Jacob Lemon

Daniel Williamson

CS 136L

Lab 10: Grid Games

**Understanding the Problem:**

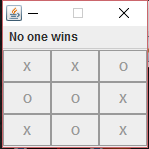
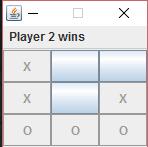
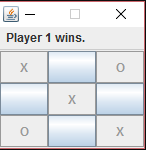
For this lab our goal was to create a GUI program that created a small suite of games along with a graphical interface for them. We will be utilizing polymorphism, interfaces and inheritance to reduce coupling in our program. We will also be implementing the Java Swing library for our GUI. Essentially, we are making a program that lets a user select between multiple grid games. Once a games is selected, it will begin the selected game and allow the user to interact with it in a GUI.

**Planning:**

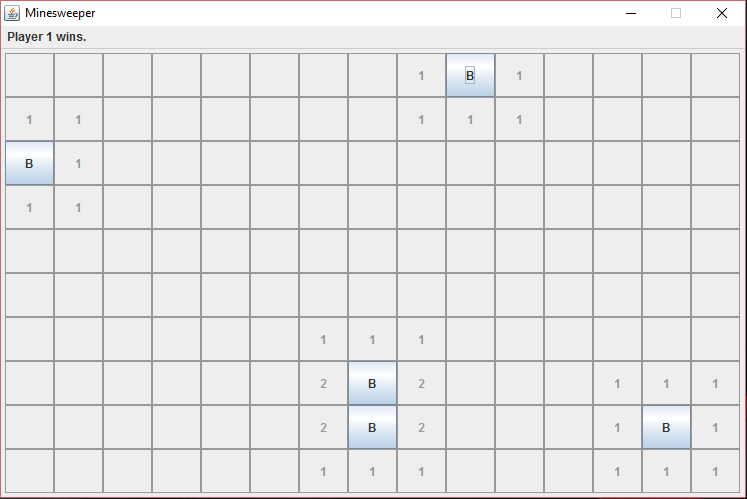
The planning for this plan was rather typical. We started by discussing different ideas on how to tackle this lab. After our discussion we decided upon beginning with creating the first prompt that asks the user what game s/he would like to start. Once we had that working correctly and the buttons didn’t do anything, yet, we went ahead and implemented TicTacToe which was given to us by our instructor. After much confusion and research on how an interface works, we inevitably figured out how to run TicTacToe. Next, we started working on the hardest part of the lab, getting the other games to work, correctly. The logic behind getting Minesweeper to randomly place bombs and when a selected square has no bombs near it, it would reveal the rest of the board was rather tough. Eventually, we got Minesweeper to function correctly and make it so it can be open with our first user interface. Lastly, we started on Connect Four, it took much trial and error. The hardest part being able to make the pieces drop to the bottom of the board. Which was solved with a simple while loop. And once again, we made it so it can be selected in our first user interface.

**Implementation:**

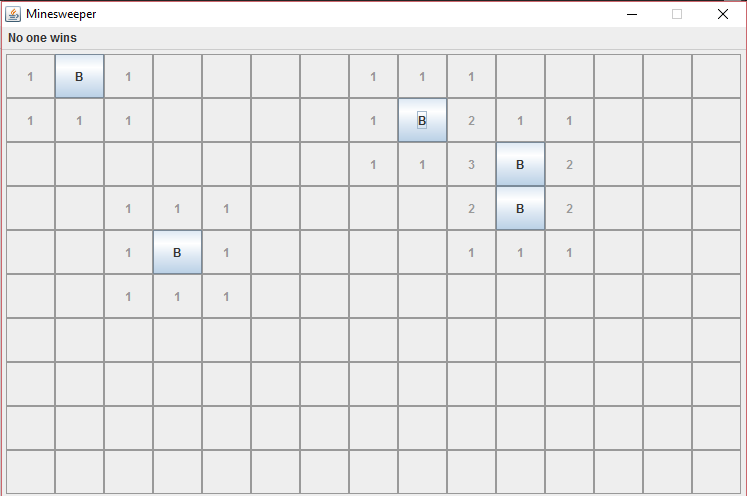
Here is a depiction of player 1 winning diagonally, player 2 winning horizontally and no one winning in TicTacToe:



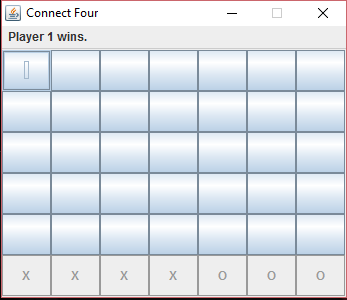
Here is a depiction of a player winning at Minesweeper. Note that all unnecessary tiles were cleared away if there wasn’t a bomb nearby.



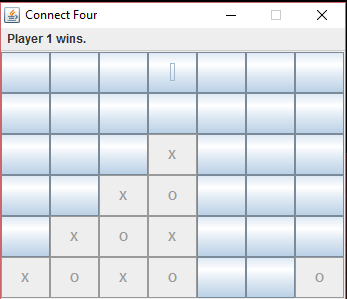
Here is a depiction of a player losing at Minesweeper by clicking on a bomb. Note that all unnecessary tiles were cleared away if there wasn’t a bomb nearby.



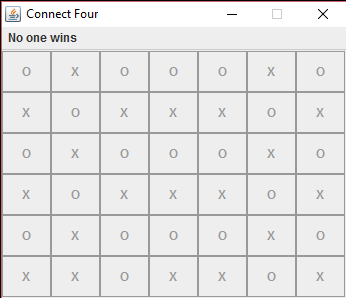
Here is a depiction of player 1 winning at connect four horizontally. Also note that all tokens were “dropped” from the top and fell to the bottom of the board.



Here is a depiction of player 1 winning at connect four diagonally. Also note that all tokens were “dropped” from the top and fell to the bottom of the board.



Here is a depiction of no one winning at connect four. Also note that all tokens were “dropped” from the top and fell to the bottom of the board.



**Conclusion:**

In conclusion this was a rather difficult lab. The most difficult parts, previously stated, were getting Minesweeper to randomly place bombs throughout the grid. Once we finally understood working with a 2D array it became more easier. We just didn’t think to use one at first. Another difficulty in Minesweeper was getting the field to open up if there were no bombs nearby. Which was solved with a for loop. And lastly, a difficult part was getting Connect Four to drop the tokens down to the button. This problem drove us insane and we finally solved with a simple while loop. Overall I think we did really well. We did not implement any of the colors, purely because we couldn’t figure out how.