사용자 가이드

AWS DeepRacer Student

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS DeepRacer Student: 사용자 가이드

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

AWS DeepRacer Student란 무엇인가요? 1
AWS AI 및 ML 장학금 프로그램에 관심이 있으신가요? 1
AWS DeepRacer Student를 처음 사용하시나요? 2
AWS DeepRacer Student League에서 경쟁하는 데 관심이 있으신가요? 2
AWS AI 및 ML 장학금 프로그램이란 무엇인가요? 3
AWS AI 및 ML 장학금에서 받게 되는 혜택 3
AWS AI 및 ML 장학금 프로그램에 등록하고 사전 자격을 얻으려면 어떻게 해야 하나요? 4
AWS AI 및 ML 장학금 프로그램과 Udacity 플랫폼에 관해 자주 묻는 질문(FAQ) 5
AWS 플레이어 계정이란 무엇입니까? 7
지원되는 서비스에 대한 AWS 플레이어 계정 생성 7
AWS 플레이어 계정 삭제 7
시작 9
사전 조건 9
1단계: AWS DeepRacer Student에 가입 9
2단계: AWS DeepRacer Student 가입 완료 10
3단계: (선택 사항) AWS AI 및 ML 장학금 신청에 관하여 읽고 동의하기 11
4단계: 내 프로필 업데이트 하기 11
5단계: 홈 페이지에서 Explore AWS DeepRacer Student 12
강화 학습 모델 훈련하기 14
1단계: AWS DeepRacer Student를 사용하여 강화 학습 모델 훈련 14
2단계: 내 모델 이름 지정하기 14
3단계: 트랙 선택하기 15
4단계: 알고리즘 선택하기 15
5단계: 보상 함수 사용자 지정하기 15
6단계: 기간을 선택하고 리더보드에 모델 제출하기 17
7단계: 리더보드에서 모델 성능 보기 17
8단계: 복제를 사용하여 모델 개선하기 18
9단계: (선택 사항) 모델 다운로드하기 18
레이스 참가 20
학생 리그 레이스 참가하기 20
학생 커뮤니티 레이스 참가하기 21
보상 함수 사용자 지정하기 22
Python 코드를 편집하여 보상 함수를 사용자 지정하세요 22
보상 함수 입력 파라미터 23

all_wheels_on_track 24
closest_waypoints 26
closest_objects 27
distance_from_center 27
heading 29
is_crashed 29
is_left_of_center 30
is_offtrack 30
is_reversed 30
objects_distance 30
objects_heading 31
objects_left_of_center 31
objects_location 31
objects_speed 31
progress 32
속도 32
steering_angle 33
steps 34
track_length 35
track_width 35
x, y 36
중간 지점 37
보안 39
데이터 보호 39
AWS DeepRacer Student 포털에서 캡처된 데이터 40
AWS DeepRacer Student 포털의 저장 시 암호화 40
In AWS DeepRacer Student 포털에서 전송 중 암호화 41
자격 증명 및 액세스 관리 41
규정 준수 확인 41
복원성 42
인프라 보안 42
일반적인 문제 해결 43
내 AWS DeepRacer Student 계정에서 자동으로 로그아웃된 이유는 무엇인가요? 43
AWS AI 및 ML 장학금 프로그램에서 탈퇴하려면 어떻게 해야 하나요? 43
my AWS DeepRacer Student 계정을 삭제할 수 없습니다. 44
드롭다운 목록에서 학교 이름을 찾을 수 없어요 44

모델 훈련을 계속할 수 없어요 44
'이 이메일에 등록된 계정이 있습니다'라는 오류 메시지가 뜹니다. 44
Gmail 계정으로 가입했는데 인증 코드를 못 찾겠어요 45
할당량 46
계정 삭제 47
문서 기록 49
..... I

AWS DeepRacer Student란 무엇인가요?

AWS DeepRacer Student는 전 세계 고등학교 및 대학 등록 학습자가 기계 학습(ML) 기술을 개발할 수 있는 곳입니다. 교육 자료, 선택적 AWS AI 및 ML 장학금 프로그램에 대한 액세스 권한과 AWS DeepRacer Student League에 대한 강화 학습(RL) 모델을 훈련하고 테스트할 수 있는 기회를 제공합니다. 시작하려면 ???섹션을 참조하세요.

AWS DeepRacer Student 기능

- 홈 — 예정된 이벤트에 대한 세부 정보를 찾고, RL 모델 훈련을 연습하고, ML 교육 콘텐츠에 접속하고, 내 모델 훈련 시간을 추적하세요. 왼쪽 탐색 창에서 AWS DeepRacer Student 프로필 및 계정 정보를 관리할 수도 있습니다.
- 학습 - AWS 전문가가 개발한 비디오를 포함한 ML 콘텐츠에 액세스합니다. 사전 경험이 없는 학생도 이해하기 쉽고 자기 주도형 자료를 통해 기계 학습의 기초를 배울 수 있습니다.
- 연습 — 트랙, 알고리즘, 보상 함수를 선택하여 강화 학습 모델을 생성하세요. 선택 사항으로, 보상 함수 Python 코드 설명을 듣고 사용자 정의할 수 있습니다. AWS DeepRacer 서비스를 사용하여 시뮬레이션된 3D 레이싱 환경에서 모델을 훈련합니다. 최고 성능의 모델을 복제하고 보상 함수를 반복하여 AWS DeepRacer Student League 리더보드에 오르세요.
- 경쟁 — 모델을 월간 가상 레이스에 제출하여 성과를 달성하고 상품을 받을 기회를 얻으세요. 지역 최고의 학생 레이서들은 학생 세계 선수권 대회(Student World Championships)에 출전할 기회를 얻습니다. 자세한 내용은 이용 약관을 참조하세요. 모든 학생은 매달 10시간 동안 AWS DeepRacer Student League에 대한 RL 모델을 훈련할 수 있습니다. AWS DeepRacer Student를 사용하여 훈련된 모델만 AWS DeepRacer Student 리그에서 사용할 수 있습니다.

주제

- AWS AI 및 ML 장학금 프로그램에 관심이 있으신가요?
- AWS DeepRacer Student를 처음 사용하시나요?
- AWS DeepRacer Student League에서 경쟁하는 데 관심이 있으신가요?

AWS AI 및 ML 장학금 프로그램에 관심이 있으신가요?

본인이 기술 분야 교육의 혜택을 받지 못하거나 소외되었다고 느끼면 AWS AI 및 ML 장학금 프로그램에 지원해보세요. AWS AI 및 ML 장학금 프로그램이란 무엇인가요?에서 자격 대상자, 신청 방법, 수상 내역을 자세히 알아보세요.

AWS DeepRacer Student를 처음 사용하시나요?

AWS DeepRacer Student를 처음 사용하는 경우 다음 섹션을 읽는 것으로 시작하는 것이 좋습니다.

- AWS DeepRacer Student 시작하기
- AWS DeepRacer Student 레이스 참가
- AWS DeepRacer Student에서 강화 학습 모델 훈련

AWS DeepRacer Student League에서 경쟁하는 데 관심이 있으신가요?

The AWS DeepRacer Student League는 3월부터 9월까지 매월 가상 레이스에서 경쟁하여 국가 및 리전 수준에서 경쟁하고 10월에 Student World Championships에서 경쟁할 기회를 얻을 수 있는 AWS후원 대회입니다. 학생들은 보상과 상품을 받을 기회가 있습니다. 자세한 내용은 이용 약관을 참조하세요.

내 첫 레이스에 참가하려면 AWS DeepRacer Student 레이스 참가을 참조하세요.

AWS AI 및 ML 장학금 프로그램이란 무엇인가요?

AWS DeepRacer Student의 일부로 출시되는 AWS AI 및 ML 장학금 프로그램은 성공적인 신청자에게 최대 2개의 Udacity Nanodegree를 획득할 수 있는 기회를 제공하여 인공 지능(AI) 및 기계 학습(ML) 분야에 다양성을 제공하도록 설계되었습니다. 유다시티 나노디그리는 학습과 경력 목표 사이의 간극을 해소하고자 설계된 대규모 공개 온라인 교육 과정(MOOC) 입니다. 자세한 내용은 Udacity 지원 문서의 나노디그리 프로그램이 무엇인가요? 항목을 참조하세요. 합격자에게는 경력을 발전시키는 데 도움이 되는 독점 이벤트와 멘토링도 제공됩니다. 자세한 내용은 AWS AI 및 ML 장학금에서 받게 되는 혜택 항목을 참조하세요.

이 장학금은 기술 분야에서 교육의 혜택을 덜 받고 소외된 인재를 대상으로 합니다. 지원자는 16세 이상이어야 하며 현재 고등학교, 2년제 또는 4년제 대학교에 재학 중이어야 합니다. 사전 자격 획득에 대한 자세한 내용은 AWS AI 및 ML 장학금 프로그램에 등록하고 사전 자격을 얻으려면 어떻게 해야 하나요? 항목을 참조하세요.

AWS AI 및 ML 장학금 프로그램 참가는 무료입니다.

AWS AI 및 ML 장학금 프로그램은 2022년 4월 11일에 공식적으로 출범했습니다.

AWS AI 및 ML 장학금 프로그램은 코호트 기반 접근 방식으로 운영됩니다. 매년 1,000명의 학생으로 구성된 두 개의 코호트(총 2,000명)가 선발됩니다. 첫 번째 코호트 신청은 2024년 5월 31일까지 Udacity에 제출해야 하며, 두 번째 코호트 신청 마감일은 2024년 9월 30일입니다.

AWS AI 및 ML 장학금에서 받게 되는 혜택

AWS AI 및 ML 장학금 프로그램의 경우 AWS 는 Udacity와 협력합니다.

매년 AWS AI 및 ML 장학금 프로그램은 2,500개의 유다시티 나노디그리 장학금을 전 세계에서 서로 다른 두 코호트에 나눠 제공합니다. 모든 학생에게 제공되는 혜택은 다음과 같습니다.

- Python을 이용한 AI 프로그래밍 나노디그리 무료 수강권. 이 교육 과정은 AI 기술을 사용하기 시작하고 프로그래밍, 선형 대수학, 신경망 기술을 개발하는 데 필요한 기초 기술을 가르칩니다.
- 주중 업무 시간에는 Udacity 강사와 함께 수업 내용에 관한 문답을 나눕니다. 또한 학생들은 Udacity 강사가 진행하는 주간 사례 연구 연습에 참여할 수 있습니다.
- 업계 전문가 세션, 이력서 편집 안내, 면접 팁 등 멘토십 및 커리어 정보 리소스에 독점적으로 접속할 수 있습니다.

또한 Python을 이용한 AI 프로그래밍 강의 상위 500명의 학생(수업 평가 성과 기준)은 딥 러닝의 기초를 다루고 기계 학습 엔지니어가 되는 데 필요한 기술을 제공하는 고급 나노디그리 과정을 수강할 수 있습니다.

장학금 사전 조건을 완료하고 Udacity 애플리케이션을 제출하는 모든 학생은 AWS 및 Udacity에서 공동 생성한와 함께 생성형 AI 소개라는 새로운 생성형 AI AWS 과정에 무료로 액세스할 수 있습니다. 한 달 동안 진행되는 이 과정은 학생들이 생성형 AI 분야의 기술을 더욱 향상시킬 수 있도록 도와주는 선택적 리소스입니다. 이 과정은 장학금 신청서를 제출한 모든 학생이 이용할 수 있는 반면, 나노디그리 장학금은 장학금 수혜자만 이용할 수 있습니다.

