Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

Дисциплина «Объектно-ориентированное программирование и проектирование»

|  |  |
| --- | --- |
|  | «К ЗАЩИТЕ ДОПУСТИТЬ» |
|  | Руководитель курсового проекта  ассистент кафедры ПИКС  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Листратенко |
|  | \_\_\_.\_\_\_\_.2022 |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовому проекту

на тему:

**«Игровое программное средство “Карточная игра” для мобильных устройств»**

БГУИР КП 1-40 05 01-10

|  |  |
| --- | --- |
|  | Выполнил студент группы 114301  Тюленев Юрий-Франтишек Александрович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |
|  | Курсовой проект представлен на проверку \_\_\_.\_\_\_\_.2022  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |

Минск 2022

**РЕФЕРАТ**

БГУИР КР 1-40 05 01-10 017 ПЗ

ИГРОВОЕ ПРОГРАММНОЕ СРЕДСТВО “Карточная игра” ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ/ Тюленев Ю-Ф.А.– Минск: БГУИР, 2022. – 38 с.,

чертежей(плакатов) – 4 л. формата A3

ИГРОВОЕ ПРОГРАМНОЕ СРЕДСТВО, Unity, C#, ОБЪЕКТНООРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ.

Предмет: игровое программное средство.

Объект: разработка игрового программного средства в среде Unity на языке C#.

Цель: разработка игрового программного, в котором пользователи смогут играть в карточкую игру против друг друга удаленно.

Методология проведения работы: в процессе решения поставленных задач использованы принципы разработки приложения через отдельные модули и принципы взаимодействия игроков, используя удаленный сервер.

Результаты работы: в ходе выполнения курсовой работы осуществлена разработка игрового программного средства с возможностью играть “по сети”.

Область применения результатов: личное использование для игры с друзьями.

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

|  |  |  |
| --- | --- | --- |
|  |  | УТВЕРЖДАЮ  Заведующий кафедрой ПИКС  \_\_\_\_\_\_\_\_\_\_\_\_\_\_В.В.Хорошко  «\_\_\_\_» сентября 2022 |
|  |  |  |

**ЗАДАНИЕ**

**к курсовому проекту по дисциплине**

«Объектно-ориентированное проектирование и программирование»

Группа *114301*

Студенту *Тюленеву Юри-Франтишеку Александровичу*

**1.Тема проекта:** *«Игровое программное средство “Карточная игра”».*

**2.Сроки сдачи студентом законченного проекта*:*** 12-14.12.2022 г.

**3.Исходные данные к проекту:**

3.1. Описание системы – *мобильное игровое программное средство с возможностью игры по сети.*

3.2. Назначение системы – *возможность играть в карточкую игру против друг друга.*

3.3. Язык и среда программирования – C#, Unity, организация базы данных – Player Prefs. Разработанное программное обеспечение должно быть реализовано на объектно-ориентированном языке.

3.4.Нормативные источники: 3.4.1.Положение о курсовом проектировании БГУИР. 3.4.2. СТП 01-2017. Стандарт предприятия. Дипломные проекты (работы). Общие требования. 3.4.3. ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения.

**4.Содержание расчетно-пояснительной записки** (перечень подлежащих разработке вопросов):

Титульный лист. Реферат. Задание. Содержание. Перечень условных обозначений, символов и терминов.

Введение

4.1. Анализ исходных данных на курсовое проектирование. 4.1.1. Анализ исходных данных к курсовому проекту. 4.1.2. Обоснование и описание выбора языка программирования, средств разработки, используемых технологий и сторонних библиотек.

4.2. Проектирование и разработка программного средства. 4.2.1. Проектирование архитектуры и описание состояний программного средства. 4.2.2. Проектирование и разработка графического интерфейса. 4.2.3. Описание и реализация используемых в программном средстве алгоритмов.

4.3. Эксплуатация программного средства. 4.3.1. Ввод в эксплуатацию и обоснование минимальных технических требований к оборудованию. 4.3.2. Руководство по эксплуатации программного средства.

Заключение .

Список использованных источников

Приложения (листинг программного кода; справка о проверке курсового проекта на плагиат; ведомость курсового проекта).

**5.Перечень графического материала** (с указанием обязательных чертежей и графиков):

5.1.Схема алгоритма (формат А3).

5.2.UML диаграмма классов (плакат, формат А3).

5.3.Диаграмма состояний (плакат, формат А3).

5.4.Структура графического пользовательского интерфейса (формат А3).

**6.Консультанты по проекту:** старший преподаватель ГОРБАЧ Антон Петрович (ауд. 415а-1 корп.), ассистент КУПРИЯНОВ Никита Игоревич (ауд. 412-1 корп.), ассистент ЛИСТРАТЕНКО Владислав Викторович (ауд. 412-1 корп.), ассистент ЛАРЬКИН Антон Дмитриевич (ауд. 435а-1 корп.), ассистент ГОРОХ Павел Игоревич (ауд. 412-1 корп.).

