

[Home](#)[Service](#)[About Us](#)[Contact](#)

DEEP RACER

4109053049 張康磊

4110053037 王盈穎



aws

re:Invent 產品 解決方案 定價 文件 了解 合作夥伴網路 AWS Marketplace 客戶支援 事件 進一步探索 登入主控台

AWS DeepRacer 標觀 聯盟 活動 學生 AI 和 ML 獎學金 賽車技巧 入門 定價 常見問答集 機器人專案

開發人員們，發動您的引擎

無論技能水平如何，開發人員都可以透過以雲端為基礎的 3D 賽車模擬器、由強化學習驅動的 1/18 比例全自動賽車和全球賽車聯盟來親身體驗機器學習。

[建置模型](#) [立即購買](#)



flow chart

▲ 亞馬遜開發之網站，可讓所有開發人員以強化學習模擬自駕賽車
Link:<https://aws.amazon.com/tw/deepracer/>

Get started with reinforcement learning

How AWS DeepRacer works

```
graph LR; A[Learn basics<br/>Learn reinforcement learning basics] --> B[Create model<br/>Choose action space, algorithm, and reward function]; B --> C[Train & evaluate<br/>See your strategy in simulation and your metrics visualized]; C --> D[Join DeepRacer League<br/>Race models in the Virtual Circuit]; D --> E[Model iteration & Upskill<br/>Clone, ideate, and create a winning strategy. Attend workshops to learn advanced RL techniques]; E -- feedback loop --> A
```

Learn basics
Learn reinforcement learning basics

Create model
Choose action space, algorithm, and reward function

Train & evaluate
See your strategy in simulation and your metrics visualized

Join DeepRacer League
Race models in the Virtual Circuit

Model iteration & Upskill
Clone, ideate, and create a winning strategy. Attend workshops to learn advanced RL techniques



