

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(национальный исследовательский университет)
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой, доцент, к. т. н.
_____ К.А. Домбровский
« ____ » _____ 2017 г

Разработка приложения для подготовки детей к школе

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ- 09.03.01.2017.266 ПЗ ВКР

Руководитель проекта, доцент, к. т. н.
_____ Е. С. Ярош
« ____ » _____ 2017 г.

Автор проекта
студент группы КЭ-445
_____ И. А. Филиппов
« ____ » _____ 2017 г.

Нормоконтролер, ст. преп. каф.
«Электронные вычислительные
машины»
_____ В. В. Лурье
« ____ » _____ 2017 г.

Челябинск 2017

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ.....	5
1.1 Назначение и цели создания системы	5
1.2 Требования к приложению со стороны заказчика	6
1.3 Обзор родственных разработок.....	7
1.4 Выбор среды реализации	17
2 ОПИСАНИЕ ПРОЕКТНОГО РЕШЕНИЯ	26
2.1 Требования к разрабатываемому приложению	26
2.1.1 Основные элементы	26
2.1.2 Меню приложения	26
2.1.3 Пользовательский интерфейс	28
2.1.4 Ход выполнения занятия	29
2.1.5 Управление	30
2.1.6 Варианты использования системы.....	30
2.2 Архитектура приложения	32
2.2.1 Создание базы данных.....	32
2.2.2 Реализация приложения	36
3 ПРИМЕРЫ РАБОТЫ.....	49
ЗАКЛЮЧЕНИЕ	56
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	57

ВВЕДЕНИЕ

В современных условиях компьютер стал неотъемлемой частью нашей жизни. Он используется и как рабочий инструмент, и как средство организации досуга. Поэтому растет актуальность разработки разного рода приложений, в том числе видеоигр.

Более 80% детей и множество взрослых играют в видеоигры. Однако если взрослый человек может сам для себя определить полезность и актуальность игр, то для детей это определяют их родители или воспитатели в детских учреждениях. Большинство видеоигр, которые рассчитаны на массового пользователя, имеют развлекательный характер и не всегда могут развивать какие-либо способности ребенка, а узконаправленных развивающих игр на данный момент очень мало.

Развивающие игры – это игры, которые разработаны специально для активизации и совершенствования необходимых навыков. Играя в данные игры, дети тренируют такие способности, как мышление, воображение, память, внимание, речь и т.д. Для каждого детского возраста и для каждого отдельного ребенка может быть подобрана игра, которая развивает то, что необходимо именно сейчас. Для самых маленьких подойдут игры на развитие речи, крупной и мелкой моторики рук, а также упражнения, развивающие органы чувств. А для детей, которые готовятся перейти к школьному обучению, важны занятия с упором на умственное развитие.

Для родителей и воспитателей крайне важно, чтобы ребенок развивался по строго определенным критериям. При возникновении каких-либо проблем в развитии тех или иных навыков, эти проблемы необходимо фиксировать и анализировать.

Целью данной выпускной квалификационной работы является разработка приложения для подготовки детей старшего дошкольного возраста к школе. Приложение содержит комплекс занятий, а также собирает

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		3

и анализирует статистику выполнения данных занятий, что очень важно для родителей и воспитателей.

Разработка выполнялась на основе игрового движка Unity 5.3 с использованием Visual Studio 2015 и СУБД SQLite.

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		4

1 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ

1.1 Назначение и цели создания системы

Приложение предназначено для развития детей старшего дошкольного возраста и подготовке их к школе.

Приложение состоит из комплекса игровых развивающих занятий длительностью не более 10 минут. Каждое занятие содержит несколько заданий, которые направлены на развитие различных умений и навыков, таких как: логическое мышление, речь и грамота, элементарные математические представления, внимание и память, а также восприятие и воображение.

Важной особенностью приложения является структурированная система заданий, суть которой заключается в группировании всех имеющихся заданий по типу развития тех или иных навыков и способностей ребенка. Таким образом, каждый раз для ребенка можно выбрать один из навыков и развивать только его.

Также система предназначена для родителей ребенка или воспитателя, которым необходимо вести учет неудач для дальнейшего анализа и принятия необходимых решений.

Отличительным аспектом приложения является наличие звукового сопровождения. Правильные и неправильные ответы на задания будут отмечаться соответствующими звуками. Также по запросу пользователя может быть включено озвучивание текста задания (если ребенок занимается один и не умеет читать).

Если ребенок устал выполнять задания или они ему не нравятся, он может прервать занятие в любое время.

Цели создания системы:

1. Подготовка детей старшего дошкольного возраста к школе;
2. Облегчение поиска игровых занятий для родителей и воспитателей;

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		5

3. Развитие и совершенствование у детей различных навыков и умений (развитие логического мышления, закрепление представления об окружающем мире, развитие речи и обучение грамоте и т.д.);

4. Занятие и развлечение ребенка на непродолжительное время;

5. Разнообразие игровой деятельности детей;

6. Предоставление удобной системы сбора статистики по определенному ребенку.

Система предназначена для решения перечисленных ниже задач:

1. Создание комплекса занятий, состоящих из нескольких различных заданий;

2. Группирование заданий по типу совершенствования навыков;

3. Обучение детей дома и в детском саду;

4. Создание структурированной статистики, показывающей неудачи ребенка;

5. Знакомство ребенка с персональным компьютером.

1.2 Требования к приложению со стороны заказчика

Со стороны заказчика к приложению предъявляются следующие требования:

1. Наличие необходимых заданий для развития ребенка;

2. Ограничение на количество выполняемых заданий в занятии (не более 5);

3. Отсутствие ненужных, отвлекающих объектов на экране, а также большого количества информации;

4. Спокойная и яркая, но не раздражающая цветовая палитра приложения и графических рисунков;

5. Легко читаемый текст, грамотное использование шрифтов;

6. Наличие голосовых сопровождений;

7. Учет статистики выполнения занятий для каждого ребенка;

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		6

8. Возможность отображения подсказки на экране по запросу для каждого задания;
9. Наличие всплывающих верных/неверных ответов в зависимости от выбранного ответа;
10. Возможность самостоятельного выполнения заданий ребенком без присмотра родителей или воспитателя;
11. Возможность прервать выполнение задания в любой момент.

1.3 Обзор родственных разработок

Обучающая компьютерная игра «Тренажер мозга», разработчик Бука софт.[6]

Данная компьютерная игра позволяет развивать многие важнейшие качества и усовершенствовать их посредством игровых упражнений. Выполняя разнообразные задания, ребенок будет тренировать свое внимание, память, логику, реакцию и умение считать. А в режиме экзамена есть возможность проверить полученные навыки.

Достоинства:

1. Обучение работе с интерфейсом игры;
2. Наличие разноплановых заданий;
3. Наличие заданий для детей старшего школьного возраста от 5 лет и выше.

Недостатки:

1. Неудобное расположение и отсутствие текстового оформления многих интерфейсных кнопок;
2. Сложный пользовательский интерфейс;
3. Невнятное и отталкивающее звуковое сопровождение;
4. Очень мелкие графические рисунки и картинки;

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		7

5. Наличие отсчета времени на каждое задание, что мешает ребенку сфокусироваться на его выполнении.

Отсутствуют возможности:

1. Выполнять смешанные задания на развитие отличных друг от друга навыков;
2. Озвучивание текста заданий и верных/неверных ответов по запросу пользователя;
3. Просмотр подсказок при выполнении задания;
4. Выбор уровня сложности заданий;
5. Прерывать занятие в любое время.

Игра предназначена для работы исключительно на персональном компьютере пользователя и не требует подключения к сети интернет или локальной сети.

Пример работы Обучающей компьютерной игры Тренажер мозга изображен на рис. 1.1 и рис. 1.2.

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		8



Рис. 1.1 – Начальный экран в игре Тренажер мозга

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		9

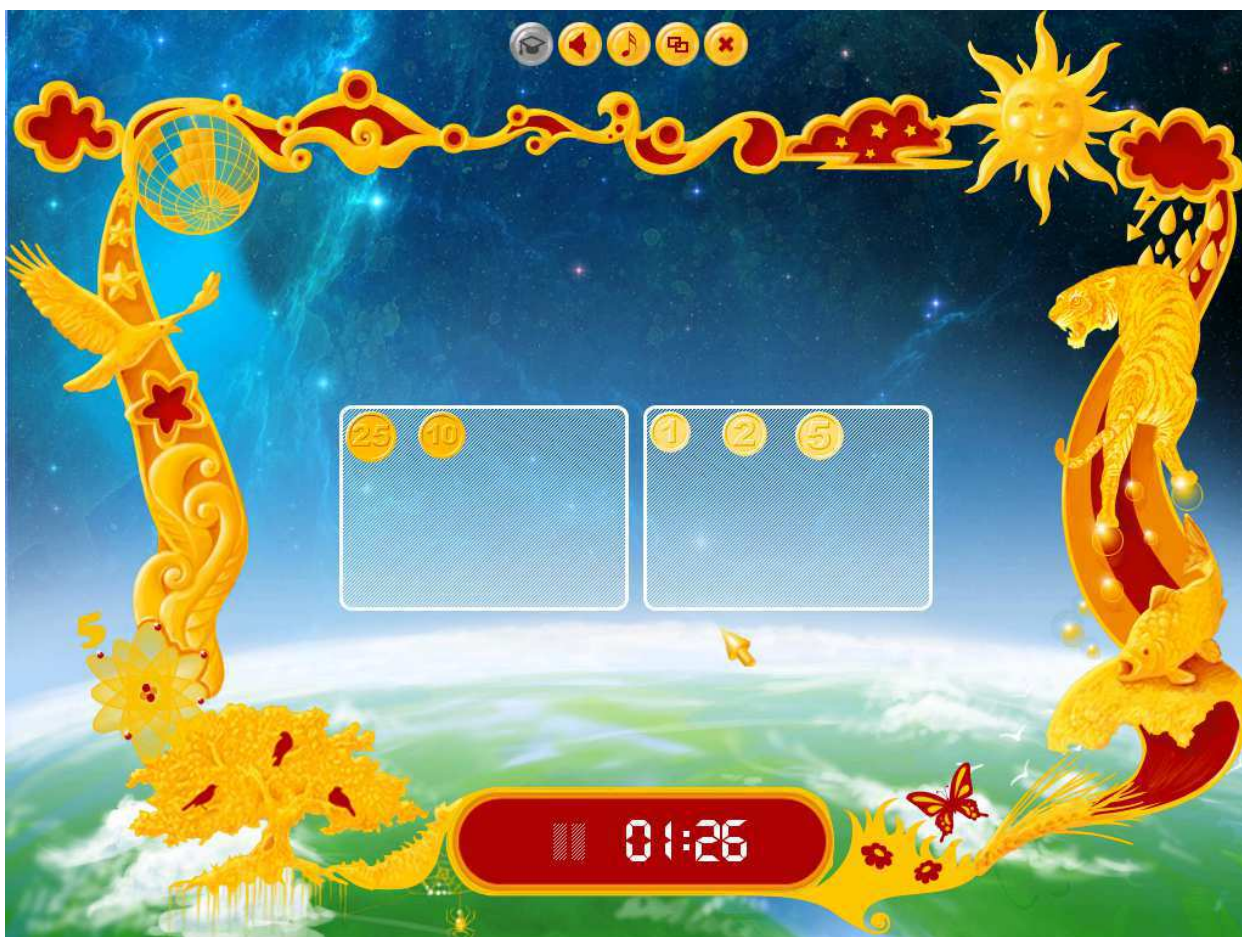


Рис. 1.2 – Задание на умение считать в игре Тренажер мозга

Компьютерная игра Раннее развитие: Логика и память, разработчик «Издательство, ООО».[8]

Эта игра выполнена в виде небольшого путешествия по сказочному миру вместе с домовёнком «Кузей», помогая ребенку в выполнении разнообразных заданий, тренируя исключительно логическое мышление и слуховую и зрительную память. Уникальная возможность данной игры состоит в том, что родителю или воспитателю не нужно уделять много времени для работы с ребенком, т.к. приложение выполнено в виде игрового модуля.

Достоинства:

1. Красивое графическое оформление игры;
2. Наличие звукового сопровождения как описания задания, так и некоторого общения с ребенком;

Изм	Лист	№ докум.	Подпись	Дата

09.03.01.2017.266.00 ПЗ

Лист

10

3. Элементы интерактивного общения с ребенком;
4. Наличие обучения.

Недостатки:

1. Невозможность прервать длинные текстовые, аудио и видео вставки;
2. Сложный для ребенка пользовательский интерфейс;
3. Наличие заданий на развитие и совершенствования малого количества навыков;
4. Невозможность сохранить игровой процесс.

Отсутствуют возможности:

1. Ведение статистики по конкретному пользователю;
2. Развитие и усовершенствование математических и речевых способностей ребенка;
3. Возможность использования подсказок;
4. Отключение звуковых сопровождения во время выполнения заданий.

Игра предназначена для работы исключительно на персональном компьютере пользователя и не требует подключения к сети интернет или локальной сети.

Пример работы игры Раннее развитие: Логика и память изображен на рис. 1.3 и рис. 1.4.

