



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

**Институт информационных технологий (ИТ)**  
**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**КУРСОВАЯ РАБОТА**

по дисциплине: Разработка программных приложений

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: 09.03.04 «Программная инженерия»

Тема: «Разработка программного приложения для хранения информации о видеоиграх»

Студент: Петров Анатолий Валерьевич

Группа: ИКБО-03-18

Работа представлена к защите 15.05.2020

\_\_\_\_\_/Петров А.В./  
(подпись и ф.и.о. студента)

Руководитель: доцент Буравцев Алексей Владимирович

Работа допущена к защите \_\_\_\_\_ (дата)

\_\_\_\_\_/Буравцев А.В./  
(подпись и ф.и.о. рук-ля)

Оценка по итогам защиты:

\_\_\_\_\_/\_\_\_\_\_, доцент Буравцев Алексей Владимирович /  
\_\_\_\_\_/\_\_\_\_\_/

(подписи, дата, ф.и.о., должность, звание, уч. степень двух преподавателей, принявших защиту)

2020г.

И. о. заведующего кафедрой

инструментального и прикладного  
программного обеспечения (ИиППО)  
Института информационных технологий (ИТ)

Болбакову Роману Геннадьевичу

От студента Петрова Анатолия Валерьевича  
ФИО  
ИКБО-03-18  
группа  
2 курс  
курс

Контакт: 8(968)562-08-10,  
petrovanatoly2000@gmail.c  
om

### Заявление

Прошу утвердить мне тему курсовой работы по дисциплине «Разработка программных приложений» образовательной программы бакалавриата 09.03.04 (Программная инженерия). Тема: «Разработка программного приложения для хранения информации о видеоиграх».

Приложение: лист задания на КР в 2-ух экземплярах на двухстороннем листе (проект).

Подпись студента /Петров А.В./  
подпись ФИО

Дата \_\_\_\_\_

Подпись руководителя / к.т.н., доц. Буравцев А.В./  
подпись Должность, ФИО

Дата \_\_\_\_\_



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА - Российский технологический университет»  
РТУ МИРЭА

---

Институт информационных технологий (ИТ)  
Кафедра инструментального и прикладного программного обеспечения (ИиППО)

### ЗАДАНИЕ на выполнение курсовой работы

по дисциплине: Разработка программных приложений

по профилю: Разработка программных продуктов и проектирование информационных систем

Студент: Петров Анатолий Валерьевич

Группа: ИКБО-03-18

Срок представления к защите: 15.05.2020

Руководитель: доцент Буравцев Алексей Владимирович

**Тема** «Разработка программного приложения для хранения информации о видеоиграх»

**Исходные данные:** документация по языку Java, ГОСТ 19.105-78, ГОСТ 19.201-78, статьи по работе с Android Studio.

**Перечень вопросов, подлежащих разработке, и обязательного графического материала:**

1. Провести анализ предметной области разработки программных приложений, 2. Определить перечень функциональных требований к программе, 3. Выбрать инструментальные средства для разработки программы, 4. Спроектировать модули программного приложения. 5. Реализовать модули программного приложения 6. Представить пояснительную записку.

Руководителем произведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

И. о. зав. кафедрой ИиППО: \_\_\_\_\_ / Болбаков Р. Г. /, «\_\_\_\_\_» \_\_\_\_\_ 2020 г.

Задание на КР выдал: \_\_\_\_\_ / Буравцев А. В. /, «\_\_\_\_\_» \_\_\_\_\_ 2020 г.

Задание на КР получил: \_\_\_\_\_ / Петров А. В. /, «\_\_\_\_\_» \_\_\_\_\_ 2020 г.

# АННОТАЦИЯ

Представленная курсовая работа на тему «Разработка программного приложения для хранения информации о видеоиграх» состоит из введения, пяти разделов, заключения, списка использованных источников и приложений.

Во введении рассматривается актуальность выбранной темы, задачи, выполнение которых является необходимым условием для достижения цели курсовой работы, а также объект и предмет исследования.

В основной части работы рассматриваются общие сведения, такие как обозначение и наименование программы, программное обеспечение, необходимое для функционирования программы, языки программирования, на которых написана программа. Далее описано функциональное назначение, логическая структура, включая декомпозицию, также следует описание зависимостей, деталей, алгоритмов и графического интерфейса. Затем описываются используемые технические средства, вызов и загрузка.

# СОДЕРЖАНИЕ

АННОТАЦИЯ .....	4
СОДЕРЖАНИЕ .....	5
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ.....	7
ВВЕДЕНИЕ .....	8
1 ОБЩИЕ СВЕДЕНИЯ .....	10
1.1 ОБОЗНАЧЕНИЕ И НАИМЕНОВАНИЕ ПРОГРАММЫ.....	10
1.2 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, НЕОБХОДИМОЕ ДЛЯ ФУНКЦИОНИРОВАНИЯ ПРОГРАММЫ .....	10
1.3 ЯЗЫКИ ПРОГРАММИРОВАНИЯ, НА КОТОРЫХ НАПИСАНА ПРОГРАММА .....	10
2 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ .....	11
3 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ .....	12
3.1 ДЕКОМПОЗИЦИЯ.....	12
3.1.1 Пакет UI .....	12
3.2 ОПИСАНИЕ ДЕТАЛЕЙ.....	12
3.2.1 Класс MainActivity.....	12
3.2.2 Класс Choose .....	13
3.2.3 Класс EndActivity.....	14
3.2.4 Прочие классы .....	15
3.3 ОПИСАНИЕ АЛГОРИТМОВ.....	16
3.3.1 Алгоритм перехода игры.....	16
3.3.2 Алгоритм выбора типа игр.....	17
3.3.3 Другие алгоритмы .....	17
3.4 ОПИСАНИЕ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА.....	18
4 ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА.....	28
5 ВЫЗОВ И ЗАГРУЗКА .....	29

ЗАКЛЮЧЕНИЕ .....	30
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	31
ПРИЛОЖЕНИЯ .....	32
Приложение А .....	33
Приложение Б .....	37

## **ПЕРЕЧЕНЬ СОКРАЩЕНИЙ**

ОС — операционная система;

JVM — Java Virtual Machine (виртуальная машина Java);

UML — Unified Modeling Language (унифицированный язык моделирования);

XML — eXtensible Markup Language (расширяемый язык разметки).

