СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc501531313)

[1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ 5](#_Toc501531314)

[2 ПРОГРАММНОЕ ПРОЕКТИРОВАНИЕ 7](#_Toc501531315)

[3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ](#_Toc501531316) 12

[3.1 Детальная реализация функциональных частей ПО](#_Toc501531317) 12

[3.2 Сопроводительная документация 16](#_Toc501531318)

[3.3 Анализ ПО 16](#_Toc501531319)

[3.4 Тестирование ПО 17](#_Toc501531320)

[ЗАКЛЮЧЕНИЕ 24](#_Toc501531321)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 25](#_Toc501531322)

[ПРИЛОЖЕНИЕ А 26](#_Toc501531323)

[ПРИЛОЖЕНИЕ Б 29](#_Toc501531324)

ВВЕДЕНИЕ

Темой данного курсового проекта является создание мобильной игры «Шашки».

Мобильная игра – это игровая программа для мобильных устройств. В настоящее время разработка мобильных игр становится все популярнее. Это связано с появлением улучшенных операционных и аппаратных систем, с доступностью смартфонов, а также с использованием мобильных устройств, как средство для отдыха и развлечения.

Android – как система поддерживает огромный спектр всяческих отраслей жизнедеятельности человека, начиная от всем привычных телефонов и наручных часов, и заканчивая поддержкой автомобильных развлекательных систем и бытовых роботов. Но и это далеко не весь ее потенциал, так как с повышением тенденций развития многих сфер жизнедеятельности человека, так же повышается и многофункциональность систем под управлением Android.

Отдел управления инвестициями данных Kantar опубликовал статистику распространения мобильных OC на ключевых рынках по итогам первого квартала 2018 года. Так, в Китае доля Android сейчас составляет 77,4%. В целом на большинстве основных рынков распределение долей между основными ОС практически не изменилось. В Великобритании, Франции, Италии и Японии Android за год несколько нарастила позиции (на 1-5 процентных пунктов), а это значит, что разработчиков привлекает данная отрасль. [6]

В рамках курсового проекта будет использоваться именно данная операционная система. Преимущества OS Android [1]:

1. Частично открытый исходный код;
2. Сервисы Google и синхронизация;
3. Стоимость и функциональность;
4. Быстрорастущий магазин приложений Google Play Market;
5. Удобный и интуитивно понятный интерфейс;
6. Многозадачность, когда без проблем работает одновременно несколько приложений;
7. Разнообразие игр и приложений.

Низкая стоимость, достаточный функционал, синхронизация google-аккаунтов и магазин бесплатных и платных приложений несомненно делают Android популярной площадкой для развития. Кроме этого, независимые разработчики тоже прилагают немало усилий по совершенствованию своих приложений, быстро адаптируя их под новые версии ОС.

1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ

Целью данного курсового проекта является разработка игрового приложения «Шашки» под операционную систему Android версии 4.4. и выше.

Данное приложение является развлекательным. Шашки по праву призваны одной из самых интеллектуальных игр. Она стимулирует мыслительную деятельность детей, способствует их логическому мышлению, развивает пространственное воображение, память и внимание, что очень важно для подготовки к школе. Большая часть детей в дошкольном возрасте по своей природе очень подвижны и активны, им трудно сосредоточить свое внимание – эта волшебная игра ненавязчиво формирует у них такое качество, как усидчивость. Также игра в шашки воспитывает в детях такие немаловажные для их будущей жизни качества, как умение самостоятельно думать и нести ответственность за принятое решение, адекватно относиться к неудачам и поражениям.

Правила игры в шашки следующие:

1. Шашка может ходить только вперед, а бить - и вперед, и назад.
2. Если шашка, передвигаясь по доске, достигает поля в последнем ряду, она становится дамкой.
3. Дамка может делать ходы на любое свободное поле по диагонали и вперед, и назад.
4. Шашка обязательно должна побить (снять с доски) шашку (или дамку) противника при своем ходе, если за ней есть свободное поле.
5. Если за шашкой противника есть несколько свободных полей подряд, то дамка, взяв эту шашку, может остановиться на любом из свободных полей.
6. Если можно взять несколько шашек (или дамок) разными ходами, игрок имеет право выбрать любой из ходов.
7. Игра заканчивается победой, когда все шашки противника накрыты или заблокированы.

Выбор исходной версии Android был сделан опираясь на статистику компании Google составленную в июле 2018, где доля телефонов с версией KitKat 4.4 - 9.1% от общего количества. Что фактически означает, что приоритетная доля рынка, составляющая 95.9% относится к версиям Android 4.4 и выше. [5]

Сам же рынок мобильных приложений составляет более 3 миллионов размещенных приложений. И разработчикам приходится сталкиваться с огромной конкуренцией на таких крупных Android-магазинах как GooglePlay или AmazonAppStore. Собственно, что дает огромный толчок в развитии жанров игровой индустрии и применению все более изощрённых подходов к созданию андроид приложений, таких как Pokemon Go, которая была одним из основателей жанра дополнительной реальности.

Подавляющее большинство игр написано в жанре «аркада» и «головоломка». Основным ограничением для «разнообразия жанров» является устройство ввода. Например, для удобной игры в 3D-шутер желательна возможность одновременного наведения на цель и стрельбы — использование двух клавиш одновременно, что затруднено на маленьких клавиатурах мобильного телефона, а стратегии в реальном времени изначально ориентировались на управление джойстиком или компьютерной мышью. Руководствуясь данными было принято решение создания игры в жанре «головоломка». [4]

В качестве игрового движка был выбран Unity2D, который является современным кроссплатформенным движком для создания игр и приложений. Использование данного игрового движка было охарактеризовано следующими плюсами:

1. В среду разработки Unity интегрирован игровой движок, что позволяет протестировать свою игру, не выходя из редактора;
2. Unity поддерживает импорт огромного количества различных форматов, что позволяет разработчику игры конструировать сами модели в более удобном приложении, а Unity использовать по прямому назначению — разработки продукта;
3. Написание сценариев (скриптов) осуществляется на наиболее популярных языках программирования — C# и JavaScript.

Среда разработки - Visual Studio 2017, Unity.

Язык программирования - C#.

