
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

DPR202-R6

AWS DeepRacer: Get hands-on with machine learning

PRESENTED BY

intel

Tim O'Brien

Principal Solutions Architect
AWS

aws

Agenda

Introducing AWS DeepRacer

Introduction to reinforcement learning

Training using the AWS DeepRacer console

Hands-on workshop

Introducing AWS DeepRacer

How can we put reinforcement learning in the hands of all developers?

Literally

A closer look

- 1:18 4WD scale car
- Intel Atom processor
- Intel distribution of OpenVINO toolkit
- Stereo camera (4MP)
- 360-degree, 12-meter scanning radius Lidar sensor
- System memory: 4 GB RAM
- 802.11ac Wi-Fi
- Ubuntu 20.04 Focal Fossa
- ROS 2 Foxy Fitzroy

OpenVINO™

intel. ATOM

Get hands-on experience with reinforcement learning

AWS DeepRacer
Evo

Get hands-on experience with reinforcement learning

AWS DeepRacer Evo

3D racing simulator

Get hands-on experience with reinforcement learning

AWS DeepRacer Evo

3D racing simulator

PRESENTED BY

intel

AWS DeepRacer League

Get hands-on experience with reinforcement learning

AWS DeepRacer Evo

3D racing simulator

PRESENTED BY

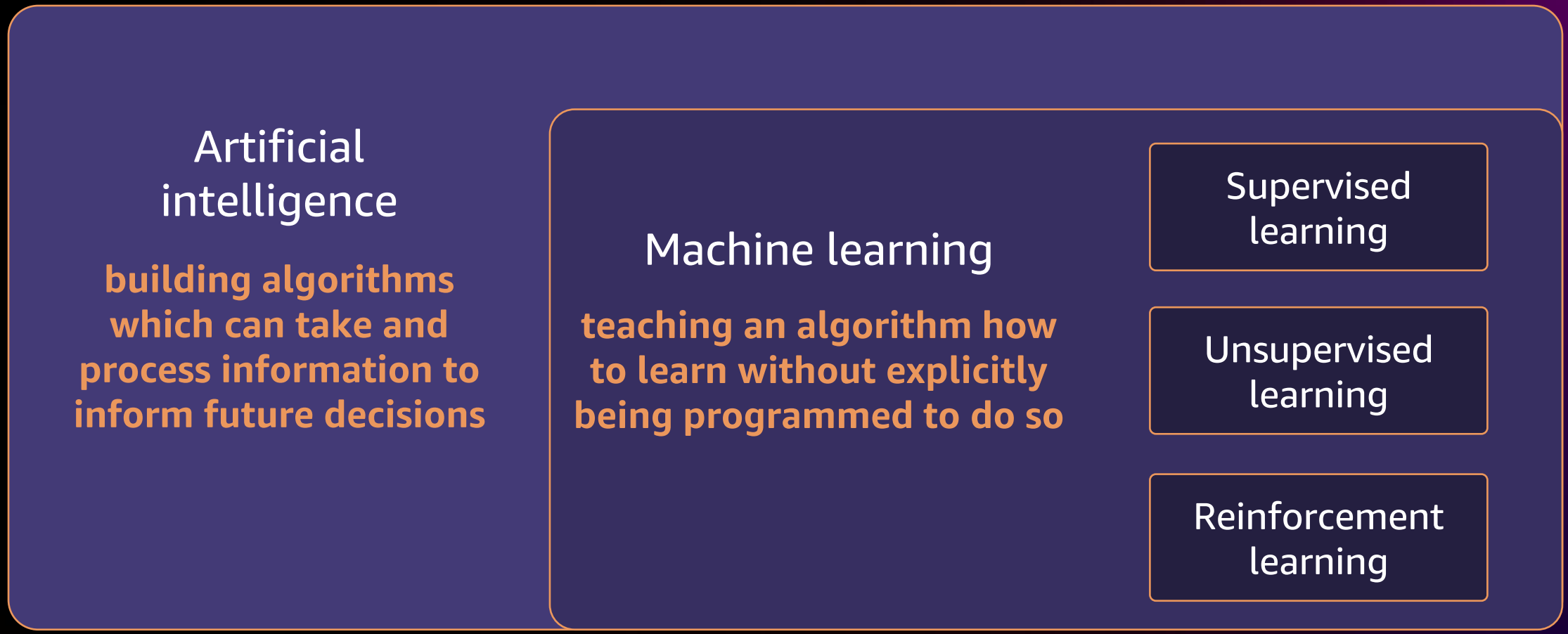
intel

AWS DeepRacer League

Community races

Introduction to reinforcement learning

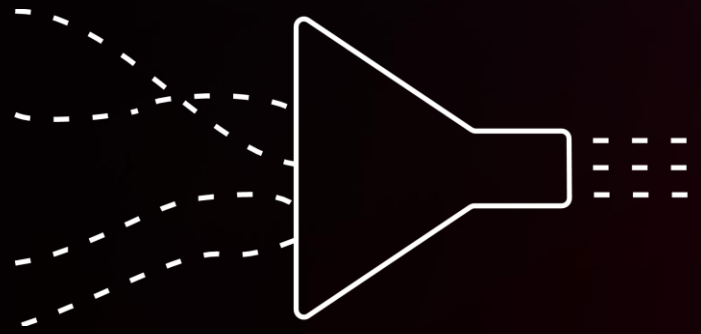
Reinforcement learning in the broad AI context



ML overview

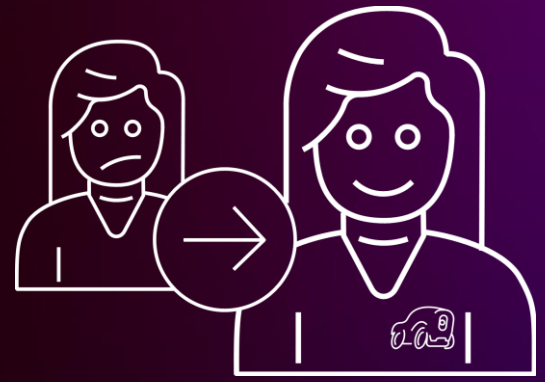
Supervised

Example-driven training; every datum has a corresponding label



Unsupervised

No labels for training data; useful for clustering like data



Reinforcement

Learns through consequences of actions in a specific environment

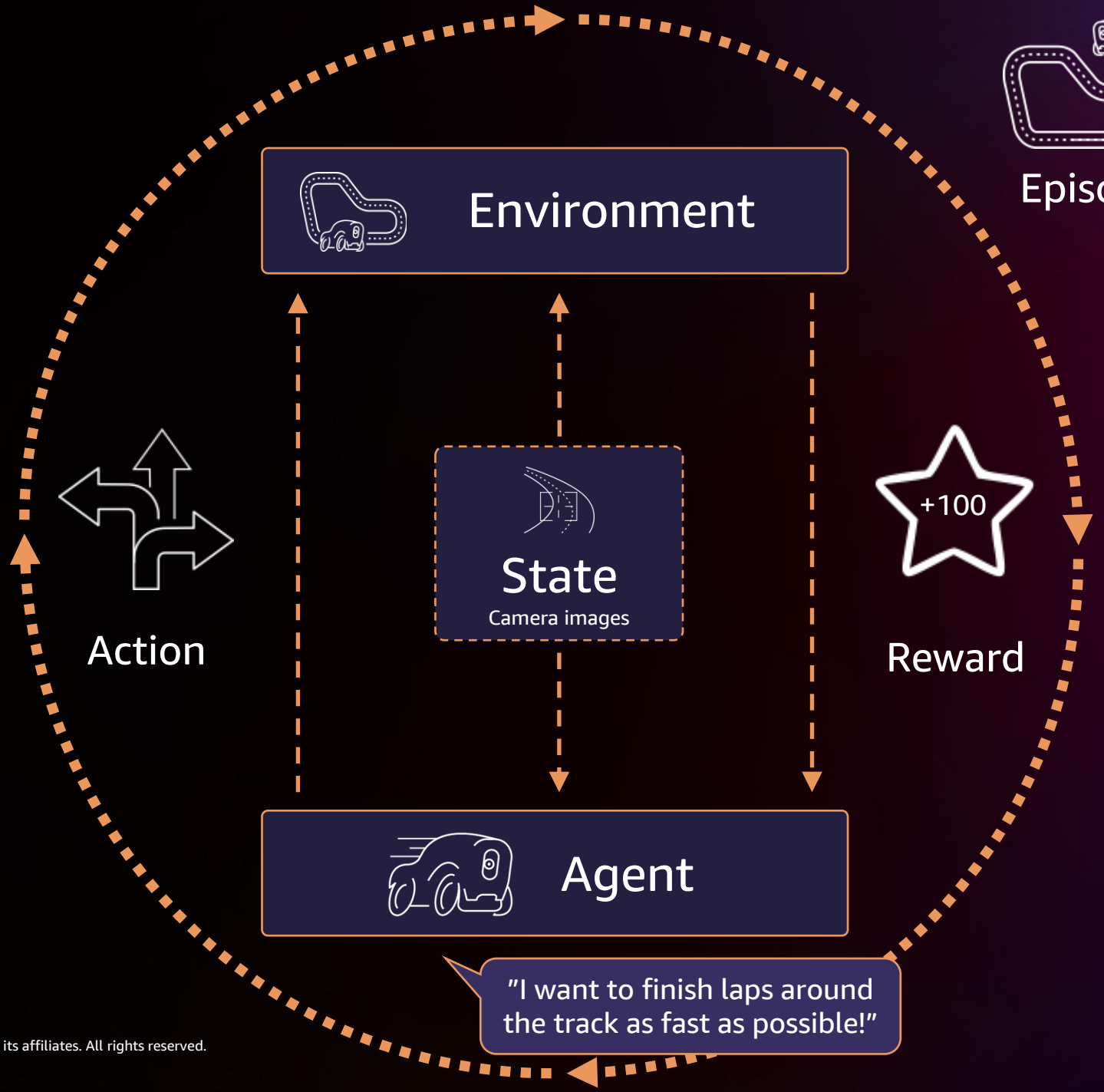
Reinforcement learning in the real world