ABOUT DEEPRACER

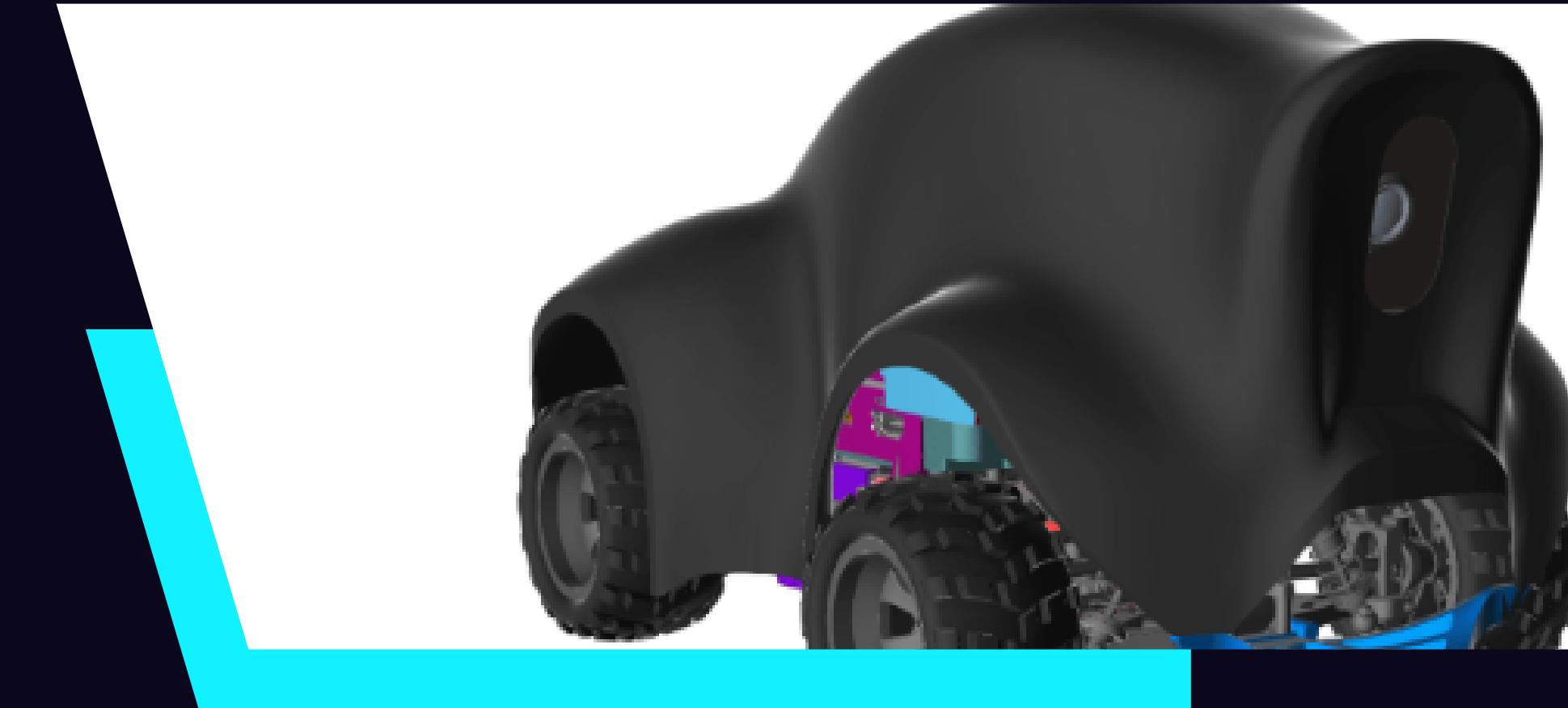
01

A COURSE ON
REINFORCEMENT LEARNING

02

CREATE A MODEL

MODEL NAME AND ENVIRONMENT SPECIFYING
CHOOSE RACING TYPE AND TRAINING ALGORITHM
DEFINITION OF ACTION SPACE
VEHICLE
COSTOMIZE REWARD FUNCTION



A COURSE ON REINFORCEMENT LEARNING

Step 1: Take a crash course on Reinforcement Learning (10min)

Reinforcement Learning (RL) is the Machine Learning technique which drives AWS DeepRacer. Learn the basics of RL to create and optimize your models to compete in the AWS DeepRacer League.

 Start the course

AWS DeepRacer Guide

- Introduction
- What is reinforcement learning?
- ▶ How does AWS DeepRacer learn to drive itself?
- ▶ How to train a reinforcement learning model?
- ▼ Parameters of reward functions
 - Intro**
 - Position on track
 - Heading
 - Waypoints
 - Track Width
 - Distance from center line
 - All wheels on track
 - Speed
 - Steering angle
 - Summary
- ▶ Reward function

Parameters of reward functions

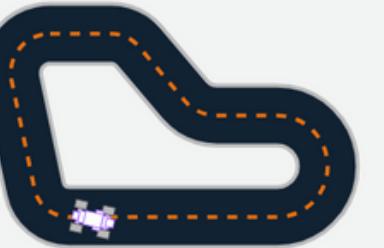
Reward function parameters for AWS DeepRacer

In AWS DeepRacer, the reward function is a Python function which is given certain parameters that describe the current state and returns a numeric reward value.

The parameters passed to the reward function describe various aspects of the state of the vehicle, such as its position and orientation on the track, its observed speed, steering angle and more.

We will explore some of these parameters and how they describe the vehicle as it drives around the track:

- Position on track
- Heading
- Waypoints
- Track width
- Distance from center line
- All wheels on track
- Speed
- Steering angle



Previous **Next**



CREATE A MODEL

Step 2: Create a model

Simply follow the steps in the console to build, train and evaluate your model and enter the AWS DeepRacer League. With [AWS Free Tier](#), you will receive 10 free hours to train or evaluate models and 5GB of free storage during your first month. This is enough to train your first time-trial model, evaluate it, tune it, and then enter it into the AWS DeepRacer League. This offer is valid for 30 days after you have used the service for the first time.

[Create model](#)

MODEL NAME AND ENVIRONMENT SPECIFYING

Training details

Model name

TopModel-re:InventTrack

The model name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and hyphens (-). No spaces or underscores (_).

Training job description - optional

Log details for quick reference

The model description can have up to 255 characters.

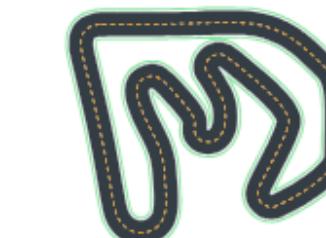
Environment simulation Info

Simulated environment emulates a track to train your model.