**2** **ПРОГРАММНОЕ ПРОЕКТИРОВАНИЕ**

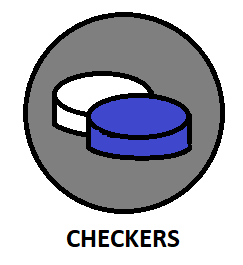
Основная задача при создании игры – максимальная адаптивность игрового кода для последующего расширения функционала. Так же при проектировании нужно учитывать особенности игровой механики, указанной в задании к курсовому проекту и простоту работы пользователя с приложением. Игровой интерфейс должен быть интуитивно понятным и визуально законченным. Так же следует помнить про адаптивность игрового пространства под различные разрешения экранов.

Цветовая гамма игры будет выполнена преимущественно из следующих цветов: розовый, желтый, молочный, салатовый, коричневый (рисунок 2.1).



**Рисунок 2.1** – Основная цветовая палитра приложения

Лого приложения будет выглядеть следующим образом (рисунок 2.2):



**Рисунок 2.2** – Лого приложения

Пользовательский интерфейс программы представлен тремя окнами:

1. Главное меню;
2. Основное поле игры;
3. Поле с информацией об авторе игры.

При запуске игры, пользователь сможет наблюдать информацию о том, с помощью чего было написано игровое приложение «Шашки» (рисунок 2.3).



**Рисунок 2.3** – Запуск приложения

После того, как игра прогрузится, откроется главное меню, которое будет состоять из двух кнопок: «Новая игра» и «Об авторе». Внешнее представление главного меню отображено на рисунке 2.4.



**Рисунок 2.4** – Графическое представление главного меню игры

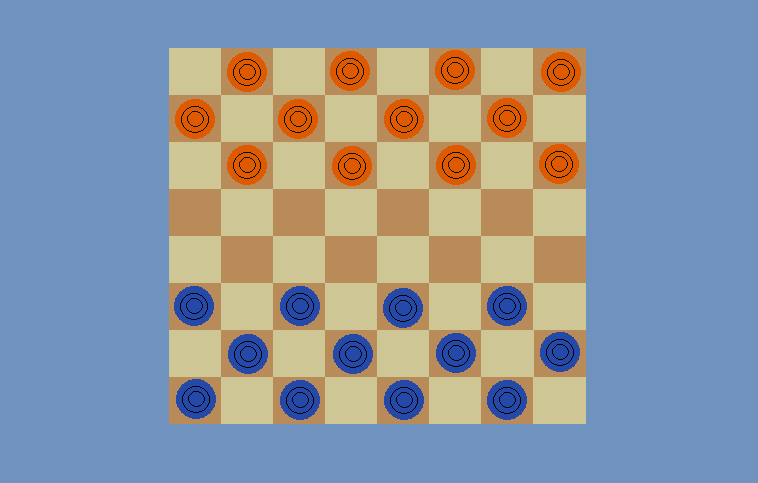
При выборе первого пункта меню «new game», пользователю будет предоставлена возможность выбрать тип игры (рисунок 2.5):

1. Локальная игра, в которую могут играть пользователи на одном телефоне.
2. Режим игры «VS Bot». Здесь уже пользователь сможет играть против виртуального соперника.



**Рисунок 2.5** – Графическое представление первого пункта меню «Новая игра»

В соответствии с требованиями при старте игры будет автоматически загружаться игровое поле с уже расставленными шашками. Графическое представление поля игры отображено на рисунке 2.6. Следует отметить, что положение игрового поля на экране адаптируется в зависимости от устройства.



**Рисунок 2.6** – Графическое представление игрового пространства игры

Одни шашки будут иметь синий цвет, другие – оранжевый. Первый ход всегда за синими шашками. Чтобы сделать ход, необходимо перетащить шашку на определённую клетку. При перетаскивании шашки выделяются серым цветом те ячейки, куда можно поставить шашку. По ходу игры можно побить одну или несколько шашек соперника, последовательно перетаскивая через них свою шашку.

Достигнув последней горизонтали на противоположной стороне доски, простая шашка должна превратиться в дамку, которая может уже перемещаться по всему игровому полю, также она должна будет изменить свой внешний вид, чтобы пользователь смог легко отличить дамку от простой шашки. Игрок может провести в дамки неограниченное число шашек.

Выигрывает партию тот, кто сумеет побить все шашки противника или лишит оставшиеся на доске чужие фишки возможности сделать ход («запрет»их). После окончания игры появится всплывающее окно с сообщением со следующей информацией: какие шашки выиграли «оранжевые» (светлые) либо «синие» (темные), количество ходов и оставшихся фигур на игровой доске (рисунок 2.7, рисунок 2.8).

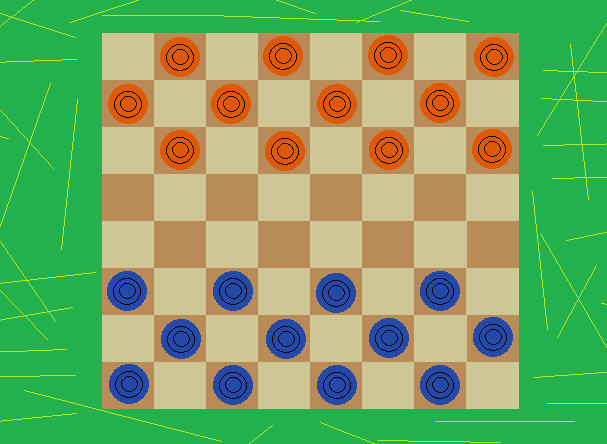


**Рисунок 2.7** – Сообщение о победе светлых



**Рисунок 2.8** – Сообщение о победе темных

Следует отметить, что при запуске игры генерируется задний фон (рисунок 2.6, рисунок 2.9). поэтому пользователь сможет увидеть один из пяти вариантов заднего фона.



**Рисунок 2.9** – Вариант фона

При выборе второго пункта меню пользователь может узнать некоторые сведения о разработчике (рисунок 2.10):

****

**Рисунок 2.10** – Графическое представление второго пункта меню «Об авторе»

Диаграмма вариантов использования представлена в приложении Б.

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

3.1 Детальная реализация функциональных частей ПО

В Unity используются сценарии - именуемые «Скрипты». Поскольку основной задачей при создании игры была максимальная адаптивность игрового кода для последующего расширения функционала. То при создании поля каждый «Скрипт» привязывается не к конкретному игровому объекту, а к его копии.

