Санкт-Петербургский государственный политехнический университет

Институт компьютерных наук и технологий

Курсовая работа

Дисциплина: Проектирование мобильных приложений

Тема: Логические комбинации.

Выполнил студент гр. 33531/2 Тюшин А. М.

Преподаватель

Санкт-Петербург

2018

Оглавление

[Список иллюстраций 3](#_Toc533404380)

[Введение: 4](#_Toc533404381)

[Цель 4](#_Toc533404382)

[Задачи 4](#_Toc533404383)

[Описание работы 4](#_Toc533404384)

[Ход работы: 5](#_Toc533404385)

[Создания макетов 5](#_Toc533404386)

[Создание логики 10](#_Toc533404387)

[Графическая обработка 17](#_Toc533404388)

[Тестирование 18](#_Toc533404389)

[Структура приложения 21](#_Toc533404390)

[Вывод 22](#_Toc533404391)

[Список источников 23](#_Toc533404392)

# Список иллюстраций

[Рисунок 1 activity\_main.xml. 5](#_Toc533739940)

[Рисунок 2 activity\_main.xml(land) 6](#_Toc533739941)

[Рисунок 3 in\_game.xml(part1) 7](#_Toc533739942)

[Рисунок 4 in\_game.xml(part2) 8](#_Toc533739943)

[Рисунок 5 strings.xml 8](#_Toc533739944)

[Рисунок 6 colors.xml 9](#_Toc533739945)

[Рисунок 7 OnClickListener 10](#_Toc533739946)

[Рисунок 8 ServiceForMusic.java 11](#_Toc533739947)

[Рисунок 9 Методы вызванные в MainActivity.java 11](#_Toc533739948)

[Рисунок 10 Обработка нажатия на кнопку звука 12](#_Toc533739949)

[Рисунок 11 InGame.java(part1) 13](#_Toc533739950)

[Рисунок 12 Метод countResult 14](#_Toc533739951)

[Рисунок 13 Метод onSaveInstanceState 15](#_Toc533739952)

[Рисунок 14 Востановление состояний 15](#_Toc533739953)

[Рисунок 15 onBackPressed 16](#_Toc533739954)

[Рисунок 16 Логотип 17](#_Toc533739955)

[Рисунок 17 Объекты для главного меню 17](#_Toc533739956)

[Рисунок 18 Тестирование 18](#_Toc533739957)

[Рисунок 19 UI Тестирование 18](#_Toc533739958)

[Рисунок 20 Конечный результат (part1) 19](#_Toc533739959)

[Рисунок 21 Конечный результат (part2) 20](#_Toc533739960)

[*Рисунок 22 Структура приложения* 21](#_Toc533739961)

# Введение:

## Цель

Целью данной работы является создание разработанного приложения на телефон с операционной системой Android. Несомненно, ещё одной из важных целей является приобретение новых знаний и умений в области проектирования мобильных приложений.

## Задачи

1. Создание макетов
2. Создание логики
3. Графическая обработка
4. Тестирование

## Описание работы

В качестве курсовой работы мной было выбрано создание приложения, в котором игроку нужно расставить знаки между цифрами от 1 до 9 так, чтобы получилось число, которое задано на экране. Из возможных для пользователя знаков будут такие арифметические операции как сложение, вычитание, умножение, деление, возведение в степень, а также соединения двух соседних цифр в одну. Ещё будет реализован сервис для проигрывания фоновой музыки. И наконец, в работе будет отдельная страница с описанием игрового процесса.

# Ход работы:

## Создания макетов

Макет определяет визуальную структуру пользовательского интерфейса, например, пользовательского интерфейса операции или виджета приложения. В работе используется **объявление элементов пользовательского интерфейса в XML** С помощью справочника XML-элементов, который имеется в Android, можно быстро и просто создавать макеты пользовательского интерфейса и содержащиеся в нем элементы.

Приведу примеры созданных макетов.



Рисунок 1 activity\_main.xml.

В данном макете, за основной Layout я взял LinearLayout, в который я поместил ConstrainLayuot для отображения логотипа и кнопки звука, и LinearLayout, в котором я расположил все кнопки, отвечающие за переходы на другие Activity.

Также был создан альтернативный макет, который отображается при повороте.



Рисунок 2 activity\_main.xml(land)

В нем я меняю отображение кнопок, помещая их в TableLayout, таким образом они намного лучше выглядят при повороте экрана.

