

Интеграция JavaScript и Java в Android-приложении

Андриенко Иван Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассматривается практический подход к интеграции веб-игры на JavaScript с нативным Android-приложением на Java. Описывается механизм передачи данных между WebView и нативной частью приложения для реализации системы сохранения рекордов. Приводятся конкретные примеры кода, демонстрирующие использование JavascriptInterface для двустороннего взаимодействия, а также организацию хранения данных через SharedPreferences. Результаты показывают эффективность предложенного решения для гибридных мобильных приложений.

Ключевые слова: Android, WebView, JavaScript, Java, гибридные приложения, SharedPreferences, игровые рекорды

Integration of JavaScript and Java in an Android application

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

The article discusses a practical approach to integrating a JavaScript web game with a native Java Android application. The mechanism of data transfer between WebView and the native part of the application for the implementation of a record-keeping system is described. Specific code examples are provided that demonstrate the use of JavascriptInterface for two-way interaction, as well as the organization of data storage through SharedPreferences. The results show the effectiveness of the proposed solution for hybrid mobile applications.

Keywords: Android, WebView, JavaScript, Java, hybrid applications, SharedPreferences, game records

1 Введение

1.1 Актуальность

Разработка гибридных приложений, сочетающих веб-технологии и нативные возможности платформы, остается востребованной задачей в мобильной разработке. Особый интерес представляет передача данных между JavaScript и Java при создании игровых приложений, где важно сохранять прогресс пользователя. Предлагаемое решение упрощает процесс интеграции

javascript-игр в Android-приложения без необходимости полного переписывания кода на нативные технологии.

1.2 Обзор исследований

В своей работе М.Д. Иванов предложил методы оптимизации архитектур мобильных приложений на базе операционной системы android [1]. А.Н. Казакова рассмотрела множество способов хранения данных, каждый из которых имеет свои преимущества и недостатки [2]. В своей работе В.В. Осокин, В.А. Бендик провели реализацию CRM-системы для платформы Android. Система позволяет управлять клиентами, контактными лицами и контактами. В статье описан интерфейс данной системы и его реализация, а также реализация синхронизации данных с серверной частью [3]. В своей работе Ю.В. Мурашко рассмотрел методы хранения данных в операционной системе Android. Также рассматриваются особенности хранения данных во внутреннем и внешнем хранилищах. Выделяются достоинства и недостатки каждого из хранилищ [4]. В.С Дейч в своей работе описал библиотеки, которые облегчат разработку мобильного приложения под ОС Android, предоставляя набор общих функциональных возможностей. Использование библиотек позволяет сэкономить время и силы при создании приложений [5].

1.3 Цель исследования

Цель исследования – разработка и апробация метода передачи игровых результатов из JavaScript-кода в нативную часть Android-приложения с последующим сохранением в постоянное хранилище.

2 Материалы и методы

В качестве основного инструмента разработки использовалась Android Studio с языком программирования Java. Для веб-части применялся JavaScript, встроенный в приложение через WebView.

3 Результаты и обсуждения

Гибридные Android-приложения часто сочетают нативный код Java/Kotlin и веб-технологии JavaScript. В данном случае игра реализована на JavaScript и запускается внутри WebView, а интерфейс рекордов – нативный, написанный на Java. Проблема в том, что JavaScript не может напрямую работать с Android-API для сохранения данных, а Java не имеет доступа к переменным игры. Решение — JavascriptInterface, специальный мост между слоями. В коде создается Java-класс с методами, доступными из JavaScript.

Для организации взаимодействия между JavaScript-кодом игры и нативной Java-частью приложения необходимо выполнить несколько ключевых шагов. Первым этапом является настройка WebView для работы с JavaScript. В методе onCreate активности создается экземпляр WebView, который будет использоваться как контейнер для веб-контента. Важно отметить, что перед загрузкой HTML-страницы необходимо явно разрешить выполнение кода через вызов метода setJavaScriptEnabled. Создание класса-

моста. Этот класс должен содержать методы, которые будут доступны для вызова из JavaScript-кода. В данном случае создается внутренний класс, помеченный аннотацией `@JavascriptInterface`. Именно этот класс будет обрабатывать передачу данных о рекордах из игры в нативную часть приложения (рис. 1).

```
9  ▶ </> public class GameActivity extends AppCompatActivity {
10
11      private WebView webView; 5 usages
12
13      @Override
14      protected void onCreate(Bundle savedInstanceState) {
15          super.onCreate(savedInstanceState);
16
17          webView = new WebView(context: this);
18          setContentView(webView);
19
20          WebSettings webSettings = webView.getSettings();
21          webSettings.setJavaScriptEnabled(true);
22
23          // связь JS -> Android
24          webView.addJavascriptInterface(new WebAppInterface(), name: "Android");
```

Рис. 1. Создание моста

В данной работе внимание уделялось реализации метода сохранения рекордов. Метод принимает целочисленное значение счета из JavaScript и сохраняет его в `SharedPreferences` (рис. 2).

```
29      // передача данных из JS в Android
30      public class WebAppInterface { 1 usage
31          @android.webkit.JavascriptInterface no usages
32          public void saveScore(int score) {
33              SharedPreferences prefs = getSharedPreferences(name: "GamePrefs", MODE_PRIVATE);
34              SharedPreferences.Editor editor = prefs.edit();
35
36              int bestScore = prefs.getInt(s: "best_score", i: 0);
37              if (score > bestScore) {
38                  editor.putInt(s: "best_score", score);
39              }
40
41              for (int i = 9; i >= 1; i--) {
42                  int prevScore = prefs.getInt(s: "last_score_" + i, i: -1);
43                  if (prevScore != -1) {
44                      editor.putInt(s: "last_score_" + (i + 1), prevScore);
45                  }
46              }
47              editor.putInt(s: "last_score_1", score);
48
49              editor.apply();
50          }
51      }
```