Метод инициализации игрового поля представлен в методе Create (). Функция инициализации игрового поля представлена в листинге 3.1.1. Входными параметрами функции служат X и Y, которые представляют собой изометрические координаты игрового пространства. На данных изометрических данных будет строиться позиционирование всех остальных копий игровых объектов.

**Листинг 3.1.1** – Метод Create()

1. public void Create()
2. {
3. for (int y = 0; y < 8; y++)
4. {
5. for (int x = 0; x < 8; x++)
6. {
7. GameObject newCell = Instantiate(сellPrefab, transform);
8. RectTransform rectTransform = newCell.GetComponent<RectTransform>();
9. rectTransform.anchoredPosition = new Vector2((float)((x \* xImageLength) + imageDistance), (y \* yImageLength) + imageDistance);
10. mAllCells[x, y] = newCell.GetComponent<Cell>();
11. mAllCells[x, y].Setup(new Vector2Int(x, y), this);
12. }
13. }
14. for (int x = 0; x < 8; x += 2)
15. {
16. for (int y = 0; y < 8; y++)
17. {
18. int offset = (y % 2 != 0) ? 0 : 1;
19. int finalX = x + offset;
20. mAllCells[finalX, y].GetComponent<Image>().color = new Color32(230, 220, 187, 255);
21. }
22. }
23. }

Метод PlacePieces(int startRow, int sideFlag, List<BaseObject> pieces, Board board), который инициализирует расстановку шашек на поле, представлен в листинге 3.1.2.

**Листинг 3.1.2** – Метод PlacePieces(int startRow, int sideFlag, List<BaseObject> pieces, Board board)

1. PlacePieces(int startRow, int sideFlag, List<BaseObject> pieces, Board board)
2. {
3. int column, midleColumn;
4. if (sideFlag == 0)
5. {
6. column = 0;
7. midleColumn = 1;
8. }
9. Else
10. {
11. column = 1;
12. midleColumn = 0;
13. }
14. for (int i = 0; i < 4; i++)
15. {
16. pieces[i].Place(board.mAllCells[column, startRow]);
17. pieces[i + 4].Place(board.mAllCells[midleColumn, startRow + 1]);
18. pieces[i + 8].Place(board.mAllCells[column, startRow + 2]);
19. column += 2;
20. midleColumn += 2;
21. }
22. }

Place(Cell newCell) – метод постановки конкретоной шашки на игровое поле во время игры – представлен в листинге 3.1.3.

**Листинг 3.1.3** – Метод Place(Cell newCell)

1. Place(Cell newCell)
2. {
3. mCurrentSell = newCell;
4. mOriginalSell = newCell;
5. mCurrentSell.mCurrentPiece = this;
6. transform.position = newCell.transform.position;
7. gameObject.SetActive(true);
8. }

После того, как поле задано, у игрока появляется возможность нажимать на игровые объекты. Перемещение шашки осуществляется с помощью трех методов:

1. OnBeginDrag(PointerEventData eventData) – нажатие на шашку (листинг 3.1.4);
2. OnDrag(PointerEventData eventData) – перемещение шашки (листинг 3.1.5);
3. OnEndDrag(PointerEventData eventData) – отпускание шашки (листинг 3.1.6).

**Листинг 3.1.4** – Метод OnBeginDrag(PointerEventData eventData)

1. OnBeginDrag(PointerEventData eventData)
2. {
3. base.OnBeginDrag(eventData);
4. mHighlightedCells = new List<Cell>();
5. if (mPieceManager.attackPosibilityFlag == false)
6. {
7. CheckPathing();
8. }
9. else
10. mHighlightedCells = mPieceManager.attackCells;
11. if (mPieceManager.AttackerCheck(mCurrentSell) && mPieceManager.attackPosibilityFlag == true)
12. {
13. ShowWriteCells();
14. }
15. else if(mPieceManager.attackPosibilityFlag == false)
16. {
17. ShowCells();
18. }
19. }

**Листинг 3.1.5** – Метод OnDrag(PointerEventData eventData)

1. OnDrag(PointerEventData eventData)
2. {
3. base.OnDrag(eventData);
4. transform.position += (Vector3)eventData.delta;
5. foreach(Cell cell in mHighlightedCells)
6. {
7. if(RectTransformUtility.RectangleContainsScreenPoint(cell.mRectTransform, Input.mousePosition) && cell.mOutLineImage.enabled)
8. {
9. mTargetCell = cell;
10. break;
11. }
12. mTargetCell = null;
13. }
14. }

**Листинг 3.1.6** – Метод OnEndDrag(PointerEventData eventData)

1. OnEndDrag(PointerEventData eventData)
2. {
3. base.OnEndDrag(eventData);
4. ClearCells();
5. if (!mTargetCell)
6. {
7. transform.position = mCurrentSell.gameObject.transform.position;
8. return;

Окончание листинга 3.1.6

1. }
2. Else
3. mPieceManager.attackPosibilityFlag = false;
4. Move();
5. if (mPieceManager.attackPosibilityFlag == false)
6. {
7. mPieceManager.SwitchSides(mColor);
8. AttackPosibility(mColor, "Enemy");
9. }
10. }

Метод CheckPathing(), который отображает доступные для перемещения клетки (листинг 3.1.7).

**Листинг 3.1.7** – Метод CheckPathing()

1. CheckPathing()
2. {
3. CreateCellPath(1, 1, mMovement.z);
4. CreateCellPath(-1, 1, mMovement.z);
5. CreateCellPath(-1, -1, mMovement.z);
6. CreateCellPath(1, -1, mMovement.z);
7. }

Взятие шашки описывается в следующем листинге 3.1.8. Также идет проверка на возможность повторного взятия шашки.

**Листинг 3.1.8** – Часть метода Move()

1. if (backCellState == CellState.Enemy)
2. {
3. PlaceChanger();
4. mPieceManager.RemoveFromList(mColor, backCell);
5. backCell.RemovePiece();
6. AttackPosibility(mColor, "Friendly", mCurrentSell);
7. }

Преобразование простой шашки в дамку: клетка, в которую стала шашка, является крайней с противоположной стороны доски (листинг 3.1.9).

