

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

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

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

РАБОТА ПРОВЕРЕНА

Рецензент,
Программист СКЦ,
ЛСМ ЮУрГУ (НИУ)

_____ В.И. Козырев
“ ____ ” _____ 2018 г.

ДОПУСТИТЬ К ЗАЩИТЕ

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

_____ Л.Б. Соколинский
“ ____ ” _____ 2018 г.

**РАЗРАБОТКА КОМПЬЮТЕРНОЙ ИГРЫ
В ЖАНРЕ “ГОЛОВОЛОМКА”
ДЛЯ ОС ANDROID**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.03.02.2018.115-013.ВКР

Научный руководитель,
преподаватель кафедры СП
_____ К.Ю. Беседин

Автор работы,
студент группы КЭ-401
_____ А.Б. Локонцев

Ученый секретарь
(нормоконтролер)
_____ О.Н. Иванова
“ ____ ” _____ 2018 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	6
1.1. Анализ аналогичных проектов	6
1.2. Анализ существующих решений для реализации проекта.....	7
2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ	11
2.1. Общие сведения	11
2.2. Жанр и аудитория	11
2.3. Основная концепция игры.....	11
2.4. Интерфейс	12
2.5. Цель игры	14
2.6. Игровые возможности	14
2.7 Варианты использования игрового приложения	14
3. РЕАЛИЗАЦИЯ ИГРЫ	16
3.1. Файловая структура игры.....	16
3.2. Реализация компонентов игры	16
3.3. Игровые сцены	21
4. ТЕСТИРОВАНИЕ СИСТЕМЫ.....	24
ЗАКЛЮЧЕНИЕ	26
СПИСОК ЛИТЕРАТУРЫ	27
ПРИЛОЖЕНИЕ	29

ВВЕДЕНИЕ

Актуальность

В наше время почти у каждого второго человека есть смартфон, они стали неотъемлемой частью нашей жизни. Раньше не каждый человек, у которого среднестатистическая зарплата, мог позволить себе приобрести смартфон, но теперь ситуация кардинально изменилась. Дети, начиная с самых маленьких, начинают знакомиться с портативными устройствами, в том числе смартфонами, у многих с самого раннего детства появляются телефоны личного пользования. Люди более старшего поколения начали привыкать к жизни со смартфонами [3].

С течением времени, все больше людей покупают себе новейшие смартфоны для личного пользования, т.к. компании, изготавливающие телефоны, с каждым годом стараются сделать товар более доступным в ценовом сегменте и более мощным, по сравнению с предыдущими моделями.

Раньше телефон использовался, исключительно, как средство связи, а не развлечения, но теперь все наоборот. С каждым годом стали появляться все больше и больше приложений, и игр для развлечения. Именно поэтому разработка игр для мобильного телефона в наше время является хорошо оплачиваемой работой.

Актуальность разработки игр для смартфонов ОС Android подтверждается статистикой. В конце 2017 года распределение на рынке мобильных ОС доля Android устройств составляла 87,7 % рынка, а IOS – 12,1 %. Благодаря широкой линейке продукции от мировых брендов Samsung, Acer, HTC и многих других в различных ценовых категориях, что способствует распространению смартфонов на базе Android среди всех слоев населения – от студентов до руководителей крупных компаний. Так как Android смартфоны доминируют на мировом рынке, количество пользователей этой системы будет только увеличиваться с каждым новым годом.

Головоломка – это жанр игр, которые требуют от игрока задействовать его интуицию, стратегию и логику. Существует множество разновид-

ностей головоломок, начиная от настольных, заканчивая популярными телепередачами.

Цель и задачи

Целью данной работы является разработка игры в жанре «головоломка» для ОС Android.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ аналогичных игровых приложений, выделить особенности разрабатываемой игры;
- спроектировать приложение;
- реализовать игру;
- протестировать игровое приложение.

Структура и объем работы

Работа состоит из введения, четырех глав, заключения, библиографии и приложения. Объем работы составляет 28 страниц, объем библиографического списка – 15 источников, объем приложения – 2 страницы.

Содержание работы

В первой главе, «Анализ предметной области», рассматриваются примеры существующих игр в жанре «головоломка» для Android, выделены особенности разрабатываемой игры и ее интерфейса.

Вторая глава содержит определение функциональных требований к разрабатываемой игре, диаграмму вариантов использования. Представляется проектирование интерфейса.

В третьей главе, представлена файловая структура приложения.

В четвертой главе, «Тестирование игрового приложения», проведено функциональное тестирование разрабатываемой игры.

В заключении сделаны выводы о проделанной работе.

В приложении продемонстрирован графический интерфейс и игровой процесс.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Анализ аналогичных проектов.

Unblock me [10]

Игра в жанре головоломка. Целью данной игры является как можно быстрее вывести красный блок с поля за минимальное число ходов, при этом перемещая другие блоки. В игре присутствуют четыре уровня сложности – от уровня для «Новичков» до «Профессионала». Всего, в данной игре, более десяти тысяч головоломок, и, чтобы пройти данную игру, понадобится очень много времени. Игра была выпущена в 2013 году, имеет очень много положительных отзывов и более пятидесяти миллионов установок. Средняя оценка игры в PlayMarket – 4,3 из 5.