Также были созданы макеты и для остальных Activity.



Рисунок 3 in\_game.xml(part1)



Рисунок 4 in\_game.xml(part2)

При создании различных свойств объекта необходимо держать константы в специальных отведённых для них местах.

Например все используемые строки описываются в папке ресурсов в соответствующем файле strings.xml.

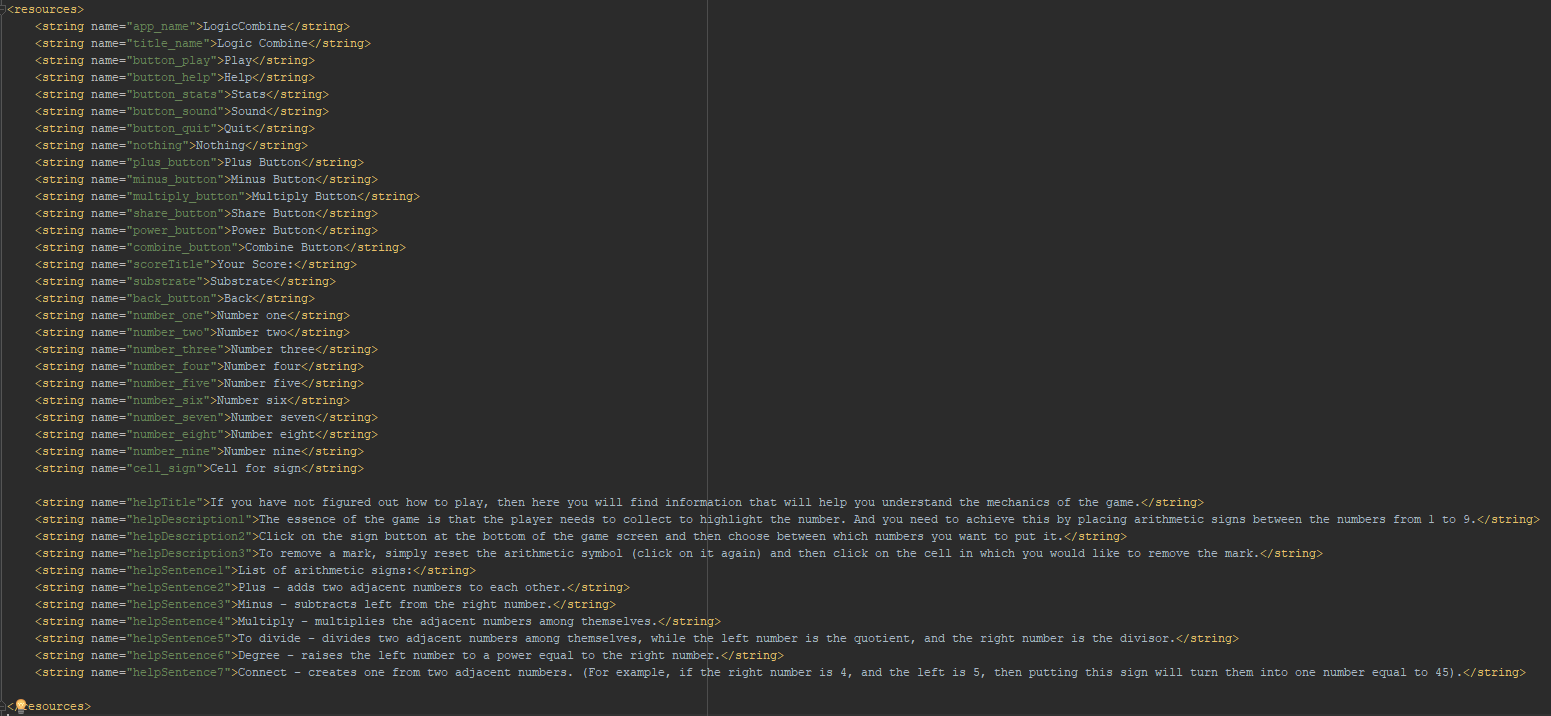


Рисунок 5 strings.xml

А цвета в colors.xml.

