

CISC 489 Program 2

Program 1, but now we're competing against other students in the class.

Detailed below is my thought process in solving this problem.

Test Cases

I ran some demo tests to get a baseline for what's happening in this environment.

Test Case	Description	Player A Score	Player B Score
case = 0	Player A controllable (no inputs), Player B random	0	20
case = 0 *	Player A controllable (by me), Player B random	65	20
case = 1	Two Random Agents	-47	-77

Theory

My agents are going to very greedy, and walk right up to the edge of colliding with each other. Both of my agents will just head for the nearest coin (weighted on value). To correct some errors that I had with Programming Assignment 1, I will institute the general strategy:

1. Find the nearest coin weighted on value.
2. Find a path to that coin (ignoring the position of the other agent), and only reconsider if the coin has disappeared.
3. If immediately adjacent to another agent, cancel the current plan (for one action step), going in the other direction.

I believe this should result in a nice aggressive agent that does not go too far.

Development

I am copying over all of my old code from Programming Assignment 1, because I will need most of it. I am not going to use my heatmap code, because there's just not enough range in the variables (coin values only 1 through 9) to be usable.

I had to include a `requirements.txt` for the `pygame` import, and `numpy` , as I added `numpy` to handle a few expensive functions.

I realized pretty quickly that I was unable to extract the other player's position. I went into debug mode, and I searched through all of the variables in scope, but there was no way that I could tell what the position of the other player is. And also, with how late this project is, I think its better to submit now rather than spend more time on it than I already have.

My Test Results

Player 1 Description	Player 1 Score	Player 2 Description	Player 2 Score
Random (case = 2)	42	My Player A: Implementing my theory	60
My Player A (case = 3)	26	My Player B (case = 3)	28

My score is definitely worse than what it was in Programming Assignment 1, but I've traded that for reliability.

Conclusion

My agents will always keep moving forward, and they now reconsider their plans as they go about them. They will never stop moving. They may waste one turn as they consider a new path, but that's alright with me.

Really now what I realize is I should have also considered the lifetime of the coin, and if a coin was not reachable by the minimum distance it would take to reach it, then do not even consider it.

But I am out of time with this project, and with this being the last one of my college career, I'm satisfied enough it.

Thank you for a great year, and have a good day, — James