а возможность изменения уровня сложности отсутствует. Также при многократном прохождении теряется коррекционный эффект в связи с запоминанием встречающихся в упражнениях слов.

1.1.6 Букварь

Электронное пособие «Букварь» ориентировано на обучение детей с ОВЗ дошкольного и младшего школьного возраста по методике профессора А. Ф. Малышевского. Структура программы представлена в виде отдельных разделов для каждой буквы, в которые входит ознакомительная информация и несколько простых заданий. Перемещение между буквами осуществляется по нажатию на порядковый номер или с помощью стрелок. Также на экране доступна кнопка

для перехода в режим обучения. Программа снабжена красочными иллюстрациями, задания и стихотворения озвучены дикторами. Поскольку дети быстро теряют интерес к однообразным заданиям, а из всех видов мотивации применяется только положительная обратная связь, данный способ обучения наиболее эффективен в присутствии педагога или родителя.

1.1.7 Игры для тигры

Компьютерная коррекционная технология «Игры для тигры» [20] предназначена для исправления дефектов речи у детей дошкольного и младшего школьного возраста. В состав программы входят 50 упражнений, разделенных на 4 тематических блока: фонематика, просодика, лексика и звукопроизношение. Занятия основаны на методике кандидатов педагогических наук Л. Р. Лизуновой и З. А. Репиной и представлены в легкой и понятной ребенку форме. Большинство упражнений требует наличия качественного микрофона, а также присутствия взрослого для контроля правильности произношения, дыхания и т. д. Интерфейс программы представлен на рисунке 4.



Рисунок 4 – Интерфейс программы Игры для Тигры [20]

1.1.8 Мерсибо

На сегодняшний день на портале Мерсибо [21] представлено 337 развивающих онлайн-игр для детей от 4 до 9 лет, 33 из которых используются для обучения звуковому анализу и чтению слов по слогам. Помимо возраста сложность некоторых заданий может быть отрегулирована рядом параметров (скорость, количество повторений и др.). В процессе игры за выполнение некоторых заданий игрок получает достижения.

При регистрации на портале предлагается выбрать тип учетной записи: специалист или родитель. Для специалистов программное обеспечение распространяется в форме подписки, количество доступных игр и наличие дополнительных возможностей зависит от выбранного тарифа. Учетная запись родителя включает 8 бесплатных игр, а для открытия каждой новой игры требуется одноразовый взнос в 50-70 билетиков, в зависимости от ее уровня сложности. Последующие открытия данных игр будут стоить 15 билетиков в сутки. Билетики – это внутриигровая валюта, которую можно получить игровым путем, либо приобрести на сайте.

Специалистам доступны дополнительные возможности для работы с детьми, такие как создание собственных пособий с помощью встроенного конструктора картинок, подключение родителей и их детей, ведение речевой и психологической карты ребенка, выдача домашних заданий и др. В дневнике ребенка отображается время и число повторов по каждой игре, но отсутствует информация о количестве и характере допускаемых ошибок, динамике успеваемости и т. д.

1.1.9 Уроки мудрой совы

Сборник мини-игр «Уроки мудрой совы» [22] был разработан с целью оказания помощи взрослому в подготовке детей к успешному обучению в школе. Учебный материал программы представлен в виде 37 последовательных глав, каждая из которых содержит несколько упражнений, разбитых на три группы по уровню сложности. Большинство заданий однообразны и непонятны, поэтому для повышения эффективности занятий требуется наличие внешнего контроля со стороны педагога или родителя.

1.1.10 Играемся

Детский портал «Играемся» [23] содержит множество интересных и познавательных игр, направленных на развитие внимания, памяти и мышления – базовых навыков, без которых невозможно научиться читать. Целью данного проекта является подготовка детей к школе, поэтому для тех, кто уже учится в начальных классах, игровые задания могут показаться слишком простыми.

1.1.11 Вывод

Сравнение рассмотренных комплексов и программ представлено в таблице 1.

Таблица 1 – Сравнение аналогов

Критерий	SLO GY	Fast ForW ord	Grap hoGa me	Дэль фа- 142.1	Баба- Яга учится читать	Бук вар ь	Игры для тигр ы	Мер сибо	Уроки мудро й совы	Игр аемс я
ФГОС	–	–	–	+	–	–	+	+	–	–
Стратегия повторения	+	+	+	+	+	+	+	+	+	–
Уровни сложности	+	+	+	+	–	–	–	+	+	–
Последовательные и лаконичные инструкции	+	+	+	+	+	+	+	+	–	+
Положительная обратная связь	+	–	–	–	+	–	+	+	–	+
Автоматическая регистрация ответов	+	+	+	+	+	+	+	+	+	+
Возможность вывода отчетов	+	+	+	–	–	–	–	–	–	–
Незагроможденный понятный интерфейс	+	+	+	+	+	+	+	–	+	+
Минимальные навыки работы с клавиатурой	+	+	+	+	+	+	+	+	+	+

Большинство существующих приложений имеют такие недостатки, как отсутствие возможности индивидуализации обучения, недостаточно эффективная система мотивации или являются устаревшими. Например, часть из них построена на основе технологии Flash [37], поддержка которой была прекращена 31 декабря 2020 года. Только три приложения соответствуют ФГОС.

Новизна настоящей работы заключается в возможности получения отчетов по каждому ученику или по группе в виде табличных и графических диаграмм прямо из приложения. В качестве источника данных для формирования отчетов используется база данных, в которой хранится вся аналитическая и статистическая информация по игроку, в том числе общее время выполнения упражнения, количество допущенных ошибок, средняя скорость реакции. Для рисования графиков на текстуре используются возможности движка Unity для работы с 2D графикой, в частности класс Texture2D. Также для более точного анализа реализована возможность отображения значений в конкретной точке графика при наведении курсора.

Используемые в работе стимульные материалы (буквы, слоги, слова) выбираются из набора пользовательских словарей, что обеспечивает вариативность содержания и сложности упражнений, повышает заинтересованность учащихся.

В процессе обучения с использованием разрабатываемого программного обеспечения не требуется наличие интернет-соединения или дополнительного оборудования, а также контроль со стороны взрослого.

1.2 Анализ основных технологических решений

В начале 2000-х годов многие разработчики игр (Blizzard Entertainment, Crytek и др.) были вынуждены создавать собственные движки [31] по причине высокой стоимости лицензирования существующих решений или отсутствия на рынке технологий, подходящих для реализации проекта. На сегодняшний день данная проблема неактуальна, поскольку существует огромное разнообразие игровых движков, и разработчик может выбрать оптимальный вариант по цене, спектру решаемых задач и другим критериям.

В рамках работы рассмотрим следующие наиболее популярные средства разработки:

- Unreal Engine [32],
- Unity [33],
- Godot Engine [34].

Игровой движок *Unreal Engine* был выпущен компанией Epic Games в 1998 году. Все эти годы он активно развивался, но основную популярность приобрел после перехода на условно-бесплатную модель распространения в 2015 году. Движок предъявляет высокие требования к аппаратному обеспечению компьютера, а его функционал избыточен для большинства проектов, также используемый язык программирования C++ имеет более низкий уровень абстракции по сравнению с другими языками высокого уровня и считается одним из сложнейших в изучении. В Unreal Engine 5 было добавлено множество улучшений [35], среди которых наиболее важными являются технологии Nanite и Lumen. Первая отвечает за визуализацию объектов в зависимости от их удаления от наблюдателя и предоставляет возможность использовать неоптимизированные модели без потери производительности и качества изображения. А вторая позволяет добавлять в игру динамическое освещение, которое адаптируется к изменениям обстановки или расположения источников света в режиме реального времени.

Межплатформенная среда разработки *Unity* от компании Unity Technologies на данный момент является самой популярной по количеству игр [36]. Несмотря на то, что по количеству возможностей и уровню технологий Unity уступает Unreal Engine 5, интуитивно понятный интерфейс, простой в освоении язык программирования C#, обилие обучающих материалов, активное сообщество – все это привлекает как начинающих программистов, так и профессионалов. Бесплатная версия движка может быть использована при

условии, что доход компании не превышает 100 000 долларов в год и имеет следующие ограничения:

- закрытый исходный код;
- стандартная экранная заставка;
- низкий приоритет заявки в техническую поддержку.

Godot Engine – открытый полностью бесплатный игровой движок без каких-либо ограничений по доходам и встроенных заставок с логотипом. Данный инструмент появился на рынке относительно недавно – в 2014 году. Основной отличительной особенностью *Godot Engine* является минимальный объем занимаемого пространства (около 70 Мбайт). Движок имеет незагроможденный интерфейс с минимальным количеством вкладок, встроенный редактор кода, а также собственный язык программирования GDscript. Из недостатков можно отметить небольшое количество обучающих материалов.

2 ПРОЕКТИРОВАНИЕ

2.1 Архитектура предлагаемого решения

Архитектура разрабатываемого программного обеспечения представлена на рисунке 5.

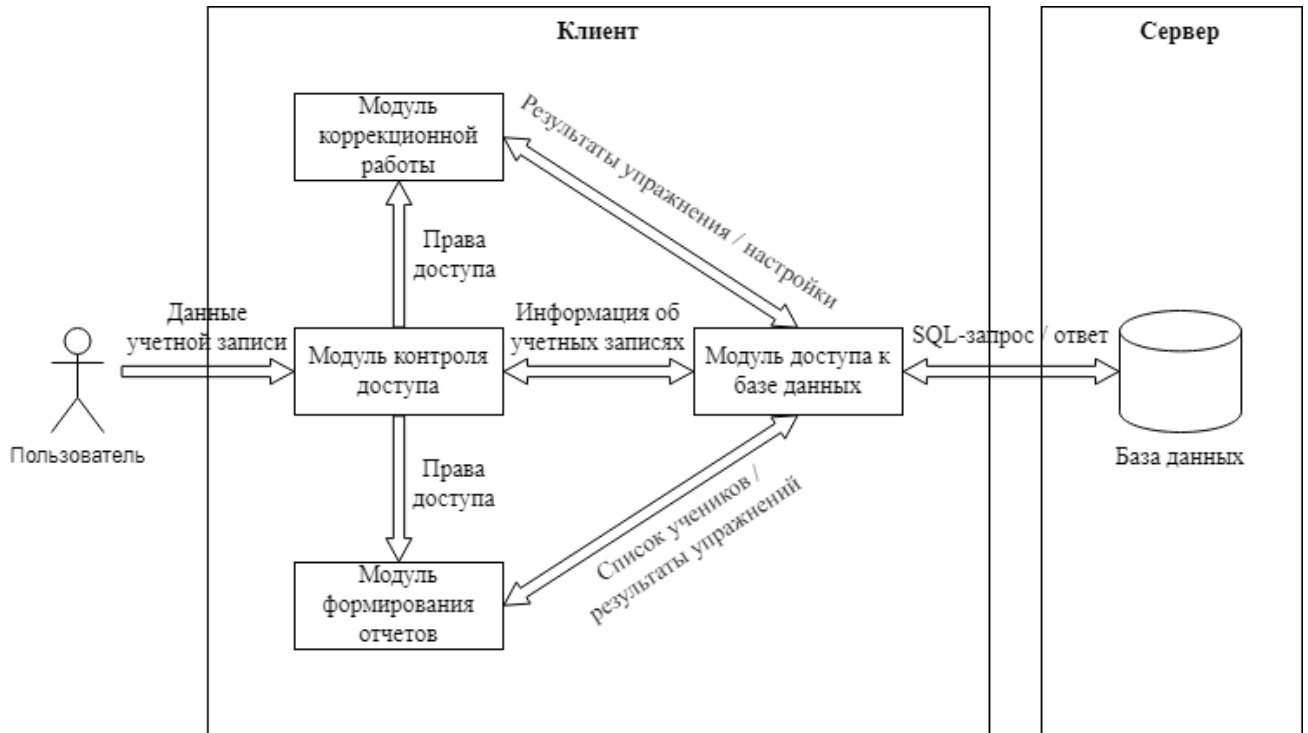


Рисунок 5 – Архитектура программного обеспечения

Можно выделить следующие функциональные модули:

- модуль контроля доступа,
- модуль коррекционной работы,
- модуль формирования отчетов,
- модуль доступа к базе данных.

Модуль контроля доступа состоит из трех взаимосвязанных, выполняемых последовательно процедур: идентификация, аутентификация, авторизация. На этапе идентификации по введенному имени пользователя из

хранилища данных считывается его идентификатор. Если пользователь с таким именем существует, то в зависимости от типа учетной записи проверяется на соответствие введенный пароль, т.е. осуществляется аутентификация, либо происходит переход к следующему шагу. После подтверждения личности пользователя на этапе авторизации ему выдаются соответствующие права доступа.

Модуль коррекционной работы содержит набор скриптов, отвечающих за логику и поведение каждой мини-игры, отправку результатов в базу данных, настройку уровня сложности и других параметров или ресурсов приложения.

Модуль формирования отчетов обрабатывает полученные из базы данных результаты упражнений, преобразовывает в формат таблицы или диаграммы для улучшения восприятия записей пользователем.

Модуль доступа к базе данных устанавливает соединение с хранилищем для считывания/редактирования существующих и внесения новых данных, таких как параметры учетных записей, настройки, результаты упражнений и др.