Vivalas Speedway

Inspired by a historic Las Vegas track of yesteryear, the Vivalas Speedway is the second longest, and most difficult track to be released in 2021. 5 consecutive opposing hairpins are framed in by the Vivalas Loop perimeter; a modified oval full of high speed straightaways primed for passing and all out speed. Which racers will go all in and gamble it all for the jackpot?

Direction: Counterclockwise

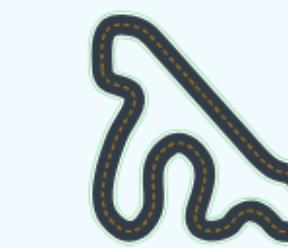


[Virtual circuit](#)

2022 re:Invent Championship

Get ready to rev your engines on the official 2022 re:Invent Championship track! This is an intensely difficult track (35.87 m) featuring a technical chicane section that will challenge even the most skilled developers.

Direction: Clockwise, Counterclockwise



Jennens Super Speedway

The Jennens Super Speedway (62.07 m) is named in honor of the first ever racing family and 2021 re:Invent finalists James "JJ" and Timothy "Flatearth" Jennens. Pros this month will need to navigate the technical section without off tracks, and apply top speeds on the drag strips to climb the leaderboard.

Direction: Clockwise, Counterclockwise



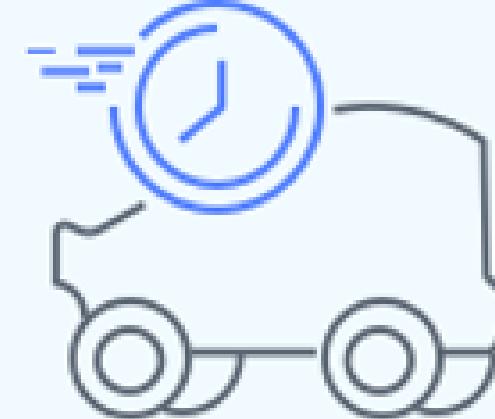
[View more race tracks](#)



CHOOSE RACING TYPE

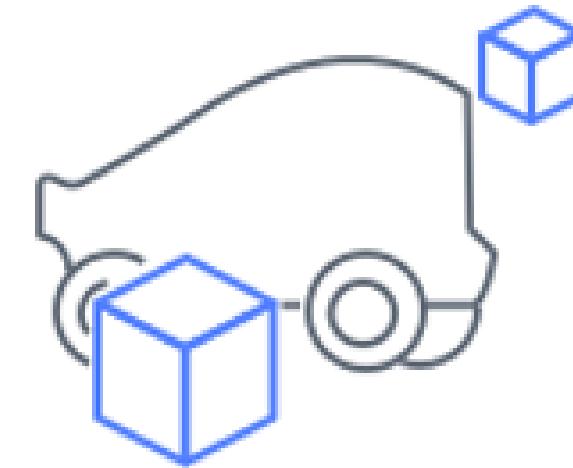
Time trial

The agent races against the clock on a well-marked track without stationary obstacles or moving competitors.



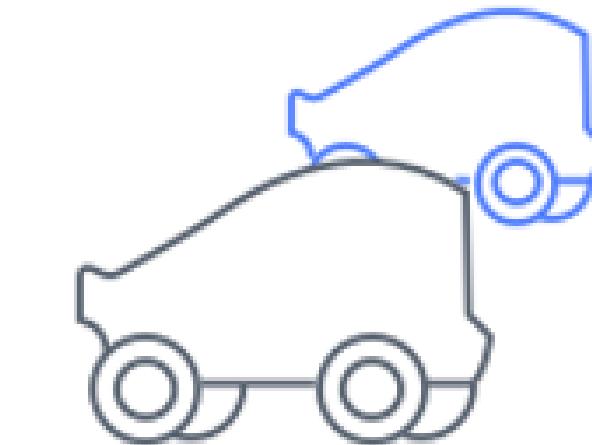
Object avoidance

The vehicle races on a two-lane track with a fixed number of stationary obstacles placed along the track.



Head-to-head racing

The vehicle races against other moving vehicles on a two-lane track.



TRAINING ALGORITHM

Training algorithm and hyperparameters [Info](#)

PPO

A state-of-the-art policy gradient algorithm which uses two neural networks during training – a policy network and a value network.

SAC

Not limiting itself to seeking only the maximum of lifetime rewards, this algorithm embraces exploration, incentivizing entropy in its pursuit of optimal policy.