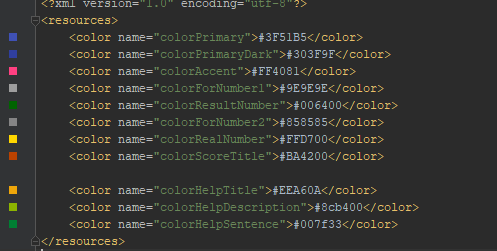


Рисунок 6 colors.xml

## Создание логики

Для того, чтобы пользователь мог взаимодействовать с объектами, описанными в макете, нужно написать java код, который будет обрабатывать действия пользователя.

Например для того, чтобы при нажатии на кнопку игрок переходил в соответствующее Activity нужно обработать нажатие программно.

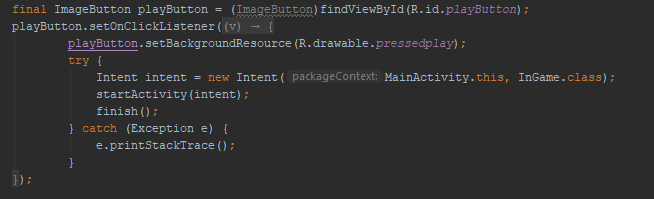


Рисунок 7 OnClickListener

Для реализации фоновой музыки был использован сервис, который запускается если у пользователя стоит воспроизведение музыки, или же отключается, если пользователь выключает её.

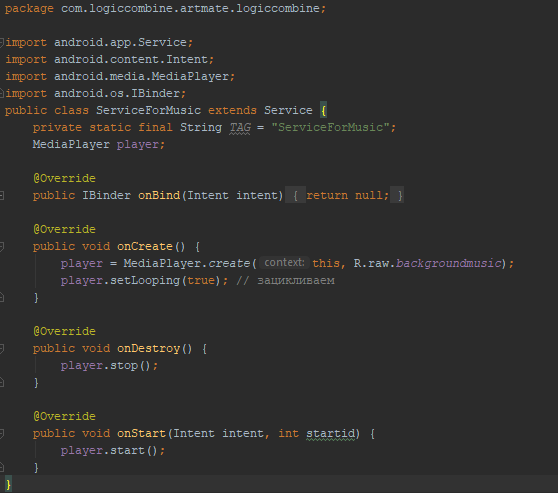


Рисунок 8 ServiceForMusic.java

Музыка была помещена в ресурсы. А методы данного класса непосредственно используются при изменении музыки в остальных Activity.

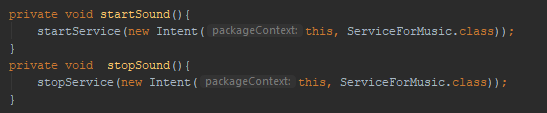


Рисунок 9 Методы вызванные в MainActivity.java

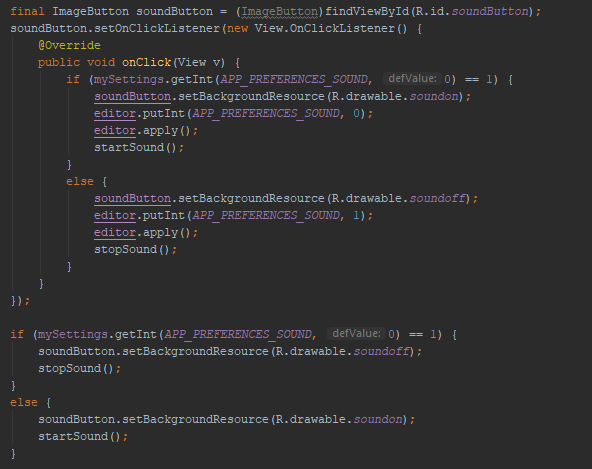


Рисунок 10 Обработка нажатия на кнопку звука

Можно заметить, что я также использовал задачу настроек приложения, а именно звука. Это используется для того, чтобы выбор игрока оставался таким же и при перезапуске приложения.

