Documentation on TicTacToe mini-project

Aim: The aim of the project is to understand the development of a Tic-Tac-Toe for Android devices. The project involves creating a mobile app that allows players to engage in the classic two-player Tic-Tac-Toe game as well as play against an AI opponent with varying levels of difficulty.

* **The main features of this project are:**

1. **Two-Player Gameplay**: The app allows two players to take turns playing Tic-Tac-Toe on the same device.
2. **Single-Player Mode with AI**: The app provides an option for a single player to play against an AI opponent. The AI is implemented with different levels of difficulty: easy and hard.
3. **User Interface**: The project includes well-designed user interfaces using XML layouts, with buttons and animations to enhance the user experience.
4. **Player Name Customization**: The players' names can be customized, and their scores are tracked as they win rounds and games.
5. **Pause and Reset**: The game includes functionality to pause the game, reset the current round, start a new game, and view credits.
6. **Animations**: The app utilizes Lottie animations to provide visual feedback to players, including win animations and celebratory effects.
7. **State Management**: The game retains its state during device rotation or other interruptions using ‘onSaveInstanceState’ and ‘onRestoreInstanceState’.
8. **Dialogs:** Custom dialogs are used for pause menus, reset, and displaying credits.
9. **Intents and Activities**: The app uses intents and activities to navigate between different screens, such as game options and the main game board.

Requirements:

Minimum Hardware/software requirements: Any device running Android 11(API LEVEL 30) with a method of installing offline ‘.apk’ files.

Recommended Hardware/software requirements: Any device running Android 13(API LEVEL 33) with a method of installing offline ‘.apk’ files.

Internet connectivity: not required

Version history TTT3

0.1-TTT1

0.2-TTT2

0.3-TTT3(continued this)

1.1-buttons and text design

1.2-base game logic and linking buttons

1.3-base design implementations and modifications

2.1-loading screen 1st implementation

2.2- loading screen modifications and design changes

2.3-loading bar implementation and final design loading screen changes

3.1-pause menu implementation

3.2-credit screen implementation

3.3-college logo addition with link to college website

4.1-overall win system introduction

5.1-game start menu creation with more game options

6.1-introduction to one player and two player mode

7.1-animations for round wins and overall wins

8.1- Added options for 2 difficulties for the bot, easy and hard(1 player mode)

9.0-QC with final modifications and small changes

Basic explanation of the use of xml files and java files:

The XML layout serves as the visual representation of the game options screen, while the Java class handles the logic and behaviour associated with user interactions on that screen. The XML layout and Java class are linked through the ‘setContentView’ call and the use of ‘findViewById’ to reference UI elements. This connection enables the user interface to interact with the underlying code and data.

**References:**

1. LoadingActivity.java and activity\_loading.xml: Loading screen when the user first opens the app.
2. GameoptionsActivity.java and activity\_game\_options.xml: The game options screen the user gets after the loading is complete, allowing the user to choose the options for the game.
3. MainActivity.java and activity\_main.xml: The main user playable game after the game options menu.
4. Pause\_menu.xml: An extra clickable and interactive dialog which has extra game options during a running game.
5. Credits\_dialog.xml: A secondary dialog which can be activated from the game starting screen or the pause menu.