**7.Дата выдачи задания**: 03.09.2022 г.

**8.Календарный график работы над проектом на весь период проектирования** (с указанием сроков выполнения и трудоемкости отдельных этапов):

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование этапов курсового проекта | Срок выполнения этапов проекта | Примечание |
| 1. | 1-я опроцентовка (4.1, 5.1) | 03-06.10.2022 | 30% |
| 2. | 2-я опроцентовка (4.2, 5.2-5.3) | 31.10-03.11.2022 | 60% |
| 3. | 3-я опроцентовка (введение, 4.3, 5.4, заключение) | 28.11-01.12.2022 | 80% |
| 4. | Сдача курсового проекта на проверку | 12-14.12.2022 | 100% |
| 5. | Защита курсового проекта | 19-23.12.2022 | Согласно графику |

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Листратенко

Задание принял к исполнению 03.09.2022 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(*подпись студента*)  *(расшифровка подписи)*

**СОДЕРЖАНИЕ**

[Введение 5](#_Toc121884563)

[1 Анализ исходных данных на курсовое проектирование 6](#_Toc121884564)

[1.1 Анализ исходных данных к курсовому проекту 6](#_Toc121884565)

[1.2 Обоснование и описание выбора языка программирования, средств разработки, используемых технологий и сторонних библиотек 6](#_Toc121884566)

[2 Проектирование и разработка программного средства 8](#_Toc121884567)

[2.1 Проектирование архитектуры и описание состояний программного средства 8](#_Toc121884568)

[2.2 Проектирование и разработка графического интерфейса 11](#_Toc121884569)

[3 Эксплуатация программного средства 22](#_Toc121884570)

[3.1 Ввод в эксплуатацию и обоснование минимальных технических требований к оборудованию 22](#_Toc121884571)

[3.2 Руководство по эксплуатации программного средства 22](#_Toc121884572)

# **ВВЕДЕНИЕ**

Сегодня мобильные устройства занимают не последнее место в нашей жизни. В частности мобильные игры. Благодаря мобильности и мощности современных телефонов, они дают возможность играть в игры, по возможностям не сильно уступающим играм на персональных компьютерах, причем играть в совершенно любых местах.

Люди играют пока едут в метро, чтобы скоротать время, в перерывах на работе, чтобы расслабится, на парах в университете, когда им скучно.

Много мобильных игр позволяет играть людям вместе, совместно или против друг друга, находясь на тысячи километров друг от друга.

В данной работе рассматривается разработка игрового программного средства “Карточная игра”.

Цель курсового проекта – дать возможность интересно и просто играть людям вместе, через интернет.

Для достижения цели курсового проекта необходимо решить следующие задачи:

– изучить предметную область;

– построить алгоритм программы;

– разработать интерфейс программы;

– построить архитектуру программы;

– наладить взаимодействие клиента и сервера

–протестировать программу, исправить обнаруженные ошибки.

1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ

* 1. Анализ исходных данных к курсовому проекту

Тема курсовой работы: разработка игрового программного средства “Карточная игра” для пользователей под ОС Android.

Cреда разработки Unity

Язык программирования С#.

Парадигма программирования – объектно-ориентированная.

Способ организации данных – классы.

Способ хранения данных – Player Prefs

К защите курсовой работы представляются:

1. мобильное программное средство;
2. схема алгоритма (формат А2/А3);
3. UML диаграмма классов (плакат, формат А2/А3);
4. диаграмма состояний (плакат, формат А2/А3);
5. структура графического пользовательского интерфейса (плакат, формат А2/А3) и пояснительная записка.

Текст пояснительной записки оформляется в соответствии со стандартом предприятия СТП 01–2017.

1.2 Обоснование и описание выбора языка программирования, средств разработки, используемых технологий и сторонних библиотек

В качестве средства разработки я выбрал Unity. Это межплатформенная среда разработки компьютерных игр, которая позволяет создавать приложения, работающие под более чем 20 различными операционными системами, включающими персональные компьютеры, игровые консоли, мобильные устройства и интернет-приложения.

Преимущества Unity:

* Кроссплатформенность
* Удобный и функциональный интерфейс
* Компонентно-ориентированный подход разработки
* Много встроенных библиотек, плагинов и поддерживаемых API
* Наличие отладчика
* Большое сообщество разработчиков

В Unity уже реализованы многие игровые модули: Физический,

звуковой, графический, менеджер ресурсов, пользовательский ввод и многие другие. Это позволяет быстро и качественно разрабатывать приложение, сосредотачиваться только на постройке игровой архитектуры и логике.

Язык программирования я выбрал исходя из средства разработки. В новых версиях Unity поддерживает только C#.

С# («Си Шарп») – один из наиболее быстро растущих, востребованных и при этом «удобных» языков программирования. Это модификация фундаментального языка С от компании Microsoft, призванная создать наиболее универсальное средство для разработки программного обеспечения для большого количества устройств и операционных систем.