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		11

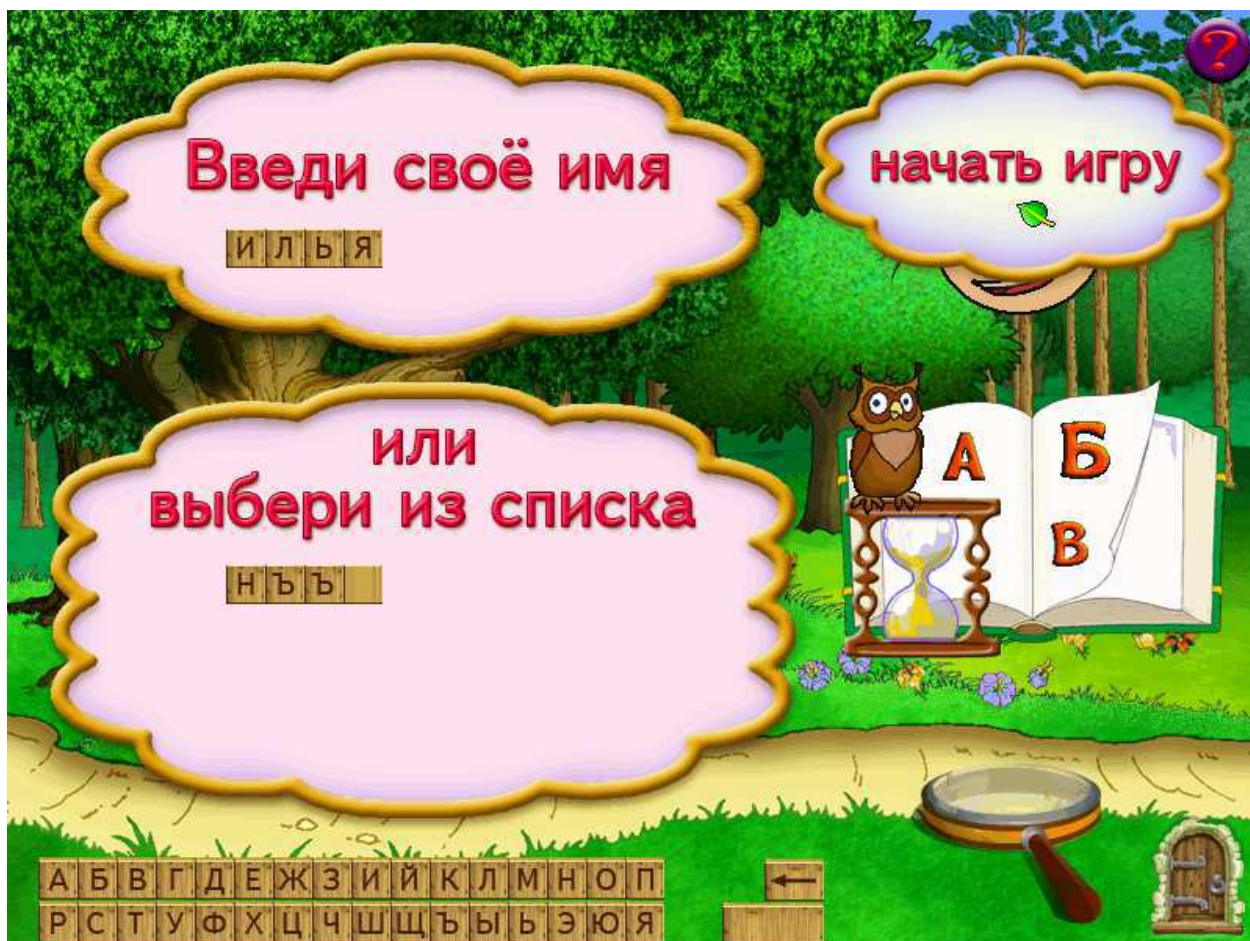


Рис. 1.3 – Начальное меню игры Раннее развитие: Логика и память

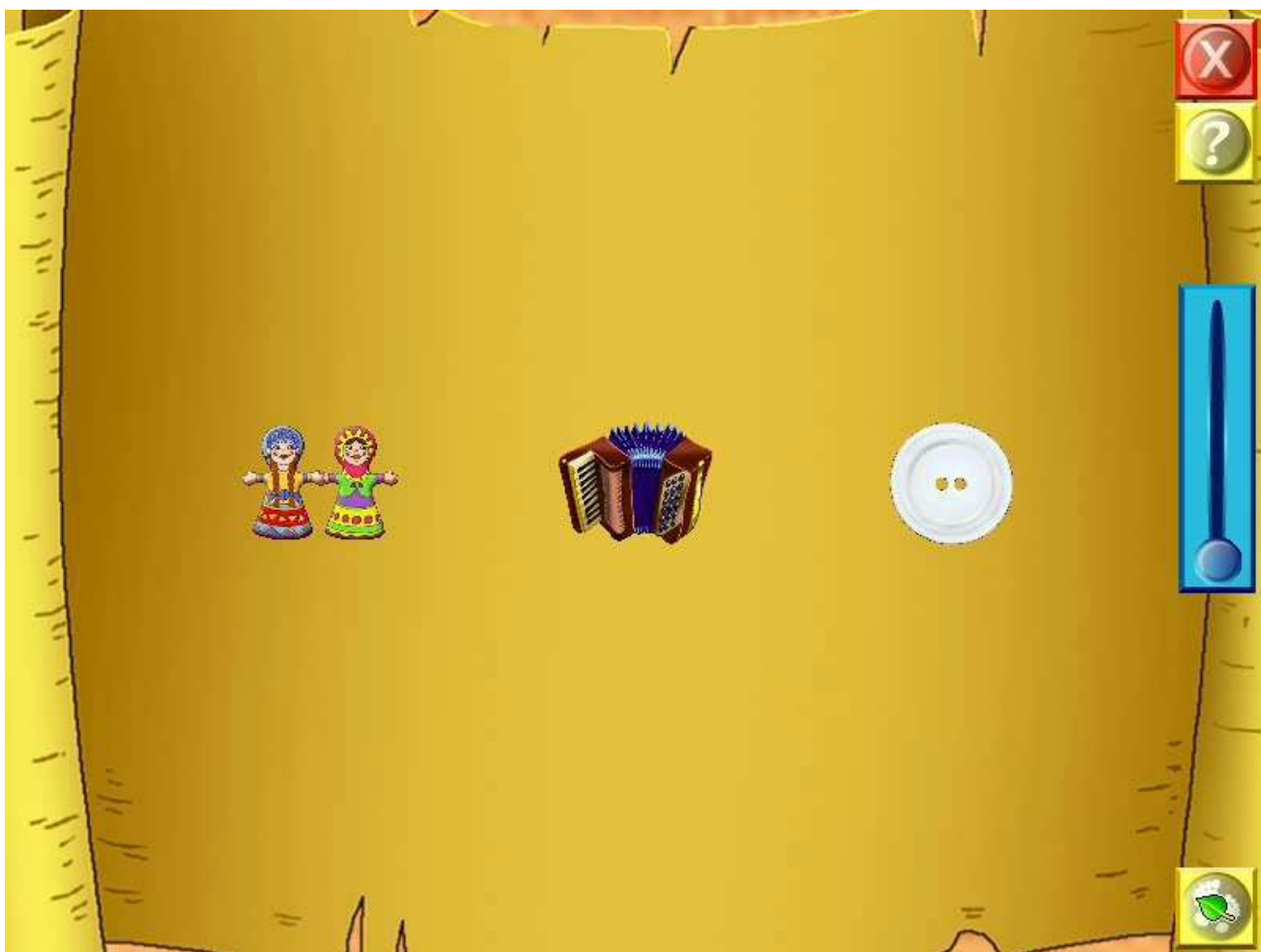


Рис. 1.4 – Выполнение заданий на память в игре Раннее развитие:
Логика и память

Компьютерная игра Скоро в школу: Сказочный учебник, разработчик ID Company[9]

Скоро в школу: Сказочный учебник – это уникальная компьютерная игра, разработанная специально для того, чтобы подготовить детей дошкольного возраста к первому походу в школу. С помощью любимых персонажей из известных детских сказок игра помогает развивать и тренировать такие важные навыки, как внимание, логическое мышление, память, речь. Помимо этого, ответы на любопытные вопросы помогут дошкольнику узнать много нового, интересного и неизведанного.

Достоинства:

1. Наличие сказочных персонажей;

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		13

2. Наличие заданий на развитие большинства важных навыков и умений;
3. Элементы интерактивного общения с ребенком.

Недостатки:

1. Сложный пользовательский интерфейс;
2. Отвлекающие внимание ребенка различные анимации и изображения;
3. Плохо читаемый текст, чересчур яркая цветовая палитра.

Отсутствуют возможности:

1. Звукового сопровождения;
2. Развития исключительно конкретного навыка или умения;
3. Сбор статистических данных для конкретного пользователя.

Игра предназначена для работы исключительно на персональном компьютере пользователя и не требует подключения к сети интернет или локальной сети.

Пример работы игры Скоро в школу: Сказочный учебник приведен на рис. 1.5 и рис. 1.6.

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		14



Рис. 1.5 – Начальное меню игры Скоро в школу: Сказочный учебник

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		15

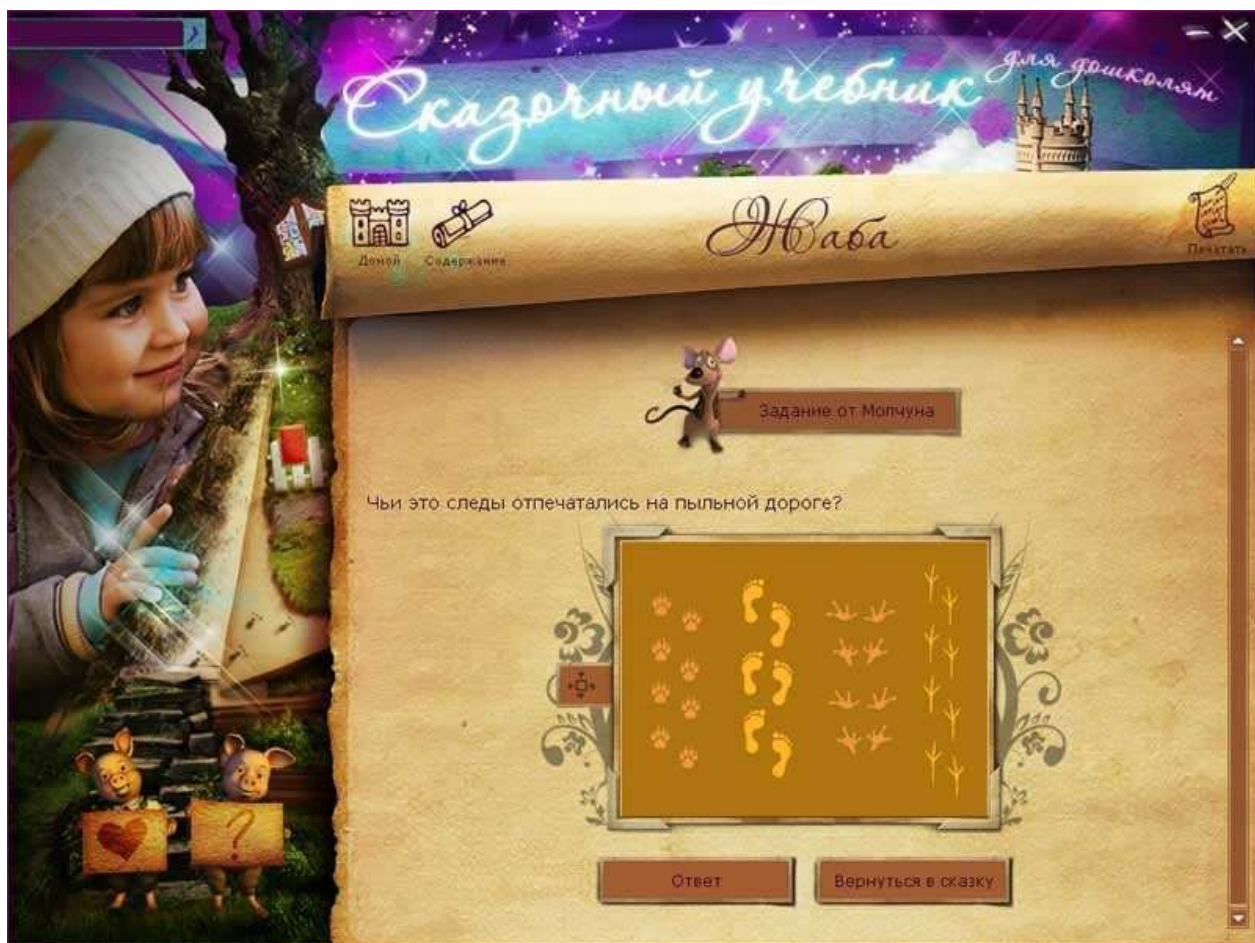


Рис. 1.6 – Выполнение задания на логику в игре Скоро в школу:
Сказочный учебник

Рассмотренные выше родственные разработки хоть и выполняют основную функцию подготовки детей старшего дошкольного возраста к школе, но имеют ряд важных недостатков и ограничений:

1. Наличие сложного пользовательского интерфейса, который ограничит возможности ребенка в самостоятельной работе с приложением;
2. Отсутствие, либо сложность поиска важных кнопок (отключение звукового сопровождения, пропуск видео и текстовых вставок);
3. Ни у одной из игр нет пропуска задания или полного окончания занятия;
4. Не у всех игр есть возможность выбора категорий заданий;

5. Ни у одной из игр нет возможности сбора и анализа статистических данных.

Разрабатываемое приложение имеет особенности: при выполнении заданий пользователю не дается ограничение по времени, также есть возможность пропустить задание или вообще отказаться от его выполнения, а выполняемые задания имеют различные категории развития навыков. Одной из важных функций приложения является сбор статистических данных, которые требуются родителям или воспитателям для анализа показателей ребенка и работы над совершенствованием конкретных навыков.

1.4 Выбор среды реализации

При построении программного продукта важно учитывать множество различных аспектов, одним из которых является кроссплатформенность приложения.

Кроссплатформенность помогает добиться результата работы приложения не только на одной, а на множестве аппаратных платформ и операционных систем. Обеспечивается это благодаря разработке исходного продукта на высокоуровневых языках программирования, средствах разработки и выполнения, компиляции, компоновки и исполнения программного кода на нескольких платформах. Так, например, одно приложение может выполняться сразу на Windows и Android одновременно.

Исходя из этого, разработка кроссплатформенного приложения на рынке ценится выше приложений, которые рассчитаны исключительно на одну платформу, и именно поэтому было решено в качестве инструмента разработки использовать один из игровых движков.

Логично, что для разрабатываемого приложения не нужно выбирать платные движки или движки с ограниченной функциональностью, учитывая, что свободно распространяемых продуктов более чем достаточно.

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		17

Одним из наиболее старых и весьма популярных Freeware движков является *Unreal Engine*, который уже давно используется начинающими и профессиональными разработчиками игр. Движок написан на высокоуровневом языке программирования C++, поддерживается разработка на большинстве популярных современных операционных системах: Windows, Mac OS, Linux. В Unreal Engine существует огромное количество мощных инструментов для работы с графическими материалами, светом, физикой материалов и прочим.

Достоинства:

1. Кроссплатформенность разработки игр;
2. Разработка на языке высокого уровня;
3. Возможность разработки 3D игр с использованием большого количества рендеринг систем;
4. Поддержка работы с сетью;
5. Бесплатный для начинающих разработчиков.

Недостатки:

1. Сложность освоения для начинающих разработчиков, непростой графический интерфейс;
2. Отсутствие в магазине движка бесплатных наработок, в том числе готовых скриптов;
3. Тяжелый с точки зрения вычислительных мощностей персонального компьютера разработчика.

Интерфейс движка Unreal Engine 4 представлен на рис. 1.7.

