УДК 004

Разработка игры «Виселица» на Android

Андриенко Иван Сергеевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье представлено пошаговое руководство по разработке игры "Виселица" на платформе Android с использованием Android Studio. В статье описывается процесс создания пользовательского интерфейса, реализация игровой логики и обработки пользовательского ввода. Читатели получат подробное представление о разработке игр на Android. Конечный результат - полноценная игра, готовая к установке и использованию на устройствах Android.
Ключевые слова: Android, Виселица, игра, разработка, мобильное приложение, Android Studio, пользовательский интерфейс.

Development of “the Gallows” game on Android

Andrienko Ivan Sergeevich
Sholom-Aleichem Priamursky State University
Student

Abstract
This article provides a step-by-step guide to the development of the game "Gallows" on the Android platform using Android Studio. The article describes the process of creating a user interface, implementing game logic and processing user input. Readers will get a detailed idea of game development on Android. The end result is a full-fledged game ready for installation and use on Android devices.
Keywords: Android, gallows, game, development, mobile application, Android Studio, User interface.

1 Введение

1.1 Актуальность

Разработка игры Виселица на платформе Android имеет актуальность, поскольку игры на мобильных устройствах становятся все более популярными и востребованными среди пользователей. Реализация данной игры предоставляет разработчикам возможность погрузиться в процесс создания мобильного приложения, овладеть основными концепциями разработки на платформе Android и научиться применять их на практике. Также, разработка игры Виселица предоставляет возможность изучить особенности работы с игровой логикой, обработкой пользовательского ввода и созданием привлекательного пользовательского интерфейса. Эти навыки могут быть

применены в различных областях разработки мобильных игр и приложений, открывая новые перспективы для разработчиков и предоставляя возможность создавать интересные и увлекательные игровые продукты на платформе Android.

1.2 Обзор исследований

В своей работе Т.Ю. Батюшкина, В.С. Новгородова представили разработка 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]. С.В. Гуляева, А.А. Бурнашева описали возможности мобильных технологий и эффективность их использования в процессе развития логики детей дошкольного возраста [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 работе Т.Ю. Батюшкина, Р.С. Манасытов представили создание мобильного приложения с помощью Android Studio. Проведен сравнительный анализ характеристик игровых движков. Обосновывается актуальность исследования, указываются цели, новизна и практическая значимость. Были рассмотрены методы и возможности среды разработки Android Studio для создания мобильной игры. Приведены описание технологий создания мобильной игры [4].

1.3 Цель исследования

Цель исследования – разработать игру полнофункциональную игру Виселица, которая будет готова к развертыванию на устройствах Android, используя Android Studio

2 Материалы и методы

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

3 Результаты и обсуждения

Для разработки игры в Android Studio создадим новый проект. Шаблон Empty Activity предоставляет базовую структуру проекта с одной активностью. Также необходимо указать невысокую версию SDK, чтобы обеспечить совместимость с широким спектром устройств. В качестве языка программирования будет использоваться java.

В приложении "Виселица" для платформы Android были реализованы следующие ключевые элементы:

1. Поле для ввода буквы: Для угадывания буквы в слове игрок может использовать поле для ввода, где он может ввести одну букву из алфавита.

2. Поле с вопросом: В игре "Виселица" каждый раунд связан с определенным вопросом или загадкой. Этот вопрос отображается в поле с вопросом, чтобы игрок мог попробовать угадать правильный ответ, вводя буквы.

3. Изображение виселицы: Визуальное представление состояния игры отображается через изображение виселицы. При каждой неправильной попытке угадать букву, изображение виселицы меняется, приближая игрока к проигрышу.

4. Кнопка "Проверить": После ввода буквы игрок нажимает на кнопку "Проверить", чтобы проверить свой ответ. Это инициирует процесс проверки угаданной буквы и обновление состояния игры.

5. Поле с текущим состоянием слова: Для обратной связи и отображения прогресса игроку предоставляется поле с текущим состоянием слова. Каждая угаданная буква отображается в соответствующем месте в слове, а неразгаданные буквы обозначаются пустыми местами.

6. Кнопка "Сыграть еще": По окончании игры, независимо от результата, игроку предлагается кнопка "Сыграть еще", чтобы начать новый раунд игры.

В качестве виселицы были нарисованы 7 изображений (рис. 1). Это значит, что у пользователя будет 6 попыток, так как первое изображение используется сразу.