В приложении предусмотрены два типа пользователей («Ребенок», «Специалист») и три режима функционирования («Игра», «Настройки» и «Отчет»).

Ребенок – это обычный пользователь, имеющий полный доступ к режиму «Игра», в котором он выполняет упражнения для коррекции нарушений письменной речи. Возможности режима «Отчет» ограничены выводом информации только по данному пользователю. Для уровня доступа «Ребенок» предусмотрен беспарольный вход в приложение.

Специалист – пользователь с повышенными правами, ему доступны все функции режима «Отчет» и дополнительный режим «Настройки», в котором настраивается сложность упражнений, редактируются словари. Учетная запись специалиста защищена паролем.

2.2 Концепция, игровой процесс и сценарий

Игра разбита на четыре локации, три из которых являются отдельными мини-играми, при этом каждая из них направлена на решение определенной коррекционной задачи. Оставшаяся локация представляет собой карту, на которой находятся объекты, описывающие своим видом то или иное упражнение и осуществляющие переход на соответствующую локацию. Сюжет каждой мини-игры сводится к тому, что нужно помочь игровому персонажу решить некоторую задачу или отгадать его загадку.

Приложение содержит 3 мини-игры: «Шарики», «Фокусник» и «Конвейер».

В мини-игре «*Шарики*» на экране в несколько рядов расположены разноцветные воздушные шары, на каждом из которых изображена некоторая буква. Цель мини-игры – лопнуть все шары, а чтобы это сделать, необходимо найти в каждом ряду слово, выбранное случайным образом из словарей, и выделить его. Расположенные по бокам от загаданного слова буквы формируются путем перестановки одного символа в слове, относящегося к этому же словарю, чтобы еще больше запутать игрока.

«*Фокусник*» – мини-игра, в которой игроку необходимо определить спрятанный Джокером слог в слове, случайно выбранном из словаря, и выбрать его из предложенных вариантов.

В мини-игре «*Конвейер*» необходимо помочь роботу-сортировщику правильно распределить упаковки по конвейерным линиям. На экране конвейера появляется некоторая буква, а на ленте упаковка, подписанная случайным словом из словаря. Игрок должен провести анализ слова на наличие отображаемой на экране буквы и принять решение о запуске конвейера.

Управление во всех мини-играх осуществляется только с использованием компьютерной мыши и сводится к наведению курсора на игровой объект и щелчку по нему.

Каждое успешно выполненное упражнение оценивается по 100-балльной шкале. Количество баллов и другие результаты упражнения автоматически сохраняются в базе данных и доступны для просмотра из игры при нажатии на иконку дневника. Игрок может проходить мини-игры в любом порядке неограниченное количество раз, либо по согласованию со взрослым.

2.3 Описание данных

В базе данных хранятся настройки приложения, данные учетных записей (имя, хеш пароля, уровень доступа), ресурсы упражнений – коллекция понятных ребенку слов, объединенных в тематические группы, а также аналитическая и статистическая информация, необходимая для формирования отчетов: общее время выполнения упражнения, количество баллов, сложность упражнения, информация о допущенных ошибках (пользователь, ответ пользователя, правильный ответ).

Схема базы данных, спроектированная в CASE-системе DBDesigner [40], представлена на рисунке 6.

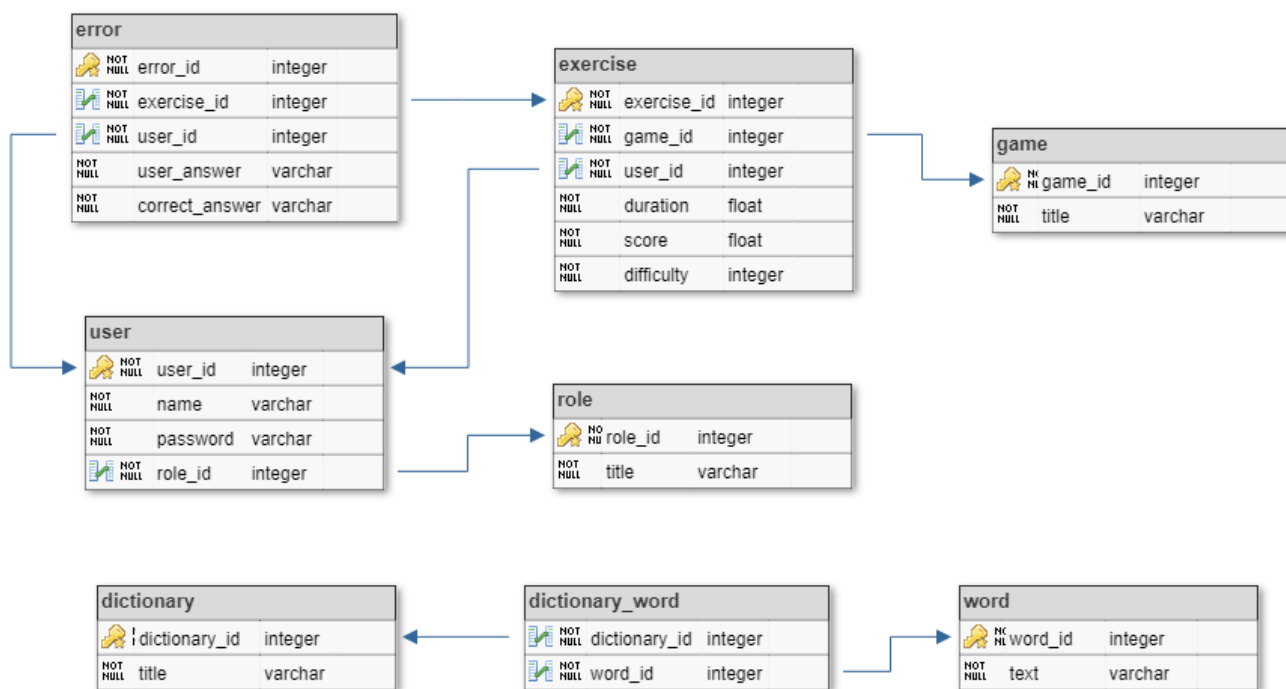


Рисунок 6 – Схема базы данных

Сущности базы данных и их описание приведены в таблице 2.

Таблица 2 – Сущности базы данных

Название	Описание
user	Данные учетной записи
role	Уровень доступа, определяет права пользователя
game	Мини-игра
exercise	Результаты упражнения с указанием пользователя и мини-игры
error	Информация об ошибках, допущенных пользователем в упражнении
word	Используемые в упражнениях слова
dictionary	Тематический словарь
dictionary_word	Соотношение слов и словарей

3 РЕАЛИЗАЦИЯ

3.1 Средства разработки

На основе проведенного анализа средств разработки для реализации настоящего проекта был выбран игровой движок Unity. Это связано с тем, что он имеет подробную официальную документацию на русском языке, а также огромное количество обучающих материалов, библиотек и ресурсов, позволяющих ускорить процесс разработки, из-за чего идеально подходит для работы в одиночку или в составе небольшой команды. Также согласно требованиям разрабатываемое приложение должно работать как на операционной системе Windows, так и на Linux, а разработанный на Unity проект может быть перенесен на различные платформы с минимальными изменениями.

Основным языком программирования в данном движке является C#. Помимо того, что он высокоуровневый и имеет низкий порог вхождения, многие элементы и приемы здесь уже реализованы и нужно всего лишь ими воспользоваться, это также позволяет ускорить разработку проекта.

Для написания, отладки и сборки кода используется интегрированная среда разработки Microsoft Visual Studio. Она имеет множество полезных инструментов, обеспечивающих высокий уровень безопасности кода за счет проверки, обнаружения ошибок и предупреждений, выдаваемых компилятором. Сборщик мусора .NET управляет выделением и освобождением памяти и позволяет устранить связанные с ней проблемы без участия разработчика. Также следует отметить высокую совместимость Microsoft Visual Studio с игровым движком Unity.

Для работы с базой данных используется система управления базами данных (СУБД) PostgreSQL [41]. Данная СУБД базируется на языке и стандартах SQL, но не реализует весь представленный в них функционал, за счет чего имеет более высокую производительность. Она поддерживает множество различных типов данных и даже позволяет создавать пользовательские типы данных и индексов. Особенностью PostgreSQL перед другими СУБД является отсутствие ограничения на максимальный размер базы данных.

3.2 Реализация модуля доступа к базе данных

Для работы с базой данных PostgreSQL был разработан класс DbManager, который использует библиотеку Npgsql. Класс содержит методы для открытия и закрытия соединения с базой данных и выполнения SQL-запросов на выборку или изменение данных. Адрес и порт сервера, информация о базе данных загружаются из конфигурационного файла dbsettings.json (листинг 1) в объект класса DbSettings при запуске приложения.

Листинг 1 – Структура файла dbsettings.json

```
{
  "Server": "localhost",
  "Port": "5432",
  "Database": "logocor",
  "UserID": "postgres",
  "Password": "postgres"
}
```

3.3 Реализация модуля контроля доступа

После запуска приложения на экране отображается меню входа (рисунок 7), где нужно выбрать пользователя из выпадающего списка, элементы которого можно отфильтровать по введенному значению. Если выбранный пользователь имеет уровень доступа «Ребенок», то для начала игры достаточно нажать на кнопку «Войти». При выборе пользователя с уровнем доступа «Специалист» также потребуется ввести пароль от учетной записи, а после успешного входа будет предложено выбрать один из трех режимов функционирования.

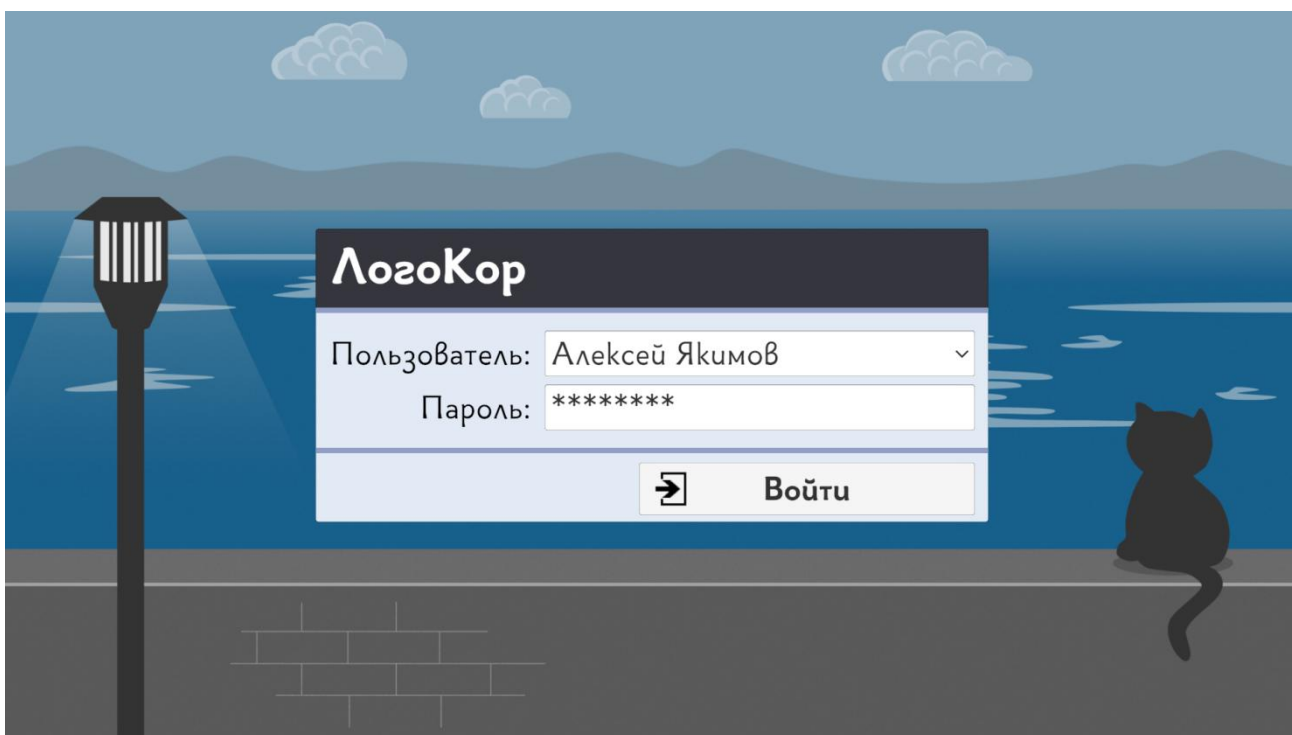


Рисунок 7 – Меню входа

Для обеспечения безопасности пароли в базе данных хранятся в зашифрованном виде с использованием алгоритма шифрования SHA512. Когда пользователь вводит свой пароль и нажимает на кнопку «Войти», вызывается метод ComputeHash класса NashConverter, и полученное хеш-значение сравнивается с хранящимся в базе данных. Если пароли совпадают, то

пользователь аутентифицируется и получает доступ к ресурсам, в противном случае выводится сообщение об ошибке.

3.4 Реализация модуля коррекционной работы

После перехода в режим игры на экране отображается игровая карта (рисунок 8), на которой размещены различные объекты, в том числе для запуска мини-игры.