## ВВЕДЕНИЕ

В данной курсовой работе было решено разработать мобильное приложение, нацеленное на любого пользователя, который любит видеоигры. Многим людям интересно следить за тем, когда выходит новая игра или же узнать, во что можно поиграть прямо сейчас. Чтобы узнать эту информацию пользователю обычно приходится искать о играх в Интернете и это не очень удобно. С помощью данного приложения пользователь сможет просматривать нужную информацию о тех играх, которые занесены программой. В программе можно выбрать платформу, ведь некоторые игры являются “эксклюзивами” (игры, в которые нельзя поиграть на других платформах). Об игре можно узнать её название, постер, жанр и год выхода.

Из данной темы прослеживается цель курсовой работы — создание мобильного приложения, решающего описанную проблему и имеющего привлекательный графический интерфейс. Для достижения поставленной цели необходимо выполнить определённые задачи. В первую очередь следует рассмотреть архитектуру ОС Android [1] для базового понимания технологических особенностей устройств, поддерживающих данную систему. Также важно изучить жизненный цикл мобильных приложений и все обязательные компоненты их разработки. Владение не только языком программирования Java для описания логики программы, но и знание языка разметки XML [2] поможет объединить графику приложения с его необходимыми процессами жизнедеятельности.

Объектом исследования курсовой работы является разработка мобильных приложений для ОС Android. Предметом исследования в данном случае выступает приложение для хранения информации о видеоиграх. Для правильного выполнения всей работы необходимо соблюдать определённые научные методы, среди которых особенно выделяются логические методы: анализ, системный подход и моделирование. Информационная база



исследования предстаёт в виде научной литературы, источников сети Интернет и, в частности, официальной документации [3] по разработке для ОС Android.

Структура данной работы для корректного описания процессов приложения состоит из разделов. Ниже будут представлены общие сведения, функциональное назначение, описание логической структуры, используемые технические средства, вызов и загрузка.

# **1 ОБЩИЕ СВЕДЕНИЯ**

В контексте данной курсовой работы было разработано приложение для хранения информации о видеоиграх.

## **1.1 Обозначение и наименование программы**

Наименование мобильного приложения — "DataGame", что в переводе означает данные игры.

## **1.2 Программное обеспечение, необходимое для функционирования программы**

Для функционирования программы необходимо установить приложение на мобильное устройство, работающее на базе ОС Android с версией Android 4.4 KitKat [4] или выше.

## **1.3 Языки программирования, на которых написана программа**

Данная программа написана на языке программирования Java. Также для построения графического интерфейса использовался язык разметки XML.

Java — строго типизированный объектно-ориентированный язык программирования. Данный язык активно применяется для создания программного обеспечения для многих устройств: обычных ПК, планшетов, смартфонов и мобильных телефонов. Характерной особенностью языка Java является то, что его код сначала транслируется в специальный байт-код, независимый от платформы, а затем этот байт-код выполняется виртуальной машиной JVM [5].

Язык XML применяется в макетах при создании программного продукта, которые определяют визуальную структуру пользовательского интерфейса операции или виджета приложения.

## **2    ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ**

Приложение "DataGame" предназначено для хранения информации о видеоиграх. Необходимо предоставить возможность выбора платформы пользователю. Во время использования программы пользователь может менять платформу. После выбора платформы пользователь может смотреть актуальные и ожидаемые игры для данной платформы.

Для понятного отображения информации об играх должен быть реализован механизм позволяющий переключаться между наблюдаемой игрой и остальными играми. Это выполнено стрелками “влево” и “вправо”.

## 3 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

В ходе разработки программного продукта создание логической структуры занимало важную часть всего процесса, так как от неё зависит эффективная и правильная работа приложения.

### 3.1 Декомпозиция

Исходный код приложения состоит из одного главного пакета.

#### 3.1.1 Пакет UI

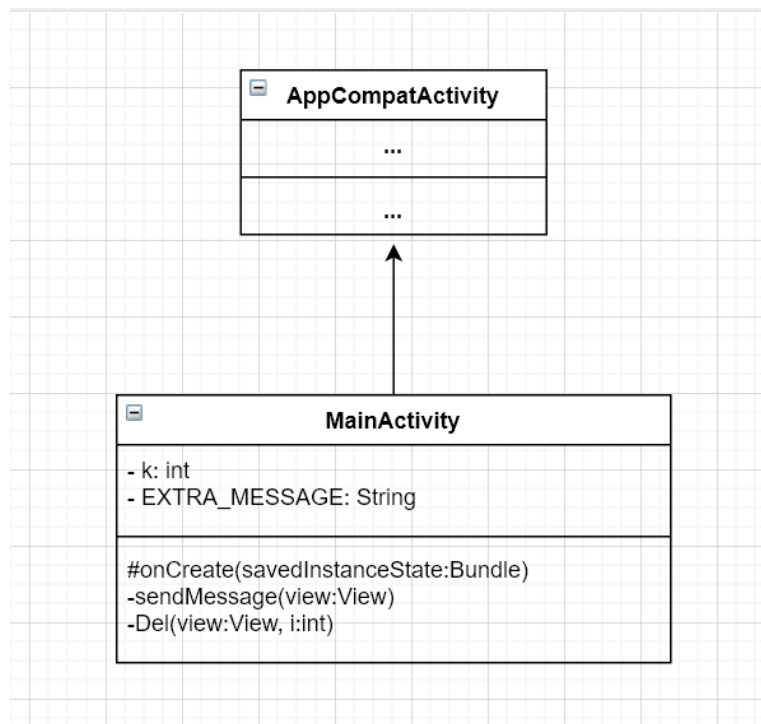
В пакете UI находятся следующие классы:

- 1) MainActivity — класс основной Activity приложения, запускающий отображение всех популярных платформ для выбора пользователя, реализованных анимированными кнопками с картинками, и после этого отображается кнопка для подтверждения.
- 2) Choose — класс, позволяющий выбрать тип игр.
- 3) EndActivity — класс фрагмента, отображающий картинку с игрой и всей информации об игре, а также содержащий две кнопки для переключения игр.
- 4) Game — класс фрагмента, логически описывающий игру.
- 5) Global — класс фрагмента для использования глобальных переменных.

### 3.2 Описание деталей

#### 3.2.1 Класс MainActivity

На рисунке 3.1 представлены UML-диаграммы [6] классов MainActivity и AppCompatActivity. Он наследуется от класса AppCompatActivity библиотеки Android.