**Листинг 3.1.9** – Часть метода Move()

1. if (mCurrentSell.mBoardPosition.y == (mColor == Color.white ? 7 : 0))
2. {
3. mPieceManager.QueenUpgarder(mCurrentSell);
4. }

Удаление шашки с поля описывается в методе Kill() (листинг 3.1.10).

**Листинг 3.1.10** – Метод Kill()

1. Kill()
2. {
3. mCurrentSell.mCurrentPiece = null;
4. gameObject.SetActive(false);
5. }

3.2 Сопроводительная документация

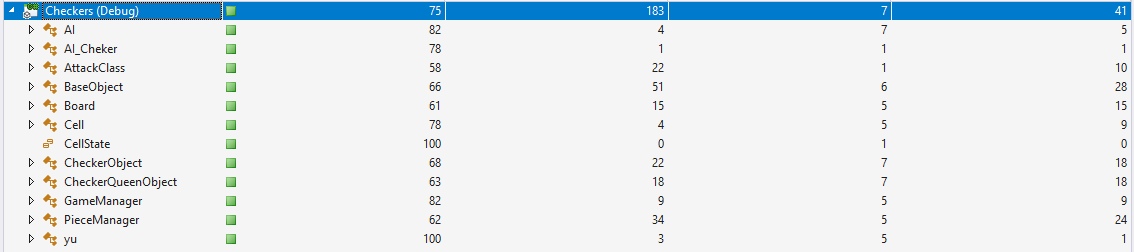
Сопроводительная документация по разработанному игровому приложению «Шашки» представлена в приложении А.

3.3 Анализ ПО

Для анализа данного программного обеспечения используем анализ метрик кода.

Метрика программного обеспечения – мера, позволяющая получить численное значение некоторого свойства программного обеспечения или его спецификаций. [4]

Microsoft предоставляет встроенное в Visual Studio средство, которое предоставляет возможность сделать анализ ПО. При анализе метрик будем учитывать следующие критерии: индекс удобства поддержки – оценивает простоту обслуживания кода, сложность организации циклов – определяет число ветвей, глубина наследования – определяет число уровней в иерархии наследования объекта, взаимозависимость классов – определяет число классов, на которые есть ссылки. Результаты метрик кода игры представлен на рисунке 3.3.1.



**Рисунок 3.3.1** — Результат метрик кода сервера

Индекс удобства поддержки кода игры составил 75 из 100, что является хорошим показателем. Сложность организации циклов – 183. Глубина наследования – 7. Взаимозависимость классов – 41.

К достоинствам этого приложения стоит отнести легкость и простоту в использовании, быстродействие, малую потребность в ресурсах системы.  
Анализ ресурсов: разработанное приложение потребляет следующее количество ресурсов: CPU – в диапазоне от 10 до 40%, оперативная память – в диапазоне от 100 до 150 мб.

3.4 Тестирование ПО

Для тестирования приложения используем SmokeTest. Smoke Test (дымовое тестирование) в тестировании программного обеспечения означает минимальный набор тестов на явные ошибки. «Дымовой тест» обычно выполняется самим программистом; не имеет смысла отдавать программу, не проходившую этот тест, на более глубокое тестирование. [4]

Список тестов, которые прошла игра, представлен ниже:

1. Запуск игры;
2. Нажатие кнопки старта игры;
3. Проверка целостности игрового поля и наличие на нем светлых и темных шашек;
4. Проверка возможности передвижения шашек по игровому полю;
5. Проверка взятия шашки;
6. Проверка преобразования простой шашки в дамку;
7. Проверка возможности начать новую игру после завершения предыдущей;
8. Проверка появления сообщения о победе;
9. Возможность зайти в раздел «About»;
10. Возможность вернуться обратно в главное меню из раздела «About».

Лого приложения выглядит следующим образом (рисунок 3.4.1):



**Рисунок 3.4.1** – Лого приложения

При запуске игры пользователь наблюдает информацию о том, с помощью чего было написано игровое приложение «Шашки» (рисунок 3.4.2).



**Рисунок 3.4.2** – Запуск приложения

Успешное отображение главного меню мобильного приложения представлено на рисунке 3.4.3