Hexa Puzzle [9]

Игра в жанре головоломка. Это очень простая и увлекательная игра, как для самых маленьких, так и для старшей аудитории. Целью данной игры является перемещение блоков, чтобы заполнить какую-либо фигуру, чтобы попасть на следующий уровень. Игра была выпущена в 2017 году, пользуется наибольшей популярностью у Android-устройств и насчитывает больше пятидесяти миллионов установок. Средняя оценка игры в PlayMarket – 4,5 из 5.

2048 [8]

Игра в жанре головоломка. Цель игры состоит в том, чтобы игрок передвигал квадраты в одном направлении, после чего будет появляться еще один квадрат ценностью в два очка. Соединяя квадраты с одинаковым значением, будет получаться сумма. Игра подходит для всех возрастов, начиная от 3 лет и до людей более старшего возраста. Игра была создана в 2015 году, у игры очень много положительных отзывов и более шестидесяти миллионов установок. Средняя оценка игры в PlayMarket – 4,4 из 5.

После анализа данных приложений было сформировано мнение, что подобного жанра игры очень интересны аудитории с Android-устройствами [7].

Основываясь на результатах этого анализа, было сформировано представление об основных принципах этого жанра игр.

Все данные, полученные на основе анализа, позволяют правильно спроектировать и реализовать игровое приложение.

1.2. Анализ существующих решений для реализации проекта

На сегодняшний день существует огромное количество как платных, так и бесплатных инструментов по разработке игр, начиная от простых библиотек для известных языков программирования, заканчивая крупными редакторами с обширным функционалом.

Чтобы создать качественную игру, требуется знать, какой инструмент для разработки приложения придется использовать, в процессе создания игры. Поэтому было решено использовать уже готовый инструмент для разработки игр, который успел себя зарекомендовать, как и на Российском рынке, так и за рубежом.

Проведем обзор наиболее популярных игровых платформ, которые имеют обширный функционал для разработки игр, в которых используются наиболее популярные языки программирования, а также имеются хорошие отзывы.

Unreal Engine 4 – это инструмент для разработки игр, который был создан в 1998 году, компаний Epic Games. Данная платформа была написана на языке C++ и поддерживает работоспособность на операционных системах: Microsoft Windows, OSX и Linux [12].

Платформа использует упрощенное портирование; использует модульную систему зависимых компонентов; имеется поддержка разных систем рендеринга, воспроизведения звука, модули для работы с сетью, а также поддержка самых разных устройств ввода. Позволяет создавать игры для множества операционных систем и платформ: Mac OS, Microsoft Windows, Linux, Android, iOS и многие других.

Преимущества:

- 1) полный доступ к исходным кодам;
- 2) система Blueprint для написания игровой логики;
- 3) компиляция C++ кода во время тестирования игрового приложения;
- 4) содержит множество полезных функций;
- 5) пользуется большой популярностью у программистов.

Недостатки:

- 1) высокие системные требования;
- 2) сложность в освоении для новых пользователей;
- 3) плохая оптимизация игровой физики и теней.

Unity3d – это современная среда разработки для создания игр и приложений, разработанная компанией Unity Technologies. Данная среда разработки позволяет создавать приложения и игры не только для компьютеров, работающих на операционных системах: Microsoft Windows и Mac OS, но и для мобильных устройств на базе: iOS, Android, Windows Phone, игровых консолях: Sony PlayStation, Xbox, Wii, и других девайсов. В Unity используются простые, для освоения, языки программирования: C# и JavaScript [11]. Огромной плюс данной платформы – это простота разработки приложений и игр для мобильных устройств, как от самых простых в разработке, так и до самых сложных. В Unity есть возможность запускать любое приложение в веб-плагине.

Преимущества:

- 1) самое большое комьюнити в мире среди всех существующих платформ, для разработки игр и приложений;
- 2) хороший игровой редактор;
- 3) огромное количество инструментов для разработки приложений и игр;
- 4) присутствует внутренний магазин, где можно найти много всего полезного для создания своего проекта;

5) большое количество библиотек, которые позволяют расширить функционал платформы;

6) быстрая компиляция.

Недостатки:

1) у разработчиков нет доступа к исходному коду платформы на системном уровне;

2) условно-бесплатный.

Defold – кросс-платформенный инструмент для разработки 2D игр, выпущенный компанией King в 2009 года, но в бета-тестировании только с марта 2016 года. В состав Defold входит: редактор карт, редактор физики и многое другое. Есть возможность разрабатывать игры подключая других разработчиков с помощью email на сервисе Yahoo или Gmail. В данном инструменте не используется программирование на прямую, все игры создаются с помощью скриптов написанных на языке программирования LUA. Позволяет создавать игры для таких платформ, как: Microsoft Windows, Mac OS, Linux, iOS, Android и HTML5 [6].

Преимущества:

1) мультиплатформенность;

2) полностью бесплатный и компания King не собирается монетизировать данный инструмент разработки;

3) прост в освоении;

4) платформа нацелена на мобильные игры.

Недостатки:

1) маленькое комьюнити;

2) сложно найти обучающие материалы;

3) поддержка только одного языка программирования;

4) закрыт доступ к исходному коду платформы.

После анализа трех наиболее подходящих платформ для разработки мобильных игр на базе ОС Android, было принято решение выбрать Unity3d.

Unity3d поддерживает язык программирования C#, обладает наибольшим числом достоинств, а так же есть представление о разработке игр в данное платформе.