**Рисунок 3.1 — UML-диаграмма класса MainActivity**

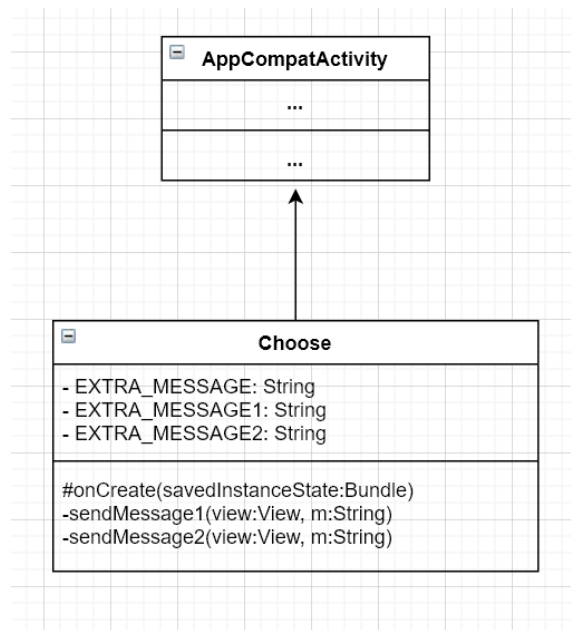
Рассмотрим подробнее методы класса MainActivity.

С метода `onCreate()` начинается выполнение основной активности. В ней производится подготовительная настройка, в частности, связываются кнопки с кодом, в котором описывается анимация кнопок и вызова метода `Del()` при нажатии на одну из кнопок.

Работа активности завершается методом `sendMessage()`, который возникает при подтверждении выбора платформы. После нажатие на подтверждающую кнопку продолжение программы переходит в следующую активность, передавая ей номер выбранной платформы. Выбор платформы повлияет на те игры, которые мы увидим для этой платформы.

### 3.2.2 Класс Choose

Перейдём к классу Choose. На рисунке 3.2 изображена его UML-диаграмма. Данный класс наследуется также только от класса AppCompatActivity библиотеки Android.



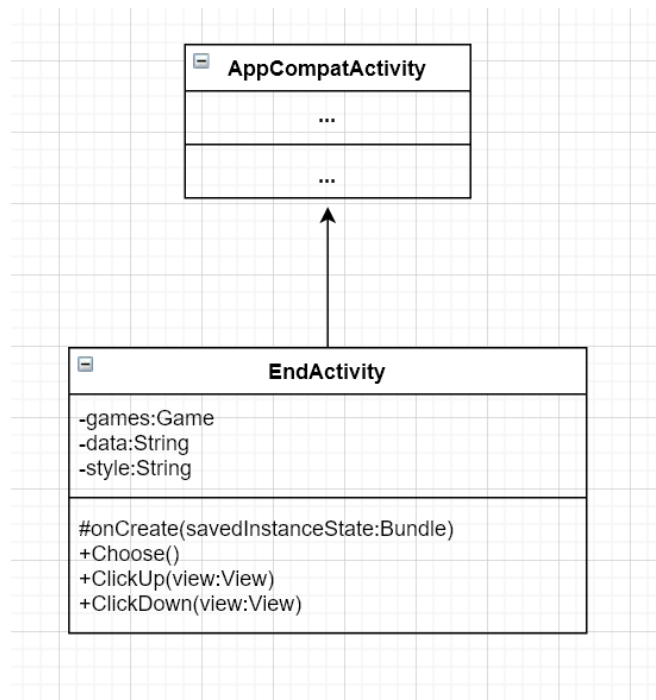
**Рисунок 3.2 — UML-диаграмма класса Choose**

С метода onCreate() начинается выполнение следующей активности. В ней производится подготовительная настройка, в частности, связываются кнопки с кодом, а также получается аргумент из основного активити.

Методы SendMessage1() и SendMessage2() определяют какая кнопка была нажата и в зависимости от этого передается аргумент в последнюю активность.

### **3.2.3 Класс EndActivity**

На рисунке 3.3 представлена UML-диаграмма класса EndActivity. Он наследуется от класса AppCompatActivity библиотеки Android как и все остальные классы.



**Рисунок 3.3 — UML-диаграмма класса EndActivity**

Метод `onCreate()` получает аргумент от прошлой активности и благодаря этому определяет какая была нажата кнопка до этого. В зависимости от этого заполняется массив данными.

Метод `Choose()` переключает игру через массив.

А в методах `ClickUp()` и `ClickDown()`, обращаясь к методу `Choose()`, обрабатывается нажатие на кнопки для переключения, при этом учитывается цикличность.

Собственно многомерный массив `games` и хранит всю информацию об играх. В методе `onCreate()` он заполняется данными.

### 3.2.4 Прочие классы

Остальные классы, такие как `Game` и `Global`, представленные в исходном коде программного продукта, являются утилитами и обращение к ним происходит статично.

## 3.3 Описание алгоритмов

### 3.3.1 Алгоритм перехода игры

В самом конце работы программы пользователь смотрит игры. Это возможно с помощью данного алгоритма из класса EndActivity. Для наглядности приведём блок-схему (рисунок 3.4) алгоритма.

