

Разработка игры на Android с использованием JavaScript

Андриенко Иван Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье описывается процесс разработки игры на Android с использованием JavaScript. Рассматриваются ключевые аспекты создания игры, включая интеграцию JavaScript в Android-приложение, создание игрового процесса и обеспечение плавного взаимодействия между пользователем и игрой. В качестве примера приводится реализация 2D-игры «Башня», в которой игрок должен строить башню из блоков, размещая их как можно ровнее. Показано, как с помощью JavaScript можно эффективно реализовать игровой процесс и интегрировать его в Android-приложение, обеспечивая совместимость и гибкость разработки.

Ключевые слова: Android, JavaScript, разработка игр, html, css, игровой процесс.

Developing an Android game using JavaScript

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article describes the process of developing an Android game using JavaScript. The key aspects of creating a game are considered, including integrating JavaScript into an Android application, creating gameplay, and ensuring smooth interaction between the user and the game. As an example, the implementation of the 2D Tower game is given, in which the player must build a tower of blocks, placing them as evenly as possible. It shows how JavaScript can be used to effectively implement the gameplay and integrate it into an Android application, ensuring compatibility and flexibility of development.

Keywords: Android, JavaScript, game development, html, css, gameplay.

1 Введение

1.1 Актуальность

В условиях современного мобильного мира разработка игр становится все более востребованной и популярной задачей среди программистов. JavaScript, являясь одним из наиболее распространенных языков программирования, предоставляет широкие возможности для создания кроссплатформенных игр. Интеграция JavaScript в Android-приложение

позволяет разработчикам использовать единую кодовую базу для создания игр, которые могут быть запущены на различных платформах, обеспечивая гибкость и сокращение времени разработки. В свете возрастающего спроса на качественные мобильные игры, использование JavaScript в разработке игр на Android становится актуальным и перспективным направлением.

1.2 Обзор исследований

С.И. Федосенко исследует платформы для разработки мобильных приложений на JavaScript, акцентируя внимание на их мультиплатформенных возможностях [1]. В работе А.С. Ильясова описывает процесс создания гибридных приложений для Android с применением технологий Java, подчеркивая их эффективность [2]. А.Г. Иванов исследует алгоритмы взаимодействий и обмена данными между различными платформами, что позволяет оптимизировать процессы разработки для Web и мобильных устройств [3]. В своей работе А.С. Винокуров и Р.И. Баженов представляют проект разработки мультиплатформенного мобильного приложения для фитнес-клуба с использованием JavaScript, демонстрируя его успешную реализацию в условиях реального бизнеса [4]. В.К. Харлампики в своем исследовании рассматривает возможности Framework7 для мультиплатформенной разработки, выделяя его преимущества для создания современных мобильных приложений [5].

1.3 Цель исследования

Цель исследования – разработать эффективный подход к созданию игры на Android с использованием JavaScript. Это включает интеграцию JavaScript-кода в Android-приложение, разработку игрового процесса, а также обеспечение плавного и отзывчивого взаимодействия с пользователем.

2 Материалы и методы

Для исследования использовались Android Studio, язык программирования JavaScript, и WebView для интеграции игрового процесса в Android-приложение. Основное внимание уделялось созданию игры «Башня», где пользователь должен строить башню из блоков, размещая их как можно ровнее. Использование JavaScript позволило обеспечить кроссплатформенную совместимость и гибкость разработки.

3 Результаты и обсуждения

Первоначально нужно создать новую папку в проекте в среде разработки Android Studio. Для этого необходимо щелкнуть правой кнопкой мыши на папке app, расположенной в левой верхней части интерфейса, и в появившемся контекстном меню выбрать опцию New → Folder → Assets Folder.

После создания папки внутри проекта, в этой папке будут размещены файлы игры: файлы javascript, файл css и html файл.

Далее добавляем элемент WebView в макет Android-приложения. Это позволяет интегрировать веб-контент непосредственно в пользовательский интерфейс приложения. Благодаря параметрам `match_parent`, WebView занимает все доступное пространство на экране устройства. Это особенно важно для игр, чтобы пользователи видели игру в полный экран, не отвлекаясь на другие элементы интерфейса. (рис.1).

