САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Отчёт по курсовой работе

**«Игра в стиле монополии»**

Дисциплина

«Проектирование мобильных приложений»

выполнили:

Залата А.С

Купцов А.О.

группа: 33531/2

преподаватель:

Алексюк А.О.

Санкт-Петербург

2018

Оглавление

[Список иллюстраций 2](#_Toc533738143)

[Введение 3](#_Toc533738144)

[Цель 3](#_Toc533738145)

[Задачи 3](#_Toc533738146)

[Дополнительная информация 3](#_Toc533738147)

[Описание работы 4](#_Toc533738148)

[Структура проекта 4](#_Toc533738149)

[1. SplashScreenActivity 4](#_Toc533738150)

[2. MainActivity 5](#_Toc533738151)

[3. AmountOfPlayersActivity 5](#_Toc533738152)

[4. GameActivity 6](#_Toc533738153)

[5. WinScreenActivity 8](#_Toc533738154)

[6. Структура проекта 8](#_Toc533738155)

[7. Тесты 9](#_Toc533738156)

[8. Warnings 9](#_Toc533738157)

[9. Исправления 9](#_Toc533738158)

[Выводы 10](#_Toc533738159)

[Приложение 10](#_Toc533738160)

# 

# Список иллюстраций

[Рисунок 1 SplashScreenActivity 4](#_Toc533378473)

[Рисунок 3 MainActivity 5](#_Toc533378473)

[Рисунок 3 AmountOfPlayersActivity 5](#_Toc533378473)

[Рисунок 4 GameActivity 6](#_Toc533378473)

[Рисунок 5 WinScreenActivity 8](#_Toc533378473)

# Введение

## Цель

Основной целью данного курсового проекта является изучение взаимодействий с API операционной системы Android, а также получение практических навыков работы в среде разработки Android Studio.

## Задачи

1. Составление общего концепта проекта
2. Проектирование структуры проекта
3. Описание логики и механики проекта
4. Связь логики и интерфейса пользователя
5. Написание jUnit и UI тестов
6. Исправление недочётов

## Дополнительная информация

* Минимальная версия SDK: 22 (Android 5.0 Lollipop)

# Описание работы

## Структура проекта

В данном разделе будут перечислены и описаны *Activity*, из которых состоит приложение.

### SplashScreenActivity

Данное активити появляется перед главным меню. Оно представляет из себя лого игры.



Рисунок 1 SplashScreenActivity

### MainActivity

Данное активити появляется сразу после SplashScreenActivity. В нем содержится пара кнопок с базовыми действиями, TextView в котором написано название игры и FadingTextView в котором меняется текст. Значения для данного поля берутся из ресурсов массивов строк. Кнопка «Exit» вызывает метод выхода из приложения, а также срабатывает небольшая анимация закрытия приложения. Кнопка «Play» создает новое активити и запускает её. Переходы между активити сопровождаются анимацией свайпа вниз, а возвращение в предыдущее активити сопровождаются анимацией свайпа вверх.

