**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное автономное образовательное   
учреждение высшего образования   
«Санкт-Петербургский политехнический университет Петра Великого»**

**ИНСТИТУТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

|  |  |
| --- | --- |
|  | **Допустить к защите**  Заместитель директора по  учебно-методической работе  \_\_\_\_\_\_\_\_\_\_\_ \_\_Е. Г. Конакина\_\_\_\_\_  (Подпись) (И.О.Ф.)  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |
|  |

**ДИПЛОМНЫЙ ПРОЕКТ**

Тема Разработка обучающий игры «Математика для дошкольников»

специальность 09.02.07 группа 42919/5

Студент (ка) Рябов Н.А.

(подпись) (ФИО)

Руководитель Андреев В.А.

(подпись) (ФИО)

Санкт-Петербург

2023

**ЛИСТ ГОТОВНОСТИ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ  
К ЗАЩИТЕ**

Фамилия, имя, отчество студента (ки) Рябов Никита Александрович

Специальность 09.02.07 «Информационные системы и программирование»

Тема ВКР Разработка обучающий игры «Математика для дошкольников»

Замечаний нет:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Консультант** | **Фамилия, имя, отчество** | **Подпись** | **Дата** |
| 1 | Экономическая часть | Вимберг С.В. |  |  |
| 2 | Охрана труда и безопасность жизнедеятельности | Никитина Е.С. |  |  |
| 3 | Графическая часть | Крестинин Н.С. |  |  |
| 4 | Нормоконтроль | Крестинин Н.С. |  |  |
| 5 | Руководитель выпускной квалификационной работы | Андреев В.А. |  |  |

**ДОПУСТИТЬ К ЗАЩИТЕ:**

Председатель ПЦК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Андреев В.А. «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 202\_\_ г.

(Подпись) (ФИО)

Назначить защиту выпускной квалификационной работы на «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 202\_\_ г.

Заведующий отделением \_\_\_\_\_\_\_\_\_\_\_ / Сухорукова О. А. «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 202\_\_ г.

(Подпись) (ФИО)

**СОДЕРЖАНИЕ**

[Введение 4](#_Toc137193510)

[1 Общая часть 5](#_Toc137193511)

[1.1 Анализ предметной области 5](#_Toc137193512)

[1.2 Постановка задачи 5](#_Toc137193513)

[1.3 Функции проектируемого приложения 6](#_Toc137193514)

[1.4 Анализ рынка существующих решений 7](#_Toc137193515)

[1.5 Анализ методов решения 8](#_Toc137193516)

[1.6 Обоснование и выбор методики, технологии и инструментальных средств проектирования разработки 8](#_Toc137193517)

[2 Специальная часть 10](#_Toc137193520)

[2.1 Описание структуры приложения 10](#_Toc137193521)

[2.2 Объектно-ориентированное проектирование системы 10](#_Toc137193522)

[2.3 Разработка пользовательского интерфейса 11](#_Toc137193524)

[3 Экономическая часть 20](#_Toc137193526)

[3.1 Область применения программного продукта и его преимущества перед аналогичным программным продуктом 20](#_Toc137193527)

[3.2 Трудоемкость разработки программного продукта, квалификация исполнителя и его оклад 20](#_Toc137193528)

[3.3 Расчет затрат на разработку 23](#_Toc136891053)

[4 Техника безопасности и охрана труда 30](#_Toc137193529)

[4.1 Анализ условий труда разработчика игр 30](#_Toc137193530)

[4.2 Расчет искусственного освещения в помещении 31](#_Toc137193531)

[4.3 Электробезопасность на предприятии 33](#_Toc137193532)

[4.4 Пожарная безопасность на предприятии 35](#_Toc137193533)

[ЗАКЛЮЧЕНИЕ 37](#_Toc137193534)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 38](#_Toc137193535)

[Приложение А(справочное) Листинг кода 39](#_Toc137193536)

Введение

В рамках современного общества компьютерные и мобильные игры занимают не последнее место в жизни людей. Люди разных возрастов приобщаются к данной культуре. Сами игры могут нести различный характер, и совмещать в себе как развлекающий, так и развивающий контент.

Целью данной дипломной работы является разработка развивающей игры для детей в форме викторины, обеспечивающая проведения образовательного процесса в виде развлекательного контента.

В соответствии с целью определим следующие задачи дипломной работы:

* проанализировать предметную область приложения;
* описать задачи проекта;
* проанализировать задачи для разработки приложения;
* спроектировать интерфейс приложения;
* реализовать приложение.

Структура дипломной работы состоит из введения, четырех разделов, разделенных на подразделы, заключения, списка использованных источников и приложений.

# Общая часть

## Анализ предметной области

Приложение «Математика для дошкольников» предназначено для развития знаний и интеллектуальных способностей детей в сфере математики.

Приложение «Математика для дошкольников» имеет следующий функционал:

* просмотр главной страницы;
* выбор уровня;
* прохождение уровней;

Пользователями приложения является детская аудитория. В целом, игра викторина для детей является полезным и интересным способом развивать знания и навыки, а также проводить время с пользой и удовольствием.

## Постановка задачи

Сейчас человечество находится в эпоху цифровой трансформации, насыщенную технологиями, неожиданными открытиями, диктующую современные тренды управления. Это побуждает нас использовать самые современные цифровые технологии, быстро принимать решения, быть мобильными и гибкими.

Основное назначение разрабатываемого программного продукта заключается в разработке приложения с графическим интерфейсом «Математика для дошкольников»: пользователь заходит в игру, выбирает понравившийся уровень на определенную тематику.

Существует несколько оснований для разработки игры для детей:

– развитие когнитивных навыков. Данная игра может помочь развивать у детей когнитивные навыки, такие как память, логическое мышление, внимание и умение анализировать информацию.

– познавательный аспект. Данная игра ориентирована на образовательный процесс и используется для учебных целей. Игра может помочь детям запомнить цифры, их порядок, а так же научит простым математическим действиям, таким как сложение и вычитание.

Данный продукт должен эксплуатироваться на детских предприятиях и мобильных устройствах пользователей данной программы. Конечными пользователями являются клиенты, в частности дети, обучаемые в данном предприятии, пользователи, заинтересовавшиеся продуктом, и сотрудники предприятия.

## Функции проектируемого приложения

Программа должна быть интуитивно понятна и проста для пользователей, так как большинство из них будут являться детьми дошкольного возраста.

Для удержания интереса пользователей должны применяться элементы, которые позволят задействовать воображение ребенка, например озвучка различных уровней голосовым помощником.

Должны учитываться следующие функциональные требования для данного продукта:

1. приложение должно иметь главную страницу, с которой можно перейти на уровни;
2. приложение должно давать возможность выйти;
3. приложение должно иметь возможность генерировать вопросы к уровням случайным образом;
4. приложение должно иметь отдельные окна каждого уровня.

Нефункциональные требования были поставлены следующие:

1. приложение должно работать без зависаний;
2. доступ к уровням должен осуществляться не более чем за 2 нажатия на иконку;
3. программа должна открываться и запускаться менее чем за 20 секунд;
4. программа должна запускаться без сбоев и ошибок, должна обрабатывать исключения возникновения ошибок.

## Анализ рынка существующих решений

Рынок игр подобного характера довольно широк и разнообразен, включая множество игр на разных платформах. Ниже приведен анализ рынка игр:

1) тип платформы, математические игры викторины доступны на разных платформах, включая мобильные устройства, компьютеры, игровые консоли и интерактивные телевизоры. В последнее время мобильные игры становятся все более популярными, что отражается на рынке игр в.

2) целевая аудитория, математические игры викторины могут быть нацелены на разные возрастные группы и уровни сложности, от игр для детей до взрослых.

3) конкуренция, на рынке существует множество игр викторин, в том числе и известных брендов, таких как 1C-Publishing LLC, RV AppStudios. Однако существует также множество маленьких игровых разработчиков, которые создают свои собственные игры викторины.

4) модель монетизации, многие развивающие игры бесплатны, но могут содержать рекламу или микротранзакции. Также существуют платные игры, которые могут быть куплены в магазинах приложений или на игровых платформах.

5) инновации, существует множество новых игр, которые включают инновационные функции, такие как использование искусственного интеллекта, многопользовательские режимы и использование дополненной реальности.

Рынок игр является довольно разнообразным и динамичным, с большим количеством игр на разных платформах и различной целевой аудиторией. Стоит отметить, что мобильные игры становятся все более популярными, и существует множество новых игр, которые используют инновационные технологии. Конкуренция на рынке также довольно высока, что требует от разработчиков постоянного совершенствования и инноваций.

## Анализ методов решения

Анализ рынка существующих решений показал, что на данный момент развивающих игр в России набирает обороты популярности.

Главным вывод является, что создание собственного проекта для дошкольников будет наиболее эффективным, так как дети будут получать важную информацию в виде развлекательного контента. Такая программа поможет детям легче изучить различные темы.

