Mobile Application Development (COMP2008)
Assignment 1 (Part A) – 32 Marks
Build a Tic-Tac-Toe Game for Android

Objective: The objective of this assignment is to build a feature-rich Tic-Tac-Toe game for Android. By completing this assignment, students will demonstrate their understanding of user interface design, event handling, game logic, and Android app architecture.

Tic-Tac-Toe game: Tic-Tac-Toe, also known as Noughts and Crosses or Xs and Os, is a classic two-player game played on a square grid (typically played on a 3x3 grid). Two players take turns, one using X markers and the other using O markers. Players place their markers in empty cells of the grid. The first player to form a horizontal, vertical, or diagonal line of their markers wins. If all cells are filled, and no player has formed a winning line, the game is a draw.

Tic-Tac-Toe is known for its simple rules and quick gameplay, making it a popular choice for children and casual players. Despite its simplicity, the game has been studied for its mathematical properties and strategic possibilities, especially in analysing optimal moves and strategies to win or at least secure a draw. It's often one of the first games introduced to teach game theory concepts, including dominance, strategy, and decision-making. Additionally, it serves as a foundation for understanding more complex games and algorithms in the field of artificial intelligence.

This is a group assignment (a group of four or three members). The maximum mark is placed right after each requirement. In the marking criteria, there are six levels of implementation, i.e., zero attempts (0% -10% complete), novice (efforts shown – around 10%-30% are completed), promising (around 30%-50% are completed), intermediate (about 50%-70% of the requirements are achieved), competent (about 70%-90% of the requirements are achieved, and proficient (90% or above level are achieved).

Functional Requirement of your application:

1. **Implement multiple game modes (2-player, and vs. AI) [3 Marks]:** In 2-player mode, two human players take turns playing against each other on the same device. In AI mode, a human player plays against an AI opponent. Do not worry; the AI is dumb, i.e., it doesn't think, just **randomly** places its markers on empty grid cells.

2. **Enable personalisation to the Tic-Tac-Toe game [3 Marks]:** Allowing users to customise game rules, such as board size, win conditions, and player markers, adds a layer of flexibility and personalisation to the Tic-Tac-Toe game. This customisation feature enhances the user experience and accommodates different preferences. The application provides users to customise the following settings:
    a. **Board Size:** Implement a mechanism to adjust the size of the game board. This involves changing the dimensions of the grid (i.e., 3x3, 4x4, and 5x5) and updating the UI accordingly.
    b. **Win Conditions:** Users can specify the number of consecutive markers required to win the game (minimum three and maximum five). This can be achieved by adjusting the win condition logic to consider the user's input.

c. **Player Markers:** Let users choose their preferred markers (X, O, or even custom icons). Update the game logic to use the selected markers for the players.

3. **User Profile Creation and Avatar Selection [3 Mark]:** The users will be able to create their profile, i.e., name and select their avatar from a collection of avatar images. You can assume that all information is volatile, i.e., when the app closes, it is okay to have all information lost. Also enable users to edit their profile settings, including their avatar selection.

4. **Gameplay Statistics Tracking [3 Marks]:** The application should incorporate leaderboards for tracking high scores. It stores gameplay statistics such as total games played, wins, losses, draws, and win percentages for each user. Again, when the app closes, it is okay to have all information lost.

5. **In-game information [10 Marks]:** When playing the Tic-Tac-Toe game, players should have access to various pieces of information that enhance their understanding of the game state, their opponent's moves, and their progress. Here's the essential information that should be available to players while playing the game:
    a. Game Board: The main visual representation of the game, showing the current state of the cells. Clearly marked cells to indicate where each player has placed their markers (X or O or the customised icon).
    b. Player Indicators: Display the names or labels of the players (e.g., "Player 1" and "Player 2"). Indicate the current player's turn (e.g., "Your Turn" or "Opponent's Turn").
    c. Game Progress: If applicable, display the number of moves made and the total available moves. This helps players gauge their proximity to a potential win or draw.
    d. Timer: Include a countdown timer for each player's turn.
    e. Notifications and Messages: Provide informative messages or notifications about important game events, such as a win, draw, or invalid move. Display messages indicating the game's outcome, such as "Player X wins!" or "It's a draw!"
    f. Undo or Reset Option: Allow players to undo their last move or reset the game board if they make a mistake or want to start over.
    g. Settings and Menu: Provide access to settings, allowing players to adjust customisation options or return to the main menu. Include a way to pause or exit the game.
    h. User Profile: Display the user's profile information, including their avatar and username.