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		18

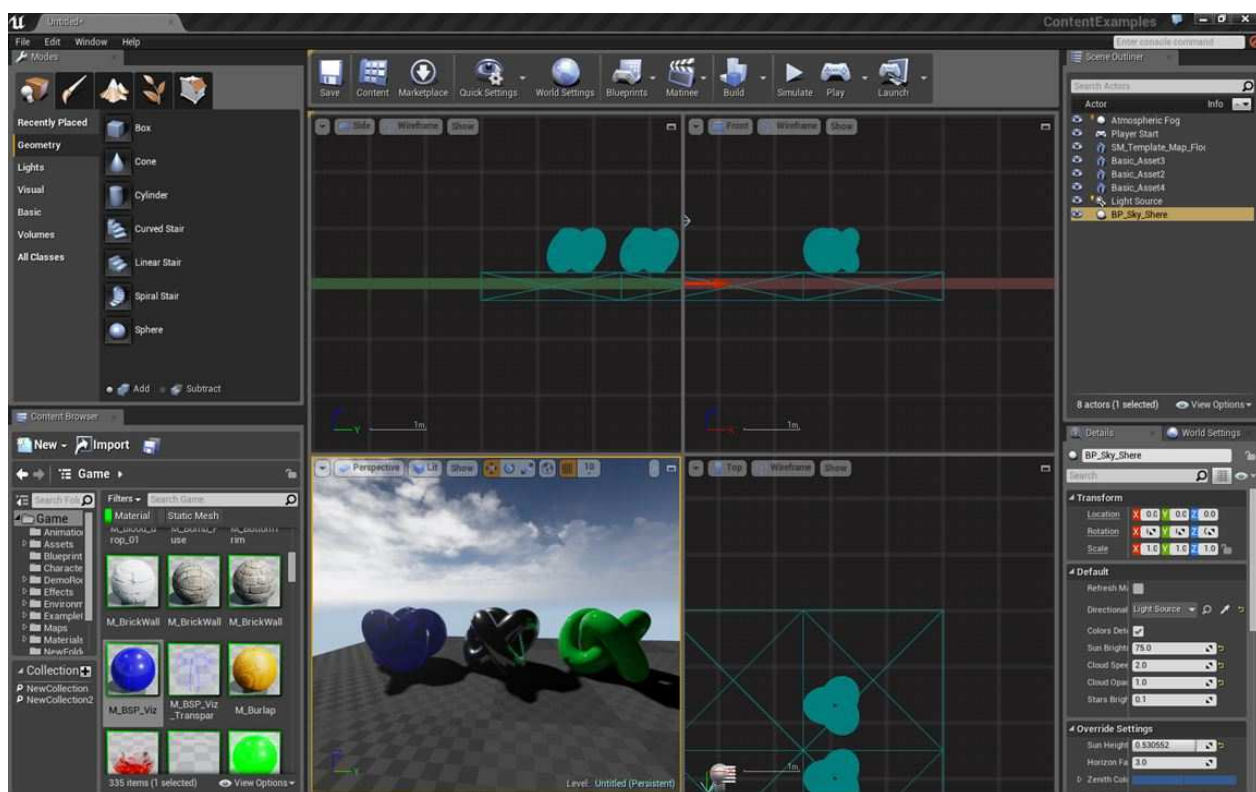


Рис. 1.7 – Графический интерфейс движка Unreal Engine 4

Какими бы достоинствами ни обладал данный движок, он «заточен» исключительно под разработку высококачественных 3D игр. Большинство функций вообще не будет использоваться в разработке текущей системы.

Game Maker: Studio – это проприетарный и весьма популярный среди разработчиков движок, который является продолжением своего предшественника Game Maker. Главной отличительной особенностью между ними является наличие кроссплатформенной разработки 2D и 3D игр. Бесплатность данной платформы ограничивается компиляцией и работой исключительно под Windows.

Текущая версия движка написана на Delphi и позволяет писать программный код исключительно на собственном языке программирования GML (Game Maker Language).

Достоинства:

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		19

1. Кроссплатформенность разработки игр;
2. Поддержка библиотек и расширений, в том числе на разных языках;
3. Поддержка работы с сетью;
4. Не требует больших вычислительных средств персонального компьютера разработчика;
5. Бесплатный для начинающих разработчиков.

Недостатки:

1. Отсутствие магазина бесплатных наработок, в том числе готовых скриптов;
2. Ограниченный функционал бесплатной версии;
3. Разработка игр только на одной операционной системе Microsoft Windows;
4. Собственный язык программирования GML (Game Maker Language);
5. Непростой графический интерфейс.

Интерфейс движка Game Maker: Studio представлен на рис. 1.8.

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		20

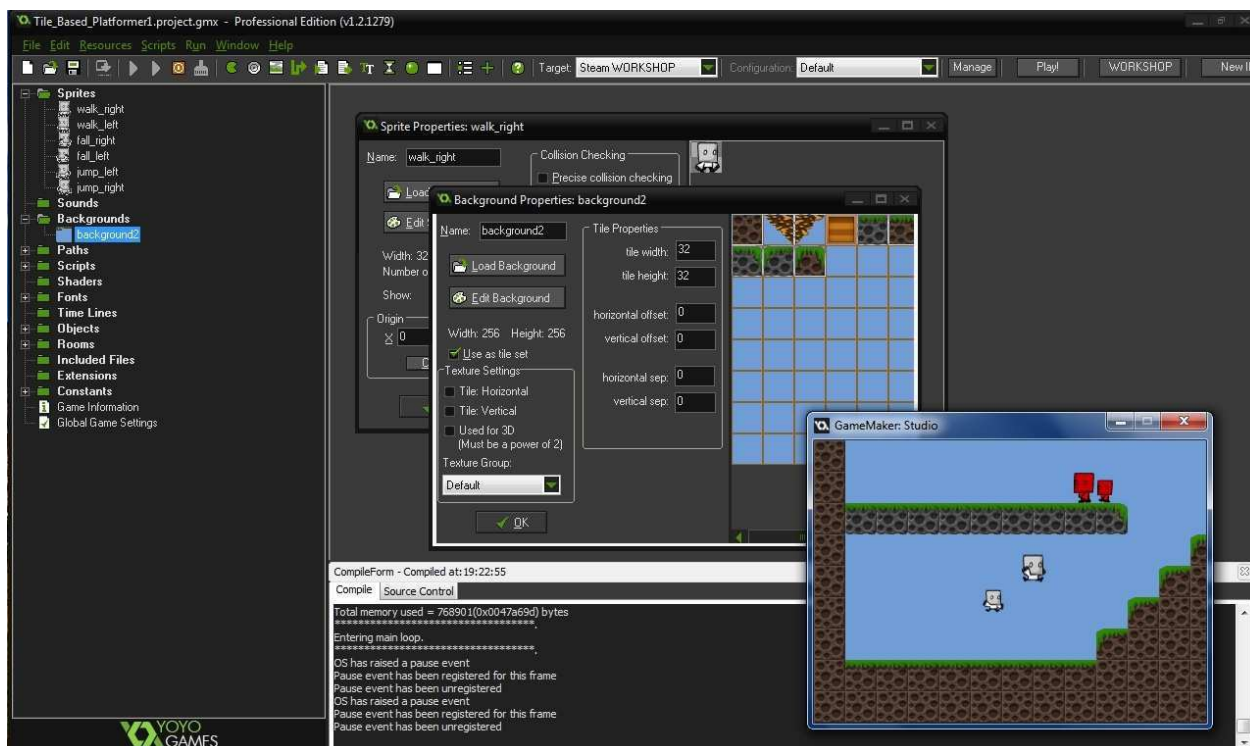


Рис. 1.8 – Графический интерфейс движка Game Maker: Studio

Главный недостаток ограниченного функционала бесплатной версии Game Maker: Studio – весьма неудобный графическим интерфейс и необходимость написания программного кода исключительно на собственном языке программирования. Это заставляет отбросить выбор данного конструктора в качестве разработки приложения.

Последний рассматриваемый игровой движок уже давно зарекомендовал себя как среди начинающих индивидуальных разработчиков, так и среди профессиональных команд разработчиков игр. *Unity3D* обладает всеми возможностями создания двух- и трехмерных приложений и игр сразу на огромное количество платформ: Windows OS, Linux, Android, Apple iOS, на большинстве популярных игровых приставках, а также для запуска в браузере.

Drag and Drop интерфейс *Unity3D* разбит на несколько окон с различным функционалом, который очень просто настраивать и перемещать. Для разработчика предоставлен выбор написания программного кода между

двумя сценарными языками: C# и JavaScript. Движок поддерживает множество популярных редакторов кода и интегрированных сред разработки.

Созданный проект на Unity3D делится на так называемые сцены (уровни) – отдельные файлы, каждый из которых содержит свою реализацию окружающего мира, объектов, сценариев и настроек.

Достоинства:

1. Кроссплатформенность разработки игр;
2. Поддержка работы с сетью;
3. Присутствие магазина с бесплатными наработками и скриптами;
4. Понятный и удобный графический интерфейс;
5. Предоставление мануалов и документации на официальном сайте разработчика движка;
6. Использование сразу двух языков программирования;
7. Не требует больших вычислительных средств персонального компьютера разработчика;
8. Система контроля версий;
9. Оптимизация для больших проектов;
10. Бесплатный для начинающих разработчиков.

Недостатки:

1. Ограниченный функционал бесплатной версии для разработки под iOS, Linux и на игровые приставки;
2. Трудность в подключении внешних модулей и библиотек;
3. Отсутствие полноценного тестирования программного кода;
4. Невозможность получить исходный код движка.

Интерфейс движка Unity3D представлен на рис. 1.9.

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		22

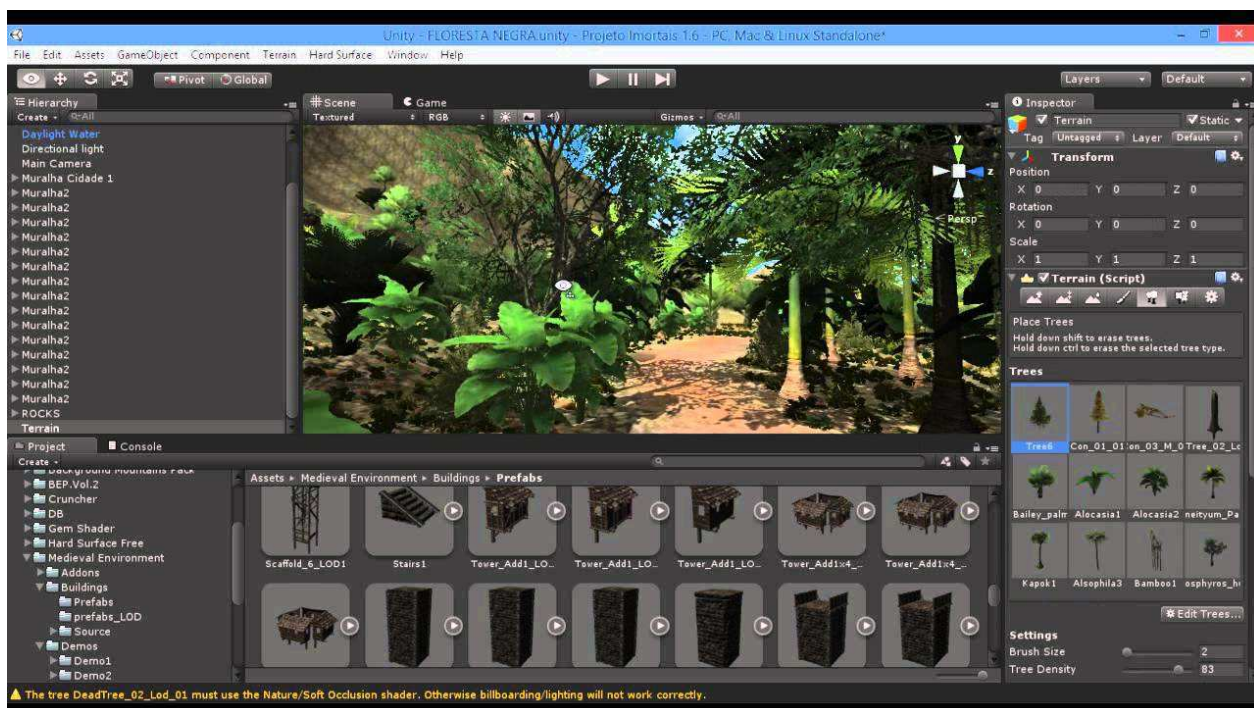


Рис. 1.9 – Графический интерфейс движка Unity3D

Unity3D обладает мощной базой компонентов для разработки как 2D так и 3D игр разного жанра: это и интуитивно понятный графический интерфейс, и наличие разработки кроссплатформенных игр и приложений, и бесплатность использования на многих операционных системах, а также огромное хранилище документации, мануалов, обучающих материалов и исходного кода.

В сумме всех достоинств и небольшого количества недостатков, Unity3D будет наиболее подходящей платформой для создания приложения для подготовки детей к школе.

Исходя из вышесказанного, разработка будет вестись на сценарном языке программирования C#, поддерживаемом Unity3D.

Язык C# - это детище компании Microsoft. В него входит много полезных особенностей - простота, объектная ориентированность, типовая защищенность, "сборка мусора", поддержка совместимости версий и многое другое. Данные возможности позволяют быстро и легко разрабатывать приложения, особенно COM+ приложения и Web сервисы. При создании C#, его авторы учитывали достижения многих других языков программирования: C++, C, Java, SmallTalk, Delphi, Visual Basic и т.д. Надо заметить, что по

причине того, что C# разрабатывался с чистого листа, у его авторов была возможность (которой они явно воспользовались), оставить в прошлом все неудобные и неприятные особенности (существующие, как правило, для обратной совместимости), любого из предшествующих ему языков. В результате получился действительно простой, удобный и современный язык, по мощности не уступающий C++, но существенно повышающий продуктивность разработок [1].

За хранение различной информации, необходимой для работы приложения, будет использоваться система управления базами данных SQLite.

SQLite – это бесплатная компактная реляционная СУБД, которая встраивается в приложение и используется им. Кроссплатформенность SQLite дает возможность встраивать эту базу данных во многие популярные платформы. В сравнении с серверными БД, предоставляет отличный набор инструментов для более простой обработки любых видов данных. Приложение, в котором используется данная СУБД, всегда будет более быстрым и производительным, т.к. связь между ними происходит с помощью прямых вызовов файлов, а не какого-то определенного интерфейса.

