Math Quiz, Flip It and Add up Android App

Requirements

Project Overview

The Math Quiz Android App is designed to provide users with an interactive math quiz game. Users are presented with math questions, and they need to choose the correct answer from multiple choices. The app keeps track of their score and presents the final score when the quiz is completed.

The Card Flipping Game is a web-based memory game that challenges players to match pairs of cards with identical symbols. The game features a grid of face-down cards, and players must flip them over to reveal the symbols. The objective is to find all matching pairs within a certain time limit or a set number of moves.

The Add up game is a digital puzzle where players combine numbered tiles to reach the elusive 2048 tile. It challenges strategic thinking and offers an addictive gameplay experience. The objective is to slide and merge tiles to achieve the highest score possible.

Functional Requirements

1. The app should generate random math questions with options.

2. Questions should include addition, subtraction, and multiplication operations.

3. Users should be presented with three answer options for each question.

4. Users should be able to select an answer by tapping one of the options.

5. The app should provide immediate feedback on whether the selected answer is correct or not.

6. After each question, the app should update the user's score and display the current question number.

7. The quiz should consist of 10 questions.

8. At the end of the quiz, the app should display the user's final score.

9. Users should have the option to restart the quiz.

Non-Functional Requirements

1. The app should be responsive and provide a user-friendly interface.

2. Questions and answer choices should be generated randomly to ensure variety.

3. The app should be performant and not lag during question transitions.

4. The app should be compatible with Android devices running Android 5.0 (Lollipop) and above.

Design

Architecture

The app follows a basic Model-View-Controller (MVC) architecture:

Model: Represents the data (e.g., question, answer choices, score).

-View: Represents the user interface, including layout and widgets.

-Controller: Handles user interactions and updates the model and view accordingly.

User Interface

The app consists of the following key elements:

- Main screen with the current question number, score, and question.

- Three buttons representing answer choices.

- Feedback on whether the selected answer is correct.

- A result screen displaying the final score.

Implementation

MainActivity

- Initializes the app and user interface.

- Generates random math questions.

- Handles user interactions and updates the score and question number.

- Provides feedback on selected answers.

- Displays the result screen when the game is completed.

Question Generator

- Generates random math questions with appropriate answer choices.

- Ensures variety in operations (addition, subtraction, multiplication).

- Randomly positions the correct answer among the choices.

ResultActivity

- Displays the final score to the user.

- Offers an option to restart the quiz.

\*\*Installation\*\*

To install the app on an Android device:

1. Download the APK file from the PlayStoree

2. Enable installation from unknown sources in your device's settings.

3. Open the APK file and follow the installation instructions.

Usage

1. Launch the app.

2. Start the game.

3. Answer the questions presented.

4. See your final score at the end of the quiz, or on top

**ROLES**

Our team, consisting of Mpumelelo, Thulani, Othandwayo, and Laina, worked collaboratively to develop the Math Quiz Android App. As a group, we collectively wrote the Java code and learnt how to navigate Android Studio together, making it a shared effort. We also conducted research collectively to gather insights and ideas for the project. Each team member played a vital role in contributing to the development process, and our collective effort resulted in the successful creation of the app.

1. Othandwayo

- Role:Team Leader

- Responsibilities:

- Led the coding efforts and architecture design.

- Implemented core functionality, including generating math questions and user interaction handling.

- Ensured code quality and app performance.

2.Laina

- Role: UI & UX developer

- Responsibilities:

- Designed the user interface (UI) layout.

- Created graphical assets like buttons and icons.

- Collaborated with the development team to integrate the UI elements.

3. Mpumelelo

- Role: Researcher & Project Coordinator

- Responsibilities:

- Conducted thorough testing of the app.

- Identified and reported bugs and issues.

- Ensured the app's overall quality and responsiveness.

4. Thulani

- Role: Project manager

-Responsibilities:

- Coordinated team meetings and discussions.

- Assisted with project planning and task distribution.