Reward positive behavior

Don’t reward negative behavior

The result

Reinforcement learning key terms

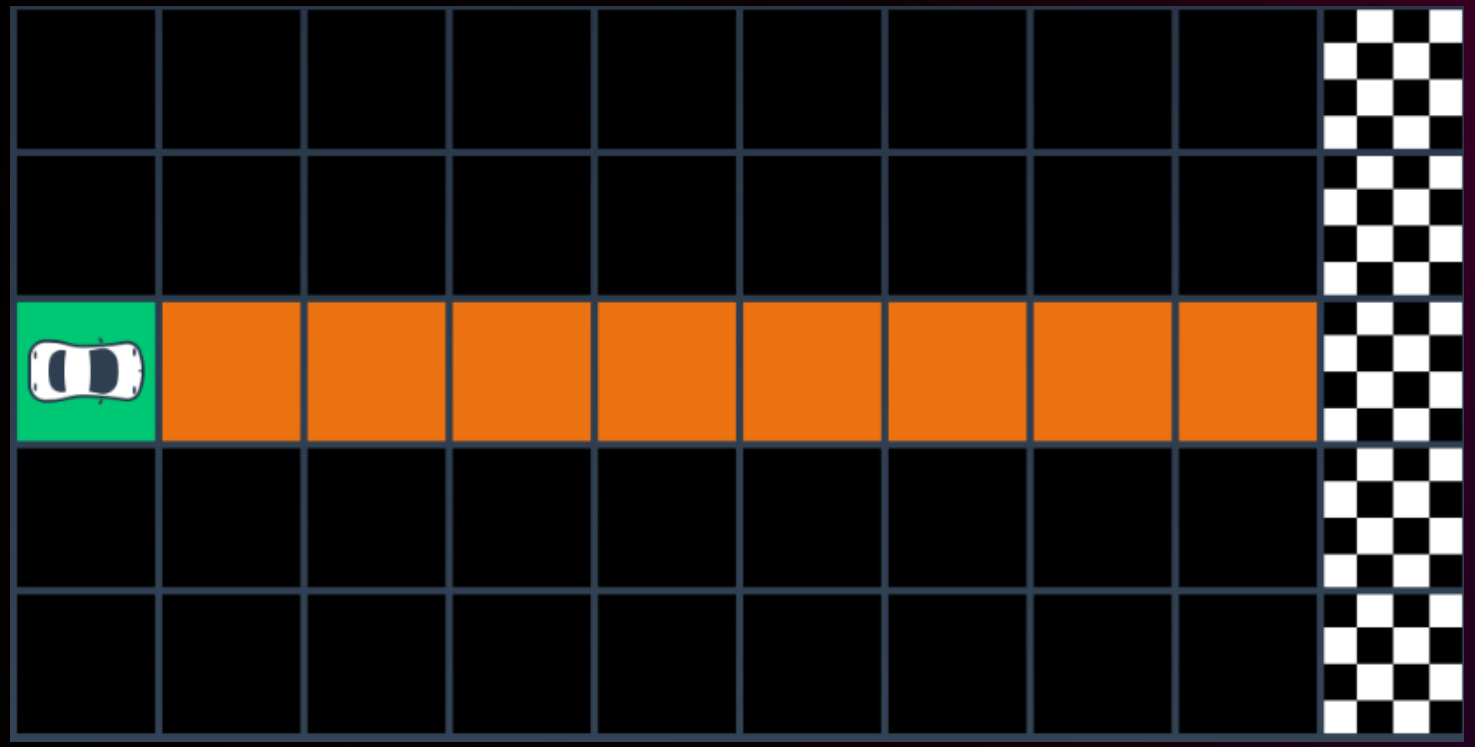


The reward function

The reward function incentivizes particular behaviors and is at the core of reinforcement learning

The reward function: Straight track race

Agent

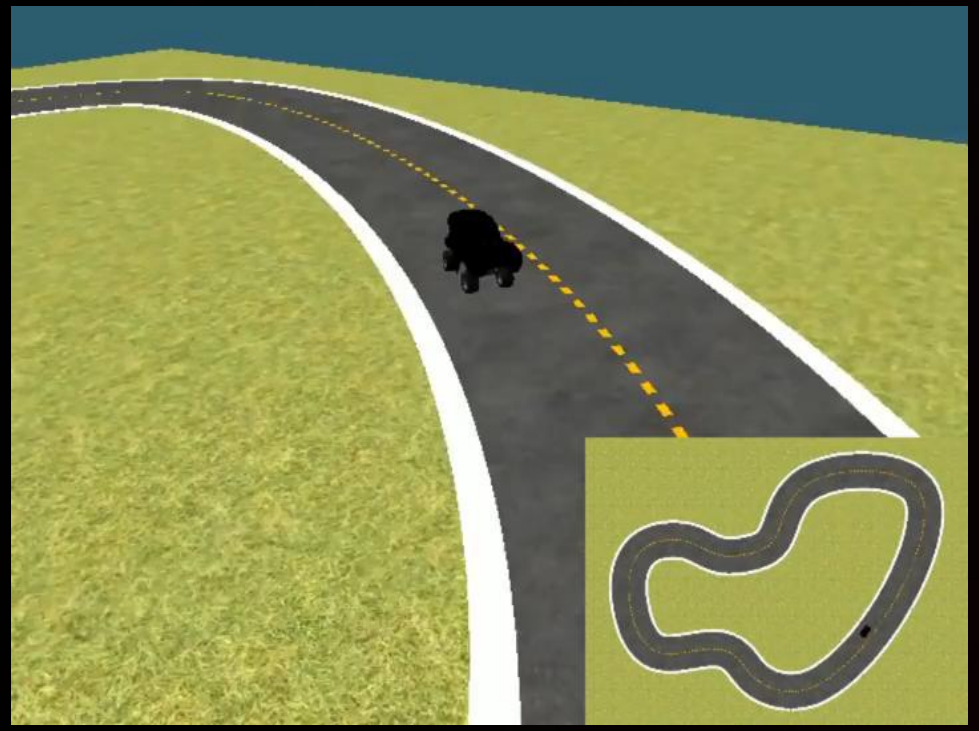


Goal

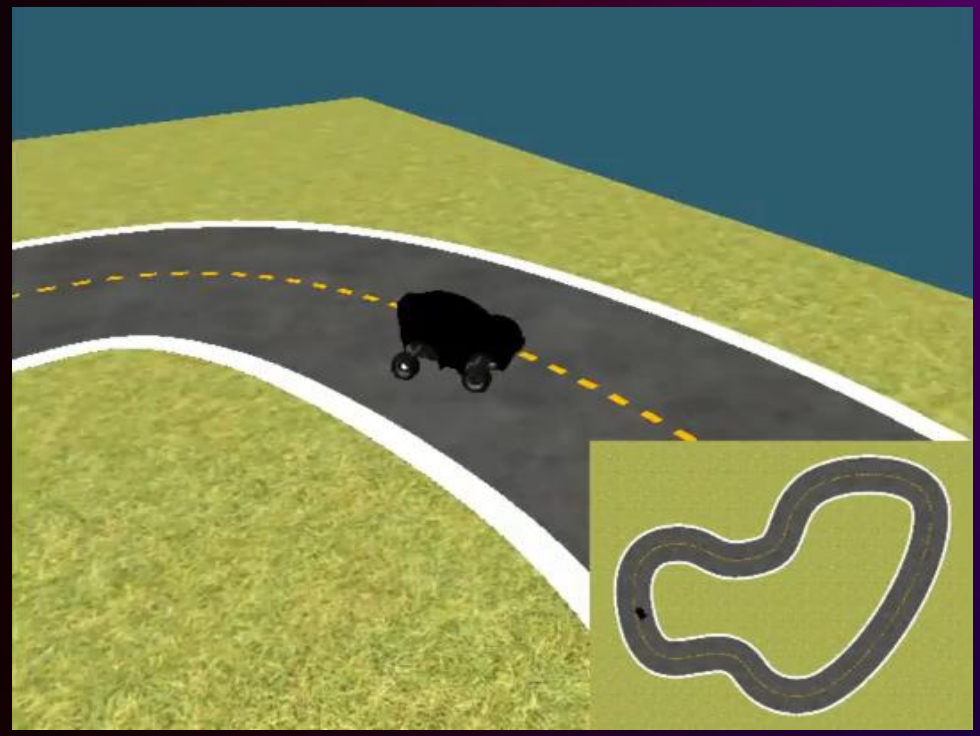
Rewards that incentivize center-line driving