Язык входит в семью С-подобных языков. Синтаксис приближен к Java и C++. Его особенности:

* статистическая типизация,
* поддерживается полиморфизм,
* поддерживается перегрузка операторов,
* доступна делегация, атрибуты, события, обобщенные типы и анонимные функции.

В рамках Unity язык C# отвечает за игровую логику. В Unity представлены базовые классы для взаимодействие с игровым окружением. В частности MonoBehavior, который имеет базовые события. Основные из них:

* Start() – Вызывается при старте сцены
* Update() – Вызывается при каждом обновлении кадра
* FixedUpdate() – Вызывается каждые N секунд, не зависит от частоты кадров

В качестве клиент-серверной части я использую библиотеку Photon Unity Networking. Она позволяет быстро и просто создавать клиент-серверное взаимодействие между игроками. Преимущества Photon Unity Networking:

* Простой и многофункциональный API
* Встроенный подбор оппонентов
* Понятная документация
* Условно бесплатный

# **2** **ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА**

**2.1 Проектирование архитектуры и описание состояний программного средства**

Проектирование игрового программного средства состоит из разработки отдельных игровых модулей, разделенных логически между собой. Каждый игровой модуль представляет из себя отдельную сборку – Assembly Defenition.

Каждая сборка компилируется отдельно в .dll файл, что сильно ускоряет время компиляции проекта, так как не нужно компилировать весь код проекта, а только отдельный модуль. Между сборками не может быть циклических. Проект состоит из 9 модулей:

* модуль TouchInput;
* модуль UI;
* модуль Audio;
* модуль PhotonUnityNetworking;
* модуль Extention;
* модуль Data;
* модуль Lobby;
* модуль Cards;
* модуль Managament.

Модуль TouchInput содержит в себе класс TouchInput, который отвечает за считываение жестов пользователя и преобразование их в события. Класс CardMovement считывает события классы TouchInput и отвечает за перемещение карт пользователем.

Модуль UI отвечает за элементы пользовательского интерфейса. Содержит в себе классы, отвечающие за анимации элементов, считывание нажатия, отображение списков и прочие. Так же содержит класс PlayerUI, который отвечает за интерфейс игрока.

Модуль Audio содержит класс SoundManager, в котором хранится массив звуков, доступных для проигрывания из любого места программы.

Модуль PhotonUnityNetworking представляет собой импортированную библиотеку для клиент-серверного взаимодействия.

Модуль Extention содержит в себе класс Extention, который содержит методы-расширения для встроенных классов. Например метод Shuffle для IList, который случайно перемешивает лист.

Модуль Data содержиь в себе класс DataBase который отвечает за сохранение пользовательской информцации: имя пользователя, имя комнаты, количество матчей и информацию о каждом сыгранном матче. Для сохранения информации используется встроенный в Unity класс PlayerPrefs, который хранит информацию в виде ключ-значение.

Модуль Lobby представляет собой класс Lobby, который отвечает за работу главное меню, создание и присоединение к комнате, просмотр игровой статистики, изменении псевдонима игрока.

Модуль Cards представляет классы игровых объектов. Класс Card представляет собой сущность карты. Класс Hand представляет собой сущность руки игрока, держащая карты. Класс Deck представляет собой сущность колоды карт. Класс DiscardPile представляет собой сущность отбоя. Класс Table представляет собой сущность игрового стола, который отвечает за прием карты, проверки может ли одна карта побить другую, закончен ли ход, кто победил в данном ходу, кто победил в матче.

Модуль Managament представляет классы, отвечающие за правление игровым процессом. Класс GameContext содержит в себе ссылки на все классы, отвечающие за игровой процесс и связывает их в единое целое. Он подписывается на все события игровых классов и отвечает за их жизненный цикл. Класс HostController отвечает за клиент-серверное взаимодействие, он подписывается на события всех игровых объектов и дублирует нужные событие на другом клиенте.