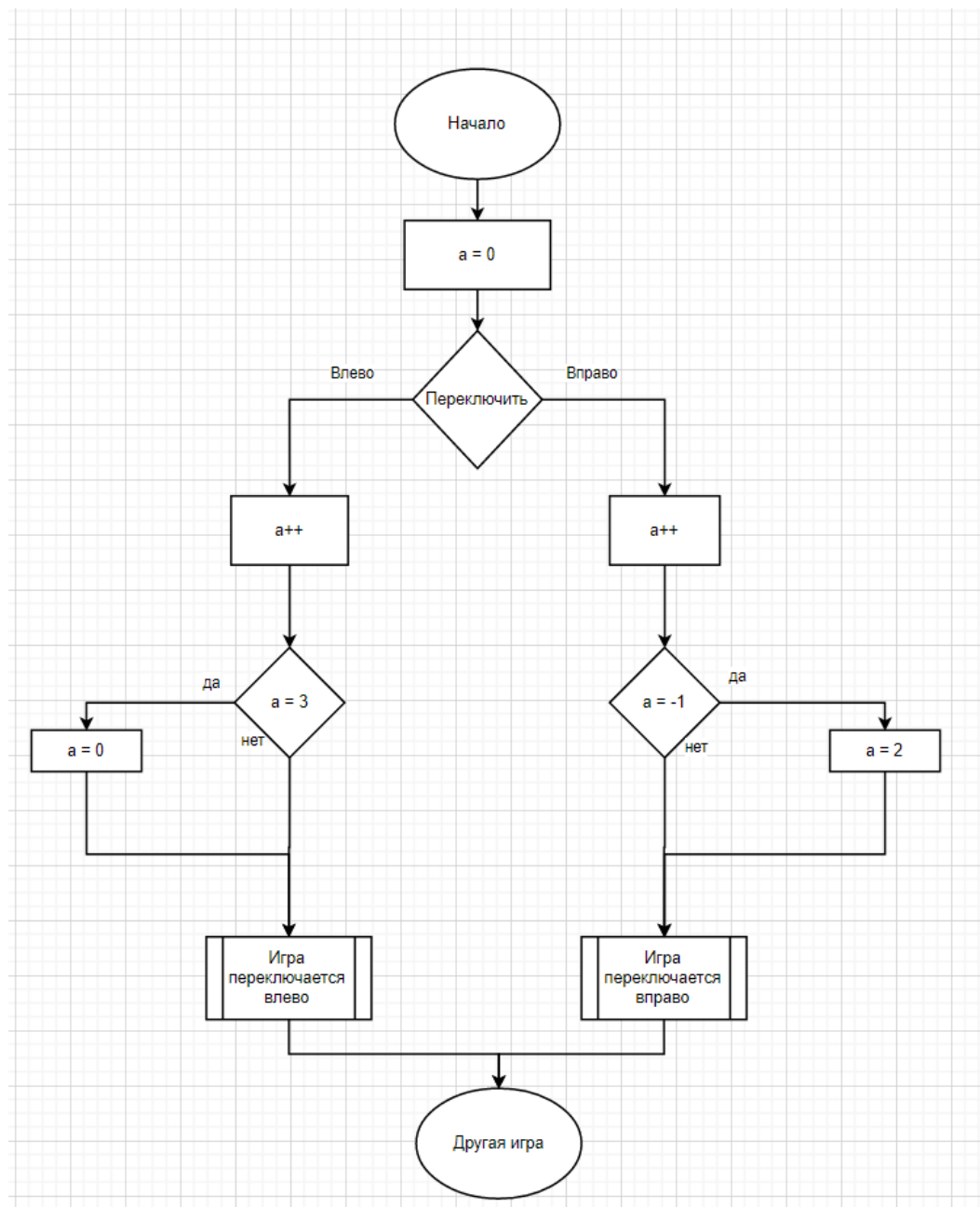


Рисунок 3.4 — Блок-схема алгоритма перехода игры

В данном алгоритме происходит считывание клика по выбранной кнопки пользователя и логичное смена игр.



### 3.3.2 Алгоритм выбора типа игр

Алгоритмы присутствуют и в классе Choose. В качестве примера приведём блок-схему кода передачи данных о типе игр. Соответствующие блок-схемы представлены на рисунках 3.5.

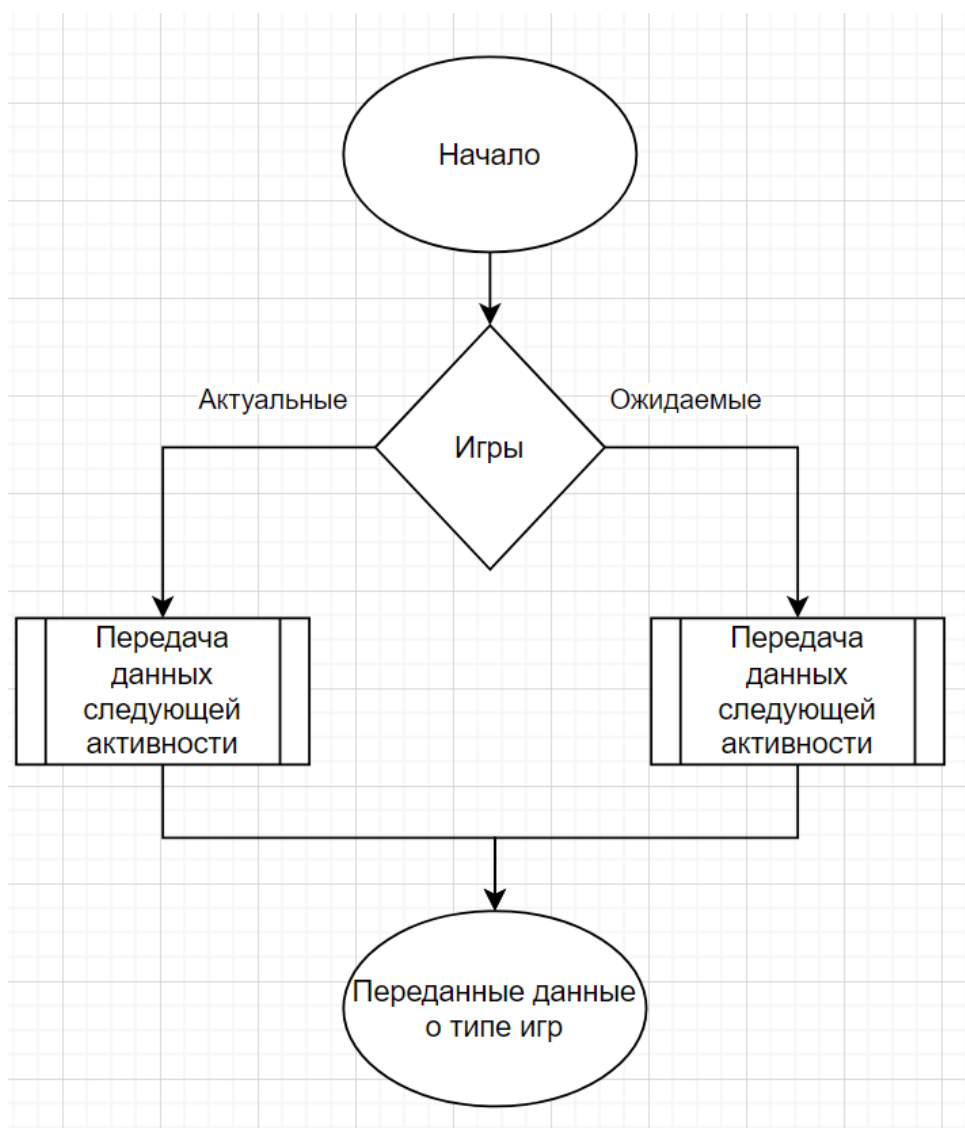


Рисунок 3.5 — Блок-схема алгоритма добавления данных

### 3.3.3 Другие алгоритмы

В классе EndActivity находятся алгоритмы обработки перехода к другой игре с помощью кнопок и переопределение названия, картинки, жанра и год выхода игры.

### 3.4 Описание графического интерфейса

Графический интерфейс приложения реализуется одним экраном, представленным в программном коде фрагментами. На рисунках ниже это будет продемонстрировано.

При запуске приложения пользователь будет видеть главный экран (рисунок 3.6). На нем изображены пять кнопок, четыре из которых, это игровые платформы. После выбора платформы, необходимо нажать на кнопку “Подтвердить”.