Exploration versus exploitation

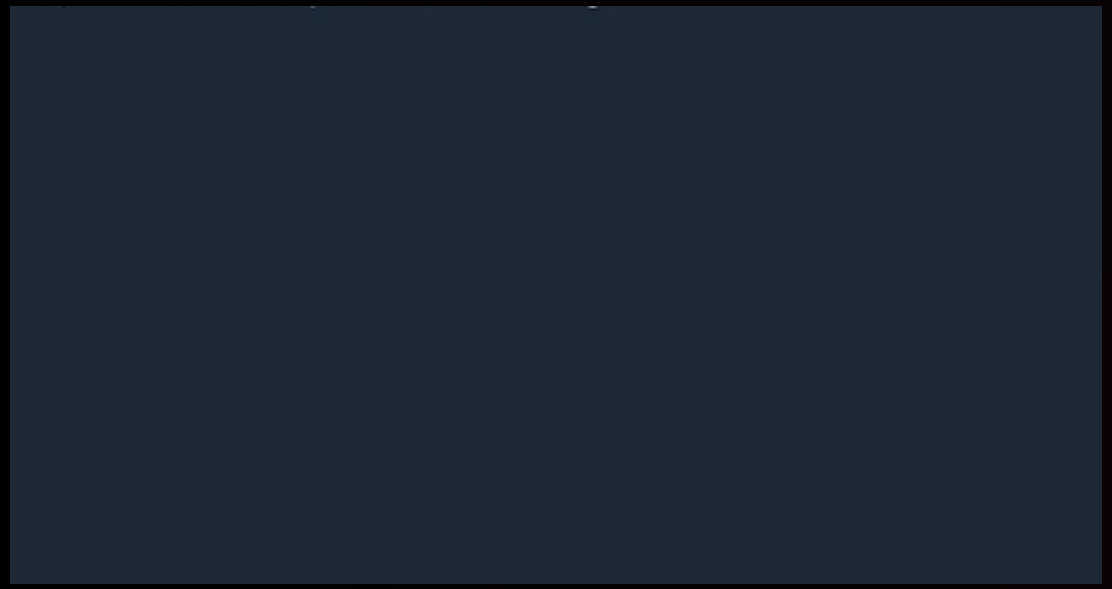
Exploration

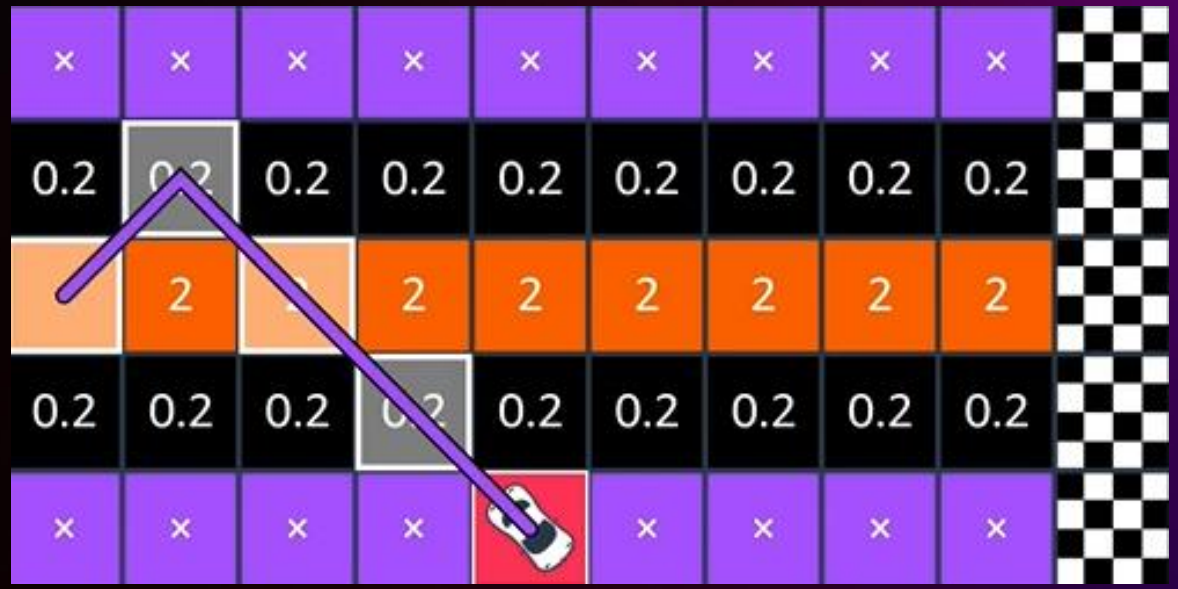


Exploitation

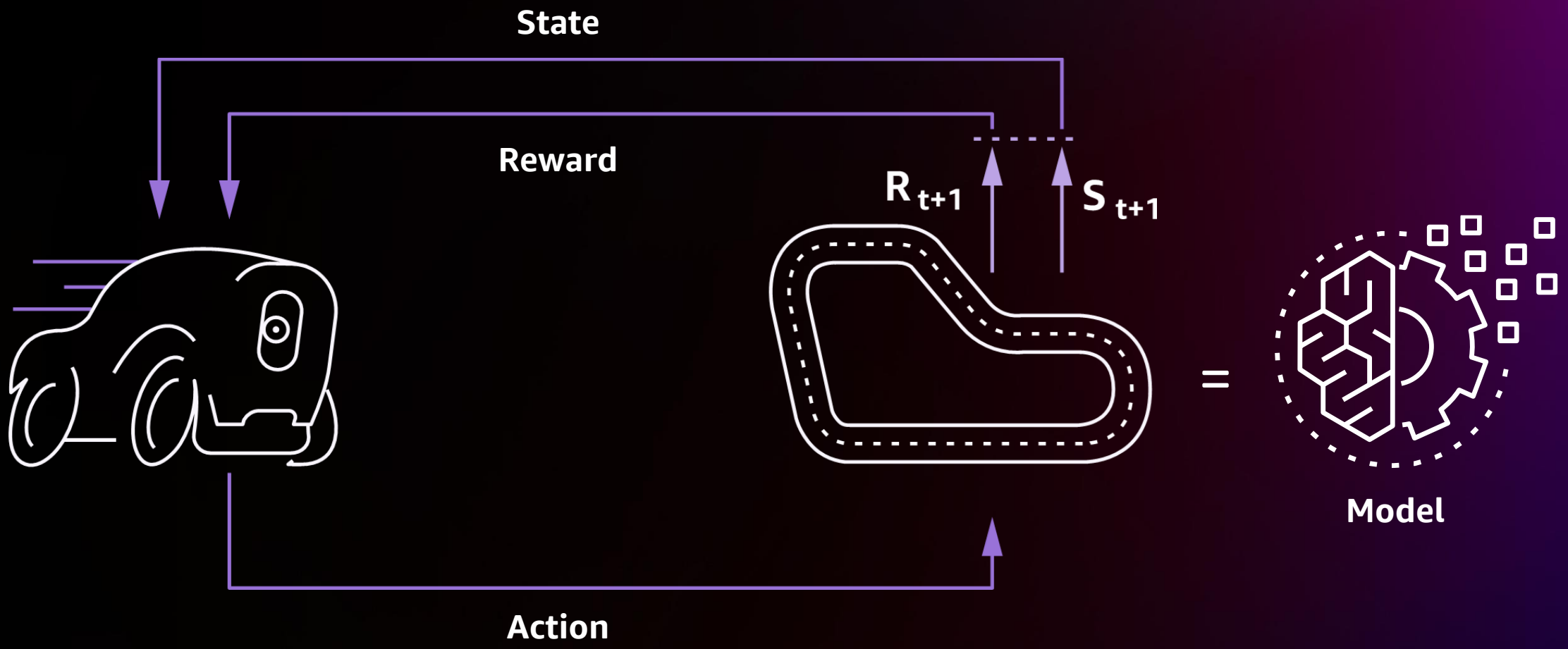


Iteration and convergence

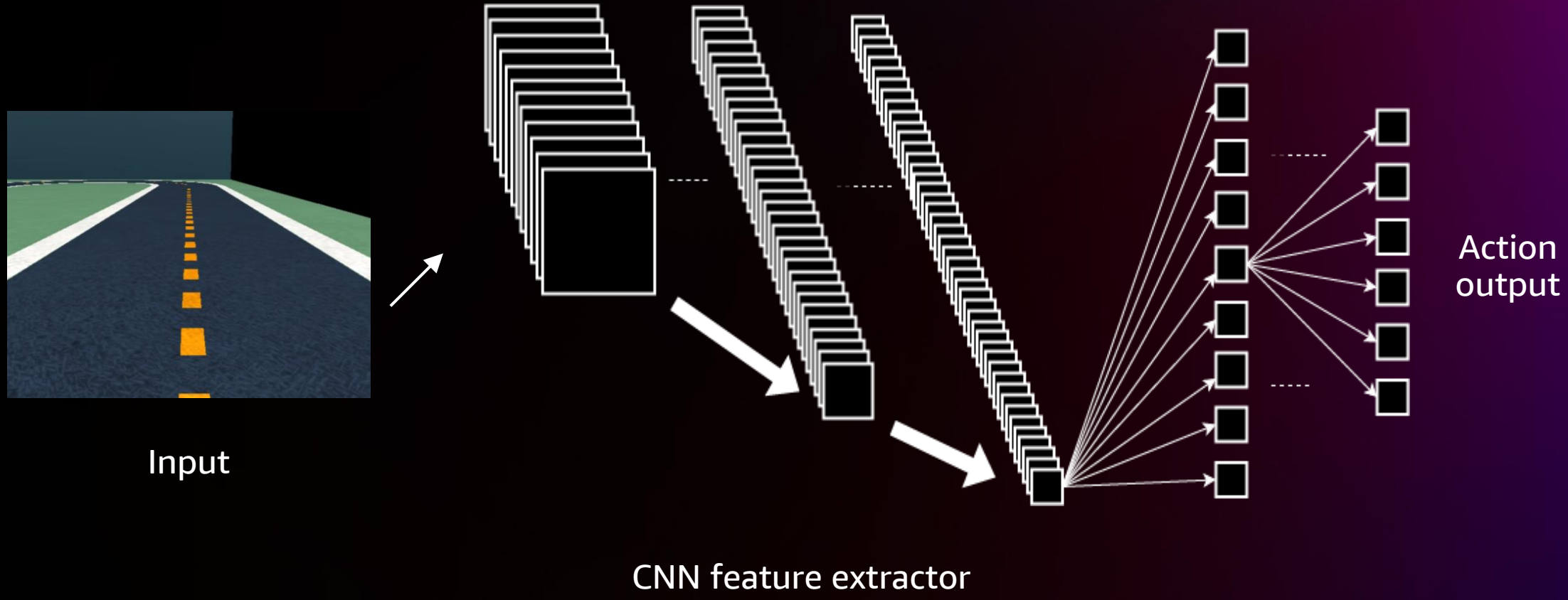




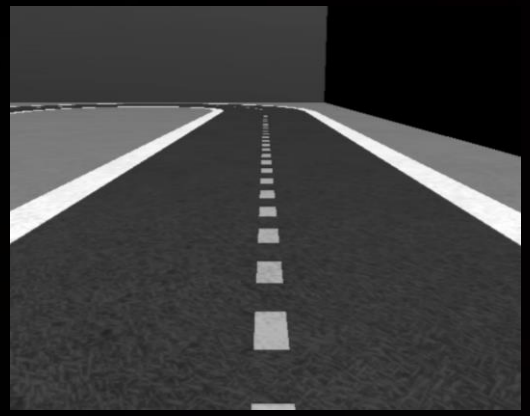
How does the learning (training) happen?



