CS 333 – Project 1

Kaan Alp S018037

For the beginning of the project, I decided to start by first creating the game and making sure that 2 person play works correctly. In the first implementation of the game, I tried to use a one dimensional array but decided against it as it made the implementation harder. I decided to create the game in a different class and implemented all the methods that were needed into the game into there as this would declutter the main class.

I decided to put all the game checks into a single method. After the player or computer makes their move, the check looks at the specified node on the array and checks it horizontally, vertically, and diagonally.

After the game worked, I started on the computer logics implantation.

I wanted the logic to first look at all the board after the players moves. If the player is close to winning, then the logic will focus on stopping the player. If not, then the logic will try to win itself. I implemented this by giving weights to the logics moves. If the player is going to get a 4 in a row in their next move the weight will be higher, if the player is not going to win the next turn and there is a move that will block the player and get the logic closer to winning it will be a 4, If there is no good move the ai will still block the player with weight 3, If the player has at most 2 pucks next to each other the logic will try to win itself up by trying to put a random “O” puck next to one of their own. With these kinds of weights, the computer should choose the most appropriate move.

For the search algorithm I chose breadth first algorithm and start at (0,7) and look at every node it goes to and assign weights to these paths. Then the algorithm parses the next node and looks at its adjacent weight. The algorithm doesn’t need to look everywhere as pucks can only put on the bottom or on top of a different puck so the only nodes that should be parsed are the ones with these criteria. When the processes complete the column with the highest weight is played then rinse and repeat until a winner can be applied with the conditions set in the game.