CS 3600 Project 2 Wrapper

CS 3600 - Fall 2022

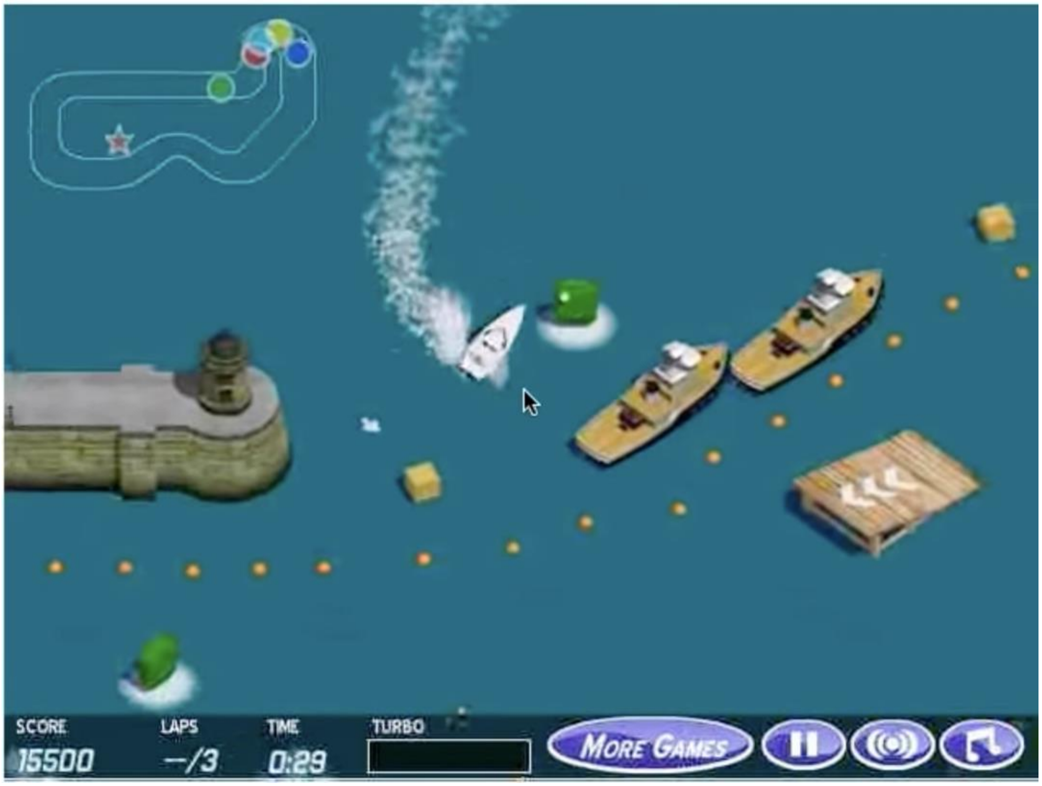
Due October 19th 2022 at 11:59pm EST via Gradescope

Introduction

This Project Wrapper is composed of 4 questions, each worth 1 point. Please limit your responses to a maximum of 200 words. The focus of this assignment is to train your ability to reason through the consequences and ethical implications of computational intelligence, therefore do not focus on getting ”the right answer”, but rather on demonstrating that you are able to consider the impacts of your designs.

Context

Reinforcement learning is a powerful technique for problem-solving in environments with stochastic actions. As with any Markov Decision Process, the reward function dictates what is considered optimal behavior by an agent. Since a reinforcement learning agent is trying to find a policy that maximizes expected future reward, changing when and how much reward the agent gets changes its policy.



However, if the reward function is not specified correctly (meaning rewards are not given for the appropriate actions in the appropriate states) the agent's behavior can differ from what is intended by the AI designer. Consider the boat racing game pictured above. The goal, as understood by people, is to quickly finish the race. Humans have no difficulty playing the game and driving the boat to the end of the course. However, when a reinforcement learning agent learns how to play the game, it never completes the course. In fact, it finds a spot and goes in circles until time runs out. You can see the RL agent in action in this video: https://youtu.be/tlOIHko8ySg.The agent's reward function is the score the player receives while playing the game. Score is given for collecting power-ups and doing tricks, but no points are given to players for completing the course.