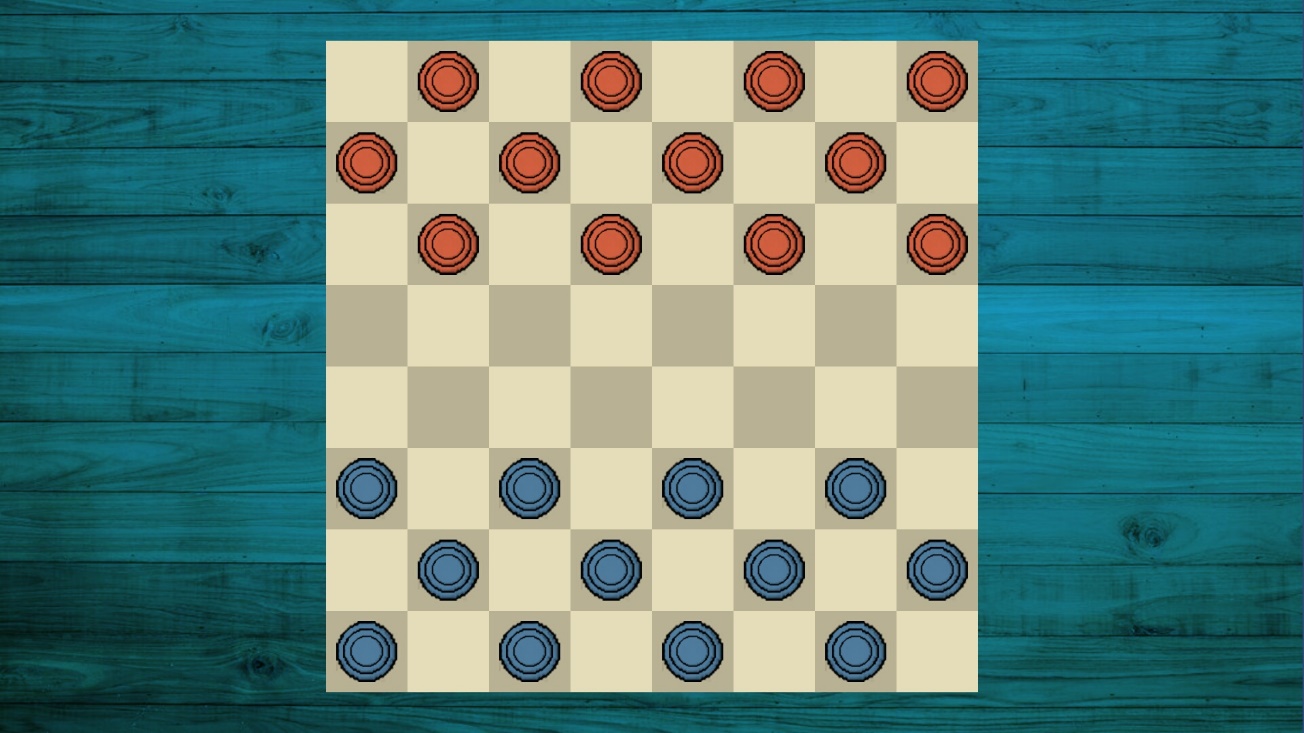


**Рисунок 3.4.3** – Успешное отображение главного меню мобильного приложения

Далее была проверена функция возможности нажатия на кнопку «new game», после нажатия которой должно открыться еще одно подменю с выбором типа игры (рисунок 3.4.4). Проверка целостности игрового поля и наличие на нем шашек. Успешное позиционирование фона, игровой доски и светлых и темных шашек представлено на рисунке 3.4.5.