Преимущества данной СУБД:

1. Вся база данных хранится в одном файле, что помогает при перемещении;
2. Кроссплатформенность;
3. Стандартизированность;
4. Надежность;
5. Занимает мало места на жестком диске;
6. Динамическое типизирование данных;
7. Масштабируемость и огромный набор функций.

В качестве интегрированной среды разработки будет использоваться Microsoft Visual Studio, которая включает в себя редактор исходного кода, встроенный отладчик и подключаемые внешние библиотеки.

Вывод по разделу:

С учетом требований заказчика и на основании анализа полученных данных, были выбраны бесплатные программные средства для решения поставленной задачи.

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		25

2 ОПИСАНИЕ ПРОЕКТНОГО РЕШЕНИЯ

2.1 Требования к разрабатываемому приложению

Разрабатываемое приложение должно представлять собой двумерную развивающую игру для детей старшего дошкольного возраста.

2.1.1 Основные элементы

Основной элемент разрабатываемого приложения – задания, которые разделяются по уровням сложности и категориям развития навыков. Задания представляют собой некую развивающую задачу в игровой форме, цель которой – правильный выбор ответа.

Уровни сложности заданий:

- 1) Легко;
- 2) Сложно.

Категории заданий:

- 1) Внимание;
- 2) Логика;
- 3) Математика;
- 4) Память;
- 5) Речь.

По типу взаимодействия с пользователем задания должны быть на перемещение элементов по экрану, клику клавишей мыши по определенному элементу задания или же клику по нескольким элементам.

Различия в уровнях сложности зависит как от категории заданий, так и от того, как реализовано взаимодействие с пользователем.

2.1.2 Меню приложения

При запуске приложения, открывается главное меню. Основным пунктом данного меню является кнопка «Играть!», которая предназначена специально для детей, которые хотят без выбора различных настроек начать выполнять игровые задания. Также в меню присутствует пункт

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		26

«Администратор», который перенаправляет пользователя в администраторское меню.

Меню администратора должно состоять из трех ключевых пунктов – «Начать занятие», «Новый игрок» и «Статистика».

Пункт меню «Начать занятие» позволяет пользователю стартовать занятие и выполнять задания. В зависимости от режима пользователь может выполнять задания по-разному:

- 1 режим занятия заключается в том, чтобы сформировать 5 случайных заданий на развитие различных навыков (по одному заданию из категории заданий);
- 2 режим занятия формирует случайные задания только на развитие одного из навыков (предоставляется список категорий на выбор).

Также перед стартом занятия пользователю предоставляется выбор уровня сложности и выбор ФИО игрока (ребенка). Если все это не будет выбрано, то система выведет соответствующее сообщение на экран.

Пункт меню «Новый игрок» необходим для регистрации нового пользователя и занесения его в базу данных для дальнейшего сохранения статистики.

Последний пункт «Статистика» выводит статистические данные по конкретно выбранному пользователю. База данных обеспечивает хранение статистики для каждого из выполненных/невыполненных заданий игрока. В строчку выводится следующая информация: дата выполнения задания, наименование задания, сложность, категория, выполнено или нет, было ли игроку сложно и/или не интересно при выполнении задания, а также что не получилось.

В каждом из выбранных пунктов меню присутствует кнопка «Назад», которая возвращает пользователя в основное главное меню приложения.

Схема связи основного меню приложения представлена на рис. 2.1

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		27



Рис. 2.1 – Схема связи основного меню приложения

2.1.3 Пользовательский интерфейс

Интерфейс приложения не должен быть как громоздким, так и слишком маленьким, элементы не должны наезжать друг на друга, либо смещаться относительно основного экрана приложения.

Пользовательский интерфейс приложения располагается на экране при выполнении заданий. Каждая кнопка, переключатель и поле для ввода текста предназначено для достижения конкретных целей.

Переключатель «Сложное» служит для обозначения пользователем текущего задания сложным для выполнения. Соответствующий выбор заносится в базу данных.

Переключатель «Не интересное» необходим в том случае, если пользователю не интересно выполняемое задание. Соответствующий выбор заносится в базу данных.

При нажатии на кнопку «Озвучить», выполнится полное озвучивание текста задания голосовым сопровождением.

Поле ввода текста «Что не получилось» необходимо для организации обратной связи с пользователем, куда он вводит информацию о том, что не получилось при выполнении текущего задания. Соответствующая информация заносится в базу данных.

Кнопка «Продолжить» выполняет переключение следующего задания. Если при выполнении текущего задания не был выбран ни один из вариантов

ответа, то в базу данных заносится запись о том, что задание не было выполнено.

Кнопка «Окончить занятие» полностью прекращает выполнение занятия и переносит пользователя в главное меню, статистика при этом сохраняется после последнего выполненного задания.

Кнопка «Подсказка» выводит на экран правильные ответы для текущего задания.

2.1.4 Ход выполнения занятия

После того, как администратор выбрал необходимый режим и уровень сложности, начинается занятие. Формируется список из случайных заданий, которые выводятся один за другим. Для каждого задания есть текст задачи, находящийся в верхней части экрана.

Основная задача игрока – дать правильный выбор ответа на все задания. Для этого ему необходимо будет с помощью манипулятора «мышь» кликать по соответствующим картинкам либо перемещать их по экрану для правильного расположения.

При выполнении заданий в левом нижнем углу экрана сказочный персонаж будет в виде текстового «диалога» обозначать правильность выбора того или иного ответа. Также это будет сопровождаться озвученным голосовым сообщением. Для заданий, где необходимо выбрать несколько вариантов ответа, выводятся «галочки» и «крестики» при выборе соответствующего правильного и не правильного ответа на вопрос.

Завершение занятия происходит по двум причинам:

- 1) Пользователь выполнил все задания;
- 2) Пользователь самостоятельно с помощью кнопки завершил занятие.

После завершения на экране выводится текст соответствующего содержания с возможностью возвращения в главное меню.

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		29

Если игровые задания были запущены без использования администраторского режима (пункт главного меню “Играть!”), задания формируются случайным образом без учета всех настроек (режима занятия, уровня сложности, категории заданий). При выполнении заданий все статистические данные, которые имеются в режиме занятий, сохраняться не будут.

2.1.5 Управление

Управление приложением должно быть наиболее простым и понятным пользователю. Все взаимодействия с приложением, такие как выбор пунктов меню, ответ на задания, пользование графическим интерфейсом должны осуществляться с помощью левой клавиши манипулятора «мышь».

2.1.6 Варианты использования системы

Исходя из анализа требований к системе, была сформирована модель использования (рис. 2.2.).

					09.03.01.2017.266.00 ПЗ	Лист
						30
Изм	Лист	№ докум.	Подпись	Дата		

USE CASE

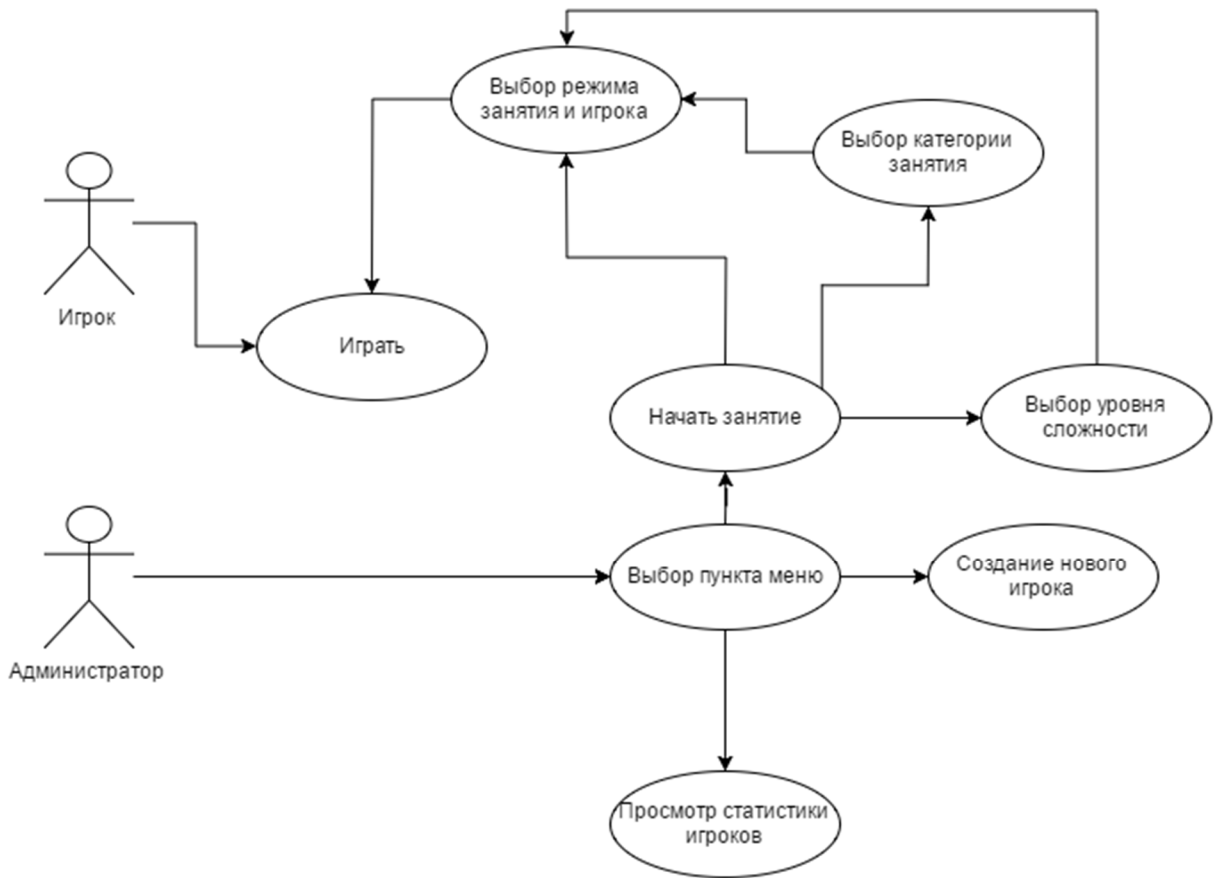


Рис. 2.2 – Диаграмма использования

Можно выделить следующие варианты использования системы:

- выбор пункта меню – пользователь может выбрать один из пунктов меню;
- просмотр статистики игроков – пользователю предоставляется полная статистика по каждому из игроков;
- создание нового игрока – пользователю предоставляется возможность создать нового игрока по ФИО;
- начать занятие – пользователь может начать занятие с выбором ФИО игрока;
- выбор режима занятия – пользователь выбирает режим занятия между двумя представленными;

Изм	Лист	№ докум.	Подпись	Дата

- выбор уровня сложности и категории – в зависимости от уровня сложности и категории будут подбираться соответствующие задания;
- играть – запуск заданий.

2.2 Архитектура приложения

2.2.1 Создание базы данных

Схема базы данных для приложения по подготовке детей старшего дошкольного возраста к школе представлена на рис. 2.3.

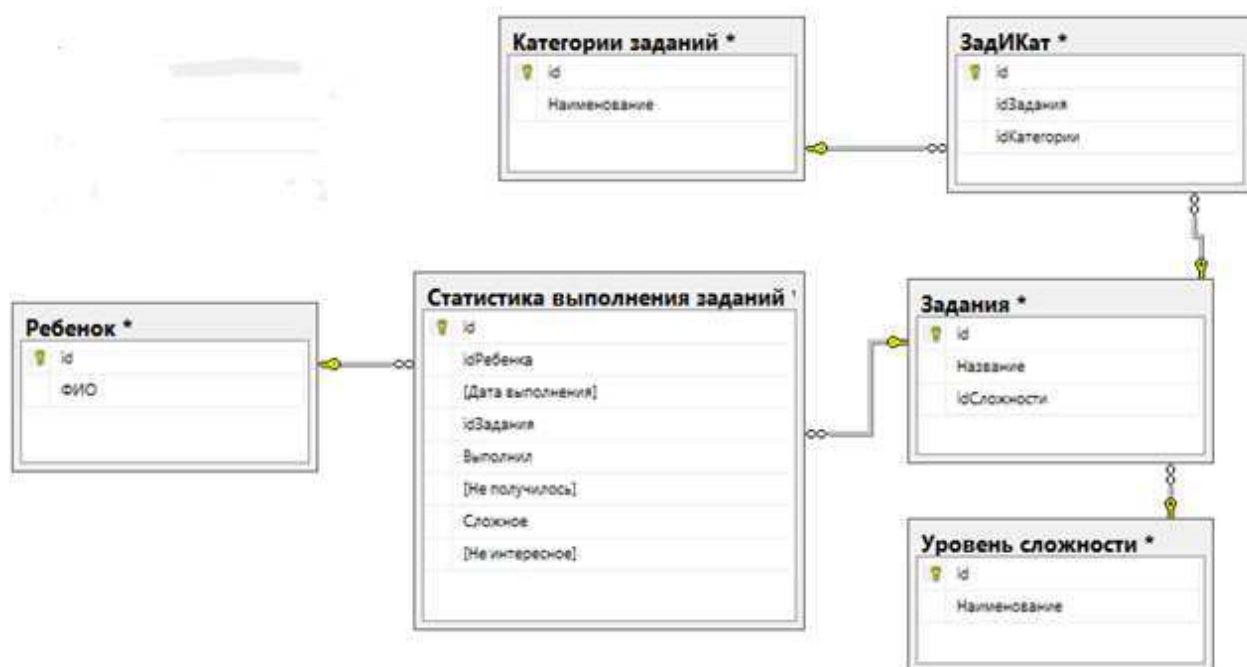


Рис. 2.3 – Схема базы данных приложения по подготовке детей к школе

В таблице 2.1 описано назначение каждой таблицы базы данных.

Таблица 2.1 – Описание таблиц базы данных

Наименование таблицы	Назначение
Задания	Список всех доступных заданий

ЗадИКат	Справочник для соответствующей сущности, позволяет избежать избыточности и аномалии обновлений
Категории заданий	Категории заданий
Ребенок	Списокт всех доступных пользователей (детей)
Статистика выполнения заданий	Статистика выполнения заданий по каждому пользователю
Уровень сложности	Уровни сложности заданий

Таблица «Задание» включает следующие поля (Рисунок 2.5):

- id – первичный ключ БД;
- Название – атрибут, служащий для хранения названия каждого задания;
- idСложности – внешний ключ, служит для связи с таблицей «Уровень сложности».

