Zach Forster

CS 335

Game Tree

This assignment was fairly close to the convex hull assignment in difficulty, making it the second hardest assignment we’ve had this semester. The issue was judging the correctness of the tic-tac-toe AI when it was playing with some apparent purpose, but quite badly. I had a few issues with the actual tic-tac-toe part of the assignment, but that part actually rolled over for me pretty quickly. The MiniMax algorithm was a non-issue, since you gave it to us in the writeup. One of my issues was the fact that I wasn’t updating my nextOpenPosition variable after the user moved (only after the computer moved). The other was that I had no idea how to interact with the GUI code. That was much easier after I figured out that you use the Graphics object to draw, and that you don’t call paintComponent() directly. Luckily, I was able to get most of my code working without the GUI by taking your advice about making a test case that utilizes the board’s toString() method.

After finishing the TicTacToeBoard early, I decided to give the ConnectFourBoard a try. At first, I thought that it was going to be incredibly simple. I copied most of the methods from the TicTacToeBoard class directly. I pre-computed a list of all possible winning combinations in connect four. I believe that there were only 64 of them, so the check to see if the game is over was very speedy. With full confidence, I ran my code, only to realize that when you drop a disc into a connect four game, it can’t stop and hover at any hole you want it to. For some reason, it was quite the struggle implementing the logic to find the next open space (probably because I was writing the code past 4:00 a.m.). Anyway, I’m pretty sure that everything works, since I can’t beat the tic-tac-toe AI on hard mode, and I struggle to beat the connect four AI (it happens once in a while). This assignment was really fun! +1