В табл. 1 представлены сравнения платформ для разработки игр

Табл. 1. Анализ платформ для разработки игр

Платформа для разработки	Мультиплатформенность	Язык программирования	Лицензия	Открытый исходный код
Unreal Engine	+	C++	Free/1500\$ pro	-
Unity3d	+	C#/JavaScript	Free/3000\$ pro	-
Defold	+	LUA	Free	-

2.ПРОЕКТИРОВАНИЕ

2.1. Общие сведения

Название данной игры – MIND, где игрок тренирует свою логику, мышление и рациональность действий. Игрок должен правильно анализировать ситуацию на игровом поле, чтобы как можно правильной складывать плитки с цифрами между собой. Чтобы выиграть в игре MIND требуется набрать на плитке число 2048.

2.2. Жанр и аудитория

Игра MIND относится к жанру головоломки.

Игра ориентирована на аудиторию людей старше десяти лет и подходит для русскоговорящих людей. Игра будет интересна для людей, которым нравится мыслить нестандартно и искать наиболее правильные решения, даже в самых сложных ситуациях.

2.3. Основная концепция игры

Игра MIND разработана для устройств на базе операционной системы Android.

Основная задача игрока – правильно перемещать игровые плитки на игровом поле, чтобы они складывались между собой при условии, что на плитках одинаковые числа.

Игрок может перемещать «плитки» вверх, вниз, вправо и влево. При каждом перемещении на игровом поле появляется одна новая плитка либо с числом два, либо с числом три.

Если игровое поле будет заполнено плитками, и, возможности сложить данные плитки не будет, то игра будет проиграна.

Как только игрок складывает плитки между собой, то ему начисляется количество очков, зависящее от суммы, полученной после сложения плиток и от количества плиток, которые сложились между собой за одно действие игрока [1].

Игра является не одноразовой, так как чтобы добиться победы, игрок может использовать разные стратегии и разрабатывать собственную тактику, для достижения выигрыша.

2.4. Интерфейс

Дизайн меню игры состоит из двух частей:

- 1) кнопка начала новой игры;
- 2) кнопка настроек;
- 3) фон игры;
- 4) поле, где написано название игры.

При нажатии кнопки начала игры начинается новая игра.

При нажатии кнопки настроек игрок может изменить громкость звука в игре.

Схема расположения элементов интерфейса меню изображена на рис. 1.

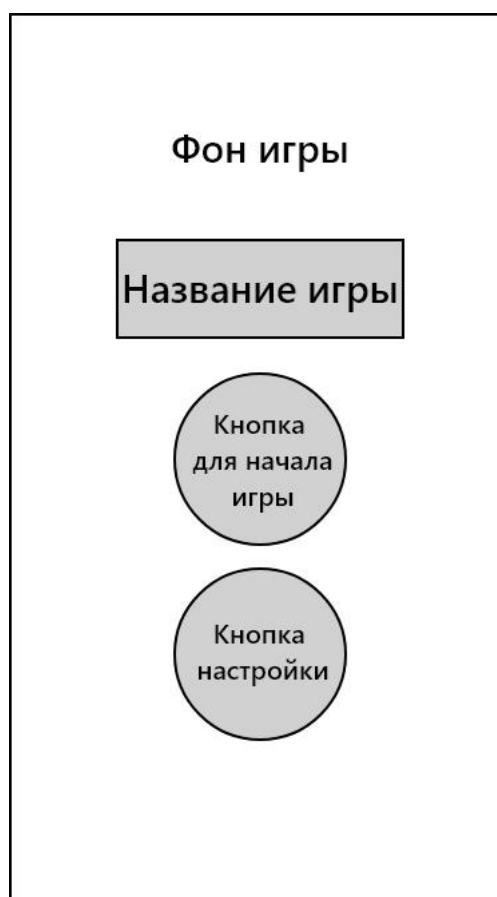


Рис. 1. Пример интерфейса меню

Дизайн игрового процесса состоит из 4 частей:

- 1) поле для текущих очков;
- 2) кнопка паузы;
- 3) поле генерации плиток размером четыре на четыре;
- 4) поле для лучшего результата за все игры.

В поле текущих очков отображаются очки, которые набрал игрок за текущую игру.

При нажатии кнопки паузы игроку предоставляется возможность: продолжить игру, начать игру заново и выйти в главное меню.

В поле генерации плиток, в начале игры, появляется две случайные плитки с цифрами либо два, либо 4. Затем, после совершения хода, появляется новая плитка с цифрой и длится это до тех пор, пока поле полностью не будет заполнено плитками с цифрами.

В поле лучшего результата отображается наиболее количество очков, которые игрок набрал за все сыгранные игры.

Схема расположения элементов интерфейса изображена на рис. 2.

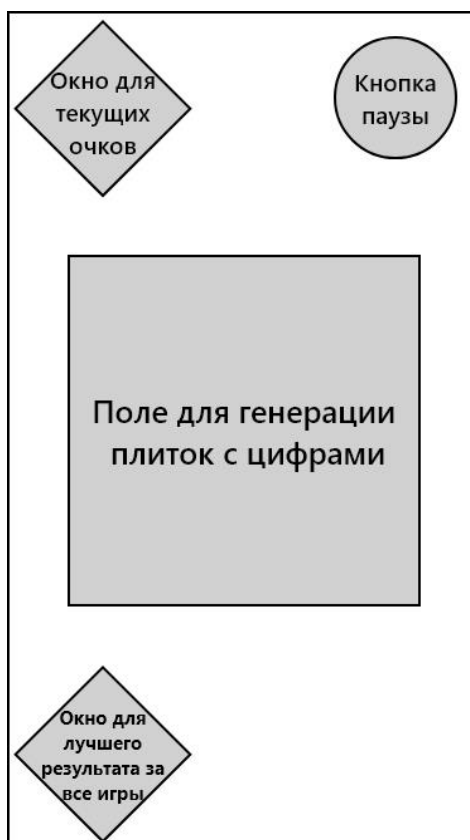


Рис. 2. Пример интерфейса

2.5 Цель игры

Цель игры – собрать плитку с числом 2048.