Name	Тип	Not Null	Default Value	Primary Key
id	INTEGER	0	null	1
Название	TEXT	0	null	0
idСложности	INTEGER	0	null	0

Рисунок 2.5 – Проект таблицы «Задание»

Таблица «ЗадИКат» включает следующие поля (Рисунок 2.6):

- id – первичный ключ БД;
- idЗадания – внешний ключ, служит для связи с таблицей «Задания»;
- idКатегории – внешний ключ, служит для связи с таблицей «Категории заданий».

Name	Type	Not Null	Default Value	Primary Key
id	INTEGER	1	null	1
idЗадания	INTEGER	1	null	0
idКатегории	INTEGER	0	null	0

Рисунок 2.6 – Проект таблицы «ЗадИКат»

Таблица «Категории заданий» включает следующие поля (Рисунок 2.7):

- id – первичный ключ БД;
- Наименование – атрибут, служащий для хранения названия категории заданий.

Name	Type	Not Null	Default Value	Primary Key
id	INTEGER	1	null	1
Наименование	TEXT	1	null	0

Рисунок 2.7 – Проект таблицы «Категории заданий»

Таблица «Ребенок» включает следующие поля (Рисунок 2.8):

- id – первичный ключ БД;
- ФИО – атрибут, служащий для хранения ФИО пользователя.

Name	Type	Not Null	Default Value	Primary Key
id	INTEGER	1	null	1
ФИО	TEXT	1	null	0

Рисунок 2.8 – Проект таблицы «Ребенок»

Таблица «Статистика выполнения заданий» включает следующие поля (Рисунок 2.9):

- id – первичный ключ БД;
- idРебенка – внешний ключ, служит для связи с таблицей «Ребенок»;
- Дата выполнения – атрибут, служащий для хранения даты выполнения задания;
- idЗадания – внешний ключ, служит для связи с таблицей «Задания»;

- Выполнил – атрибут, служащий для хранения информации о том, смог ли пользователь выполнить задание;
- Не получилось – атрибут, служащий для хранения информации о том, что не получилось у пользователя при выполнении задания;
- Сложное – атрибут, служащий для хранения информации о том, что задание было сложным для пользователя;
- Не интересное – атрибут, служащий для хранения информации о том, было ли интересно задание пользователю.

Name	Тип	Not Null	Default Value	Primary Key
id	INTEGER	1	null	1
idРебенка	INTEGER	1	null	0
Дата выполнения	DATETIME	1	null	0
idЗадания	INTEGER	1	null	0
Выполнил	BOOLEAN	1	null	0
Не получилось	TEXT	0	null	0
Сложное	BOOLEAN	1	null	0
Не интересное	BOOLEAN	1	null	0

Рисунок 2.9 – Проект таблицы «Статистика выполнения заданий»

Таблица «Уровень сложности» включает следующие поля (Рисунок 2.10):

- id – первичный ключ БД;
- Наименование – атрибут, служащий для хранения наименования уровня сложности задания.

Name	Тип	Not Null	Default Value	Primary Key
id	INTEGER	1	null	1
Наименование	TEXT	1	null	0

Рисунок 2.10 – Проект таблицы «Уровень сложности»

2.2.2 Реализация приложения

В данном разделе будут рассмотрены основные компоненты, на которых базируется приложение. Подробнее будет расписано большинство классов, функций и методов.

Движок Unity содержит в себе систему разбиения игровых уровней на так называемые сцены. В основном, сцены представляют собой отдельно описываемый компонент, состоящий из своего набора классов, объектов, изображений, звуков, анимаций и т.д. В приложении сцены будут использоваться для создания модульной системы, то есть каждое из заданий будет являться отдельной сценой. Также отдельно будет представлена сцена с основным главным меню.

На рисунке 2.11 представлен редактор сцен в Unity.

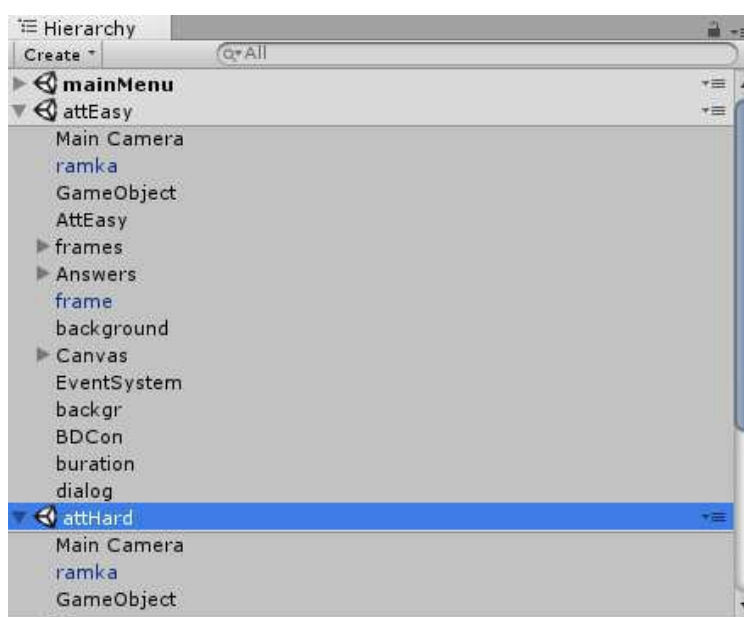


Рисунок 2.11 – Редактор сцен Unity

Список присутствующих сцен в приложении:

- mainmenu – сцена, на которой расположено главное меню и меню администратора;
- attEasy, attHard, logEasy, logHard, mathEasy, mathHard, memEasy, memHard, speEasy, speHard – сцены-модули, каждый из которых отвечает за отдельно взятое задание.

Для каждой из сцен заранее подготовлен свой набор используемых компонентов и классов. Некоторые из сцен используют практически одинаковый набор классов, незначительно отличающихся между собой реализацией того или иного задания.

Сцены attEasy, attHard, logEasy, logHard, mathEasy, mathHard, memEasy, memHard, speEasy, speHard представляют собой модули, которые разделены друг с другом. В основном, практически все из них имеют одинаковую архитектуру, различаясь лишь в некоторых мелочах.

На рисунке 2.12 показана диаграмма, демонстрирующая классы перечисленных выше сцен.

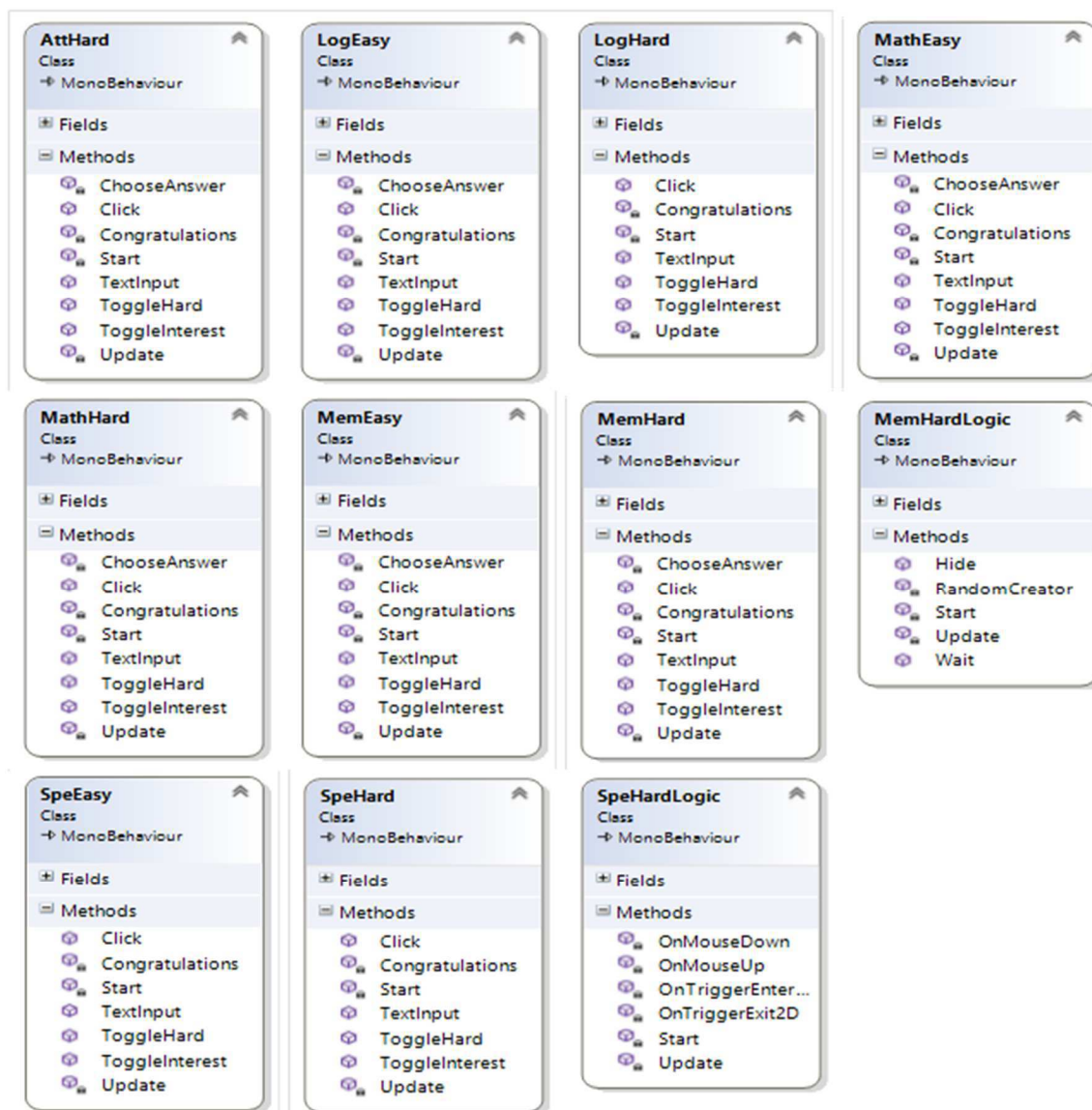


Рисунок 2.12 – Диаграмма классов сцен-модулей

Ввиду громоздкости и практически схожей архитектуры, будет представлен только один класс `class public AttHard`, его методы и атрибуты (рисунок 2.13):

- `Transform answers` – переменная типа `Transform`, хранит правильный ответ задания;
- `Transform BDcon` – переменная типа `Transform`, необходима для соединения с базой данных;
- `Bool completeTask` – логическая переменная, проверяет завершение задания;
- `Transform dialog` – переменная типа `Transform`, отвечает за объект с «диалогом» (вывод правильности выбранного ответа);
- `Text falseAnswer` – текстовое поле, выводит сообщение о неправильном ответе;
- `RaycastHit2D hit` – переменная, которая отвечает за 2D луч, считывающий, куда кликнул пользователь;
- `Bool isHard` – логическая переменная, проверяет нажатие пользователем галочки «Сложное»;
- `Bool notIntrested` – логическая переменная, проверяет нажатие пользователем галочки «Не интересное»;
- `Int rightAnswer` – идентификатор количества правильных ответов;
- `String textOnDontHappend` – строковое поле для хранения того, что не получилось у пользователя;
- `Text trueAnswer` – текстовое поле, выводит сообщение о правильном ответе;
- `ChooseAnswer()` – метод, осуществляет вывод о правильности выбранного ответа в задании;

					09.03.01.2017.266.00 ПЗ	Лист
						38
Изм	Лист	№ докум.	Подпись	Дата		

- Click() – метод, связанный с кнопкой для перехода на следующий вопрос, передает в класс DBConnection информацию о статистике выполнения задания;
- Congratulation() – метод, осуществляющий вывод правильности/неправильности выбранного ответа на задание;
- Start() – метод, задает начальные параметры для многих объектов;
- TextInput() – метод, связанный с текстовым полем «Не получилось», в которое пользователь может вписать, что у него не получилось;
- ToggleHard() – метод, связанный с переключателем, на который пользователь нажимает в зависимости от того, сложное задание для него или нет;
- ToggleInterest() – метод, связанный с переключателем, на который пользователь нажимает в зависимости от того, интересное задание для него или нет;
- Update() – метод, отвечающий за считывание нажатия клавиши мыши по объекту;

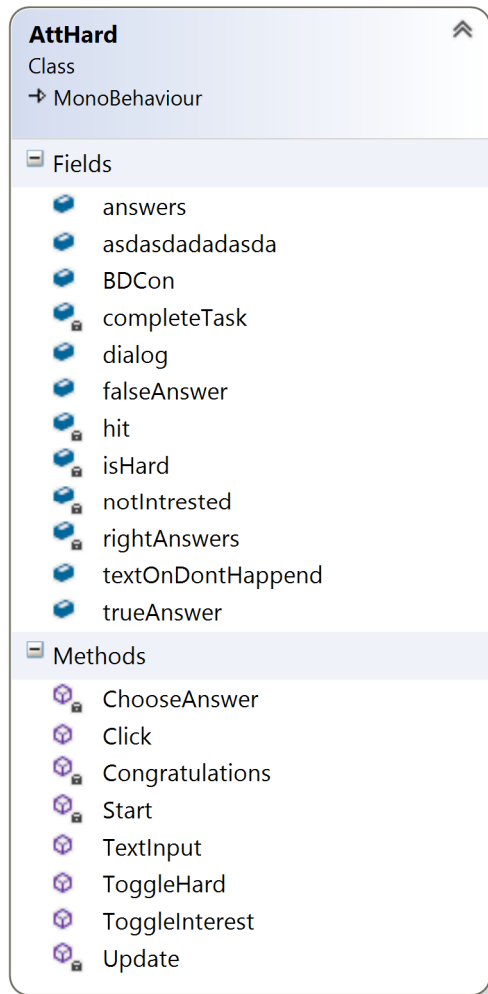


Рисунок 2.13 – Класс AttHard

Реализация сцены mainmenu содержит в себе один класс, который представлен на рис. 2.14.