AWS DeepRacer neural network architecture

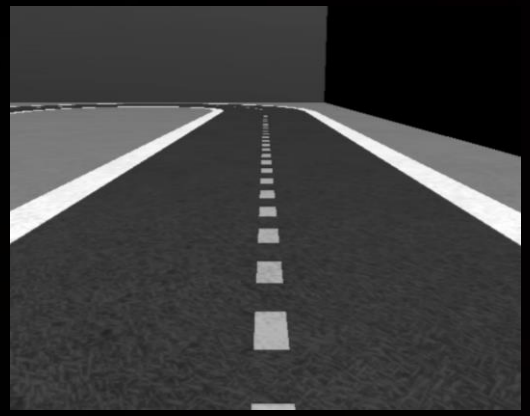


AWS DeepRacer neural network architecture

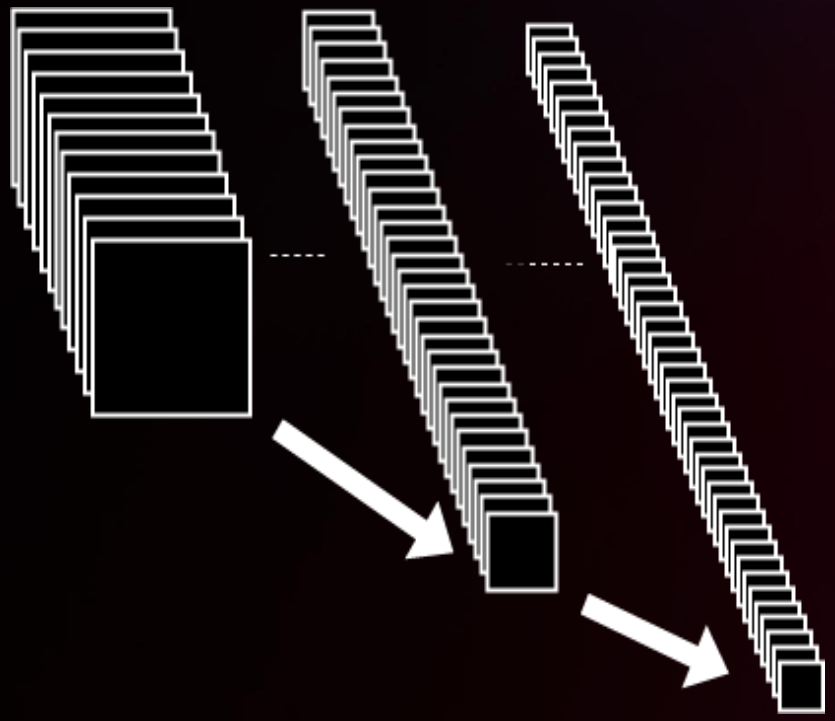


Input

AWS DeepRacer neural network architecture

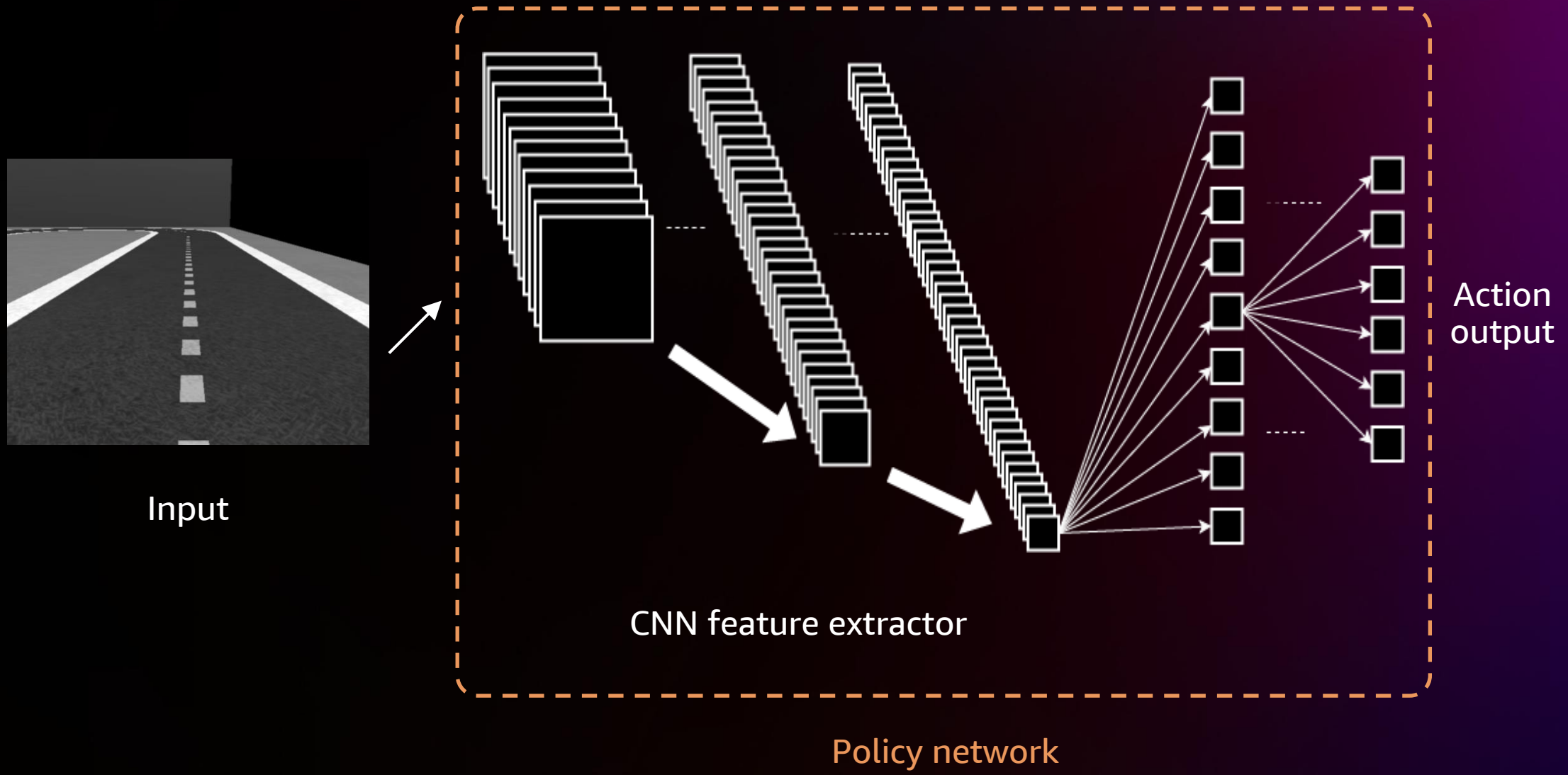


Input



CNN feature extractor

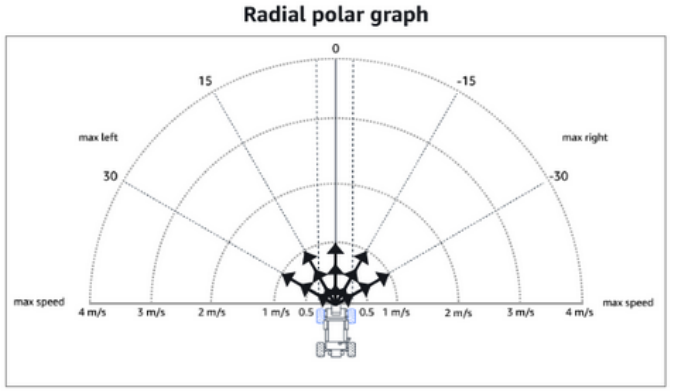
AWS DeepRacer neural network architecture



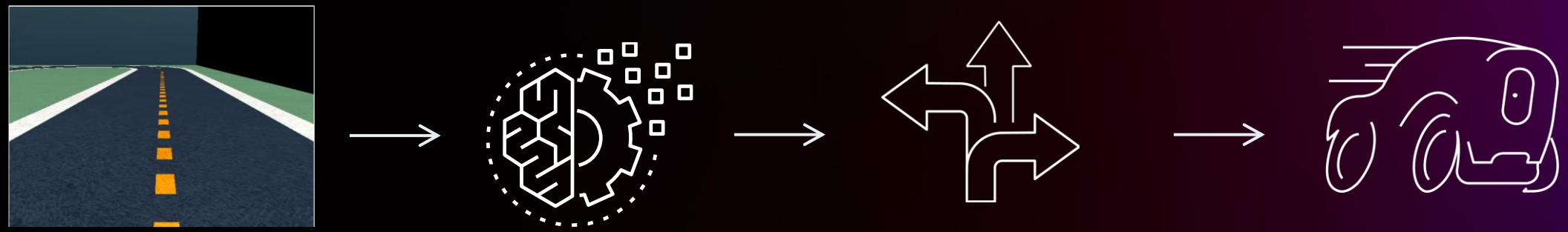
Action space

Action list — Advanced configuration

Action	Steering angle	Speed
0	-30.0 degrees	0.33 m/s
1	-30.0 degrees	0.67 m/s
2	-30.0 degrees	1.00 m/s
