Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

**Отчет по курсовой работе**

**Дисциплина**: Проектирование мобильных приложений

**Тема**: Разработка приложения Insect Killer

Выполнила студентка гр. 33531/3 _____ В.С. Янковская
(подпись)

Преподаватель _____ А.Н. Кузнецов
(подпись)

“___”_____ 2018 г.

Санкт-Петербург
2018

СОДЕРЖАНИЕ

**Введение**

Целью работы является разработка Android-приложения «Insect Killer». В качестве среды разработки была выбрана Android Studio, язык программирования — Kotlin.
Перед нами стоят следующие задачи:

- Необходимо реализовать:
  - Начальный экран с возможностью перехода к новой игре, экрану с настройками и экрану со списком лучших результатов;
  - Основной экран с игровым полем;
  - Экран с сообщением об окончании игры и возможностью сохранить результат игры;
  - Экран настроек, позволяющий настраивать приложение;
  - Экран со списком лучших результатов, а также базу данных для хранения этих результатов;
- Провести тестирование разработанного приложения.

**1. Начальный экран**

На начальном экране в качестве макета используется ConstraintLayout. Здесь размещаются четыре компонента: один ImageView и три ImageButton’а.
При нажатии на центральную кнопку происходит переход на экран с новой игрой, на левую кнопку — переход на экран с настройками, на правую кнопку — переход на экран со списком лучших результатов.


Рисунок 1. Начальный экран в вертикальной и горизонтальной ориентациях

## 2. Экран настроек

Для экрана настроек используется специальное Activity для работы с настройками — PreferenceActivity.
Экран настроек (PreferenceScreen) содержит два компонента:

- CheckBoxPreference, с помощью которого можно включить/выключить звуки в игре;
- ListPreference, с помощью которого можно выбрать один из трех уровней сложности игры.

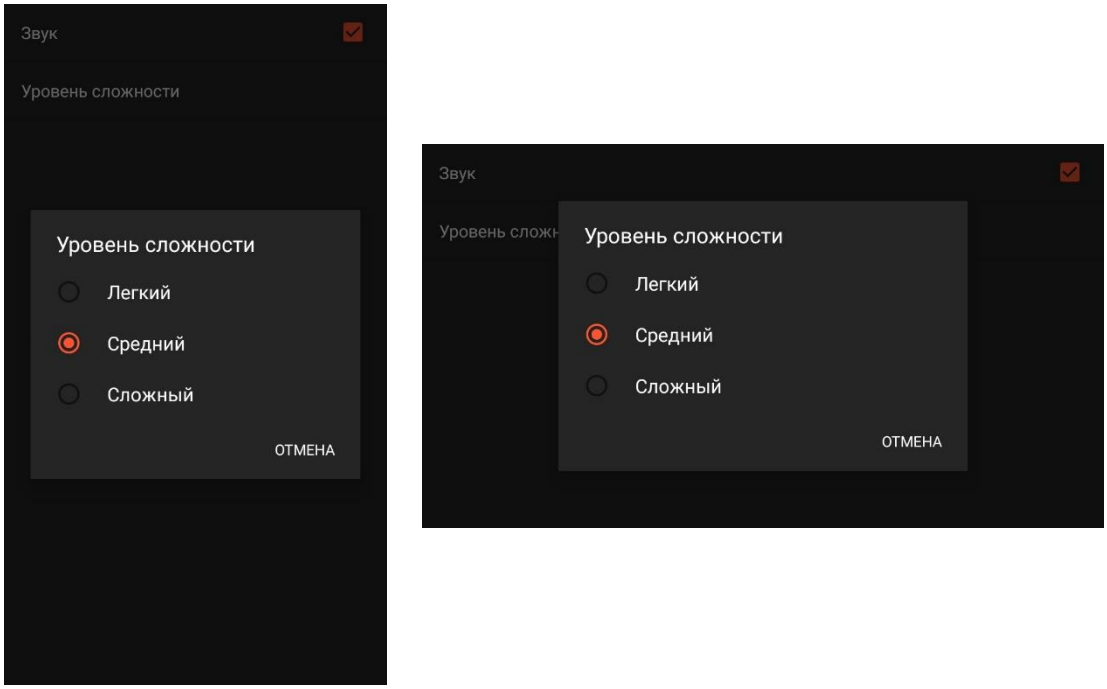


Рисунок 2. Экран настроек в вертикальной и горизонтальной ориентациях

## 3. Игровой экран

Суть игры заключается в следующем: необходимо защитить пиццу, находящуюся в центре экрана, от жуков, которые пытаются эту пиццу съесть. Для этого нужно давить жуков, получая за это очки. Также нужно пытаться не раздавить хороших жуков — божьих коровок, так как за это отнимаются очки. Когда пицца полностью съедена жуками, игра заканчивается. Цель игры — набрать как можно больше очков.
Игровой экран состоит из двух основных областей: собственно игрового поля, на котором лежит пицца и по которому бегают насекомые, и верхней панели, которая содержит информацию о состоянии игры (индикатор текущего «здоровья» пиццы и количество набранных очков). Также на верхней панели находится кнопка, с помощью которой можно поставить игру на паузу.