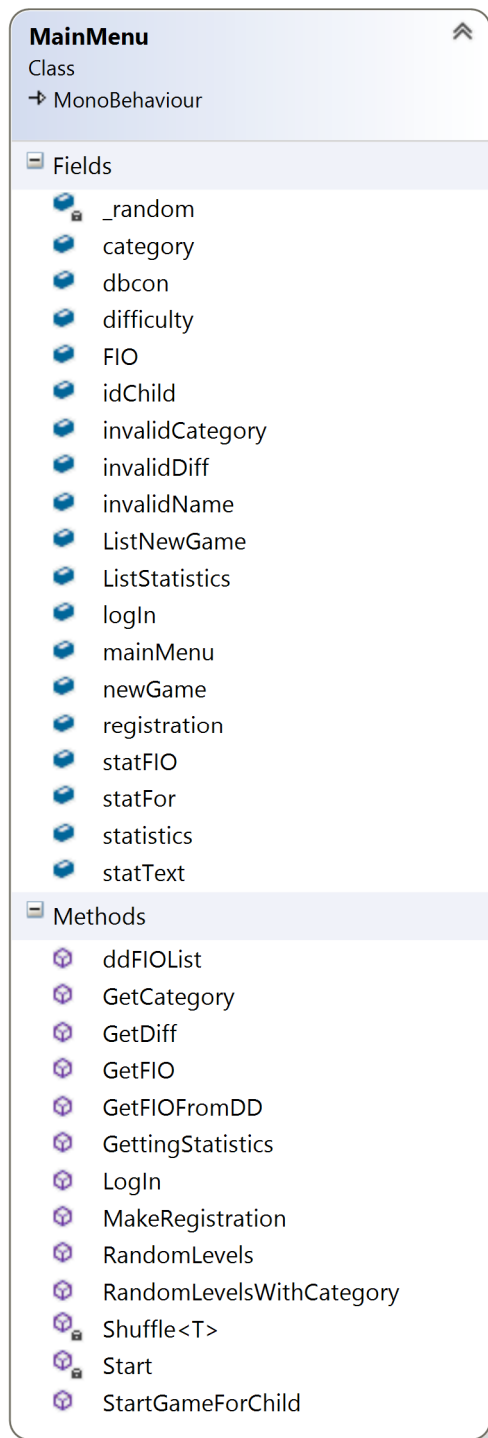


Рисунок 2.14 – Класс MainMenu

Класс `public class MainMenu` включает в себя атрибуты и методы:

- Random `_random` – выбор случайных чисел;
- Int `category` – категория заданий (внимание, память, математика, речь, логика);
- Transform `dbcon` – объект для связи с базой данных;

- Int difficulty – сложность заданий (легко, сложно);
- String FIO – строковое поле для ФИО пользователя;
- Int idChild – идентификатор пользователя (ребенка), используется для выбора ФИО ребенка в списке;
- Text invalidCategory – текстовое поле для вывода ошибки о том, что не выбрана категория;
- Text[] invalidDiff – массив текстового поля для вывода ошибки о том, что не выбран уровень сложности;
- Text[] invalidName – массив текстового поля для вывода ошибки о том, что не выбрано ФИО пользователя;
- Dropdown ListNewGame – выпадающий список пользователей;
- Dropdown ListStatistics – выпадающий список пользователей для статистики;
- GameObject logIn – поле типа GameObject для кнопки входа;
- GameObject mainMenu – поле типа GameObject для объекта главного меню;
- GameObject newGame – поле типа GameObject для объекта начала нового занятия;
- GameObject registration – поле типа GameObject для объекта регистрации нового пользователя;
- GameObject statFIO – поле типа GameObject для объекта просмотра статистики;
- Text statFor- текстовое поле для вывода ФИО в объекте статистики;
- Text statText- текстовое поле для статистики пользователя;
- ddFIOList() – метод, который выводит всех пользователей в выпадающем списке;
- GetCategory() – метод, который выводит все категории в выпадающем списке;

- GetDiff() – метод, который выводит все уровни сложности в выпадающем списке;
- GetFIOFromDD() – метод, который выводит всех пользователей в выпадающем списке статистики;
- GettingStatistics() – метод, который получает из базы и выводит статистику для конкретного пользователя;
- LogIn() – метод, необходимый для авторизации пользователя при старте нового занятия;
- MakeRegistration() – метод, необходимый для регистрации нового пользователя;
- RandomLevels() – метод, необходимый для выбора случайных заданий в занятии;
- RandomLevelsWithCategory() – метод, необходимый для выбора случайных заданий из определенной категории;
- Shuffle<T>() – метод, который перемешивает массив с заданиями;
- Start() – метод, в котором прописываются атрибуты начального значения объектов;
- StartGameForChild() – метод, который запускает игру без выбора всех параметров.

Также во всех сценах присутствует класс DBConnection, который отвечает за соединение с базой данных и за все SQL запросы от остальных классов. Класс public class DBConnection включает в себя атрибуты и методы (рисунок 2.15):

- Bool authSuccess – логическая переменная, проверяет правильность ввода пароля администратора для показа подсказки на экране;
- ArrayList childList – список всех пользователей из базы данных;

- Bool completeTask – логическая переменная, необходима для отправки запроса о выполнении задания в таблицу базы данных «Статистика выполнения заданий»;
- Int idTask – необходима для отправки запроса идентификатора задания в таблицу базы данных «Статистика выполнения заданий»;
- Bool IsHard – логическая переменная, необходима для отправки запроса о том сложности задания в таблицу базы данных «Статистика выполнения заданий»;
- String nePoluchilos – строковое поле, необходимо для отправки запроса о том, что не получилось у пользователя при выполнении задания в таблицу базы данных «Статистика выполнения заданий»;
- Bool notInterested – логическая переменная, необходимая для отправки запроса о том, интересно ли задание для пользователя в таблицу базы данных «Статистика выполнения заданий»;
- Bool TaskMaked – логическая переменная, отвечает за то, будет ли отправлена статистика в базу данных;
- String tempCon – строковое поле, хранит строку подключения к базе данных;
- Auth() – метод, который отвечает за проверку вводимого пароля для администратора;
- GetLevelsWithSomeCategory() – метод, который делает запрос к базе данных для выборки заданий по категориям;
- GetLevelsWithSomeDifficulty() – метод, который делает запрос к базе данных для выборки заданий по уровню сложности;
- GetStatistics() – метод, который делает запрос к базе данных для получения статистики;
- InsertChild() – метод, отвечает за добавление нового пользователя в базу данных;

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		44

- `makeChildList()` – метод, отвечает за получение всех пользователей из базы данных;
- `Start()` – метод, отвечает за подключение к базе данных;
- `TaskMaked()` – метод, отвечает добавление в статистику новой записи в базе данных;
- `Update()` – метод, который несколько раз в секунду проверяет, была ли добавлена запись о завершении задания в базу данных;

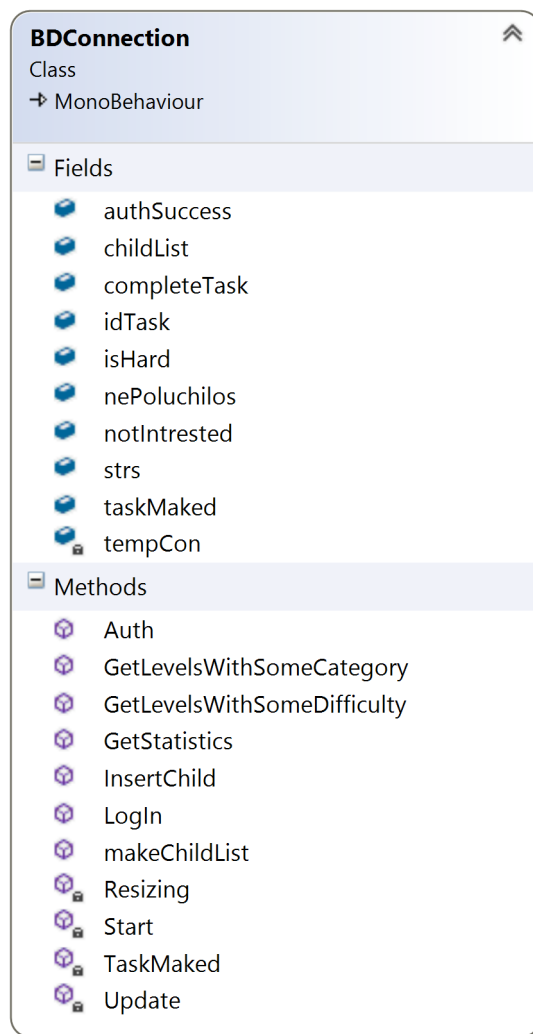


Рисунок 2.15 – Класс DBConnection

Ниже будут рассмотрены ключевые функции класса DBConnection.

1) `Start`.

Метод `Start`, исходя из логики Unity, выполняется сразу после старта приложения. Содержит в себе строку подключения к базе данных, к

которой обращаются все остальные методы класса DBConnection. Листинг метода представлен на рисунке 2.16.

```
void Start()
{
    string connectionString = "URI=file:" + UnityEngine.Application.dataPath
                             + "/Db/TestingDB.sqlite";
    tempCon = connectionString;
}
```

Рисунок 2.16 – Листинг метода Start

2) InsertChild.

Функция выполняет простой SQL-запрос на добавление новой записи в таблицу базы данных «Ребенок», таким образом регистрируется новый пользователь для работы с приложением. Листинг метода представлен на рисунке 2.17.

```
public void InsertChild(string FIO)
{
    using (IDbConnection dbcon = (IDbConnection)new SqliteConnection(tempCon))
    {
        dbcon.Open();
        var sql = "INSERT INTO Ребенок VALUES ((SELECT MAX(id) from Ребенок)+1, '" + FIO + "')";
        using (IDbCommand dbcmd = dbcon.CreateCommand())
        {
            dbcmd.CommandText = sql;
        }
    }
}
```

Рисунок 2.17 – Листинг метода InsertChild

3) GetStatistics.

Помимо добавления в базу данных новых записей, необходимо выбирать из нее уже существующие записи. Данная функция создает SQL-запрос и в дальнейшем формирует строковое поле для вывода на экран статистики выполнения заданий. Листинг метода представлен на рисунке 2.18.

```

public void GetStatistics(string FIO)
{
    using (IDbConnection dbcon = (IDbConnection)new SqliteConnection(tempCon))
    {
        int i = 0;
        dbcon.Open();
        string[] strArray = new string[1000];
        var sql = "SELECT "
            + "ст.[Дата выполнения], "
            + "зад.Название, "
            + "ур.Наименование, "
            + "кат.Наименование, "
            + "ст.Выполнил, "
            + "ст.Сложное, "
            + "ст.[Не интересное], "
            + "ст.[Не получилось] "
            + "FROM 'Статистика выполнения заданий' ст "
            + "JOIN Ребенок реб ON реб.id = ст.idРебенка "
            + "JOIN Задания зад ON зад.id = ст.idЗадания "
            + "JOIN 'Уровень сложности' ур ON ур.id = зад.idСложности "
            + "JOIN 'ЗадКат' зк ON зк.idЗадания = зад.id "
            + "JOIN 'Категории заданий' кат ON кат.id = зк.idКатегории "
            + "WHERE реб.ФИО = '" + FIO + "'";
        using (IDbCommand dbcmd = dbcon.CreateCommand())
        {
            dbcmd.CommandText = sql;
            using (IDataReader reader = dbcmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    const string frmt = "Дата:{0}, Задание:{1}, Сложность:{2}, Категория:{3},
                    Выполнено:{4}, Сложное:{5}, Не интересное:{6}, Не получилось:{7};\n";
                    strArray[i] = string.Format(frmt, reader.GetDateTime(0), reader.GetString(1),
                    reader.GetString(2), reader.GetString(3), reader.GetBoolean(4),
                    reader.GetBoolean(5), reader.GetBoolean(6), reader.GetString(7));
                    i++;
                }
            }
            dbcon.Close();
        }
    }
}

```

Рисунок 2.18 – Листинг метода GetStatistics

4) GetLevelsWithSomeCategory.

Для генерации заданий необходимо делать SQL-запрос в базу данных из соответствующей таблицы. Данный метод содержит реализацию необходимого запроса, а также выбирает задания исключительно по выбранному пользователем уровню сложности. Листинг метода представлен на рисунке 2.19.

```

public void GetLevelsWithSomeCategory(int category, int difficulty)
{
    using (IDbConnection dbcon = (IDbConnection)new SQLiteConnection(tempCon))
    {
        dbcon.Open();
        int i = 0;
        int temp;
        var sql = "";
        if (difficulty == 1 || difficulty == 2)
            sql = "SELECT зад.ид FROM Задания зад JOIN 'ЗадКат' эк ON эк.идЗадания = зад.ид
                JOIN 'Категории заданий' кат ON кат.ид = эк.идКатегории
                JOIN 'Уровень сложности' ур ON ур.ид = зад.идСложности
                WHERE кат.ид = " + category + " AND ур.ид = " + difficulty;
        else if (difficulty == 3)
            sql = "SELECT зад.ид FROM Задания зад JOIN 'ЗадКат' эк ON эк.идЗадания = зад.ид
                JOIN 'Категории заданий' кат ON кат.ид = эк.идКатегории WHERE кат.ид = " + category;
        using (IDbCommand dbcmd = dbcon.CreateCommand())
        {
            dbcmd.CommandText = sql;
            using (IDataReader reader = dbcmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    temp = reader.GetInt32(0);
                    Global.levelsWithCategory[i] = temp;
                    i++;
                }
            }
        }
    }
}

```

Рисунок 2.19 – Листинг метода GetLevelsWithSomeCategory

5) TaskMaked.

Для генерации заданий необходимо делать SQL-запрос в базу данных для соответствующей таблицы. Данный метод содержит реализацию необходимого запроса, а также выбирает задания исключительно по выбранному пользователем уровню сложности.

Вывод по разделу: в данной главе были рассмотрены требования с точки зрения IT-специалиста, и на основании этих требований была разработана база данных архитектура и реализация приложения.

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		48

3 ПРИМЕРЫ РАБОТЫ

Главное меню приложения состоит из двух пунктов: «Играть!» и «Администратор». Пункт меню «Играть!» необходим для запуска заданий без выбора настроек, пункт «Администратор» предоставляет возможность пользователю выбрать задания по критериям.