3	-15.0 degrees	0.33 m/s
4	-15.0 degrees	0.67 m/s
5	-15.0 degrees	1.00 m/s
6	0.0 degrees	0.33 m/s
7	0.0 degrees	0.67 m/s
8	0.0 degrees	1.00 m/s
9	15.0 degrees	0.33 m/s
10	15.0 degrees	0.67 m/s
11	15.0 degrees	1.00 m/s
12	30.0 degrees	0.33 m/s
13	30.0 degrees	0.67 m/s
14	30.0 degrees	1.00 m/s



Optimizing and inferencing with OpenVINO



Input data

OpenVINO optimized model

OpenVINO inference results

Racing

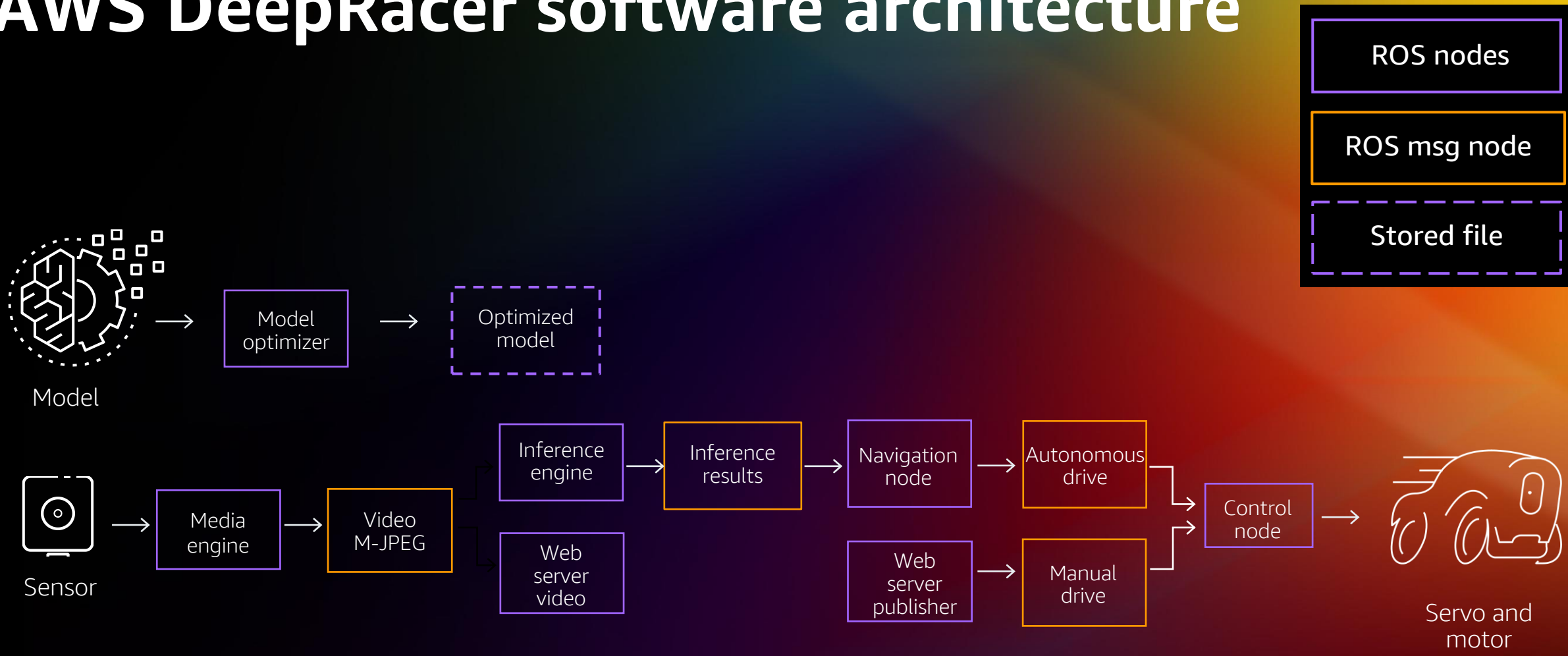
Free download: software.intel.com/openvino-toolkit
Open-source version: github.com/openvinotoolkit/openvino