2.6 Игровые возможности

Игрок может перемещать плитки в направлениях: вверх, вниз, влево и вправо. Игрок может ставить игру на паузу, а так начинать новую игры, и выходить в главное меню.

2.7 Варианты использования игрового приложения

На основе концепции игры, была разработана UML диаграмма вариантов использования (use case) игрового меню, которое присутствует в каждой игре, где игрок может начать игры, зайти в настройки или совершить ряд других действий, доступных в главном меню [5]. Игрок – единственный актер системы [15]. Варианты использования игрового приложения представлены на рис. 3.

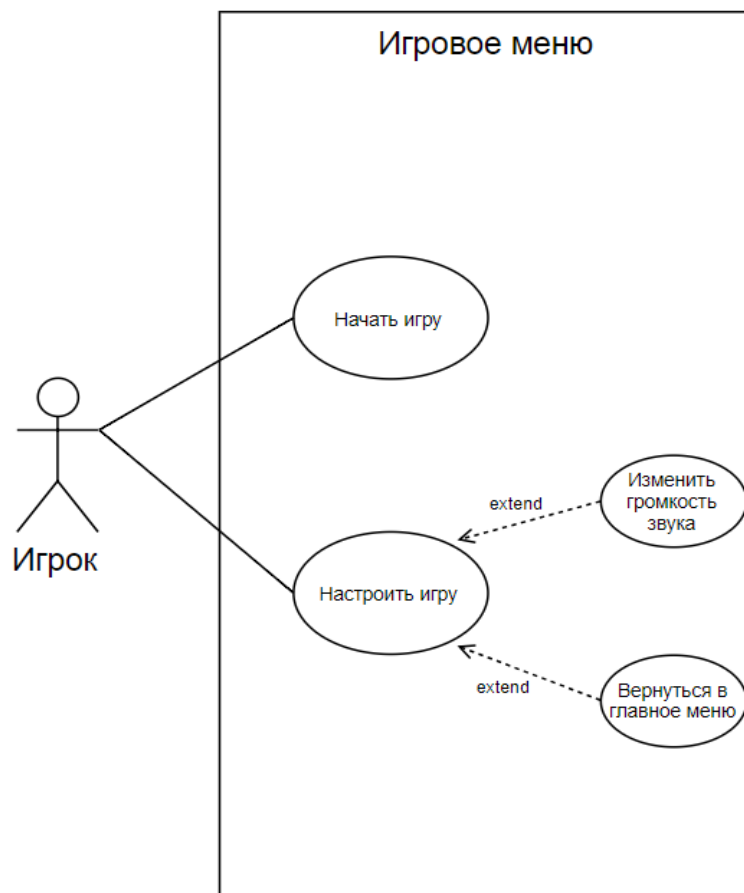


Рис. 3. Диаграмма вариантов использования в игровом меню

На основе концепции игры была разработана UML диаграмма вариантов использования в игровой сцене, представленная на рис. 4.

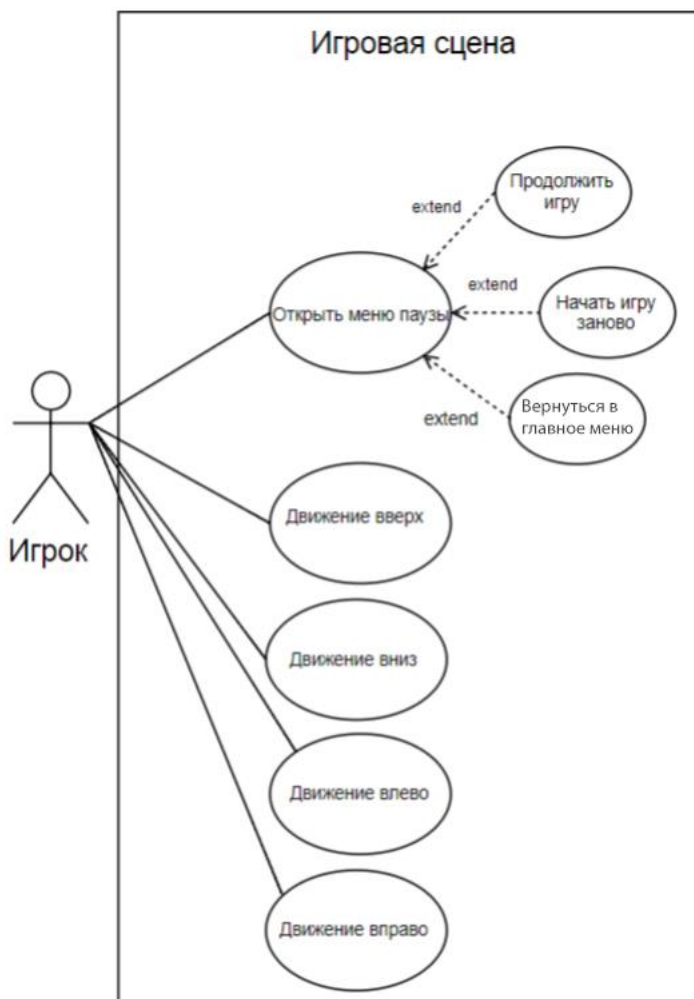


Рис. 4. Диаграмма вариантов использования в игровой сцене

3. РЕАЛИЗАЦИЯ ИГРЫ

3.1. Файловая структура

В результате разработки игрового приложения была создана файловая структура, содержащая ряд каталогов, которые включают в себя все файлы и приложения, такие как:

- 1) скрипты;
- 2) звуки;
- 3) шрифты;
- 4) изображение интерфейса, иконок и кнопок.