**1.6** **Обоснование и выбор методики, технологии и инструментальных средств проектирования разработки**

**1.6.1 Выбор программы для разработки**

Существует несколько популярных движков для создания игр, которые предоставляют разработчикам мощные инструменты и ресурсы для реализации их творческих идей. Ниже описаны несколько из них:

– Unreal Engine. Движок, разработанный Epic Games , также является мощным и популярным движком для создания игр. Он предлагает высококачественную графику, продвинутую физику и реалистичные эффекты. Unreal Engine также имеет широкий набор инструментов для разработки игр на различных платформах и поддерживает различные языки программирования, такие как C++ и Blueprint , где Blueprints – язык визуального программирования, визуальная скриптовая среда, используемая для функционала игрового проекта ;

– Godot Engine. Является бесплатным и открытым исходным кодом движком для создания игр. Он предлагает простой в использовании визуальный редактор и поддерживает несколько языков программирования, включая GDScript, C# и C++. Godot Engine обладает мощными инструментами для создания 2D и 3D игр и поддерживает различные платформы, такие как Android, IOS, Windows, Linux, MacOS;

– Cocos2d. Является популярным движком для создания мобильных игр. Он предоставляет разработчикам возможность создавать 2D игры с использованием языков программирования, таких как C++, JavaScript и Lua. Cocos2d имеет широкий набор функций, включая анимацию, физику, управление аудио и сценами;

– Construct. Это движок для создания 2D игр без необходимости программирования. Он предлагает визуальный интерфейс и интуитивный редактор, который позволяет создавать игры путем перетаскивания и настройки объектов и событий. Construct поддерживает экспорт игр на различные платформы, включая веб, мобильные устройства и настольные компьютеры.

– Unity. Является одним из самых популярных и широко используемых движков для создания игр. Он поддерживает разработку игр для различных платформ, включая мобильные устройства, компьютеры, консоли и виртуальную реальность. Unity предлагает обширный набор инструментов и ресурсов, включая визуальный редактор, систему физики, анимации, аудио и многое другое;

Unity – кроссплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies.

Unity – среда, которая предоставляет множество функциональных возможностей для разработки игр, в которые входят моделирование физических сред, карты нормалей, динамические тени и многое другое. Первый фактор включает не только инструментарий визуального моделирования, но и интегрированную среду, цепочку сборки, что направлено на повышение производительности разработчиков, в частности, этапов создания прототипов и тестирования. Под межплатформенной поддержкой предоставляется не только места развёртывания (установка на персональном компьютере, на мобильном устройстве, консоли и т. д.), но и наличие инструментария разработки (интегрированная среда может использоваться под Windows и Mac OS).

В Unity модульная система компонентов, с помощью которой происходит конструирование игровых объектов, когда последние представляют собой комбинируемые пакеты функциональных элементов. В отличие от механизмов наследования, объекты в Unity создаются посредством объединения функциональных блоков, а не помещения в узлы дерева наследования [1]. Такой подход облегчает создание прототипов, что актуально при разработке игр.

Unity используется как крупными разработчиками, так и независимыми студиями. На Unity сделаны не малоизвестные игры, такие как Beat Saber, Fallout Shelter, Ori and the Blind Forest, Outer Wilds, и многое другое. На Unity написаны тысячи игр, приложений, визуализации математических моделей, которые охватывают множество платформ и жанров.

Для разработки дипломного проекта будет целесообразным использовать среду разработки Unity, так как он обладает большим спектров инструментов для создания игры с ориентиром на двумерное пространство.

**1.6.2 Выбор языка для разработки**

Язык программирования C# – это единственный язык, который поддерживается Unity .

У данного языка имеется рад плюсов:

1. объектно-ориентированный язык, позволяющий разработать многофункциональные приложения [2];
2. независимость от аппаратного функционала, программа не нуждается в адаптации под многочисленные платформы;
3. ориентация на безопасность кода;
4. расширенная поддержка событийно-ориентированного программирования [3].

# Специальная часть

# 2.1 Описание структуры приложения

Приложение представляет собой программный продукт, состоящий из нескольких компонентов:

* + пользовательский интерфейс, для взаимодействия с системой;
  + функции, для обработки запросов пользователей.

# 2.2 Объектно-ориентированное проектирование системы

# 2.2.1 Диаграмма использования

Диаграмма вариантов использования – диаграмма, описывающая функционал разрабатываемой программы (рисунок 1).

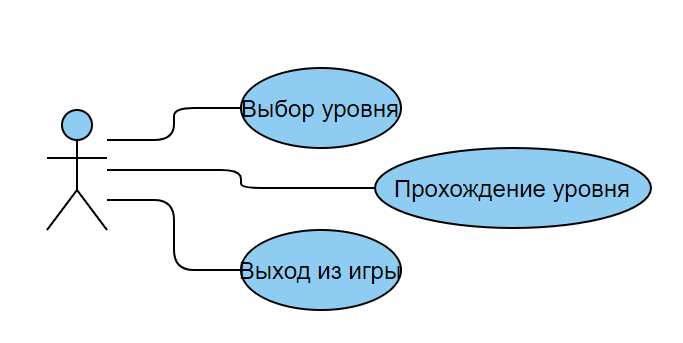


Рисунок 1 – Диаграмма вариантов использования

# 2.3 Разработка пользовательского интерфейса

# 2.3.1 Разработка пользовательского интерфейса для пользователя

Пользовательский интерфейс приложения «Математика для дошкольников» выполнен в светлых тонах [4](рисунок 2).

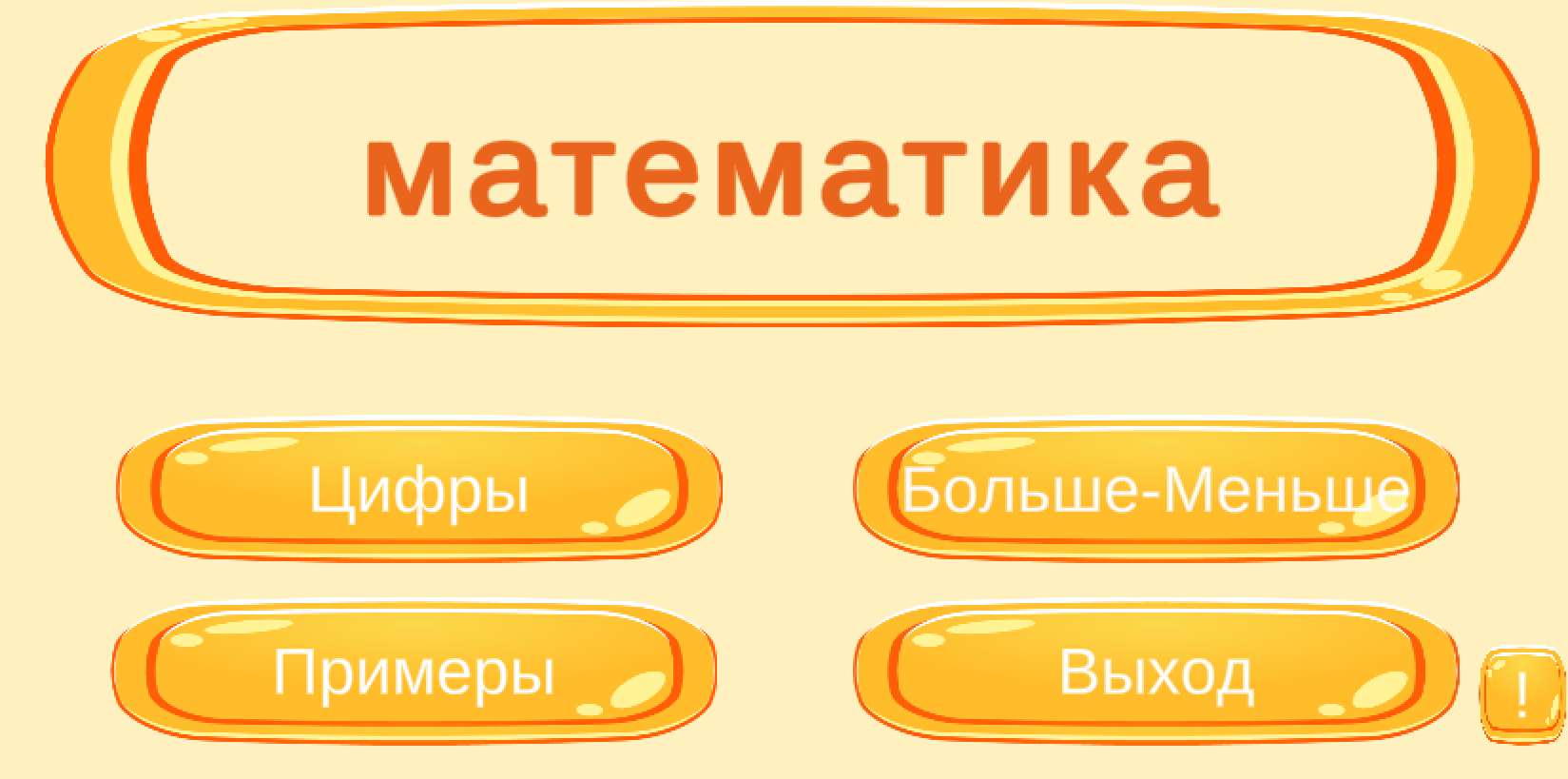


Рисунок 2 – Главная страница

Главная страница выполнена в легком стиле для удобства детей [5]. На ней находится 5 кнопок, при наведении на которые кнопки изменят состояние (рисунок 3).