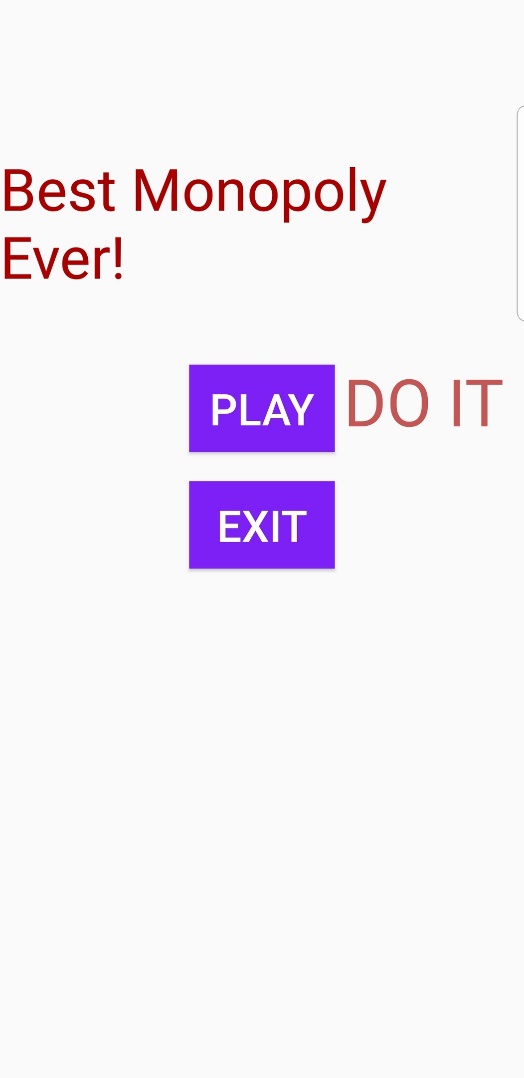


Рисунок 2 MainActivity

### AmountOfPlayersActivity

Содержит поле для ввода имен игроков, текст в данном поле должен быть от 1 до 15 символов, и не совпадать с ранее введёнными именами. Кнопка “Enter” отвечает за ввод имени игрока, так же она становится неактивной, если количество введённых имен меньше минимального количества – 2 или равно максимальному числу игроков (в данный момент 4). Кнопка “Start” отвечает за создание и запуска нового активити (с самой игрой), но она не будет доступна если количество игроков меньше двух. 4 TextView отвечают за отображение имен, при нажатии выбранное имя выделяется и его можно удалить с помощью кнопки “Delete”. Можно выбирать сразу несколько игроков для удаления. При добавлении игрока появляется кнопка, с помощью которой можно назначить данного игрока ботом и в игре он будет совершать действия сам.

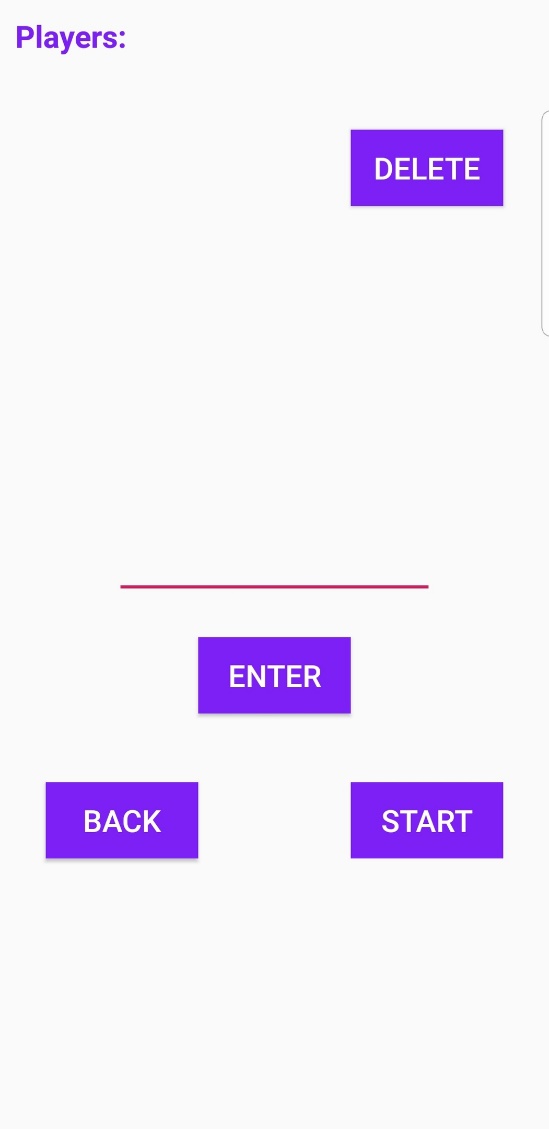
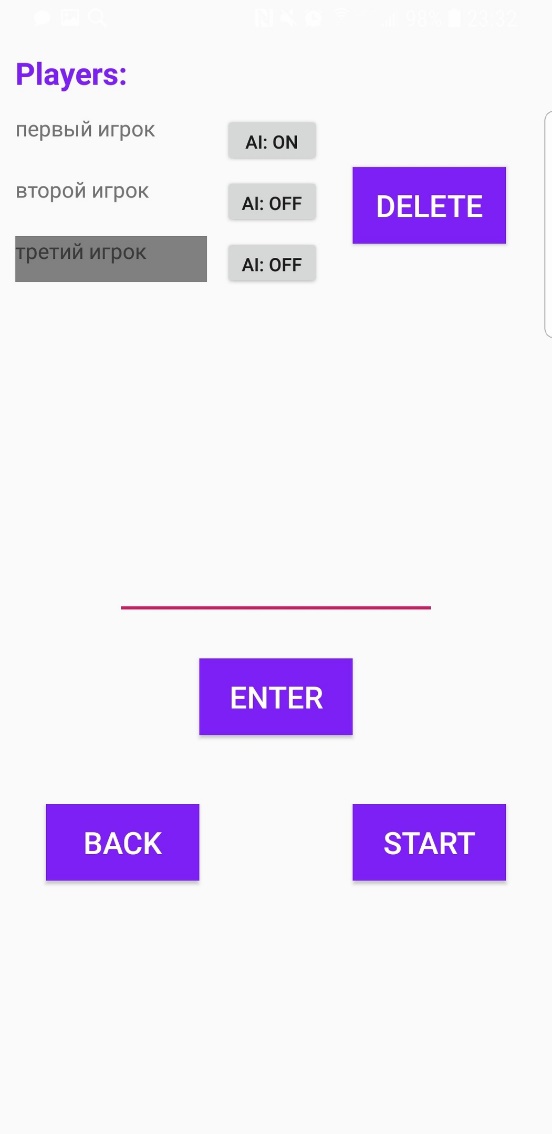
 