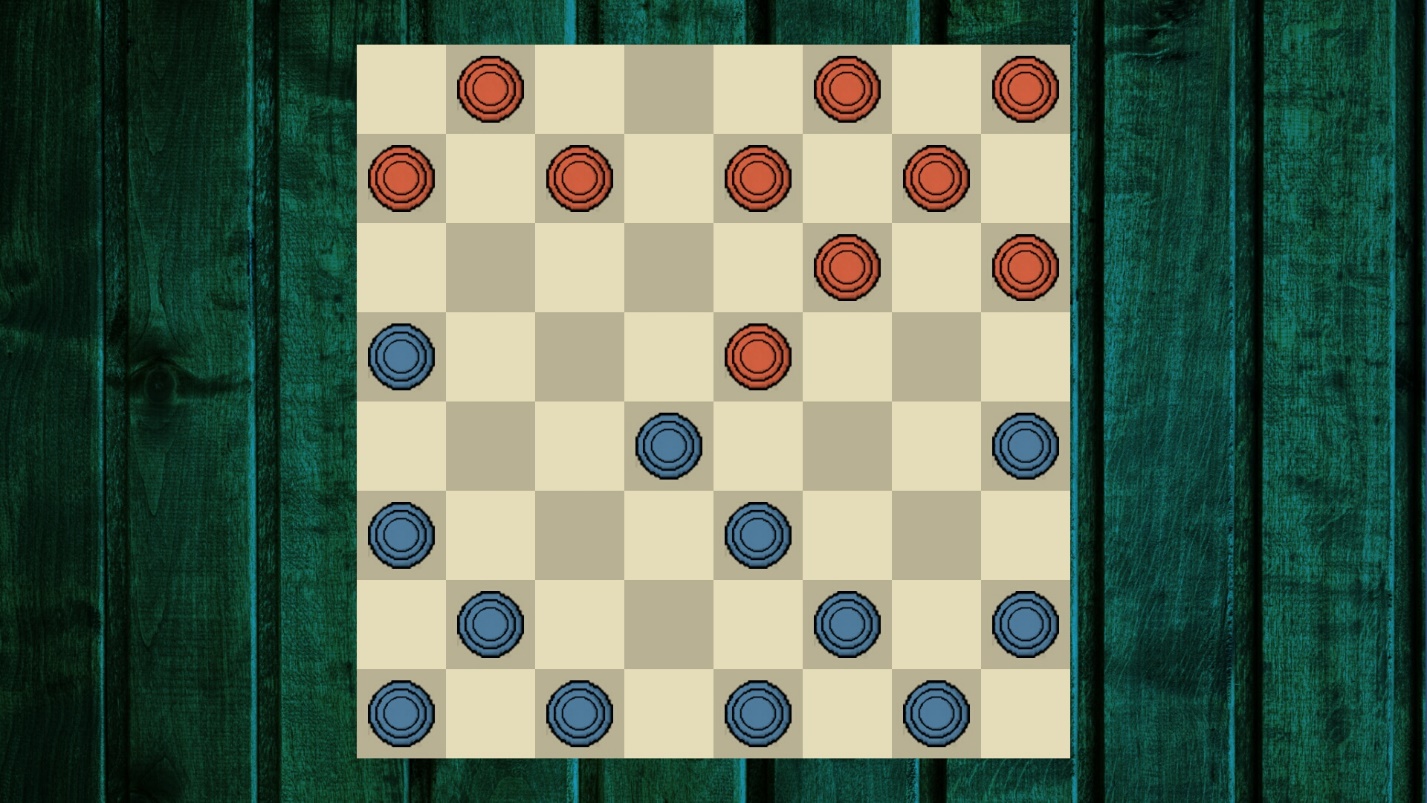


**Рисунок 3.4.4** – Корректное представление первого пункта меню «Новая игра»

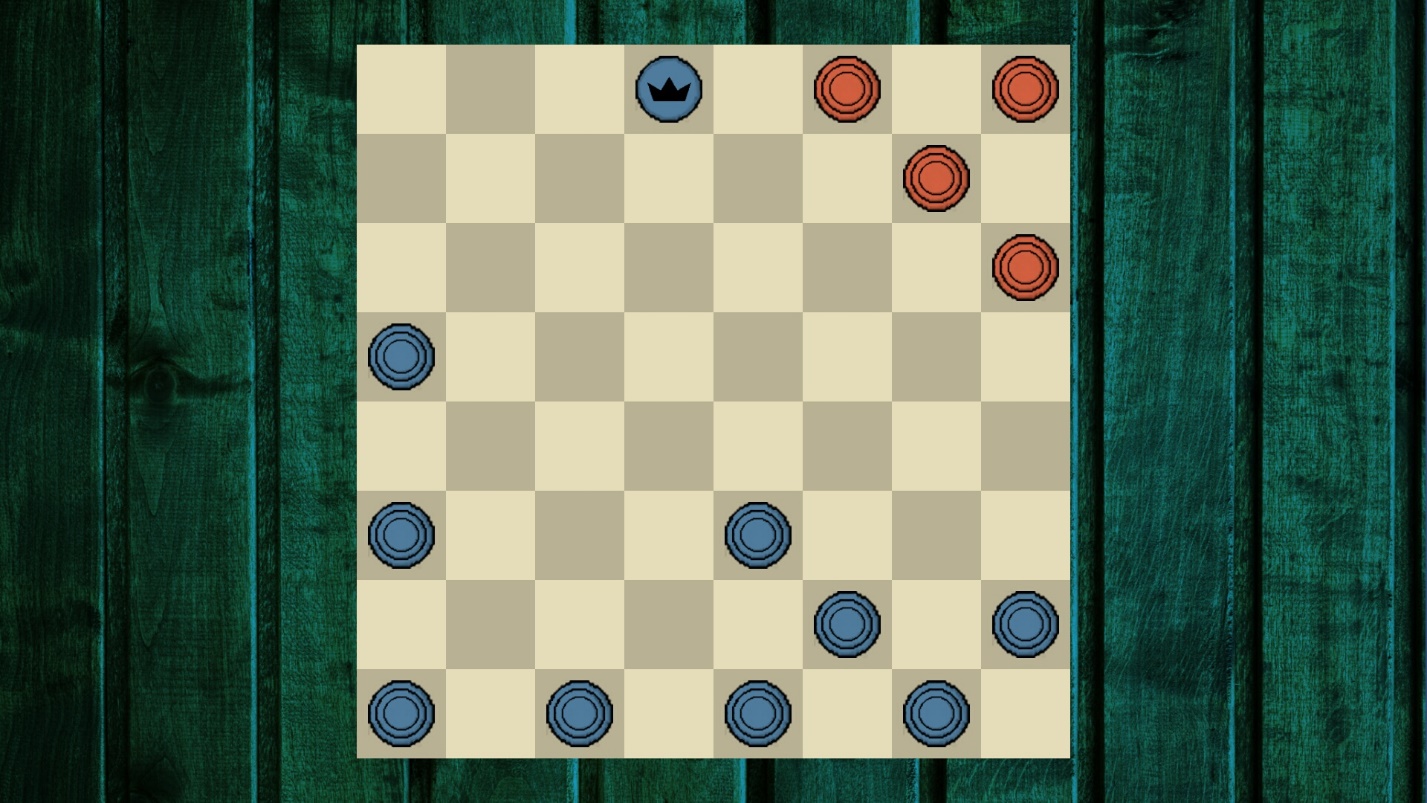


**Рисунок 3.4.5** – Успешное представление игрового пространства игры

Также было протестирование корректное перемещение шашек по игровой доске, взятие шашки соперника, преобразование простой шашки в дамку при достижении противоположной стороны доски (рисунок 3.4.6, рисунок 3.4.7).



**Рисунок 3.4.6** – Успешное перемещение шашек по игровому полю



**Рисунок 3.4.7** – Успешное преобразование простой шашки в дамку

Далее было протестировано завершение игры. Результаты приведены на рисунках 3.4.8 и 3.4.9.



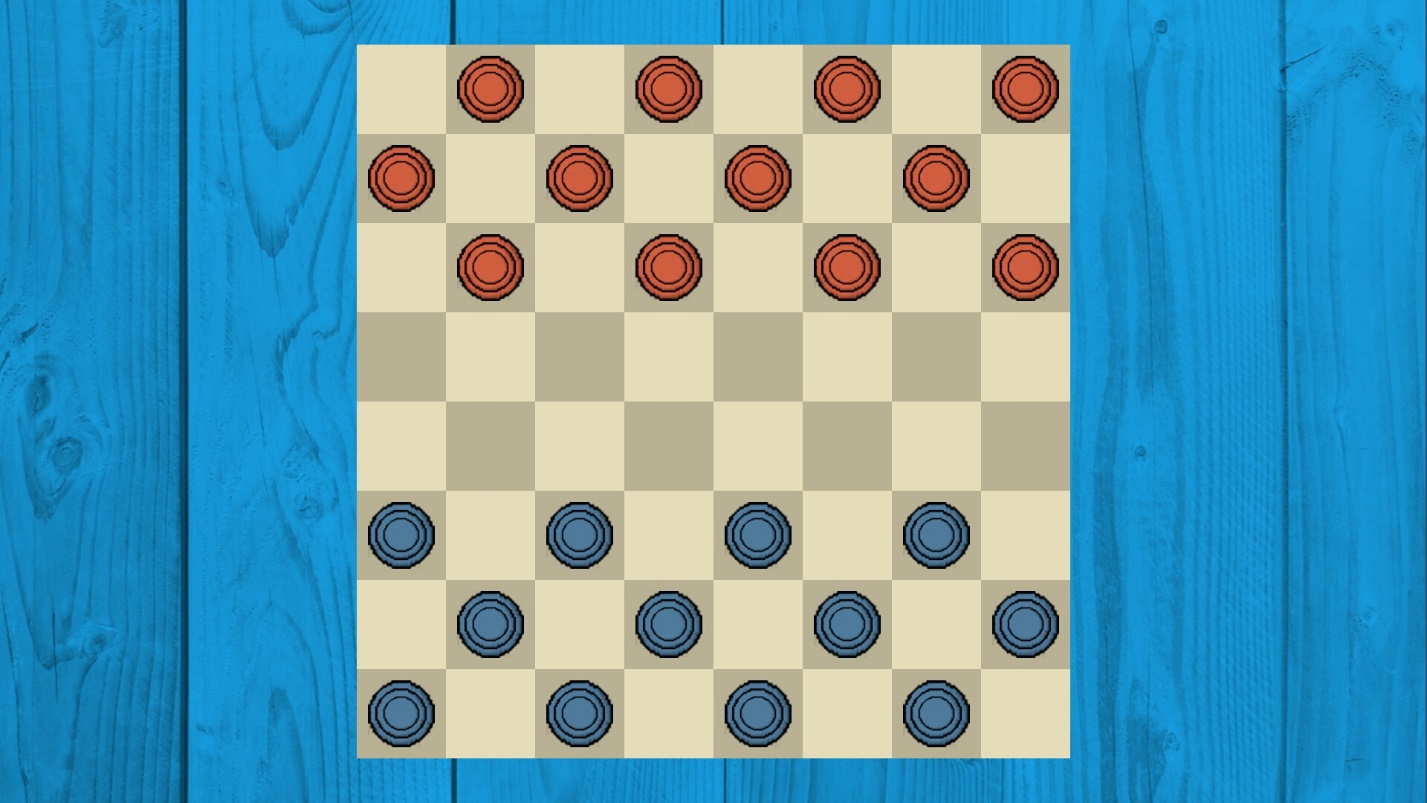
**Рисунок 3.4.8** – Сообщение о победе светлых