Рисунок 3 – Изменение состояния кнопки

Элементы главной странице показаны на рисунке 4.

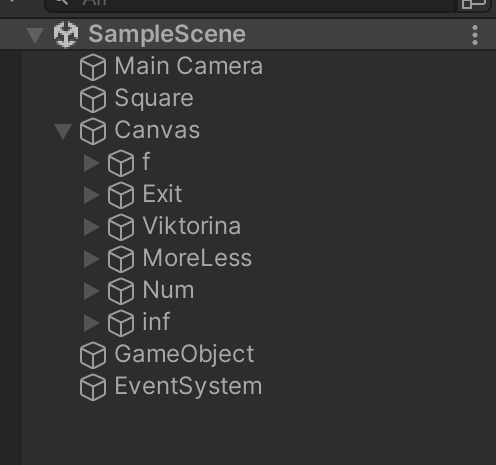


Рисунок 4 – Элементы главной страницы

Все кнопки настроены одинаково, и имеют разные текстуры для разных состояний кнопки (рисунок 5).

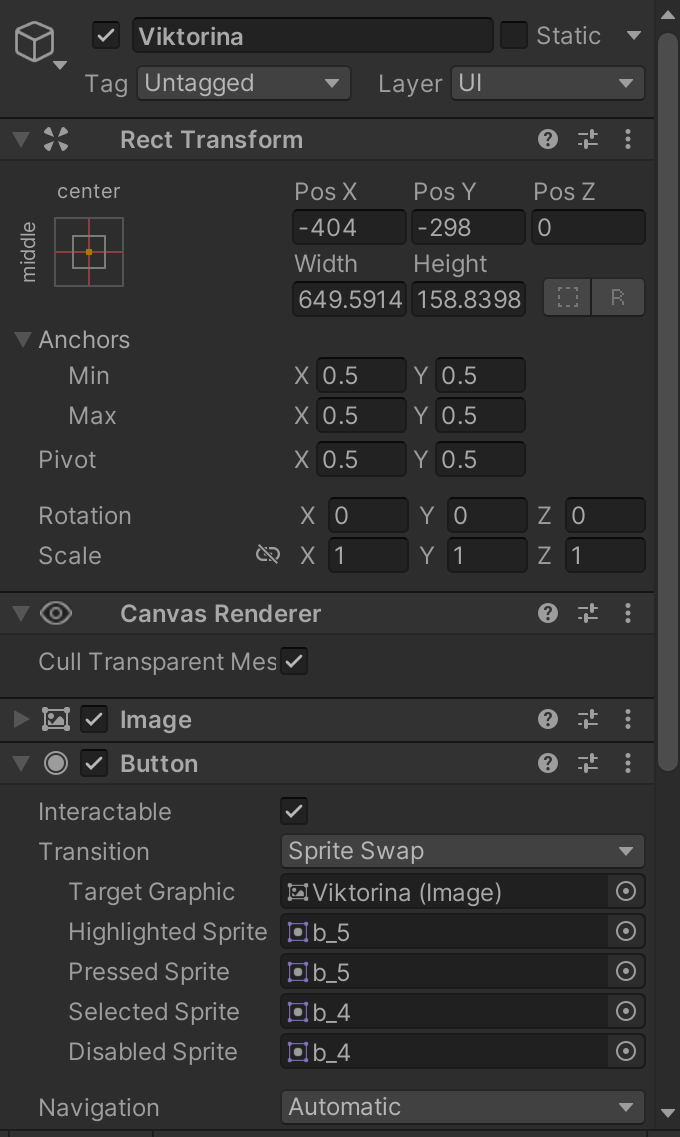


Рисунок 5 – Настройки кнопки

При нажатии на кнопку происходит событие, оно прописано в скрипте [6], для каждой кнопки есть свое событие (рисунок 6).

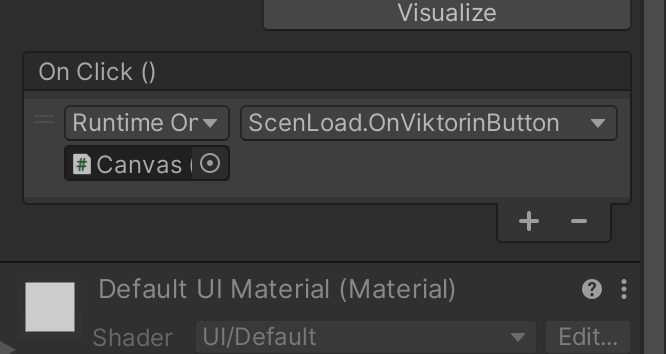


Рисунок 6 – Событие при нажатии кнопки

В приложение есть три уровня сложности, первый уровень — это цифры. Игроку выводится число монеток, нужно посчитать и выбрать правильный ответ (рисунок 7).

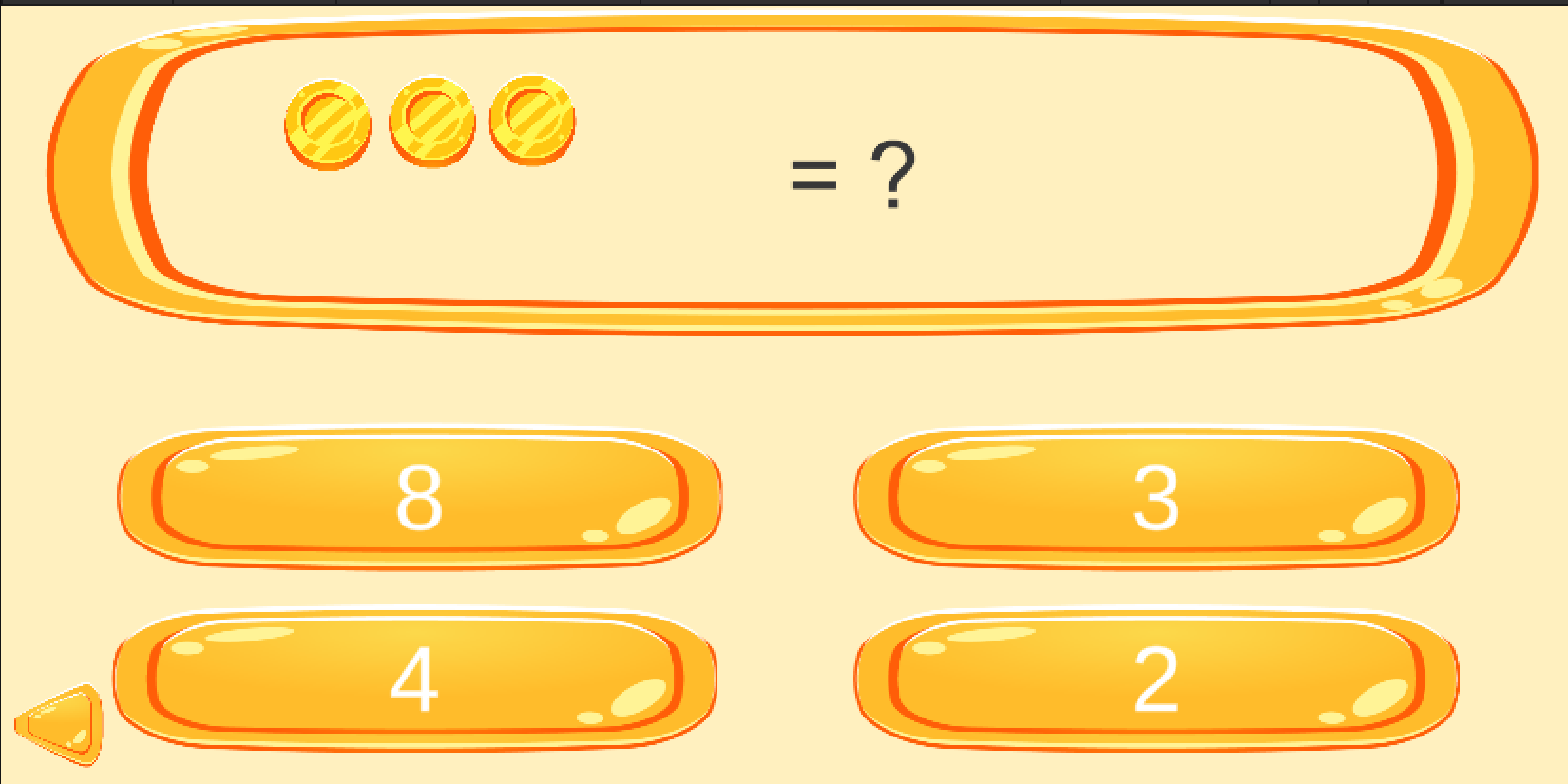


Рисунок 7 – «Цифры»

Элементы уровня «Цифры» показаны на рисунке 8.

