

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ПЕТРА ВЕЛИКОГО

Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Отчет
по курсовой работе «Разработка приложения “Пятнашки”»
Дисциплина
«Проектирование мобильных приложений»

Выполнила:
студентка группы 33531/3
Макарова Ю.С.

Преподаватель:
Кузнецов А.Н.

Санкт-Петербург
2018

Содержание

Введение	3
Цели	3
Задачи	3
Разработка и отладка приложения	3
1. Главное меню	3
2. Рекорды	4
3. Основной экран игры	6
4. Хранение рекордов	10
5. Анализатор кода	10
6. Тестирование	11
Заключение	12
Список использованных источников	12
Приложение 1	12

Введение

Все разрабатываемые приложения условно делятся на два типа: те, которые используются в рабочих целях, например, утилиты, способные контролировать протекания бизнес-процессов или составлять аналитический отчет, и те, которые включают в себя развлекательный софт – игры, программы для воспроизведения видео, аудио материалов, средства коммуникации.

Разработка под платформу Android — это серьёзный, трудозатратный процесс. Перед разработкой программного продукта необходимо понимать цель, задачи и ресурсы, которые требуются для реализации данного продукта. Необходимо принимать взвешенные решения начиная от выбора шаблона проектирования до распределения всех задач по времени, в целях более оптимизированной разработки проекта. Вся разработка происходила с использованием среды Android Studio. Весь код был написан на языке Java.

Цели

Целью данной работы является разработка развлекательного приложения-головоломки под мобильную ОС Android.

Задачи

Для достижения поставленной цели необходимо решить следующие задачи:

1. Разработать экран меню для навигации по разделам приложения
2. Разработать экран с таблицей рекордов, которые может отображать результаты игр и выполнять некоторые действия с ними
3. Обеспечить сохранение результатов игр в базе данных
4. Разработать экран с игровым процессом
5. Провести тестирование

Разработка и отладка приложения

Все разделы приложения было решено сделать в качестве отдельных активностей, чтобы сильно не нагружать код, так как, например, использование фрагментов, с моей точки зрения, немного нагружает логику работы приложения, но для больших проектов бывает необходимо использовать именно их, а данный проект не требует каких-либо сложных взаимодействий.

1. Главное меню

Главное меню состоит из трех Button'ов и одного TextView, содержащего название игры. Кнопка «Играть» отвечает за переход на GameActivity, где происходит игровой процесс. Кнопка «Рекорды» отвечает за переход на RecordsActivity, которая отображает результаты успешных игр. А кнопка «Выход» закрывает игру.



Рисунок 1 Главное меню

Для выполнения действий к каждой кнопке был привязан слушатель `OnClickListener`, который при нажатие на кнопку вызывает метод `onClick()`, в котором выполняются соответствующие действия.

2. Рекорды

Этот раздел содержит `TextView` с заголовком и две кнопки «Меню» и «Очистить», первая возвращает нас в главное меню, а вторая очищает список рекордов, но прежде просит подтвердить свое намерение с помощью диалогового окна.

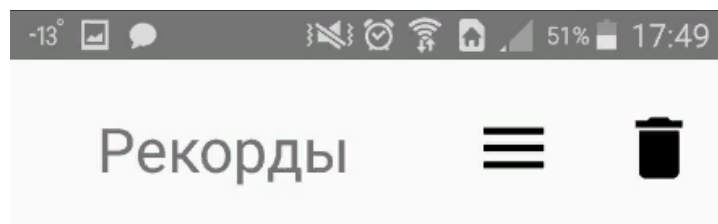


Рисунок 2 Верхнее меню раздела "Рекорды"