6. **Special Technical requirements**: There are different ways to achieve the above requirements (naïve to advanced approaches). However, the following requirements ensure that the application follows standard software design principles:
    a. Ensure that the UI layout and components adapt seamlessly to both landscape and portrait orientations and phone and tablet. Test and adjust UI elements to avoid overlapping or misalignment when the device orientation changes. [**3 Marks**]

b. Ensure in-game information is not lost when the orientation changes. [**1 Mark**]
c. Implement the game using fragment-based architecture to separate different UI components and manage their lifecycles efficiently. Create separate fragments for the main menu, settings screen, user profile, and the actual game board. [**4 Marks**]
d. Provide a grid or list of avatar options using RecyclerView (or other techniques) to display a selection of avatars for users to choose from. It means the list of avatars should be scrollable. [**2 Marks**].

Submission due date: 17 September 2023 23:59 AWST.

Submission Guidelines:
Submit your assignment electronically, via Blackboard, before the deadline. **If you submit in a group, all the members must submit the same files.**
To submit, do the following:

1. Fill out and sign a declaration of originality. A photo, scan or electronically-filled out form is fine. Whatever you do, ensure the form is complete and readable!
Place it (as a .pdf, .jpg or .png) inside your project directory.
2. Zip your entire project directory. Leave nothing out.
3. Submit your zip/tar.gz file to the assignment area on Blackboard.
4. Re-download, open, and run your submitted work to ensure it has been submitted correctly.

You are responsible for ensuring that your submission is correct and not corrupted. You may make multiple submissions, but only your newest submission will be marked. The late submission policy (see the Unit Outline) will be strictly enforced. Please note:

- DO NOT use WinRar.
- DO NOT have nested zip/tar.gz files. One is enough!
- DO NOT try to email your submission as an attachment. Curtin's email filters are configured to discard emails with potentially executable attachments silently. In an emergency, if you cannot upload your work to Blackboard, please instead upload it to Google Drive, a private GitHub repository, or another online service that preserves immutable timestamps and is not publicly accessible.

Marking Demonstration: You must demonstrate and discuss your application with a marker in a one-to-one online/in-person session. Most of the marks for your assignment will be derived from this demonstration. The demonstrator will ask you to rebuild and run your application (provided by the demonstrator) and demonstrate its major features. They may ask you about any aspect of your submission. **You will be asked about your contribution percentages during the demo. Your mark from the group submission will be adjusted based on your contribution percentage. You must specify your contribution in the individual report (check part B of the assignment).**

The demonstration schedule and policy will be published later (Check blackboard announcements). We may also cancel the demonstration-based marking due to unavoidable circumstances. In that case, it will be a full inspection-based marking.

Academic Integrity: This is an assessable task. If you use someone else's work or obtain someone else's assistance to help complete part of the assignment that is intended for you to complete yourself, you will have compromised the assessment. You will not receive marks for any parts of your submission that are not your original work.

Further, if you do not reference any external sources, you are committing plagiarism and collusion, and penalties for Academic Misconduct may apply. Please see Curtin's Academic Integrity website for information on academic misconduct (which includes plagiarism and collusion).

The unit coordinator may require you to provide an oral justification of or to answer questions about any piece of written work submitted in this unit. Your response(s) may be referred to as evidence in an Academic Misconduct inquiry.

End of Assignment