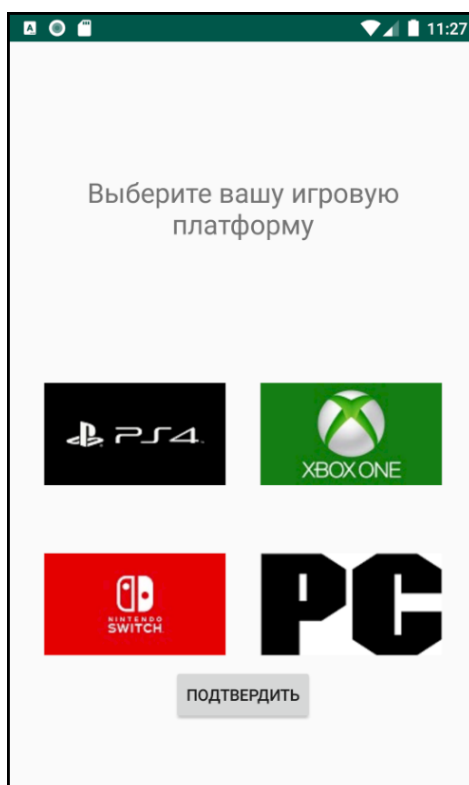


Рисунок 3.6 — Главный экран приложения

В зависимости от того какая платформа у пользователя, выбирается игровая платформа. От платформы будет зависеть список игр.

На рисунке 3.7 изображено второе окно приложения.



**Рисунок 3.7 — Второе окно приложения**

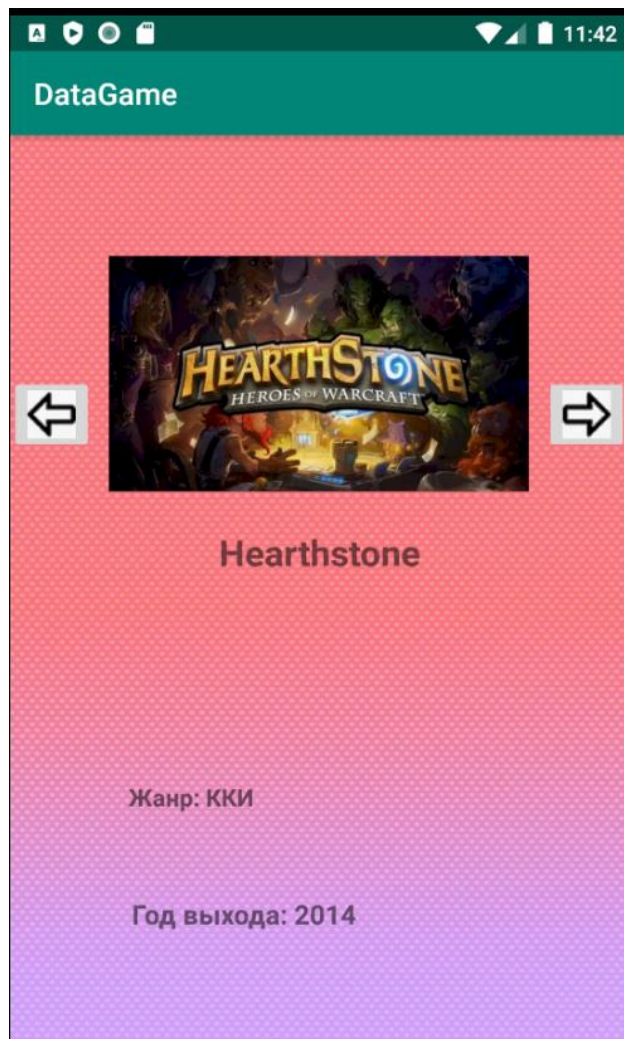
На втором окне представлены две кнопки, с помощью которых осуществляется подбор игр выбранного типа.

Появляется последнее окно с основной информацией об играх (рисунок 3.8).



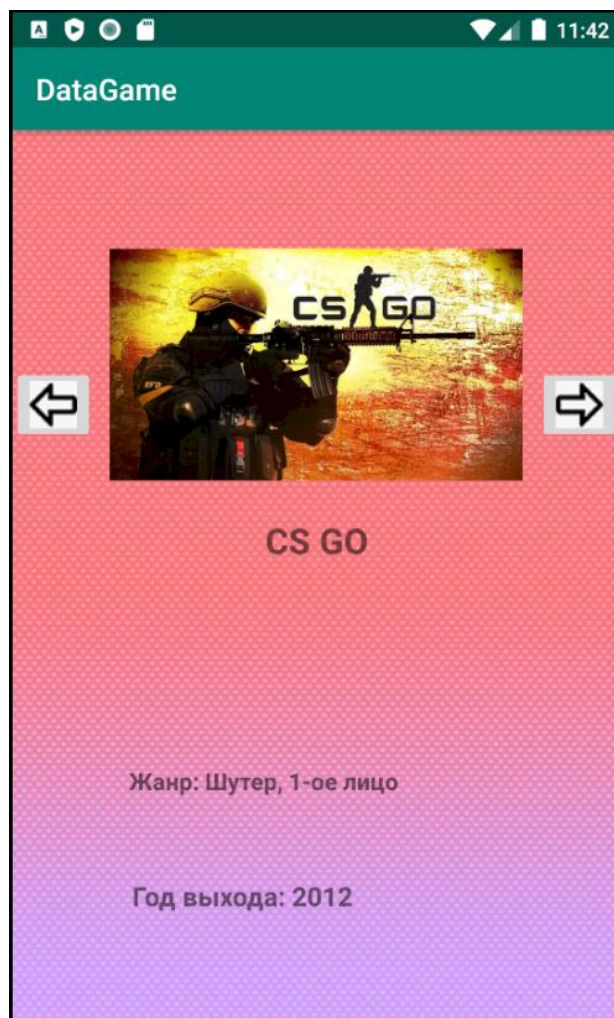
**Рисунок 3.8 — Окно с информацией об игре №1**

Если нажать на кнопку “влево”, то окно с игрой обновится (рисунок 3.9).



**Рисунок 3.9 — Окно с информацией об игре №2**

Игра поменялась. Нажмем еще раз на кнопку “влево” (рисунок 3.10).