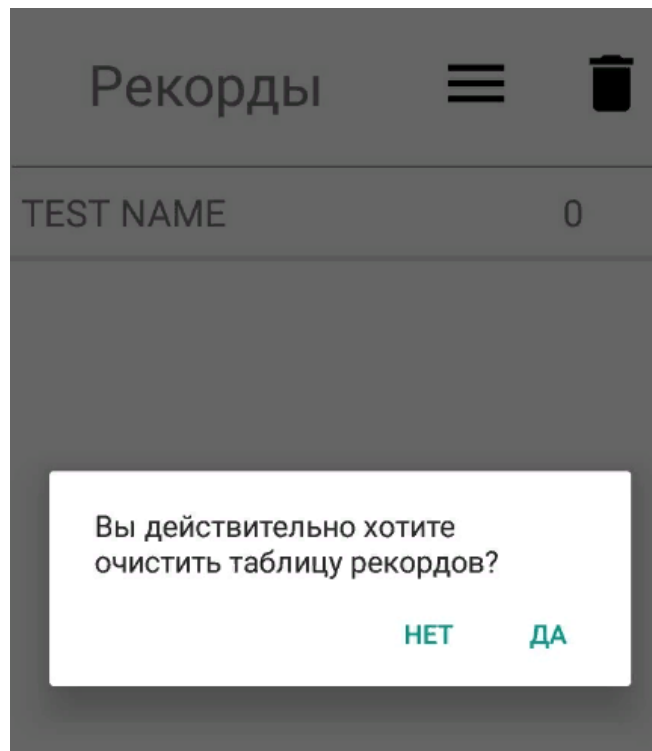


Рисунок 3 Диалог подтверждения очистки рекордов

Для отображения рекордов было решено использовать RecyclerView - представляющий абстракцию списка, который позволяет работать с большим количеством данных и отображать их на экране. Его особенность от своего предшественника ListView в том, что RecyclerView переиспользует графическое представление, а не создает отдельное – для каждого. Таким образом, можно отображать любое количество данных, используя минимум ресурсов.

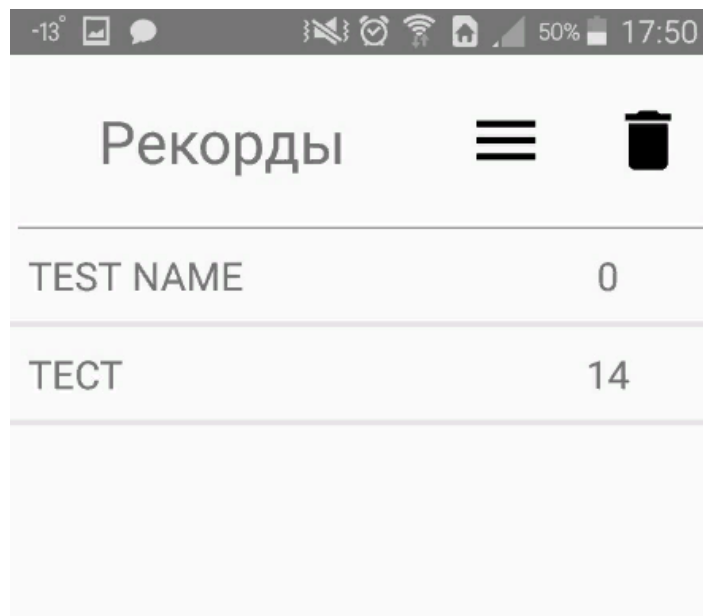


Рисунок 4 Раздел "Рекорды"

Чтобы создать разделите между элементами использовался класс DividerItemDecoration, позволяющий задать Drawable, в качестве разделителя.

Основным контейнерам были заданы веса, чтобы они всегда занимали одинаковое количество пространства независимо от экрана устройства. Это используется не только в разметке этого раздела, но и в других.

3. Основной экран игры

Данным раздел тоже содержит меню с кнопками, а также TextView, который отображает количество сделанных ходов. Но в этот раз в портретной ориентации меню находится над игровым полем, а в горизонтальном виде меню находится сбоку. Для горизонтального вида была создана альтернативная разметка, чтобы не нагружать код лишними условиями.