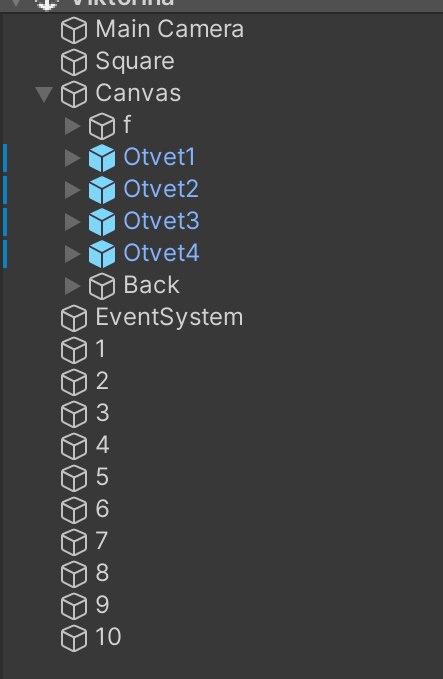


Рисунок 8 – Элементы уровня «Цифры»

Второй уровень сложности – это больше или меньше, в нем

случайным образом генерируется количество монеток, число и три варианта ответа, нужно выбрать правильный [7](рисунок 9).



Рисунок 9 – «Больше-Меньше»

Элементы уровня «Больше-Меньше» показаны на рисунке 10.

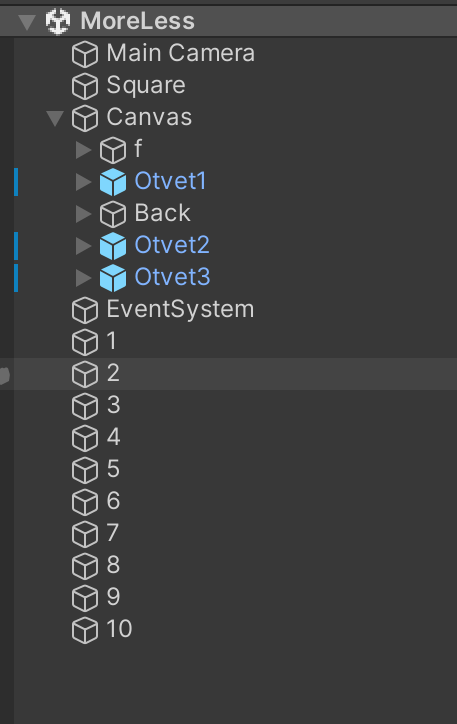


Рисунок 10 – Элементы уровня «Цифры»

Третий уровень сложности – это пример, в нем случайным образов генерируется пример и дается на выбор 4 варианта ответа [8](рисунок 11).

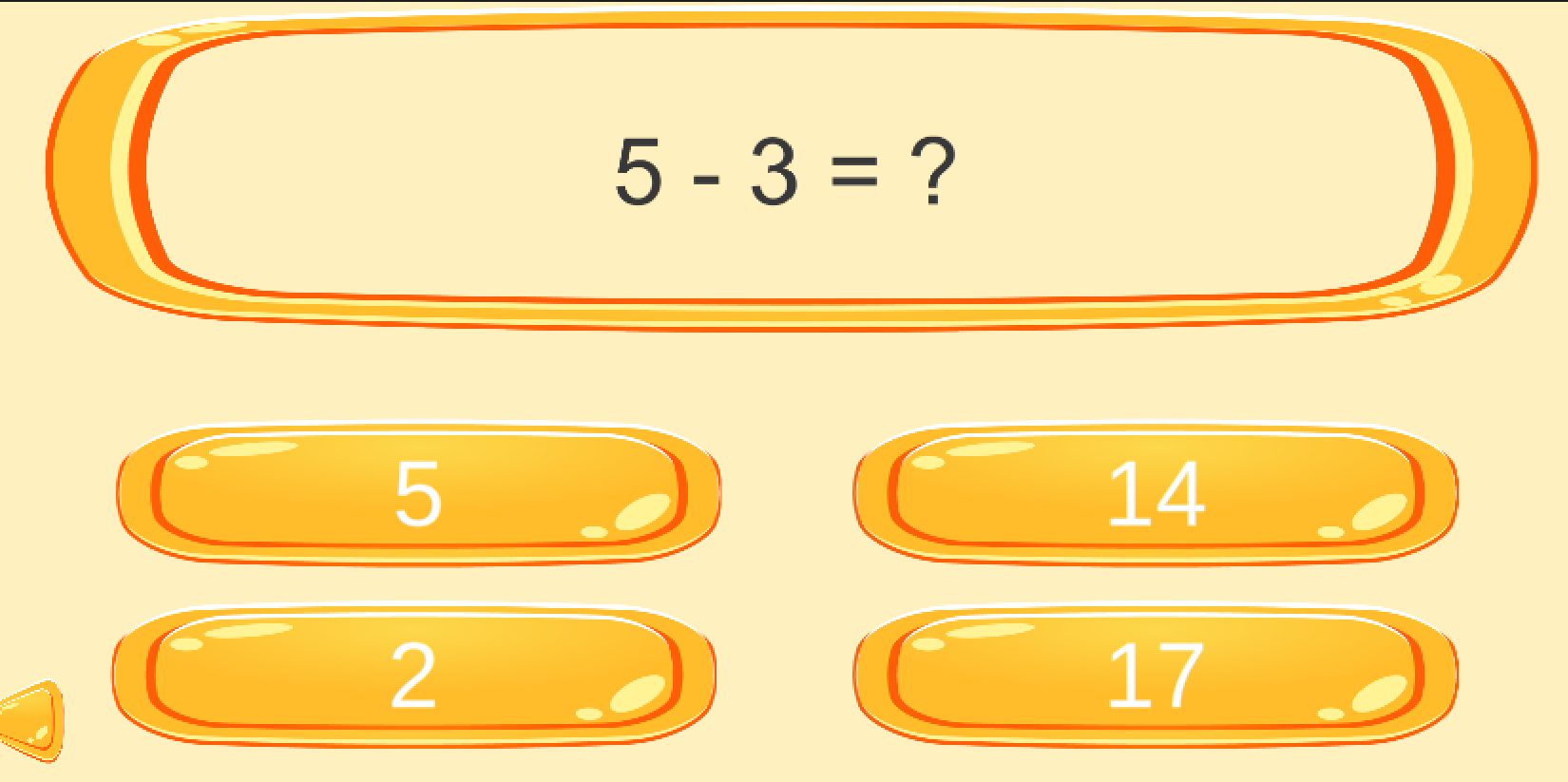


Рисунок 11 – «Примеры»

Элементы уровня «Примеры» показаны на рисунке 12

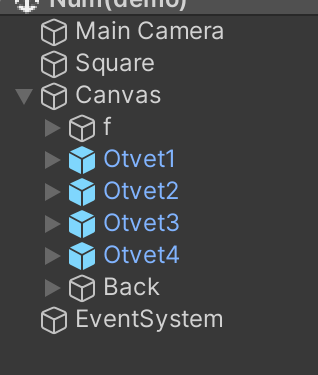


Рисунок 12 – Элементы уровня «Примеры»

При выборе правильного ответа на месте задания в течении небольшого времени будет надпись "молодец!" [9](рисунок 13).

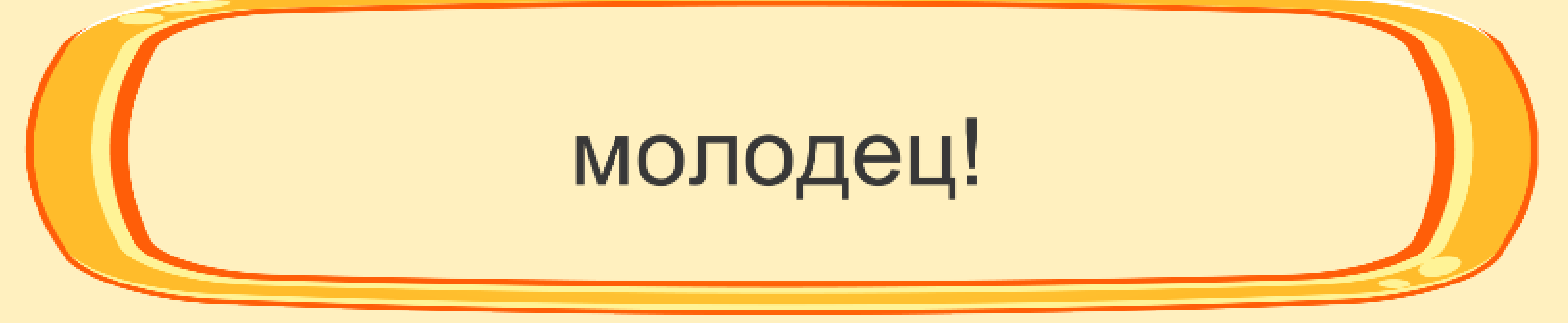


Рисунок 13 – Выбор правильного ответа

При выборе неправильного ответа на месте задания в течении небольшого времени будет виден надпись "попробуй еще раз" далее задание будет выведено повторно [10] (рисунок 14).



Рисунок 14 – Выбор неправильного ответа

**2.4 Код программы**

Листинг кода программы представлен в приложении А.

