Programming Assignment 3 – Two-Player Game (w/ Bluetooth)

Assigned/Due: See online for exact due date (11:59pm)

YOU MAY WORK WITH OR WRITE CODE TOGETHER WITH YOUR PARTNER ON THIS PROJECT.

Functional Requirements

In this project, you will push the boundaries of your knowledge to add new features based on some functionality we haven't quite covered in class (but should be within reach with a bit of research).

The below requirements must be met to receive full credit (see **Grading Breakdown** section at the end), but you are given considerable freedom and flexibility to implement these features however you see fit; that is, you don't need to write code in a certain way and are generally free to design your screens as you feel best, with minimal exceptions.

Create a multi-device solution that allows you to play a two-player game by communicating game information over Bluetooth. Ensure your solution has ALL of the following:

a) **Gameplay over Two Devices**: Use two M5Core2 devices or an M5Core2 and Mobile Phone device to play the game.
b) **Communicates over Bluetooth**: Use one or more Bluetooth Low Energy (BLE) service characteristics to communicate data:
   i) Data must be sent from device 1 (D1) to device 2 (D2) via Bluetooth
   ii) Data must be sent from device 2 (D2) to device 1 (D1) via Bluetooth
c) **Synced Game State**: Data must be synced across D1 and D2
   i) Screens should show synced game state data in near real-time during game play
      (1) i.e., there should be no noticeable/significant lag between the states/displays of D1 and D2
   ii) At "game over", D1 and D2 must display differing game statuses (e.g., "Winner" and "Loser", different scores, etc.)
d) **Touch Screen**: Game utilizes touch screen on both D1 and D2
e) **Overall Challenge & Implementation Quality**: The overall challenge and implementation quality of your game/solution will be graded (subjectively by the instructor); for example, the following may be considered:
   i) How simple/complex your game state data (being sent over Bluetooth) is
   ii) How simple/complex/clean your game visualizations are
   iii) How smooth your game play is (with respect to lag and responsiveness)
   iv) How in-sync your two devices are (is data being communicated reliably or do the two devices seem to be displaying two different game states)
   v) How flawless your execution is (are there errors evident in the solution)
   vi) etc.

Grading Breakdown

Grading will be broken down as follows:

- **10% - Turned in all source code**
- **50% - Video demonstration**
    - **20% - Hardware Usage**
        - 10% - Device utilizes touch screen on both D1 and D2 for user input
        - 10% - 2-Way BLE Communication
            - 5% - Clear evidence of D1 sending data to D2 via BLE
            - 5% - Clear evidence of D2 sending data to D1 via BLE
    - **20% - Synced Game State**
        - 10% - Game state is synced in near instantaneous time (no noticeable lag) from:
            - 5% - D1 to D2
            - 5% - D2 to D1
            - i.e., delays that CAN and SHOULD be minimized will be penalized
        - 10% - Game displays must show final game results from the perspective of each user's device:
            - 5% - D1 shows clean summary/result of game over (e.g., "You Win")
            - 5% - D2 shows clean summary/result of game over (e.g., "You Lose")
    - **10% - Showed and discussed key code portions, explaining how you implemented each of the functional requirements**
- **40% - Overall Challenge & Implementation Quality**
    - **20% - Level of CHALLENGE of implementation**
        - 20% - Excellent Challenge
        - 14% - Good Challenge
        - 8% - Average Challenge
        - 0% - Minimal/No Challenge
    - **20% - Level of EXECUTION of implementation**
        - 20% - Flawless Execution
        - 14% - Near flawless Execution (key functionality worked, but with small non-lag-related errors)
        - 8% - Manageable Execution (the point general solution is seen, but key functionality did not work)
        - 0% - Execution of solution did not work at all

Submission Instructions

1.) Submit to Blackboard the following:
   a. **Video recording** (preferably YouTube link as a submission comment) demonstrating the functionality of your program and source code explanation:
      i. Make sure to cover ALL of the areas receiving points in the Grading Breakdown section above
      ii. Your video should be a 1-3 minute demo with 3-5 minutes of code highlights (no need to go into excessive detail, although you won't be penalized if you do)
      iii. Make sure there is sound
      iv. Phone recordings are fine, just make sure the video clearly shows the LCD screen and what is happening with button presses
         1. If the instructor cannot see your screen clearly, you will not get any points for demoing features
   b. **Source files**, including:
      i. All **.cpp files**
      ii. All **.h files**
      iii. Your **platform.ini file**