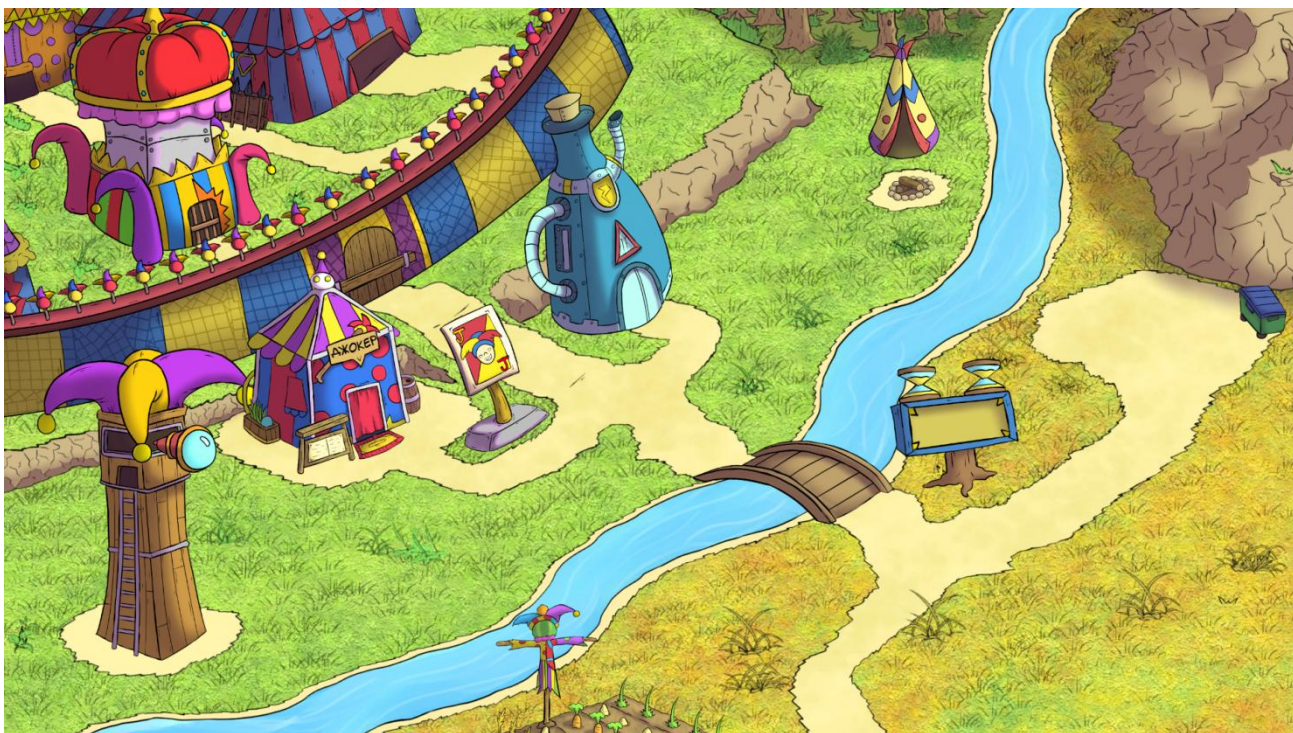


Рисунок 8 – Игровая карта

Для перехода к мини-игре игрок может взаимодействовать с одним из следующих объектов: наблюдательная вышка, шатер Джокера и конвейерный завод. При наведении курсора мыши на интерактивный объект (рисунок 9), у него воспроизводится анимация.



Рисунок 9 – Интерактивные объекты

В мини-игре «Шарики» (рисунок 10) после нахождения игроком слова в одном из рядов, проигрывается анимация, и все шарики на этом ряду исчезают. Когда шариков не останется, выполняется переход на следующий уровень.



Рисунок 10 – Мини-игра «Шарики»

В мини-игре «Фокусник» (рисунок 11) игрок выбирает один из предложенных вариантов, и если он правильный, то уровень генерируется заново, иначе предоставляется возможность повторного выбора.



Рисунок 11 – Мини-игра «Фокусник»

В мини-игре «Конвейер» (рисунок 12) при нажатии на кнопку с галочкой воспроизводится анимация конвейера, и упаковка начинает движение по ленте. Как только она достигает сканера, на экране загорается зеленый или красный цвет в зависимости от результата проверки. Когда упаковка полностью окажется вне зоны видимости камеры, конвейер останавливается и генерируется новый уровень.

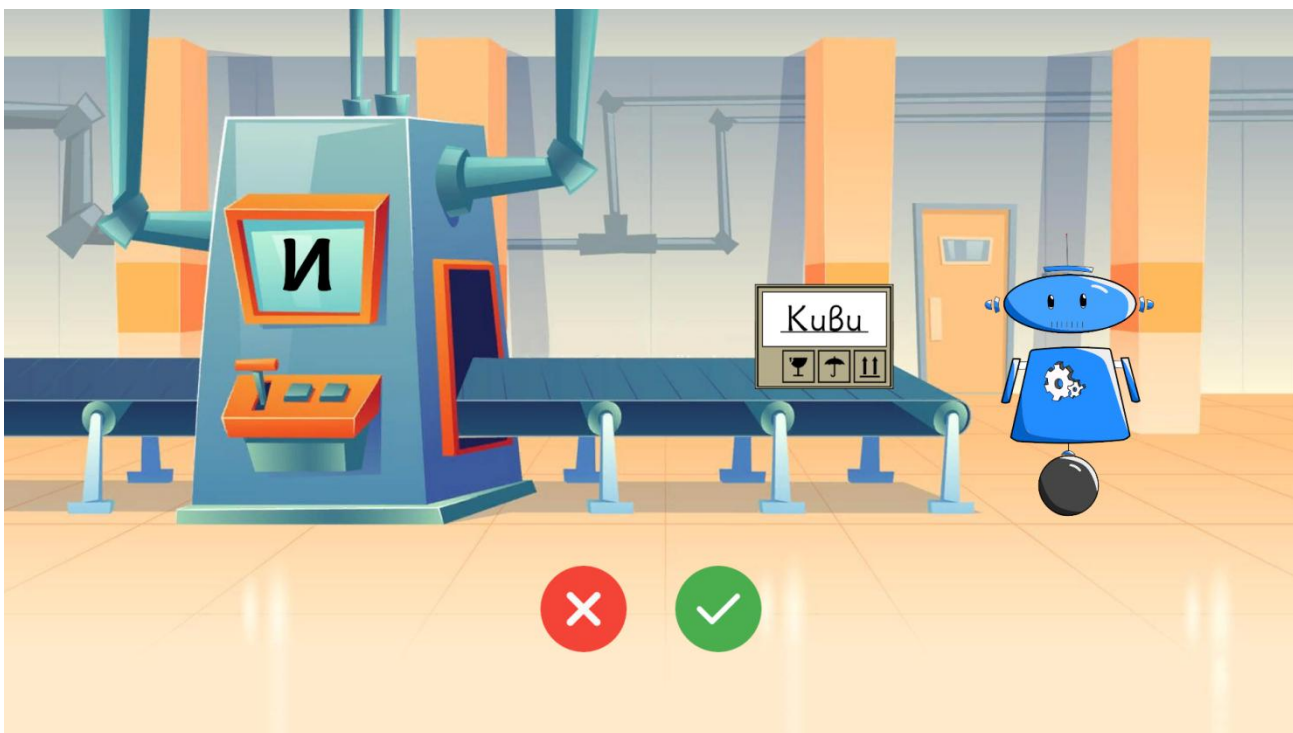


Рисунок 12 – Мини-игра «Конвейер»

Используемые в упражнениях стимульные материалы (буквы, слоги, слова) выбираются из набора пользовательских словарей. Все операции над словарями выполняются в меню настроек на вкладке «Словари» (рисунок 13).

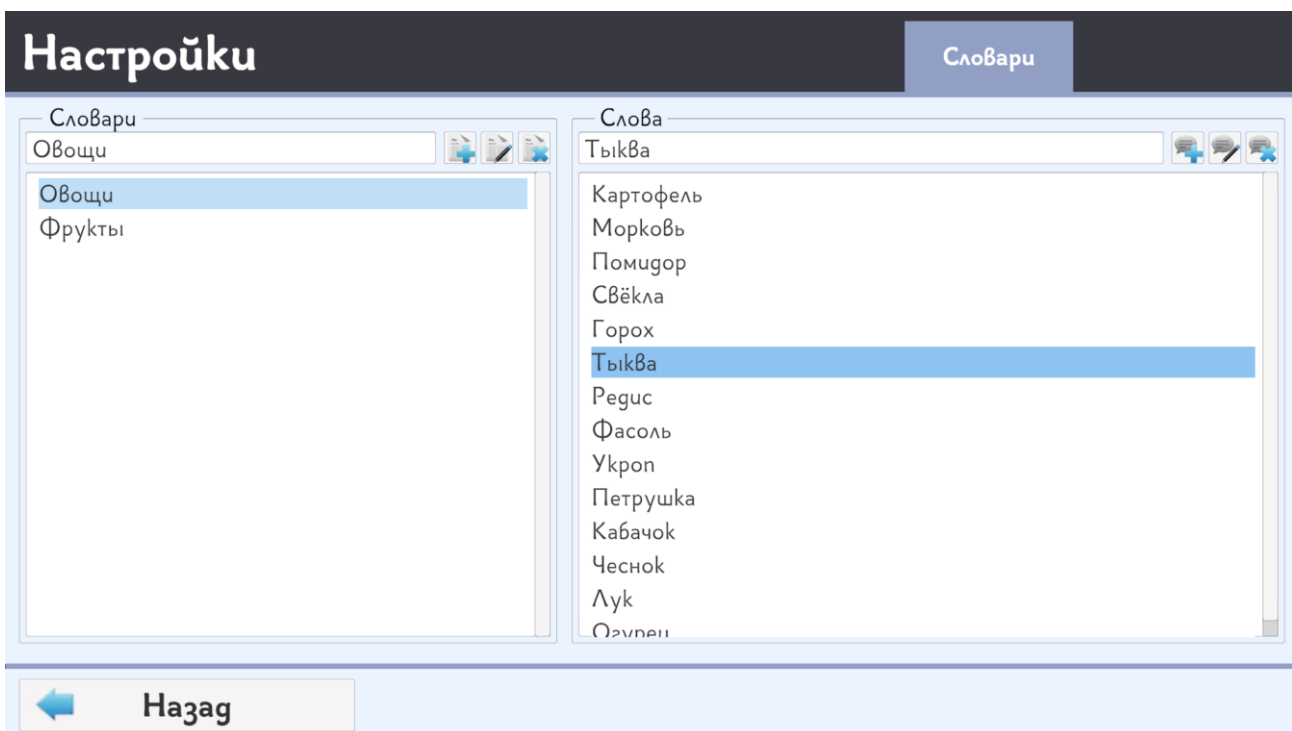


Рисунок 13 – Настройка словарей

Деление слов на слоги осуществляется по правилам русского языка школьной программы (традиционная школа) в автоматическом режиме с использованием метода `DivideIntoSyllables` класса `WordButton` (листинг 2).

Листинг 2 – Деление слов на слоги

```
private string DivideIntoSyllables(string word)
{
    char[] vowels = { 'a', 'e', 'ё', 'и', 'o', 'y', 'ы', 'э', 'ю', 'я' };
    word = word.ToLower();

    List<int> vowelIndices = new List<int>();
    for (int i = 0; i < word.Length; i++)
    {
        char letter = word[i];
        for (int j = 0; j < vowels.Length; j++)
        {
            if (letter == vowels[j])
            {
                vowelIndices.Add(i);
                break;
            }
        }
    }

    List<string> syllables = new List<string>();
    for (int i = vowelIndices.Count - 1; i > 0; i--)
    {
```

```

char letter = word[vowelIndices[i] - 1];
if (letter == 'б' || letter == 'Б')
{
    syllables.Add(word.Substring(vowelIndices[i]));
    word = word.Remove(vowelIndices[i]);
}
else
{
    int n = vowelIndices[i] - vowelIndices[i - 1] - 1;
    int index = vowelIndices[i - 1] + 1 + n / 2;
    letter = word[index];
    if (letter == 'б' || letter == 'Б') index++;
    syllables.Add(word.Substring(index));
    word = word.Remove(index);
}
}
syllables.Add(word);
syllables.Reverse();

StringBuilder result = new StringBuilder();
foreach (string syllable in syllables)
{
    if (DbManager.ExecuteQueryWithAnswer($"SELECT syllable_id FROM
public.syllable WHERE text = '{syllable}'") == null)
        DbManager.ExecuteQueryWithoutAnswer($"INSERT INTO
public.syllable (text) VALUES('{syllable}')");
    if (result.Length > 0) result.Append("-");
    result.Append(syllable);
}
return result.ToString();
}

```

Таким образом, нет никаких ограничений на ввод, кроме языка. Можно вводить слова в любом числе, роде, падеже и любой другой форме склонения слова.

3.5 Реализация модуля формирования отчетов

При переходе в режим «Отчет» открывается меню, в котором пользователь может выбрать отчет, ученика из выпадающего списка и определить временной промежуток для формирования отчета, после чего нажать на кнопку «Сформировать». На рисунке 14 представлен отчет по эффективности нагрузки.



Рисунок 14 – Отчет по эффективности нагрузки

Для рисования линий используется класс `UILineRenderer` из библиотеки `UIExtensions`. Исходные данные для построения графика загружаются из базы данных с применением указанных пользователем фильтров по дате и ученику.