Рисунок 3 AmountOfPlayers

### GameActivity

Данное активити использует ConstraintLayout. За основу клетки игрового поля была взята кнопка. Для удобного расположения поля, размер кнопок зависит от ширины экрана. Поле создается программно. Первая клетка поля привязывает к верху и левому краю layout’а, а остальные кнопки поочередно привязываются к предыдущей таким образом, чтобы получилось квадратное поле. Для создания фона в клетках поля используется layer-list, в котором задается цвет фона, картинка данной клетки и картинка рамки. Каждая из картинок содержится в 6 разных экземплярах разного разрешения. При нажатии на кнопку “ThrowDices” происходит вызов метода, который отвечает за передвижение игроков по полю. После этого, если игрок попал на поле, которое он может купить, то появится предложение купить клетку с двумя кнопками “Yes” и “No”. Если игрок нажмет “Yes” то сработает метод, который присвоит ему эту клетку и поменяет цвет фона на цвет этого игрока. Если же игрок нажмет кнопку “No”, то вызовется метод, который сменит игрока. При нажатии на любую клетку выведет информация о ней (Название, стоимость покупки, стоимость продажи, цена сбора). Так же под полем находятся TextView, в который находятся имена игроков и количество их денег. При нажатии на кнопку “Suicide” игрок сдается, все его клетки становятся доступными для покупки, а сам игрок удаляется. Когда количество оставшихся игроков становится равно одному, то игра прекращается и имя последнего игрока передается в новое активити.