**2.5 Реализация программы**

**2.5.1 Загрузка и запуск программы**

Для запуска программы необходимо скачать её и нажать на иконку, которая появится на рабочем столе. После этого откроется главная страница игры, представленная на рисунке 2.

**2.5.3 Завершение программы**

Для завершения программы во время игры необходимо перейти к главной странице и нажать на кнопку «Выход» (рисунок 15).

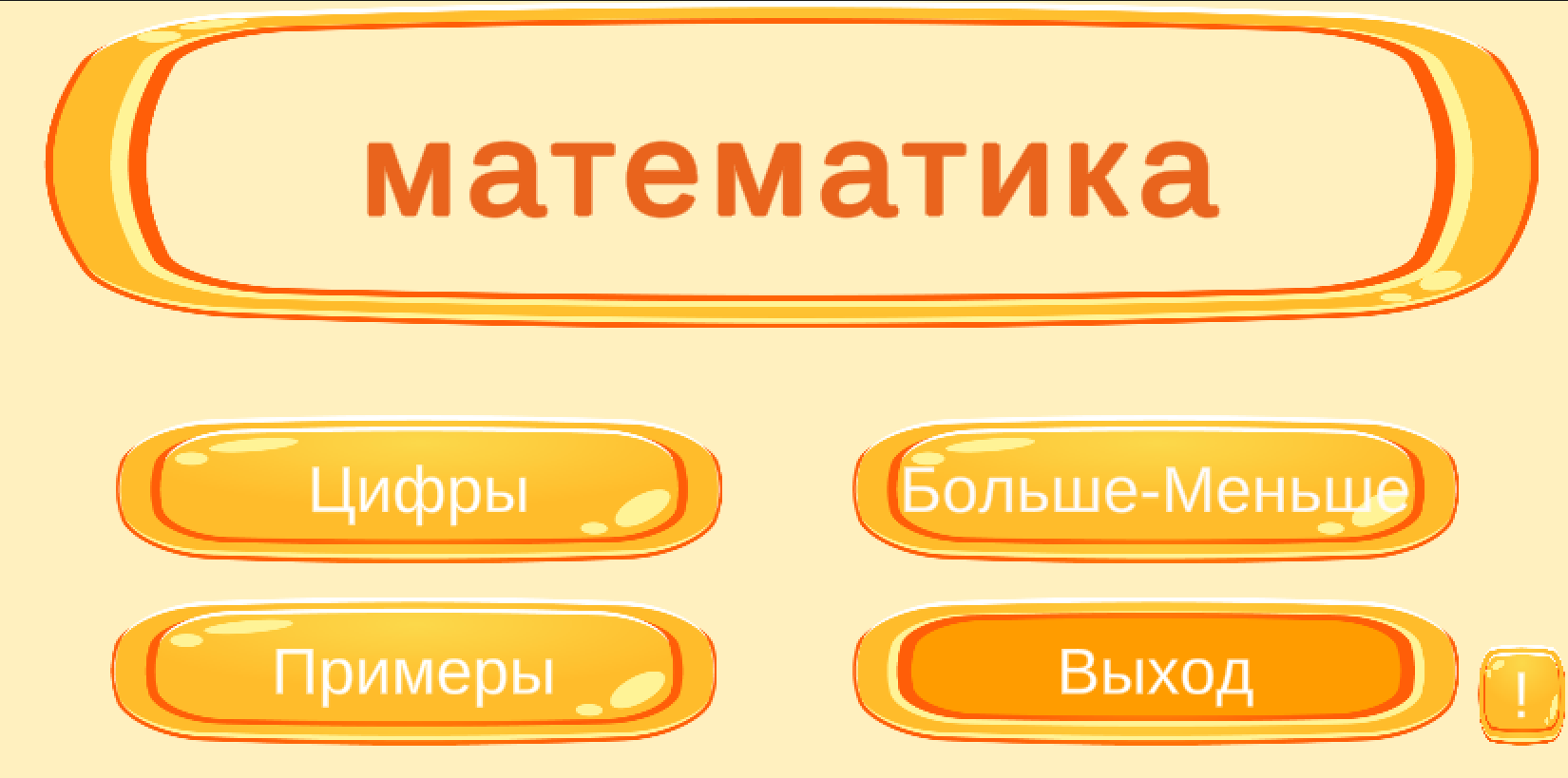


Рисунок 15 – Изменение состояния кнопки

# Экономическая часть

## 3.1 Область применения программного продукта и его преимущества перед аналогичным программным продуктом

Игра «Математика для дошкольников» предназначена для организации полезного досуга для пользователей дошкольного возраста, а также для развития навыков пользователей в сфере математики.

Преимуществами игры являются, реализация различных уровней сложности, поэтапно обучающих детей математическим основам, начиная с ознакомления с цифрами и их значениями и заканчивая изучением простейших примеров со знаками «плюс» и «минус», отсутствие всплывающей рекламы, наличие аудио-сопровождения игры.

# 3.2 Трудоемкость разработки программного продукта, квалификация исполнителя и его оклад

Трудоемкость разработки можно определить в таблице 1. Строка «Всего» отображает общую трудоемкость разработки.

Таблица 1 – Трудоемкость разработки программного продукта

| **Наименование этапа** | **Условное обозначение** | **Трудоемкость выполнения этапа, час** |
| --- | --- | --- |
| Сбор информации, описание задания | То | 40 |
| Разработка архитектуры приложения | Тра | 16 |
| Разработка интерфейса приложения | Три | 65 |
| Разработка функционала приложения | Тфп | 120 |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| **Наименование этапа** | **Условное обозначение** | **Трудоемкость выполнения этапа, час** |
| Отладка программы | Топ | 48 |
| Оценка качества программы | Тоц | 20 |
| Оформление документации | Тд | 43 |
| Всего | Тобщ | 352 |

Оклад и тарифная ставка разработчика программного продукта представлены в таблице 2.

Таблица 2 – Оклад и тарифная ставка разработчика

|  |  |  |  |
| --- | --- | --- | --- |
| **Исполнители** | **Оклад, руб** | **Часовая тарифная ставка, руб./час** | **Количество сотрудников** |
| Разработчик-программист | 53 000 | 331,25 | 1 |

Часовая тарифная ставка ЧТС, руб./час, определяется исходя из месячного оклада, количества рабочих дней в месяце и продолжительности рабочего дня по формуле:

, (1)

где Ом – оклад исполнителя в месяц, руб./мес;

Д – количество рабочих дней в месяце (для расчета Д = 20 раб. дней);

Тс – продолжительность рабочего дня (для расчета Тс = 8час.).

В таблице 3 показана стоимость технических средств для разработки системы.

Таблица 3 – Стоимость технических средств разработки ИСУ

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование компонента** | **Цена, руб.** | **Количество, шт.** | **Стоимость, руб.** |
| Ноутбук ASUS Zenbook 14 Flip OLED | 60000 | 1 | 60000 |
| МФУ Epson L350 | 20990 | 1 | 20990 |
| Итого: | | | 80990 |

В таблице 4 представлены затраты на расходные материалы.

Таблица 4 – Планируемые затраты на расходные материалы

|  |  |  |  |
| --- | --- | --- | --- |
| **Затраты** | **Стоимость** | **Количество** | **Сумма, руб.** |
| Интернет | 420 руб./мес. | 2 месяца | 820 |
| Электричество | 4 руб./КВт\*ч | 490 КВт\*ч | 1960 |
| Бумага | 250 руб. | 2 упаковки | 500 |
| Ручка | 45 руб. | 1 штука | 45 |
| Заправка МФУ | 1800 руб. | 1 раз | 1800 |
| Итого: | | | 5125 |

**3.3 Расчет затрат на разработку**

Исходные данные, связанные с разработкой программного продукта приведены в таблице 5.

Таблица 5 – исходные данные

| **Наименование** | **Обозначение** | **Значение** |
| --- | --- | --- |
| Оклад разработчика | Ор | 53 000 руб. |
| Время разработки | Трп | 2 месяца |
| Машинное время разработки | Тмч | 1,5 месяца |
| Коэффициент дополнительной заработной платы | Кд | 0,12 |
| Коэффициент страховых взносов | Кст | 0,3 |
| Количество единиц техники | Q | 1 шт |
| Себестоимость содержания техники | См/ч | 15 руб./час |
| Коэффициент готовности техники | Кгт | 0,95 |
| Число рабочих дней в месяце | ЧРД | 20 дней |
| Продолжительность смены | Тст | 8 часов |
| Коэффициент сменности | Ксм | 1 |
| Коэффициент транспортных расходов | Кт | 0,15 |
| Коэффициент накладных расходов | Кнр | 0, 54 |

Произведя вычисления было выявлено, что полные затраты на разработку составляют 300 829,86 рублей. Промежуточные результаты вычислений представлены в таблице 6.