**Рисунок 3.4.9** – Сообщение о победе темных

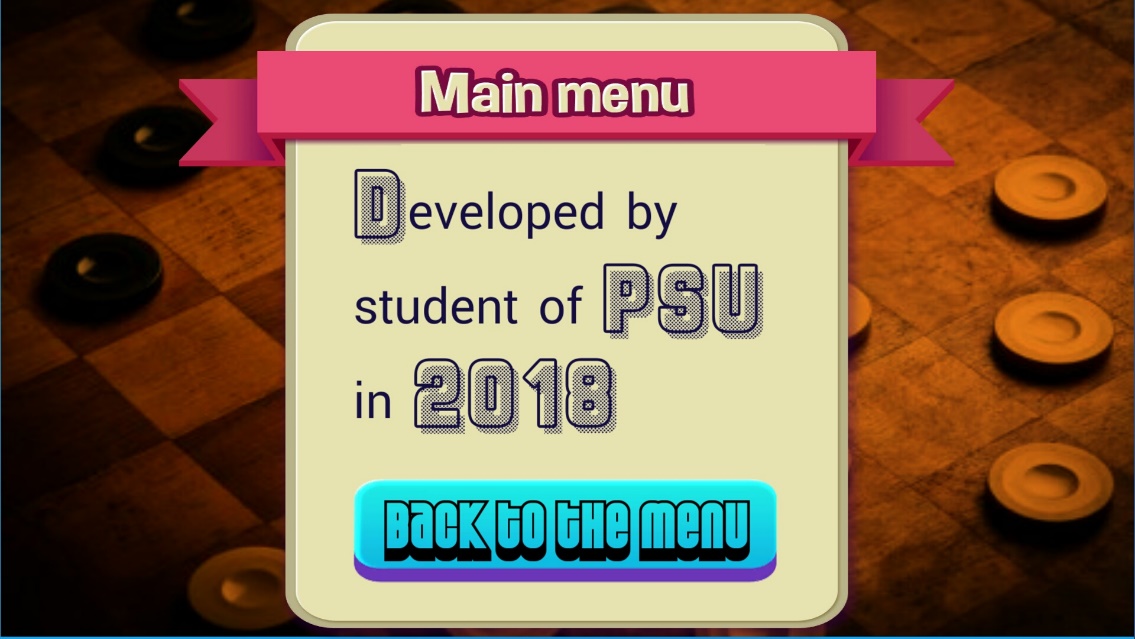
Также была проверена функция возможности нажатия на кнопку «back to the menu» (рисунок 3.4.3).

Проверка начала новой игры после проигрывания предыдущей. Результаты проверки приведены на рисунке 3.4.10.



**Рисунок 3.4.10** – Проверка возможности начать новую игру после завершения предыдущей

Далее была проведена проверка работоспособности кнопки «About». Результаты представлены на рисунке 3.4.11.



**Рисунок 3.4.11** – Переход на сцену

Была проверена возможность выхода из «About». Результаты теста приведены на рисунке 3.4.12 самого конца, до начала.





**Рисунок 3.4.12** – Возможность возвращаться в главное меню по кнопке back

В результате тестирования не было обнаружено никаких сбоев в работе, всё работает корректно – согласно требованиям, описанным в техническом задании (приложение Б).

**ЗАКЛЮЧЕНИЕ**

В результате выполнения данного курсового проекта было реализовано игровое приложение «Шашки». Оно написано на языке С# под операционную систему Android c применением движка Unity.

Мною были закреплены навыки работы на языке С#, приобретены основные навыки разработки приложений под операционную систему Android в среде разработки Unity. Я изучила основы работы с паттернами программирования, закрепила знания в области объектно-ориентированного программирования. Так же были изучены основные моменты работы с библиотеками Unity.

Во время данного курсового проекта я узнала новую для себя информацию об ОС Android, выступила в роли разработчика, тестировщика и дизайнера собственного Android-приложения.

В ходе проведения тестирования, ошибок обнаружено не было. Программа прошла все тесты с положительным результатом.

Разработанная игра получилась максимально адаптированной к экранам мобильных устройств с разным расширением, что значительно облегчит внедрение новых элементов без вреда основному коду.

Разработанное приложение было протестировано на трех разных устройствах с тремя разными дисплеями. Что свидетельствует об адаптивности игры к экранам разных мобильных устройств.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Преимущества ОС Андроид [Электронный ресурс] – Режим доступа: <https://www.compgramotnost.ru/android/plyusy-i-minusy-androida>. Дата обращения: 17.09.2018
2. Каталог API (Microsoft) и справочных материалов [Электронный ресурс] – Режим доступа: <https://msdn.microsoft.com/library> . Дата обращения: 25.11.2018
3. Википедия [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org>. Дата обращения: 01.11.2018.
4. Статистика версий Android на июль 2018 [Электронный ресурс] – Режим доступа: <https://tproger.ru/news/android-stat-july-2018/>. Дата обращения: 26.10.2018.
5. Unity [Электронный ресурс] – Режим доступа: <https://unity3d.com/ru> . Дата обращения: 15.09.2018
6. Доля Android на рынке мобильных устройств [Электронный ресурс] – Режим доступа: [https://www.ixbt.com/news/2018/05/03/doli-operacionnyh-sistem-android-i-ios-za-poslednij-god-v-mire-osobo-ne-izmenilis.html](https://www.ixbt.com/news/2018/05/03/doli-operacionnyh-sistem-android-i-ios-za-poslednij-god-v-mire-osobo-ne-izmenilis.html%20). Дата обращения 25.10.2018.