При наведении курсора на точку графика рядом с ней отображается подсказка со значением этой точки.

3.6 Файловая структура

Файловая структура приложения представлена на рисунке 15.

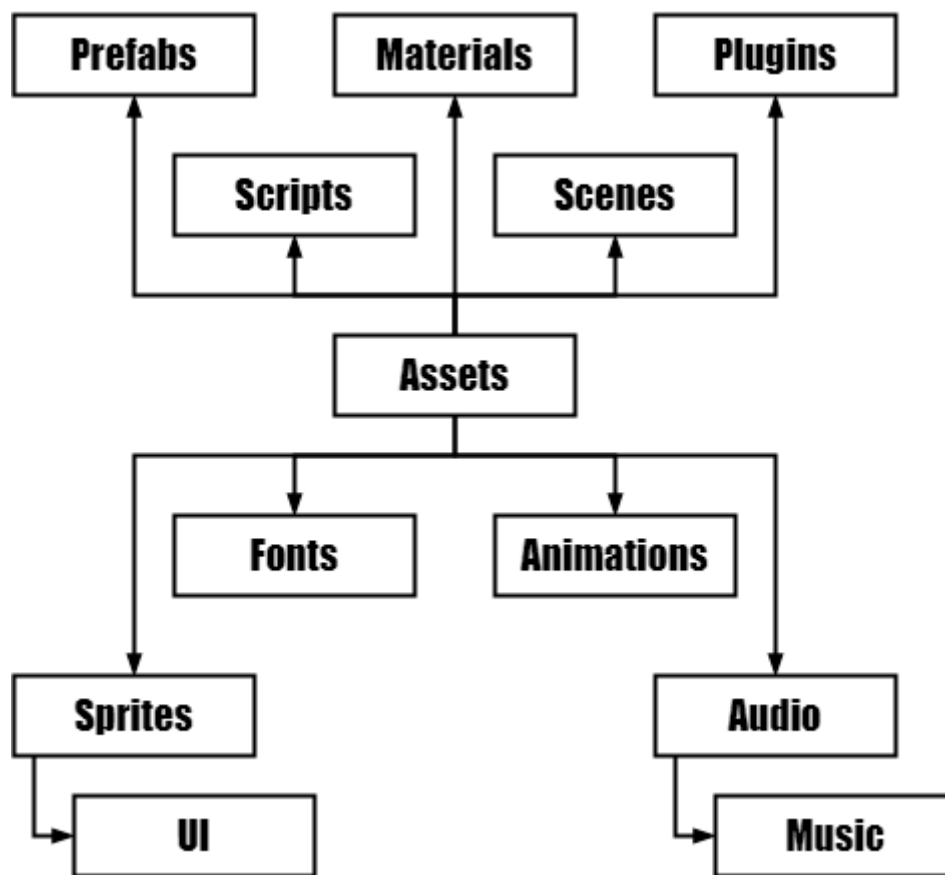


Рисунок 15 – Файловая структура

Каталоги и их описание приведены в таблице 3.

Таблица 3 – Каталоги

Название	Описание
Animations	Анимации игровых объектов
Audio	Музыка и звуки для игры
Fonts	Используемые в приложении шрифты
Materials	Материалы графических эффектов и шрифтов
Plugins	Динамически подключаемые библиотеки
Prefabs	Шаблоны игровых объектов и элементов UI
Scenes	Игровые сцены
Scripts	Скрипты с описанием классов и интерфейсов
Sprites	Изображения игровых объектов и элементов UI

Для представления текстовых и цифровых данных в проекте используется разработанный специально для людей с дислексией кириллический шрифт Adys [24].

Всего в проекте используется 22 скрипта, написанных на языке C#. Названия и описания скриптов представлены в таблице 4.

Таблица 4 – Скрипты

Название	Описание
GameManager	Выполняет загрузку и применение настроек приложения, содержит информацию о текущем пользователе
DbSettings	Содержит конфигурацию базы данных, методы для сохранения и загрузки настроек из файла, генерации настроек по умолчанию
DbManager	Обеспечивает взаимодействие с базой данных
ViewportHandler	Изменяет поле зрения камеры в зависимости от соотношения сторон экрана, чтобы охватить все ключевые объекты игры
LoginMenu	Выполняет функции контроля доступа: проверяет корректность введенных пользователем данных учетной записи
HashConverter	Преобразовывает входную строку в хеш-значение, используется для шифрования пароля
MainMenu	Обеспечивает взаимодействие пользователя с элементами интерфейса в главном меню
IScrollView	Интерфейс, описывающий контейнер с возможностью прокрутки содержимого
DictionaryScrollView	Отвечает за логику работы элемента интерфейса, содержащего список словарей, реализует IScrollView
WordScrollView	Отвечает за логику работы элемента интерфейса, содержащего список слов, реализует IScrollView
ScrollViewItem	Отвечает за логику работы элемента списка IScrollView
DictionaryButton	Содержит методы для кнопок добавления, изменения, удаления словаря
WordButton	Содержит методы для кнопок добавления, изменения, удаления слова из словаря
LineGraph	Отвечает за построение графического отчета
Tooltip	Отвечает за отображение всплывающей подсказки при наведении курсора на элемент

BalloonsGame	Отвечает за логику работы мини-игры «Шарики»
MagicianGame	Отвечает за логику работы мини-игры «Фокусник»
ConveyorGame	Отвечает за логику работы мини-игры «Конвейер»
SpeechBubble	Содержит набор фраз для похвалы, мотивации игрока и методы для отображения сообщения над головой персонажа в течение заданного периода времени
Selector	Управляет выделением объектов, реализующих интерфейс ISelectable
ISelectable	Интерфейс, описывающий объект, который может быть выделен экземпляром класса Selector
SelectableLetter	Текстовый объект с возможностью выделения, реализует интерфейс ISelectable

4 ТЕСТИРОВАНИЕ

Для обнаружения и устранения ошибок в работе приложения был спроектирован и разработан набор тестов по каждому функциональному модулю. Затем была проведена процедура тестирования и интерпретация полученных результатов в табличном формате.

В таблице 5 представлены результаты тестирования модуля контроля доступа.

Таблица 5 – Результаты тестирования модуля контроля доступа

№	Действие	Ожидаемый результат	Полученный результат
1	Выбор пользователя с уровнем доступа «Ребенок»	Поле пароля неактивно	Поле пароля неактивно
2	Выбор пользователя с уровнем доступа «Специалист»	Поле пароля активно	Поле пароля активно
3	Выбор пользователя с уровнем доступа «Ребенок» и нажатие на кнопку «Войти»	Запуск игры	Запуск игры
4	Выбор пользователя с уровнем доступа «Специалист», ввод правильного пароля и нажатие на кнопку «Войти»	Переход в меню специалиста	Переход в меню специалиста
5	Выбор пользователя с уровнем доступа «Специалист», ввод неправильного пароля и нажатие на кнопку «Войти»	Вывод сообщения об ошибке	Вывод сообщения об ошибке
6	Выбор пользователя с уровнем доступа «Специалист», поле пароля не заполнено	Кнопка «Войти» неактивна	Кнопка «Войти» неактивна
7	Ввод несуществующего имени пользователя	Кнопка «Войти» неактивна	Кнопка «Войти» неактивна
8	Ввод части имени	Фильтрация списка	Фильтрация списка

В таблице 6 представлены результаты тестирования модуля коррекционной работы.

Таблица 6 – Результаты тестирования модуля коррекционной работы

№	Действие	Ожидаемый результат	Полученный результат
1	Наведение указателя мыши на интерактивный объект на игровой карте	Воспроизведение анимации	Воспроизведение анимации
2	Нажатие на интерактивный объект на игровой карте	Запуск мини-игры	Запуск мини-игры
2	Выбор правильного ответа в мини-игре	Начисление баллов, похвала	Начисление баллов, похвала
3	Выбор неправильного ответа в мини-игре	Вычитание баллов, вывод мотивирующего сообщения	Вычитание баллов, вывод мотивирующего сообщения
4	Выполнение всех заданий в мини-игре	Вывод результатов по данному упражнению	Вывод результатов по данному упражнению
5	Нажатие на кнопку «Продолжить» после завершения мини-игры	Переход на игровую карту	Переход на игровую карту
5	Нажатие на кнопку «Повторить» после завершения мини-игры	Перезапуск мини-игры	Перезапуск мини-игры
6	Нажатие на иконку дневника	Открытие окна с результатами упражнений	Открытие окна с результатами упражнений
7	Нажатие на иконку двери	Выход из игры	Выход из игры
8	Нажатие на кнопку «Настройки» в меню специалиста	Открытие меню настроек	Открытие меню настроек
9	Нажатие на кнопку «Выход» в меню специалиста	Закрытие приложения	Закрытие приложения

В таблице 7 представлены результаты тестирования модуля формирования отчетов.

Таблица 7 – Результаты тестирования модуля формирования отчетов

№	Действие	Ожидаемый результат	Полученный результат
1	Нажатие на кнопку «Отчет» в окне с результатами игрока	Получение отчета по данному игроку	Получение отчета по данному игроку
2	Нажатие на кнопку «Отчет» в меню специалиста	Переход в меню отчетов	Переход в меню отчетов
3	Ввод текста в поле «Ученик»	Вывод списка учеников по введенному шаблону	Вывод списка учеников по введенному шаблону
4	Нажатие на кнопку «Добавить»	Добавление выбранного ученика в отчет	Добавление выбранного ученика в отчет
5	Нажатие на кнопку «Удалить»	Удаление выбранного ученика из отчета	Удаление выбранного ученика из отчета
6	Нажатие на кнопку «Сформировать»	Получение отчета по выбранным ученикам	Получение отчета по выбранным ученикам
7	Нажатие на кнопку «Меню»	Возврат в меню специалиста	Возврат в меню специалиста

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы было разработано приложение для обучения грамоте детей младшего школьного возраста с расстройствами чтения и письма.

Для достижения поставленной цели были решены следующие задачи:

- 1) проанализированы аналогичные решения;
- 2) спроектирована архитектура программного обеспечения;
- 3) выполнена программная реализация в соответствии с заданными требованиями;
- 4) протестировано разработанное программное обеспечение.

Разработанное приложение позволяет получать данные об эффективности занятий в удобном формате и обеспечивает гибкость и вариативность содержания упражнений за счет возможности создания пользовательских словарей. Проект является полностью работоспособным и готов к опытной эксплуатации.

Дальнейшее развитие проекта предусматривает:

- 1) возможность подключения к удаленному серверу;
- 2) адаптацию приложения под мобильные устройства;
- 3) возможность экспорта отчета в Excel, печати, отправки по электронной почте;
- 4) добавление новых мини-игр;
- 5) добавление новых отчетов.

В ходе реализации проекта были получены навыки работы с межплатформенной средой разработки Unity. Было создано 5 сцен, 14 шаблонов объектов, 9 анимаций, 76 спрайтов, 3 материала, 22 скрипта (около 1200 строк кода), 20 классов и 2 интерфейса. Также с использованием СУБД PostgreSQL

была разработана база данных, состоящая из 9 таблиц и 3 функций. Объем созданного программного средства составляет 90 Мбайт.

По теме данной работы был представлен доклад на 75-й студенческой научной конференции ЮУрГУ.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Пигаль, А. Ю. Развитие экспорта образовательных услуг в условиях COVID-19: проблемы и возможности / А. Ю. Пигаль // Молодежь, наука, практика : сборник научных трудов 61-й Международной практической конференции студентов, магистрантов и аспирантов. – Ярославль : Союз-Пресс, 2021. – Том 1. – С. 203-207.

2. Короткова, А. Дети. Медиапотребление. 2017 : ежегодный доклад / А. Короткова, Ю. Верлина, Т. Цай, С. Никитина, А. Сосновая, А. Хохлова. – Текст : электронный. – 2017. – URL: https://momri.org/wp-content/uploads/2018/08/momri.-deti_mediapotreblenie_2017.pdf (дата обращения: 13.06.2022).

3. Кареш, Г. Исследование развивающих игр и эффекта игрушек во время обучения детей дошкольного возраста / Г. Кареш, Е. Тезель // Сибирский педагогический журнал. – 2017. – № 2. – С. 130-134.

4. Каткова, А. Л. Использование компьютерных игр в процессе обучения информатике для стимулирования познавательного интереса студентов / А. Л. Каткова // Омский научный вестник. – 2006. – № 10 (48). – С. 182-185.

5. Горбунова, Л. И. Использование информационных технологий в процессе обучения / Л. И. Горбунова, Е. А. Субботина. – Текст : электронный // Молодой ученый : научный журнал. – 2013. – № 4 (51). – С. 544-547. – URL: <https://moluch.ru/archive/51/6685/> (дата обращения: 13.06.2022).

6. Гилязова, И. Р. IT-технологии для организации учебного процесса / И. Р. Гилязова // Инновации и современные технологии в системе образования : материалы III международной научно-практической конференции. – Пенза : Социосфера, 2013. – С. 289-292.

7. Анкудинова, С. А. Развитие интеллектуальных способностей детей дошкольного и младшего школьного возраста с помощью информационно-коммуникационных технологий / С. А. Анкудинова, И. А. Непочатых // Инновационная наука : международный научный журнал. – Уфа : Аэтерна, 2016. – С. 31-35.