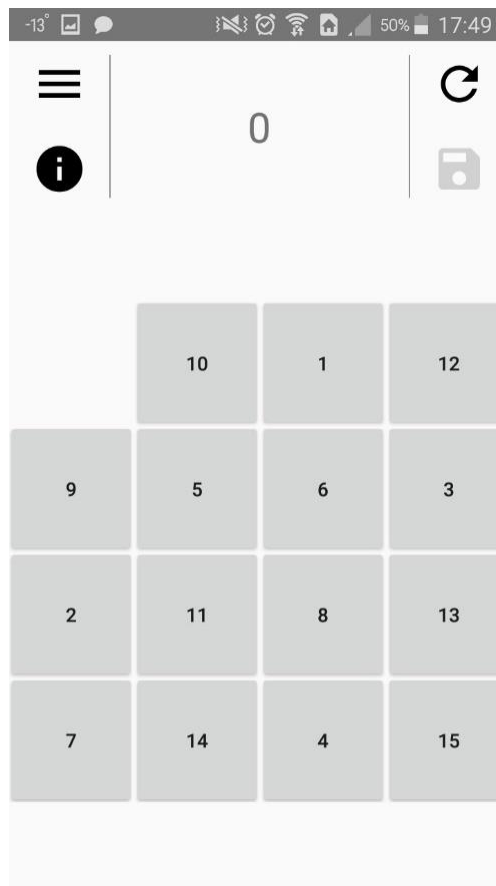


Рисунок 5 Игра: вертикальный вид

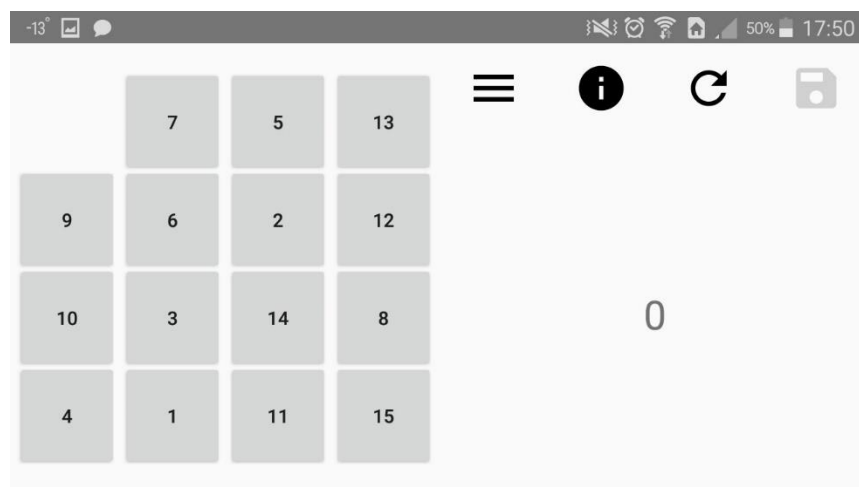


Рисунок 6 Игра: горизонтальный вид

Игровое поле представляет из себя GridView, которое содержит в себе 16 кнопок с уникальной цифрой. Для размещения элементов на GridView использовался собственный адаптер, который наследуется от базового адаптера BaseAdapter. Количество столбцов фиксировано и задано в разметке, их количество равно четырём. Чтобы пятнашки не вылезали за границы экрана, их размеры динамически вычисляются при создании активности.

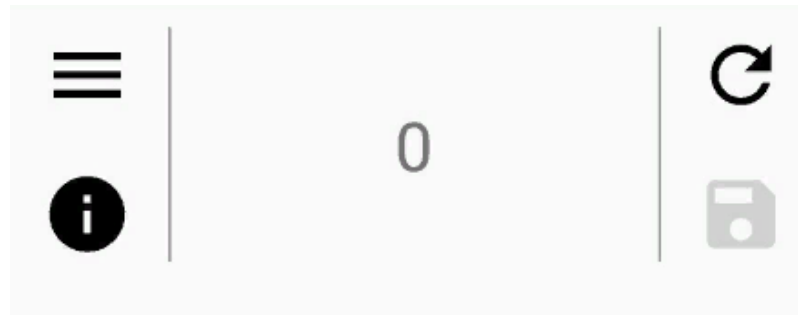


Рисунок 7 Меню со счетчиком ходов

Кнопка «Меню» возвращает нас в меню, точно так же, как и в других разделах. Кнопка «Информация» открывает диалоговое окно с описанием целей игры и небольшой исторической справкой.