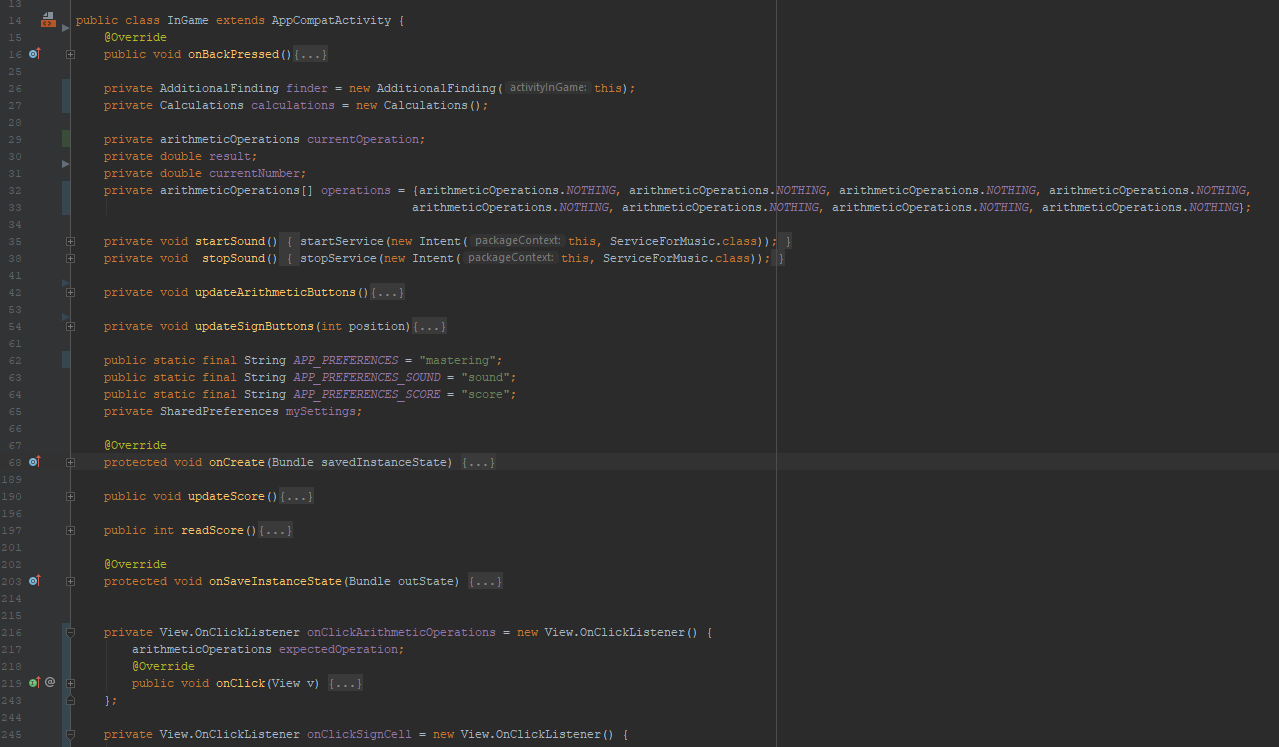
Логика самой игры описана в файле InGame.java

Рисунок 11 InGame.java(part1)

Для удобной структуризации коды использовано разбиение на регионы, при помощи //region…//endregion.

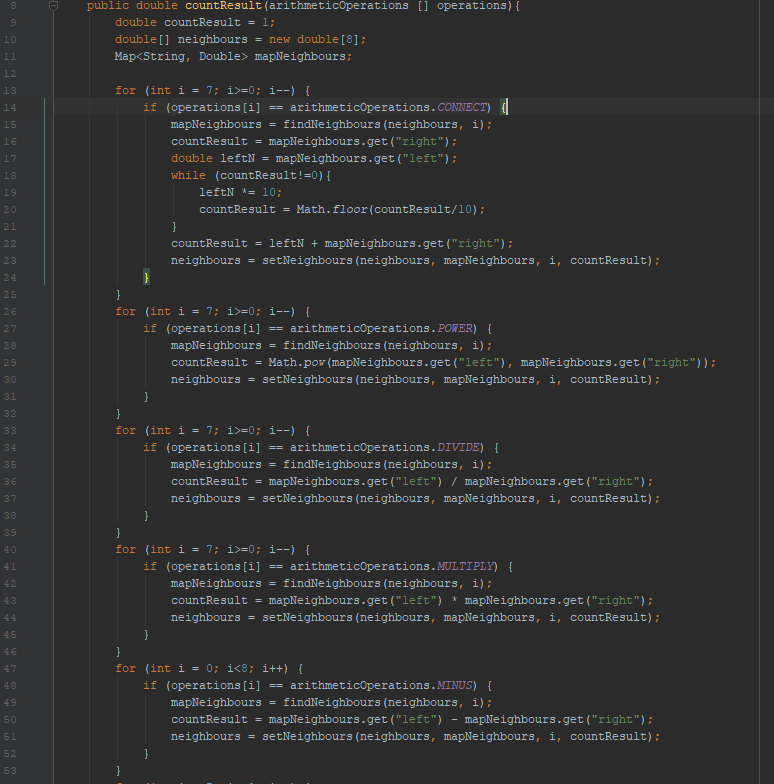


Рисунок 12 Метод countResult

…

Подсчет результата ведём в отдельном классе Calculations.