OpenVINO™

intel ATOM

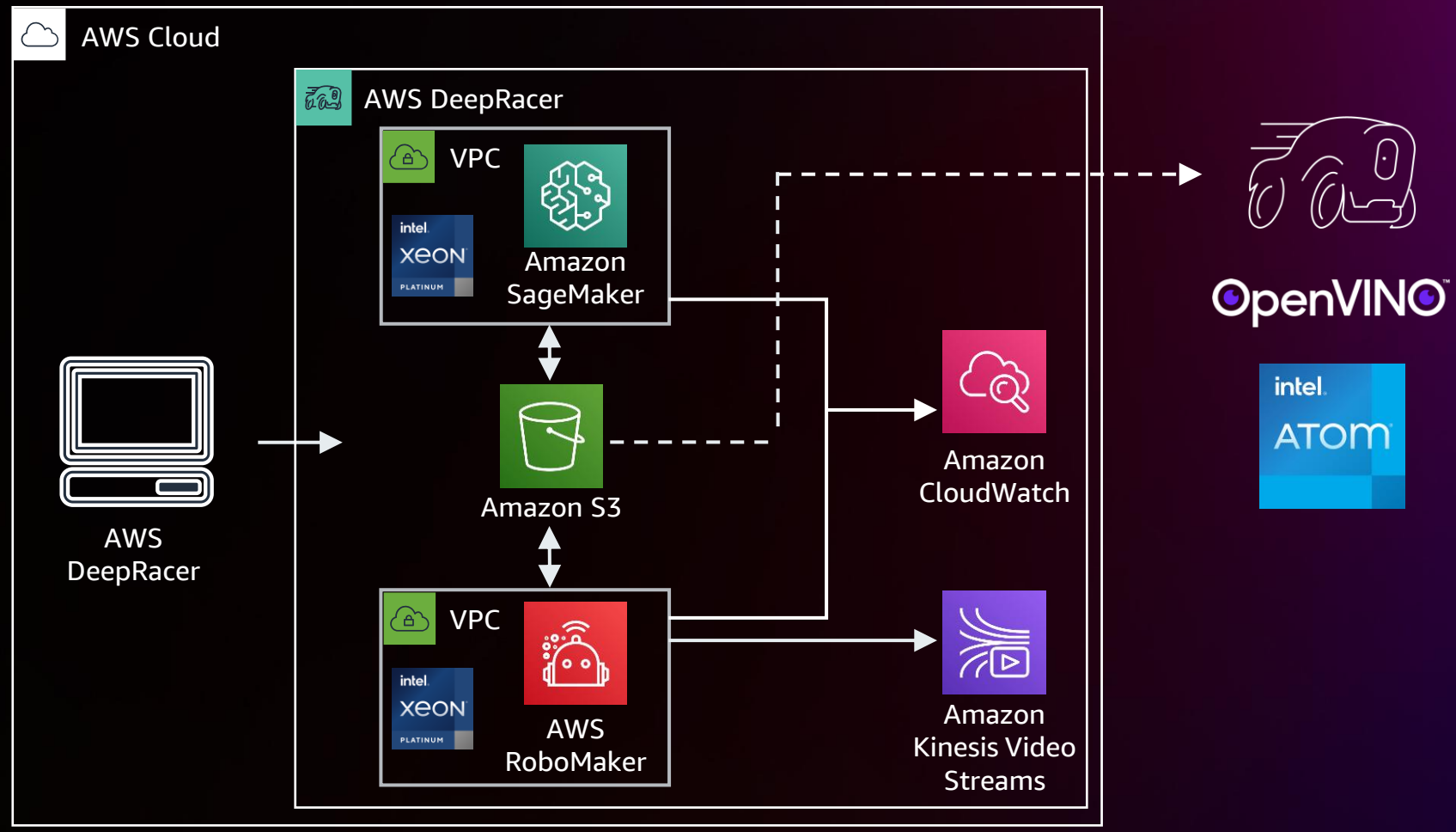
aws

AWS DeepRacer software architecture

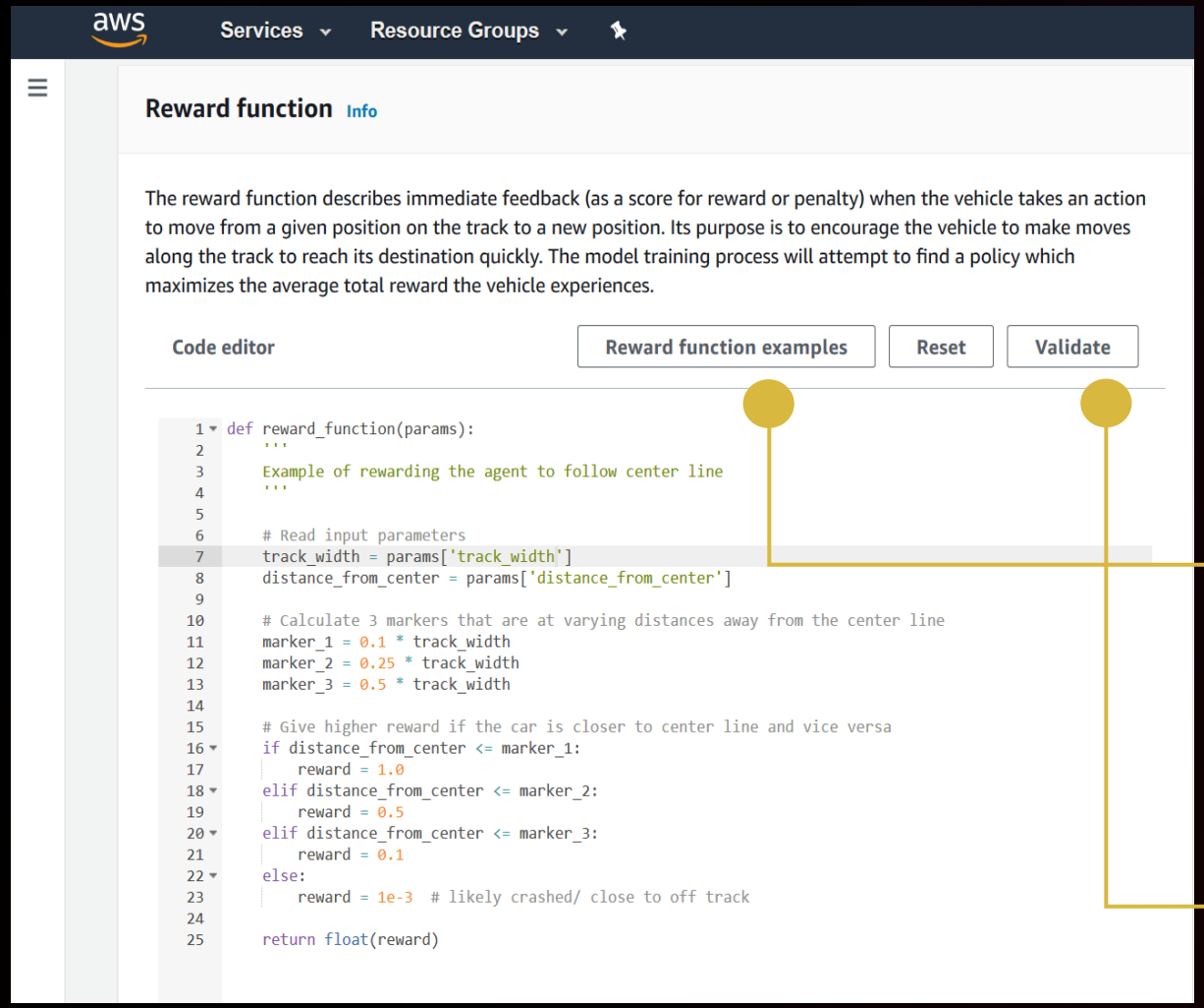


Training using the AWS DeepRacer console

AWS DeepRacer simulator architecture



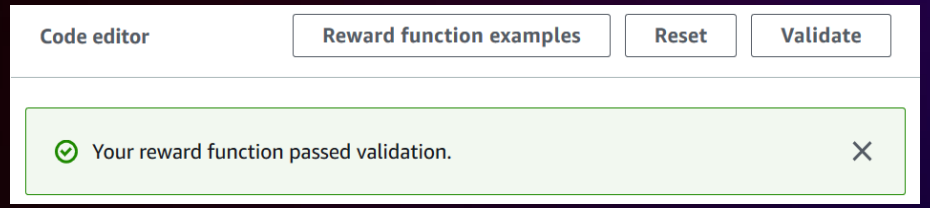
Programming your own reward function



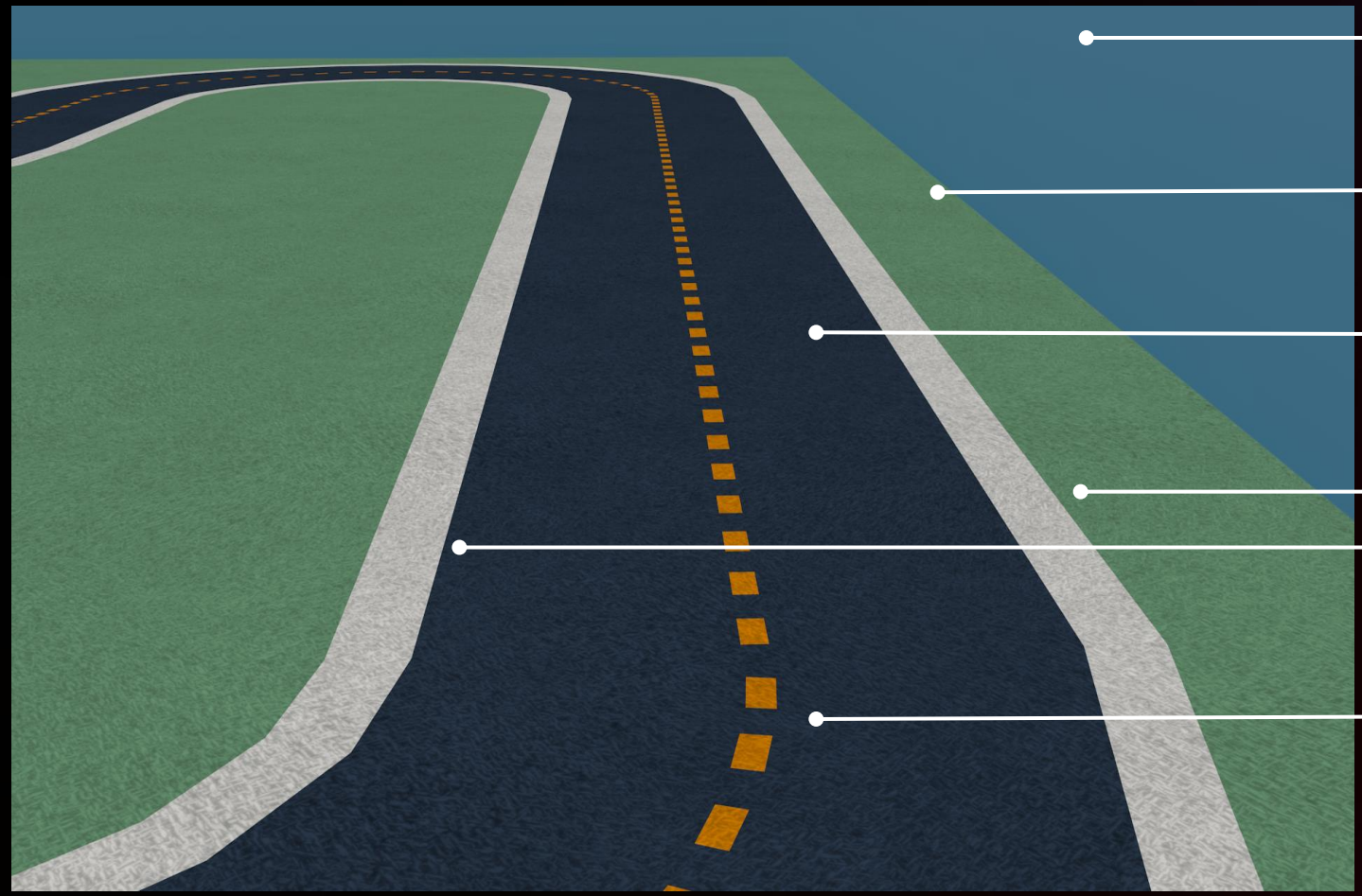
Code editor: Python 3 syntax

Three example reward functions

Code validation via Lambda



Track components



Track wall

Field (off-track)

Track surface (on-track)