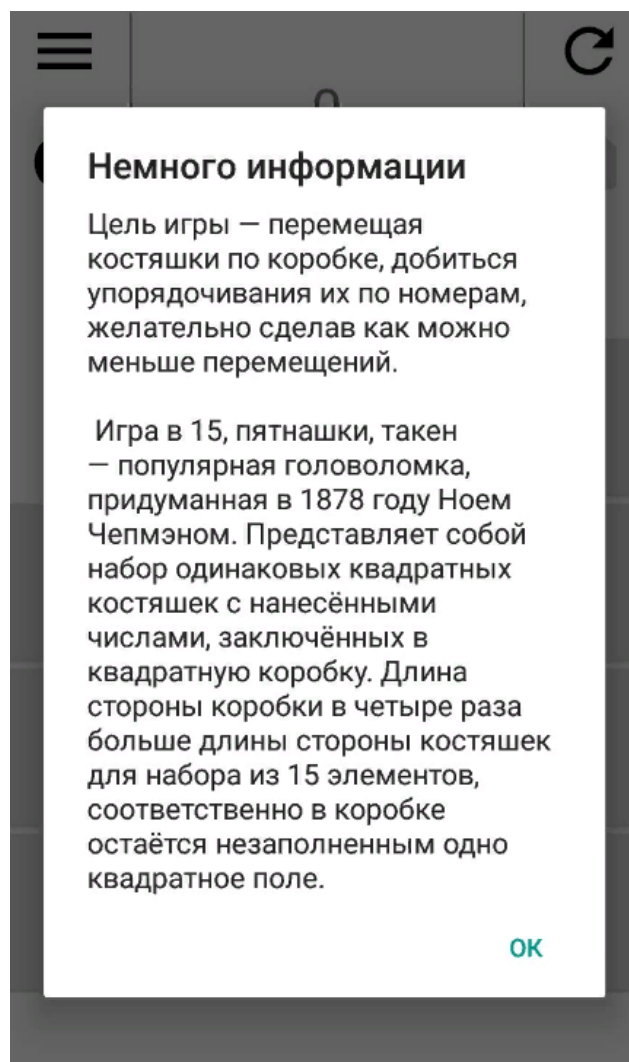


Рисунок 8 Диалог с информацией

Кнопка «Рестарт» генерирует новое поле и начинает игру заново, но прежде выводит диалоговое окно с просьбой подтвердить ваше намерение.

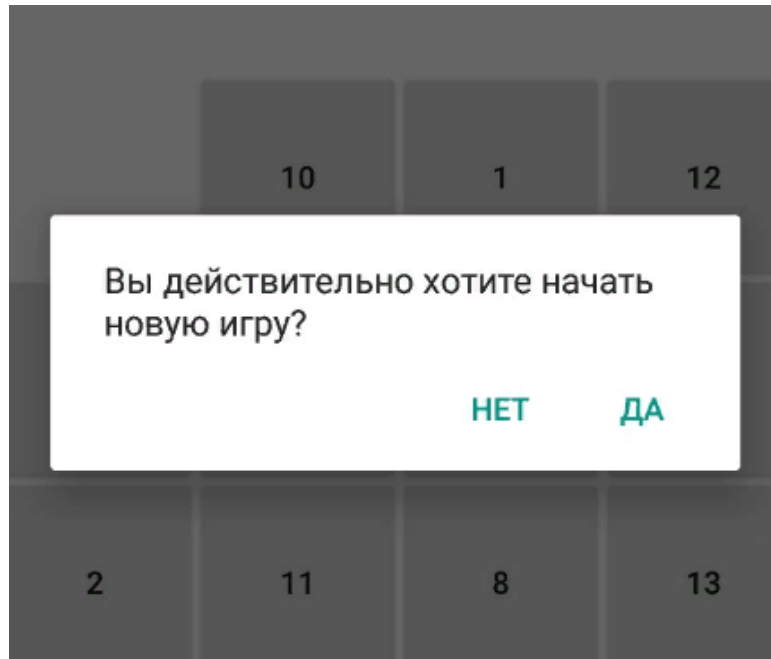


Рисунок 9 Диалог подтверждения начала новой игры

Как известно пятнашки являются решаемыми только в 50% случаев, поэтому для проверки правильности сгенерированного поля использовался следующий алгоритм: пусть квадратик с числом i расположен до (если считать слева направо и сверху вниз) k квадратиков с числами меньшими i . Будем считать $n_i = k$, то есть если после костяшки с i -м числом нет чисел, меньших i , то $k = 0$. Также введем число e — номер ряда пустой клетки (считая с 1). Если сумма

$$N = \sum_{i=1}^{15} n_i + e$$

является нечётной, то решения головоломки не существует. А чтобы не генерировать долго игровое поле, поле случайным образом заполняется цифрами от 1 до 14 по позициям от 1 до 14; потом нулевая позиция заполняется нулем (пустая ячейка), а последняя позиция заполняется 15 и проверяется: является ли пятнашка решаемой, если да, то так и оставляем, а если нет, то согласно правилу, мы можем поменять наш ноль с пятнадцатью и тогда сумма выше станет четной и пятнашка станет решаемой.