8. Явич, М. П. Теоретические основы информатики. Концепции создания электронных обучающих средств и интеллектуальных компьютерных игр для детей / М. П. Явич // Современные информационные технологии и ИТ-образование : международный научный журнал. – 2012. – № 8. – С. 890-897.

9. Ярмухаметова, И. А. Развивающие компьютерные игры – катализатор интеллектуальных способностей детей старшего дошкольного возраста / И. А. Ярмухаметова // Международный журнал прикладных и фундаментальных исследований : научный журнал. – 2015. – № 12. – С. 176–179.

10. Роллингз, Э. Проектирование и архитектура игр / Э. Роллингз, Д. Моррис ; пер. с английского А. А. Чекаткова. – 2-е изд. – Москва : Вильямс, 2006. – 1040 с. : ил.

11. Интернет-Гномик : познавательное развитие дошкольника : [сайт]. – URL: <http://www.i-gnom.ru/> (дата обращения: 13.06.2022).

12. Лалаева, Р. И. Нарушения чтения и пути их коррекции у младших школьников : учебное пособие / Р. И. Лалаева ; науч. ред. Н. В. Серебрякова. – Санкт-Петербург : КАРО, 2019. – 256 с. : ил.

13. Филенкова, С. А. Нарушение письменной речи у младших школьников / С. А. Филенкова. – Текст : электронный // Молодой ученый : научный журнал. – 2019. – № 39 (277). – С. 272-274. – URL: <https://moluch.ru/archive/277/62555/> (дата обращения: 13.06.2022).

14. Польшина, М. А. Нарушение письма и чтения : учебно-методическое пособие / М. А. Польшина. – Оренбург : ОГПУ, 2021. – 100 с.

15. Фокина, О. А. Использование компьютерных технологий в коррекции дисграфии / О. А. Фокина // Коррекционная педагогика : теория и практика. – 2010. – № 4. – С. 81-86.

16. Игнатъев, А. Е. Игры и упражнения для преодоления дислексии и дисграфии : [видеозапись] / А. Е. Игнатъев. – 21.04.2020. – Изображение (движущееся) : электронное // Ассоциация родителей детей с дислексией : [канал на YouTube]. – URL: <https://www.youtube.com/watch?v=CLOTerbGePE> (дата обращения: 13.06.2022).

17. Корнев А. Н. Нарушения чтения и письма у детей : учебно-методическое пособие / А. Н. Корнев. – Санкт-Петербург : МиМ, 1997. – 286 с.

18. SLOGY : онлайн-платформа для обучения чтению и коррекции дислексии : [сайт]. – URL: <https://slogy.ru/> (дата обращения: 13.06.2022).

19. Дэльфа-142.1 : логопедические тренажеры : [сайт]. – URL: <https://www.delfam.ru/> (дата обращения: 13.06.2022).

20. Игры для Тигры : компьютерная технология коррекции общего недоразвития речи : [сайт]. – URL: <http://www.logopunkt.ru/tigra.htm> (дата обращения: 13.06.2022).

21. Мерсибо : развивающие игры для детей, логопедов, педагогов, психологов : [сайт]. – URL: <https://mersibo.ru/> (дата обращения: 13.06.2022).

22. Логозаврия : детские компьютерные игры : [сайт]. – URL: <https://www.logozavr.ru/> (дата обращения: 13.06.2022).

23. Играемся : детские развивающие игры онлайн : [сайт]. – URL: <https://www.igraemsa.ru/> (дата обращения: 13.06.2022).

24. Шрифт Adys : [сайт]. – URL: <http://www.adysfont.com/> (дата обращения: 13.06.2022).

25. ФГОС Начальное общее образование. – Текст : электронный. – 2009. – URL: <https://fgos.ru/fgos/fgos-noo/> (дата обращения: 13.06.2022).

26. International Dyslexia Association – Dyslexia Basics : fact sheet. – Text : electronic. – 2020. – URL: <https://dyslexiaida.org/dyslexia-basics/> (date of treatment: 6/13/2022).

27. Doolan, K. Student life in the EHEA during the COVID-19 pandemic : Europe-wide insights / K. Doolan, V. Barada, I. Burić, K. Krolo, Ž. Tonković. – Text : electronic. – 2020. – 14 p. – URL: http://www.ehea.info/Upload/BFUG_HR_UA_71_8_1_Survey_results.pdf (date of treatment: 6/13/2022).

28. Google for Games : global frontier report. – Text : electronic. – 2021. – URL: <https://games.withgoogle.com/reports/> (date of treatment: 6/13/2022).

29. Wijman, T. Global games market to generate \$175.8 billion in 2021; despite a slight decline, the market is on track to surpass \$200 Billion in 2023 / T. Wijman. – Text : electronic. – 2021. – URL: <https://newzoo.com/insights/articles/global-games-market-to-generate-175-8-billion-in-2021-despite-a-slight-decline-the-market-is-on-track-to-surpass-200-billion-in-2023/> (date of treatment: 6/13/2022).

30. Main, S. Focus on the journey, not the destination : digital games and students with disability / S. Main, J. O'Rourke, J. Morris, H. Dunjey. – Text : electronic // Issues in Educational Research : refereed academic research journal. – 2016. – Vol. 26 (2). – PP. 315-331. – URL: <http://www.iier.org.au/iier26/main.pdf> (date of treatment: 6/13/2022).

31. Custom game engines : a small study. – Text : electronic. – 2020. – URL: <https://gist.github.com/raysan5/909dc6cf33ed40223eb0dfe625c0de74/> (date of treatment: 6/13/2022).

32. Unreal Engine : the most powerful real-time 3D creation tool : [site]. – URL: <https://www.unrealengine.com/> (date of treatment: 6/13/2022).

33. Unity : real-time development platform | 3D, 2D VR & AR engine : [site]. – URL: <https://unity.com/> (date of treatment: 6/13/2022).
34. Godot Engine : free and open source 2D and 3D game engine : [site]. – URL: <https://godotengine.org/> (date of treatment: 6/13/2022).
35. Unreal Engine 5.0 Release Notes : Unreal Engine Documentation. – Text : electronic. – 2022. – URL: <https://docs.unrealengine.com/5.0/en-US/unreal-engine-5-0-release-notes/> (date of treatment: 6/13/2022).
36. Doucet, L. Game engines on Steam : the definitive breakdown / L. Doucet, A. Pecorella. – Text : electronic. – 2021. – URL: <https://www.gamedeveloper.com/business/game-engines-on-steam-the-definitive-breakdown/> (date of treatment: 6/13/2022).
37. Adobe Flash Player End of Life : general information page. – Text : electronic. – 2021. – URL: <https://www.adobe.com/products/flashplayer/end-of-life.html> (date of treatment: 6/13/2022).
38. Fast ForWord : computer-based reading program : [site]. – URL: <https://www.scilearn.com/> (date of treatment: 6/13/2022).
39. GraphoGame : academically researched learning app : [site]. – URL: <https://www.graphogame.com/index.html> (date of treatment: 6/13/2022).
40. DB Designer : online database schema design and modeling tool : [site]. – URL: <https://www.dbdesigner.net/> (date of treatment: 6/13/2022).
41. PostgreSQL : open source relational database : [site]. – URL: <https://www.postgresql.org/> (date of treatment: 6/13/2022).

ПРИЛОЖЕНИЕ А

Листинг А.1 – Исходный код класса GameManager

```
public class GameManager : MonoBehaviour {  
  
    public static GameManager Instance;  
  
    public int UserId { get; set; }  
  
    private void Awake() {  
        if (Instance == null) Instance = this;  
        else Destroy(gameObject);  
  
        System.Threading.Thread.CurrentThread.CurrentCulture = new  
System.Globalization.CultureInfo("en-US");  
        DbSettings.LoadSettings("dbsettings.json");  
    }  
}
```

Листинг А.2 – Исходный код класса DbSettings

```
[JsonObject(MemberSerialization.OptIn)]  
public class DbSettings {  
    private static DbSettings _instance;  
    public static DbSettings Instance { get => _instance ??= new DbSettings();  
private set => _instance = value; }  
    [JsonProperty] public string Server { get; private set; }  
    [JsonProperty] public string Port { get; private set; }  
    [JsonProperty] public string Database { get; private set; }  
    [JsonProperty] public string UserID { get; private set; }  
    [JsonProperty] public string Password { get; private set; }  
  
    public static void SaveSettings(string fileName) {  
        using (StreamWriter sw = new StreamWriter(fileName)) {  
            string json = JsonConvert.SerializeObject(Instance,  
Formatting.Indented);  
            sw.Write(json);  
        }  
    }  
  
    public static void LoadSettings(string fileName) {  
        if (File.Exists(fileName)) {  
            using (StreamReader sr = new StreamReader(fileName)) {  
                string json = sr.ReadToEnd();  
                Instance = JsonConvert.DeserializeObject<DbSettings>(json);  
            }  
        }  
        else {  
            WriteDefaultData();  
            SaveSettings(fileName);  
        }  
    }  
  
    private static void WriteDefaultData() {  
        Instance.Server = "localhost";  
    }  
}
```

```

Instance.Port = "5432";
Instance.Database = "logocor";
Instance.UserID = "postgres";
Instance.Password = "postgres";
    }
}

```

Листинг А.3 – Исходный код класса DbManager

```

public class DbManager {
    private static readonly string _connectionString =
        $"Port = {DbSettings.Instance.Port};" +
        $"Server = {DbSettings.Instance.Server};" +
        $"Database = {DbSettings.Instance.Database};" +
        $"User ID = {DbSettings.Instance.UserID};" +
        $"Password = {DbSettings.Instance.Password};";

    private static NpgsqlConnection _connection;
    private static NpgsqlCommand _command;

    private static void OpenConnection() {
        _connection = new NpgsqlConnection(_connectionString);
        _connection.Open();
        _command = _connection.CreateCommand();
    }

    public static void CloseConnection() {
        _connection.Close();
        _command.Dispose();
    }

    public static void ExecuteQueryWithoutAnswer(string query) {
        OpenConnection();
        _command.CommandText = query;
        _command.ExecuteNonQuery();
        CloseConnection();
    }

    public static string ExecuteQueryWithAnswer(string query) {
        OpenConnection();
        _command.CommandText = query;
        var answer = _command.ExecuteScalar();
        CloseConnection();
        return answer?.ToString();
    }

    public static DataTable GetTable(string query) {
        OpenConnection();
        NpgsqlDataAdapter adapter = new NpgsqlDataAdapter(query, _connection);
        DataSet DS = new DataSet();
        adapter.Fill(DS);
        adapter.Dispose();
        CloseConnection();
        return DS.Tables[0];
    }
}

```

Листинг А.4 – Исходный код класса ViewportHandler

```

public class ViewportHandler : MonoBehaviour {
    private readonly float _targetAspect = 16f / 9f;

    [SerializeField] private float _cameraSize = 5;
    private Camera _camera;

    private void Awake() {
        _camera = GetComponent<Camera>();
    }

    private void OnGUI() {
        _camera.orthographicSize = _cameraSize * _targetAspect / _camera.aspect;
    }
}

```

Листинг А.5 – Исходный код класса LoginMenu

```

public class LoginMenu : MonoBehaviour {
    [SerializeField] private GameObject _mainMenu;
    [SerializeField] private TMP_InputField _inputLogin;
    [SerializeField] private TMP_InputField _inputPassword;
    [SerializeField] private Button _loginButton;
    [SerializeField] private TMP_Dropdown _dropdown;

    private List<TMP_Dropdown.OptionData> _items = new
List<TMP_Dropdown.OptionData>();

    void Start() {
        _dropdown.onValueChanged.AddListener(_ => DropdownItemSelected());
        _inputLogin.onValueChanged.AddListener(_ => ValueChangedHandler());
        _inputPassword.onValueChanged.AddListener(_ => UpdateButtonState());
        Refresh();
    }

    private void ValueChangedHandler()
    {
        if (!_items.Any(i => i.text == _inputLogin.text && i.text != "[Не
выбрано]")) {
            _dropdown.value = 0;
            Refresh(_inputLogin.text);
        }

        bool isSpecialist = DbManager.ExecuteQueryWithAnswer($"SELECT role_id
FROM public.user WHERE name = '{_inputLogin.text}';") == "1";
        _inputPassword.interactable = isSpecialist;

        UpdateButtonState();
    }

    public void Refresh(string filter = null) {
        _dropdown.options.Clear();
        _items = new List<TMP_Dropdown.OptionData>();
        if (_items.Count <= 1) {