На рис. 5 представлена файловая структура игрового приложения.

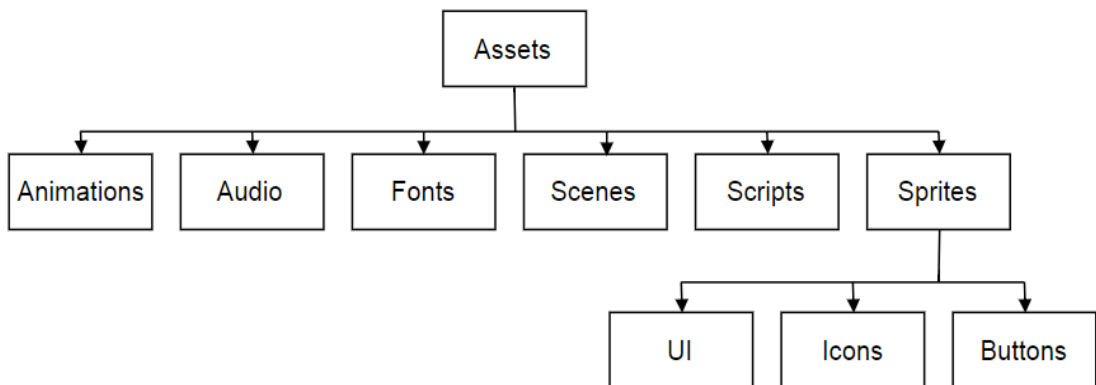


Рис. 5. Файловая структура игрового приложения

3.2. Реализация компонентов игры

На рис. 6 представлена диаграмма классов игрового приложения.

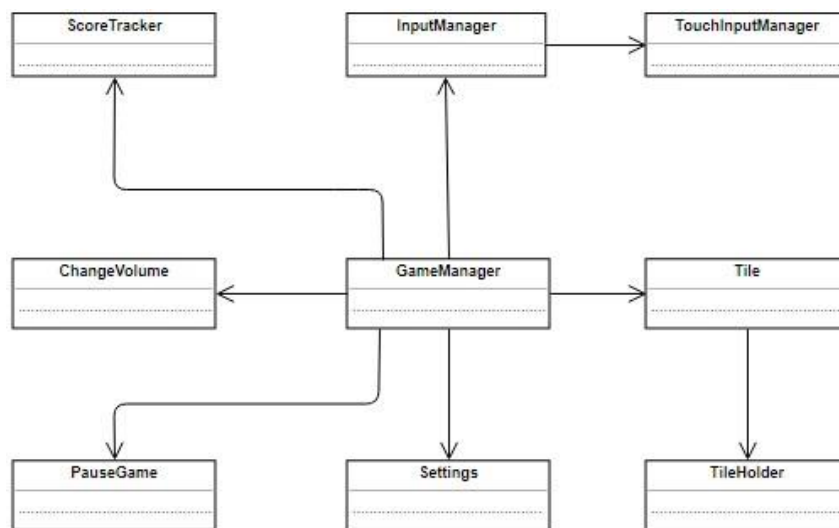


Рис. 6 Диаграмма классов

Исходный код метода класса `InputManager`, отвечающий за передвижение плиток по игровому полю представлен на рис. 7. Игрок управляет передвижением плиток посредством нажатия левой, верхней, нижней и правой стрелки на клавиатуре, которое реализовано следующим образом:

- 1) если нажата правая стрелка, игрок перемещает плитки в правую сторону игрового поля;
- 2) если нажатая левая стрелка игрок перемещает плитки в левую сторону игрового поля;
- 3) если нажата верхняя стрелка, игрок перемещает плитки в верхнюю часть игрового поля;
- 4) если нажата нижняя стрелка, игрок перемещает плитки в нижнюю часть игрового поля.

```
void Update ()
{
    if (gm.State == GameState.Playing)
    {
        if (Input.GetKeyDown (KeyCode.RightArrow))
        {
            gm.Move (MoveDirection.Right);
        }
        else if (Input.GetKeyDown (KeyCode.LeftArrow))
        {
            gm.Move (MoveDirection.Left);
        }
        else if (Input.GetKeyDown (KeyCode.UpArrow))
        {
            gm.Move (MoveDirection.Up);
        }
        else if (Input.GetKeyDown (KeyCode.DownArrow))
        {
            gm.Move (MoveDirection.Down);
        }
    }
}
```

Рис. 7. Реализация класса `InputManager`

Для подсчета количества очков, которые начисляются игроку за сложение плиток на игровом поле был разработан класс `ScoreTracker`, ис-

ходный код которого представлен на рис. 8. Принцип работы данного класса:

