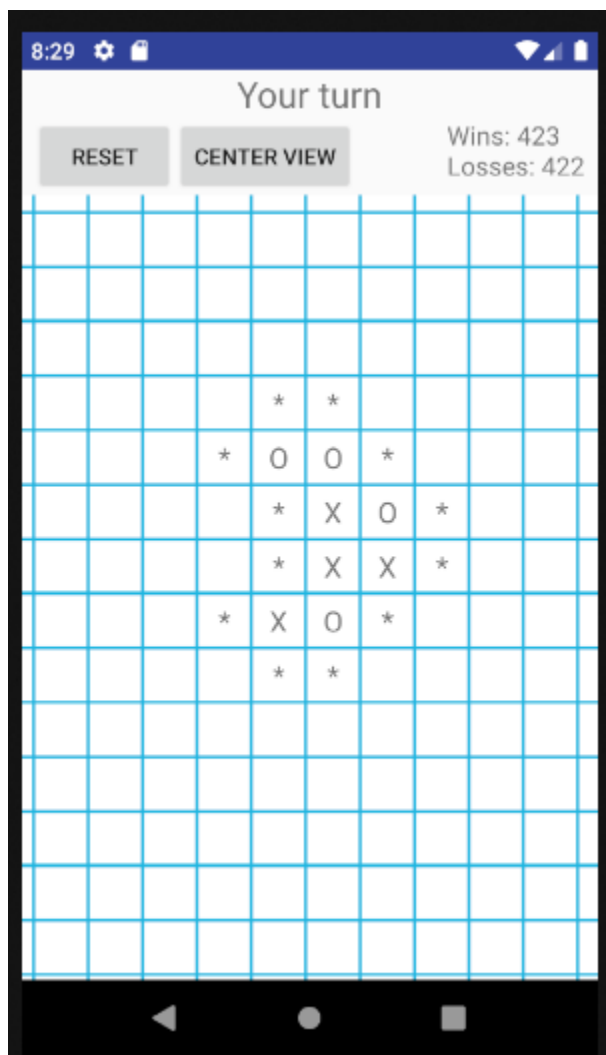


1. Project Statement

Abra is a board game. This game is the reason why I started coding. I've made this game as a website but it has always been my goal to develop an app for it. This particular game makes much more sense as a mobile app than as a browser game and I think it has some potential to actually have real players enjoying the game. Though this game looks like Tic Tac Toe, I would argue that its complexity is much closer to checkers. Checkers is a great game because children can understand the rules but advanced players can perfect their strategy to dominate the game. Abra is similar in this way. The rules are very simple but the gameplay can get quite complex. Anyone can download this game off the app store (once it's made available to the public) and quickly start playing. Though I have not added online features, that would of course require an internet connection.



2. Application Design

The user opens the app and is sent right into the action. They can start playing immediately. Here is a components list:

I. Announcements: there is one text box at the top of the screen which displays all the announcements. The first announcement is instructions that tell the user to "connect five in a row". Once the game begins, the announcement box will indicate who's turn it is. In this image, it is telling the user that it is "your turn". When someone wins, there will be an announcement that states the winner.

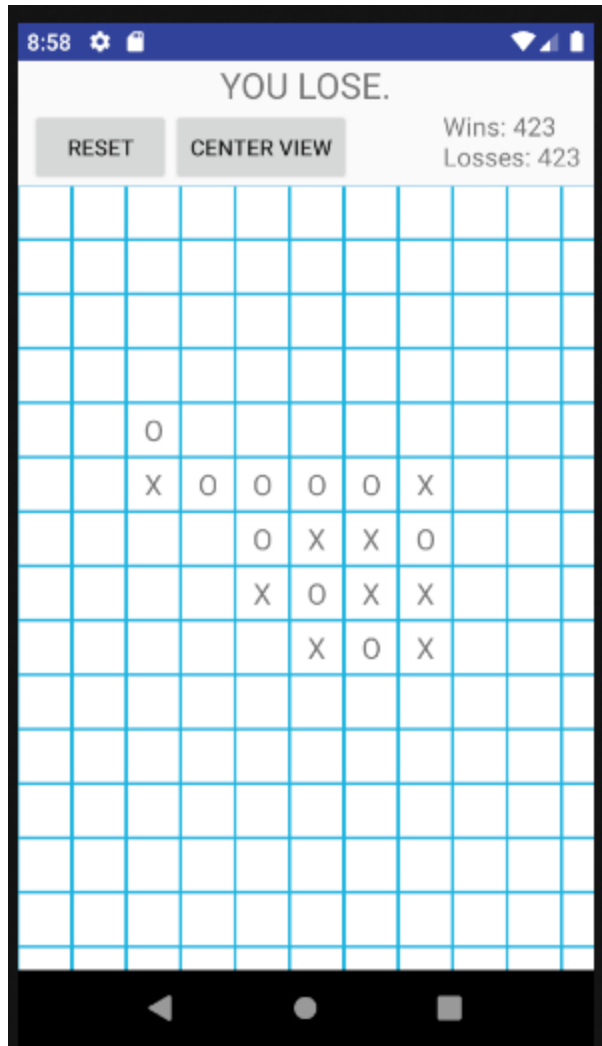
II. Reset: the reset button ends the current game and resets everything back to the

beginning (except for the wins and losses). It will also center the view.