```

```

        DataTable dt = DbManager.GetTable("SELECT name AS \"Name\" FROM
public.user;");
        string[] users = dt.AsEnumerable().Select(r =>
r.Field<string>("Name")).ToArray();
        _items.Add(new TMP_Dropdown.OptionData("[Не выбрано]"));
        foreach (string user in users)
            _items.Add(new TMP_Dropdown.OptionData(user));
    }

    if (filter == null)
        _dropdown.options = _items;
    else {
        _items = _items.Where(i =>
i.text.ToLower().Contains(filter.ToLower()) || i.text == "[Не
выбрано]").ToList();
        _dropdown.options = _items;
    }
}

private void DropdownItemSelected() {
    if (_dropdown.value != 0 || _items.Any(i => i.text == _inputLogin.text))
        _inputLogin.text = _dropdown.options[_dropdown.value].text;

    Refresh();
}

private void UpdateButtonState() {
    _loginButton.interactable = _items.Any(i => i.text == _inputLogin.text
&& i.text != "[Не выбрано]") && (!_inputPassword.interactable ||
_inputPassword.text.Length > 0);
}

public void OnClick() {
    if (_inputPassword.interactable)
    {
        string storedPassword = DbManager.ExecuteQueryWithAnswer($"SELECT
password FROM public.user WHERE name = '{_inputLogin.text}';");
        string inputPassword =
HashConverter.ComputeHash(_inputPassword.text, HashAlgorithmName.SHA512);
        if (storedPassword != inputPassword) {
            Debug.LogError("Введен неверный пароль!");
            return;
        }

        _mainMenu.SetActive(true);
        gameObject.SetActive(false);
    }
    else SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +
1);

    string userId = DbManager.ExecuteQueryWithAnswer($"SELECT user_id FROM
public.user WHERE name = '{_inputLogin.text}';");
    GameManager.Instance.UserId = Convert.ToInt32(userId);
    Debug.Log(GameManager.Instance.UserId);
}
}

```

Листинг А.6 – Исходный код класса HashConverter

```

public static class HashConverter {
    public static string ComputeHash(string data, HashAlgorithmName
hashAlgorithm) {
        byte[] dataInBytes = Encoding.Default.GetBytes(data);
        byte[] hash =
HashAlgorithm.Create(hashAlgorithm.ToString()).ComputeHash(dataInBytes);
        return GetHexString(hash);
    }

    private static string GetHexString(byte[] hash) =>
BitConverter.ToString(hash).Replace("-", string.Empty).ToLower();
}

```

Листинг А.7 – Исходный код класса MainMenu

```

public class MainMenu : MonoBehaviour {
    public void PlayGame() {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }

    public void QuitGame() {
        Application.Quit();
    }
}

```

Листинг А.8 – Исходный код интерфейса IScrollView

```

public interface IScrollView {
    GameObject ItemPrefab { get; }
    Transform Content { get; }
    Scrollbar Scrollbar { get; }
    ScrollViewItem SelectedItem { get; set; }

    void Refresh();
    bool Contains(string itemTitle);
}

```

Листинг А.9 – Исходный код класса DictionaryScrollView

```

public class DictionaryScrollView : MonoBehaviour, IScrollView {
    [SerializeField] private WordScrollView _wordScrollView;
    [SerializeField] private GameObject _itemPrefab;
    [SerializeField] private Transform _content;
    [SerializeField] private Scrollbar _scrollbar;
    private ScrollViewItem _selectedItem;
    private List<ScrollViewItem> _items;

    public event Action SelectedItemChanged;

    public GameObject ItemPrefab => _itemPrefab;
    public Transform Content => _content;
    public Scrollbar Scrollbar => _scrollbar;
}

```

```

    public ScrollViewItem SelectedItem { get => _selectedItem; set {
        _selectedItem = value; SelectedItemChanged.Invoke(); } }

    public void Start() {
        SelectedItemChanged += _wordScrollView.Refresh;
        Refresh();
    }

    public void Refresh() {
        foreach (Transform item in Content) {
            Destroy(item.gameObject);
        }

        var dt = DbManager.GetTable("SELECT title AS \"Title\" FROM
public.dictionary;");
        string[] models = dt.AsEnumerable().Select(r =>
r.Field<string>("Title")).Distinct().ToArray();

        foreach (var model in models) {
            var viewGameObject = Instantiate(ItemPrefab, Content);
            viewGameObject.GetComponent<ScrollViewItem>().Init(this, model);
        }
    }

    public bool Contains(string itemTitle)
    {
        return _items.Any(i => i.Title == itemTitle);
    }

    public void ScrollTo(float value)
    {
        StartCoroutine(ScrollToCoroutine(value));
    }

    IEnumerator ScrollToCoroutine(float value)
    {
        yield return null; // Waiting one frame
        Scrollbar.value = value;
    }
}

```

Листинг А.10 – Исходный код класса WordScrollView

```

public class WordScrollView : MonoBehaviour, IScrollView {
    [SerializeField] private DictionaryScrollView _dictionaryScrollView;
    [SerializeField] private GameObject _itemPrefab;
    [SerializeField] private Transform _content;
    [SerializeField] private Scrollbar _scrollbar;
    [SerializeField] private TMP_InputField _inputField;
    private ScrollViewItem _selectedItem;
    private List<ScrollViewItem> _items;

    public GameObject ItemPrefab => _itemPrefab;
    public Transform Content => _content;
    public Scrollbar Scrollbar => _scrollbar;
    public ScrollViewItem SelectedItem { get => _selectedItem; set {
        _selectedItem = value; _inputField.text = _selectedItem.Title; } }
}

```

```

public bool Contains(string itemTitle)
{
    return _items.Any(i => i.Title == itemTitle);
}

public void Refresh() {
    foreach (Transform item in Content) {
        Destroy(item.gameObject);
    }
    _items = new List<ScrollViewItem>();

    var dictionaryTitle = _dictionaryScrollView.SelectedItem.ToString();

    var dt = DbManager.GetTable("SELECT text AS \"Text\" FROM
public.dictionary_word " +
                                "JOIN public.dictionary ON
dictionary_word.dictionary_id = dictionary.dictionary_id " +
                                "JOIN public.word ON
dictionary_word.word_id = word.word_id " +
                                $"WHERE title = '{dictionaryTitle}'");
    string[] words = dt.AsEnumerable().Select(r =>
r.Field<string>("Text")).Distinct().ToArray();

    foreach (var word in words) {
        var scrollViewItem = Instantiate(ItemPrefab,
Content).GetComponent<ScrollViewItem>();
        scrollViewItem.Init(this, word);
        _items.Add(scrollViewItem);
    }
    ScrollTo(1f);
}

public void ScrollTo(float value) {
    StartCoroutine(ScrollToCoroutine(value));
}

IEnumerator ScrollToCoroutine(float value) {
    yield return null; // Waiting one frame
    Scrollbar.value = value;
}
}

```

Листинг А.11 – Исходный код класса ScrollViewItem

```

public class ScrollViewItem : MonoBehaviour {
    private TextMeshProUGUI _textMeshProUGUI;
    private IScrollView _viewModel;

    public string Title { get; private set; }

    public void Init(IScrollView viewModel, string title) {
        _viewModel = viewModel;
        Title = title;
        _textMeshProUGUI = GetComponentInChildren<TextMeshProUGUI>();
        _textMeshProUGUI.text = title;
    }
}

```



```

public void OnClick() {
    if (_viewModel.SelectedItem != null)
        _viewModel.SelectedItem.GetComponent<Graphic>().color = Color.white;
    _viewModel.SelectedItem.GetComponent<Graphic>().color = Color.white;
    _viewModel.SelectedItem.GetComponent<Graphic>().color =
        _viewModel.SelectedItem.GetComponent<Button>().colors.selectedColor;
    }

public override string ToString() {
    return _textMeshProUGUI.text;
}
}

```

Листинг А.12 – Исходный код класса DictionaryButton

```

public class DictionaryButton : MonoBehaviour {
    [SerializeField] private DictionaryScrollView _dictionaryScrollView;
    [SerializeField] private WordScrollView _wordScrollView;
    [SerializeField] private TMP_InputField _inputField;

    public void AddDictionary() {
        if (!_dictionaryScrollView.Contains(_inputField.text)) {
            DbManager.ExecuteQueryWithoutAnswer($"INSERT INTO public.dictionary
(title) VALUES ('{_inputField.text}')");

            _dictionaryScrollView.Refresh();
            _dictionaryScrollView.ScrollTo(0f);
            _wordScrollView.Refresh();
            _wordScrollView.ScrollTo(0f);
        }
    }

    public void DeleteDictionary() {
        if (_dictionaryScrollView.SelectedItem != null) {
            DbManager.ExecuteQueryWithoutAnswer("DELETE FROM
public.dictionary_word WHERE " +
dictionary_id FROM public.dictionary WHERE title = '{_inputField.text}'); " +
            $"dictionary_id = (SELECT
dictionary_id FROM public.dictionary WHERE title = '{_inputField.text}'); " +
            $"DELETE FROM public.dictionary
WHERE title = '{_inputField.text}');");

            var scrollbarValue = _dictionaryScrollView.Scrollbar.value;
            _dictionaryScrollView.Refresh();
            _dictionaryScrollView.ScrollTo(scrollbarValue);
            _wordScrollView.Refresh();
            _wordScrollView.ScrollTo(0f);
        }
    }

    public void UpdateDictionary() {
        if (!_wordScrollView.Contains(_inputField.text) &&
        _dictionaryScrollView.SelectedItem != null) {
            DbManager.ExecuteQueryWithoutAnswer($"UPDATE public.dictionary SET
title = '{_inputField.text}' " +
            $"WHERE title =
'{_dictionaryScrollView.SelectedItem.Title}');");
        }
    }
}

```

```

        var scrollbarValue = _dictionaryScrollView.Scrollbar.value;
        _dictionaryScrollView.Refresh();
        _dictionaryScrollView.ScrollTo(scrollbarValue);
    }
}

```

Листинг А.13 – Исходный код класса WordButton

```

public class WordButton : MonoBehaviour
{
    [SerializeField] private DictionaryScrollView _dictionaryScrollView;
    [SerializeField] private WordScrollView _wordScrollView;
    [SerializeField] private TMP_InputField _inputField;

    public void AddWord() {
        string dictionaryTitle = _dictionaryScrollView.SelectedItem.ToString();

        if (!_wordScrollView.Contains(_inputField.text)) {
            bool isExist = DbManager.ExecuteQueryWithAnswer($"SELECT text FROM
public.word WHERE text = '{_inputField.text}';") != null;
            if (!isExist) {
                DbManager.ExecuteQueryWithoutAnswer($"INSERT INTO public.word
(text, syllables) VALUES('{_inputField.text}',
'{DivideIntoSyllables(_inputField.text)}');");
            }

            DbManager.ExecuteQueryWithoutAnswer("INSERT INTO
public.dictionary_word (dictionary_id, word_id) VALUES(" +
public.dictionary WHERE title = '{dictionaryTitle}'), " +
public.word WHERE text = '{_inputField.text}'));");
            _wordScrollView.Refresh();
            _wordScrollView.ScrollTo(0f);
        }
    }

    public void DeleteWord()
    {
        string dictionaryTitle = _dictionaryScrollView.SelectedItem.ToString();

        if (_wordScrollView.SelectedItem != null) {
            DbManager.ExecuteQueryWithoutAnswer("DELETE FROM
public.dictionary_word WHERE " +
dictionary_id = (SELECT
dictionary_id FROM public.dictionary WHERE title = '{dictionaryTitle}') AND " +
word_id = (SELECT word_id FROM
public.word WHERE text = '{_wordScrollView.SelectedItem.Title}');" +
"DELETE FROM public.word WHERE
text = '{_wordScrollView.SelectedItem.Title}';");
            var scrollbarValue = _wordScrollView.Scrollbar.value;
            _wordScrollView.Refresh();
            _wordScrollView.ScrollTo(scrollbarValue);
        }
    }
}

```

```

public void UpdateWord()
{
    string dictionaryTitle = _dictionaryScrollView.SelectedItem.ToString();

    if (!_wordScrollView.Contains(_inputField.text) &&
        _wordScrollView.SelectedItem != null)
    {
        bool isExist = DbManager.ExecuteQueryWithAnswer($"SELECT text FROM
public.word WHERE text = '{_inputField.text}';") != null;
        if (!isExist)
        {
            DbManager.ExecuteQueryWithoutAnswer($"INSERT INTO public.word
(text, syllables) VALUES('{_inputField.text}',
'{DivideIntoSyllables(_inputField.text)}');");
        }

        DbManager.ExecuteQueryWithoutAnswer("DELETE FROM
public.dictionary_word " +
                                           $"WHERE word_id = (SELECT
word_id FROM public.word WHERE text = '{_wordScrollView.SelectedItem.Title}') " +
                                           $"AND dictionary_id = (SELECT
dictionary_id FROM public.dictionary WHERE title = '{dictionaryTitle}'); " +
                                           "INSERT INTO
public.dictionary_word (dictionary_id, word_id) VALUES(" +
                                           $"(SELECT dictionary_id FROM
public.dictionary WHERE title = '{dictionaryTitle}'), " +
                                           $"(SELECT word_id FROM
public.word WHERE text = '{_inputField.text}'))");
        _wordScrollView.Refresh();
        _wordScrollView.ScrollTo(0f);
    }
}

private string DivideIntoSyllables(string word) {
    char[] vowels = { 'a', 'e', 'ë', 'и', 'o', 'y', 'ы', 'э', 'ю', 'я' };
    word = word.ToLower();

    List<int> vowelIndices = new List<int>();
    for (int i = 0; i < word.Length; i++) {
        char letter = word[i];
        for (int j = 0; j < vowels.Length; j++) {
            if (letter == vowels[j]) {
                vowelIndices.Add(i);
                break;
            }
        }
    }

    List<string> syllables = new List<string>();
    for (int i = vowelIndices.Count - 1; i > 0; i--) {
        char letter = word[vowelIndices[i] - 1];
        if (letter == 'ь' || letter == 'Ь') {
            syllables.Add(word.Substring(vowelIndices[i]));
            word = word.Remove(vowelIndices[i]);
        }
        else {
            int n = vowelIndices[i] - vowelIndices[i - 1] - 1;