**Рисунок 3.10 — Окно с информацией об игре №3**

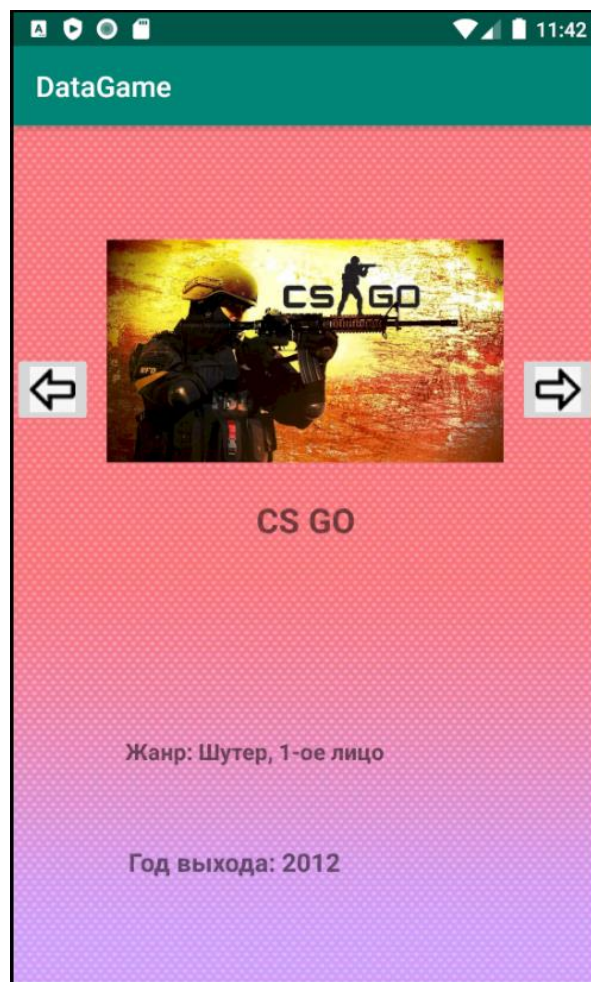
И игра снова изменилась. Нажав ещё раз на кнопку “влево”, (рисунок 3.11) вернется начальная игра, потому что для каждого типа игр и платформ было выбрано по 3 игры.



**Рисунок 3.11 — Окно с информацией об игре №3**

Теперь, если нажать на кнопку “вправо”, то игра меняется на предыдущую (рисунок 3.12).





**Рисунок 3.12 — Окно с информацией об игре №4**

Вернемся к выбору типа игр и выберем “Ожидаемые игры” (рисунок 3.13).





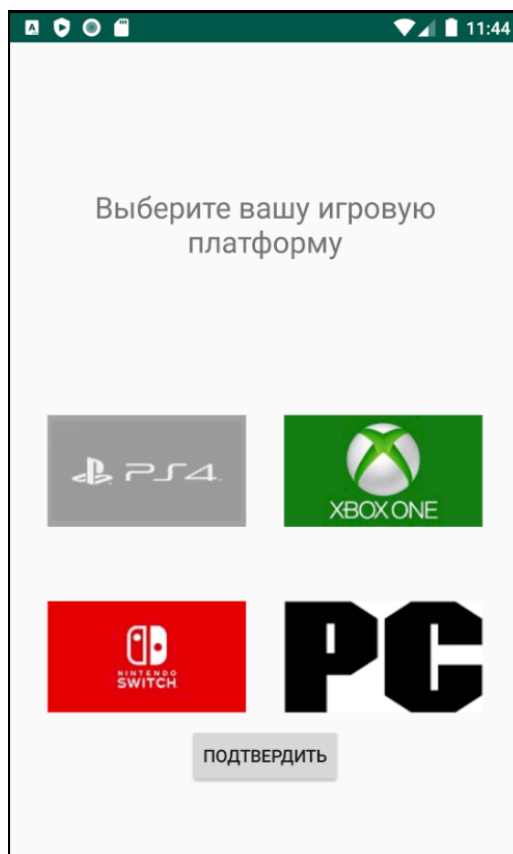
**Рисунок 3.13 — Окно с информацией об игре №5**

Представленная игра еще не вышла. И наконец сменим платформу и убедимся, что для других тоже все работает (рисунок 3.14).



**Рисунок 3.14 — Окно с информацией об игре №6**

Выбранная игра уже вышла на платформу “PS4”. Все получилось. В конце также надо продемонстрировать анимацию кнопок при выборе платформ. (рисунок 3.15)



**Рисунок 3.15 — Анимация кнопок в первом окне**  
При нажатии на кнопку, она мигает.

## **4 ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА**

Для полноценного функционирования программного продукта необходимо 10,84 МБ свободного места во внутреннем хранилище мобильного устройства.

## 5 ВЫЗОВ И ЗАГРУЗКА

Для начала работы с приложением "DataGame" достаточно установить его на мобильное устройство на базе ОС Android, загрузив файл DataGame.apk, в который упакован архив программы. Установка занимает мало времени и после её завершения появится соответствующая иконка, изображённая на рисунке 5.1, с названием приложения, что является точкой входа в программу.



**Рисунок 5.1 — Иконка приложения**

## ЗАКЛЮЧЕНИЕ

В процессе выполнения курсовой работы была выполнена поставленная в самом начале цель — разработать мобильное приложение, удобно хранящее информацию об вышедших и ожидаемых под разные платформы.

Для достижения цели работы были выполнены все сопутствующие задачи по мере разработки мобильного приложения. Особое внимание уделялось теоретической части создания программного продукта, благодаря чему удалось сформулировать ряд выводов.

Также были сформулированы определённые практические предложения. Например, при разработке мобильного приложения очень важно внимательно изучить доступные библиотеки системы Android. С их помощью можно достигнуть эффективной работы всех частей программы.

Отдельно стоит отметить разработку интерфейса приложения. Благодаря большому разнообразию доступных графических компонентов можно создать свой уникальный дизайн. Но перед этим стоит внимательно рассмотреть работу выбранного элемента для интерфейса. Соблюдая ряд данных рекомендаций, в результате получится удобное в использовании приложение.

Перед вводом приложения в полную эксплуатацию оно должно пройти ряд обязательных тестов. Следует рассмотреть все возможные варианты использования программного продукта и предотвратить последующие ошибки при их возникновении. Только тогда можно утверждать о законченности и целостности мобильного приложения.

В результате курсовой работы было освоено большое количество теоретического материала и получены определённые практические навыки. Созданное мобильное приложение работает без ошибок.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гриффитс, Д. Head First. Программирование для Android / Д. Гриффитс, Д. Гриффитс — 2-е изд. — СПб.: Питер, 2018 — 912 с.: ил.
2. Одиночкина, С. В. Основы технологий XML: учебное пособие / С. В. Одиночкина — СПб: НИУ ИТМО, 2013. — 56 с.
3. Документация по платформе Android [Электронный ресурс]. — URL: <https://developer.android.com/guide?hl=ru> (дата обращения: 11.05.2020).
4. Версии ОС Android [Электронный ресурс]. — URL: [https://en.wikipedia.org/wiki/Android\\_version\\_history](https://en.wikipedia.org/wiki/Android_version_history) (дата обращения: 11.05.2020).
5. Шилдт, Г. Java. Полное руководство / Г. Шилдт — 10-е изд.: Пер. с англ. — СПб.: ООО «Альфа-книга», 2018 — 1488 с.: ил. — Парал. тит. англ.
6. Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, И. Якобсон — 2-е изд.: Пер. с англ. Мухин Н. — М.: ДМК Пресс, 2006. — 496 с.: ил.