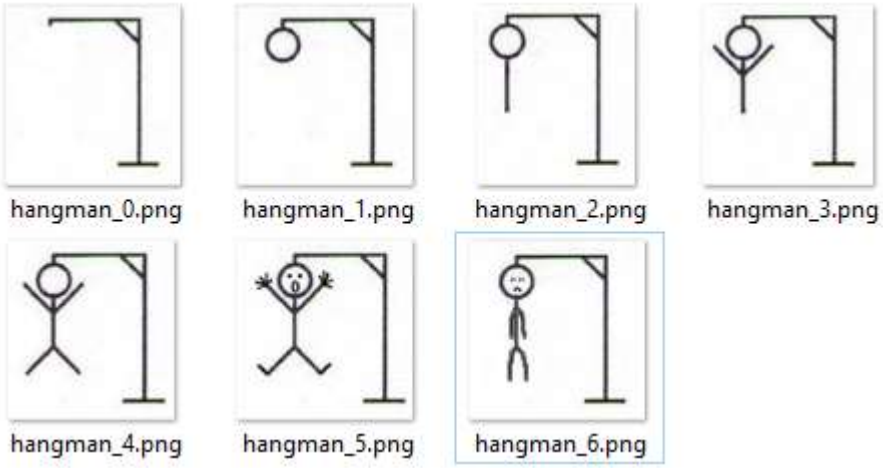

Рисунок 1 – Изображения виселицы

Переходи к созданию интерфейса. Сверху будет находиться вопрос, на который пользователь должен ответить. Затем идет поле для отгаданных букв и само поле вода, далее идет кнопка и изображение виселицы. Прописываем код в «activity_main.xml» (рис. 2).