Track boundaries

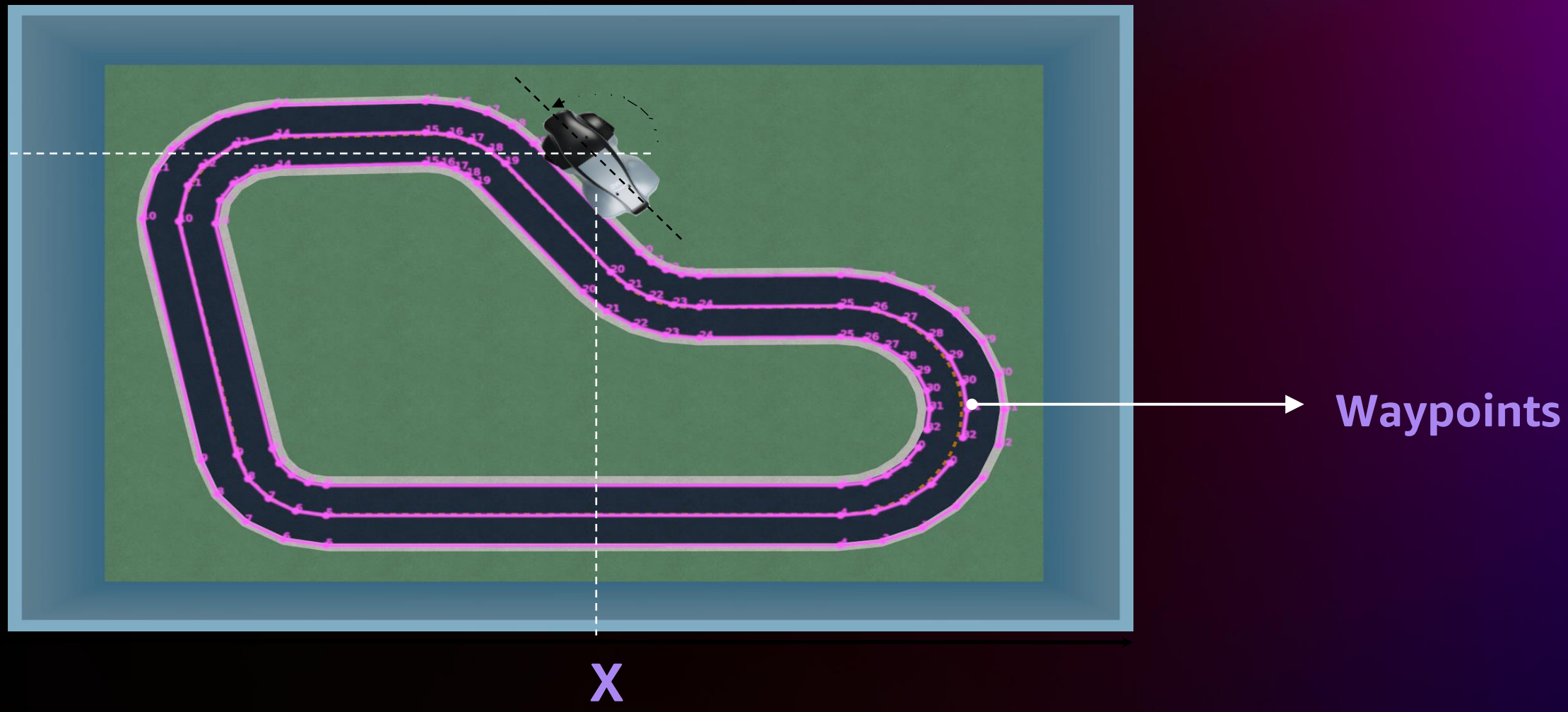
Track center

Reward function input parameters

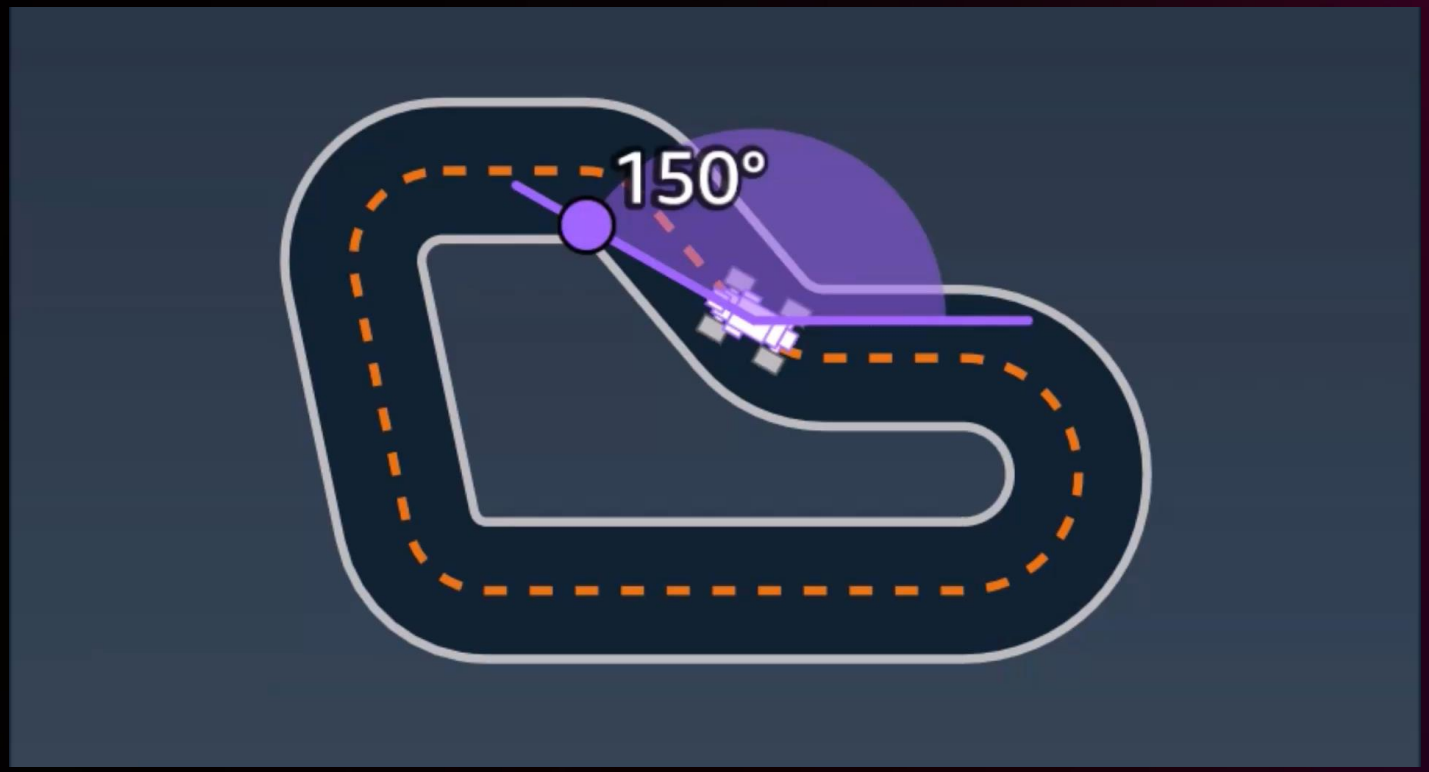
```
{
    "all_wheels_on_track": Boolean,        # flag to indicate if the agent is on the track
    "x": float,                            # agent's x-coordinate in meters
    "y": float,                            # agent's y-coordinate in meters
    "closest_objects": [int, int],         # zero-based indices of the two closest objects to the agent's current position of (x, y).
    "closest_waypoints": [int, int],       # indices of the two nearest waypoints.
    "distance_from_center": float,         # distance in meters from the track center
    "is_crashed": Boolean,                 # Boolean flag to indicate whether the agent has crashed.
    "is_left_of_center": Boolean,          # Flag to indicate if the agent is on the left side to the track center or not.
    "is_offtrack": Boolean,                # Boolean flag to indicate whether the agent has gone off track.
    "is_reversed": Boolean,                # flag to indicate if the agent is driving clockwise (True) or counter clockwise (False).
    "heading": float,                      # agent's yaw in degrees
    "objects_distance": [float, ],         # list of the objects' distances in meters between 0 and track_length in relation to the starting line.
    "objects_heading": [float, ],          # list of the objects' headings in degrees between -180 and 180.
    "objects_left_of_center": [Boolean, ], # list of Boolean flags indicating whether elements' objects are left of the center (True) or not (False).
    "objects_location": [(float, float),], # list of object locations [(x,y), ...].
    "objects_speed": [float, ],            # list of the objects' speeds in meters per second.