AWS AI 및 ML 장학금 프로그램에 등록하고 사전 자격을 얻으려면 어떻게 해야 하나요?

AWS AI 및 ML 장학금 프로그램은 16세 이상의 교육 혜택을 덜 받고 소외된 학생을 대상으로 합니다. 교육 혜택을 덜 받고 소외된 학생에는 여성, 장애인, 유색인종(흑인, 라틴계, 원주민), 성소수자 커뮤니티 구성원이 포함되지만 이에 국한되지는 않습니다.

시작하려면 학생이 AWS DeepRacer Student에 가입하고 계정을 생성해야 합니다. 가입 시 확인란에 체크 표시하여 AWS AI 및 ML 장학금 프로그램 신청에 동의합니다. 계정이 생성되었으면 홈 페이지에 장학금 신청 사전 자격 작업을 추적할 수 있는 타일이 표시됩니다. 학생은 AWS DeepRacer Student에서 장학금 프로그램 신청을 제출하기 위한 두 가지 사전 조건을 완료해야 합니다.

AWS AI 및 ML 장학금 사전 자격을 갖춘 학생에게는 Udacity 사이트에서 사용할 수 있는 고유 액세스 코드가 제공되며, 이 코드로 유다시티 나노디그리 장학금 신청 양식에 접속할 수 있습니다. 사전 자격을 갖춘 학생에게 유다시티 나노디그리 장학금이 보장되는 것은 아닙니다. Udacity는 사전 자격을 갖춘 학생 중 유다시티 나노디그리 장학금 대상자를 결정합니다.

장학금 수혜 자격은 다음 두 가지 기준에 따라 결정됩니다.

교과 과정 검토 및 합격 평가

사전 자격을 갖추려면 모든 필수 평가에서 80% 이상의 점수를 받아야 합니다. 각 평가는 AWS DeepRacer Student 탐색 창의 학습 섹션에서 찾을 수 있는 다양한 장을 기반으로 합니다. 장학금 신청은 2024년 2월 1일에 시작됩니다.

최소 랩 타임을 달성하세요

2월부터 9월까지 매달 새로운 트랙이 포함된 AWS DeepRacer 리그에서 새 리더보드가 출시됩니다. 장학금을 받기 위한 사전 자격을 얻으려면 어떤 싱글 레이스에서든 싱글 랩 타임 2분 미만을 달성해야 합니다.

학생은 두 사전 조건을 모두 완료한 후에서 고유한 액세스 코드와 Udacity Nanodegree 장학금용 애플리케이션 링크가 AWS 포함된 이메일을 받습니다. 장학금 수혜 대상자로 고려되려면, 누락된 사항 없이 신청서를 작성해야 합니다. 신청이 완료되면 Udacity는 모든 지원자에게 제공되는 선택 사항인 AWS로 만나는 생성형 AI 입문 과정의 등록 절차를 학생들에게 안내합니다. Udacity는 유다시티 나노디그리 장학금 수혜자로 선정된 지원자에게 연락을 취합니다.

AWS AI 및 ML 장학금 프로그램과 Udacity 플랫폼에 관해 자주 묻는 질문(FAQ)

공식 AWS AI 및 ML 장학금 이용 약관은 어디에서 볼 수 있나요?

공식 이용 약관을 보려면 공식 AWS AI 및 ML 장학금 프로그램 이용 약관을 참조하세요.

이 장학금은 누구를 위한 것인가요?

AWS AI 및 ML 장학금 프로그램은 전 세계에서 교육 혜택을 덜 받고 소외된 16세 이상의 학생을 대상으로 합니다. 교육 혜택을 덜 받고 소외된 학생에는 여성, 장애인, 유색인종(흑인, 라틴계, 원주민), 성소수자 커뮤니티 구성원이 포함되지만 이에 국한되지는 않습니다.

나노디그리(Nanodegree)란 무엇인가요?

나노디그리란 학습과 직업 기술 간의 간극을 해소하는 데 도움이 되는 온라인 기술 기반 교육 프로그램입니다.

AWS AI 및 ML 장학금 사전 자격을 갖추면 유다시티 나노디그리 장학금을 자동으로 받게 되나요?

아닙니다. AWS AI 및 ML 장학금 사전 자격을 갖춘 학생은 유다시티 나노디그리 장학금을 신청할 수 있습니다. Udacity는 사전 자격을 갖춘 학생 중 유다시티 나노디그리 장학금 대상자를 결정합니다.

AWS AI 및 ML 장학금에 선정되려면 재학 증명서를 제출해야 하나요?

AWS AI 및 ML 장학금을 받으려면 재학 증명서(예: 대학 성적 증명서)를 제출해야 할 수도 있습니다.

AWS AI 및 ML 장학금 프로그램을 신청할 때 현재 고등학교에 재학 중임을 어떻게 표시하나요?

고등학교에 재학 중임을 표시하려면 AWS DeepRacer Student 사용 안내서를 참조하세요4단계: 내 프로필 업데이트 하기.

개인 식별 정보(PII) 데이터는 어떻게 보호되나요?

데이터는 AWS 클라우드에서 보호됩니다. 자세한 내용은 AWS 개인 정보 보호 정책을 참조하세요.

AWS AI 및 ML 장학금을 신청하는 데 연령 제한이 있나요?

예. AWS AI 및 ML 장학금 프로그램에 신청하려면 신청 시점에 최소 16세 이상이어야 합니다.

AWS AI 및 ML 장학금을 받게 되면 어떻게 알림을 받나요?

신청 마감일이 지나면 Udacity에서 나노디그리 장학금 신청 현황에 관한 이메일을 받게 됩니다.

AWS AI 및 ML 장학금을 받게 되면 언제 알려 주나요?

Udacity가 지원자에게 연락하여 장학금 현황을 알려줄 것입니다. 장학금 수혜자로 선정된 학생에게는 중요한 날짜에 대한 정보가 제공됩니다.

Udacity에서 AWS 생성형 AI 과정에 액세스하려면 어떻게 해야 하나요?

학생은 on AWS DeepRacer Student의 사전 자격을 갖추고 Udacity에서 장학금 신청을 제출해야 합니다. Udacity는 신청서를 온전히 작성해 제출한 학생에게 연락하여 AWS로 만나는 생성형 AI 입문 과정에 등록할 수 있는 단계를 안내합니다.

AWS AI 및 ML 장학금 프로그램에 합격하면 어떻게 되나요?

AWS AI 및 ML 장학금을 받았다는 통지가 간 후 Udacity 온보딩 팀의 담당자가 신청서를 작성할 때 Udacity에 제공한 이메일로 연락을 드릴 것입니다.

유다시티 나노디그리 장학금을 받으면 나노디그리 학습에 얼마나 많은 시간을 할애해야 하나요?

학생들은 주당 약 10시간을 유다시티 나노디그리 학습에 할애해야 합니다.

영어 능력은 필수인가요?

영어 능력이 필수는 아니지만 학습 자료는 영어로 제공됩니다. 학습을 성공적으로 완료하려면 영어 읽기 및 쓰기 능력이 원활해야 합니다.

인종/성별 목록에서 나에게 해당하는 옵션을 찾지 못하면 어떻게 하나요?

두 해당 항목에 자가 식별하고 싶음(선택하여 입력) 옵션을 사용할 수 있습니다.

AWS 플레이어 계정이란 무엇입니까?

AWS 플레이어 계정은에서 생성한 AWS DeepRacer 다중 사용자 및 AWS DeepRacer Student를 위한 관리형 자격 증명 솔루션입니다 AWS. AWS 플레이어 계정은 이러한 각 AWS 서비스에서 생성된 모든 리소스를 보유합니다.

지원되는 서비스에 대한 AWS 플레이어 계정 생성

AWS DeepRacer 다중 사용자 또는 AWS DeepRacer Student에 대한 계정을 생성하면 AWS 플레이어 계정이 자동으로 생성됩니다. 이러한 서비스에서 다양한 기능을 사용하면 새 리소스가 AWS 플레이어 계정에 자동으로 추가됩니다. AWS DeepRacer 다중 사용자 및 AWS DeepRacer Student를 시작하려면 다음 링크를 사용합니다.

AWS DeepRacer Student 계정 생성

AWS DeepRacer Student를 사용하려면 계정을 생성하여 시작합니다. 계정을 만드는 방법을 알아보려면 AWS DeepRacer Student 사용 안내서 1단계: AWS DeepRacer Student에 가입 항목을 참조하세요.

AWS DeepRacer 다중 사용자를 사용하여 한 계정에서 여러 참가자를 후원합니다.

AWS DeepRacer 다중 사용자 모드는 관리자 및 참가자라는 두 가지 사용자 프로필을 지원합니다. 두 가지 모두 설정 요구 사항이 다릅니다. 시작하려면 AWS DeepRacer 개발자 안내서의 다중 사용자 모드를 참조하세요.

AWS 플레이어 계정 삭제

AWS 플레이어 계정을 삭제하면 지원되는 모든 서비스에 대한 액세스 권한이 즉시 상실됩니다. 여기에는 획득한 모든 성과(배지, 점수, 아바타 등)가 포함됩니다.

AWS 플레이어 계정 계정을 삭제해도 AWS 계정은 삭제되지 않습니다. 계정을 삭제하려면 AWS 계정 해지에 설명된 단계를 사용합니다 AWS.

AWS 플레이어 계정 계정을 사용하여 AWS DeepRacer 다중 사용자에서 이벤트를 생성한 경우 AWS 플레이어 계정 계정을 삭제할 수 없습니다. 여러분이 만든 이벤트의 참가자가 불편한 경험을 하지 않도록 하기 위한 것입니다. 관리자가 AWS DeepRacer 다중 사용자 모드에서 이벤트를 생성하는 방법에 대해 자세히 알아보려면 다음 주제를 사용하세요.

AWS DeepRacer 다중 사용자 모드를 사용하여 이벤트 설정(관리자)

다중 사용자 모드를 사용하여 이벤트를 생성하는 방법을 알아보려면 AWS DeepRacer 개발자 안내서의 다중 사용자 모드 설정(관리자)을 참조하세요.

AWS 플레이어 계정은 서비스 계정에서 생성된 AWS 리소스 이외의 리소스에 액세스할 수 없습니다. 서비스 계정의 모든 AWS Identity and Access Management 정책 및 관련 리소스는 필요한 리소스로만 제한됩니다.

AWS DeepRacer Student 시작하기

이 자습서를 사용하여 AWS DeepRacer Student를 시작합니다. 이 자습서에서는 AWS DeepRacer Student에 로그인하고, 프로필을 업데이트하고, AWS AI 및 ML 장학금을 고려하도록 옵트인하고, 기계 학습(ML) 및 강화 학습(RL)에서 무료 과정을 수강하고, AWS DeepRacer 모델을 생성하는 방법을 설명합니다. 계정을 생성할 때 장학금 신청에 동의하지 않았더라도 나중에 프로필을 업데이트할 때 동의할 수 있습니다.