Пицца представляет из себя компонент ImageView, ресурсом которого является drawable ресурс типа level-list. Таким образом, по мере убывания «здоровья» пиццы изменяется её вид — понемногу с пиццы исчезает начинка. Для реализации насекомых был создан класс Insect, который является наследником класса ImageView. При создании экземпляра класса Insect нужно задать следующие параметры: начальные и конечные координаты насекомого и его тип (жук или божья коровка). Анимация передвижения насекомого от начальной до конечной точки реализуется с помощью объекта класса ObjectAnimator.

В классе GameFieldActivity определен метод addInsect(), который добавляет экземпляр класса Insect на gameFieldLayout и устанавливает ему OnTouchListener. Также определен метод generateInsects() с помощью которого создается определенное количество насекомых определенного вида. Этот метод по некоторым формулам генерирует начальные и конечные координаты для насекомого, создает экземпляр класса Insect и вызывает применительно к нему метод addInsect(). Метод generateInsects() вызывается с определенной периодичностью, для этого используется объект класса Timer. В зависимости от выбранного уровня сложности изменяется скорость движения насекомых, период генерации насекомых, а также количество божьих коровок.

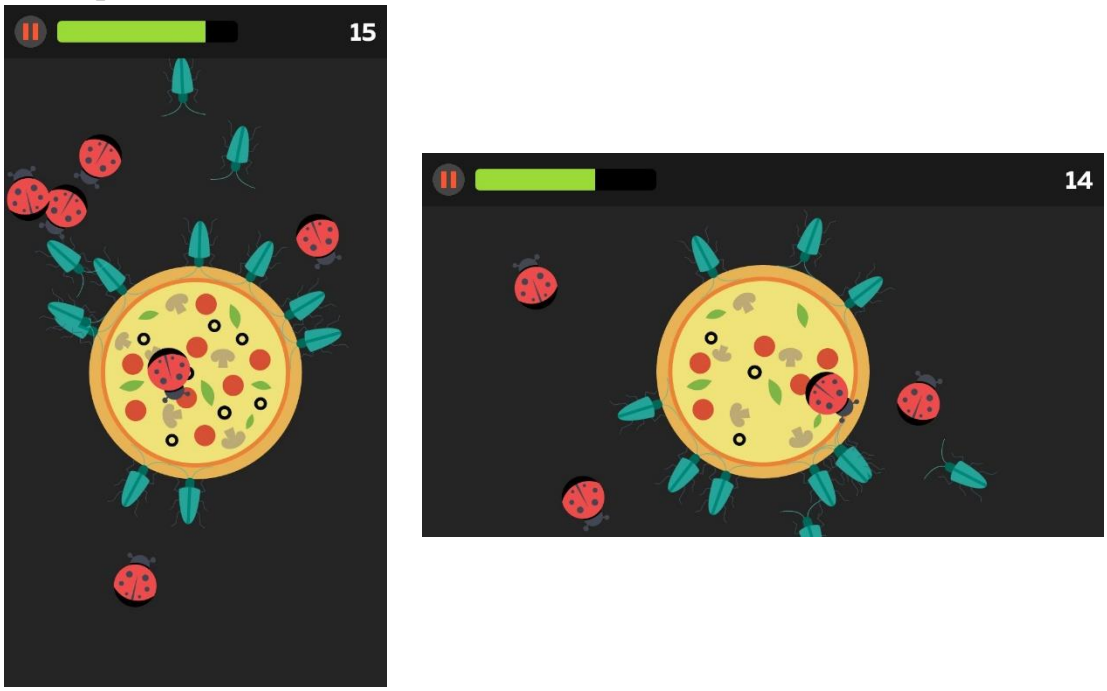


Рисунок 3. Игровой экран в вертикальной и горизонтальной ориентациях

## 4. Экран завершения игры

Экран завершения игры содержит три компонента:

- TextView с текстом «Ваш счет» и итоговым счетом игры;

- EditText, в которое игроку необходимо ввести свое имя;
- кнопку «Сохранить результат», при нажатии на которую результат игры сохраняется в базу данных.