ПРИЛОЖЕНИЕ А

(обязательное)

Техническое задание

Введение

Наименование программного продукта – «Шашки». Данное приложение является развлекательным. Шашки по праву призваны одной из самых интеллектуальных игр. Она стимулирует мыслительную деятельность детей, способствует их логическому мышлению, развивает пространственное воображение, память и внимание, что очень важно для подготовки к школе. Большая часть детей в дошкольном возрасте по своей природе очень подвижны и активны, им трудно сосредоточить свое внимание – эта волшебная игра ненавязчиво формирует у них такое качество, как усидчивость. Также игра в шашки воспитывает в детях такие немаловажные для их будущей жизни качества, как умение самостоятельно думать и нести ответственность за принятое решение, адекватно относиться к неудачам и поражениям.

А.1 Основание для разработки

Приложение под систему Android разрабатывается в рамках курсового проекта студента учреждения образования «Полоцкий государственный университет» Виноградовой А.Д. Основанием для разработки является выданное задание к курсовому проекту по теме разработки мобильного приложения на системе Android, утверждённое заведующей кафедры технологий программирования Голубевой О.В. от 03.09.2018.

А.2 Назначение разработки

Функциональное и эксплуатационное назначение приложения – развитие навыков управления шашками, улучшение стратегии поведения игрока, зависящую от игровых ходов соперника. Шашки – известная и древняя логическая настольная игра, популярная во всем мире. Правила игры проще, чем у шахмат, однако от этого они не менее интересны.

А.3 Требования к программному продукту

А.3.1 Требования к функциональным характеристикам

При разработке Android-приложения выдвинуты следующие требования к функциональным характеристикам:

1. Игра должна содержать игровое меню состоящее из следующих пунктов:
2. Новая игра;
3. Об авторе;
4. Шашки должны быть двух видов: дамка и простая шашка;
5. Дамка (или простая шашка) может бить несколько фигур соперника и должна бить до тех пор, пока это возможно;
6. Пользователь может управлять шашками следующим образом: перетаскивая её в определенную ячейку;
7. При старте игры, должно открываться поле, на котором уже расставлены шашки «свои» и соперника;
8. Возможность корректного перемещения шашки по игровой области;
9. Возможность шашки взаимодействовать с другими шашками на игровом поле:
10. Исчезновение шашки (или шашек) в случае боя шашкой противника;
11. Превращение простой шашки в дамку, если она достигнет последнего противоположного горизонтального ряда доски (дамочного поля).
12. Возможность героя начать новую игру после завершения игровой сессии;
13. Программа должна иметь интуитивно понятный интерфейс;
14. Программа должна иметь возможность корректного выхода из игры.

А.3.2 Требования к надежности

Данная программа должна надежно функционировать. При возникновении программного сбоя программа должна оповещать пользователя о проблеме.

**А.3.3 Условия эксплуатации**

Эксплуатация программы «Шашки» должна осуществляться на мобильном устройстве с версией Android не ниже 4.4.

А.3.4 Требования к составу и параметрам технических средств

Для обеспечения корректной работы программного средства от устройства требуется:

* 1. Операционная система Android;
  2. Версия Android не ниже 4.4;
  3. Базовые навыки пользования телефоном;
  4. Мобильное устройство с наличием сенсорной панели;
  5. Знания правил игры в шашки.

А.3.5 Требования к информационной и программной   
совместимости

Для успешной работы приложения необходим смартфон под управлением операционной системы Android 4.4 и выше.

А.3.6 Требования к маркировке и упаковке

Требования к маркировке и упаковке отсутствуют.

А.3.7 Требования к транспортированию и хранению

Программное средство должно храниться на электронном носителе в виде исполняемого файла.

А.4 Требования к программной документации

Программная документация по приложению «Шашки» должна быть предоставлена в следующем составе:

1. Техническое задание. Согласно ГОСТ 19.201-78;
2. Пояснительная записка.

Требования к перечисленным программным документам устанавливаются государственными стандартами ЕСПД.

А.5 Стадии и этапы разработки

Разработка программы заключается в следующем:

1. Анализ исходных данных и постановка задачи проектирования, разработка технического задания.
2. Разработка интерфейса, архитектуры и структуры программы.
3. Реализация и тестирование программы.
4. Разработка программной документации.

**А.6 Порядок контроля и приемки**

Для проверки корректности приложения применялись следующие программные средства:

1. ОС Android 4.4;
2. Среда разработки Visual Studio 2017, Unity.

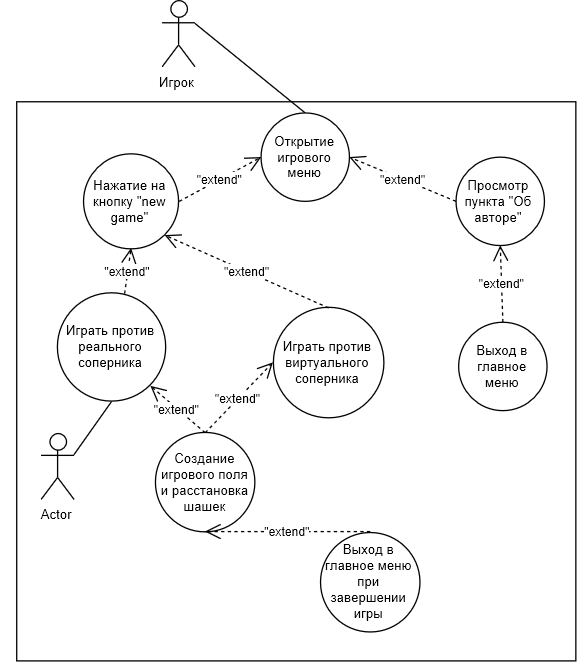
Тестирование программы состояло из проверки корректности работы ранее перечисленных функций.

Методы испытаний:

Основным методом испытания программы является визуальный контроль выполнения программой требующихся функций, корректное выполнение юнит-тестов.

ПРИЛОЖЕНИЕ Б

**Диаграмма вариантов использования**



**Рисунок Б.1** – Диаграмма вариантов использования