

Разработка мобильной игры ReactionGame

Андрienко Иван Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной работе представлена разработка мобильной игры «ReactionGame» с использованием Android Studio и языка программирования Java. В статье описаны шаги по созданию проекта, дизайну интерфейса, логике игры и отладке. Реализованные функциональности включают в себя генерацию случайных цветов, отсчет времени, подсчет очков и создание простого игрового цикла.

Ключевые слова: Android, Java, Android Studio, Мобильная игра.

Development of the Reaction Game mobile game

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This paper presents the development of the mobile game "ReactionGame" using Android Studio and the Java programming language. The article describes the steps for creating a project, interface design, game logic, and debugging. Implemented functionality includes random color generation, timing, scoring, and creating a simple game loop.

Keywords: Android, Java, Android Studio, Mobile game.

1 Введение

1.1 Актуальность

Разработка мобильных игр является важным и актуальным направлением в области программирования, особенно в контексте постоянного роста мобильных устройств и повышенного интереса пользователей к развлекательным приложениям. Создание легких и захватывающих игр, таких как «ReactionGame», не только способствует развлечению пользователя, но и предоставляет разработчикам возможность освоить основы разработки игр для мобильных устройств. Разработка игры «ReactionGame» предоставляет пример простой, но захватывающей мини-игры, подходящей для начинающих разработчиков, а также обеспечивает понимание основных этапов создания мобильного приложения и игрового цикла.

1.2 Обзор исследований

В своей работе Т.Ю. Батюшкина, В.С. Новгородова представили разработку игры «Угадай корейскую знаменитость» на платформе Android Studio. Проведен сравнительный анализ характеристик сред разработки, а также анализ аналогов мобильных игр «Угадай корейскую знаменитость». Обосновывается актуальность исследования, указываются цели и практическая значимость. В практической части представлены скриншоты разработки игры с описанием каждого уровня [1]. А.Ю. Звайгзне рассмотрел процесс разработки Android игры - Catch Star, сутью которой является в угадывании актера по фото. имена актеров и их фотографии берутся с популярного сайта Internet Movie Database, со страницы топа 100 актеров на 2014 год [2]. В своей работе А.М. Саввин описал процесс разработки мобильной игры на Android Studio. В статье рассматривается анализ и обзор языков и средств программирования, аналоги мобильных игр. Авторами выбрана тема головоломок с якутским языком с целью её продвижения и развития. Обосновано предположение о необходимости разработки такой игры. Новизной исследования указана попытка создания мобильной игры, которая бы позволила повысить формирование национального самосознания, патриотизма молодежи Якутии [3]. В своей С.В. Гуляева, А.А. Бурнашева исследовали возможности мобильных технологий и эффективность их использования в процессе развития логики детей дошкольного возраста [4]. Винокурова Л.Е описала процесс разработки мобильного приложения для ознакомления и использования настольных игр для развития мелкой моторики рук и логики детей на Android Studio. В статье рассматриваются достоинства и недостатки языков и средств программирования, приводится их сравнительная характеристика, аналоги мобильных приложений. Выбрана якутская тема с целью её продвижения и развития. Обосновано предположение о необходимости разработки такого приложения [5].

1.3 Цель исследования

Цель исследования – разработать и представить процесс создания мобильной игры «ReactionGame» для платформы Android, используя Android Studio и язык программирования Java.

2 Материалы и методы

Процесс создания мобильной игры проделан в среде Android Studio, с использованием языка программирования Java.

3 Результаты и обсуждения

Перед созданием приложения необходимо подготовить папку «drawable». В ней будет содержаться ресурс, используемый в приложении. Создаем в папке drawable resource file. Он представляет собой XML-файл, определяющий форму и цвет объекта, который будет использоваться в качестве фона для View в макете. В данном случае, создается круг, и его цвет

установлен в черный. Именно по этому кругу должен будет успеть нажать пользователь чтобы заработать очки (рис. 1).