Диаграмма классов представлена на рисунке 2.2.1

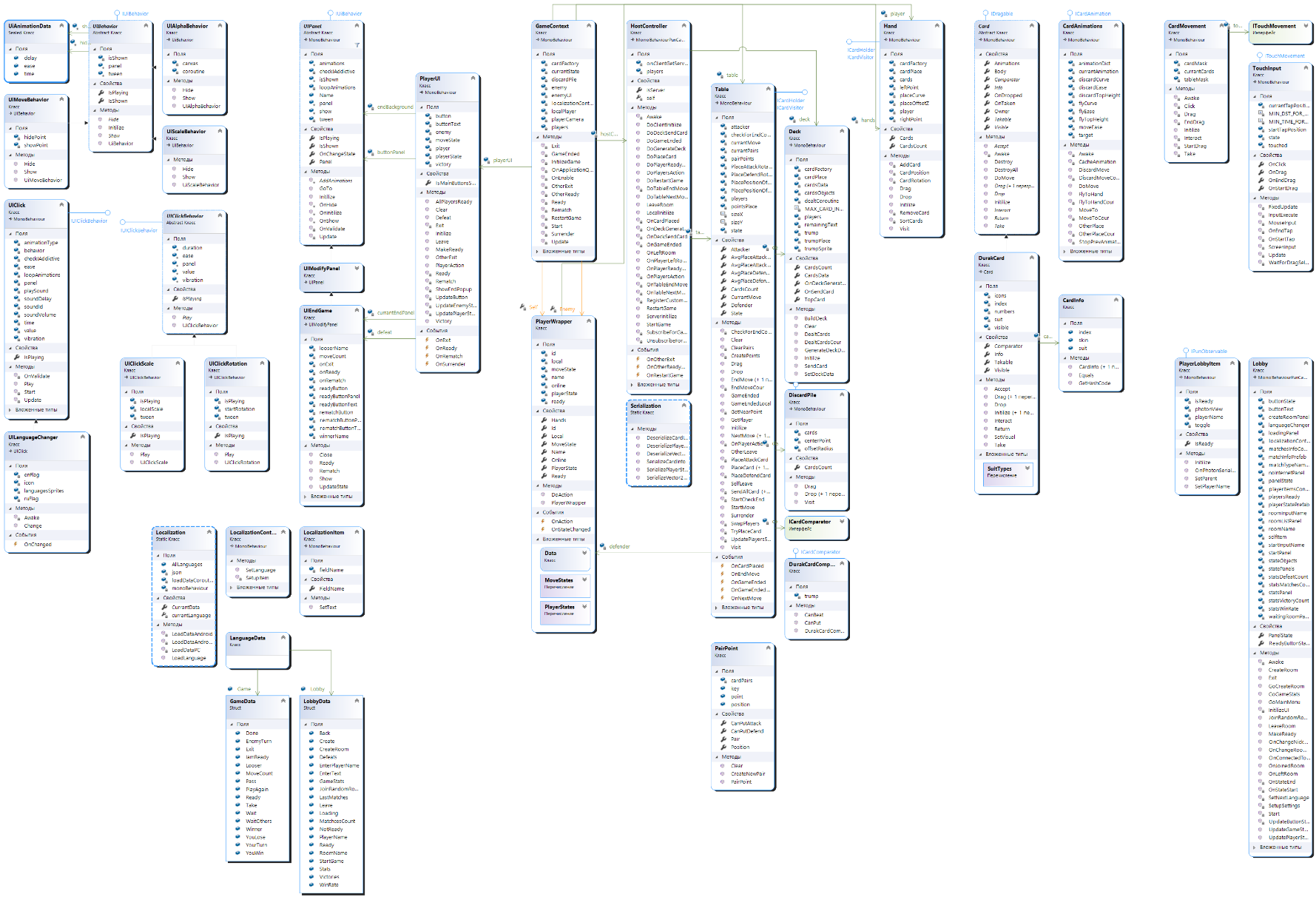


Рисунок 2.2.1 – Диаграмма классов

Диаграмма состояний — поведенческая диаграмма, отображающая состояния, в которых может находится программное средство, система или объекты этой системы. На диаграмме состояний программного средства отображены следующие ключевые состояния:

* Отображение экрана загрузки;
* Отображение экрана главного меню;
* Отображение экрана создания комнаты;
* Отображение экрана присоединения к комнате;
* Отображение экрана игровой статистики;
* Отображение игрового процесса;
* Отображение панели конца игры.

Диаграмма состояний изображена на рисунке 2.2.2.