1) если текущий счет выше рекордного, то происходит установка рекорда на текущее значение;

2) обновляется информация о текущем и рекордном счете на экране, при завершении игр текущий счет сохраняется.

```
public class ScoreTracker : MonoBehaviour {

    private int score;
    public static ScoreTracker Instance;
    public Text ScoreText;
    public Text HighScoreText;

    public int Score
    {
        get
        {
            return score;
        }
        set
        {
            score = value;
            ScoreText.text = score.ToString();

            if (PlayerPrefs.GetInt("HighScore") < score)
            {
                PlayerPrefs.SetInt("HighScore", score);
                HighScoreText.text = score.ToString();
            }
        }
    }

    void Awake()
    {
        Instance = this;

        if (!PlayerPrefs.HasKey("HighScore"))
            PlayerPrefs.SetInt("HighScore", 0);

        ScoreText.text = "0";
        HighScoreText.text =
        PlayerPrefs.GetInt("HighScore").ToString();
    }
}
```

Рис. 8. Принцип работы класса ScoreTracker

Для того чтобы игрок мог начать игру заново, продолжить текущую игру или вернуться в главное меню был разработан класс PauseGame, исходный код которого представлен на рис. 9. Принцип работы класса PauseGame:

- 1) если игрок нажимает кнопку `Escape`, то игра ставится на паузу, во время которой время сцены становится равно нулю;
- 2) при нажатии кнопки `Restart` загружается сцена новой игры;
- 3) как только игрок нажимает на кнопку `Continue`, то игра продолжается и время сцены становится равно единице;
- 4) если игрок нажмет на кнопку `Back to menu`, то загружается сцена **Menu**.

```
public class PauseGame : MonoBehaviour {

    private bool isPaused = false;
    public GameObject PP;

    private void Update()
    {
        if(Input.GetKeyDown(KeyCode.Escape) && !isPaused)
        {
            PP.SetActive(true);
            Time.timeScale = 0;
            isPaused = true;
        }
        else if (Input.GetKeyDown(KeyCode.Escape) && isPaused)
        {
            PP.SetActive(false);
            Time.timeScale = 1;
            isPaused = false;
        }
    }
    public void pauseOn()
    {
        PP.SetActive(true);
        Time.timeScale = 0;
        isPaused = true;
    }
    public void _Continue()
    {
        PP.SetActive(false);
        Time.timeScale = 1;
        isPaused = false;
    }
    public void BackToMenu()
    {
        Time.timeScale = 1;
        SceneManager.LoadScene("Menu");
    }
    public void _Restart()
    {
        Time.timeScale = 1;
        SceneManager.LoadScene("Game");
    }
}
```

Рис. 9. Принцип работы класса `PauseGame`

Для регулирования громкости звука был разработан класс ChangeVolume, исходный код которого представлен на рис. 10. В данном классе реализуется регулировка громкости звука в игре [2].

```
public class ChangeVolume : MonoBehaviour {  
  
    private AudioSource audioSrc;  
    private float musicVolume = 1f;  
  
    void Start()  
    {  
        audioSrc = GetComponent<AudioSource>();  
    }  
  
    void Update()  
    {  
        audioSrc.volume = musicVolume;  
    }  
  
    public void SetVolume(float vol)  
    {  
        musicVolume = vol;  
    }  
  
}
```

Рис. 10. Реализация класса ChangeVolume

Для перехода между сценами был разработан класс Settings, исходный код которого представлен на рис. 11. Принцип работы класса:

- 1) если игрок нажимает кнопку настройки, то переходит в сцену настроек;
- 2) если нажата кнопка Play, то игрок переходит в сцену игры;
- 3) если игрок нажимает на кнопку возврата в сцене настроек, то происходит переход в главное меню.

```
public class Settings : MonoBehaviour {  
    public void LoadSettings()  
    {  
        Application.LoadLevel("Settings");  
    }  
    public void LoadPlayGame()  
    {  
        Application.LoadLevel("Game");  
    }  
    public void BacktoMenu()  
    {  
        Application.LoadLevel("Menu");  
    }  
  
}
```

Рис. 11. Принцип работы класса Settings

Для реализации цвета плиток и цвета текста с разными числами был реализован класс `TileStyleHolder`, исходный код которого представлен на рис. 12.

```
public class TileStyle
{
    public int Number;
    public Color32 TileColor;
    public Color32 TextColor;
}

public class TileStyleHolder : MonoBehaviour {

    public static TileStyleHolder Instance;

    public TileStyle[] TileStyles;

    void Awake()
    {
        Instance = this;
    }
}
```

Рис. 12. Принцип работы класса `TileStyleHolder`

3.3. Игровые сцены

Для реализации игрового приложения использовалась система сцен, которую представляет система `Unity3d` [13]. Сцена состоит из набора различных игровых объектов. В каждой сцене игрок может использовать функции, которые были добавлены. Приложение было разбито на три независимых сцены:

- 1) сцена главного меню;
- 2) сцена настроек игры;
- 3) сцена игры;

Главное меню – первая сцена, которая появляется перед игроком, после запуска игры. Из нее игрок может перейти в сцену игры, где происходит игровой процесс, а также может перейти в сцены настроек, где можно регулировать громкость звука на всех сценах. Сцена главного меню представлена на рис. 13.



Рис. 13. Сцена главного меню

В сцене настроек игрок может изменить громкость звука передвигая ползунок, а также вернуться в главное меню [14]. Сцена настроек изображена на рис. 14.