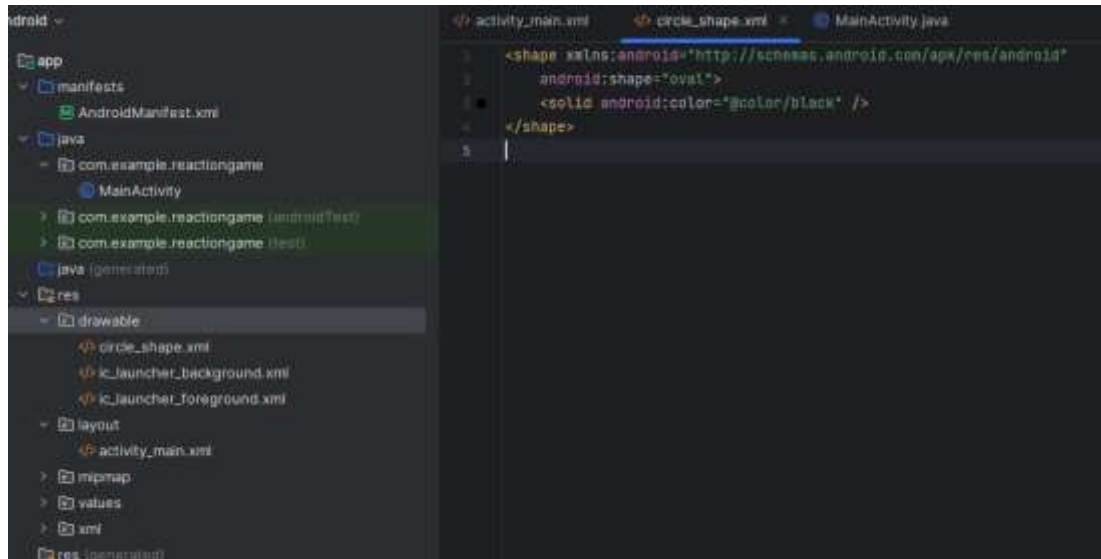


Рисунок 1 – Создание circle_shape.xml

Далее создадим макет интерфейса. Добавим основные компоненты приложения в файл activity_main.xml. Всего в макете будет три элемента: View – базовый строительный блок для создания графических элементов, Button – элемент для запуска начала игры и TextView – используется для отображения текста, в данном случае счет игрока. Настроим параметры элементов, такие как размеры, расположение и внешний вид в соответствии с дизайном (рис. 2).

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/btnStartGame"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/colorCircle"
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="140dp"
        android:layout_marginBottom="53dp"
        android:text="Начать игру" />
    <View
        android:id="@+id/colorCircle"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_centerInParent="true"
        android:background="@drawable/circle_shape" />
    <TextView
        android:id="@+id/scoreText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/colorCircle"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp"
        android:text="Счет: 0" />
</RelativeLayout>
```

Рисунок 2 – Создание главного макета

Пропишем логику приложения. В главном классе объявим переменные необходимые для работы приложения. Создадим переменную «score» со значением ноль. Именно эта переменная будет выводиться и меняться при игре в качестве заработанных очков (рис. 3).

```
1 package com.example.reactiongame;
2
3 > import ...
13
14 <> public class MainActivity extends AppCompatActivity {
15
16     3 usages
17     private Button btnStartGame;
18     4 usages
19     private View colorCircle;
20     2 usages
21     private TextView scoreText;
22
23     4 usages
24     private Handler handler;
25     3 usages
26     private Runnable colorChangeRunnable;
27     4 usages
28     private boolean isGameRunning = false;
29     3 usages
30     private int score = 0;
```

Рисунок 3 – Создание переменных

Создадим метод `onCreate`. В этом методе происходит инициализация элементов пользовательского интерфейса, установка обработчиков событий для кнопки и цветного круга, и запуск игры при нажатии на кнопку "Начать игру". Вначале вызывается метод, который выполняет базовую инициализацию активности. Затем устанавливается макет, определенный в файле `activity_main.xml`. Далее инициализируются элементы интерфейса – кнопка, изображение круга и счет, с использованием метода `findViewById`. Для кнопки "Начать игру" и цветного круга устанавливаются обработчики событий `OnClickListener`. При нажатии на кнопку вызывается метод `startGame()`, который запускает игру. При нажатии на цветной круг, если игра запущена (`isGameRunning` равно `true`), вызываются методы `increaseScore()` для увеличения счета и `changeCircleColor()` для изменения цвета круга.

```
24
25     @Override
26     protected void onCreate(Bundle savedInstanceState) {
27         super.onCreate(savedInstanceState);
28         setContentView(R.layout.activity_main);
29
30         btnStartGame = findViewById(R.id.btnStartGame);
31         colorCircle = findViewById(R.id.colorCircle);
32         scoreText = findViewById(R.id.scoreText);
33
34         handler = new Handler();
35
36         btnStartGame.setOnClickListener(new OnClickListener() {
37             @Override
38             public void onClick(View v) {
39                 startGame();
40             }
41         });
42
43         colorCircle.setOnClickListener(new OnClickListener() {
44             @Override
45             public void onClick(View v) {
46                 if (isGameRunning) {
47                     increaseScore();
48                     changeCircleColor();
49                 }
50             }
51         });
52     }
53 }
```

Рисунок 4 – Создание метода «onCreate»

Для того чтобы игра начиналась, когда пользователь готов, пропишем метод `startGame`. Он инициализирует начало игры, устанавливая флаг `isGameRunning` в `true`, обнуляя счет, обновляя отображение счета на экране, изменяя цвет круга и скрывая кнопку "Начать игру". Метод `changeCircleColor` управляет циклическим изменением цвета круга в течение игры. Сначала он удаляет предыдущий коллбэк (если существует) для предотвращения конфликтов. Затем создается объект `Runnable`, представляющий задачу изменения цвета. В этом объекте запускается анимация: круг становится видимым, затем через секунду снова становится невидимым. Далее, рекурсивно вызывается сам метод `changeCircleColor` для повторения процесса. Время задержки перед появлением круга устанавливается в случайное значение от 2 до 5 секунд с использованием `Random`. Это создает эффект случайного и непредсказуемого появления круга, что является ключевой частью игровой динамики (рис. 5).