▼ Hyperparameters

Gradient descent batch size

- 32
- 64
- 128
- 256
- 512

Number of epochs

10

Integer between 3 and 10.

Learning rate

0.0003

Real number between 0.00000001 (1e-8) and 0.001 (1e-3).

Entropy

0.01

Real number between 0 and 1.

Discount factor

0.99

DEFINITION OF ACTION SPACE



Select action space [Info](#)

Action spaces

Continuous action space
A continuous action space allows the agent to select an action from a range of values for each state.

Discrete action space
A discrete action space represents all of the agent's possible actions for each state in a finite set.

Define continuous action space [Info](#)

In a continuous action space setting, the agent learns to pick the optimal speed and steering values from the min/max bounds you provide through training. Providing a range of values for the model to pick from seems to be the better option but the agent has to train longer to learn to choose the optimal actions.

Steering angle

The steering angle determines the range of steering angles in which the front wheels of your agent can turn.

Left steering angle range

degrees

Values are between 0 and 30.

Right steering angle range

degrees

Values are between -30 and 0.

Speed

The speed determines how fast your agent can drive.

Min/max speed defines the range of speeds available to the agent while training.

Minimum speed

m/s

Values are between 0.1 and 4.

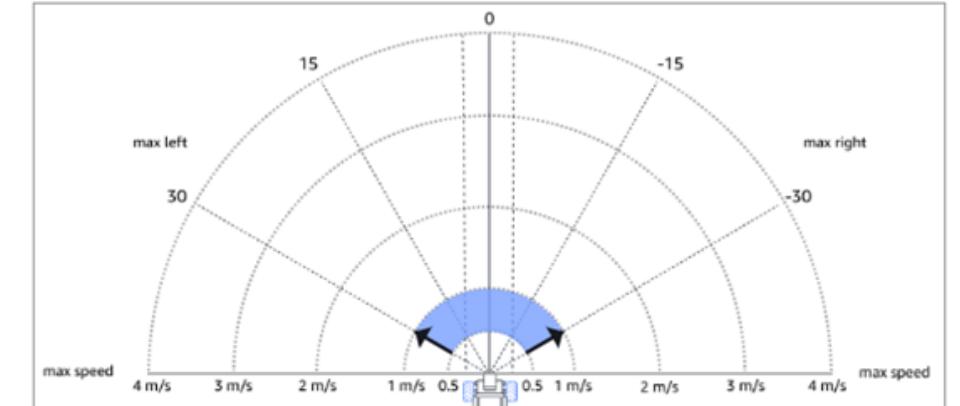
Maximum speed

m/s

Values are between 0.1 and 4.

[Reset to default values](#)

Dynamic sector graph



Select and drag an arrow to change the steering angle and speed.



Vehicle shell with sensor configuration (1)

Search

The Original DeepRacer

Sensor(s): Camera; Shell: DeepRacer



VEHICLE

CUSTOMIZE REWARD FUNCTION

Create model

Reward function Info

The reward function describes immediate feedback (as a score for reward or penalty) when the vehicle takes an action to move from a given position on the track to a new position. Its purpose is to encourage the vehicle to make moves along the track to reach its destination quickly. The model training process will attempt to find a policy which maximizes the average total reward the vehicle experiences. [Learn more](#) about the reward function and the reward input parameters you can use in your function.

Code editor

[Reward function examples](#)[Reset](#)[Validate](#)

```
1 def reward_function(params):
2     """
3     Example of rewarding the agent to follow center line
4     """
5
6     # Read input parameters
7     track_width = params['track_width']
8     distance_from_center = params['distance_from_center']
9
10    # Calculate 3 markers that are at varying distances away from the center line
11    marker_1 = 0.1 * track_width
12    marker_2 = 0.25 * track_width
13    marker_3 = 0.5 * track_width
14
15    # Give higher reward if the car is closer to center line and vice versa
16    if distance_from_center <= marker_1:
17        reward = 1.0
18    elif distance_from_center <= marker_2:
19        reward = 0.5
20    elif distance_from_center <= marker_3:
21        reward = 0.1
22    else:
23        reward = 1e-3 # likely crashed/ close to off track
24
25    return float(reward)
```

參考網址:

<https://engineering.linecorp.com/zh-hant/blog/aws-deepracer-2019>

<https://us-east-1.console.aws.amazon.com/deepracer/home?region=us-east-1#createModel>