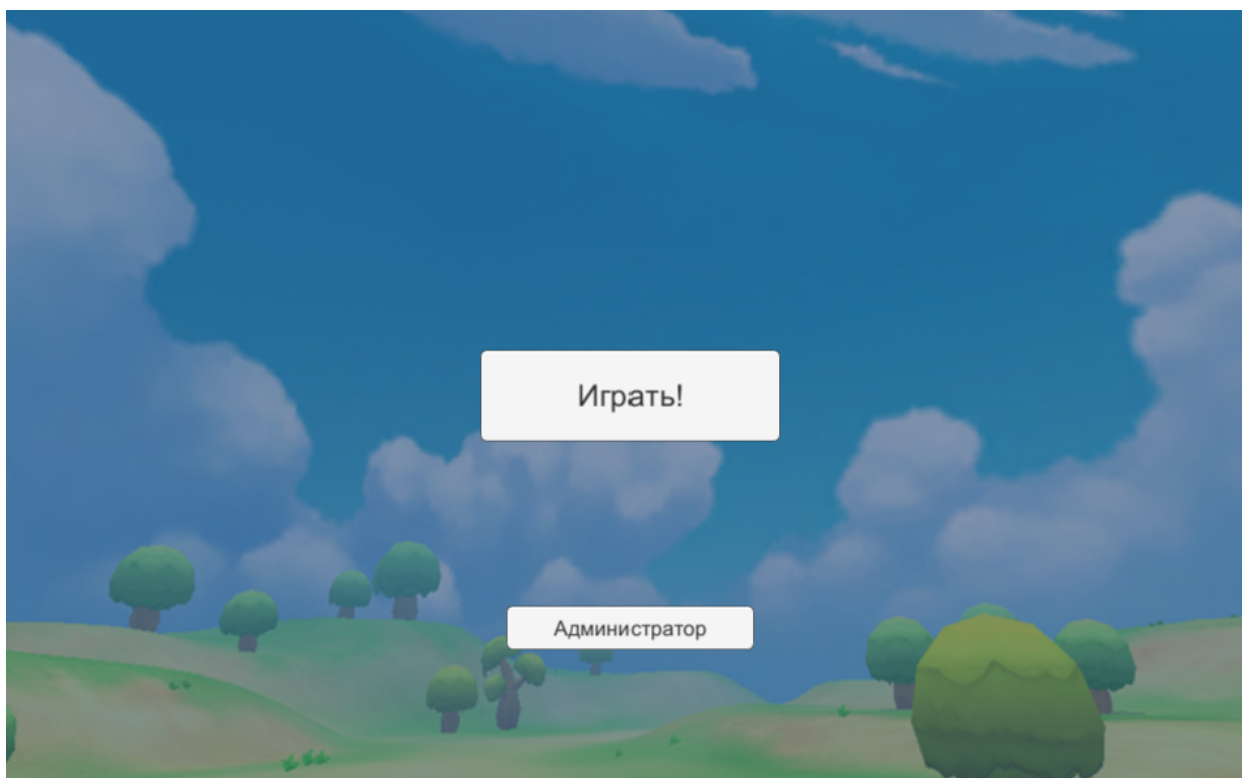


Рисунок 3.1 Главное меню приложения

На рисунке 3.2 представлено меню администратора приложения, с которым работает пользователь. При переходе из главного меню, а также после завершения занятия появляется данное окно, из которого можно выбрать один из пунктов меню: «Начать занятие», «Новый игрок» и «Статистика».

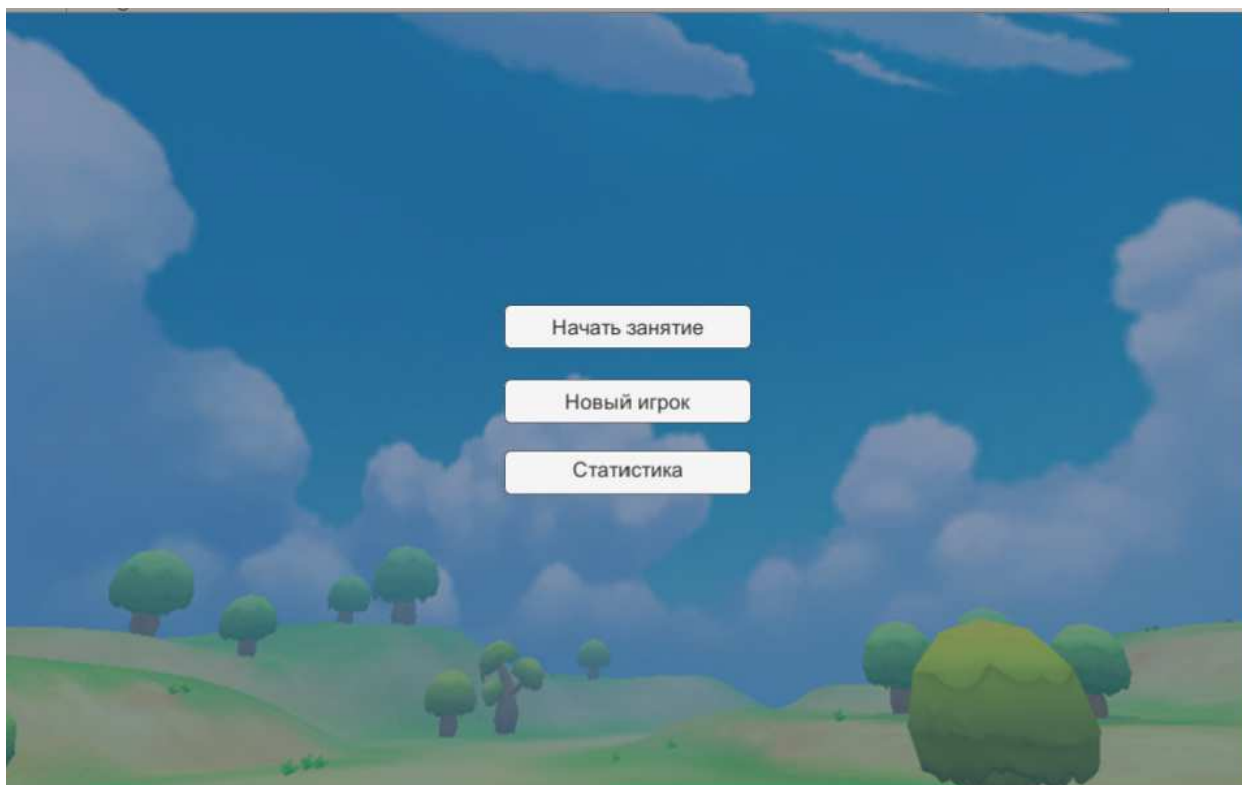


Рисунок 3.2 Меню в режиме администратора

Из выпадающего списка на рисунке 3.3 можно выбрать необходимого пользователя. Данное окно появляется как при начале занятия, так и при показе статистики.

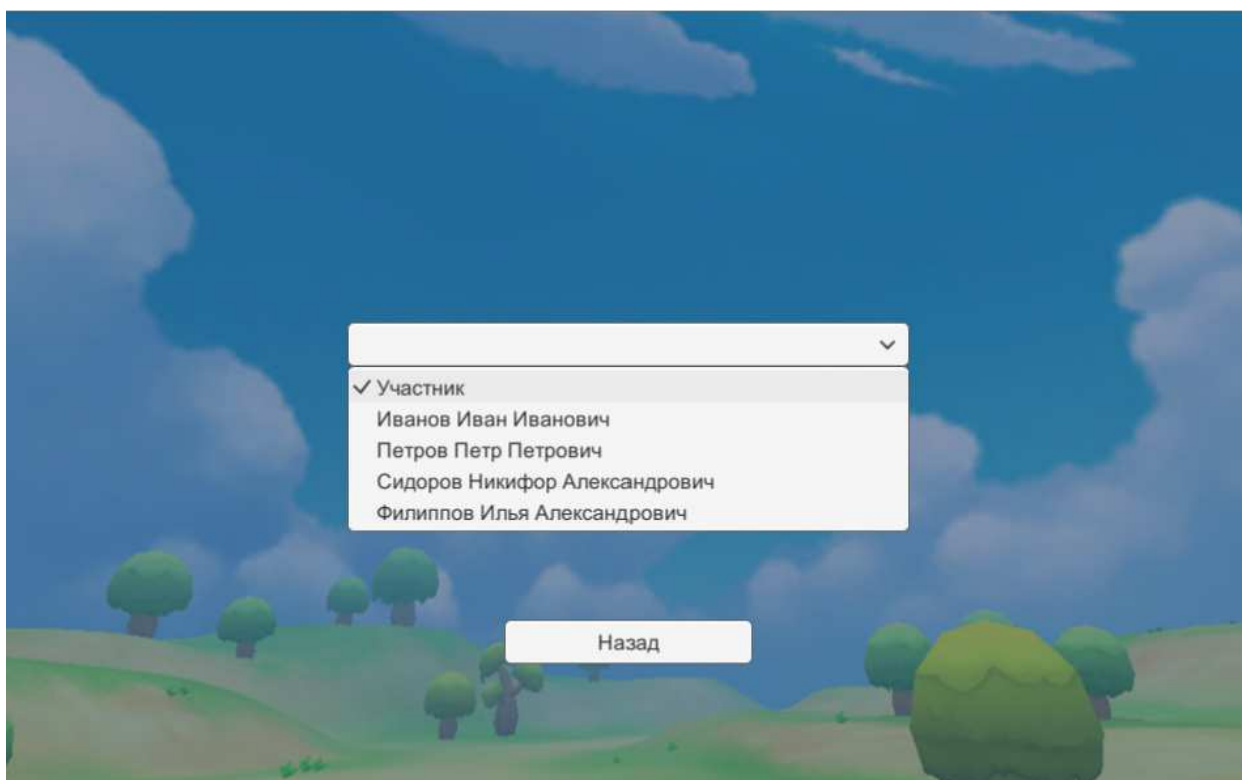


Рисунок 3.3 Выбор пользователя для начала занятия и показа статистики

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		50

На рисунке 3.4 представлено окно регистрации нового пользователя.

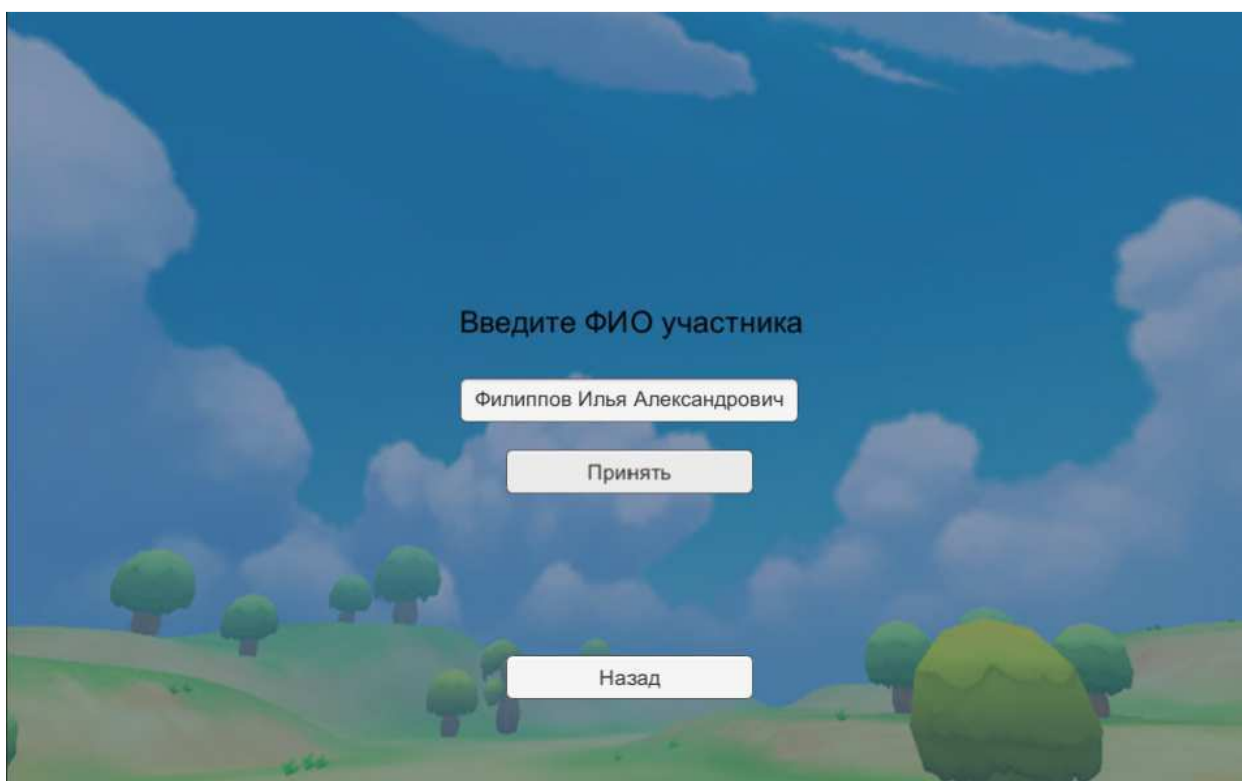


Рисунок 3.4 Регистрация нового пользователя

На рисунке 3.5 продемонстрирован выбор уровня сложности занятия. Также, при выборе другого режима занятий (на развитие только одной категории навыков), будет добавлен еще один выпадающий список с выбором категории заданий.

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		51

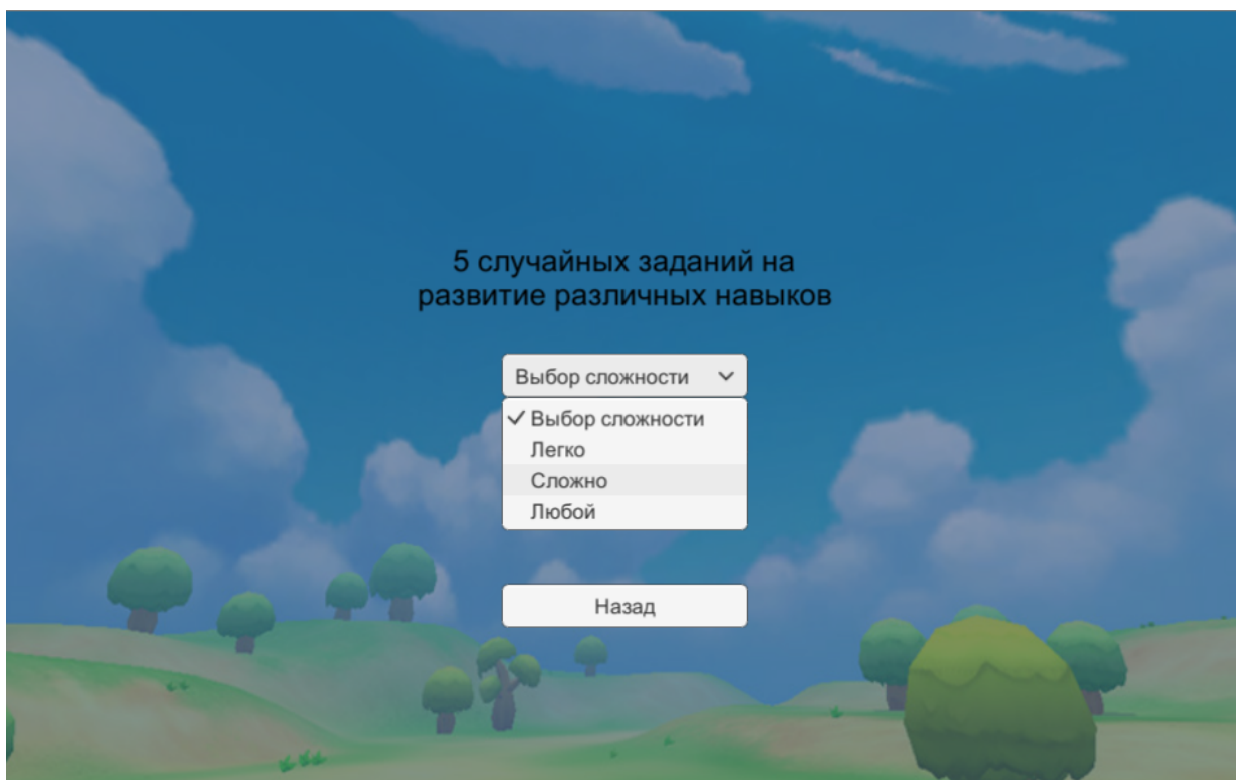


Рисунок 3.5 Выбор уровня сложности занятия

Статистика для определенного пользователя представлена на рисунке 3.6. Она поделена на несколько строк, каждая строка отвечает за одно задание. В таблице указаны все пункты базы данных из таблицы «Статистика выполнения заданий».