Question 1

Watch the video and explain why the agent’s policy has learned this circling behavior instead of progressing to the end of the course like we expect from a human player. Explain the behavior in terms of utility and reward.

Answer:

As mentioned above, there is not reward or points given to players for completing the course itself, so the utility associated with this goal state is probably lower than many other states. We can see in the video, the AI begins driving the boat in circles to collect three turbo power-ups in a row repeatedly and this results in a very large reward cycle. I believe that in the training cycles of the AI, it must have reinforced this cycle since it is able to collect points almost infinitely since there is no count down.

If there is a reward for completing the course in a faster time, it is probably the case that the AI was never able to explore that case since that goal state is so far away and the reward associated with the quicker time is probably still less that constantly getting points for collecting the power-ups.

Question 2

When humans play, the rules for scoring are the same. Why do humans play differently then, always completing the course? Why don't humans circle in the same spot in the course endlessly if they are receiving the same score feedback as the agent?

Answer:

This is because humans can see the bigger picture of the map wen playing the game, and due to intuition and prior knowledge, we understand that the goal of the game is actually to complete the course as fast as possible rather than to get the highest score.

I assume for people who figure out that spamming the powerups as the AI did yields such a large point scoring, they might do that as well.

The AI yields its given policy since it explores the map with an Epsilon-Greedy algorithm, so it either explores randomly or it reinforces prior success states. This causes loops such as the one we see to thrive with time.

Question 3

The agent's original reward function is:

$$R(s_t,a) = game\ score(s_t) - game\ \_score(s_{t-1})$$

Describe in terms of utility, reward, and score **two** ways one could modify the reward function to get the agent to behave more like a human player. That is, what do we need to change to make the agent complete the course every single time? Assume the agent has access to state information such as the position and speed of the boat and all rival racers, but we cannot change how the game itself provides scores through the call $game\ score(s_t)$.

Answer:

The first way I would modify the reward function could be to consider:

$$R(s_t,a) = (position(s_{t-1}) - position(s_t)) * 5000$$

This incentivizes gaining/ overtaking other rival racers while also punishing losing positions in the race. This would hopefully result in a utility that would gain as many positions at the beginning and then avoid losing positions after that, allowing the AI to win the race

The second way I would modify the reward function would be to consider:

$$R(s_t,a) = game\ score(s_t) - game\ \_score(s_{t-1}) - \ speed(s_t) - speed(s_{t-1})$$

This incentivizes getting the highest score possible whilst also maintaining the highest speed possible. This hopefully would prevent the AI from forming loops again and instead find a fast path with a lot of reward on the way too.

I think a certain combination of all three reward functions would results in the best acting AI that acts more like a human player.

Question 4

Self-driving cars do not use reinforcement learning for a variety of reasons, including the difficulty of teaching RL agents in the real world, and the dangers of a taxi accidentally learning undesired policies as we saw with the boat game example. Suppose however, that you tried to make a reinforcement learning agent that drove a taxi. The agent is given reward based on how much fare is paid for the ride, including tips given by the passenger. Describe a scenario in which, after the taxi agent has learned a policy, the autonomous car might choose to do an action that puts either the rider, pedestrians, or other drivers in danger.

Answer:

There could be a scenario where the Taxi threatens to either not end the ride or potentially hurt other pedestrians or other drivers unless the passenger doesn't agree to leave a tip above a certain threshold.

Another scenario might be that the Taxi agent realizes that faster rides allow for more tips and a greater reward per unit time, so it begins attempting to take shortcuts or practices unsafe driving that puts the passenger and those around them at danger.