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:id="@+id/main"
7      android:layout_width="match_parent"
8      android:layout_height="match_parent"
9      tools:context=".MainActivity">
10
11      <WebView
12          android:layout_width="match_parent"
13          android:layout_height="match_parent"
14          android:id="@+id/webview" />
15
16  </androidx.constraintlayout.widget.ConstraintLayout>
```

Рис. 1. Настройка FileProvider в манифесте

Для инициализации и настройки WebView пропишем код в MainActivity. Важной частью является поиск и настройка WebView. Сначала WebView находится в макете с помощью метода `findViewById(R.id.webview)`. Затем включается поддержка JavaScript в WebView с помощью `webView.getSettings().setJavaScriptEnabled(true);`, что позволяет исполнять скрипты, необходимые для работы игры. Далее загружается HTML-страница с игрой из папки assets (рис. 2).

```
1 package com.modeg.tower;
2
3 > import ...
12
13 ></> public class MainActivity extends AppCompatActivity {
14
15     @SuppressWarnings("SetJavaScriptEnabled")
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         EdgeToEdge.enable( $this$enableEdgeToEdge: this);
20         setContentView(R.layout.activity_main);
21         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
22             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
23             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
24             return insets;
25         });
26
27         setContentView(R.layout.activity_main);
28         WebView webView=findViewById(R.id.webview);
29         webView.getSettings().setJavaScriptEnabled(true);
30         webView.loadUrl("file:///android_asset/index.html");
31     }
32 }
```

Рис. 2. Инициализация WebView

Важным аспектом разработки было создание игры «Башня» на JavaScript. Игра представляет собой 2D-проект, где игрок строит башню, размещая блоки друг на друге. Основной игровой цикл и обработка взаимодействия пользователя были реализованы на JavaScript. В начале настраивается canvas, устанавливаются его размеры и создается объект игры, который управляет логикой и состоянием. Основной игровой цикл очищает canvas, обновляет и рисует игровые элементы, а также проверяет, окончена ли игра. Если игра окончена, отображается финальный счет и сообщение о завершении. Также реализуем класс tower. Он управляет блоками в игре, где каждый блок представляет собой площадку. Он создает новые блоки, управляет их движением и падением, проверяет их столкновения с другими блоками или краем экрана. При успешном размещении блока обновляется счет, а при ошибке создаются частицы для визуального эффекта. Код также включает функции для отрисовки блоков и частиц, а также для сброса состояния игры (рис. 3, 4).

```

1  const canvas = document.getElementById('gameCanvas');
2  const context = canvas.getContext('2d');
3  canvas.width = 400;
4  canvas.height = 600;
5
6  const tower = new Tower(context, canvas.width, canvas.height);
7
8  function gameLoop() {
9      context.clearRect(0, 0, canvas.width, canvas.height);
10
11      if (!tower.gameOver) {
12          tower.moveFloor();
13          tower.drawFloors();
14          tower.drawCurrentFloor();
15          tower.drawParticles();
16          requestAnimationFrame(gameLoop);
17      } else {
18          document.getElementById('final-score').innerText = `Баш чыет: ${tower.score}`;
19          document.getElementById('game-over').style.display = 'block';
20      }
21  }
22
23  window.addEventListener('click', () => {
24      if (!tower.currentFloor) {
25          tower.createFloor();
26      } else if (!tower.isDropping) {
27          tower.dropFloor();
28      }
29  });
30
31  document.getElementById('restart-button').addEventListener('click', () => {
32      document.getElementById('game-over').style.display = 'none';
33      tower.reset();
34      tower.createFloor();
35      gameLoop();
36  });
37
38  tower.createFloor();
39  gameLoop();

```

Рис. 3. Реализация класса game.js