Рисунок 4. Экран завершения игры в вертикальной и горизонтальной ориентациях

## 5. База данных с результатами игр

Для хранения результатов игр используется встроенная СУБД SQLite. База данных содержит три таблицы — для хранения результатов игр легкого, среднего и сложного уровней сложности. Таблицы имеют простую структуру:

| Id | Name | Score |
|---|---|---|

Для работы с БД был создан класс DBHelper, в котором были переопределены методы onCreate() и onUpgrade(). Также был создан файл DBMethods.kt, в котором определены две функции для добавления записей в БД: addScore() и addDefaultScores(). Изначально БД с помощью функции addDefaultScores() инициализируется случайными результатами (при этом имена игроков случайным образом выбираются из некоторого списка имен, а счет игр генерируется случайно в заданном диапазоне). Новые результаты добавляются в БД при нажатии кнопки «Сохранить результат» на экране завершения игры, для этого вызывается метод addScore().

## 6. Экран со списком лучших результатов

Экран со списком лучших результатов состоит из трех фрагментов — по фрагменту для показа результатов игр каждого из трех уровней сложности. Для управления фрагментами используются компоненты TabLayout (вкладки)

и ViewPager (для возможности перелистывания вкладок влево/вправо). При создании активити берет данные из БД, описанной выше, сортирует их по убыванию счета, и показывает по десять лучших результатов на каждый уровень сложности.

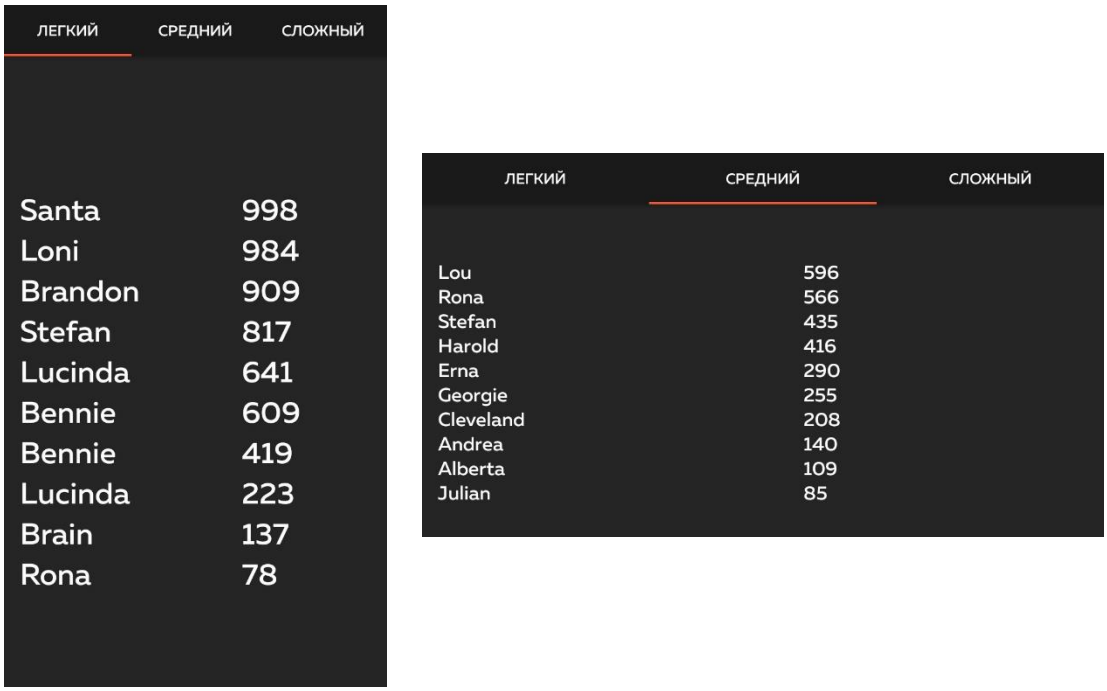


Рисунок 5. Экран со списком лучших результатов в вертикальной и горизонтальной ориентациях

## 7. Тестирование

Для создания UI тестов был использован фреймворк Espresso. Были протестированы появление на экране TextView, ImageView, ImageButton, нажатия на ImageButton. Основными тестами, проверяющими работу приложения, были ручные тесты.

Пример проверки появления на экране ImageView с заданным id:

```
onView(withId(R.id.logo)).check(matches(isDisplayed()))
```

Пример проверки нажатия на ImageButton:

```
onView(withId(R.id.startButton)).perform(click())
```

## Заключение

В результате проделанной работы было разработано Android-приложение с игрой «Insect Killer». Во время разработки мы изучили структуру Android-приложения и жизненный цикл Activity, обработку событий, таких как onClick() и onTouch(). Научились использовать ресурсы различных типов, а также альтернативные ресурсы для смены языка интерфейса. Также мы познакомились с работой с СУБД SQLite.

Большая часть поставленных задач были выполнены, таким образом цель работы достигнута.

**Приложение 1**

Ссылка на исходный код проекта: https://github.com/Lestwald/AndroidProject