주제

- 사전 조건
- 1단계: AWS DeepRacer Student에 가입
- 2단계: AWS DeepRacer Student 가입 완료
- 3단계: (선택 사항) AWS AI 및 ML 장학금 신청에 관하여 읽고 동의하기
- 4단계: 내 프로필 업데이트 하기
- 5단계: 홈 페이지에서 Explore AWS DeepRacer Student

사전 조건

AWS DeepRacer Student에 액세스하고 참여하려면 다음이 필요합니다.

- 현재 고등학교, 2년제 또는 4년제 대학에 재학 중인 16세 이상의 학생이어야 합니다.
- 또는 고등학교, 2년제 또는 4년제 대학 학생들을 위한 교육자 또는 이벤트 주최자여야 합니다.
- 유효한 이메일 주소가 있어야 합니다.

1단계: AWS DeepRacer Student에 가입

이 절차에 제공된 URL을 사용하여 AWS DeepRacer Student에 가입할 수 있습니다. 가입할 때 AWS 플레이어 계정을 생성합니다. 이 계정은 다른 특정 AWS 서비스와 함께 사용할 수 있습니다. AWS Player 계정이 이미 있는 경우, 해당 계정을 AWS DeepRacer Student에서 사용할 수 있습니다.

1. http://deepracerstudent.com/ 랜딩 페이지를 여세요.
2. Get started를 선택합니다.
3. 로그인 페이지에서 AWS 플레이어 계정이 아직 없는 경우 가입을 선택합니다.

> **Note**
> AWS 플레이어 계정이 이미 있는 경우 여기에 정보를 입력합니다. AWS 플레이어 계정에 대한 자세한 내용은 섹션을 참조하세요. AWS 플레이어 계정이란 무엇입니까?

4. 로그인 페이지에서 다음 정보를 입력합니다:
   - 이메일 주소
   - 암호
5. 가입을 선택합니다. 지정한 이메일 주소로 확인 코드가 포함된 이메일이 전송됩니다.
6. 팝업이 나타나면 인증 번호를 입력하고 확인을 선택합니다.
7. AWS DeepRacer Student 로그인 페이지에서 이메일 주소와 암호를 입력하고 로그인을 선택합니다.
8. AWS DeepRacer Student 시작 팝업에서 가입 완료를 선택합니다.

> **Note**
> 나중에 할게요를 선택할 수도 있습니다. 나중에 가입하려면 일단은 로그아웃합니다.

2단계: AWS DeepRacer Student 가입 완료

섹션을 완료하여 AWS DeepRacer Student 계정을 생성합니다. 별도의 설명이 없는 한 모든 입력란은 필수입니다.

1. AWS DeepRacer Student 계정 생성하기 섹션의 입력란에 개인 정보를 추가하여 내 계정을 생성하세요.
2. 확인란을 선택하여 고등학교, 2년제 또는 4년제 대학에 재학 중인 학생임을 증명하세요.

> **Note**
> 고등학생인 경우 다음과 같이 합니다:
>
> - 학교에서 기타를 선택합니다. 그런 다음 학교 이름 입력에 내가 다니는 고등학교 이름을 추가합니다.

- 현재 또는 장래 희망 전공란에는 미정을 선택하거나 목록에서 장래 희망 전공을 선택합니다.

3단계: (선택 사항) AWS AI 및 ML 장학금 신청에 관하여 읽고 동의하기

AWS AI 및 ML 장학금 프로그램에 지원하고 싶으신가요? 섹션에서 AWS AI 및 ML 장학금 및 장학금 대상자에 관한 정보를 읽어보세요.

1. AWS AI 및 ML 장학금 기준을 충족하는 경우, 장학금 수혜 대상에 고려되는 데 동의할 수 있습니다. 확인란을 선택하여 장학금 수혜 고려에 동의한다는 것을 확인합니다.

> **Note**
>
> 장학금 수혜 대상이 되고 싶지 않거나 아직 결정하지 않은 경우 확인란을 선택하지 않은 상태로 두고 나중에 할게요를 선택하십시오. 일단 로그아웃.을 선택해 로그아웃하거나 제출을 선택해 동의를 진행합니다. 프로필을 업데이트할 때도 동의할 수 있습니다.

2. (선택 사항) 드롭다운 목록을 사용하여 성별 선택 및 인종 선택 (미국 참가자만 해당) 입력란에 정보를 입력합니다.
3. 제출을 선택합니다.

4단계: 내 프로필 업데이트 하기

내 프로필을 업데이트하려면 내 프로필 페이지를 이용하세요. AWS AI 및 ML 장학금 프로그램에 동의할 수도 있습니다. AWS DeepRacer Student League에서 성과에 대한 업적을 받은 경우 프로필 페이지에서 소셜 미디어에 공유할 수 있습니다.

내 프로필을 업데이트하려면

1. In AWS DeepRacer Student의 왼쪽 탐색 창에서 프로필을 선택합니다.
2. 내 프로필 페이지의 레이서 이름에서 내 레이서 이름 변경을 선택합니다.

> **Note**
> 레이서 이름은 2~24자 사이에서 입력할 수 있습니다. 문자, 숫자, 붙임표(-)가 허용됩니다.

3. 레이서 이름 모달에 내 레이서 이름을 입력하고 저장을 선택합니다. 레이서 이름을 바꾸고 싶지 않으면 취소를 선택합니다.
4. 내 프로필 정보 섹션에서 다음 입력란 정보를 수정할 수 있습니다:
   - 명칭
   - 레이서 이름
   - 학교 이름
   - 전공 이름
   - 졸업 연도
   - 거주 국가

   > **Note**
   > AWS DeepRacer Student League 가상 레이스에 참가하고 상품을 받으려면 거주 국가를 선택해야 합니다. 특정 레이스에 내 모델을 제출하면 해당 레이싱 시즌 동안은 거주 국가를 바꿀 수 없습니다.

5. (선택 사항) AWS AI 및 ML 장학금 프로그램에 고려되고 싶으신가요? 섹션에서 장학금 및 신청 방법에 대한 정보를 볼 수 있습니다.
   a. 장학금 프로그램에 수혜 대상으로 고려되는 데 동의하려면 확인란을 선택합니다.
   b. (선택 사항) 성별 선택 및 인종 선택 입력란에 정보를 입력합니다.
   c. 제출을 선택합니다.

5단계: 홈 페이지에서 Explore AWS DeepRacer Student

AWS DeepRacer Student 홈 페이지는 AWS DeepRacer Student가 제공해야 하는 모든 항목을 탐색할 수 있는 완벽한 장소입니다. 홈 페이지에서는 다음을 수행할 수 있습니다:

기계 학습(ML) 의 기초 학습 시작하기

AWS DeepRacer Student의 자세히 알아보기 섹션에서 제공되는 무료 강좌를 이용할 수 있습니다. 이 강력한 제품은 AWS DeepRacer Student를 사용한 기계 학습 여정의 기반을 구축하는 데 도움이 됩니다.

기계 학습 지식 실습하기

학습 과정을 사용하여 시간을 보낸 후에는 AWS DeepRacer 모델을 생성하고 훈련할 준비가 되었습니다. 자세한 내용은 AWS DeepRacer Student에서 강화 학습 모델 훈련 단원을 참조하십시오.

in AWS DeepRacer Student 레이스에서 경쟁

첫 번째 AWS DeepRacer 모델 훈련을 마치면 레이스에 참가할 준비가 된 것입니다. 학생 리그 월간 레이스에 참가하려면 오픈 레이스 섹션으로 가서 레이스 참가를 선택하세요. 레이싱을 완료한 후에는 내 모델이 어땠는지 리더보드에서 확인하고 모델을 변경하고 개선하는 데 필요한 데이터도 확인하세요. 월간 가상 레이스에서 경쟁하면 점수를 획득할 수 있습니다. 이 점수로 전국 및 지역 시즌 순위가 결정됩니다. 매월 상위 3명의 레이서가 메달과 상품을 획득하게 됩니다. 자세한 내용은 AWS DeepRacer Student 레이스 참가 섹션을 참조하세요.

모델 훈련 시간 확인하기

모델을 훈련하고 복제할 때는 무료 모델 훈련 시간을 일부 사용합니다. 홈페이지에서 남은 모델 훈련 시간을 확인할 수 있습니다.

AWS DeepRacer Student 레이싱 일정 확인

레이싱 일정을 보고 레이스 날을 위한 계획을 세워보세요.

AWS AI 및 ML 장학금에 관해 자세히 알아보기

AWS AI 및 ML 장학금에 관해 알아보고 준비 방법을 확인하세요. 자세한 내용은 AWS AI 및 ML 장학금 프로그램이란 무엇인가요? 단원을 참조하세요.

기타 리소스 보기

Discord 채널 및 AWS DeepRacer 웹 사이트와 같이 AWS DeepRacer Student를 탐색하는 데 도움이 되는 다른 리소스를 찾을 수 있습니다. 이러한 리소스를 통해 팁과 인사이트를 공유하는 레이서 및 팬 커뮤니티에 연결될 수 있습니다.

AWS DeepRacer Student에서 강화 학습 모델 훈련

이 연습에서는 AWS DeepRacer Student에서 첫 번째 모델을 훈련하는 방법을 보여줍니다. 또한 경험을 최대한 활용하고 학습 속도를 높이는 데 도움이 되는 몇 가지 유용한 팁도 제공합니다.

1단계: AWS DeepRacer Student를 사용하여 강화 학습 모델 훈련

모델 생성 버튼을 찾을 수 있는 위치를 알아보고 첫 번째 모델 훈련을 시작하여 AWS DeepRacer Student에서 여정을 시작하세요. 모델 생성과 훈련은 반복적인 프로세스라는 점을 염두에 두세요. 다양한 알고리즘과 보상 함수를 실험해 보고 최상의 결과를 얻으세요.

강화 학습 모델을 훈련하려면

1. AWS DeepRacer Student 홈 페이지에서 모델 생성을 선택합니다. 또는 왼쪽 탐색 창에서 내 모델로 이동할 수도 있습니다. 모델 페이지의 내 모델에서 모델 생성을 선택합니다.
2. 개요 페이지에서 강화 모델을 훈련하는 방법에 관해 읽어보세요. 이 페이지에는 프로세스의 각 단계가 설명되어 있습니다. 마쳤으면 다음을 선택합니다.

2단계: 내 모델 이름 지정하기

내 모델의 이름을 지어 주세요. 모델을 개선하고 복제하려는 경우, 개별 모델을 빠르게 찾을 수 있도록 모델에 고유한 이름을 붙이는 것이 좋습니다. 예를 들면, 이니셜-날짜-버전과 같이 일관된 이름 지정 규칙에 따라 모델에 이름을 붙일 # 있습니다.

내 모델 이름을 지정하려면

1. 내 모델 이름 지정 페이지의 모델 이름 입력란에 이름을 입력합니다.

   > **Note**
   >
   > 모델 학습을 시작하면 모델 이름이 고정되어 더 이상 변경할 수 없게 됩니다.

2. 다음을 선택합니다.

3단계: 트랙 선택하기

시뮬레이션 트랙을 선택하세요. 트랙은 환경 역할을 하며 차량에 데이터를 제공합니다. 매우 복잡한 트랙을 선택하면 차량의 총 훈련 시간이 더 오래 걸리고 사용하는 보상 함수도 더 복잡해집니다.