```
61     1 usage
62     private void startGame() {
63         isGameRunning = true;
64         score = 0;
65         updateScoreText();
66         changeCircleColor();
67         btnStartGame.setVisibility(View.INVISIBLE);
68     }
69
70     3 usages
71     private void changeCircleColor() {
72         handler.removeCallbacks(colorChangeRunnable);
73
74         colorChangeRunnable = new Runnable() {
75             @Override
76             public void run() {
77                 isGameRunning = true;
78                 colorCircle.setVisibility(View.VISIBLE);
79                 handler.postDelayed(new Runnable() {
80                     @Override
81                     public void run() {
82                         isGameRunning = true;
83                         colorCircle.setVisibility(View.INVISIBLE);
84                         changeCircleColor();
85                     }
86                 }, delayMillis: 500);
87             }
88         };
89
90         int delay = new Random().nextInt(bound: 3000) + 2000;
91         handler.postDelayed(colorChangeRunnable, delay);
92     }
```

Рисунок 5 – Создание методов startGame и changeCircleColor

Сделаем счет динамическим. Увеличиваем значение переменной score на единицу, представляющую текущий счет в игре. После этого вызывается метод updateScoreText(), который обновляет отображение счета в текстовом поле (TextView) с идентификатором scoreText. Новое значение счета присваивается текстовому полю, отображая текущий счет игры в пользовательском интерфейсе (рис. 6).

```
92     1 usage
93     private void increaseScore() {
94         score++;
95         updateScoreText();
96     }
97
98     2 usages
99     private void updateScoreText() { scoreText.setText("Счет: " + score); }
100
101 }
```

Рисунок 6 – Создание динамического счета

Тестируем игру (рис. 7, 8).

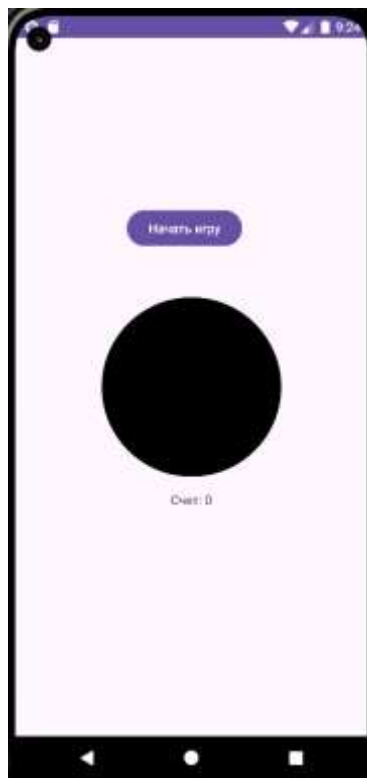


Рисунок 7 – Начальный экран



Рисунок 8 – Пропадание круга при нажатии

Выводы

В данной статье был представлен процесс разработки мобильной игры «ReactionGame» для платформы Android, используя Android Studio и язык программирования Java. В результате исследования была успешно создана

увлекательная мини-игра, предоставляющая пользователю возможность проверить свою реакцию на появление цветного круга в случайный момент времени. Разработка приложения включала в себя создание графического интерфейса, логики игры, а также использование таймеров для управления временем отображения круга. Полученный результат демонстрирует базовые этапы разработки мобильных приложений и может служить стартовой точкой для новичков в области мобильной разработки.

Библиографический список

1. Батюшкина Т.Ю., Новгородова В.С. Разработка игры "угадай корейскую знаменитость" на платформе Android Studio. // DIGITAL EDU. Цифровые компетенции в образовании. Сборник материалов Всероссийского научного форума с международным участием. Киров, 2023. С. 489-492.
2. Звайгзне А.Ю. Разработка игры catch star в Android Studio. // Постулат. 2023. № 1 (87).
3. Саввин А.М. Разработка мобильной игры «якутские кроссворды». // Современная экономика и право: опыт теоретического и эмпирического анализа. Сборник статей VI Международной научно-практической конференции. Петрозаводск, 2023. С. 298-302.
4. Гуляева С.В., Бурнашева А.А. Разработка развивающей игры на android studio. // Молодежная наука как фактор и ресурс инновационного развития. сборник статей II Международной научно-практической конференции. Петрозаводск, 2020. С. 120-124.
5. Винокурова Л.Е. Разработка мобильного приложения «якутские настольные игры». // Современная экономика и право: опыт теоретического и эмпирического анализа. Сборник статей VI Международной научно-практической конференции. Петрозаводск, 2023. С. 276-287.