    "progress": float,                     # percentage of track completed
    "speed": float,                        # agent's speed in meters per second (m/s)
    "steering_angle": float,               # agent's steering angle in degrees
    "steps": int,                          # number steps completed
    "track_length": float,                 # track length in meters.
    "track_width": float,                  # width of the track
    "waypoints": [(float, float), ]        # list of (x,y) as milestones along the track center

}
```

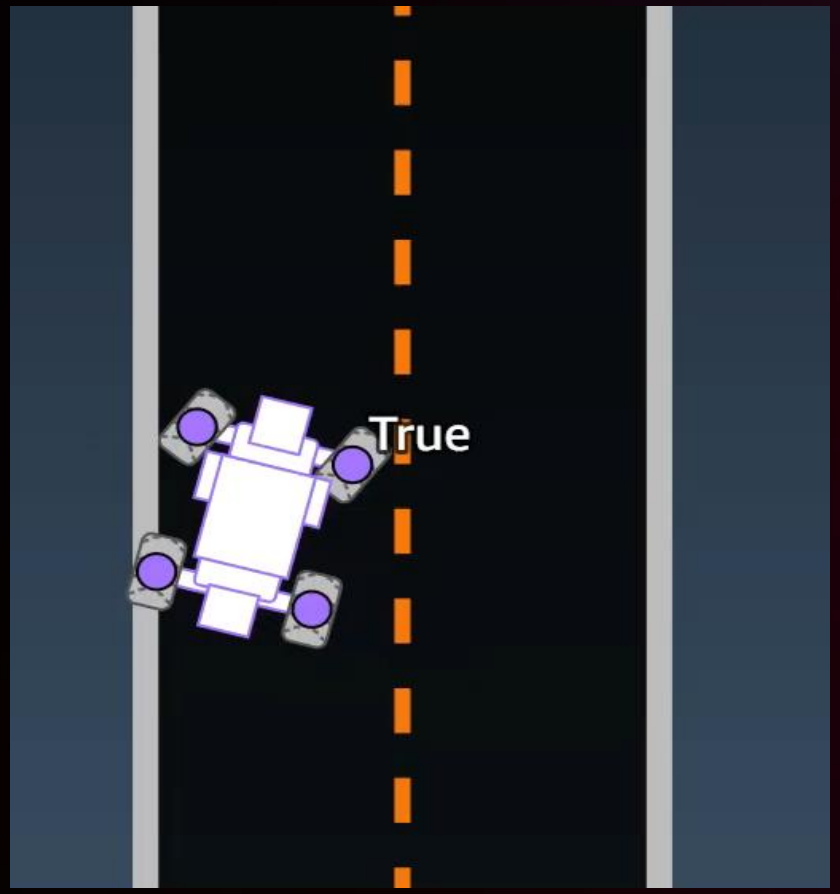
Coordinate system and track waypoints



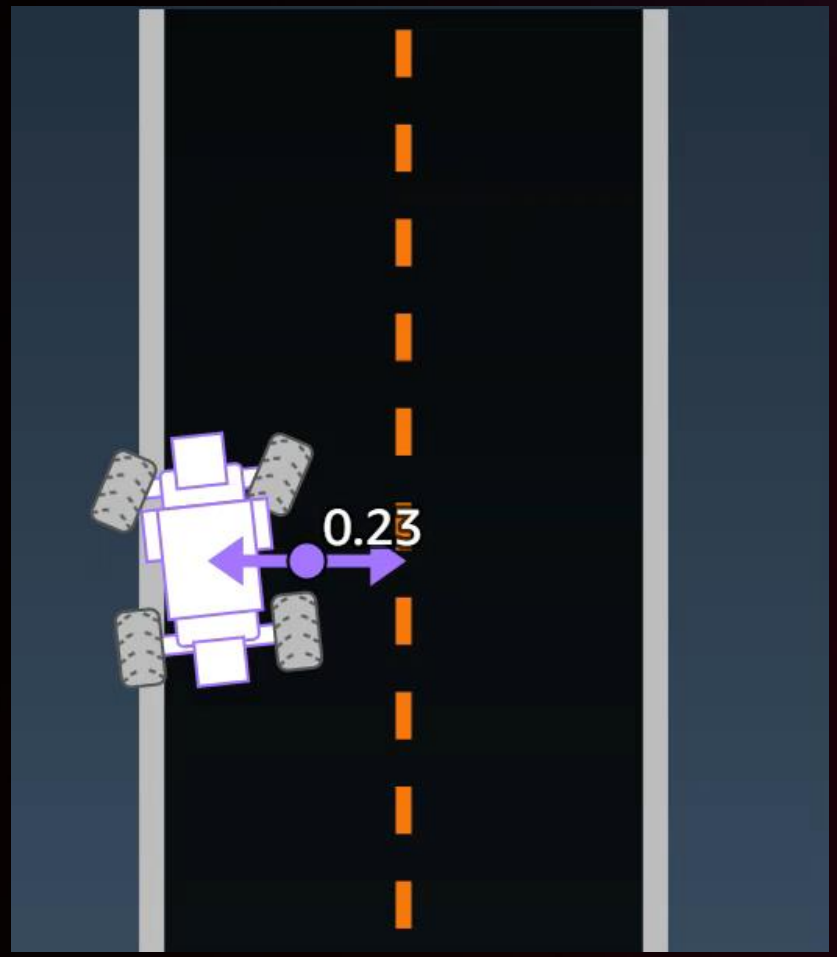
Example parameter: heading



Example parameter: all_wheels_on_track



Example parameter: distance_from_center



Hands-on workshop

Getting started with this workshop

https://catalog.workshops.aws/deepracer-200l/en-US

Next steps

Win an in-person driving experience!

Get behind the wheel of a DeepRacer Arcade or take your model for a spin in our open tracks in the MGM between Monday at 8:00 AM and Wednesday at 4:00 PM

You'll be automatically entered to win one of four spots to race full size exotic cars alongside our 2022 Champions and AWS VIP's at the **DeepRacer Winner's Circle Driving Experience at the Las Vegas Speedway** - no competition required!

Next steps

Win a wildcard spot in the 2023 Championship Cup at a ticket to next year's re:Invent

Race your model at any DeepRacer track on Thursday (10:00 AM to 5:00 PM) during the **re:Invent 2023 Open** to see how you stack up against the competition

All tracks across re:Invent will lead to one leaderboard, and the fastest developer will secure their spot (and ticket!) at next year's finals

Thank you!

Tim O’Brien
tpobrien@amazon.com

Please complete the session survey in the **mobile app**