내 트랙(환경)을 선택하려면

1. 트랙 선택 페이지에서 차량의 훈련 환경으로 사용할 트랙을 선택합니다.
2. Next(다음)를 선택합니다.

4단계: 알고리즘 선택하기

AWS DeepRacer Student에는 선택할 수 있는 두 가지 훈련 알고리즘이 있습니다. 각 알고리즘은 다른 방식으로 보상을 극대화합니다. AWS DeepRacer Student 경험을 최대한 활용하려면 두 알고리즘을 모두 실험해 보세요. 알고리즘에 관한 자세한 내용은 AWS DeepRacer 훈련 알고리즘 항목을 참조하세요.

훈련 알고리즘을 선택하려면

1. 알고리즘 유형 선택 페이지에서 알고리즘 유형을 선택합니다. 다음 두 가지 알고리즘 유형을 사용할 수 있습니다.
   - 근접 정책 최적화(PPO). 안정적이지만 데이터를 많이 사용하는 이 알고리즘은 훈련을 반복할 때 일관되게 실행됩니다.
   - 소프트 액터 크리틱(SAC). 불안정하지만 데이터 효율성이 높은 이 알고리즘은 훈련을 반복할 때 성능이 일관되지 않을 수 있습니다.
2. Next(다음)를 선택합니다.

5단계: 보상 함수 사용자 지정하기

보상 함수는 강화 학습의 핵심 부분으로. 이를 사용하여 차량(에이전트)가 이 트랙(환경)을 탐색해 가면서 특정 동작을 취하도록 장려하세요. 반려동물의 특정 행동을 장려하고 억제하는 것과 마찬가지로 이 도구를 사용하여 차량이 최대한 빨리 랩을 마치도록 장려하고 트랙을 벗어나거나 지그재그로 주행하지 않게 할 수 있습니다.

첫 모델을 훈련할 때는 기본 설정 샘플 보상 함수를 사용하는 것이 좋습니다. 모델을 실험하고 최적화할 준비가 되면 코드 편집기에서 코드를 편집하여 보상 함수를 사용자 지정할 수 있습니다. 보상 함수를 사용자 지정하는 방법에 관한 자세한 정보는 보상 함수 사용자 지정하기 항목을 참조하세요.

보상 함수를 사용자 지정하려면

1. 보상 함수 사용자 지정 페이지에서 샘플 보상 함수를 선택합니다. 사용자 지정할 수 있는 세 가지 샘플 보상 함수가 있습니다.

   - 중앙선을 따라가세요. 차량이 트랙 중앙선에 최대한 가깝게 자율 주행하면 보상을 주세요.
   - 경계선 안에서 주행하게 하세요. 네 바퀴가 모두 트랙 경계 안에 있는 상태로 자율 주행할 때 차량에게 보상을 주세요.
   - 지그재그를 방지하세요. 차량이 중앙선에 가까이 붙어 주행하면 보상을 주세요. 조향 각도가 크거나 트랙을 벗어나면 차량에 벌점을 주세요.

   > **Note**
   > 보상 함수를 사용자 지정하지 않으려면 다음을 선택하세요.

2. (선택 사항) 보상 함수 코드를 수정하세요.

   - 샘플 보상 함수를 하나 선택하고 이 코드를 설명해주세요를 선택합니다.
   - 코드의 각 섹션에서 + 선택하면 설명 텍스트가 포함된 팝업 텍스트 상자가 표시되고 자세한 정보를 볼 수 있습니다. 각 팝업에서 다음을 선택하여 코드 설명을 진행하세요. 팝업 텍스트 상자를 종료하려면 모서리에 있는 X를 선택합니다. 설명을 종료하려면 완료를 선택합니다.

     > **Note**
     > 샘플 보상 함수 코드를 편집하지 않으려면 기본 설정 코드로 진행을 선택하면 됩니다.

   - 선택 사항으로 샘플 보상 함수를 선택하고 샘플 코드 편집을 선택하여 샘플 보상 함수를 편집할 수도 있습니다. 코드를 편집하고 확인을 선택하여 코드를 확인합니다. 코드를 확인하지 못하거나 코드를 원래 상태로 재설정하려면 재설정을 선택합니다.

3. Next(다음)를 선택합니다.

6단계: 기간을 선택하고 리더보드에 모델 제출하기

모델 학습 기간은 모델의 성능에 영향을 미칩니다. 훈련 초기 단계에서 실험할 때는 이 파라미터를 작은 값으로 시작한 다음 점진적으로 훈련 기간을 늘려야 합니다.

내 모델을 훈련하는 이 단계에서는 내 훈련된 모델이 리더보드에 제출됩니다. 확인란의 선택을 해제하여 제출 동의를 거부할 수도 있습니다.

기간을 선택하고 리더보드에 모델을 제출하려면

1. 기간 선택 페이지의 모델 훈련 기간 선택에서 시간을 선택합니다.
2. 모델 설명 입력란에 내가 선택한 내용을 기억하는 데 도움이 되도록 모델에 대한 유용한 설명을 입력합니다.

   > **Tip**
   > 보상 함수 및 알고리즘에 대한 현재 선택 및 수정 사항, 모델이 어떻게 작동할지에 대한 가설 등 모델에 대한 정보를 추가하는 것이 좋습니다.

3. 훈련이 완료된 후 모델을 AWS DeepRacer Student 리더보드에 자동으로 제출하려면 확인란을 선택합니다. 확인란의 선택을 해제하여 모델 제출을 거부할 수도 있습니다.

   > **Tip**
   > 리더보드에 모델을 제출하는 것을 권장합니다. 모델을 제출하면 내 모델이 다른 모델과 어떻게 비교되는지 확인하고 피드백을 받아 모델을 개선할 수 있습니다.

4. 내 모델 훈련을 선택합니다.
5. 모델 훈련 초기화 팝업 창에서 확인을 선택합니다.
6. 훈련 구성 페이지에서 모델의 훈련 상태 및 구성을 검토할 수 있습니다. 훈련 상태가 진행 중일 때는 선택한 트랙에서의 모델 훈련 동영상을 볼 수도 있습니다. 동영상을 시청하면 모델을 개선하는 데 사용할 수 있는 귀중한 인사이트를 얻는 데 도움이 될 수 있습니다.

7단계: 리더보드에서 모델 성능 보기

모델을 훈련하고 리더보드에 제출한 후에는 성능을 확인할 수 있습니다.

모델 성능을 보려면

1. 왼쪽 탐색 창에서 경쟁을 찾아 펼칩니다. 시즌을 선택하세요. 리더보드 페이지에서 내 모델과 순위가 한 섹션에 표시됩니다. 이 페이지에는 제출된 모델 목록, 레이싱 세부 정보, 레이싱 세부 정보 섹션을 담은 리더보드 섹션도 있습니다.
2. 리더보드가 표시된 페이지에서 내 프로필 섹션으로 가서 동영상 보기를 선택하여 모델의 성능이 담긴 동영상을 볼 수 있습니다.

8단계: 복제를 사용하여 모델 개선하기

모델을 훈련하고 원하는 경우 리더보드에 제출한 후에는 모델을 복제하여 개선할 수 있습니다. 모델을 복제하면 이전에 훈련한 모델을 새 모델의 출발점으로 사용하여 단계를 덜 거치면서 훈련 효율성을 높일 수 있습니다.

모델 복제 및 개선하기

1. In AWS DeepRacer Student의 왼쪽 탐색 창에서 모델로 이동합니다.
2. 내 모델 페이지에서 모델을 하나 선택하고 복제를 선택합니다.
3. 내 모델 이름 지정 입력란에 복제된 모델의 새 이름을 입력하고 다음을 선택합니다.
4. 보상 함수를 사용자 지정하기 페이지에서 보상 함수를 사용자 지정하고 다음을 선택합니다. 보상 함수를 사용자 지정하는 방법에 관한 자세한 정보는 5단계: 보상 함수 사용자 지정하기 항목을 참조하세요.
5. 기간 선택 페이지에서 모델 훈련 기간 선택 입력란에 시간을 입력하고 모델 설명 입력란에 설명을 입력한 다음 확인란을 선택하여 복제된 모델을 리더보드에 제출합니다.
6. 내 모델 훈련을 선택합니다. 훈련이 초기화됩니다. 복제된 모델에 관한 정보가 포함된 훈련 구성 페이지가 나타납니다. 훈련 상태가 진행 중일 때는 선택한 트랙에서의 모델 훈련 동영상을 볼 수도 있습니다.
7. 사전 훈련된 모델을 계속 복제하고 수정하여 리더보드에서 최고의 성과를 달성하세요.

9단계: (선택 사항) 모델 다운로드하기

모델을 훈련하고 선택적으로 리더보드에 제출한 후 나중에 AWS DeepRacer 물리적 디바이스에서 사용할 수 있도록 모델을 다운로드할 수 있습니다. 모델은 `.tar.gz` 파일로 저장됩니다.

모델을 다운로드하려면

1. In AWS DeepRacer Student의 왼쪽 탐색 창에서 모델로 이동합니다.
2. 내 모델 페이지에서 모델을 하나 선택하고 다운로드를 선택합니다.
3. 브라우저에서 모델 다운로드 진행 상황을 볼 수 있습니다. 모델이 다운로드되면 로컬 하드 드라이브나 기타 선호하는 저장 장치에 저장할 수 있습니다.

   AWS DeepRacer 디바이스 작업에 대한 자세한 내용은 AWS DeepRacer 안내서의 Operation YourDeepRacer Vehicle을 참조하세요. AWS DeepRacer

AWS DeepRacer Student 레이스 참가

시뮬레이션에서 모듈 훈련과 평가를 성공적으로 마쳤으면 레이스에 참가하여 자신의 모델 성능을 다른 레이서의 모델과 비교할 수 있습니다. 레이싱은 모델에 대한 피드백을 얻고, 상과 상품을 얻고, 다른 AWS DeepRacer Student 커뮤니티 멤버를 가상으로 만나고, 기술을 배우고 개선할 수 있는 기회에 대해 듣고, 즐거운 시간을 보낼 수 있는 좋은 방법입니다. 학생 레이스에는 학생 리그와 커뮤니티라는 두 가지 유형이 있습니다.

학생 리그 레이스는 모든 학생이 참가할 수 있는 월간 가상 대회입니다. 학생 커뮤니티 레이스는 학생이 초대를 통해서만 참가할 수 있는 AWS 콘솔의 교육자 또는 이벤트 주최자가 만든 프라이빗 레이스입니다.

이 섹션에서는 AWS DeepRacer Student 학생 리그 레이스 및 학생 커뮤니티 레이스에 참여하는 방법을 설명합니다.

학생 리그 레이스 참가하기

이 섹션에서는 모델을 AWS DeepRacer Student 학생 리그 레이스에 제출하는 방법을 알아봅니다. 훈련된 모델을 학생 리더보드에 바로 제출하여 매달 레이스에 참가할 수 있습니다. 모델 훈련에 관한 자세한 정보는 AWS DeepRacer Student에서 강화 학습 모델 훈련 항목을 참조하세요.