Так как у нас при повороте используется альтернативный ресурс, то наш Activity пере собирается, а следственно теряет важные данные, которые были записаны во время игры. Чтобы не допустить этого я воспользуюсь специальным методом onSaveInstanceState.

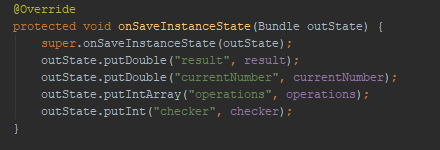


Рисунок 13 Метод onSaveInstanceState

После чего при его существовании учтём это в методе onCreate.

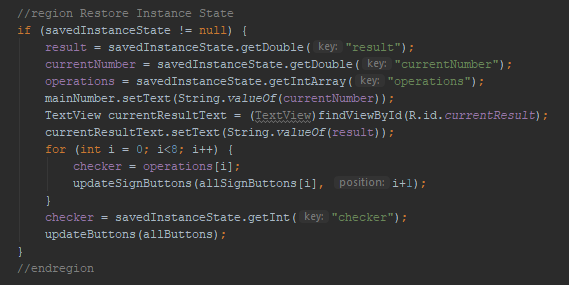


Рисунок 14 Востановление состояний

Также обработаем все действия пользователя, как на этом Activity, так и на всех остальных, включая нажатие на кнопку назад непосредственно на самом телефоне.

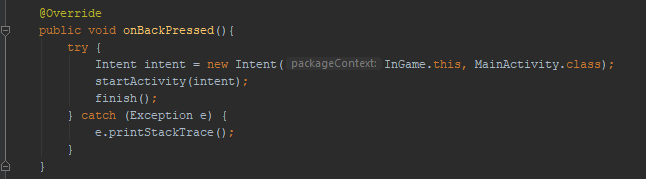


Рисунок 15 onBackPressed

## Графическая обработка

Практически все изображения используемы в работе я создавал в программе Photoshop.

Примеры:



Рисунок 16 Логотип

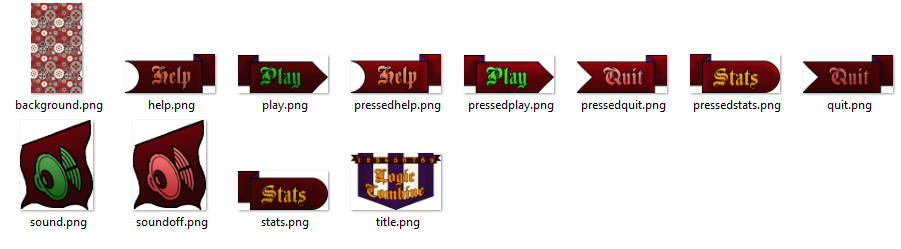


Рисунок 17 Объекты для главного меню

## Тестирование

Для проверки работоспособности приложения были написаны и успешно выполнены Unit тесты.

Пример проверка работоспособности метода подсчёта.