Таблица 6 – Результаты вычислений

| **Наименование** | **Обозначение** | **Значение** |
| --- | --- | --- |
| Оплата труда | Зот | 118720 руб.; |
| Страховые взносы | Зст | 35616 руб.; |
| Содержание ЭВМ | Зэвм | 3420 руб.; |
| Затраты на специальные программы | Зсп | 58590 руб.; |
| Расходные материалы | Зрм | 5893,75 руб.; |
| Накладные расходы | Знр | 64108,8 руб.; |
| Итого затрат на разработку | Зрп | 286 348,55 руб. |

Расчет полных затрат на разработку проектного решения (программного продукта) осуществляется по формуле:

Зрп = Зот + Зст + Зэвм + Зрм + Знр + Зсп, (2)

где Зот – затраты на оплату труда разработчика (разработчиков), руб.;

Зст – страховые взносы по оплате труда во внебюджетные фонды, руб.;

Зэвм – затраты, связанные с содержанием вычислительной техники, руб.;

Зрм – затраты на расходные материалы, необходимые при разработке программного продукта, руб.;

Знр – затраты по накладным расходам, приходящиеся на разработку программного продукта, руб..

Зрп = 118720 + 35616 + 3420 + 58590 +5893,75 + 64108,8 = 286348,55

**3.3.1 Затраты на оплату труда разработчику (Зот), руб.**

Размер фонда оплаты труда разработчиков (Зот) рассчитывается по формуле:

Зот = Ор \* Трп \* (1 + Кд), (3)

где Ор – месячный оклад разработчика проектного решения, руб./мес.;

Трп – время разработки проектного решения разработчиком, мес.(час) включает в себя машинное время работы над проектом (Тмрп);

Кд – коэффициент дополнительной заработной платы разработчика;

Если время разработки программного продукта менее месяца, т.е. определено в часах, то в формулу подставляется часовая тарифная ставка, определяется по формуле 1.

Зот = 53000 \* 2 \* (1 + 0,12) = 118720

**3.3.2 Затраты по страховым взносам (Зст), руб.**

Сумма страховых взносов определяется по формуле:

Зст = Кст \* Зот, (4)

где Кст – коэффициент страховых взносов для расчета отчислений во внебюджетные фонды.

Зст = 0,3 \* 118720 = 35616

**3.3.3 Затраты по содержанию Эвм (Зэвм), руб.**

Затраты, связанные с эксплуатацией и содержанием ЭВМ, определяются по формуле:

Зэвм = Тмрп \* Кгт \* Q \* См/ч, (5)

где Тмрп – машинное время на разработку проектного решения, час;

Кгт – коэффициент готовности техники;

Q – количество условных единиц используемой техники;

См/ч – стоимость машино-часа, эксплуатации оборудования, руб. в час.

Зэвм = 240 \* 0,95 \* 1 \*15 = 3420

Так как машинное время может измеряться в месяцах, а себестоимость машино-часа за один час, то машинное время необходимо перевести в часы.

Перевод рабочего времени в часы осуществляется по формуле:

Тмрп = Тмч \* Чрд \* Тсм \* Ксм, (6)

Где Тмч рабочее время в месяцах;

Чрд – число рабочих дней в месяце;

Тсм – продолжительность рабочей смены;

Ксм – количество рабочих смен.

Тмрп = 1,5 \* 20 \* 8 \* 1 = 240

**3.3.4 Затраты на расходные материалы**

Затраты на расходные материалы, необходимые для разработки проектного решения определяются по формуле:

, (7)

Где Qi – количество i – го вида материала, шт.;

Рi – цена i – го вида материала, руб.;

n – количество видов материалов;

Кт – коэффициент транспортных расходов.

Зрм = 5893,75

**3.3.5 Затраты по накладным расходам (Знр)**

Затраты по накладным расходам определяются по формуле:

Знр = Кнр \* Зот, (8)

где Кнр – коэффициент накладных расходов (или процент) принимается для расчета по данным предприятия;

Зот – затраты по оплате труда, руб.

Знр = 0,54 \* 118720 = 64108,8

**3.4 Расчет цены и прибыли**

Исходные данные, связанные с разработкой программного продукта приведены в таблице 7.

Таблица 7 – исходные данные

|  |  |  |
| --- | --- | --- |
| **Наименование** | **Обозначение** | **Значение** |
| Коэффициент рентабельности | Кр | 0,25 |
| Коэффициент налога на добавочную стоимость | Кндс | 0,2 |
| Ставка налога | Кнп | 0,2 |
| Стоимость затрат на разработку и внедрение | Квпр | 286 348,55 руб. |

Произведя вычисления было выявлено, что чистая прибыль будет составлять 57 269,71 рублей. Промежуточные результаты вычислений представлены в таблице 8.

Таблица 8 – Результаты вычислений

|  |  |  |
| --- | --- | --- |
| **Наименование** | **Обозначение** | **Значение** |
| Плановая прибыль | Ппл | 71 587,138 |
| Налог на добавочную стоимость | НДС | 71 587,138 |
| Цена программного продукта | Цпп | 429 522,826 |
| Налог на прибыль | НП | 14 317,428 |
| Чистая прибыль | ЧП | 57 296,71 |
| Поступление в бюджет | ПБ | 85 904,566 |

Цена программного продукта, который разработан одной организацией по заказу другой и не предназначен для тиражирования, определяется по формуле:

Цпп = Квпр +Ппл + НДС, руб., (9)

Цпп = 286 348,55 + 71 587,138 + 71 587,138 = 429 522,826

где Ппл – планируемая прибыль рассчитывается по формуле:

Ппл = Квпр \* Кр, руб исходя из нормативной рентабельности (коэффициента) Кр = 0,25.

Ппл = 286 348,55 \* 0,25 = 71 587,138

НДС – налог на добавленную стоимость определяется исходя из Кндс = 0,2 (ставка налога 20%) по формуле:

НДС = (Квпр + Ппл) \* Кндс, руб. (10)

НДС = (286 348,55 + 71 587,138) \* 0,2 = 71 587,138

Каждое предприятие с полученной прибыли перечисляет государству налог на прибыль. На сегодня ставка налога 20% (Кнп = 0,2) от полученной прибыли, и определяется по формуле:

НП = Ппл \* Кнп, руб. (11)

НП = 71 587,138 \* 0,2 = 14 317,428

Чистая прибыль, остающаяся в распоряжении предприятия:

ЧП = Ппл – НП, руб. (12)

ЧП = 71 587,138 – 14 317,428 = 57 269,71

Поступление в бюджет складываются из налога на прибыль и НДС.

ПБ = НП + НДС, руб. (13)

ПБ = 14 317,428 + 71 587,138 = 85 904,566

Исходя из расчетов экономической части затраты на разработку составили 286 348,55 руб. рублей, цена программного продукта составляет 429 522,826 рублей, а чистая прибыль составляет 57 269,71 рублей.

1. **Техника безопасности и охрана труда**
   1. **Анализ условий труда разработчика игр**

Рабочее место разработчика игр – это часть помещения, где он проводит большую часть рабочего времени. Совокупность показателей технических и программных средств определяет качество и производительность работы, наряду с габаритными показателями мебели и ее удобством.

Помещение для работы представляет собой кабинет длиной 6 м, шириной 4 м и высотой потолков 2,5 м.

Согласно санитарным правилам, монитор на столе нужно располагать на расстоянии 60- 70 см от глаз пользователя, но не ближе 50 см с учетом размеров алфавитно-цифровых знаков и символов. При использовании жидкокристаллических дисплеев на каждый компьютер должно приходиться не менее 4,5 м2 площади. На дисплей ПК не должен попадать прямой солнечный свет во избежание бликов и повышенной нагрузки на зрение.

Для снижения нагрузки на технические средства, влекущей за собой перегрев компьютера и ноутбука, используются программные средства, позволяющие снизить нагрузку на технические средства за счёт отказа в использование ненужных служб.

При идентификации вредных производственных факторов было установлено, что опасными могут являться:

* электромагнитные поля и излучения от технических средств;
* статическое электричество;
* статические перегрузки, вызываемые неподвижной работой за компьютером;
* перенапряжение органов зрения.

В целях выявления вредных и (или) опасных факторов производственной среды и трудового процесса, оценке уровня их воздействия на работника с учетом отклонения их фактических значений от установленных уполномоченным Правительством Российской Федерации федеральным органом исполнительной власти гигиенических нормативов условий труда и применения средств индивидуальной и коллективной защиты работников проводится специальная оценка условий на рабочих местах. Оценке подлежат все имеющиеся в организации рабочие места.

Работа по специальной оценке условий труда заключается в исследовании следующих факторов на рабочих местах организации:

* химические факторы;
* физические факторы;
* биологические факторы;
* тяжесть и напряженность трудового процесса.

Для уменьшения нагрузки на органы зрения, а также снятия мышечного напряжения во время рабочего дня проводится проветривание в течение 10-15 минут каждые 3 часа.

* 1. **Расчет искусственного освещения в помещении**

Через зрительный анализатор человек получает около 80% из общего объема информации. Ее качество, в основном, зависит от освещения в помещении. Если оно неудовлетворительное, это приводит к утомлению организма в целом. Кроме того, нерациональное освещение может явиться причиной травматизма, плохо освещенные опасные зоны, слепящие источники света и блики от них, резкие тени ухудшают видимость настолько, что вызывают полную потерю ориентировки работающих, снижают производительность труда и увеличивают брак продукции. Поэтому необходимо достаточное искусственное освещение для рабочего места.