학생 리그 레이스에 참가하려면

1. AWS DeepRacer Student의 왼쪽 탐색 창에서 경쟁을 확장하고 Student 리그를 선택합니다.
2. 이 페이지에는 레이스 세부 정보와 리더보드가 표시됩니다. 공개 레이스 섹션에서 레이스 참가를 선택해 모델을 제출하거나, 모델을 아직 생성하지 않았으면 모델 생성을 선택합니다. 자세한 내용은 AWS DeepRacer Student에서 강화 학습 모델 훈련 단원을 참조하십시오.
3. 레이스할 모델 선택 섹션에서 드롭다운 목록을 사용하여 모델 선택 입력란에서 모델을 하나 선택합니다.
4. 레이스 참가를 선택하여 내 모델을 제출합니다.
5. 내 모델이 레이싱 기준에 따라 성공적으로 평가되면 리더보드에서 다른 참가자와 비교하여 내 모델의 순위를 확인할 수 있습니다.
6. 선택 사항으로 보기를 선택하여 내 차량의 성능을 동영상으로 볼 수도 있습니다.
7. 다른 모델로 참가하려면 다시 레이스를 선택하세요. 각 AWS DeepRacer Student League 레이스의 제출 할당량은 50입니다.

학생 커뮤니티 레이스 참가하기

이 섹션에서는 모델을 AWS DeepRacer Student 학생 커뮤니티 레이스에 제출하는 방법을 알아봅니다. 교육자 또는 이벤트 주최자로부터 이메일을 통해 초대 링크를 받아 학생 커뮤니티 레이스에 참여할 수 있습니다.

학생 커뮤니티 레이스에 참여하려면

1. 초대 링크로 이동하여 AWS DeepRacer 학생 계정에 로그인합니다.
2. 로그인한 후 레이스 참가 버튼을 선택합니다.
3. 레이스할 모델 선택 드롭다운 목록에서 커뮤니티 레이스에서 사용할 모델을 선택합니다.
4. 레이스 참가 선택해 내 모델을 제출합니다.
5. 내 모델이 레이싱 기준에 따라 성공적으로 평가되면 리더보드에서 다른 참가자와 비교하여 내 모델의 순위를 확인할 수 있습니다.
6. 선택 사항으로 보기를 선택하여 내 차량의 성능을 동영상으로 볼 수도 있습니다.
7. 다른 모델로 참가하려면 다시 레이스를 선택하세요.

보상 함수 사용자 지정하기

보상 함수를 생성하는 것은 인센티브 플랜을 설계하는 것과 같습니다. 파라미터는 인센티브 플랜을 개발하는 데 사용할 수 있는 값입니다.

인센티브 전략이 다르면 차량의 동작도 달라질 수 있습니다. 차량이 더 빨리 주행하도록 장려하려면 차량이 한 바퀴를 완주하는 데 너무 오래 걸리거나 트랙을 벗어날 경우 음수 값을 부여해 보세요. 지그재그 주행 패턴을 피하려면 조향 각도 범위 제한을 정의하고 트랙의 직선 구간에서 덜 공격적으로 조향하면 차량에 보상을 제공하세요.

트랙의 중앙선과 안쪽, 바깥쪽 가장자리를 따라 배치된 번호가 매겨진 마커인 웨이포인트를 사용하여 특정 주행 동작을 직선 도로, 커브 같은 트랙의 특성과 연관시킬 수 있습니다.

효과적인 보상 함수를 만드는 것은 창의적이고 반복적인 프로세스입니다. 다양한 전략을 시도하고 파라미터를 여러 가지로 조합해보세요. 가장 중요한 건 즐기는 것입니다!

주제

- Python 코드를 편집하여 보상 함수를 사용자 지정하세요
- AWS DeepRacer 보상 함수의 입력 파라미터

Python 코드를 편집하여 보상 함수를 사용자 지정하세요

In AWS DeepRacer Student에서는 샘플 보상 함수를 편집하여 모델에 맞는 사용자 지정 레이싱 전략을 만들 수 있습니다.

보상 함수를 사용자 지정하려면

1. AWS DeepRacer Student 모델 생성 경험의 5단계: 보상 함수 사용자 지정 페이지에서 샘플 보상 함수를 선택합니다.
2. 샘플 보상 함수 선택기 아래의 코드 편집기를 사용하여 Python 코드로 보상 함수의 입력 파라미터를 사용자 지정합니다.
3. 유효성 검사를 선택하여 코드가 작동하는지 여부를 확인합니다. 또는 재설정을 선택하여 다시 시작할 수도 있습니다.
4. 변경을 완료했으면 다음을 선택합니다.

AWS DeepRacer 보상 함수의 입력 파라미터을 사용해 각 파라미터에 관해 알아볼 수 있습니다. 보상 함수 예제에서 다양한 파라미터가 어떻게 쓰이는지 살펴보세요.

AWS DeepRacer 보상 함수의 입력 파라미터

AWS DeepRacer 보상 함수는 변수 로 전달되는 사전 객체를 입력`params`으로 사용합니다.

```
def reward_function(params) :

    reward = ...

    return float(reward)
```

`params` 딕셔너리 객체에는 다음과 같은 키-값 페어가 저장됩니다.

```
{
    "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

}
```

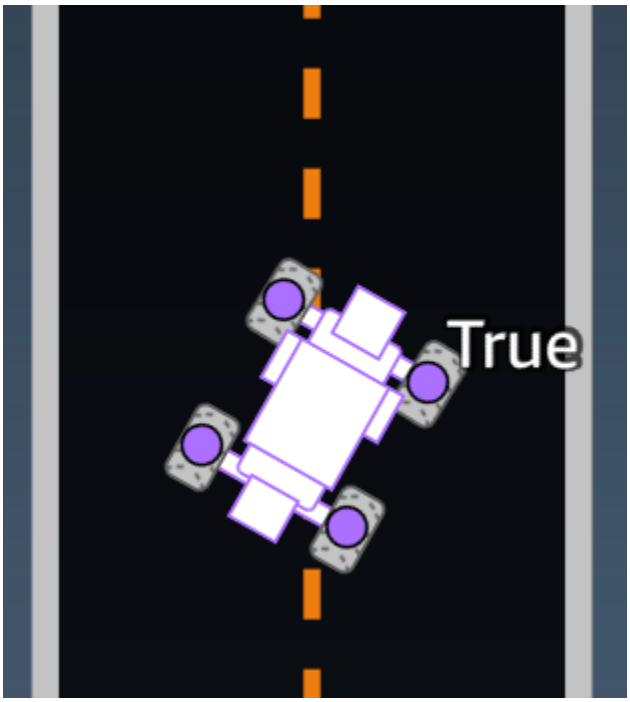
다음 참조를 사용하여 AWS DeepRacer 입력 파라미터를 더 잘 이해할 수 있습니다.

all_wheels_on_track

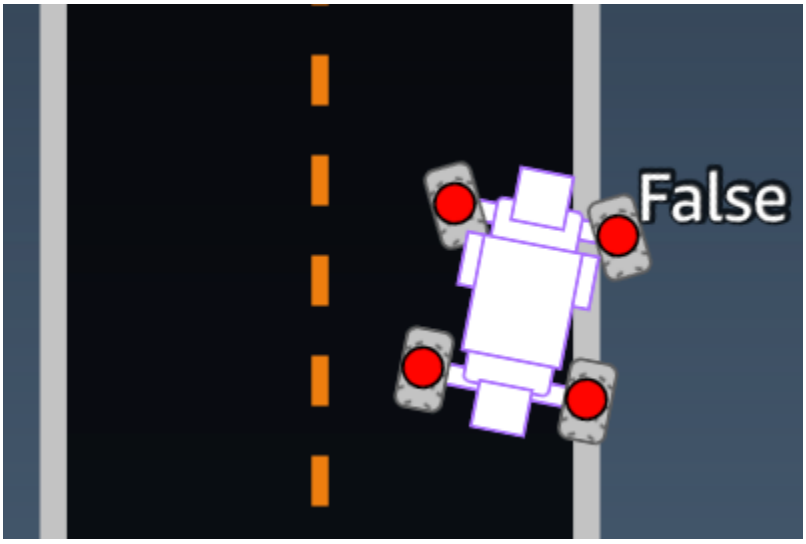
유형: `Boolean`

범위: `(True:False)`

에이전트의 트랙 주행 또는 트랙 이탈을 나타내는 `Boolean` 플래그입니다. 바퀴 하나라도 트랙 경계를 벗어나면 트랙 이탈(`False`)입니다. 네 바퀴가 안쪽과 바깥쪽 트랙 경계 안에 있으면 트랙 주행(`True`)입니다. 다음 그림은 에이전트가 트랙을 따라 주행하는 것을 나타냅니다.



다음 그림은 두 바퀴가 트랙 경계 밖에 있어서 트랙 이탈인 에이전트를 보여줍니다.



예제: *all_wheels_on_track* 파라미터를 사용하는 보상 함수

```
def reward_function(params):
    ################################################################################
    '''
    Example of using all_wheels_on_track and speed
    '''

    # Read input variables
    all_wheels_on_track = params['all_wheels_on_track']
    speed = params['speed']

    # Set the speed threshold based your action space
    SPEED_THRESHOLD = 1.0

    if not all_wheels_on_track:
        # Penalize if the car goes off track
        reward = 1e-3
    elif speed < SPEED_THRESHOLD:
        # Penalize if the car goes too slow
        reward = 0.5
    else:
        # High reward if the car stays on track and goes fast
        reward = 1.0

    return float(reward)
```

closest_waypoints

유형: `[int, int]`

범위: `[(0:Max-1),(1:Max-1)]`

에이전트의 현재 위치인 `(x, y)`에 가장 가깝게 인접한 두 `waypoint`의 제로 기반 인덱스입니다. 거리는 에이전트 중앙에서 유클리트(Eudlidean) 거리로 측정됩니다. 첫 번째 요소는 에이전트 뒤에서 가장 가까운 중간 지점을 나타내고, 두 번째 요소는 에이전트 앞에서 가장 가까운 중간 지점을 나타냅니다. `Max`는 중간 지점 목록의 길이입니다. 중간 지점의 그림에서 `closest_waypoints`는 `[16, 17]`이게 됩니다.

다음 보상 함수 예제는 `waypoints`, `closest_waypoints` 및 `heading`을 사용해 즉각적인 보상을 계산하는 방법을 나타낸 것입니다.

AWS DeepRacer는 `math`, , `random`, numpy `scipy`및 Python 라이브러리를 지원합니다`shapely`. 하나를 사용하려면 함수 정의 `def reward_function(params)` 앞에 가져오기`import` *`supported library`*를 추가합니다.