## **ПРИЛОЖЕНИЯ**

Приложение А — Техническое задание

Приложение Б — Исходный код



## **Приложение А**

# **1 Техническое задание**

## **1.1 Введение**

Составленное техническое задание по дисциплине «Разработка программных приложений» является документом к курсовой работе, который отражает все этапы разработки программного продукта, а также процесс проектирования и выявление требований, предъявляемых конечному продукту.

### **1.1.1 Наименование программы**

Наименование данного мобильного приложения связано с той же сферой жизни, которая обозначена в теме курсовой работы: «Разработка программного приложения для хранения информации о видеоиграх», а именно игровой. Из этого выходит название программного продукта — "DataGame", что в переводе с английского означает данные игры.

### **1.1.2 Краткая характеристика области применения программы**

Программный продукт предназначен для небольших областей, связанных с развлечением людей. Приложение подойдёт тем, кому нравятся игры и следить за ними.

## **1.2 Основание для разработки**

Основанием для разработки является задание на выполнение курсовой работы, утверждённое институтом информационных технологий, кафедрой инструментального и прикладного программного обеспечения, по дисциплине «Разработка программных приложений», предусмотренной учебным планом подготовки бакалавров 09.03.04 «Программная инженерия».

Также основанием для разработки является потребность в удобном в использовании и полном по функционалу мобильном приложении для хранения информации о видеоиграх.

### **1.3 Назначение для разработки**

Назначением разработанного приложения является хранения информации о видеоиграх для нескольких платформ.

### **1.4 Требования, предъявляемые к программе**

Программа должна отображать массив игр, который можно посмотреть и информацию о них.

#### **1.4.1 Взаимодействие продукта с другими продуктами и компонентами**

Программный продукт не взаимодействует с другими продуктами и компонентами.

#### **1.4.2 Функциональные требования**

Мобильное приложение должно предоставлять пользователям следующие функции:

- 1) Выбор игровой платформы.
- 2) Выбор типа игры.
- 3) Просмотр информации о играх под данную платформу и тип игр.
- 4) Возможность выбрать другую игровую платформу и другой тип игры в одном запуске приложения.

#### **1.4.3 Требования к внешним интерфейсам**

Для корректной работы приложения устройству не требуется доступ к сети Интернет.

#### **1.4.4 Требования к производительности**

Программный продукт должен стабильно работать таким образом, чтобы действия пользователя не вызвали аварийного завершения программы. Также все функции приложения должны отзываться на действия пользователя.

#### **1.4.5 Нефункциональные требования**

Требования к доступности и безопасности приложения не предъявляются.

### **1.5 Требования к программной документации**

Требования к программной документации должны соответствовать:

- 1) Рекомендации по оформлению письменных работ обучающихся по образовательным программам бакалавриата, специалитета и магистратуры (СМКО МИРЭА 7.5.1/03.П.69-18).
- 2) Отчётной документации, составленной в соответствии с ГОСТ.

В процессе создания мобильного приложения вся проделанная работа документируется, следует описать все детали разработки. Всё вышеперечисленное должно быть отражено в пояснительной записке, которая прилагается к работе.

### **1.6 Технико-экономические показатели**

Технико-экономические требования к работе не предъявляются.

### **1.7 Стадии и этапы разработки**

Этапы разработки:

- 1) Исследование концепции: 01.05.2020 – 01.05.2020.
- 2) Выработка требования и составление ТЗ: 02.05.2020 – 02.05.2020.
- 3) Проектирование: 03.05.2020 – 10.05.2020.
- 4) Написание логики программы: 10.05.2020 – 10.05.2020.
- 5) Написание графического интерфейса: 10.05.2020 – 12.05.2020.

- 6) Отладка созданного приложения: 12.05.2020 – 12.05.2020.
- 7) Оформление отчётной документации: 12.05.2020 – 14.05.2020.
- 8) Представление работы к защите: 14.05.20 – 15.05.20

Дальнейшее сопровождение программы после сдачи и защиты курсового проекта не предусмотрено.

## **1.8 Порядок контроля и приема работы**

Приём работы осуществляется путём защиты курсовой работы у преподавателя, назначенного руководителем курсовой работы. Защита осуществляется в соответствии с установленным графиком защиты курсовых работ.

## Приложение Б

### Листинг программного кода

#### Листинг 1.1 — Файл MainActivity.java

```
package com.example.datagame;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;

public class MainActivity extends AppCompatActivity {
    int k = 0;
    public final static String EXTRA_MESSAGE = "EXTRA_MESSAGE";

    public void sendMessage(View view) {
        Intent intent = new Intent(this, Choose.class);
        intent.putExtra(EXTRA_MESSAGE, k);
        startActivity(intent);
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        final Animation animAlpha = AnimationUtils.loadAnimation(this, R.anim.alpha);
        final ImageButton btnAlpha1 = (ImageButton) findViewById(R.id.alpha1);
        btnAlpha1.setOnClickListener(new ImageButton.OnClickListener() {
            @Override
            public void onClick(View view) {
                view.startAnimation(animAlpha);
                Del(view,1);
            }
        });

        ImageButton btnAlpha2 = (ImageButton) findViewById(R.id.alpha2);
```

```

btnAlpha2.setOnClickListener(new ImageButton.OnClickListener() {
    @Override
    public void onClick(View view) {
        view.startAnimation(animAlpha);
        Del(view,2);
    }
});

ImageButton btnAlpha3 = (ImageButton) findViewById(R.id.alpha3);
btnAlpha3.setOnClickListener(new ImageButton.OnClickListener() {
    @Override
    public void onClick(View view) {
        view.startAnimation(animAlpha);
        Del(view,3);
    }
});

ImageButton btnAlpha4 = (ImageButton) findViewById(R.id.alpha4);
btnAlpha4.setOnClickListener(new ImageButton.OnClickListener() {
    @Override
    public void onClick(View view) {
        view.startAnimation(animAlpha);
        Del(view,4);
    }
});
}

public void Del(View view, int i) {
    final Button Finalbutton = (Button) findViewById(R.id.button2);
    Finalbutton.setVisibility(View.VISIBLE);
    ImageButton btnAlpha1 = (ImageButton) findViewById(R.id.alpha1);
    ImageButton btnAlpha2 = (ImageButton) findViewById(R.id.alpha2);
    ImageButton btnAlpha3 = (ImageButton) findViewById(R.id.alpha3);
    ImageButton btnAlpha4 = (ImageButton) findViewById(R.id.alpha4);
    k = i;
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
}