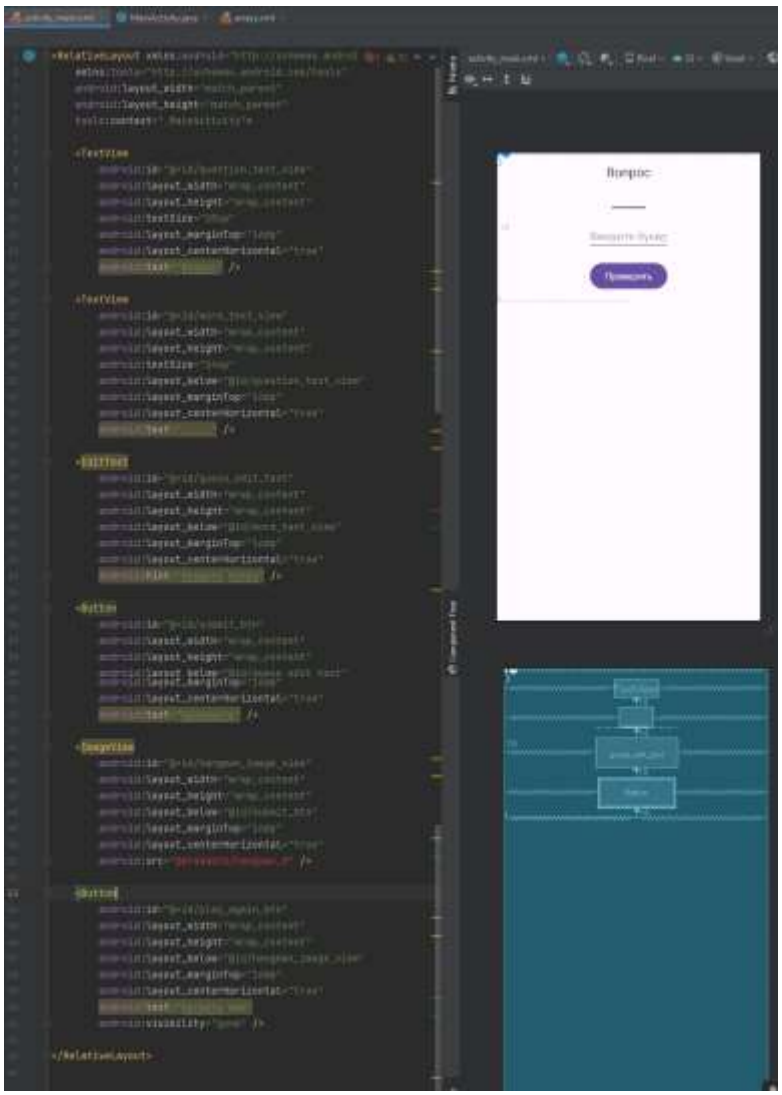

Рисунок 2 – Создание макета приложения

Пропишем основную логику для приложения в главной активности. Импортируем необходимые библиотеки. Добавим несколько переменных, таких как «guessEditText» (поле для ввода буквы), «submitBtn» (кнопка "Проверить"), «playAgainBtn» (кнопка "Сыграть еще"), «wordTextView» (поле для отображения текущего состояния слова) и «hangmanImageView» (изображение виселицы). Также объявлены переменные «currentWord» (текущее слово, которое нужно угадать), «guessedLetters» (строитель строки для отслеживания угаданных букв), «incorrectAttempts» (счетчик неправильных попыток), «questions» (список вопросов) и «currentQuestionIndex» (индекс текущего вопроса) (рис. 3).

Рисунок 3 – Импорт библиотек и создание переменных

Далее создадим инициализацию элементов пользовательского интерфейса и установим обработчика нажатия на кнопку "Сыграть еще". В методе «onCreate()» происходит установка контента активности с использованием разметки «activity_main». Затем происходит связывание переменных с соответствующими элементами интерфейса, такими как поле ввода «guessEditText», кнопка «submitBtn», кнопка "Сыграть еще" «playAgainBtn», текстовое поле с текущим словом «wordTextView», изображение виселицы «hangmanImageView» и текстовое поле с вопросом «questionTextView». Загружается список вопросов, и устанавливается первый вопрос из списка в «questionTextView». Также устанавливается обработчик нажатия на кнопку "Сыграть еще", который вызывает метод «startNewGame()» для начала новой игры, а также изменяет видимость кнопок и включает возможность ввода букв в поле «guessEditText» (рис. 4).

Рисунок 4 – Создание инициализации переменных и обработчиков

При нажатии на кнопку «Отправить», получается введенная пользователем буква из поля ввода «guessEditText». Затем проверяется, является ли введенный символ корректной буквой. Если это одиночная буква, то вызывается метод «processGuess()», который обрабатывает угаданную букву. В случае, если пользователь ввел некорректную букву, например, ввел строку или несколько символов, выводится сообщение об ошибке с помощью «Toast». После обработки введенной буквы, поле ввода очищается и запускается новая игра с помощью метода «startNewGame()» (рис. 5).

Рисунок 5 – Создание обработчиков

Добавим инициализацию новой игры. При вызове метода «startNewGame()», происходят следующие действия: получение нового случайного слова, создание пустой строки для угаданных букв «guessedLetters», сброс счетчика неправильных попыток «incorrectAttempts». Затем происходит очистка «TextView» для отображения слова, установка изображения виселицы на начальное состояние, очистка вопросов и установка

первого вопроса. Затем инициализируется строка «guessedLetters», состоящая из подчеркиваний, которые соответствуют буквам загаданного слова. Наконец, происходит обновление «TextView» с текущим состоянием слова (рис. 6).

```
2 usages
private void startNewGame() {
    currentWord = getRandomWord();
    guessedLetters = new StringBuilder();
    incorrectAttempts = 0;

    wordTextView.setText("");

    hangmanImageView.setImageResource(R.drawable.hangman_0);

    currentQuestionIndex = 0;

    TextView questionTextView = findViewById(R.id.question_text_view);
    questionTextView.setText(questions.get(currentQuestionIndex));

    for (int i = 0; i < currentWord.length(); i++) {
        guessedLetters.append("_");
    }

    wordTextView.setText(getCurrentWordState());
}
```

Рисунок 6 – Создание события «Новая игра»

Далее необходимо обработать угадывание буквы. В методе «processGuess()» сначала проверяется, есть ли угаданная буква в загаданном слове. Если угаданная буква присутствует в слове, она заменяет соответствующее подчеркивание в строке «guessedLetters», и устанавливается флаг «correctGuess» в значение «true». Если угаданная буква отсутствует в слове, увеличивается счетчик неправильных попыток «incorrectAttempts» и обновляется изображение виселицы. Затем происходит обновление «TextView» с текущим состоянием слова. После этого происходит проверка на завершение игры: если количество неправильных попыток достигло 6, вызывается метод «handleGameEnd(false)» для обработки поражения пользователя, если же все буквы слова угаданы, вызывается метод «handleGameEnd(true)» для обработки победы пользователя (рис. 7).

```
1 usage
private void processGuess(char letter) {
    boolean correctGuess = false;
    for (int i = 0; i < currentWord.length(); i++) {
        if (currentWord.charAt(i) == letter) {
            guessedLetters.setCharAt(i, letter);
            correctGuess = true;
        }
    }

    if (!correctGuess) {
        incorrectAttempts++;
        updateHangmanImage();
    }

    wordTextView.setText(getCurrentWordState());

    if (incorrectAttempts == 6) {
        handleGameEnd( isWin: false);
    } else if (String.valueOf(guessedLetters).equals(currentWord)) {
        handleGameEnd( isWin: true);
    }
}
```

