CS478: Software Development for Mobile Platforms

Project #4

Due time: 11:59 pm on 4/9/2023

Total points: 100

Instructor: Ugo Buy

TAs: Sasanka Veleti, Anupreet Paulkar and Harshil Bhatt

In this project you will design and code a strategy game called *Three Men Morris (TMM)* as an Android app. Here is a short description of the game.

The game uses a 3-by-3 board. Two players start with 3 pieces each; the board is initially empty. Next, the two players take turns placing their three pieces in an empty location in the board. The goal of the game is for each player to arrange her three pieces in a line either horizontally or vertically (but not diagonally). If there is no winner after the players have placed their pieces on the board, players will take turns moving one of their pieces to an empty location on the board. Pieces can move to *any* empty location on the board, not just adjacent locations. Additional information on TMM can be found on the Wikipedia TMM page.

Your implementation will have two Java worker threads play against each other. (The human user just watches the game without playing against the computer.) The UI thread is responsible for mainting and updating the display, and for for creating and starting the two worker threads. You must define 2 worker threads and add loopers to the threads as part of the project. Each thread will take turns taking the following actions:

1. Waiting for a short time (1-2 seconds) in order for a human viewer to notice the previous move on the display.
2. Figuring out the next move of the player associated with this thread.
3. Communicating the move to the UI thread, which will then update the device's display.
4. Waiting for the opponent to complete their move.

The UI thread is specifically responsible for the following functionality:

1. Showing the initial display (i.e., an empty 3-by-3 board).
2. Updating the TMM board after latest move by one of the worker threads.
3. Displaying a button to start the game. Pressing this button while a game is in progress will void the current game and start a new game from scratch.
4. Receiving notifications of moves by the worker threads.
5. Informing each worker thread of its opponent's move. This will also signal the thread that it should make the next move.
6. Checking on the status of the game, by determining whether one player has won, the game has ended in a tie, or the game needs to continue.
7. Displaying an appropriate message to indicate the outcome of each game.
8. Signaling the two worker threads that the game is over; the two threads should stop their execution as a result of this action.

Implementation constraints. Your project must comply with the following requirements.

1. Use handlers to implement the communication between the three threads involved. Each thread must have a handler, a job queue and a looper.
2. When coding handlers, you must use at least one runnable and one message in your code.
3. The two worker threads must use different strategies for winning the game.
4. Make sure that the game is played at such a speed that a human user can clearly see and understand the move of each player.

You must work alone on this project. For this project use a Pixel 5 device running the usual Android platform (API 33— Android 13). You are not required to provide backward compatibility with previous Android versions. Submit one Studio project as a zip archive using the submission link in the assignment's page on Blackboard. No late submissions will be accepted.