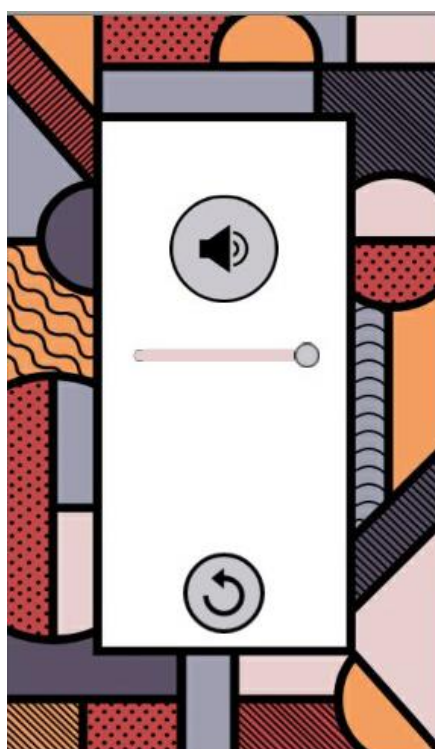


Рис. 14. Сцена настроек игры

В сцене игрового уровня осуществляется весь процесс игры. Из текущей сцены игрок может приостановить игровой процесс нажав на кнопку паузы, после чего может продолжить игру, начать новую игру и вернуться в главное меню. Сцена игры представлена на рис. 15.

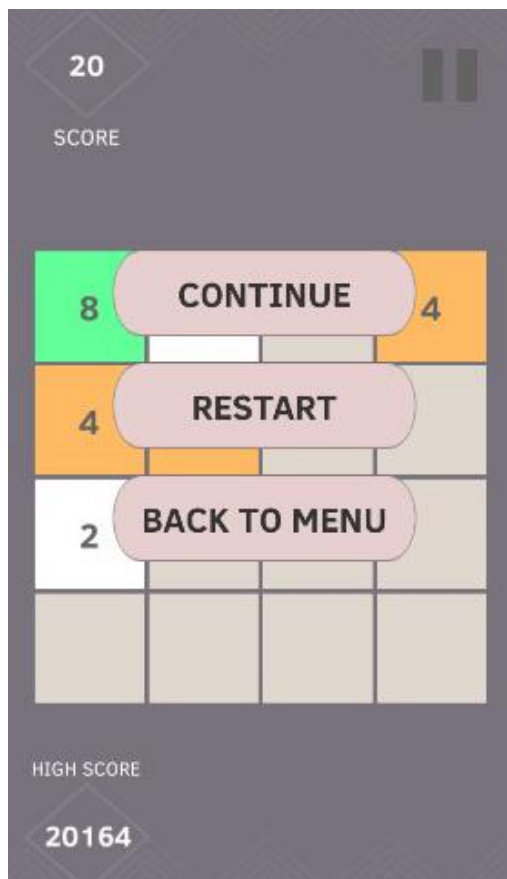


Рис. 15. Сцена игры

4. ТЕСТИРОВАНИЕ СИСТЕМЫ

Функциональное тестирование

Функциональное тестирование – это тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности ПО в определенных условиях решать задачи, нужные пользователям [4]. Результаты функционального тестирования совпали с ожидаемыми. Примеры результатов тестирования приведены в табл. 2.

Табл. 2. Результаты функционального тестирования

№	Название теста	Шаги	Ожидаемый результат	Прохождение теста
1	Разработка игры на ОС Android	Приложение запускается и полностью работает на ОС Android	Приложение запустится и будет работать на ОС Android	Да
2	Переход из главного меню в сцену игры	1. Нажать на кнопку “Play” 2. Перемещение игрока в сцену игры	При нажатии на кнопку происходит начало игры	Да
3	Переход из главного меню в сцену настроек	1. Нажать на кнопку “Настроек” 2. Перемещение игрока в сцену настроек	При нажатии на кнопку происходит переход в меню настроек	Да
4	Передвижение плиток по игровому полю	Передвижение игрока при нажатии стрелок управления	Передвижение плиток по игровому полю	Да
5	Начисление очков	При сложении плиток начисляются очки	Начисляются очки	Да

№	Название теста	Шаги	Ожидаемый результат	Прохождение теста
6	Сохранение лучшего результата	При установлении нового рекорда ,счетчик лучшего результата обновляется	Обновляется лучший результат	Да
7	Изменение громкости звука	При передвижении ползунка громкость музыки изменяется	Изменяется громкость музыки	Да

Все проделанные тесты были успешно пройдены итоговой версией приложения, следовательно, приложения работает правильно.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломной работы были выполнены следующие задачи:

- 1) проведен анализ аналогичных игровых приложений, выделены особенности разрабатываемой игры;
- 2) произведен обзор существующих платформ для разработки мобильных приложений на базе ОС Android;
- 3) выполнена разработка архитектуры приложения;
- 4) реализовано игровое приложения;
- 5) протестировано игровое приложение.