Рисунок 7 – Метод «processGuess()»

После происходит обработка завершения игры. В методе «handleGameEnd()»: в зависимости от значения параметра «isWin», которое указывает, выиграл ли пользователь, выводится соответствующее сообщение в виде всплывающего уведомления. Если пользователь угадал слово, параметр «isWin» равен «true», выводится сообщение о победе. В случае, если пользователь проиграл, параметр «isWin» равен «false», выводится сообщение о поражении. Затем элементы интерфейса (поле ввода, кнопки) блокируются, а кнопка "Играть снова" становится видимой (рис. 8).

```
2 usages
private void handleGameEnd(boolean isWin) {
    if (isWin) {
        Toast.makeText( context: this,  text: "Вы выиграли!", Toast.LENGTH_SHORT).show();
    } else {
        Toast.makeText( context: this,  text: "Вы проиграли!", Toast.LENGTH_SHORT).show();
    }

    guessEditText.setEnabled(false);
    submitBtn.setEnabled(false);
    playAgainBtn.setVisibility(View.VISIBLE);
}

2 usages
private String getCurrentWordState() {
    return guessedLetters.toString();
}
```

Рисунок 8 – Обработчик завершения игры

Для того чтобы изображение обновлялось в методе «updateHangmanImage()» происходит обновление изображения виселицы на основе количества неправильных попыток «incorrectAttempts».

Идентификатор изображения получается с помощью метода «getResources().getIdentifier()», который формирует идентификатор на основе имени ресурса. Затем метод «setImageResource()» устанавливает соответствующее изображение в «hangmanImageView». В методе «getRandomWord()» случайным образом выбирается слово из списка. Список слов получается с помощью «getResources().getStringArray()», который получает массив строк из ресурсов по их идентификатору. Затем генерируется случайный индекс с помощью «nextInt()» и выбирается соответствующее слово из массива. Полученное случайное слово возвращается из метода (рис. 9).

Рисунок 9 – Метода «getResources().getIdentifier()» и «getRandomWord()»

Тестируем разработанную игру (рис. 10, 11).

Рисунок 10 – Процесс игры

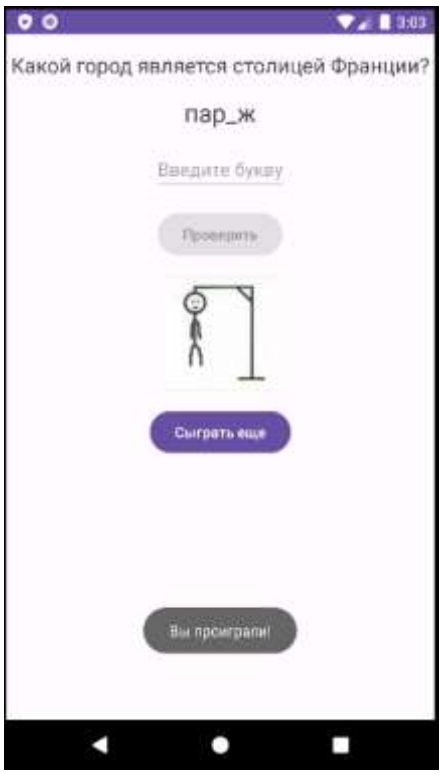

Рисунок 10 – Конец игры

Выводы

В данной научной статье был представлен процесс разработки игры "Виселица" на платформе Android. Разработанное приложение предоставляет пользователям возможность играть в игру, угадывая слова по буквам. Были реализованы основные функциональности, включая ввод букв, отображение вопросов и управление состоянием игры. Дальнейшее развитие приложения может включать расширение списка вопросов, улучшение графического интерфейса и добавление новых функций для обогащения игрового процесса и улучшения пользовательского опыта.

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. 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.
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. Батюшкина Т.Ю., Манасытов Р.С. Создание мобильного приложения с помощью android studio // Мировые научные исследования и разработки в

эпоху цифровизации. Сборник статей XV Международной научно-практической конференции. Ростов-на-Дону, 2021. С. 133-135.