```

35  dropFloor() {
36      this.isDropping = true;
37  }
38
39  moveFloor() {
40      if (this.currentFloor) {
41          if (this.isDropping) {
42              this.dropProgress += 0.0;
43              this.currentFloor.y = this.dropStartY + (this.dropProgress * this.verticalSpeed);
44
45              if (this.currentFloor.y + this.floorHeight >= this.canvasHeight ||
46                  (this.floors.length > 0 && this.currentFloor.y + this.floorHeight >= this.floors[this.floors.length - 1].y)) {
47
48                  if (this.floors.length > 0) {
49                      const lastFloor = this.floors[this.floors.length - 1];
50                      const overlap = Math.abs(this.currentFloor.x - lastFloor.x);
51                      if (overlap > this.floorWidth / 2) {
52                          this.gameOver = true;
53                          this.createParticles(this.currentFloor.x + this.floorWidth / 2, this.currentFloor.y + this.floorHeight);
54                          return;
55                      }
56                  } else {
57                      if (this.currentFloor.y + this.floorHeight >= this.canvasHeight) {
58                          this.createParticles(this.currentFloor.x + this.floorWidth / 2, this.canvasHeight);
59                      }
60                  }
61
62                  this.floors.push(this.currentFloor);
63                  this.score++;
64                  document.getElementById('score').innerText = `Чыет: ${this.score}`;
65                  this.currentFloor = null;
66                  this.createParticles(this.x + this.floorWidth / 2, this.startY);
67              }
68          } else {
69              this.currentFloor.x += this.horizontalSpeed * this.direction;
70              if (this.currentFloor.x <= 0 || this.currentFloor.x + this.floorWidth >= this.canvasWidth) {
71                  this.direction *= -1;
72              }
73          }
74      }

```

Рис. 4. Реализация класса tower.js

Для запуска игры необходимо создать HTML файл. Он создает структуру и подключает необходимые ресурсы (рис. 5).

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Башня</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10   <div id="game-container">
11     <canvas id="gameCanvas"></canvas>
12     <div id="score">Счет: 0</div>
13     <div id="game-over" style="...">
14       <p>Игра окончена!</p>
15       <p id="final-score"></p>
16       <button id="restart-button">Сыграть еще раз</button>
17     </div>
18   </div>
19   <script src="tower.js"></script>
20   <script src="game.js"></script>
21 </body>
22 </html>
23
```

Рис. 5. Создание файла index.html

После разработки было проведено тестирование. Все основные функции игры работают без сбоев (рис. 6).

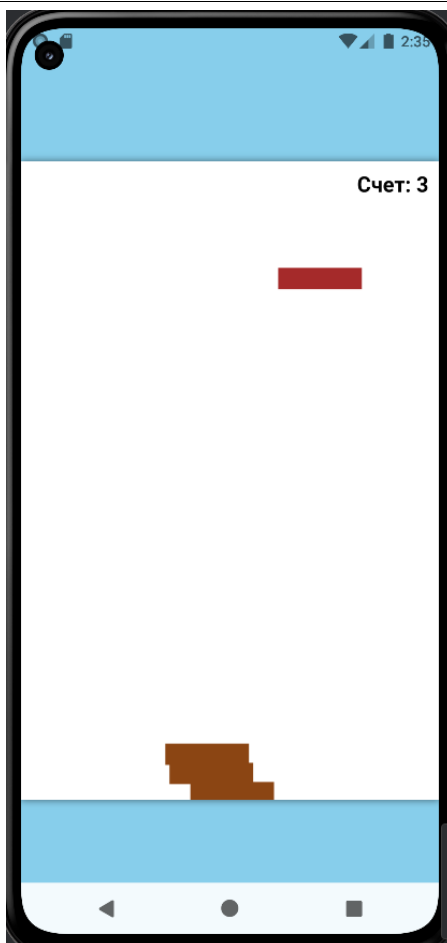


Рис. 6. Тестирование игры

Выводы

В данной статье представлен процесс разработки игры на Android с использованием JavaScript. Рассмотрены ключевые аспекты интеграции JavaScript в Android-приложение, создания игрового процесса и обеспечения взаимодействия пользователя с игрой. Приведены примеры реализации, демонстрирующие, как можно эффективно использовать JavaScript для разработки кроссплатформенных игр, обеспечивая гибкость и сокращение времени разработки. Использование JavaScript в Android-приложениях позволяет разработчикам создавать игры, которые могут легко адаптироваться к различным платформам и устройствам.

Библиографический список

1. Федосенко С.И. Платформа для разработки мобильных приложений на JavaScript // В сборнике: Электронные системы и технологии. Сборник материалов 57-й научной конференции аспирантов, магистрантов и студентов БГУИР. Минск, 2021. С. 428-431.
2. Ильясова А.С. Создание гибридных приложений для Android с применением технологий Java // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2016. № 3 (13). С. 52-59.

3. Иванов А.Г. Разработка алгоритмов взаимодействий и динамического обмена данными на разных платформах в Web и на мобильных устройствах // В сборнике: Современная наука: прогнозы, факты, тенденции развития. Сборник материалов XVI Международной научно-практической конференции. Чебоксары, 2023. С. 210-216.
4. Винокуров А.С., Баженов Р.И. Проект разработки мультиплатформенного мобильного приложения для фитнес-клуба со встроенным JavaScript модулем // Постулат. 2016. № 3 (3). С. 5.
5. Харлампики В.К. Мультиплатформенная разработка мобильных приложений с использованием Framework7 // Международный научно-исследовательский журнал. 2023. № 7 (133).