Кнопка «Сохранить рекорд» является неактивной пока головоломка не решена, и при нажатии на нее всплывает диалоговое окно с информацией об этом.

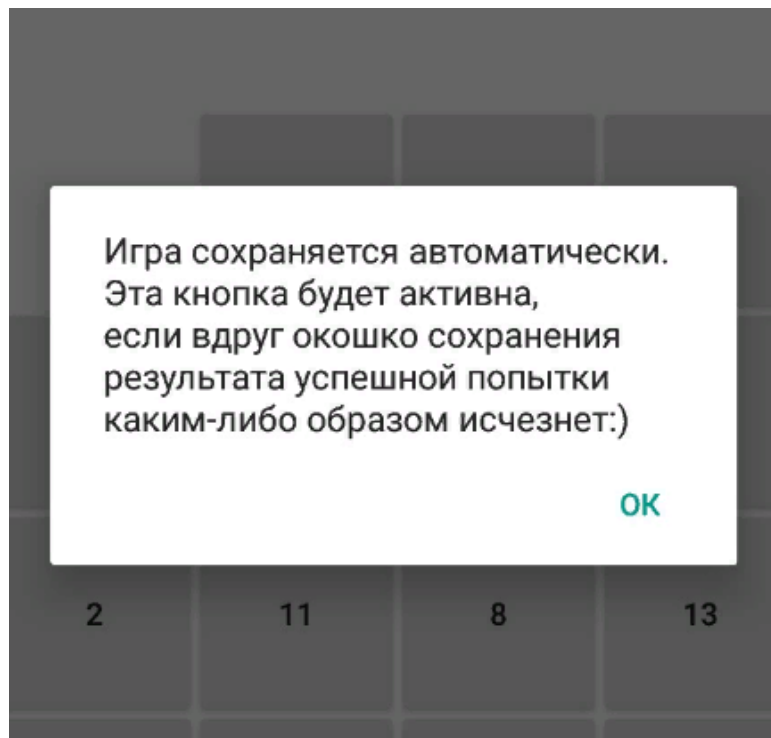


Рисунок 10 Диалоговое окно неактивной кнопки "Сохранить"

Когда головоломка решается, всплывает диалог предлагающий ввести имя для сохранения рекорда. Диалог является неотменяемым, но есть ситуация, при которой он может исчезнуть, не сохранив рекорд, для этого случая есть кнопка «Сохранить рекорд». Она становится активной, когда головоломка становится решенной и при нажатии на нее в этом случае открывается уже диалог сохранения рекорда.

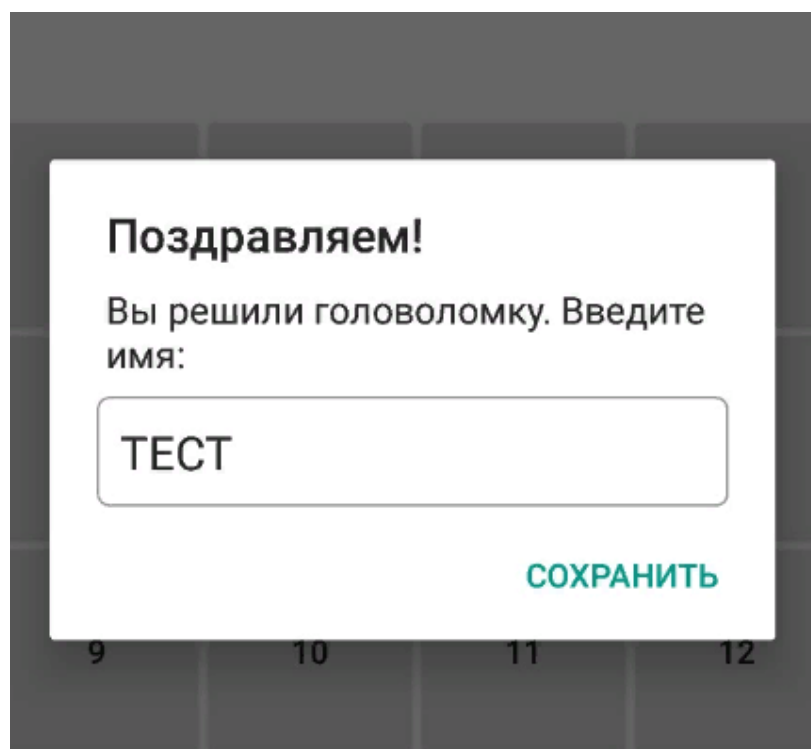


Рисунок 11 Диалог сохранения рекорда

Рекорды хранятся в базе данных. А состояние игры при выходе сохраняется с помощью SharedPreferences.