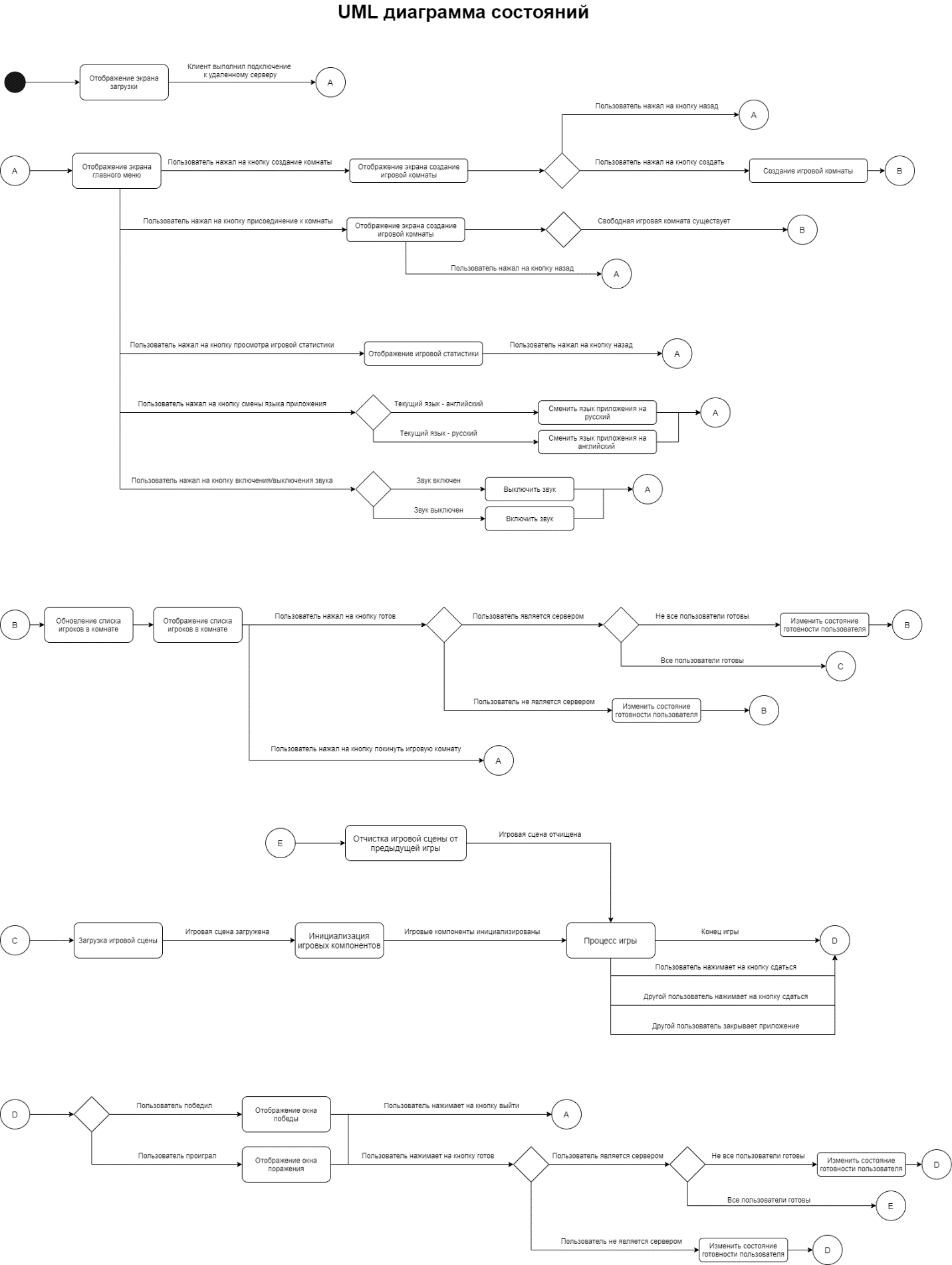


Рисунок 2.2.2 – UML диаграмма состояний

2.2 Проектирование и разработка графического интерфейса

GUI (Graphical User Interface, графический интерфейс пользователя) –

это форма пользовательского интерфейса , которая позволяет пользователям взаимодействовать с приложением используя графические элементы.

Реализация загрузочного экрана на рисунке 2.3.1.



Рисунок 2.3.1 – загрузочный экран

Реализация главного меню на рисунке 2.3.2.

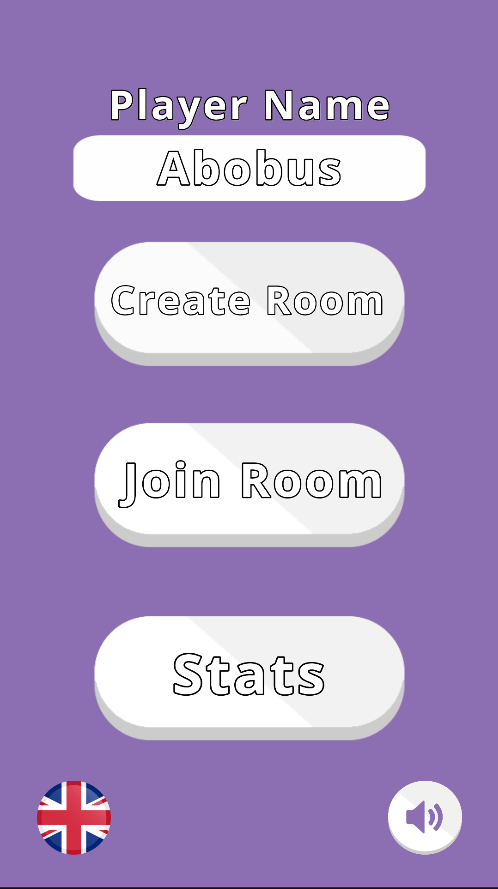


Рисунок 2.3.2 – главное меню

Реализация меню создания комнаты на рисунке 2.3.3.



Рисунок 2.3.3 – меню создания комнаты

Реализация меню присоединения к комнате на рисунке 2.3.4.



Рисунок 2.3.4 – меню присоединения к комнате

Реализация меню ожидания игроков на рисунке 2.3.5.

