Conclusions and Recommendations

While working on this project, we found that the team aspect worked very well. We were able to get ideas from each other and work on separate functions to get the project finished on time. We also found that creating the screen for the blank board went well. We were able to make it aesthetically pleasing. We were happy to find that, once finished, the buttons worked well, and our code had few errors in compiling.

We struggled to figure out how to make the menu screen and jump back to the menu from the game ending. We also had trouble figuring out how to test for a touch in the correct location on the menu to choose a specific "button". We figured out how to achieve these features by using a "do… while" loop to loop through the menu options until a choice leaves the menu, paired with a switch case of the chosen buttons.

We were initially unsure how to go about placing tokens, but we figured out to use arrays for storing the center coordinates of the slots to set the centers of tokens placed in those slots. We checked what column the player tapped in, then checked that column for the lowest open slot, and drew a circle at the center of that open slot and changed the board array to signify that the slot had become full.

We never figured out how to make sure the slots don't get overwritten when someone taps on a full column, and we never figured out how to prevent accidental double taps from stealing the opposing player's turn.

We also struggled to figure out how to check if a player won. We knew that we needed to check every possible array combination that would yield a result of four in a row. We knew that we could hard-code every combination into a function, but knew this would yield a clunky, and long result. Instead we decided to use "for" loops and "if" statements to loop through every board index to check if each row, column, and diagonal - both on the main and minor - had four in a row. This looping was done for both the red and blue players to check if either player had won after a turn. Additionally, a tie had to be checked, so we used a nested for loop to check every array index to see if there were any empty slots; if there weren't the game would register a tie.

In spite of our success with figuring out the major game components and aesthetics, we still would have liked to make the game better. We recommend future game developers make it possible for players to tap through the credits instead of making them wait through the credits. We also wish we had set tapping in the instructions to jump back to the menu instead of starting the game. If we had more time and a lot more experience, we might have made a one player option that would pit the player against the computer. We also would have liked to make the four in a row set of tokens light up in rainbow succession to make winning extra special. The game would be nicer if developers figured out how to make sure the slots don't get overwritten when someone taps on a full column, and if they prevent accidental double taps from stealing the opposing player's turn. Sound effects would have been nice to add too, but we didn't think of that. Overall, we recommend this game for helping budding developers learn.