예: *`closest_waypoints`* 파라미터를 사용하는 보상 함수.

```
# Place import statement outside of function (supported libraries: math, random, numpy,
 scipy, and shapely)
# Example imports of available libraries
#
# import math
# import random
# import numpy
# import scipy
# import shapely

import math

def reward_function(params):
    ################################################################################
    '''
    Example of using waypoints and heading to make the car point in the right direction
    '''

    # Read input variables
    waypoints = params['waypoints']
    closest_waypoints = params['closest_waypoints']
    heading = params['heading']
```

```
    # Initialize the reward with typical value
    reward = 1.0

    # Calculate the direction of the centerline based on the closest waypoints
    next_point = waypoints[closest_waypoints[1]]
    prev_point = waypoints[closest_waypoints[0]]

    # Calculate the direction in radius, arctan2(dy, dx), the result is (-pi, pi) in
radians
    track_direction = math.atan2(next_point[1] - prev_point[1], next_point[0] -
prev_point[0])
    # Convert to degree
    track_direction = math.degrees(track_direction)

    # Calculate the difference between the track direction and the heading direction of
the car
    direction_diff = abs(track_direction - heading)
    if direction_diff > 180:
        direction_diff = 360 - direction_diff

    # Penalize the reward if the difference is too large
    DIRECTION_THRESHOLD = 10.0
    if direction_diff > DIRECTION_THRESHOLD:
        reward *= 0.5

    return float(reward)
```

closest_objects

유형: `[int, int]`

범위: `[(0:len(object_locations)-1), (0:len(object_locations)-1]`

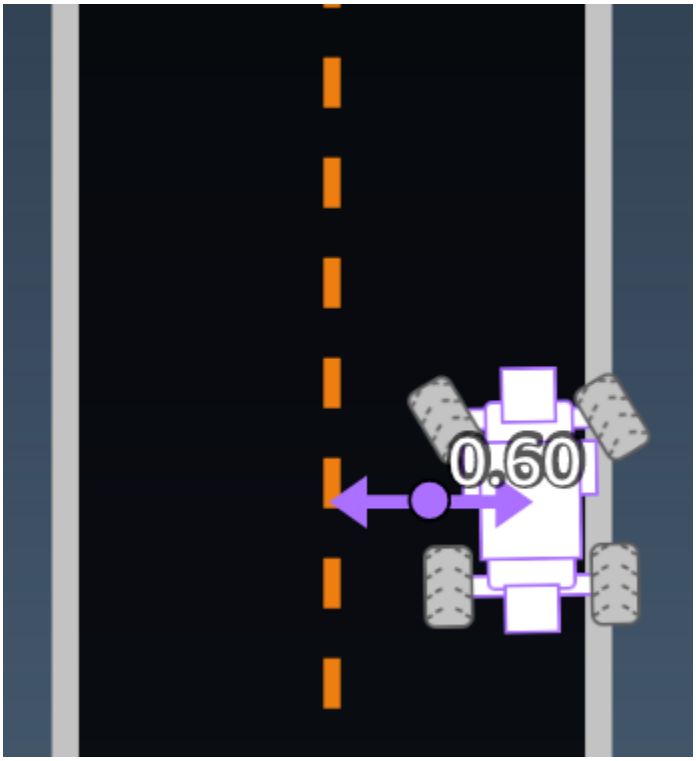
에이전트의 현재 위치(x, y)에 가장 가까운 두 객체의 인덱스(0부터 시작)입니다. 첫 번째 인덱스는 에이전트 뒤에서 가장 가까운 객체를 참조하고 두 번째 인덱스는 에이전트 앞에서 가장 가까운 객체를 참조합니다. 객체가 하나만 있는 경우 두 인덱스는 모두 0이 됩니다.

distance_from_center

유형: `float`

범위: `0:~track_width/2`

에이전트의 중앙과 트랙의 중앙 사이의 변위(미터)입니다. 에이전트의 바퀴 중 하나라도 트랙 경계를 벗어났을 때 최대 변위가 관측될 수 있으며, 이때 최대 변위는 트랙 경계의 너비에 따라 다르지만 `track_width`의 절반보다 약간 작거나 클 수 있습니다.



예: *`distance_from_center`* 파라미터를 사용하는 보상 함수

```
def reward_function(params):
    ################################################################################
    '''
    Example of using distance from the center
    '''

    # Read input variable
    track_width = params['track_width']
    distance_from_center = params['distance_from_center']

    # Penalize if the car is too far away from the center
    marker_1 = 0.1 * track_width
    marker_2 = 0.5 * track_width

    if distance_from_center <= marker_1:
        reward = 1.0
```

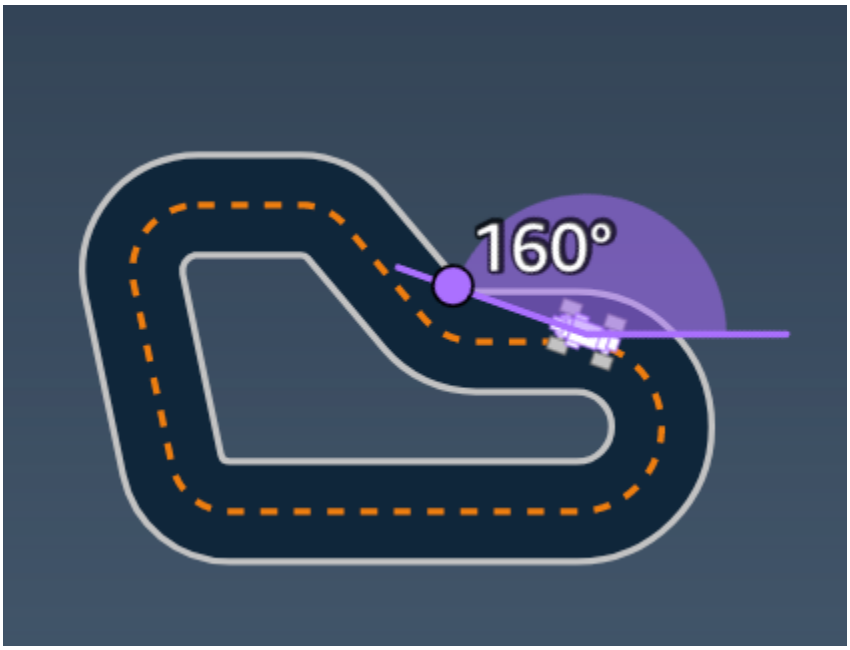
```
    elif distance_from_center <= marker_2:
        reward = 0.5
    else:
        reward = 1e-3  # likely crashed/ close to off track

    return float(reward)
```

heading

유형: `float`

범위: `-180:+180`

좌표계의 x축에 대한 에이전트 진행 방향(각도)입니다.



예: *heading* 파라미터를 사용하는 보상 함수

자세한 내용은 `closest_waypoints` 단원을 참조하십시오.

is_crashed

유형: `Boolean`

범위: `(True:False)`

에이전트가 다른 객체와 충돌했는지(True) 또는 충돌하지 않았는지(False)를 종료 상태로 나타내는 Boolean 플래그입니다.

is_left_of_center

유형: Boolean

범위: [True : False]

에이전트가 트랙 중앙선의 왼쪽에 있는지(True) 트랙 중앙선의 왼쪽에 있지 않은지(False)를 나타내는 Boolean 플래그입니다.

is_offtrack

유형: Boolean

범위: (True:False)

에이전트의 바퀴 4개가 모두 트랙의 안쪽 또는 바깥쪽 경계선 밖으로 주행했는지(True) 또는 그렇지 않은지(False)를 나타내는 Boolean 플래그입니다.

is_reversed

유형: Boolean

범위: [True:False]

에이전트가 시계 방향으로 주행(True)하는지 시계 반대 방향으로 주행(False)하는지를 나타내는 Boolean 플래그입니다.

각 에피소드에 대한 방향 변경을 활성화할 때 사용됩니다.

objects_distance

유형: [float, …]

범위: [(0:track_length), …]

시작선을 기준으로 환경 내 객체 간 거리 목록입니다. i번째 요소는 트랙 중앙선을 따라 i번째 객체와 시작선 사이의 거리(미터)를 측정합니다.

> **Note**
>
> abs | (var1) - (var2)| = how close the car is to an object, WHEN var1 = ["objects_distance"][index] and var2 = params["progress"]*params["track_length"]
> 차량 앞에서 가장 가까운 객체와 차량 뒤에서 가장 가까운 객체의 인덱스를 가져오려면 `closest_objects` 파라미터를 사용합니다.

objects_heading

유형: `[float, … ]`

범위: `[(-180:180), … ]`

객체의 방향(도)의 목록입니다. i번째 요소는 i번째 객체의 방향을 측정합니다. 고정된 객체의 방향은 0입니다. 봇 차량의 경우 해당 요소의 값은 그 봇 차량의 방향 각도입니다.

objects_left_of_center

유형: `[Boolean, … ]`

범위: `[True|False, … ]`

`Boolean` 플래그 목록. i번째 요소 값은 그 i번째 객체가 트랙 중앙선의 왼쪽에 있는지(`True`) 오른쪽에 있는지(`False`)를 나타냅니다.

objects_location

유형: `[(x,y), ...]`

범위: `[(0:N,0:N), ...]`

이 파라미터는 모든 객체 위치를 저장합니다. 각 위치는 (x, y)의 튜플입니다.

목록 크기는 트랙 위 객체의 수와 같습니다. 나열된 객체에는 고정된 장애물과 움직이는 봇 차량이 모두 포함됩니다.

objects_speed

유형: `[float, … ]`

범위: `[(0:12.0), … ]`

트랙 위 객체의 속도(m/s) 목록입니다. 정지 객체의 경우 속도는 0입니다. 로봇 차량의 경우 이 값은 훈련 시 설정한 속도입니다.

progress

유형: `float`

범위: `0:100`

주행한 트랙의 비율입니다.

예: *`progress`* 파라미터를 사용하는 보상 함수

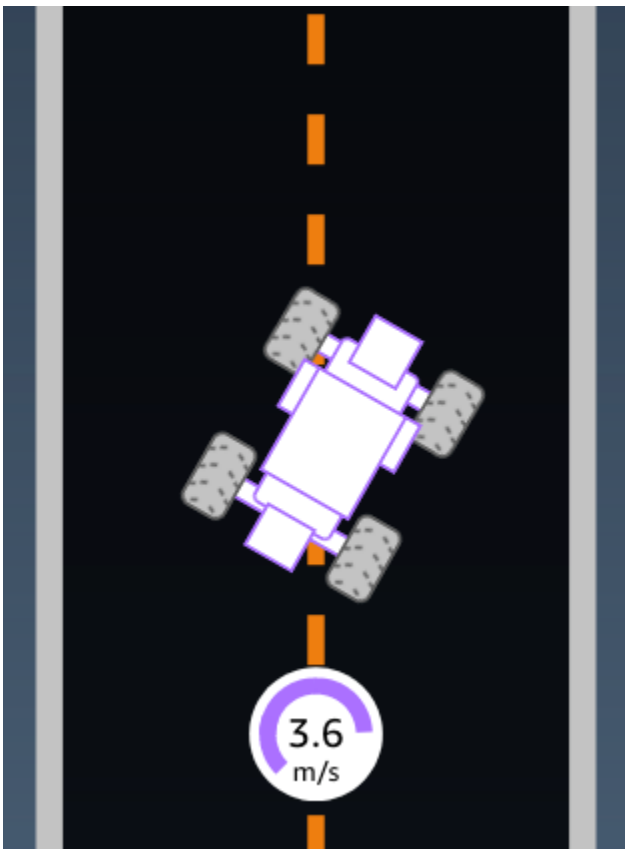
자세한 내용은 단계 항목을 참조하십시오.

속도

유형: `float`

범위: `0.0:5.0`

관측된 에이전트 속도(m/s)입니다.