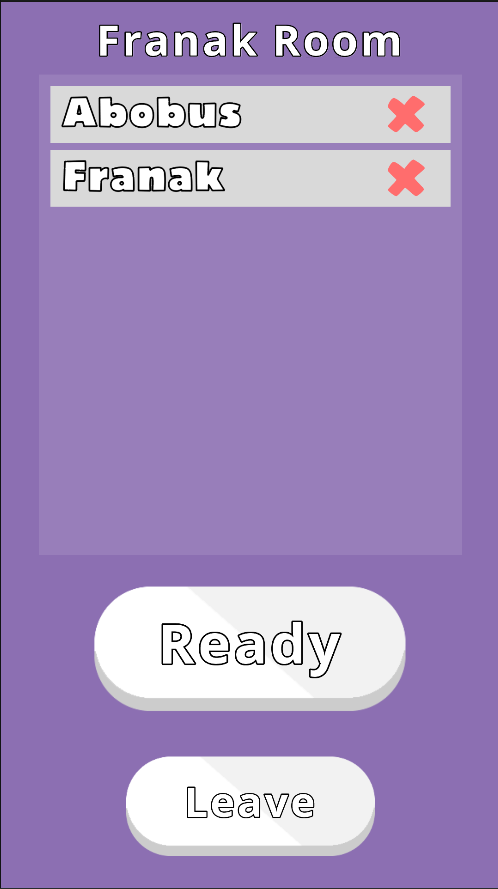


Рисунок 2.3.5 – меню ожидания игроков

Реализация меню игровой статичтики на рисунке 2.3.6.



Рисунок 2.3.6 – меню игровой статистики

Реализация игрового интерфейса на рисунке 2.3.7.

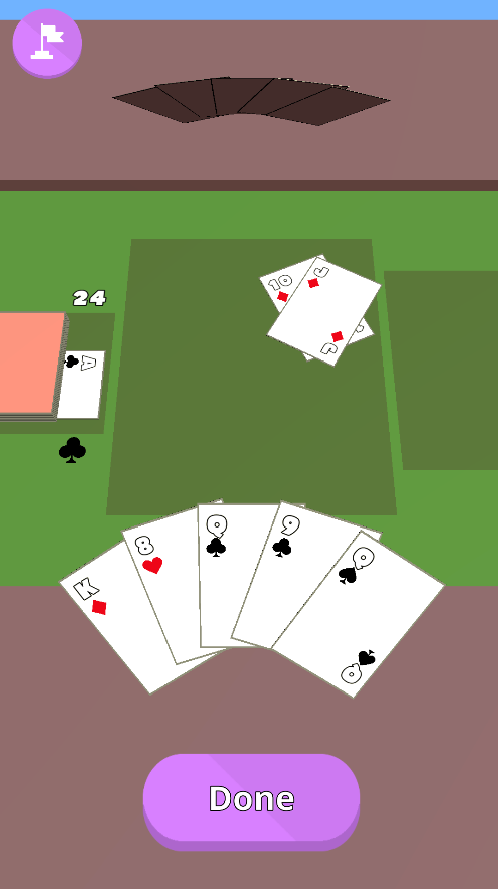


Рисунок 2.3.7 – игровой интерфейс

Реализация панели конца матча на рисунке 2.3.8.



Рисунок 2.3.8 – панель конца матча

**2.3 Описание и реализация используемых в программном средстве**

**алгоритмов.**

Основные аглоритмы, которые используются в данном игровом программном средстве это алгоритны, связаные с игровым процессом.

Схема алгоритма создания игрового поля показана на рисунке 2.4.1.

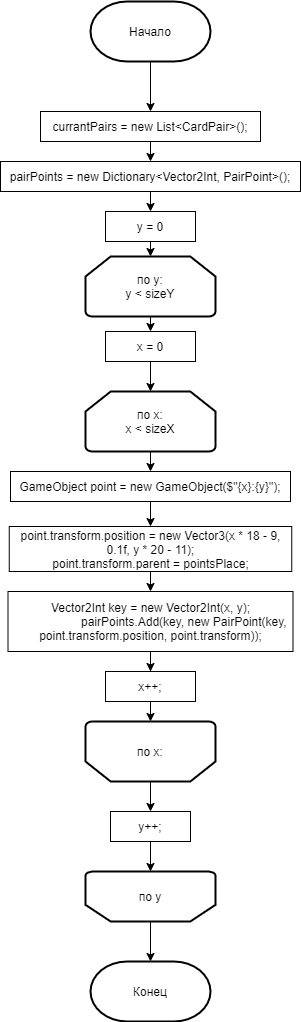


Рисунок 2.4.1 – блок-схема алгоритма создания игрового поля

Схема алгоритма проверки конца матча показана на рисунке 2.4.2. Данный метод асинхронный, реализован с помощью Coroutine. Метод вызывается только на клиенте-сервере.