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		52

Статистика для: Филиппов Илья Александрович

Дата	Задание	Сложн.	Категор.	Выполн.	Сложное	Не интер	Не полу
6/6/2017	attEasy	Легкое	Внимание	Да	Нет	Нет	
6/6/2017	attHard	Сложное	Внимание	Нет	Да	Нет	трудно

Назад

Рисунок 3.6 Статистика пользователя

Занятия делятся по уровню сложности и категориям, на рисунке 3.7 представлен пример задания из категории «Память» с уровнем сложности «Легкое». Пользователю предоставляется возможность запомнить 4 предложенные картинки, которые пропадают на несколько секунд и появляются вновь, кроме одной. Варианты правильного ответа представлены ниже. Если пользователь выбирает неправильный вариант, Буратино говорит «Не правильно», и наоборот, если правильно.

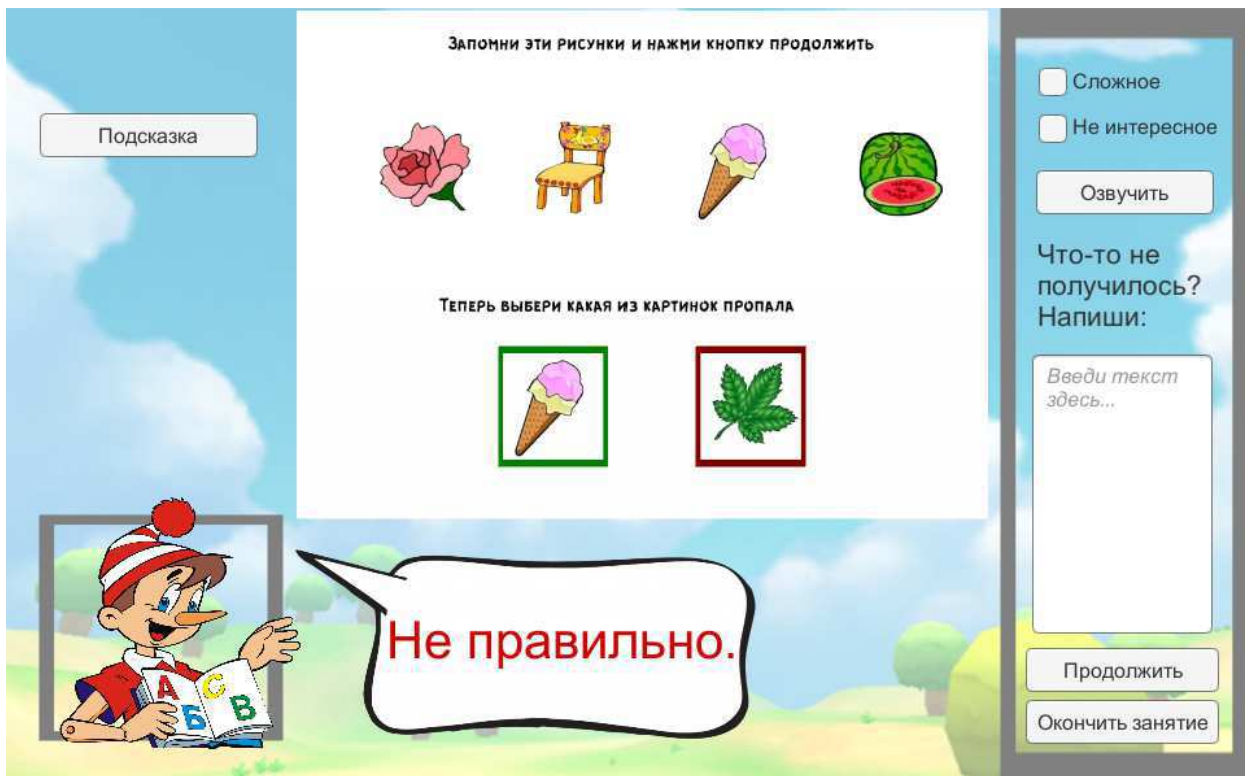


Рисунок 3.7 Пример задания уровня сложности «Легкое»

Пример задания уровня сложности «Сложное» представлен на рисунке 3.8.

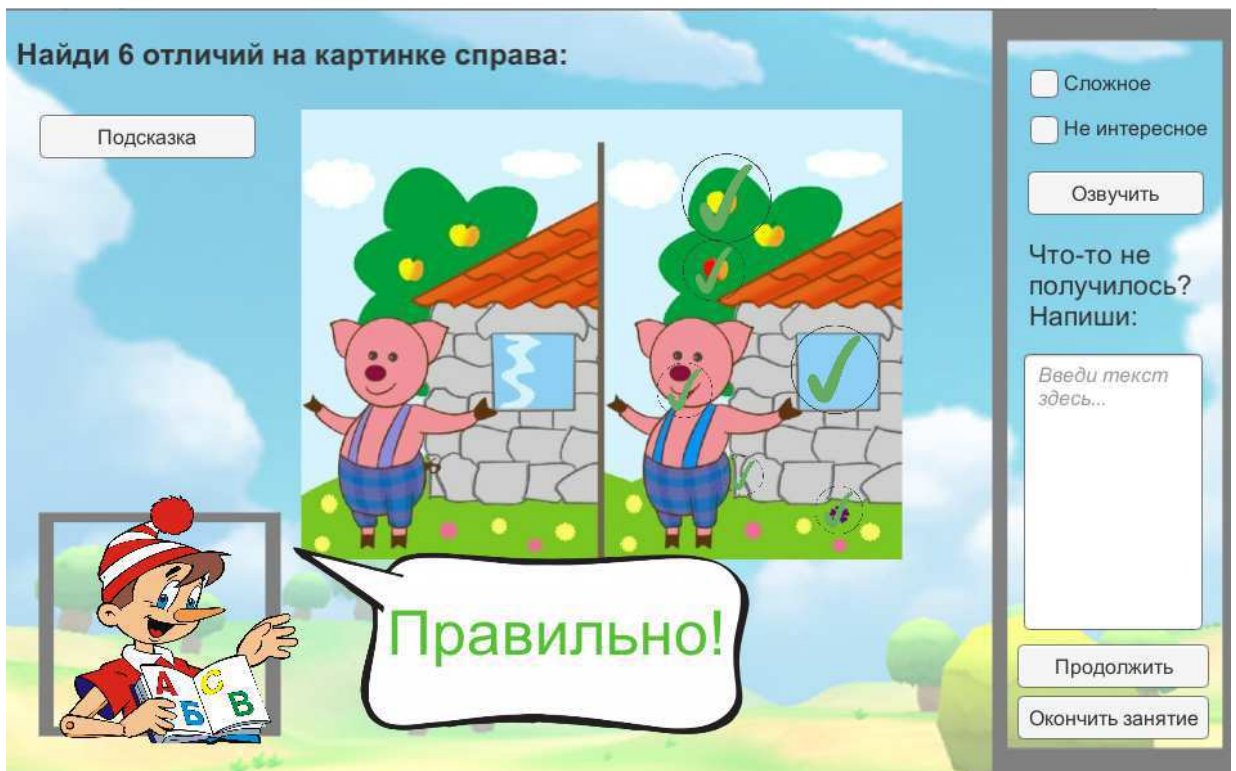


Рисунок 3.8 Пример задания уровня сложности «Сложное»

Изм	Лист	№ докум.	Подпись	Дата

09.03.01.2017.266.00 ПЗ

Лист

54

Помимо заданий на нажатие по правильному ответу, есть задания на перетаскивание игровых объектов (рисунок 3.9).

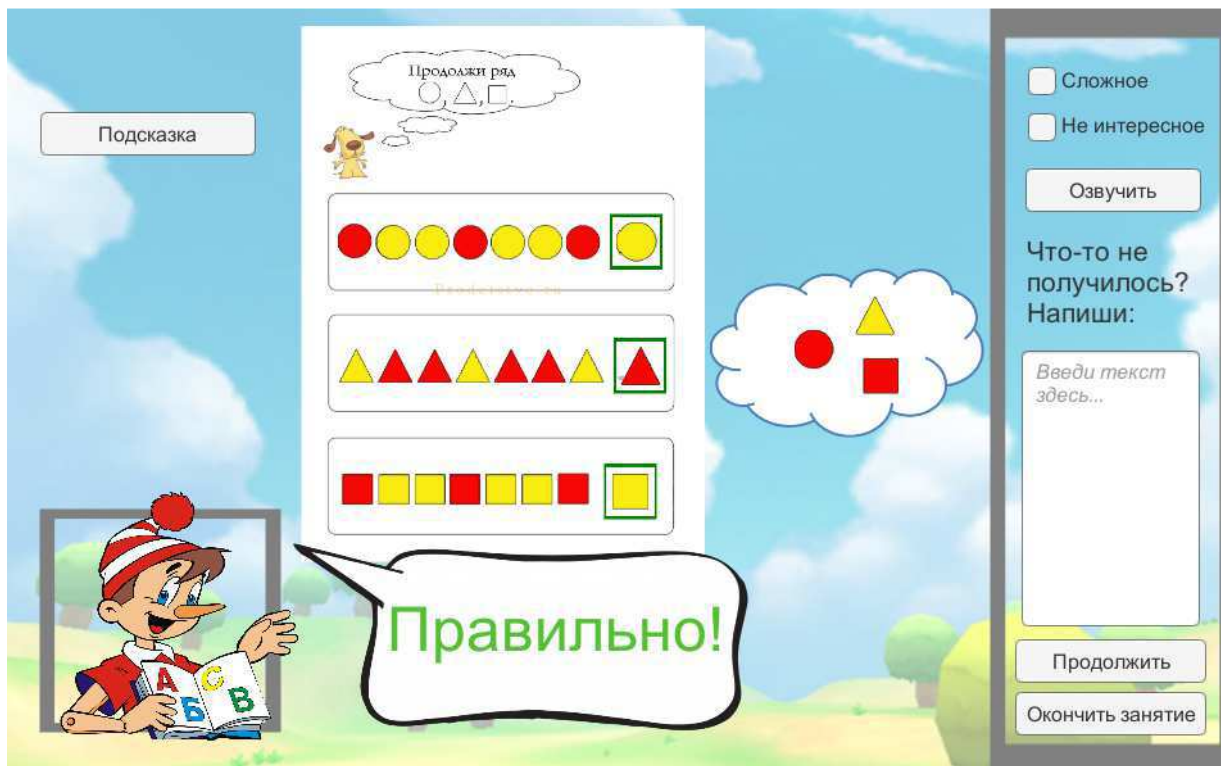


Рисунок 3.9 Пример задания на перетаскивание объекта

Вывод по разделу: в данном разделе были представлены примеры работы приложения.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы было создано приложение для подготовки детей старшего дошкольного возраста к школе.

Для достижения цели были поставлены и решены следующие задачи:

1. Выявлены цели и задачи создания системы;
2. Сформулированы и структурированы требования к системе со стороны заказчика, проведен их анализ;
3. Проведен обзор родственных разработок, выявлены их плюсы и минусы, а так же выполнено сравнение с разрабатываемым приложением;
4. Был проведен поиск средств реализации и на основе их анализа выполнен выбор необходимого движка для разработки;
5. Было составлено описание проектного решения с точки зрения it-специалиста, построена диаграмма классов, диаграмма использований, схема связи меню приложения;
6. Разработана архитектура системы, выбраны средства ее реализации;
7. Спроектирована и создана база данных;
8. Разработаны модули и компоненты системы.
9. Проведено тестирование разработанных программных средств.

На данный момент приложение готово к тестовому внедрению в компьютерные классы детских садов. Перспективы развития приложения:

- разработка приложения под мобильные платформы;
- расширение существующей базы заданий;
- добавление модулей для развития детей младшего школьного

возраста.

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		56

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Диго, С.М. Базы данных: проектирование и использование. Учебное пособие для вузов / С.М. Диго. – Москва: Финансы и статистика, 2005. – 171 с.

2 Общее описание языка С# –
http://inflib.ru/index.php?option=com_content&task=view&id=2&Itemid=2

3 Обучающая компьютерная игра Тренажер мозга –
<http://detskieradosti.ru/load/25-1-0-8144>

4 Раннее развитие. Логика и память – <http://newprogs.net/news/2012-02-04-973>

5 Руководство Unity –
<https://docs.unity3d.com/ru/current/Manual/index.html>

6 Райордан, Р. Основы реляционных баз данных / Р. Райордан. Пер. с англ. – М.:Издательско-торговый дом «Русская редакция», 2001. – 384 с.:ил. ISBN 5-7502-0150-3

7 Скоро в школу. Сказочный учебник –
<http://nd.ru/catalog/products/schoolsoonfairytetbook/>

8 Троелсен, Эндрю Язык программирования С# 5.0 и платформа .NET 4.5. 6-е изд. / Эндрю Троелсон. Пер. с англ. – М.: ООО “И.Д. Вильямс”, 2013. – 1312 с.

9 Что такое Unity, движок Unity – описание на русском –
<https://itkeys.org/about-unity/>

10 GameMaker – YoYo Games –
<https://www.yoyogames.com/gamemaker>

11 What is Unreal Engine 4 – <https://www.unrealengine.com/what-is-unreal-engine-4>

					09.03.01.2017.266.00 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		57