CS 3600 Project 2 Wrapper

CS 3600 - Fall 2023

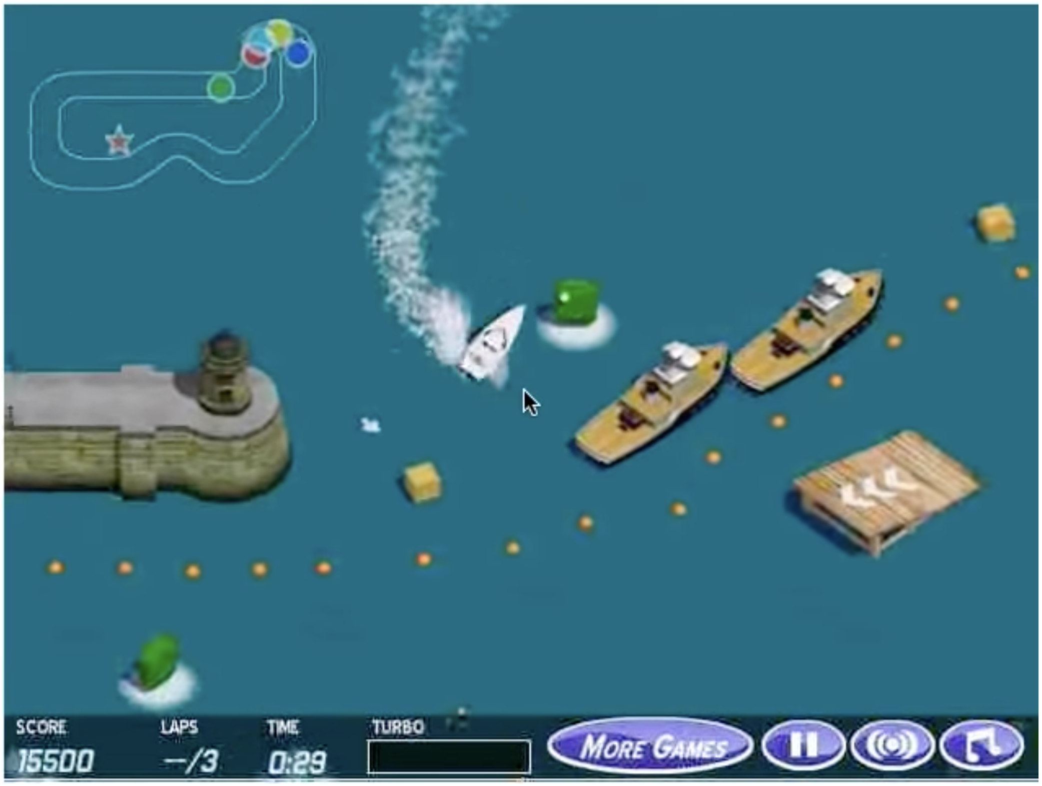
Due October 13th 2023 at 11:59pm EST via Gradescope

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:

This behaviour is attributed to the RL agent reward function and how it derives utility. As there is points for tricks and movement. However , there is no reward to make it to the end of the course that makes it appealing for the agent. Moreover, RL agents seeks to maximize thier reward (or Utility) overtime. On the other hand, Humans inherently understand the main goal of a ricing game (finishing it quickly). Due to the reward given to our agent, the agent perceives higher immediate utility in circling and accumulating points with frequent power-ups, rather than racing towards the finish line without any immediate reward. Hence, the learned policy have a looping tendency which gives optimal utility and reward.

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:

Humans are distingueshed with an inherent motivation and prior knowledge. As human would often do action in-game and in-real-life for pure fun and not to optimaly maximize thier reward but to gather satisfaction and sense of accomplishment. Additionally, they have prior knowledge that the main objective of a racing game is to finish it and reach to the main goal as quickly as possible regardless of reward. This intrinsic motivation pushes humans to complete the course rather than circle endlessly. Additionally, humans have a finite perspective of time such that endlessly moving in circles is a waste. n contrast, an AI agent doesn't have this perception of time and will happily repeat an action indefinitely if it maximizes the given reward.

Question 3

The agent's original reward function is:

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

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 $\mathit{game\_score}(s_t)$.

Answer:

Inorder for our RL agent to behave more like a human, our first method is to adjust the reward function for a position-based reward. Furthermore, our reward function will reward user for getting closer to the objective (ending the course). Consequently our reward function will beR(st,a)=game score(st) - game score(st1) + lambda * (position(st) - position(st1)). Where if we get a positive postion means that our agent approached forward movement. and where lambda indicated our scaled reward for RL agent.

Another approach to modify the reward function is based on the info of rival racers where we will give reward on overtaking rivals, or maintaining position. Furthermore, R(st,a)=game score(st) - game score (st1) + K × number of racers overtaken. Where K is our sclaed reward for incentive reward on overtaking higher positions. This approach uses relative distance to guide our RL agent.

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:

based on the given policy, our agent will be optimally increasing the fare and tips given by passenger. Subsequently, imagine a scenario where our rider is late to an interview. Therefore, our tip is maximized by following a quickler, more dangerous route in a crowded marketplace. Consequently, this behaviour leads to increased speed in narrow lanes. Moreover, ignoring padestrian crosswalks for maximum speed which puts them in risk. Additionally, taking shortcuts even if it meant driving on the wrong side of the lane. Hence, our RL agent in persuit of optimized tip and reward will make dangerous decisions because of reward policy that have no regard for safety constraints.