예: *speed* 파라미터를 사용하는 보상 함수

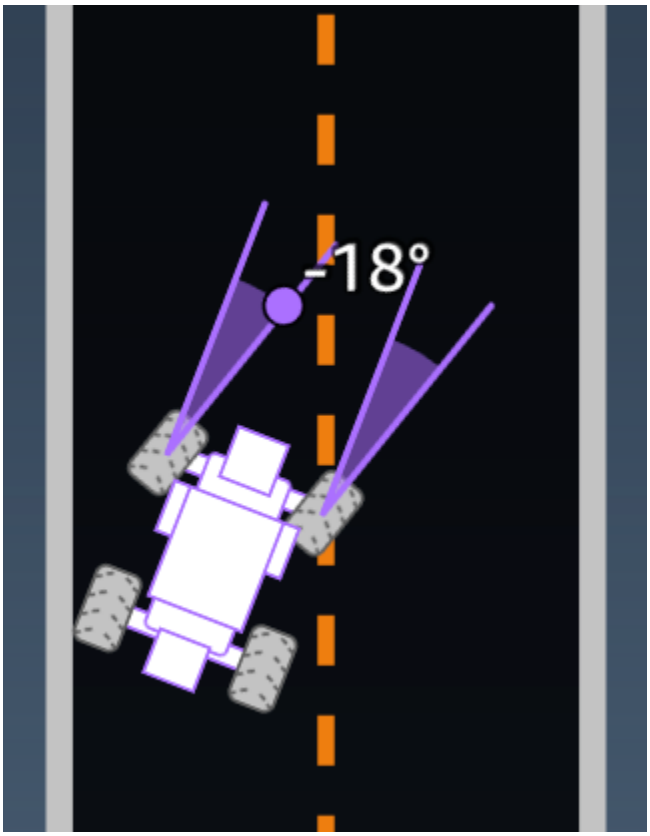
자세한 내용은 all_wheels_on_track 항목을 참조하십시오.

steering_angle

유형: `float`

범위: `-30:30`

에이전트 중앙선에 대한 전륜 조향 각도입니다. 음의 기호(-)는 오른쪽 조향을, 그리고 양의 기호(+)는 왼쪽 조향을 의미합니다. 다음 그림에 표시된 것처럼 에이전트의 중앙선이 트랙 중앙선과 반드시 평행을 이루지는 않습니다.



예: *steering_angle* 파라미터를 사용하는 보상 함수

```
def reward_function(params):
    '''
    Example of using steering angle
    '''
```

```
    # Read input variable
    abs_steering = abs(params['steering_angle']) # We don't care whether it is left or
 right steering

    # Initialize the reward with typical value
    reward = 1.0

    # Penalize if car steer too much to prevent zigzag
    ABS_STEERING_THRESHOLD = 20.0
    if abs_steering > ABS_STEERING_THRESHOLD:
        reward *= 0.8

    return float(reward)
```

steps

유형: `int`

범위: `0:`N_{step}

완료한 단계 수입니다. 한 단계는 에이전트가 현재 정책을 사용하여 완료한 하나의 관찰-조치 시퀀스에 해당합니다.

예: *`steps`* 파라미터를 사용하는 보상 함수

```
def reward_function(params):
    ###############################################################################
    '''
    Example of using steps and progress
    '''

    # Read input variable
    steps = params['steps']
    progress = params['progress']

    # Total num of steps we want the car to finish the lap, it will vary depends on the
 track length
    TOTAL_NUM_STEPS = 300

    # Initialize the reward with typical value
    reward = 1.0

    # Give additional reward if the car pass every 100 steps faster than expected
```

```
    if (steps % 100) == 0 and progress > (steps / TOTAL_NUM_STEPS) * 100 :
        reward += 10.0

    return float(reward)
```

track_length

유형: `float`

범위: [0:L_{max}]

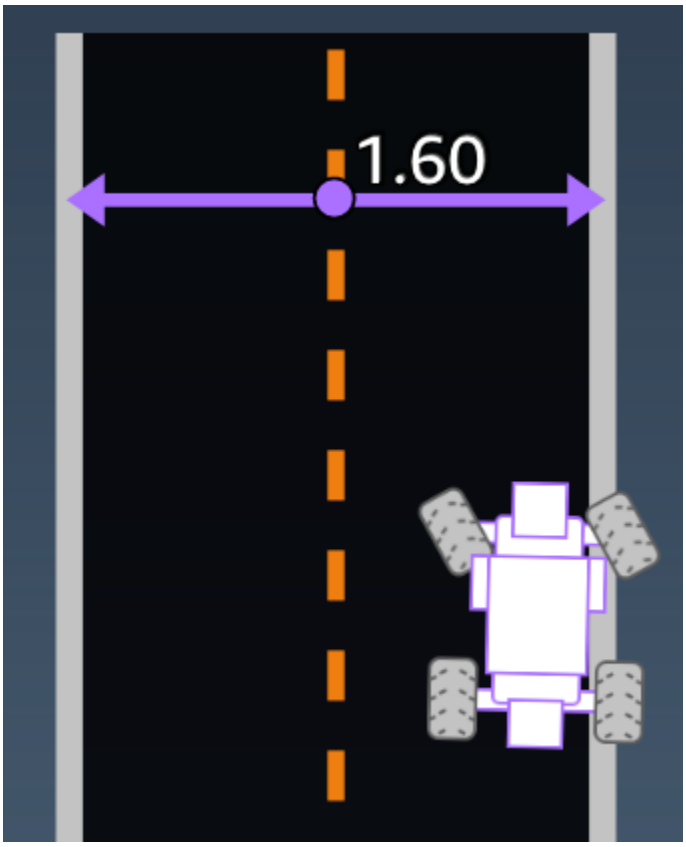
트랙 길이(미터)입니다. L_{max} is track-dependent.

track_width

유형: `float`

범위: 0:D_{track}

트랙 너비(미터)입니다.



예: *track_width* 파라미터를 사용하는 보상 함수

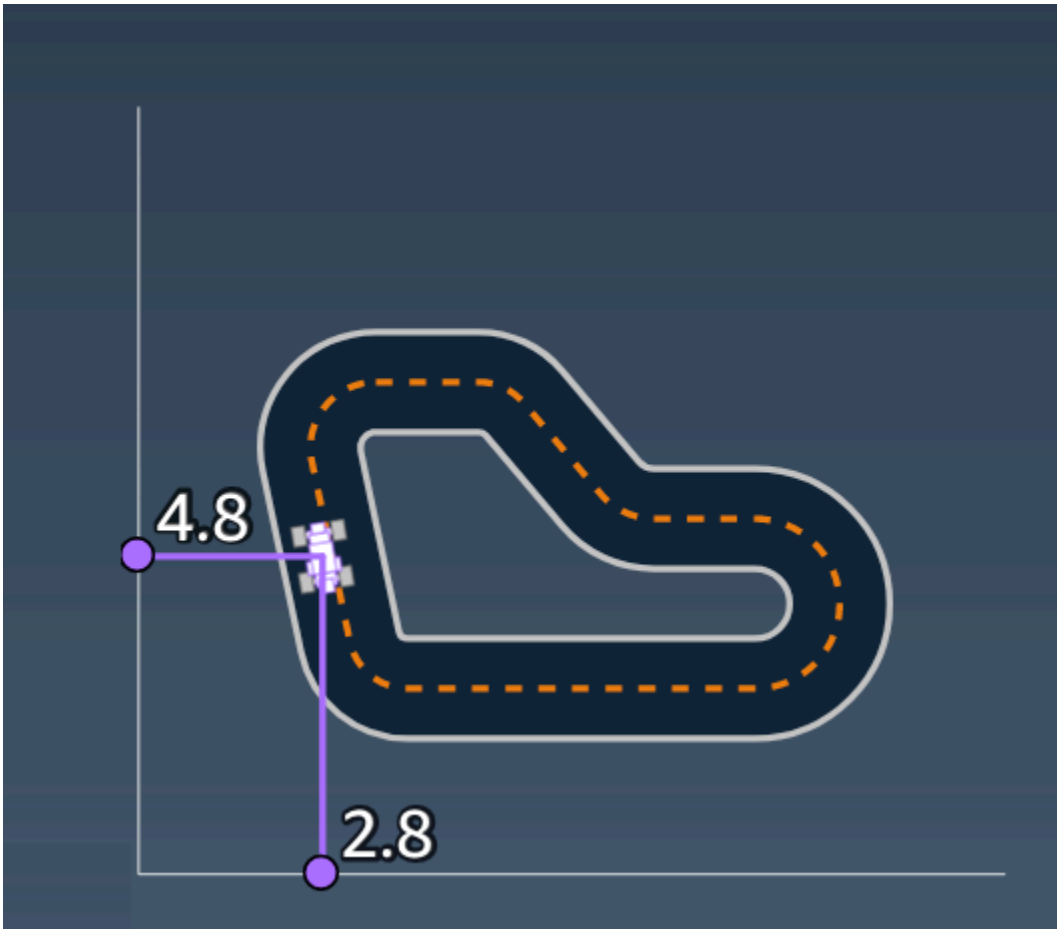
```
def reward_function(params):
    ################################################################################
    '''
    Example of using track width
    '''

    # Read input variable
    track_width = params['track_width']
    distance_from_center = params['distance_from_center']

    # Calculate the distance from each border
    distance_from_border = 0.5 * track_width - distance_from_center

    # Reward higher if the car stays inside the track borders
    if distance_from_border >= 0.05:
        reward = 1.0
    else:
        reward = 1e-3 # Low reward if too close to the border or goes off the track

    return float(reward)
```

x, y

유형: `float`

범위: `0:N`

트랙이 포함된 시뮬레이션 환경에서 x축과 y축에 따른 에이전트 중앙의 위치(미터)입니다. 원점은 시뮬레이션 환경에서 왼쪽 하단 모퉁이입니다.

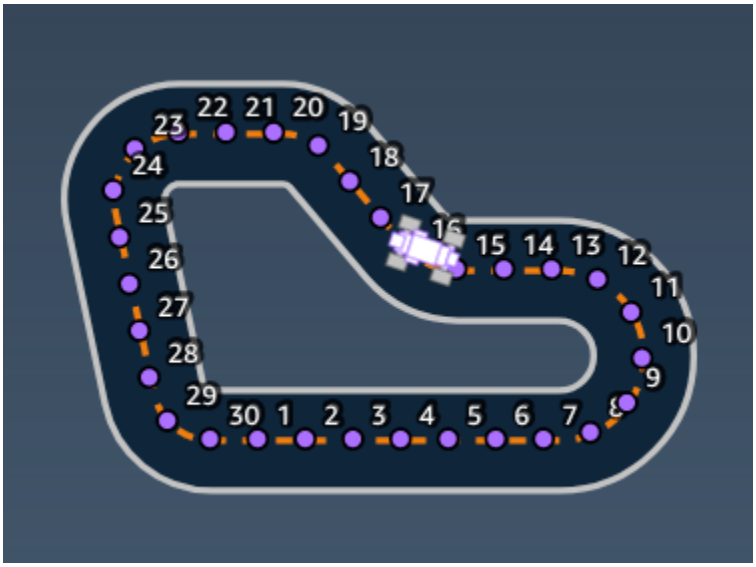


중간 지점

유형: `[float, float] list`

범위: `[[`$x_{w,0}$`,`$y_{w,0}$`] … [`$x_{w,Max-1}$`, `$y_{w,Max-1}$`]]`

트랙 중앙을 따라 순서대로 나열된 트랙에 의존하는 `Max` 이정표 목록입니다. 각 이정표는 ($x_{w,i}$, $y_{w,i}$) 좌표로 알 수 있습니다. 순환 트랙의 경우, 첫 번째와 마지막 중간 지점은 동일합니다. 직선 또는 다른 비순환 트랙의 경우, 첫 번째와 마지막 중간 지점은 다릅니다.