СПИСОК ЛИТЕРАТУРЫ

1. Арлоу Д., Нейштадт А. UML 2 и Унифицированный процесс 2-е издание. – Пер. с англ. – СПб.: Символ Плюс, 2007. – 624 с.
2. Добавление звука в игру на Android. [Электронный ресурс] URL: <https://itproger.com/course/unity-android/12> (дата обращения: 18.04.2018).
3. История развития компьютерных и видео игр. [Электронный ресурс] URL: <http://pro-gamer.org/blog/47.html> (дата обращения: 04.02.2018).
4. Канер С. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений: - Пер. с англ. – К.: Издательство «ДиаСофт», 2001. – 544 с.
5. Леоненков А. Самоучитель UML. Эффективный инструмент моделирования информационных систем. - СПб.: BHV, 2001 – 304 с.
6. Обзор платформы Defold. [Электронный ресурс] URL: <http://app2top.ru/industry/oleg-pridyuk-o-defold-e-to-tehnologiya-s-pomoshh-yu-kotoroj-my-sami-delaem-igry-78210.html> (дата обращения: 17.04.2018).
7. Обзор рынка мобильных приложений. [Электронный ресурс] URL: <https://www.openbusiness.ru/biz/business/obzor-rynka-mobilnykh-prilozheniy/> (дата обращения: 15.04.2018).
8. Официальный сайт GooglePlay. [Электронный ресурс] URL: <https://play.google.com/store/apps/details?id=com.ketchapp.play2048> (дата обращения: 11.03.2018).
9. Официальный сайт GooglePlay. [Электронный ресурс] URL: <https://play.google.com/store/apps/details?id=com.bitmango.go.blockhexapuzzle> (дата обращения: 20.03.2018).
10. Официальный сайт GooglePlay. [Электронный ресурс] URL: <https://play.google.com/store/apps/details?id=com.kiragames.unblockme> (дата обращения: 09.03.2018).
11. Официальный сайт Unity3d. [Электронный ресурс] URL: <https://unity3d.com/ru> (дата обращения: 24.04.2018).

12. Официальный сайт Unreal Engine. [Электронный ресурс] URL: <https://www.unrealengine.com/> (дата последнего обращения: 11.04.2018).
13. Программирование под Android в перспективе будущего. [Электронный ресурс] URL: <http://www.apps-trilian.ru/android/> (дата обращения: 16.04.2018).
14. Разрабатываем меню в Unity3d. [Электронный ресурс] URL: https://gcup.ru/publ/gamedev/razrabatyvaem_menju_v_unity3d_sozdanie_bazov_oj_chasti_chast_1/1-1-0-477 (дата обращения: 18.04.2018).
15. ХоккингД. Unity в действии. Мультиплатформенная разработка C#. – Пер. с англ. – СПб.: Питер, 2016. – 366 с.

ПРИЛОЖЕНИЕ



Рис. 1. Меню игры



Рис. 2. Настройки игры

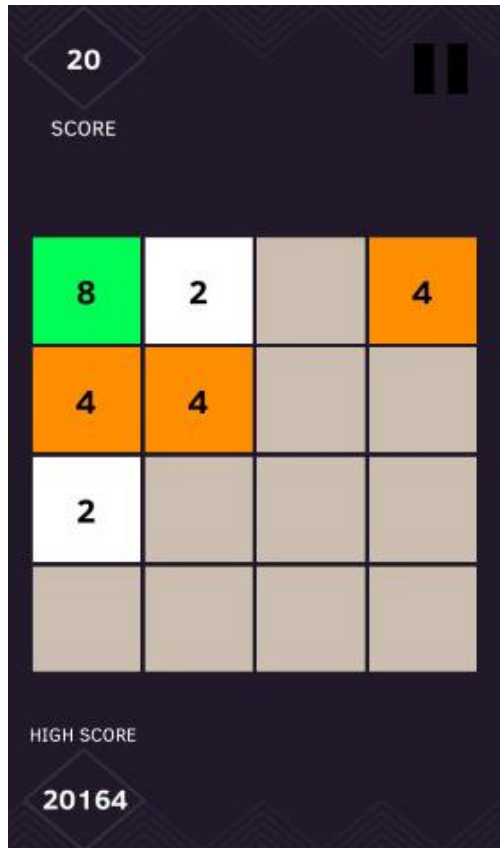


Рис. 3. Игровой процесс

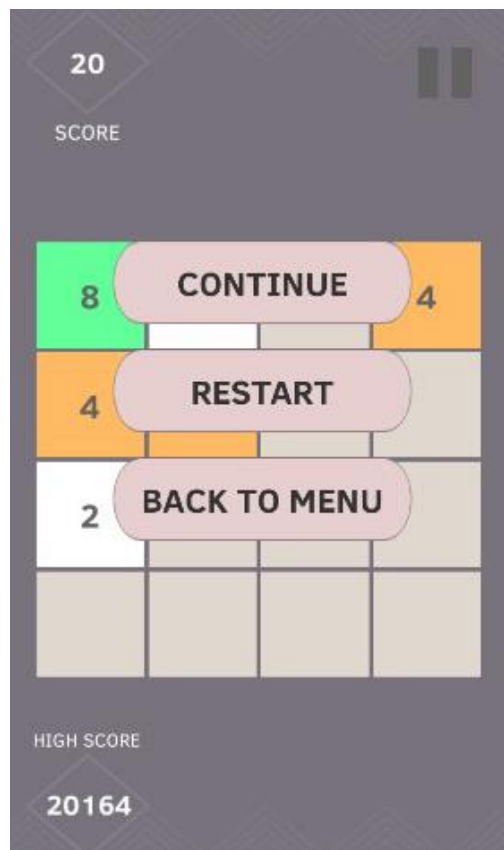


Рис. 4. Пауза в игровом процессе