

Robert Sanchez

Jonathan Schultz

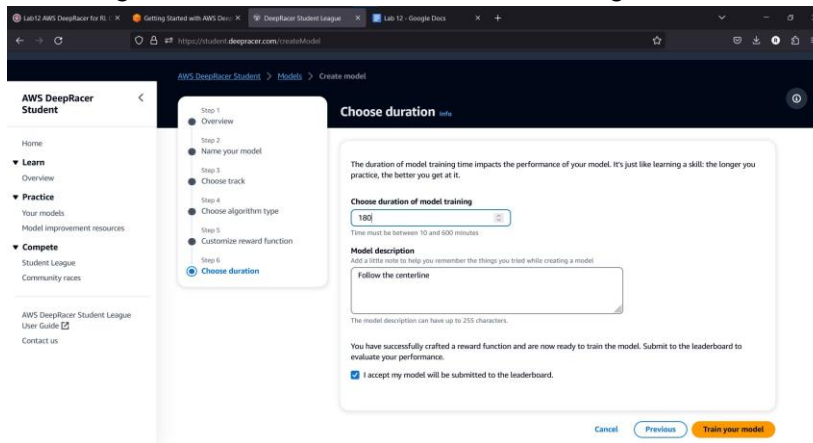
IS 160

Dr. Stephen Choi

Lab 12 AWS DeepRacer for RL Concept

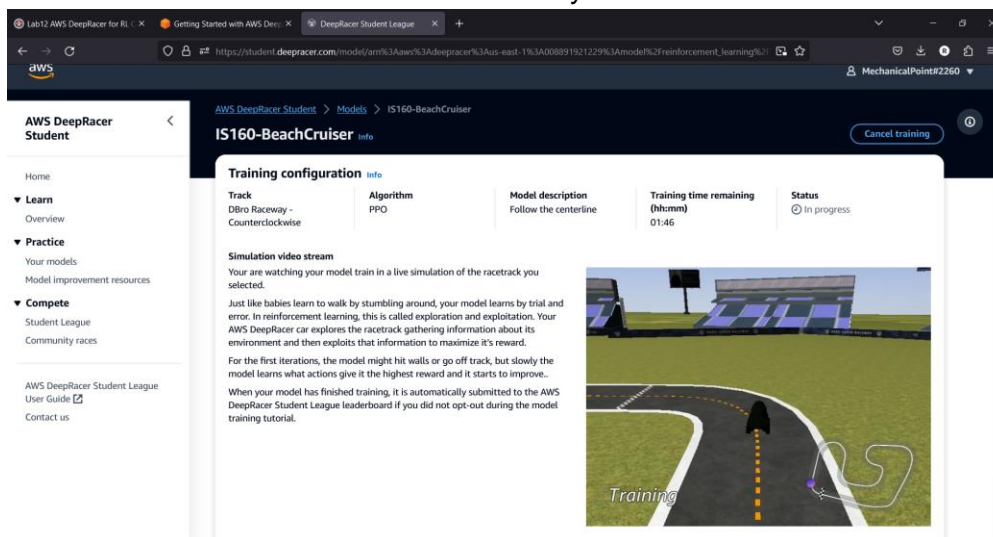
Model being created

Choosing the name and amount of time training the model



Model being trained

Trained model for 3 hours on Dbro Raceway



Model Parameters selected

Increased speed threshold to 3 from a maximum of 1 while also increasing the steering angle

Model competing in online race

Our model participating in the race on Dbro Raceway

IS160-BeachCruiser evaluation video



Second race on A to Z raceway



Race Overview of Model

▼ Race overview


Race type
Time trial

Description
Don't wait until the 2024 season starts to begin honing your DeepRacer skills, the preseason practice session is open until end of February to provide a way to practice on track and gear up for next year.