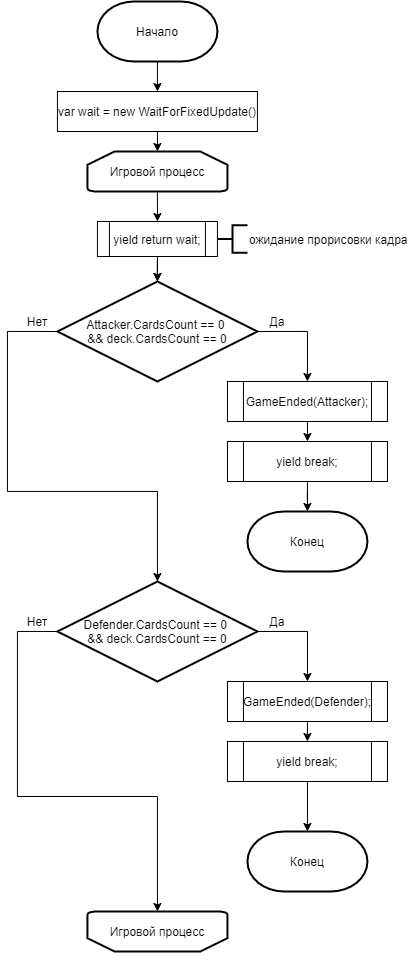


Рисунок 2.4.2 – блок-схема алгоритма проверки конца матча

Схема алгоритма раздачи карт показана на рисунке 2.4.3. Данный метод асинхронный, реализован с помощью Coroutine. Метод вызывается только на клиенте-сервере.

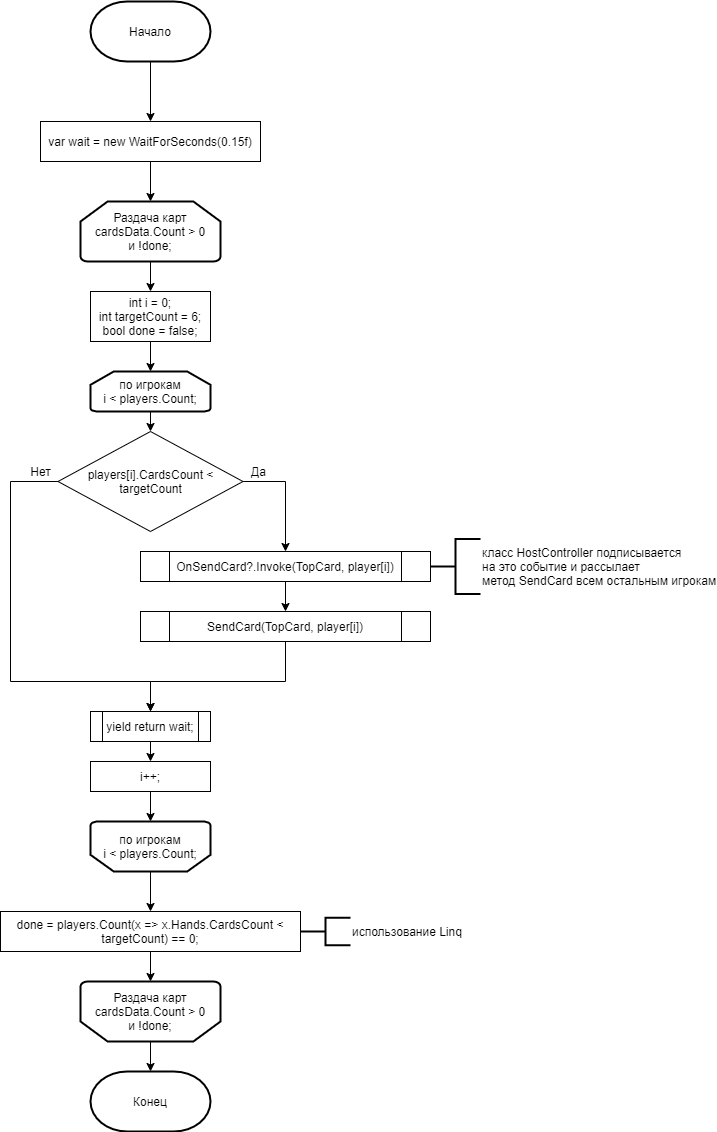


Рисунок 2.4.3 – блок-схема алгоритма проверки конца матча

3 ЭКСПЛУАТАЦИЯ ПРОГРАММНОГО СРЕДСТВА

* 1. **Ввод в эксплуатацию и обоснование минимальных технических требований к оборудованию**

Для распостранение игры среди пользователей, использующих мобильные устройства на операционной системе Android нужно собрать проект в .apk файл. При установке Unity, так же устанавливаются необходимые SDK, JDK и Gradle для сборки приложения под операционную систему Android.

При сборке проекта Unity сгенерирует дистрибутив приложения (.apk-файл) со всеми необходимыми библиотеками и сериализованными ассетами.