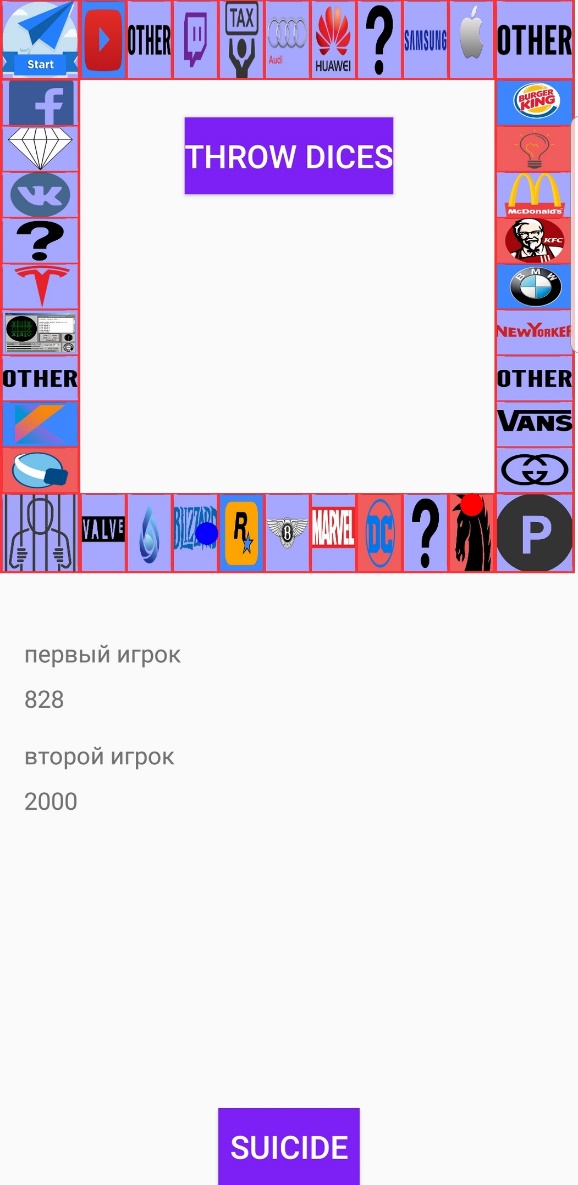


Рисунок 4 GameActivity

### WinScreenActivity

В данное активити получает имя последнего игрока и записывает его в TextView. Кнопка “Menu” отвечает за переход в MainActivity.