III. Center view: this button sends the view to the center of the board. This is useful because the board can be much bigger than the actual screen. The board expands infinitely until someone loses, so it is easy for the user to get lost. Unfortunately, the center view button only works for the x axis. I ran out of time when trying to figure out how to scroll the y axis.

IV. Wins and losses: The wins and losses text box on the right side of the screen shows the total number of wins and losses. These values are stored in a database.

V. Game board: The game board is designed using a GridView. It looks very similar to homework 2. Since it is too large for the phone screen, it is inside a HorizontalScrollView, allowing the user to scroll vertically and horizontally throughout the board. The board will grow as the players put pieces closer and closer to the edges of the board. Since it keeps growing, it is an infinite board and there is no limit to how many pieces the players can place. There are 4 types of tiles: user, opponent, available and empty tiles. The user tiles are indicated with a "X" and the opponent tiles are a "O". Available tiles are tiles that the players may place a piece on and they are indicated with a "*". Empty tiles are empty and they are completely off limits to both players. This design is pretty intuitive but it's not ideal. I would prefer to use colors instead of symbols to make things easier to understand. Also, I think empty tiles should be invisible. There's no reason to show a bunch of empty tiles that are off limits on the screen. But I did not manage to get that implemented.



Here is another state of the board. Notice that the announcements text box says “you lose”. That means the computer has beaten me. The computer managed to get a diagonal 5 in a row starting from the top left to the bottom right. You can also see that all the available tiles have now been converted to empty tiles, indicating that there are no possible moves because the game is over. In the top right, you can see that my number of losses has been incremented and I now have exactly the same amount of wins and losses, showing that I am equally as skillful as the computer. The next step is to press “reset” and start a new game.

3. Application Implementation and Evaluation

I used the homework 2 handout as a starting point for my project because my game also uses a grid. This may have been a bit of a mistake because using GridView was difficult and I felt like I lacked control over it. I needed a huge scrollable grid that can increase in size. GridView is not designed for this purpose but it worked well enough to get the project done. I had to put it inside a HorizontalScrollView to allow it to overflow on the x axis.

Once I got the basic functionality working I had to create the code for the computer opponent. It was difficult to write this code and there were a ton of errors. I managed to finish it and it is quite difficult to beat. I win about 50% of the time against it. It seems to work perfectly except that sometimes it randomly misses some very obvious moves. Usually, it’s impossible for

the computer to miss a move but it seems to happen randomly and I'm unable to replicate this error on purpose. I'm sure there is an issue in my logic somewhere, I just do not have time to refine my code. I tested the computer by just playing against it myself. It is a bit wasteful to do this type of manual testing but I don't think there's a better way to test a game that hasn't been invented yet. If I was creating a computer that plays chess, I could test it against existing chess engines. But that was not an option for Abra. I used a thread to run my computer algorithm for efficiency and so that I could update the view while the algorithm was running. I used a delay to slow down the algorithm to give the effect of "thinking" for the computer. If it hadn't been for homework 2, I think I would have wasted a lot of time figuring out why my view wasn't updating while my computer algorithm was running.

I used a simple SQL database to store the users wins and losses. My SQL table looks a bit strange because I based it on our to do list app but I only needed to store 2 ints. But it works and efficiency is not much of a concern because the database isn't used often. The database is the only other class I used besides my main class. In hindsight, I should have modularized my code more by using more classes.

There are still errors in my app but nothing major. "Center view" only centers the x axis and sometimes the computer blunders obvious moves. I was unable to get the board to center the view on the initial load. Since the grid view hasn't updated yet, I can't use it to scroll the x axis. I wish there was a callback function I could use for after the GridView updates. If you open the app, I recommend that the first thing you do is click the center view button. This won't center the y axis but it will work on the x axis and will make it easier to figure out what's going on. The last issue is that the board will sometimes suddenly move. This happens when the user or the computer places a piece that's close to the edge. The program detects this and automatically adds more tiles to the grid so that you can never actually hit the edge. The act of adding more tiles to the top or to the left of the grid causes the board to shift and it can be quite disrupting to the gameplay. In testing, I was able to figure out how to scroll the x axis to account for tiles added to the left side of the board but I couldn't get it quite right. If I had more time I could balance the x and y axis so that the user wouldn't notice when new tiles are added.

4. References

No references.

5. Experiences and Thoughts

I was not able to implement player vs player offline or online mode. It wouldn't take long to implement the offline mode but I didn't have enough time. In fact, it would have been easier to build player vs player than player vs computer but I think it was necessary to build the computer so that the app could be properly tested by a single person. I was able to complete all the core functionality for my game but the gameplay doesn't have a pleasant feel to it. The movements are clunky and the visuals aren't clear. There's not even a sound that plays when the opponent moves. So there's still a lot to be done to make the gameplay more enjoyable. I also would have liked to include some sort of help button or instructions for the user. I hope it is intuitive enough for players to figure out how to play after a few clicks. For your reference, all you need to do is connect 5 pieces in a row before your opponent does and you can only place your pieces on tiles that have an asterisk (*).

I am pretty experienced with website design but doing design on android was completely new to me. The xml looks somewhat similar to html but a lot of things are different. We didn't learn much about design in this class. I think that's a good thing, though. I don't want to waste time learning to design when I could be learning about the actual functionality of apps. I pretty much built this app using only what I learned in the homeworks and also from the to do list app, with some googling to help myself bring it all together. But, it would be nice to have made a better design for this project.