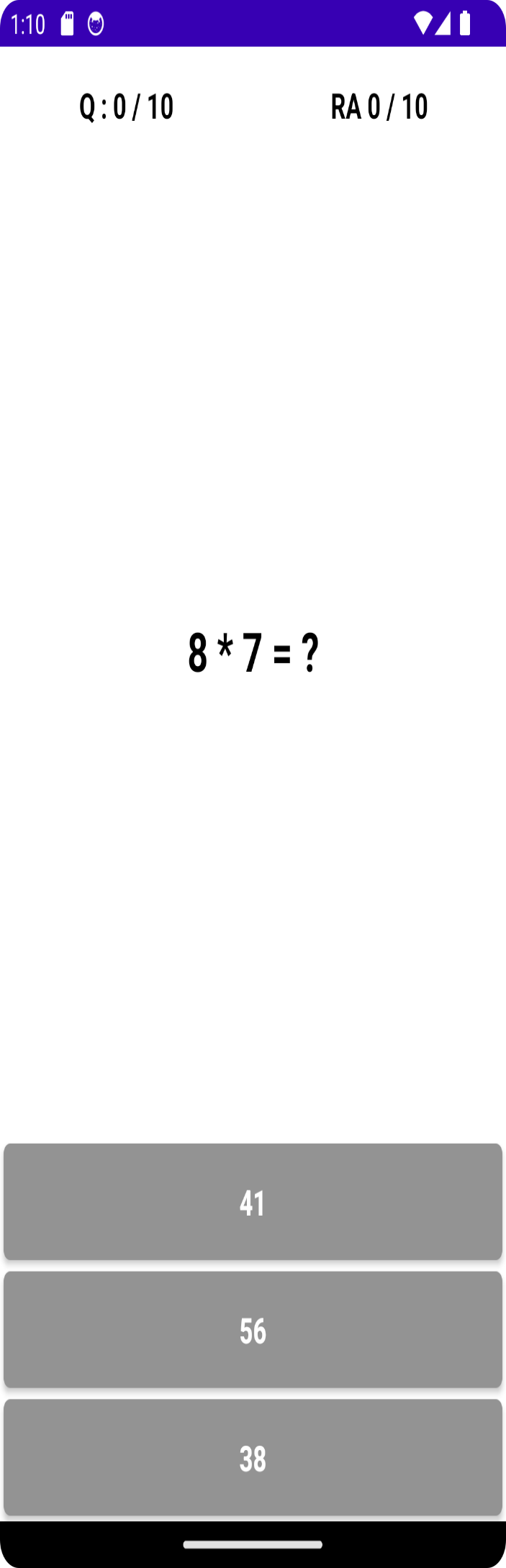
- Contributed to the coding effort and collaborated on Java code.

- Participated in research activities and collected insights for the project.

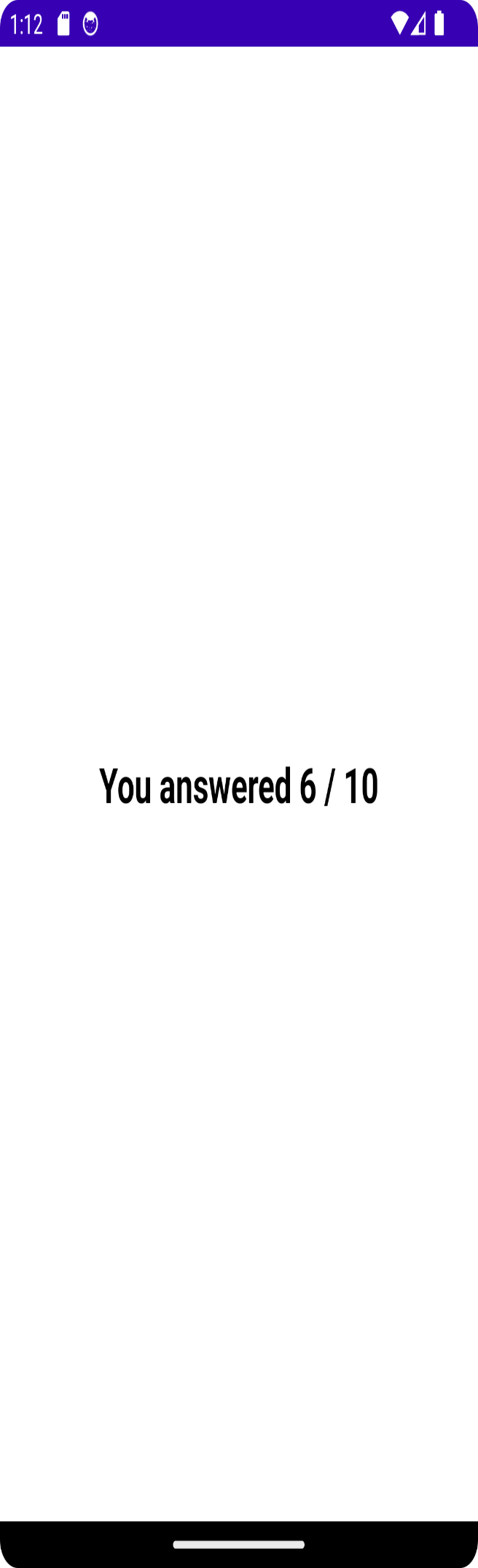
Each team member had a specific role, but collectively, we all participated in the coding and research aspects of the project, ensuring its successful development.

In compiling this document, our team leveraged a variety of resources to ensure a comprehensive and well-structured result. Notably, Mpumelelo, who had a background in IS201, brought valuable insights into app development principles. We drew from IS201 class notes to enhance our understanding of app development processes and best practices. As for the coding aspect, it was truly a collective effort. We extensively utilized Stack Overflow, a popular online community for programmers, to troubleshoot coding challenges and seek solutions. Additionally, YouTube tutorials on Android Studio played a pivotal role in guiding us through the app development journey. These resources, combined with our collaborative efforts, allowed us to create a well-informed and functional document for our project

Below, are some of the pictures from running the app on the emulator.



This is a sample picture from the application, a question is given and 3 options are given for the user to choose from, if they get it correct then the screen turns green and the RA (Right Answers) moves by 1 unit each time if not then it does not move. The Q is for Question to show the question number. This was done in questions of 10, from easy to medium to hard on a random manner each time the app is launched.



This is what happens when you have answered the 10 random Math questions, the 6/10 is the score you got based on your correct answers. After this, the app goes back to question one which is, again, random and cycles in this manner.