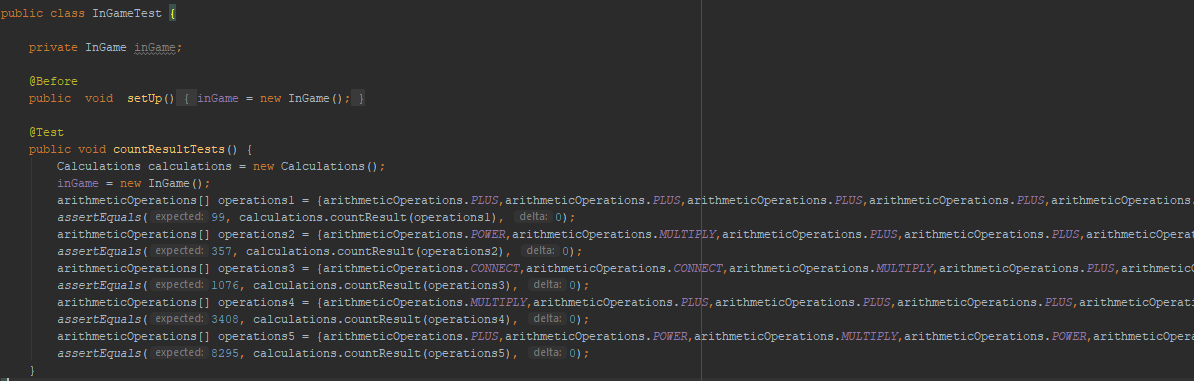


Рисунок 18 Тестирование

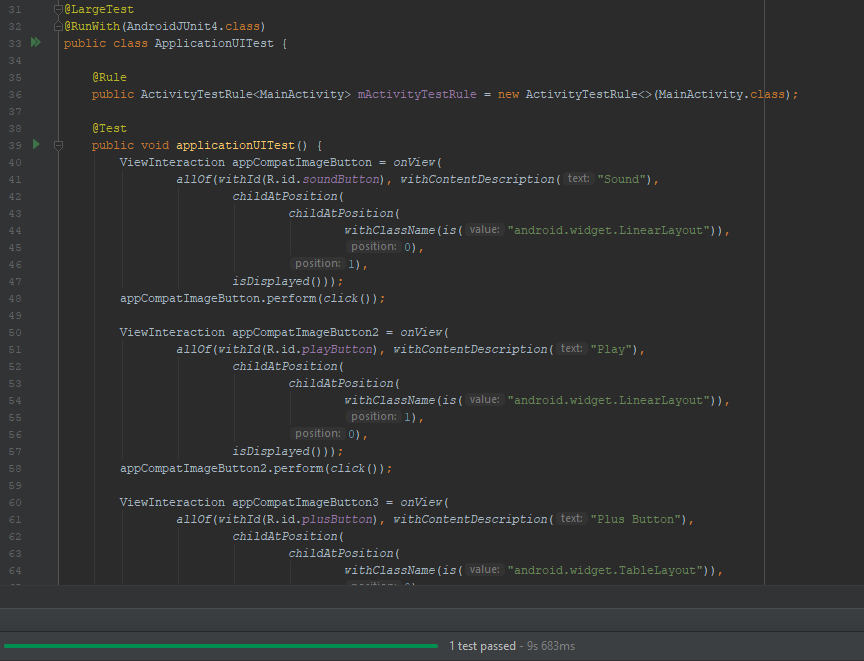


Рисунок 19 UI Тестирование

Помимо этого, были разработаны и успешно протестированы UI тесты.

Также приложение было проверено непосредственно в использовании на эмуляторе и на реальном устройстве. Помимо этого, было просмотрены несколько различных разрешений экрана для того, чтобы убедится, что объекты правильно масштабируются.

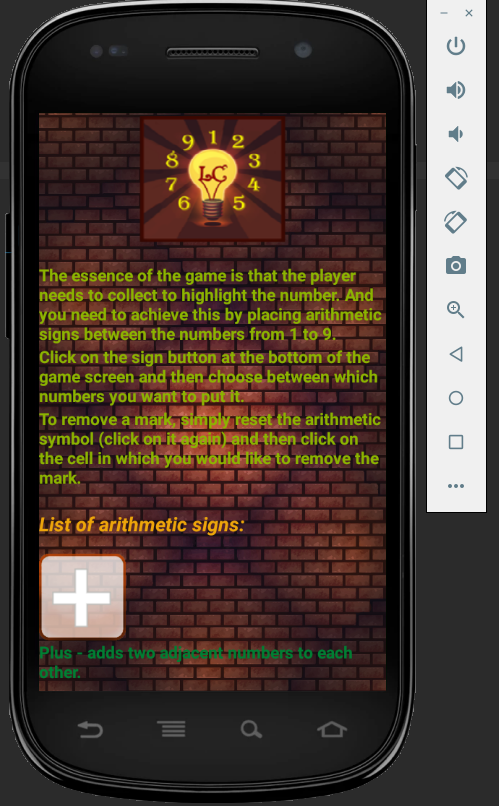
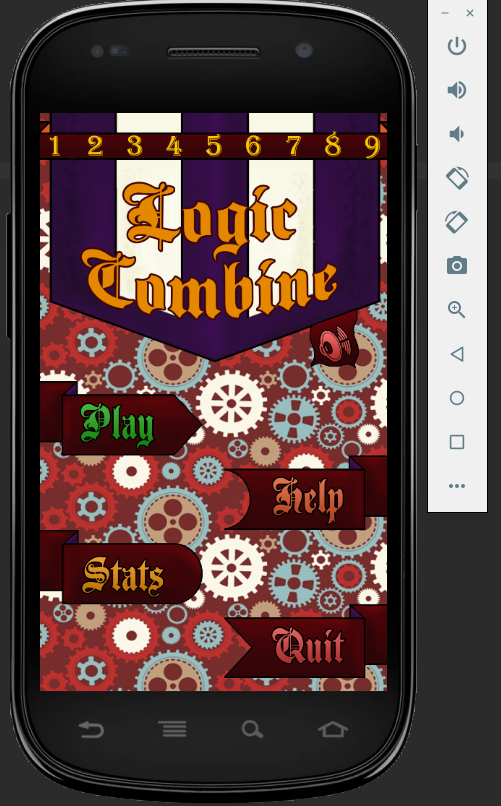