```

Листинг 1.3 — Файл MainActivity.java

```
        return true;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

## Листинг 2.1 — Файл Choose.java

```
package com.example.datagame;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;

public class Choose extends AppCompatActivity {

    public static String EXTRA_MESSAGE;
    public final static String EXTRA_MESSAGE1 = "EXTRA_MESSAGE1";
    public final static String EXTRA_MESSAGE2 = "EXTRA_MESSAGE2";

    public void sendMessage1(View view, String m) {
        EXTRA_MESSAGE = EXTRA_MESSAGE1;
        Intent intent = new Intent(this, EndActivity.class);
        intent.putExtra(EXTRA_MESSAGE, m);
        startActivity(intent);
    }

    public void sendMessage2(View view, String m) {

        EXTRA_MESSAGE = EXTRA_MESSAGE2;
        Intent intent = new Intent(this, EndActivity.class);
        intent.putExtra(EXTRA_MESSAGE, m);
        startActivity(intent);
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Intent intent = getIntent();
        Bundle arguments = getIntent().getExtras();

        final String message = arguments.get(MainActivity.EXTRA_MESSAGE).toString();

        setContentView(R.layout.activity_choose);

        final Button button1 = (Button) findViewById(R.id.button);
```



Листинг 2.2 — Файл Choose.java

```
button1.setOnClickListener(new ImageButton.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        sendMessage1(view, message);  
    }  
  
});  
  
final Button button2 = (Button) findViewById(R.id.button3);  
button2.setOnClickListener(new ImageButton.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        sendMessage2(view, message);  
    }  
  
});  
  
}  
}
```

### Листинг 3.1 — Файл EndActivity.java

```
package com.example.datagame;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;

public class EndActivity extends AppCompatActivity {
    Game[][] games = new Game[4][3];
    final String data = "Год выхода: ";
    final String style = "Жанр: ";

    public void ClickUp(View view) {
        Global.a++;
        if (Global.a == 3) Global.a = 0;
        Choose();
        // textView1.setText(games[i][k].name);
    }

    public void ClickDown(View view) {
        Global.a--;
        if (Global.a == -1) Global.a = 2;
        Choose();
    }

    void Choose()
    {
        TextView editText1 = (TextView) findViewById(R.id.nameOfGame);
        editText1.setText(games[Global.b][Global.a].name);
        TextView editText2 = (TextView) findViewById(R.id.styleGame);
        editText2.setText(style + games[Global.b][Global.a].description);

        TextView editText3 = (TextView) findViewById(R.id.yearRealize);
        editText3.setText(data + games[Global.b][Global.a].year);

        ImageView imageView = (ImageView) this.findViewById(R.id.imageView);
        imageView.setImageResource(games[Global.b][Global.a].view);
    }
}
```

### Листинг 3.2. — Файл EndActivity.java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_end);
    Bundle arguments = getIntent().getExtras();

    String message;
    boolean soon = false;
    message = arguments.get(Choose.EXTRA_MESSAGE).toString();
    if (Choose.EXTRA_MESSAGE == "EXTRA_MESSAGE2")
        soon = true;
    if (soon)
    {
        games[0][0] = new Game("Cyberpunk 2077", R.drawable.cyberpunk2077, "Экшн, РПГ,
1-ое лицо", "2020");
        games[0][1] = new Game("The Last of Us 2", R.drawable.lastofus2, "Экшн,
приключение, стелс", "2020");
        games[0][2] = new Game("Ghost of Tsushima", R.drawable.ghostoftsushima, "Экшн,
приключение, стелс", "2020" );

        games[1][0] = games[0][0];
        games[1][1] = new Game("Halo Infinite", R.drawable.halo, "Шутер, 1-ое
лицо", "2020" );
        games[1][2] = new Game("Dying Light 2", R.drawable.dl2, "Экшн, РПГ, 1-ое
лицо", "2020" );
        games[2][0] = new Game("The Legend of Zelda: Breath of the Wild 2",
R.drawable.zelda, "Экшн, приключение", "2020" );
        games[2][1] = new Game("Metroid Prime 4", R.drawable.metroid, "Экшн", "2020" );
        games[2][2] = new Game("No More Heroes 3",
R.drawable.nomanheroes3, "Боевик", "2020" );
        games[3][0] = games[0][0];
        games[3][1] = games[1][2];
        games[3][2] = new Game("Diablo 4", R.drawable.diablo4, "Экшн, РПГ, ММО", "2021" );
    }
else
    {
        games[0][0] = new Game("God of War", R.drawable.godofwar, "Экшн, приключение",
"2018");
    }
}
```

### Листинг 3.3. — Файл EndActivity.java

```
        games[0][1] = new Game("The Last of Us", R.drawable.lastofus, "Экшн,
приключение, стелс", "2013");
        games[0][2] = new Game("Horizon Zero Dawn", R.drawable.hzd,"Экшн, приключение,
РПГ","2017" );

        games[1][0] = new Game("The Witcher 3", R.drawable.w3,"Экшн, РПГ","2015" );
        games[1][1] = new Game("Doom Eternal", R.drawable.doom,"Шутер, 1-ое лицо","2020"
);
        games[1][2] = new Game("Geers 5", R.drawable.geers5,"Шутер, 3-ое лицо","2019" );
        games[2][0] = new Game("The Legend of Zelda: Breath of the Wild",
R.drawable.zelda1,"Экшн, приключение","2017" );
        games[2][1] = new Game("The Witcher 3", R.drawable.w3,"Экшн, РПГ","2019" );
        games[2][2] = new Game("Animal Crossing: New Horizons",
R.drawable.animal,"Симулятор жизни","2020" );
        games[3][0] = games[1][0];
        games[3][1] = new Game("Hearthstone", R.drawable.hs,"ККИ","2014" );
        games[3][2] = new Game("CS GO", R.drawable.csgo,"Шутер, 1-ое лицо","2012" );
    }

    switch (message)
    {
        case("1"):
            Global.b = 0;
            Choose();
            break;
        case("2"):
            Global.b = 1;
            Choose();
            break;
        case("3"):
            Global.b = 2;
            Choose();
            break;
        case("4"):
            Global.b = 3;
            Choose();
            break;
    }
}
```