Условия работы для расчета показателя освещенности рабочего места разработчика являются:

* помещение размером 24 квадратных метров;
* высота помещения 2,5 метров, длина 6 метров, ширина 4 метра;
* светильники светодиодные в количестве штук, новые;
* крупногабаритная мебель отсутствует.

Основным критерием, по которому определяется необходимое количество осветительных приборов, является нормируемый уровень освещенности.

Базовым методом проектирования осветительной установки является метод коэффициентов использования, позволяющий вручную проводить все вычислительные процедуры при решении относительно простых светотехнических задач.

По этому методу необходимое количество светильников в осветительной установке определяется с помощью следующей формулы:

 (14)

где N – количество светильников, шт;

Е – нормируемая освещенность, лк, определяется по СП 52.13330.2016 «Естественное и искусственное освещение»;

S – площадь помещения, м2;

Кз – коэффициент запаса;

U – коэффициент использования;

Фсв – световой поток светильника, лм.

Коэффициент использования характеризует эффективность использования светового прибора в помещении. Для его определения необходимо знать индекс помещения φ и коэффициент отражения стен, пола и потолка.

Индекс помещения:

 (15)

где  – индекс помещения;

S – площадь помещения, м2;

h1 – высота подвеса светильников над рабочей поверхностью, м;

h2 – высота рабочей поверхности, м;

а – длина помещения, м;

b – ширина помещения, м.

*φ* = 1,307

*N* = 4,138

Полученное количество светильников совпадает с числом светильников, установленных в помещении, что соответствует нормативным показателям искусственного освещения согласно СП 52.13330.2016 «Естественное и искусственное освещение».

* 1. **Электробезопасность на предприятии**

При поступлении на работу сотрудник должен пройти инструктаж по технике безопасности и электробезопасности. Работника знакомят с основными правилами по технике безопасности, предлагают внимательно прочитать действующие на предприятии инструкции, поясняя при этом отдельные правила и требования.

Инструктаж по технике безопасности при выполнении конкретной работы проводит руководитель соответствующего производственного участка, показывая инструктируемому правильные безопасные приемы работы.

Повторный инструктаж проводится для рабочих независимо от их квалификации, стажа и опыта работы, не реже 1 раза в 6 месяцев по программе инструктажа на рабочем месте.

При нарушении работниками правил и инструкций по технике безопасности, технологической и производственной дисциплины, а также в случаях изменения технологического процесса или вида работы проводят дополнительные инструктажи.

Все инструктажи оформляются записями в специальном журнале с указанием номеров или шифров инструкций. Журнал о проведении инструктажа хранится у руководителя подразделения.

Работник на рабочем месте должен соблюдать общие меры электробезопасности. При этом запрещается:

* прикасаться к задней панели системного блока при включенном питании;
* допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и др. устройств;
* производить самостоятельно вскрытие и ремонт оборудования;
* пользоваться неисправными электроприборами и электропроводкой;
* ремонтировать электроприборы самостоятельно;
* подвешивать электропровода на гвоздях, металлических и деревянных предметах, перекручивать провод, закладывать провод и шнуры на водопроводные трубы и батареи отопления, вешать что-либо на провода, вытягивать за шнур вилку из розетки;
* прикасаться одновременно к персональному компьютеру и к устройствам, имеющим соединение с землей (радиаторы отопления, водопроводные краны, трубы и т.п.);
* пользоваться самодельными электронагревательными приборами и электроприборами с открытой спиралью;
* наступать на переносимые электрические провода, лежащие на полу.

При работе с электроприборами и оргтехникой (персональные компьютеры, принтеры, сканеры, копировальные аппараты, факсы, бытовые электроприборы, приборы освещения) должны соблюдаться следующие условия:

* автоматические выключатели и электрические предохранители должны быть всегда исправны;
* изоляция электропроводки, электроприборов, выключателей, штепсельных розеток, ламповых патронов и светильников, а также шнуров, с помощью которых включаются в электросеть электроприборы, должны быть в исправном состоянии;
* электроприборы необходимо хранить в сухом месте, избегать резких колебаний температуры, вибрации, сотрясений;
* для подогрева воды пользоваться сертифицированными электроприборами с закрытой спиралью и устройством автоматического отключения, с применением несгораемых подставок.
  1. **Пожарная безопасность на предприятии**

Инструктаж по пожарной безопасности проводится по программе, разработанной ответственным по охране труда организации, с учетом требований стандартов, правил, норм и инструкций о мерах пожарной безопасности. Продолжительность инструктажа устанавливается в соответствии с утвержденной программой. Инструктаж по пожарной безопасности, как правило, проводится совместно с инструктажем по технике безопасности и в те же сроки.

Инструктаж по пожарной безопасности проходят все работники организации, независимо от их образования, стажа работы по данной профессии или должности, временные работники, командированные, обучающиеся и студенты, прибывшие на практику.

О проведении инструктажа по пожарной безопасности работник, проводивший инструктаж, делает запись в журнале учета проведения инструктажей по пожарной безопасности, с обязательной подписью инструктируемого и инструктирующего. При регистрации внепланового инструктажа по пожарной безопасности указывают причину его проведения.

Все работники организаций должны допускаться к работе только после прохождения противопожарного инструктажа, а при изменении специфики работы проходить дополнительное обучение предупреждению и тушению возможных пожаров в порядке, установленном руководителем.

На проведение первичного противопожарного инструктажа необходимо отводить не менее 1 часа. Инструктируемые должны ознакомиться:

* с действующими на объекте правилами пожарной безопасности и инструкциями;
* с производственными участками, наиболее опасными в пожарном отношении, где запрещается курить, применять открытый огонь;
* с возможными причинами возникновения пожара и мерами его предупреждения;
* с практическими действиями в случае возникновения пожара - вызов пожарной помощи, использование первичных средств пожаротушения, место расположения ближайшего телефона и ознакомление с правилами поведения в случае возникновения пожара, эвакуации людей и материальных ценностей.

При первичном инструктаже инструктирующий обязан рассказать о производственных установках с повышенной пожарной опасностью, мерах предотвращения пожаров и загораний, указать место курения, ознакомить вновь поступившего с имеющимися на объекте средствами пожаротушения, показать ближайший телефон и объяснить правила поведения в случае возникновения пожара.

Проведение противопожарного инструктажа в обязательном порядке должно сопровождаться практическим показом способов использования имеющихся на объекте средств пожаротушения (противогазы, респираторы, огнетушители и т.д.).

Соблюдение рассмотренных в данном разделе мероприятий позволяет снизить утомляемость и травматизм, повысить производительность труда, обеспечить комфортные условия трудовой деятельности.

## ЗАКЛЮЧЕНИЕ

Результатом разработки проекта выпускной квалификационной работы является обучающая игра «математика для дошкольников».

Посредством анализа рынка похожих приложений, анализа методов решения, а также проектирования был создан программный продукт, который служит для организации досуговой деятельности детей дошкольного возраста, а тек же в развитии и поддержании когнитивных навыков и мыслительных процессов.

Программа была создана на платформе Unity с использованием языка C#.

Продукт является экономически эффективным и удовлетворяет всем требованиям. В ходе разработки были реализованы главное меню, 3 различных по сложности уровня, помогающие в том, чтобы научить считать, сравнивать, вычитать и складывать числа.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Нейгел, К. C# 5.0 и платформа .NET 4.5 для профессионалов / К. Нейгел и др. – М. : Вильямс, 2020. – 1440 c.
2. Вагнер, Билл С# Эффективное программирование / Билл Вагнер. – М. : ЛОРИ, 2013. – 320 c.
3. Прайс, Джейсон Visual C# 2.0. Полное руководство / Джейсон Прайс , Майк Гандэрлой. – М. : Век +, Корона-Век, Энтроп, 2010. – 736 c.
4. Секунов, Н. Самоучитель C# / Н. Секунов. – М. : БХВ-Петербург, 2017. – 576 c.
5. Троелсен, Эндрю Язык программирования C# 5.0 и платформа .NET 4.5 / Эндрю Троелсен. – М. : Вильямс, 2015. – 486 c.
6. Албахари, Джозеф C# 3.0. Справочник / Джозеф Албахари , Бен Албахари. – М. : БХВ-Петербург, 2012. – 944 c.
7. Афанасьева, И. П. Вместе учимся считать. Занимательная математика для дошкольников 4-5 лет. Рабочая тетрадь №4 / И.П. Афанасьева. – М. : Детство-Пресс, 2015. – 113 c.
8. Баряева, Л. Б. Математика для дошкольников в играх и упражнениях / Л.Б. Баряева, С.Ю. Кондратьева. – М. : Каро, 2007. – 288 c.
9. Вакуленко, Ю.А. Веселая грамматика. Разработки занятий, задания, игры / Ю.А. Вакуленко. – М. : Учитель, 2017. – 780 c
10. Платов, В. Я. Деловые игры. Разработка, организация, проведение. Учебник / В.Я. Платов. – М. : Профиздат, 2010. – 192 c.