В операционной системе Android минимальные технические требования представлены двумя атрибутами: android:minSdkVersion и android:targetSdkVersion.

Android:minSdkVersion – это минимальное значение версии платформы Android, которое поддерживает приложение.

Android:targetSdkVersion – это значение версии платформы Android, для которого разработано приложение.

Минимальные требования к оборудовнию изображены в таблице 1.

Таблица 1 – минимальные требования к мобильному устройству

|  |  |
| --- | --- |
| **Требование** | **Значение** |
| Android:minSdkVersion | API LEVEL 19 |
| Android:targetSdkVersion | API LEVEL 29 |
| Свободная память | 51мб |
| Подключение к интернету | Да |

3.2 Руководство по эксплуатации программного средства

После установки .apk файла, нужно запустить приложение “Card Game” после чего запуститься загзузочный экран представленный на рисунке 2.3.1.

Для работы приложение нужно подключение к интернету для взаимодействия с серверами Photon Networking.

После загрузки приложения игрок попадает в главное меню, представленное на рисунке 2.3.2, в котором может изменить свой игровой псевдоним, при первмо запуске будет сгенерировано случайное имя. Игрок может изменить язык приложения нажав на круг слева снизу, так же может включить/выключить звук приложения.

Для проведения матчка необходимо два устройства. Первый игрок нажимает на кнопку создания комнаты, затем указывает ее имя –индентификатор, по которому второй игрок сможет к ней присоединится (игрок создавший комнату является клинтом-сервером), после чего нажимает на кнопку создать комнату, изображенную на рисунке 2.3.3

Второй игрок нажимает на кнопку присоединения к комнате, вписывает нужное имя комнаты(имя комнаты сохраняется и при последующих присоединениях к комнате одно и тоже имя комнаты вписывать не придется) и нажимает на кнопку присоединиться, изображенной на рисунке 2.3.4.

После того как игрок создал либо присоединился к комнате откроется экран ожидания игроков, изображенный на рисунке 2.3.5. Игроки нажимают на кнопку готовности и после того, как все игроки в комнате готовы, игрок, создавший комнату может запустить матч.

После запуска матча запуститься процесс игры, изображенный на рисунке 2.3.7. Игроки ходят по очереди в соответствии правил в игре “Дурак”. Атакующий игрок подкидывает карты, а защищающийся отбивается. После конца хода карты сбрасываются в отбой, если все карты отбиты, в противном случае идут в руку отбивающеегося игрока.

Если игрок нажмет кнопку “Сдаться”, или закрывает приложение, то он проигрывает, а его оппонент побеждает, матч заканчивается.

Матч заканчивается как только у одного игрока в руке нету карт и в колоде тоже.

После завершения матча, у игроков появляется панель конца матча, изображенная на рисунке 2.3.8, а результат матча записывается в базу данных каждого игрока. У игрока есть два варинта: выйти из игры и играть сново, как только все игроки готовы и игрок, создавший комнату нажимает на кнопку “Играть сново”, игра перезапускается.

**ЗАКЛЮЧЕНИЕ**

В процессе реализации всех условий курсового проекта были достигнуты все поставленные цели и в результате создано игрового программное, позволяющее играть в карточкую игру “Дурак” против друг друга.

Для реализации клиент серверного взаимодействия, необходимым стало осваивание библиотеки Photon Networking.

Были выполнены все поставленные задачи: изучение литературных источников и аналоговых программных средств по данной теме, проектирование игровых модулей, клиент-серверного взаимодействия и пользовательского интерфейса, тестирование программы и оценка результатов.

В результате выполнения курсового проекта был получен работающие, готовое к использованию приложение, удовлетворяющий всем требованиям.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Документация Unity [Электронный ресурс]. – Режим доступа:

https://docs.unity3d.com/2020.3/Documentation/Manual/index.html

[2] Документация Photon Unity Networking [Электронный ресурс]. – Режим доступа: https://doc.photonengine.com/en-us/pun/current/getting-started/pun-intro

[3] Онлайн игра: полный туториал, часть 1 [Unity 3D] [Photon] - матчмейкинг и простая синхронизация [Электронный ресурс]. – Режим доступа:

https://www.youtube.com/watch?v=aqxzrk-pl-8&ab\_channel=EmeraldPowder

[4] Онлайн игра: полный туториал, часть 2 [Unity 3D][Photon] - произвольные параметры и игровая карта [Электронный ресурс]. – Режим доступа:

https://www.youtube.com/watch?v=1q2x0IAvLBQ&ab\_channel=EmeraldPowder

[5] Онлайн игра: полный туториал, часть 3 [Unity 3D] [Photon] - события и перемещение персонажей [Электронный ресурс]. – Режим доступа:

https://www.youtube.com/watch?v=\_FkeQyW8mtk&t=12s&ab\_channel=EmeraldPowder