4. Хранение рекордов

Для работы с базой данных использовался Room — высокоуровневый интерфейс для низкоуровневых привязок SQLite. Был написан класс Record с необходимыми аннотациями.

```
@Entity(tableName = "records")
public class Record {

    @PrimaryKey(autoGenerate = true)
    public long id;

    public String name;

    public int moves;
}
```

Интерфейс RecordDao, содержащий нужные функции для взаимодействия с базой данных.

```
@Dao
public interface RecordDao {

    @Query("SELECT * FROM records ORDER BY moves ASC")
    List<Record> getAll();

    @Insert
    void insert(Record record);

    @Query("DELETE FROM records")
    void deleteAll();
}
```

И сам класс базы данных.

```
@Database(entities = {Record.class}, version = 1)
public abstract class AppDataBase extends RoomDatabase {
    public abstract RecordDao recordDao();
}
```

Такая небольшая связка дает возможность очень легко использовать базу данных для хранения информации.

5. Анализатор кода

По завершению основной работы была запущена функция среды разработки Inspect Code, которая показала не малое количество предупреждений, но все они были не критичными (область видимости переменных, орфографические ошибки в строках, не вынесение строк в ресурсы, неиспользуемые переменные). Некоторые предупреждения такого рода были устранены, оставшиеся предупреждения имеют сознательный характер.

6. Тестирование

Чтобы протестировать логику процесса игры, были написаны юнит тесты. Они тестируют класс PlayField, который хранит данные об игровом процессе и отвечает за действия над этими данными. Тесты были успешно пройдены, что подтвердило правильную работоспособность этого класса.

Для создания UI тестов был использован фреймворк Espresso. Успешно удалось протестировать все активности и почти всю функциональность приложения: открытие всех разделов; работоспособность всех кнопок; сохранение рекордов. Тестирования правильности перемещения пятнашек по игровому полю было проведено вручную. Все тесты были пройдены без проблем.

Заключение

В итоге, было разработано Android приложение «Пятнашки». Во время разработки пришлось столкнуться со множеством проблем, которые замедляли процесс. Но благодаря которым, удалось узнать и изучить, что-то новое и интересное, что в будущем безусловно пригодится. Были изучены основные элементы разработки под мобильную операционную систему Android, а также был изучен абсолютно новый для меня фреймворк Espresso и инструмент для работы с базами данных – Room.

Особенным знанием оказалось сильное удивление, что головоломка «Пятнашки» из всех возможных комбинаций игрового поля имеет только половину решаемых. Данный факт несколько растянул время потраченной на разработку, так как изначально генерация поля была абсолютно случайной и получилась комбинация, которые невозможно решить.

Исходя из общего положения и опираясь на совокупность всех ранее перечисленных и упомянутых факторов, можно утверждать, что цели, поставленные в начале работы, были достигнуты.

Список использованных источников

1. Android Developers. <https://developer.android.com/>
[Электронный ресурс]
2. Освой Android играючи. <http://developer.alexanderklimov.ru/android/>
[Электронный ресурс]
3. Основы разработки android-приложений. <https://youtu.be/ac8y518CIng>
[Видео ресурс]
4. Room. Основы. <https://startandroid.ru/ru/courses/dagger-2/27-course/architecture-components/529-urok-5-room-osnovy.html>
[Электронный ресурс]
5. Room. Тестирование. <https://startandroid.ru/ru/courses/dagger-2/27-course/architecture-components/541-urok-13-room-testirovanie.html>
[Электронный ресурс]
6. GridView и его атрибуты. <https://startandroid.ru/ru/uroki/vse-uroki-spiskom/116-urok-57-gridview-i-ego-atributy.html>
[Электронный ресурс]
7. RecyclerView. <https://metanit.com/java/android/5.11.php>
[Электронный ресурс]

Приложение 1

Исходный код доступен по следующей ссылке:

<https://github.com/JulyMak/AndroidProject>