Brian Craft

AI 1: CSC 480

Assignment 2

Github: <https://github.com/priorfire4411/Craft-Repo/blob/master/artificial_intelligence/Connect_Four_Game/somacgo_agent.py>

Youtube: <https://youtu.be/r7gIx-wM-ts>

1. What number of plys makes the game easy? Challenging? Too challenging? Explain how this would impact a consumer experience in the game?

The game was easy to beat up until ply 8, in which I would need to be careful with my moves in order to win and stay competitive. At this point, I would say the game began to get challenging. Anything after ply 8 became very challenging and almost unbeatable.

From a consumer perspective, there needs to be an AI that caters to those that want a challenge but also those learning the game that need less of a challenge and can move up. The best example of this is guitar hero. Not everyone can instantly play on the expert level, but more seasoned players aren’t going to enjoy playing on easier modes. So there is multiple difficulty levels allowing a person to progressively get better at the game. In other games, this succession is done as the player progress levels, which each level getting more challenging.

1. What number of plys makes the game run too slow? Is there a tradeoff between the game running too slow and the game being too easy?

I was using my work computer which isn’t super powerful, and I was having runtime issues at ply 9, in which it took over 30 seconds to make a move. The tradeoff is simple better runtimes mean lesser intelligent AI while longer runtimes mean a smarter AI. For me, I found ply 6 to be fun, as it ran fast, I didn’t have to think much for each move and the AI was competitive but not to the point I had to analyze every move in depth.

1. How much deeper can the game explore? Does this alter the answers to question 1 and 2?

I could run to ply 10, but with very slow runtimes. Given this, if I were to create a video game, I may allow users to set the difficult on a scale of 1-10. However, I would need to find a way to crate stable runtimes across the difficult levels, which is discussed in question 2. However, this doesn’t make me alter my answers, as I was already noticing the runtime issues and the possible tradeoff.

One thing I do question is the heuristic and if it could be smarter. I feel there could be a mix between blocking and accumulating points. In some cases, my AI would stack pieces, which is great for the score, but long term, still allows the human to consolidate space. Given the goal is to consolidate space, I’m wondering if there is a better heuristic to capture this then simply the score.