Приложение А(справочное)

Листинг кода

**А.1 Исходный код victorina**

Здесь представлен код работы с уровнем "пример"

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.SceneManagement;

using TMPro;

public class Num : MonoBehaviour

{

public GameObject[] arrM = new GameObject[10];

public Text primer;

public TextMeshProUGUI otv1;

public TextMeshProUGUI otv2;

public TextMeshProUGUI otv3;

public TextMeshProUGUI otv4;

public GameObject ot1;

public GameObject ot2;

public GameObject ot3;

public GameObject ot4;

public float delay;

private float timer;

private int a, rez;

// Start is called before the first frame update

void Start()

{

for (int i = 0; i < 10; i++)

{

arrM[i].SetActive(false);

}

timer = delay;

numGenerate();

for (int i = 0; i < a; i++)

{

arrM[i].SetActive(true);

}

}

// Update is called once per frame

void Update()

{

int count = 0;

if (primer.text == "молодец!")

{

count++;

timer -= Time.deltaTime;

if (timer <= 0)

{

timer = delay;

numGenerate();

}

}

else if (primer.text == "попробуй еще раз")

{

timer -= Time.deltaTime;

if (timer <= 0)

{

for (int i = 0; i < a; i++)

{

arrM[i].SetActive(true);

}

timer = delay;

primer.text = a + " \t = ?";

}

}

}

void numGenerate()

{

a = Random.Range(0, 10);

for (int i = 0; i < a; i++)

{

arrM[i].SetActive(true);

}

primer.text = " \t = ?";

int[] x = new int[4];

int t = 0;

while (t < 4)

{

int y = Random.Range(0, 10);

if (y != a && y != x[0] && y != x[1] && y != x[2] && y != x[3])

{

x[t] = y;

t++;

}

}

int truf = Random.Range(0, 4);

x[truf] = a;

otv1.text = x[0] + "";

otv2.text = x[1] + "";

otv3.text = x[2] + "";

otv4.text = x[3] + "";

}

public void OnButton1Click()

{

for (int i = 0; i < 10; i++)

{

arrM[i].SetActive(false);

}

if (a == int.Parse(otv1.text))

{

primer.text = "молодец!";

timer = delay;

}

else { primer.text = "попробуй еще раз"; }

}

public void OnButton2Click()

{

for (int i = 0; i < 10; i++)

{

arrM[i].SetActive(false);

}

if (a == int.Parse(otv2.text))

{

primer.text = "молодец!";

timer = delay;

}

else { primer.text = "попробуй еще раз"; }

}

public void OnButton3Click()

{

for (int i = 0; i < 10; i++)

{

arrM[i].SetActive(false);

}

if (a == int.Parse(otv3.text))

{

primer.text = "молодец!";

timer = delay;

}

else { primer.text = "попробуй еще раз"; }

}

public void OnButton4Click()

{

for (int i = 0; i < 10; i++)

{

arrM[i].SetActive(false);

}

if (a == int.Parse(otv4.text))

{

primer.text = "молодец!";

timer = delay;

}

else { primer.text = "попробуй еще раз"; }

}

public void OnBackButton()

{

SceneManager.LoadScene(0);

}

}

**А.2 Исходный код moreless**

Здесь представлен код работы с уровнем "больше – меньше"

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.SceneManagement;

using TMPro;

public class moreless : MonoBehaviour

{

public Text primer;

public GameObject[] arrM = new GameObject[10];

public GameObject ot1;

public GameObject ot2;

public GameObject ot3;

public float delay;

private float timer;

private int a, b;

// Start is called before the first frame update

void Start()

{

for (int i = 0; i < 10; i++)

{

arrM[i].SetActive(false);

}

timer = delay;

moorelessGenerat();

for (int i = 0; i < a; i++)

{

arrM[i].SetActive(true);

}

}

// Update is called once per frame

void Update()

{

int count = 0;

if (primer.text == "молодец!")

{

count++;

timer -= Time.deltaTime;

if (timer <= 0)

{

timer = delay;

moorelessGenerat();

}

}

else if (primer.text == "попробуй еще раз")

{

timer -= Time.deltaTime;

if (timer <= 0)

{

timer = delay;

for (int i = 0; i < a; i++)

{

arrM[i].SetActive(true);

}

primer.text = " \t? " + b;

}

}

}

void moorelessGenerat()

{

a = Random.Range(0, 10);

b = Random.Range(0, 10);

for (int i = 0; i < a; i++)

{

arrM[i].SetActive(true);

}

primer.text = " \t? " + b;

}

public void OnButton1Click()

{

for (int i = 0; i < 10; i++)

{

arrM[i].SetActive(false);

}

if (a > b)

{

primer.text = "молодец!";

timer = delay;

}

else { primer.text = "попробуй еще раз"; }

}

public void OnButton2Click()

{

for (int i = 0; i < 10; i++)

{

arrM[i].SetActive(false);

}

if (a < b)

{

primer.text = "молодец!";

timer = delay;

}

else { primer.text = "попробуй еще раз"; }

}

public void OnButton3Click()

{

for (int i = 0; i < 10; i++)

{

arrM[i].SetActive(false);

}

if (a == b)

{

primer.text = "молодец!";

timer = delay;

}

else { primer.text = "попробуй еще раз"; }

}

public void OnBackButton()

{

SceneManager.LoadScene(0);

}

}

**А.3 Исходный код num**

Здесь представлен код работы с уровнем "Цифры"

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.SceneManagement;

using TMPro;

public class viktorina : MonoBehaviour

{

public Text primer;

public TextMeshProUGUI otv1;

public TextMeshProUGUI otv2;

public TextMeshProUGUI otv3;

public TextMeshProUGUI otv4;

public GameObject ot1;

public GameObject ot2;

public GameObject ot3;

public GameObject ot4;

public float delay;

private float timer;

private int a, b, sim, rez;

/\*public Button GetButton(string caption)

{

var go = GameObject.Find(caption);

var button = go.GetComponent<Button>();

return button;

}\*/

// Start is called before the first frame update

void Start()

{

timer = delay;

viktGenerate();

}

// Update is called once per frame

void Update()

{

int count = 0;

if(primer.text == "молодец!"){

count++;

timer -= Time.deltaTime;

if (timer <= 0)

{

timer = delay;

viktGenerate();

}

}

else if (primer.text == "попробуй еще раз")

{

timer -= Time.deltaTime;

if (timer <= 0)

{

timer = delay;

if (sim == 1) {

primer.text = a + " + " + b + " = ?";

}

else { primer.text = a + " - " + b + " = ?"; }

}

}

}

void viktGenerate() {

int[] x = new int[4];

//for (int i = 0; i < 10; i++)

sim = Random.Range(0, 2);

//Debug.Log(sim);

if ( sim == 1) {

a = Random.Range(0, 10);

b = Random.Range(0, 10);

rez = a + b;

primer.text = a + " + " + b + " = ?";

int i = 0;

while (i < 4)

{

int y = Random.Range(0, 19);

if (y != rez)

{

x[i] = y;

i++;

}

}

int truf = Random.Range(0, 4);

x[truf] = rez;

}

else{

bool t = true;

while (t) {

a = Random.Range(0, 10);

b = Random.Range(0, 10);

rez = a - b;

if (rez > 0)

{

t = false;

}

}

primer.text = a + " - " + b + " = ?";

int i = 0;

while (i < 4)

{

int y = Random.Range(0, 18);

if (y != rez && y!= x[0] && y!= x[1] && y!= x[2] && y!= x[3] )

{

x[i] = y;

i++;

}

}

int truf = Random.Range(0, 3);

x[truf] = rez;

}

/\*or (int i = 0; i < x.getLenght(); i++) {

if (x[i] != rez) {

x[i] = Random.Range(0, 18);

}

}\*/

otv1.text = x[0] + "";

otv2.text = x[1] + "";

otv3.text = x[2] + "";

otv4.text = x[3] + "";

}

public void OnButton1Click()

{

if (rez == int.Parse(otv1.text)) {

primer.text = "молодец!";

timer = delay;

}

else { primer.text = "попробуй еще раз"; }

}

public void OnButton2Click()

{

if (rez == int.Parse(otv2.text))

{

primer.text = "молодец!";

timer = delay;

}

else { primer.text = "попробуй еще раз"; }

}

public void OnButton3Click()

{

if (rez == int.Parse(otv3.text))

{

primer.text = "молодец!";

timer = delay;

}

else { primer.text = "попробуй еще раз"; }

}

public void OnButton4Click()

{

if (rez == int.Parse(otv4.text))

{

primer.text = "молодец!";

timer = delay;

}

else { primer.text = "попробуй еще раз"; }

}

public void OnBackButton()

{

SceneManager.LoadScene(0);

}

}

**А.4 Исходный код ScenLoad**

Здесь представлен код работы с переходом между сценами

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class ScenLoad : MonoBehaviour

{

public void OnViktorinButton()

{

SceneManager.LoadScene(3);

}

public void OnNumButton()

{

SceneManager.LoadScene(1);

}

public void OnMoreLessButton()

{

SceneManager.LoadScene(2);

}

public void OnBackButton()

{

SceneManager.LoadScene(0);

}

public void OnInfButton()

{

SceneManager.LoadScene(4);

}

public void OnExitButton()

{

Application.Quit();

}

}