Рис. 2. Метод передачи данных

В момент завершения игры, когда пользователь выполнил все задания, JS-код вызывает метод сохранения очков через глобальный объект Android (рис. 3).

```

146         if (matches === gameBoard.children.length / 2) {
147             clearInterval(timer);
148             message.textContent = 'Все карточки найдены!';
149             winSound.currentTime = 0;
150             winSound.play();
151             gameStarted = false;
152             //сохранения очков в Android
153             if (window.Android) {
154                 Android.saveScore(score);
155             }
156         }
157     } else {
158         unflipCards();
159     }
160 }

```

Рис. 3. Метод сохранения очков

Для отображения сохраненных рекордов создается отдельная активность RecordsActivity. В ней реализована логика загрузки данных из SharedPreferences и их отображения в виде отсортированного списка (рис. 4).

```

33     private void loadRecords() { Usage
34
35         int bestScore = prefs.getInt("best_score", 0);
36         bestScoreTextView.setText(bestScore + " очков");
37
38         // последних 10
39         List<String> lastScores = new ArrayList<>();
40         for (int i = 1; i <= 10; i++) {
41             int score = prefs.getInt("last_score_" + i, -1);
42             if (score != -1) {
43                 lastScores.add("Игра " + i + ": " + score + " очков");
44             }
45         }
46
47         Collections.sort(lastScores, (String s1, String s2) -> {
48             int score1 = Integer.parseInt(s1.split(":")[1].replace(" очков", ""));
49             int score2 = Integer.parseInt(s2.split(":")[1].replace(" очков", ""));
50             return Integer.compare(score2, score1);
51         });
52
53         ArrayAdapter<String> adapter = new ArrayAdapter<>(
54             context, this,
55             android.R.layout.simple_list_item_1,
56             lastScores
57         );
58         recordsListView.setAdapter(adapter);
59     }

```

Рис. 4. Отображения данных

После тестирования, данные поступают в приложение, где сохраняются через SharedPreferences. Для проверки корректности передачи можно увидеть запись на экране рекордов, где система отображает последние результаты игр в отсортированном виде (рис. 5).

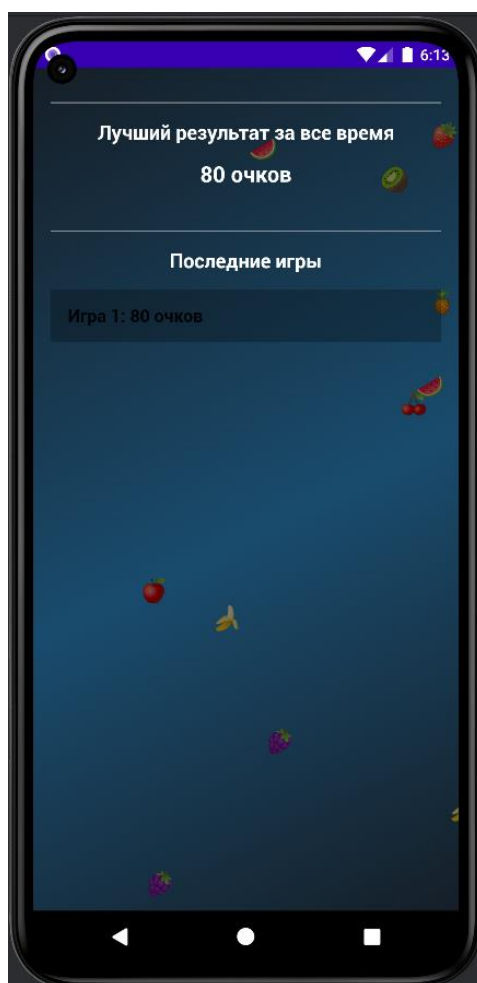


Рис. 5. Проверка передачи данных

4 Выводы

Разработанное решение успешно реализует интеграцию JavaScript-игры с нативной Android-частью через WebView и JavascriptInterface, обеспечивая надежную передачу данных о рекордах. Использование SharedPreferences позволяет сохранять результаты между сеансами игры, демонстрируя практическую эффективность гибридного подхода. Предложенный метод требует минимальных изменений в исходный код игры, сохраняя при этом все преимущества нативной платформы.

Библиографический список

1. Иванов М.Д. Оптимизация архитектуры Android-приложений. // Образование России и актуальные вопросы современной науки. сборник статей Всероссийской научно-практической конференции. 2018. С. 121-124.

2. Казакова А.Н. Исследование способов хранения информации в android приложениях. // Материалы конференций ГНИИ "НАЦРАЗВИТИЕ". Декабрь 2017. 2018. С. 38-39.
3. Осокин В.В., Бендик В.А. Android как платформа для разработки корпоративных приложений на примере сrm. // Интеллектуальные системы. Теория и приложения. 2014. Т. 18. № 2. С. 179-188.
4. Мурашко Ю.В. Методы хранения данных в ос android. // Решение. 2019. Т. 1. С. 144-146.
5. Дейч В.С. 10 библиотек, которые облегчат android-разработку в 2021 году. // Цифровизация и кибербезопасность: современная теория и практика. сборник научных трудов по материалам Международной научнопрактической конференции. Омск, 2021. С. 16-20.