```

```

        int index = vowelIndices[i - 1] + 1 + n / 2;
        letter = word[index];
        if (letter == 'ь' || letter == 'Ь') index++;
        syllables.Add(word.Substring(index));
        word = word.Remove(index);
    }
}
syllables.Add(word);
syllables.Reverse();

StringBuilder result = new StringBuilder();
foreach (string syllable in syllables) {
    if (DbManager.ExecuteQueryWithAnswer($"SELECT syllable_id FROM
public.syllable WHERE text = '{syllable}'") == null)
        DbManager.ExecuteQueryWithoutAnswer($"INSERT INTO
public.syllable (text) VALUES('{syllable}')");
    if (result.Length > 0) result.Append("-");
    result.Append(syllable);
}
return result.ToString();
}
}
}

```

Листинг А.14 – Исходный код класса LineGraph

```

public class LineGraph : MonoBehaviour {
    [SerializeField] private GameObject _xDivPrefab;
    [SerializeField] private GameObject _yDivPrefab;
    [SerializeField] private GameObject _pointPrefab;
    [SerializeField] private float _xStep = 0.1f, _yStep = 0.1f;

    private GameObject[] _points;
    private UILineRenderer _lineRenderer;
    private Vector2[] _tops;
    private float _graphWidth, _graphHeight;
    private int _startDate, _endDate;

    private string[] dates;
    private double[] grades;

    public void Start() {
        _lineRenderer = GetComponentInChildren<UILineRenderer>();
        _graphWidth = GetComponent<RectTransform>().rect.width;
        _graphHeight = GetComponent<RectTransform>().rect.height;

        int user_id = 8;
        string start_date = "2022-05-21";
        string end_date = "2022-05-31";
        DataTable dt = DbManager.GetTable($"SELECT date AS \"Date\", grade AS
\"Grade\" FROM public.daily_performance({user_id}, '{start_date}',
'{end_date}');");
        dates = dt.AsEnumerable().Select(r =>
r.Field<string>("Date")).ToArray();
        grades = dt.AsEnumerable().Select(r =>
r.Field<double>("Grade")).ToArray();

        _xStep = 1.0f / (dates.Length - 1);
    }
}

```

```

    Init();

    DrawGraph(grades);
    LocateGraph(grades);
}

public void Init() {
    Vector2 pos;
    int i = 0;
    do {
        var xDiv = Instantiate(_xDivPrefab, Vector2.zero,
Quaternion.identity);
        xDiv.transform.parent = transform.Find("X");
        pos = new Vector2(_graphWidth * i * _xStep - _graphWidth / 2, 0);
        xDiv.transform.localPosition = pos;
        xDiv.GetComponentInChildren<TMPPro.TextMeshProUGUI>().text =
dates[i];
        i++;
    } while ((pos.x < _graphWidth / 2) && (i < dates.Length));

    i = 0;
    do
    {
        var yDiv = Instantiate(_yDivPrefab, Vector2.zero,
Quaternion.identity);
        yDiv.transform.parent = transform.Find("Y");
        pos = new Vector2(0, _graphHeight * i * _yStep - _graphHeight / 2);
        yDiv.transform.localPosition = pos;
        yDiv.GetComponentInChildren<TMPPro.TextMeshProUGUI>().text = (i *
10).ToString();
        i++;
    } while (pos.y < _graphHeight / 2);
}

public void DrawGraph(double[] points) {
    _points = new GameObject[points.Length];

    for (int i = 0; i < _points.Length; i++) {
        _points[i] = Instantiate(_pointPrefab, Vector3.zero,
transform.rotation, transform);
        _points[i].GetComponent<Tooltip>().Init($"{points[i]}%");
    }
}

public void LocateGraph(double[] points)
{
    _tops = new Vector2[_points.Length];
    for (int i = 0; i < points.Length; i++) {
        _tops[i] = new Vector2((_graphWidth / (points.Length - 1) * i) -
_graphWidth / 2, (float)((_graphHeight / 100 * points[i]) - _graphHeight / 2));
    }
    for (int i = 0; i < points.Length; i++) {
        _points[i].transform.localPosition = _tops[i];
    }

    _lineRenderer.Points = _tops;
}
}

```

Листинг А.15 – Исходный код класса Tooltip

```

public class Tooltip : MonoBehaviour
{
    [SerializeField] private GameObject _tooltip;

    private TextMeshProUGUI _tooltipText;

    public void Init(string text) {
        _tooltipText.text = text;
    }

    private void Awake() {
        _tooltipText = _tooltip.GetComponentInChildren<TextMeshProUGUI>();
    }

    public void OnMouseEnter() {
        if (!_tooltip.activeSelf) _tooltip.SetActive(true);
    }

    public void OnMouseExit() {
        if (_tooltip.activeSelf) _tooltip.SetActive(false);
    }
}

```

Листинг А.16 – Исходный код класса BalloonsGame

```

public class BalloonsGame : MonoBehaviour {
    public static BalloonsGame Instance;
    public GameObject EffectPrefab;

    [SerializeField] private GameObject _letterPrefab, _rowPrefab;
    [SerializeField] private Transform _rowsParent;
    [SerializeField] private TextMeshProUGUI _scoreText;
    [SerializeField] private GameObject _gameOver;
    [SerializeField] private GameObject _letterSelectorPrefab;
    [SerializeField] private TextMeshProUGUI _wordText;

    [SerializeField] private int _roundsCount = 3;
    [SerializeField] private int _rowsCount = 5, _lettersCount = 20;
    [SerializeField] private int _pointsPenalty = 1;
    [SerializeField] private float _generationDelay = 2f;
    [SerializeField] private float _rowWidth = 20f;

    private string _dictionary;

    private GameObject _letterSelector;

    private Vector2 _selectionStartPos, _selectionEndPos;

    private List<GameObject> _rows = new List<GameObject>();
    public List<string> _objectives = new List<string>();
    private float _scorePoints;
    [SerializeField] private float _rowSpacing = 1f;
    private int _rowsLimit = 6;
    private int _rowNumber = 0;
}

```

```

private float _time;
private float _result;
private List<string> _userAnswer = new List<string>();
private List<string> _correctAnswer = new List<string>();
private int _errorCount, _correctCount;

private void Awake() {
    if (Instance == null) Instance = this;
    else Destroy(gameObject);
}

private void Start() {
    GenerateDictionary();

    if (_rowsCount < _rowsLimit) _rowsLimit = _rowsCount;
    for (int i = 0; i < _rowsLimit; i++) GenerateRow();

    Debug.Log(GameManager.Instance.UserId);
}

private void Update() {
    _time += Time.deltaTime;
}

private void GenerateDictionary() {
    var dt = DbManager.GetTable("SELECT title AS \"Title\" FROM
public.dictionary;");
    string[] dictionaries = dt.AsEnumerable().Select(r =>
r.Field<string>("Title")).Distinct().ToArray();

    if (dictionaries.Length == 0) {
        Debug.LogError("Ошибка: проверьте настройки словарей");
    }

    _dictionary = dictionaries[Random.Range(0, dictionaries.Length)];
}

private void GenerateRow() {
    _rows.Add(Instantiate(_rowPrefab, _rowsParent));
    _rows[_rows.Count - 1].transform.position += _rows.Count * _rowSpacing *
Vector3.up;

    DataTable dt = DbManager.GetTable("SELECT text AS \"Text\" FROM
public.dictionary_word " +
"JOIN public.word ON
dictionary_word.word_id = word.word_id " +
"JOIN public.dictionary ON
dictionary_word.dictionary_id = dictionary.dictionary_id " +
$"WHERE syllables LIKE '%-%' AND title
= '{_dictionary}'");
    string[] words = dt.AsEnumerable().Select(r =>
r.Field<string>("Text")).Distinct().ToArray();

    string word = words[Random.Range(0, words.Length)];

    do {
        word = words[Random.Range(0, words.Length)];
    } while (!_objectives.Any(o => o == word));
}

```

```

_objectives.Add(word);

int wordBegin = Random.Range(0, _lettersCount - word.Length + 1);
int wordEnd = wordBegin + word.Length;

string wrongWord = "";
for (int letterNumber = 0; letterNumber < _lettersCount; letterNumber++)
{
    char letterValue;

    if ((letterNumber < wordBegin) || (letterNumber >= wordEnd)) {
        if (wrongWord.Length == 0) {
            wrongWord = words[Random.Range(0, words.Length)];
            do {
                var randomIndex = Random.Range(0, wrongWord.Length);
                var randomLetter = wrongWord.Substring(randomIndex, 1);
                wrongWord = wrongWord.Remove(randomIndex, 1);
                wrongWord = wrongWord.Insert(Random.Range(0,
wrongWord.Length), randomLetter);
            } while (words.Any(w => w == wrongWord));
        }

        letterValue = wrongWord[0];
        wrongWord = wrongWord.Remove(0, 1);
    }
    else {
        letterValue = word[letterNumber - wordBegin];
        if (wrongWord.Length > 0) wrongWord = "";
    }

    GameObject letter = Instantiate(_letterPrefab, _rows[_rows.Count -
1].transform);

    letter.GetComponent<SelectableLetter>().Init(letterValue.ToString());
}

_rowNumber++;
}
public void AddScore() {
    _scorePoints += 100 / _rowsCount;
    _scoreText.text = $"Баллы: {_scorePoints:0.00}%";
    _correctCount++;

    if (_objectives.Count == 0)
    {
        _result = (float) _correctCount / (_errorCount + _correctCount) *
100.0f;
        _gameOver.SetActive(true);
        DbManager.ExecuteQueryWithoutAnswer("INSERT INTO public.exercise
(game_id, user_id, duration, score) " +
                $"VALUES (1,
{GameManager.Instance.UserId}, {System.Math.Round(_time, 2)},
{System.Math.Round(_result, 2)});");
    }
}

public void TakeScore()
{

```



```

        _scorePoints -= 1 * _pointsPenalty;
        _scoreText.text = $"Баллы: {_scorePoints:0.00}%;";
        _errorCount++;
    }

    private void OnMouseDown()
    {
        var worldMousePos = Camera.main.ScreenToWorldPoint(Input.mousePosition);

        if (_letterSelector == null) {
            if (Input.GetMouseButtonDown(0)) {
                RaycastHit2D hit = Physics2D.Raycast(worldMousePos,
                Vector2.zero, 10f, LayerMask.GetMask("Letter"));
                Transform letter = hit.transform;
                if (letter != null) {
                    _selectionStartPos = worldMousePos;
                    _letterSelector = Instantiate(_letterSelectorPrefab,
                letter.position, Quaternion.identity);
                    _letterSelector.GetComponent<Selector>().Init(letter);
                }
            }
        }
        else {
            _selectionEndPos = worldMousePos;
            if (Selector.Items.Count > 1) {
                StringBuilder selectedWord = new StringBuilder();
                foreach (SelectableLetter letter in Selector.Items)
                    selectedWord.Append(letter.Value);

                if (_selectionStartPos.x > _selectionEndPos.x) {
                    char[] selectedLetters =
                selectedWord.ToString().ToCharArray();
                    System.Array.Reverse(selectedLetters);
                    selectedWord = new StringBuilder(new
                string(selectedLetters));
                }

                string objective = _objectives.FirstOrDefault(o => o ==
                selectedWord.ToString());
                if (objective != null) {
                    int objectiveIndex = _objectives.IndexOf(objective);
                    var test = Instantiate(EffectPrefab,
                _rows[objectiveIndex].transform.position, Quaternion.identity, transform);
                    Destroy(test, 1f);
                    Destroy(_rows[objectiveIndex]);
                    _rows.RemoveAt(objectiveIndex);
                    _objectives.Remove(objective);
                    if (_rowNumber < _rowsCount) GenerateRow();
                    AddScore();
                }
                else TakeScore();
            }
            Destroy(_letterSelector);
        }
    }
}

```

Листинг А.17 – Исходный код класса MagicianGame

```

public class MagicianGame : MonoBehaviour {
    public static MagicianGame Instance;
    public GameObject EffectPrefab;

    [SerializeField] private GameObject _syllablePrefab;
    [SerializeField] private Transform _answersRow;
    [SerializeField] private GameObject _gameOver;
    [SerializeField] private GameObject _letterSelectorPrefab;
    [SerializeField] private TextMeshPro _questionText;
    [SerializeField] private TextMeshProUGUI _scoreText;
    [SerializeField] private SpeechBubble _speechBubble;

    [SerializeField] private int _roundsCount = 3;
    [SerializeField] private float _pointsPenalty = 1f;
    [SerializeField] private float _generationDelay = 2f;

    private float _scorePoints;
    [SerializeField] private float _rowSpacing = 1f;
    [SerializeField] private int _answersCount = 3;

    private string correctAnswer, userAnswer;
    private bool isPause = false;
    private int _curRound = 0;
    private string _dictionary;

    private void Awake() {
        if (Instance == null) Instance = this;
        else Destroy(gameObject);
    }

    private void Start() {
        DataTable dictDT = DbManager.GetTable("SELECT title FROM
public.dictionary;");
        List<string> dictionaries = dictDT.AsEnumerable().Select(x =>
x.Field<string>("Title")).Distinct().ToList();
        _dictionary = dictionaries[Random.Range(0, dictionaries.Count)];

        GenerateLevel();
    }

    private void GenerateLevel() {
        if (_curRound == _roundsCount) {
            _gameOver.SetActive(true);
            return;
        }
        _curRound++;

        DataTable dt = DbManager.GetTable("SELECT text AS \"Text\" FROM
public.dictionary_word " +
            "JOIN public.word ON
dictionary_word.word_id = word.word_id " +
            "JOIN public.dictionary ON
dictionary_word.dictionary_id = dictionary.dictionary_id " +
            $"WHERE syllables LIKE '%-%' AND title
= '{_dictionary}';");
    }
}