Race dates
Start September 30 at 5:00 PM
End February 29 at 3:59 PM

Time zone
UTC-0800 (Pacific Standard Time) America/Los_Angeles

Competition track
Get ready to burn rubber on the DBro Raceway named after 2021 AWS DeepRacer finalist Darren "DBro" Broderick. This track features 3 lightning fast drag strips into sweeping turns. Do you have what it takes to advance or will get caught doing DOH-nuts in the green?
Direction: Counterclockwise



Rules
Ranking method
Total time
Individual lap
Style
3 consecutive laps
Entry criteria
3 consecutive resets
Resets
Unlimited resets
Off-track penalty
3 seconds
Submission quota
250



IS160-Lab12


Your fastest time
02:23.131

Fastest model submitted
IS160-bugatti

Remaining submissions
246

Racing times and ranking

United States




Fastest time
01:59.742
AnonymousUser#321425

My race ranking
21/155

Gap to 1st
+00:23.389

North America




Fastest time
01:52.930
AdrianAI#9613

My race ranking
32/201

Gap to 1st
+00:30.201

World




Fastest time
01:44.374
OneTrickPony

My race ranking
340/1387

Gap to 1st
+00:38.757

America/Los_Angeles



**MobileFormula
#2919**


Your fastest time
01:17.05
4

Fastest model submitted
M4CSL

Remaining submissions
48

Racing times and ranking

United States




Fastest time
00:21.005
rosscomp1

My race ranking
119/269

Gap to 1st
+00:56.049

North America




Fastest time
00:21.005
rosscomp1

My race ranking
135/294

Gap to 1st
+00:56.049

World



Fastest time
00:21.005
rosscomp1

My race ranking
590/1116

Gap to 1st
+00:56.049

Race leaderboard

Your submissions (2)

AWS Deep Racer Reinforcement Learning Concept

Agent: A small autonomous car designed for reinforcement learning. Operates without human intervention. Adapts behavior through trial and error. Uses sensors to analyze its environment

Action (a): The possible moves/actions that the DeepRacer can take various degrees of accelerating, braking, and steering left or right.

Environment: A simulated racing track provided by AWS DeepRacer, where the car navigates through a series of checkpoints to complete a lap. This environment also has its own simulated physics.

State (s): The current situation returned by the environment could include the position of the car on the track, its speed, and the proximity to the next checkpoint.

Reward: An immediate return sent back from the environment to evaluate the last action by the DeepRacer. For example, the DeepRacer might receive positive rewards for staying on the track, passing through checkpoints, and completing a lap. Negative rewards could be given for going off-track, colliding with obstacles, or taking longer than a certain time to complete a lap. These rewards are given through functions that can be adjusted to encourage certain behaviors.

Markov Decision Process : The decision-making process of the robot can be modeled as a Markov Decision Process. At each state, the robot decides on an action based on its current perception, and the environment responds with a new state and a reward.

Bellman Equation: In the context of DeepRacer, the Bellman equation captures the agent's objective to maximize the cumulative expected reward over time by making optimal decisions in each state. The agent learns from experience, updating its value estimates for each state based on the immediate rewards and the expected future rewards.

In the models and tracks that we selected, the DeepRacer's goal is to learn a strategy that maximizes the cumulative reward over time. Through trial and error, it explores different actions in various states, learns from the rewards or penalties received, and adjusts its strategy accordingly. Parameters can be used to be adjusted to change the driving behavior of our models. We had to adjust these parameters to create a more balanced reward function that would not choose short term goals over the long term goal of completing the lap. A reward function that prefers short term goals over the long term goal would typically have a worse completion rate, meaning the model was being too risky with its driving and not braking in time or failing to properly reach the correct steering angle. We also discovered how important it is to select a proper amount of training time. We selected an algorithm that allows the car to select its own optimal path than rather allowing it to follow a pre drawn line. This resulted in the model to really learn by itself and take a lot longer to find the optimal race lines to maximize speed through the corners, which would provide a massive advantage compared to the other conventional methods. Doing this really gave us a very well optimized model which we were satisfied with.