Рисунок 20 Конечный результат (part1)

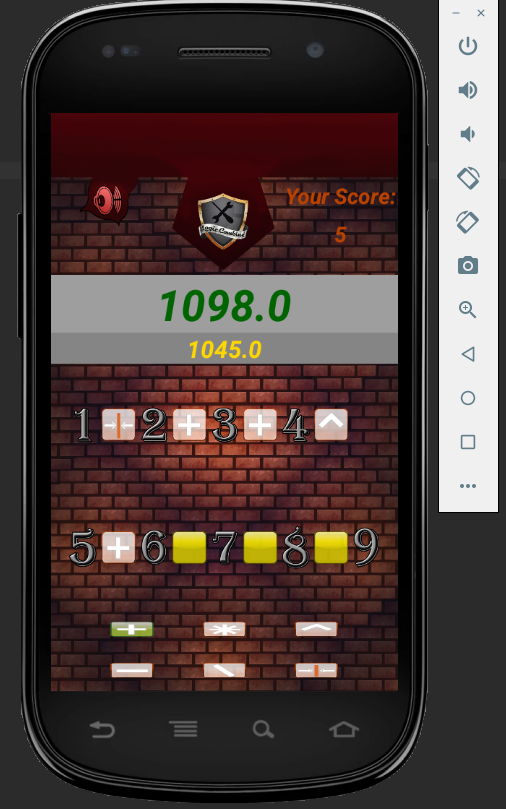
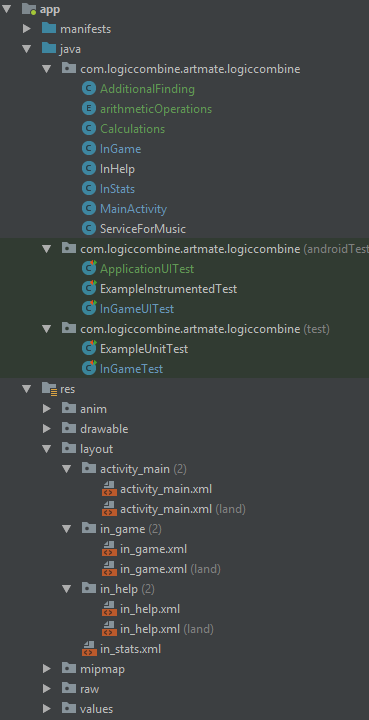
 

Рисунок 21 Конечный результат (part2)

## Структура приложения

Структура приложения приведена ниже в виде скриншота из редактора:



*Рисунок 22 Структура приложения*

# Вывод

Было получено много новой информации, а также приобретён огромный опыт работы с Android Studio, получены навыки в создании и проектировании мобильных приложений. Также были достигнуты цели, поставленные перед началом работы, в результате чего получилась задуманное изначально приложение.

# Список источников

1. Вспомогательные статьи для Android разработчиков

https://developer.android.com/guide

1. Цикл статей, объясняющих основные принципы работы в Android Studio

https://startandroid.ru/ru/

1. Форум с большим количеством ответов на затрудненные вопросы.

https://stackoverflow.com