```

```

        string[] words = dt.AsEnumerable().Select(r =>
r.Field<string>("Text")).Distinct().ToArray();

        string word = words[Random.Range(0, words.Length)];
        string[] wordSyllables = DbManager.ExecuteQueryWithAnswer($"SELECT
syllables FROM public.word WHERE text = '{word}';").Split('-');
        var randomInd = Random.Range(0, wordSyllables.Length);
        correctAnswer = wordSyllables[randomInd];
        wordSyllables[randomInd] = "...";
        _questionText.text = "";
        foreach (string s in wordSyllables) {
            _questionText.text += s;
        }

        List<string> answerOptions = new List<string>();
        var dt2 = DbManager.GetTable("SELECT text AS \"Text\" FROM
public.syllable;");
        List<string> syllables = dt2.AsEnumerable().Select(r =>
r.Field<string>("Text")).Distinct().ToList();
        syllables.Remove(correctAnswer);
        for (int i = 0; i < _answersCount - 1; i++) {
            string randomSyllable = syllables[Random.Range(0, syllables.Count)];
            answerOptions.Add(randomSyllable);
            syllables.Remove(randomSyllable);
        }
        answerOptions.Add(correctAnswer);
        System.Random r = new System.Random();
        answerOptions = answerOptions.OrderBy(a => r.Next()).ToList();

        foreach (Transform answer in _answersRow) {
            Destroy(answer.gameObject);
        }

        for (int i = 0; i < answerOptions.Count; i++) {
            var letter = Instantiate(_syllablePrefab, _answersRow);
            letter.GetComponent<Syllable>().Init(answerOptions[i]);
        }

        if (isPause) isPause = false;
    }

    public void AddScore() {
        _scorePoints += 100f / _roundsCount;
        _scoreText.text = $"Баллы: {_scorePoints:0.00}%";
    }

    public void TakeScore() {
        _scorePoints -= 1f * _pointsPenalty;
        _scoreText.text = $"Баллы: {_scorePoints:0.00}%";
    }

    private void OnMouseDown() {
        if (Input.GetMouseButtonDown(0) && !isPause) {
            var worldMousePos =
Camera.main.ScreenToWorldPoint(Input.mousePosition);
            RaycastHit2D hit = Physics2D.Raycast(worldMousePos, Vector2.zero,
10f, LayerMask.GetMask("Syllable"));
            Transform syllable = hit.transform;

```

```

        if (syllable != null) {
            userAnswer = syllable.GetComponent<Syllable>().ToString();
            if (userAnswer == correctAnswer) {
                isPause = true;
                _questionText.text = _questionText.text.Replace("...",
correctAnswer);
                AddScore();
                _speechBubble.Praise();
                StartCoroutine(GenerateWaiter(1f));
            }
            else {
                TakeScore();
                _speechBubble.Motivate();
            }
        }
    }
}

IEnumerator GenerateWaiter(float seconds) {
    yield return new WaitForSeconds(seconds);
    GenerateLevel();
}
}

```

Листинг А.18 – Исходный код класса ConveyorGame

```

public class ConveyorGame : MonoBehaviour {
    public static ConveyorGame Instance;
    public GameObject EffectPrefab;

    [SerializeField] private CardboardBox _cardboardBox;
    [SerializeField] private GameObject _gameOver;
    [SerializeField] private GameObject _letterSelectorPrefab;
    [SerializeField] private TextMeshPro _questionText;
    [SerializeField] private TextMeshProUGUI _scoreText;
    [SerializeField] private SpeechBubble _speechBubble;
    [SerializeField] private Checker _checker;
    [SerializeField] private Animator _background;

    [SerializeField] private int _roundsCount = 3;
    [SerializeField] private float _pointsPenalty = 1f;
    [SerializeField] private float _generationDelay = 2f;

    private float _scorePoints;
    [SerializeField] private float _rowSpacing = 1f;
    [SerializeField] private int _answersCount = 3;

    private string correctAnswer, userAnswer;
    public bool isPause = false;
    private int _curRound = 0;
    private string _dictionary;

    private void Awake() {
        if (Instance == null) Instance = this;
        else Destroy(gameObject);
    }
}

```

```

private void Start() {
    DataTable dictDT = DbManager.GetTable("SELECT title FROM
public.dictionary;");
    List<string> dictionaries = dictDT.AsEnumerable().Select(x =>
x.Field<string>("Title")).Distinct().ToList();
    _dictionary = dictionaries[Random.Range(0, dictionaries.Count)];

    GenerateLevel();
}

private void Update() {
    if (!isPause && !_background.GetBool("isActive"))
_background.SetBool("isActive", true);

    if (!isPause)
        _cardboardBox.transform.position += Vector3.left * Time.deltaTime *
4f;
}

public void GenerateLevel() {
    isPause = true;
    _background.SetBool("isActive", false);
    if (_curRound == _roundsCount) {
        _gameOver.SetActive(true);
        return;
    }
    _curRound++;

    DataTable dt = DbManager.GetTable("SELECT text AS \"Text\" FROM
public.dictionary_word " +
"JOIN public.word ON
dictionary_word.word_id = word.word_id " +
"JOIN public.dictionary ON
dictionary_word.dictionary_id = dictionary.dictionary_id " +
$"WHERE syllables LIKE '%-%' AND title
= '{_dictionary}'");
    string[] words = dt.AsEnumerable().Select(r =>
r.Field<string>("Text")).Distinct().ToArray();

    string word = words[Random.Range(0, words.Length)];

    _cardboardBox.Init(word);
    _checker.Refresh();
}

public void AddScore() {
    _scorePoints += 100f / _roundsCount;
    _scoreText.text = $"Баллы: {_scorePoints:0.00}%";
    _speechBubble.Praise();
}

public void TakeScore() {
    _scorePoints -= 1f * _pointsPenalty;
    _scoreText.text = $"Баллы: {_scorePoints:0.00}%";
    _speechBubble.Motivate();
}

```

```

IEnumerator GenerateWaiter(float seconds) {
    yield return new WaitForSeconds(seconds);
    GenerateLevel();
}

public void Run() {
    isPause = false;
}
}

```

Листинг А.19 – Исходный код класса SpeechBubble

```

public class SpeechBubble : MonoBehaviour {
    [SerializeField] private TextMeshPro _textMeshPro;

    private List<string> _praiseQuotes = new List<string>() { "Молодец!", "Так держать!", "Отличная работа!" };
    private List<string> _motivationalQuotes = new List<string>() { "Попробуй ещё раз.", "Я верю в тебя!", "У тебя всё получится.", "Ты справишься." };
    private string _prevQuote;
    private Coroutine _coroutine;
    [SerializeField] private float disappearDelay = 2f;

    public void Praise() {
        string randomQuote;
        do {
            randomQuote = _praiseQuotes[Random.Range(0, _praiseQuotes.Count)];
        } while (randomQuote == _prevQuote);
        _textMeshPro.text = _prevQuote = randomQuote;
        gameObject.SetActive(true);
        if (_coroutine != null) StopCoroutine(_coroutine);
        _coroutine = StartCoroutine(DisappearWaiter(disappearDelay));
    }

    public void Motivate() {
        string randomQuote;
        do {
            randomQuote = _motivationalQuotes[Random.Range(0,
            _motivationalQuotes.Count)];
        } while (randomQuote == _prevQuote);
        _textMeshPro.text = _prevQuote = randomQuote;
        gameObject.SetActive(true);
        if (_coroutine != null) StopCoroutine(_coroutine);
        _coroutine = StartCoroutine(DisappearWaiter(disappearDelay));
    }

    IEnumerator DisappearWaiter(float seconds) {
        yield return new WaitForSeconds(seconds);
        gameObject.SetActive(false);
    }
}

```

Листинг А.20 – Исходный код класса Selector

```

public class Selector : MonoBehaviour {
    public static Selector Instance;
}

```

```

private LayerMask _targetMask;

private float _origLeftEdgePosX, _origRightEdgePosX;
private float _targetWidth;

public static List<ISelectable> Items { get; private set; } = new
List<ISelectable>();

public void Init(Transform item)
{
    _targetWidth = item.GetComponent<RectTransform>().sizeDelta.x + 0.25f;
    GetComponent<SpriteRenderer>().size = new Vector2(_targetWidth,
GetComponent<SpriteRenderer>().size.y);

    _origLeftEdgePosX = transform.position.x - _targetWidth / 2;
    _origRightEdgePosX = transform.position.x + _targetWidth / 2;
    _targetMask = 1 << item.gameObject.layer;
    Select(item.GetComponent<ISelectable>());
}

void Start() {
    if (Instance == null) Instance = this;
    else Destroy(gameObject);
}

void Update() {
    //
    float halfWidth = GetComponent<SpriteRenderer>().size.x / 2;
    float leftEdgePosX = transform.position.x - halfWidth;
    float rightEdgePosX = transform.position.x + halfWidth;

    Vector3 worldMousePos =
Camera.main.ScreenToWorldPoint(Input.mousePosition);
    float centerDistanceX = worldMousePos.x - transform.position.x;
    float edgeDistanceX = Mathf.Abs(centerDistanceX) - halfWidth;
    bool isMouseRight = centerDistanceX > 0;
    bool isMouseIn = edgeDistanceX <= 0;

    float itemOffset = isMouseIn ? -_targetWidth / 2 : _targetWidth / 2;
    float moveDirectionX = isMouseRight ? rightEdgePosX + itemOffset :
leftEdgePosX - itemOffset;
    Vector2 moveDirection = new Vector2(moveDirectionX,
transform.position.y);
    RaycastHit2D hit = Physics2D.Raycast(moveDirection, Vector2.zero, 10f,
_targetMask);

    if (hit.transform != null) {
        float leftExtraWidth = _origLeftEdgePosX - leftEdgePosX;
        float rightExtraWidth = rightEdgePosX - _origRightEdgePosX;
        bool isOriginal = isMouseRight ? leftExtraWidth == 0 :
rightExtraWidth == 0;
        bool isReducible = isMouseRight ? rightExtraWidth > 0 :
leftExtraWidth > 0;
        Vector3 resizeVector = isMouseRight ? Vector3.right : Vector3.left;

        if (!isOriginal) {
            Resize(resizeVector, -_targetWidth);
        }
    }
}

```

```

        float moveDirectionX2 = isMouseRight ? leftEdgePosX +
_targetWidth / 2 : rightEdgePosX - _targetWidth / 2;
        Vector2 moveDirection2 = new Vector2(moveDirectionX2,
transform.position.y);
        RaycastHit2D hit2 = Physics2D.Raycast(moveDirection2,
Vector2.zero, 10f, _targetMask);
        Unselect(hit2.transform.GetComponent<ISelectable>());
    }
    else {
        if (!isMouseIn)
        {
            Resize(resizeVector, _targetWidth);
            Select(hit.transform.GetComponent<ISelectable>());
        }
        else if (isReducible && edgeDistanceX < -_targetWidth)
        {
            Resize(-resizeVector, -_targetWidth);
            Unselect(hit.transform.GetComponent<ISelectable>());
        }
    }
}

private void OnDestroy()
{
    for (int i = Items.Count - 1; i >= 0; i--)
    {
        Unselect(Items[i]);
    }
}

public void Select(ISelectable gameObject) {
    Items.Add(gameObject);
    gameObject.OnSelected();
}

public void Unselect(ISelectable gameObject) {
    Items.Remove(gameObject);
    gameObject.OnUnselected();
}

private void Resize(Vector3 direction, float value) {
    GetComponent<SpriteRenderer>().size += Vector2.right * value;
    transform.position += direction * _targetWidth / 2;
}
}

```

Листинг А.21 – Исходный код интерфейса ISelectable

```

public interface ISelectable {
    public void OnSelected();
    public void OnUnselected();
}

```


Листинг А.22 – Исходный код класса SelectableLetter

```
public class SelectableLetter : MonoBehaviour, ISelectable {
    [SerializeField] private SpriteRenderer _background;
    [SerializeField] private List<Sprite> _sprites;

    private TextMeshPro _text;
    private BoxCollider2D _collider;
    public GameObject Highlight;
    public string Value { get => _text.text; }

    public void Init(string text) {
        _text.text = text;
        _background.GetComponent<SpriteRenderer>().sprite =
        _sprites[Random.Range(0, _sprites.Count)];
    }

    private void Awake() {
        _text = GetComponentInChildren<TextMeshPro>();
        _collider = GetComponent<BoxCollider2D>();
    }

    private void Start() {
        _collider.size = new Vector2(1, 1);
    }

    public void OnSelected() {
        if (Highlight != null) Highlight.SetActive(true);
    }

    public void OnUnselected() {
        if (Highlight != null) Highlight.SetActive(false);
    }
}
```