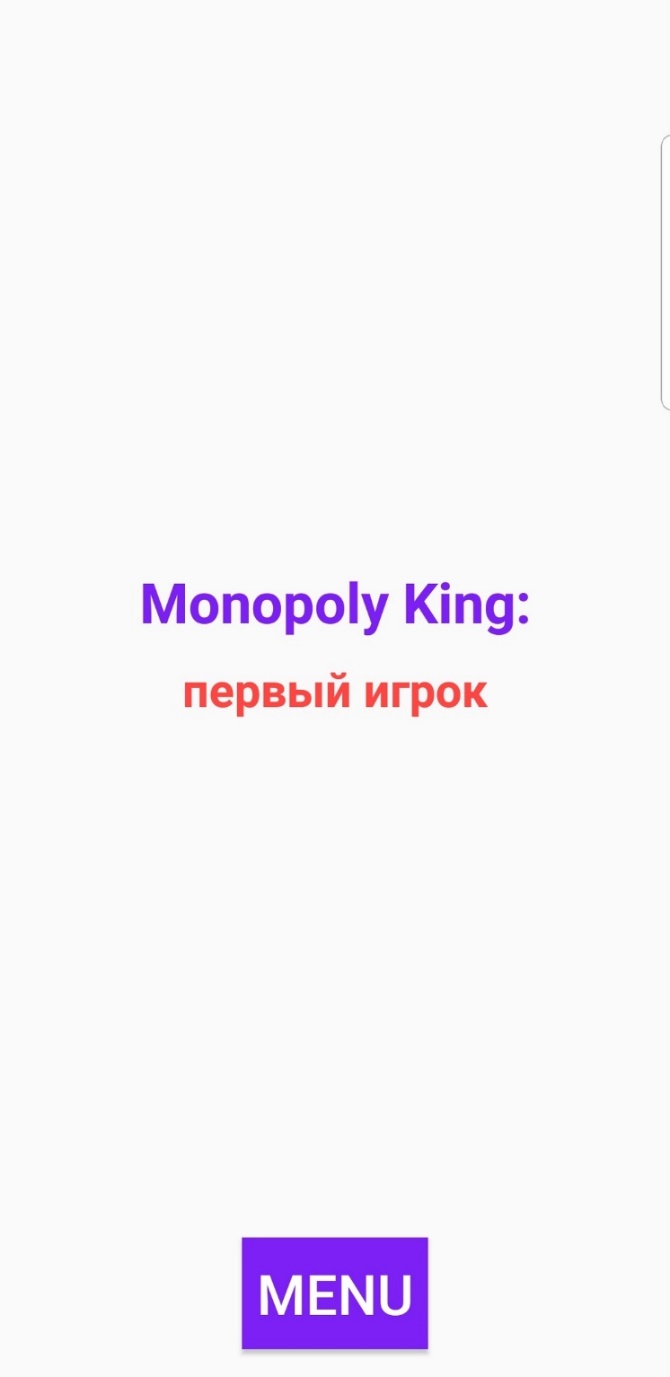


Рисунок 5 WinScreenActivity

### Структура проекта

В директории ../game расположены класс отвечающие за логику приложения. Основной управляющий класс, который также содержит информацию о состоянии приложения (данные) – GameManager. Класс Board отвечает за хранение клеток поля, перемещение игроков по данному полю, а также содержит конструктор, в котором программно создается отображение игровых клеток. Класс GameCell – реализация игровой клетки (состояние, взаимодействие с ней). Также есть множество вспомогательных классов перечислений. Есть 2 объектных (статичных) класса – GameData и ValuesData, в которых заданы начальные значения и константы. Класс GameActivityData наследуется от ViewModel и используется для хранения информации при пересоздании GameActivity.

### Тесты

Были написаны 2 вида тестов. jUnit тесты и UI тесты. jUnit тесты проверяют работу методов связанными с логикой проекта, например перемещение по клеткам, добавление/удаление игроков, различные действия игроков, такие как получение и потеря денег, проверка анализатора клетки и другие. UI тесты проверяют правильность взаимодействия кнопок с пользователем и правильное выполнение функций этих кнопок.

### Warnings

Все предупреждения, касательно кода программ, выявленные анализатором, были исправлены. Оставшиеся предупреждения, по большей части появились из-за того, что картинки находятся в различных разрешениях и некоторые картинки не подлежат предпросмотру. Так же есть предупреждения о различных названиях переменных, но они появились из-за того, что в коде присутствуют имена собственные.

### Исправления

В ходе исправления замечаний в GameActivity было решено добавить ScrollView лишь на половину экрана, чтобы в остальной части можно было расположить информацию о клетках и игроках. Для сохранения при повороте так же был использован ViewModel в связи с замечаниями. Множество констант были перенесены в отдельных класс под названием ValuesData. Так же были исправлены небольшие замечания по тестам. В замечаниях было указано описать использование метода getIdentifier. Данный метод был использован для того, чтобы получать доступ к множеству различных id, у которых небольшие различия в названии, так как с помощью R.id. это сделать нельзя.

# Выводы

В ходе данного курсового проекта были изучены различные взаимодействий с API операционной системы Android, работа с альтернативными ресурсами, обработка пересоздания активити, сохранение данных. Также были получены практические навыки по созданию приложений в среде разработки Android Studio. В ходе данной работы было встречено некоторое количество трудностей. Главная из них это плохая оценка масштаба работы, которая привела к тому, что небольшую часть начальной задумки пришлось упростить. Так же большое количество времени было потрачено на создание поля и нормальное расположение всех объектов в GameActivity. При повороте экранов так же было множество проблем, связанных с большим количеством данных, которые надо было сохранять, а сохранять их можно только в стандартных типах. Работа со всеми вышеперечисленными трудностями позволила только лучше изучить особенности разработки под Android. В результате был получен бесценный опыт проектирования и создания Android приложений.

# Приложение

1. Исходный код проекта:

[https://gitlab.com/Shurikk/monopoly](https://vk.com/away.php?utf=1&to=https%3A%2F%2Fgitlab.com%2FShurikk%2Fmonopoly)

https://github.com/AlexanderKuptsov/AndroidProject