예: *waypoints* 파라미터를 사용하는 보상 함수

자세한 내용은 closest_waypoints 단원을 참조하십시오.

Security in AWS DeepRacer Student

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. AWS DeepRacer Student에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램 제공 범위 내 AWS 서비스 규정 준수 프로그램 참조하세요.
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 여러분은 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다

이 설명서는 AWS DeepRacer Student를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 보안 및 규정 준수 목표를 충족하도록 AWS DeepRacer Student를 구성하는 방법을 보여줍니다. 또한 AWS DeepRacer Student 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법도 알아봅니다.

내용

- AWS DeepRacer Student의 데이터 보호
- AWS DeepRacer Student의 자격 증명 및 액세스 관리
- AWS DeepRacer Student에 대한 규정 준수 검증
- AWS DeepRacer Student의 복원력
- AWS DeepRacer Student의 인프라 보안

AWS DeepRacer Student의 데이터 보호

다음 섹션에서는 AWS DeepRacer Student가 캡처하는 데이터와 AWS DeepRacer Student가 데이터 암호화를 사용하여 데이터를 보호하는 위치에 대해 설명합니다.

AWS DeepRacer Student 계정을 생성할 때 AWS 플레이어 계정도 생성합니다. AWS DeepRacer Student 계정에서 생성된 리소스는 AWS 플레이어 계정에 저장됩니다. AWS 플레이어 계정에 대한 자

세한 내용은 AWS DeepRacer Student 사용 설명서AWS 플레이어 계정이란 무엇입니까?의 섹션을 참조하세요.

주제

- AWS DeepRacer Student 포털에서 캡처된 데이터
- AWS DeepRacer Student 포털의 저장 시 암호화
- In AWS DeepRacer Student 포털에서 전송 중 암호화

AWS DeepRacer Student 포털에서 캡처된 데이터

AWS DeepRacer Student 포털을 사용하려면 필요한 데이터가 AWS 플레이어 계정에 저장됩니다. AWS DeepRacer Student 포털에서 캡처된 데이터는 서비스 개선에 사용되지 않습니다.

AWS DeepRacer Student에서 캡처된 데이터.

다음은 AWS DeepRacer Student에서 생성되어 AWS 플레이어 계정에 저장된 데이터의 요약입니다.

- 계정을 등록할 때 사용한 이메일 주소와 암호.
- 내 레이서 이름
- Student League 리더보드의 내 순위
- 훈련한 내 모델
- 보상 함수 코드

AWS DeepRacer Student 포털의 저장 시 암호화

AWS DeepRacer Student 포털에서 캡처한 데이터는 기본적으로 암호화됩니다.

AWS 플레이어 계정은 Amazon Cognito를 사용하여 AWS DeepRacer Student에 로그인하는 데 사용되는 이메일과 암호를 암호화하고 저장합니다. 자세한 내용은 Amazon Cognito의 데이터 보호를 참조하세요.

AWS DeepRacer Student에서 캡처한 다른 모든 데이터는 AES-GCM AWS Key Management Service을 사용하여를 통해 AWS 소유 키를 사용하고 256비트 크기의 키를 사용하여 클라우드에서 저장 시 암호화됩니다. 이 데이터는 Amazon Simple Storage Service(S3) 및 Amazon DynamoDB에 저장되고 암호화됩니다.

In AWS DeepRacer Student 포털에서 전송 중 암호화

귀하의 등록 및 승인된 이메일 주소는 클라이언트측 암호화로 암호화됩니다. AWS DeepRacer Student에서 캡처한 다른 모든 데이터는 계정에서 복사되어 내부 AWS 시스템에서 처리됩니다. 기본적으로 AWS DeepRacer Student는 HTTPS를 통한 보안 연결을 사용하여 전송 중인 데이터를 암호화합니다.

AWS DeepRacer Student의 자격 증명 및 액세스 관리

AWS Identity and Access Management(IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있도록 지원하는 AWS 서비스입니다. IAM 관리자는 AWS 리소스를 사용할 수 있는 인증(로그인) 및 권한 부여(권한 있음)를 받을 수 있는 사용자를 제어합니다. AWS DeepRacer Student는 AWS 리소스에 대한 사용자 액세스를 제어하기 위해 IAM과 직접 통합되지 않습니다. 대신 AWS DeepRacer Student는 AWS DeepRacer에서 관리하는 인증된 프록시 API를 사용하여 사용자 리소스를 보호합니다.

AWS DeepRacer Student에 대한 규정 준수 검증

타사 감사자는 여러 규정 준수 프로그램의 일환으로 AWS DeepRacer Student의 보안 및 AWS 규정 준수를 평가합니다.

특정 규정 준수 프로그램 범위의 AWS 서비스 목록은 규정 준수 프로그램 제공 범위 내 AWS 서비스규정 준수 프로그램 제공 . 일반적인 정보는 AWS 규정 준수 프로그램을 참조하세요.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 Downloading Reports inDownloading AWS Artifact 참조하세요.

AWS 는 규정 준수에 도움이 되는 다음 리소스를 제공합니다.

- 보안 및 규정 준수 빠른 시작 안내서: 이 배포 안내서에서는 아키텍처 고려 사항에 관해 설명하고 AWS에서 보안 및 규정 준수에 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다.
- AWS 규정 준수 리소스 -이 워크북 및 가이드 모음은 업계 및 위치에 적용될 수 있습니다.
- AWS Config 개발자 안내서의 규칙을 사용하여 리소스 평가 - 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 AWS Config평가합니다.
- AWS Security Hub -이 AWS 서비스는 보안 업계 표준 및 모범 사례 준수를 확인하는 데 도움이 AWS 되는 내 보안 상태에 대한 포괄적인 보기를 제공합니다.

AWS DeepRacer Student의 복원력

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 기반으로 구축됩니다. 리전은 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며, 이러한 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크를 통해 연결되어 있습니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하세요.

AWS DeepRacer Student의 인프라 보안

관리형 서비스인 AWS DeepRacer Student는 Amazon Web Services: 보안 프로세스 개요 백서에 설명된 AWS 글로벌 네트워크 보안 절차로 보호됩니다. https://d0.awsstatic.com/whitepapers/Security/AWS_Security_Whitepaper.pdf

common AWS DeepRacer Student 문제 해결

주제

- 내 AWS DeepRacer Student 계정에서 자동으로 로그아웃된 이유는 무엇인가요?
- AWS AI 및 ML 장학금 프로그램에서 탈퇴하려면 어떻게 해야 하나요?
- my AWS DeepRacer Student 계정을 삭제할 수 없습니다.
- 드롭다운 목록에서 학교 이름을 찾을 수 없어요
- 모델 훈련을 계속할 수 없어요
- '이 이메일에 등록된 계정이 있습니다'라는 오류 메시지가 뜹니다.
- Gmail 계정으로 가입했는데 인증 코드를 못 찾겠어요

내 AWS DeepRacer Student 계정에서 자동으로 로그아웃된 이유는 무엇인가요?

AWS 보안 정책에 따라 30일 후에 AWS DeepRacer Student 계정에서 자동으로 로그아웃됩니다.

- 서비스를 계속 사용하려면 AWS DeepRacer Student 로그인 페이지로 이동한 다음 보안 인증 정보를 사용하여 다시 로그인하세요.

AWS AI 및 ML 장학금 프로그램에서 탈퇴하려면 어떻게 해야 하나요?

AWS AI 및 ML 장학금 프로그램은 선택 사항이며, 교육의 혜택을 덜 누리고 소외된 16세 이상의 학생을 대상으로 합니다. AWS DeepRacer Student에 가입하면 기본적으로 AWS AI 및 ML 장학금 프로그램에 등록되지 않습니다.

참가하려면 AWS DeepRacer Student 가입과 동시에 AWS AI 및 ML 장학금 프로그램에 지원하고 싶으신가요? 섹션의 확인란에 체크 표시하여 동의하거나 나중에 사이트 왼쪽 탐색 창을 통해 내 프로필 페이지로 가서 동의할 수 있습니다.

- 프로그램에 동의하면 지원 절차에 대한 접속 권한만 부여됩니다. 지원하지 않기를 선택할 수 있습니다.

my AWS DeepRacer Student 계정을 삭제할 수 없습니다.

AWS DeepRacer Student 계정을 삭제할 수 없는 경우 AWS DeepRacer 다중 사용자 이벤트를 생성했는지 확인합니다. AWS 플레이어 계정은 AWS for AWS DeepRacer 다중 사용자 및 AWS DeepRacer Student에서 생성한 관리형 자격 증명 솔루션입니다. AWS 플레이어 계정은 이러한 각 AWS 서비스에서 생성된 모든 리소스를 보유합니다.

- 생성한 이벤트의 참가자가 중단된 경험을 하지 않도록 하려면 AWS DeepRacer 다중 사용자 이벤트에 대한 리소스가 포함된 AWS DeepRacer Student 계정을 삭제할 수 없습니다.

드롭다운 목록에서 학교 이름을 찾을 수 없어요

드롭다운 목록에서 모든 학교, 특히 고등학교를 찾을 수 없을 수도 있습니다.

- 학교가 드롭다운 목록에 없는 경우 기타를 선택하고 학교 이름을 입력하세요.

모델 훈련을 계속할 수 없어요

월간 모델 훈련 시간 제한을 초과했을 수 있습니다.

- 홈페이지로 이동하여 남은 모델 훈련 시간 섹션에서 사용한 훈련 시간을 확인하세요. 모델 훈련 시간을 초과했으면 시간이 재설정될 때까지 기다렸다가 교육을 다시 시작하세요.

‘이 이메일에 등록된 계정이 있습니다’라는 오류 메시지가 뜹니다.

AWS 플레이어 계정 가입 페이지에 확인 코드를 입력하고 이미 동일한 이메일 주소를 사용하여 AWS DeepRacer 다중 사용자를 통해 AWS 플레이어 계정에 가입하면이 오류 메시지가 표시됩니다. 이전에 동일한 이메일 주소를 사용하여 AWS DeepRacer Student 계정에 가입한 경우에도이 오류가 발생합니다.

- 이전에 생성한 보안 인증 정보를 사용하여 AWS DeepRacer Student 로그인 페이지에 로그인하거나 암호 입력란 아래에 있는 암호 찾기를 선택하여 암호 재설정을 요청하세요.

Gmail 계정으로 가입했는데 인증 코드를 못 찾겠어요

Gmail 계정을 사용하여 AWS 플레이어 계정에 가입했는데 확인 코드 메시지를 찾을 수 없는 경우 잘못된 폴더로 전달되었을 수 있습니다.

- Gmail 계정에 로그인하고 프로모션 폴더에서 ‘AWS Player 프로필 인증 코드’라는 제목의 메시지가 있는지 확인해 보세요.

AWS DeepRacer Student 할당량

AWS DeepRacer Student에 참여하는 모든 학생은 10시간의 무료 월간 모델 훈련 컴퓨팅 리소스와 5GB의 스토리지를 받습니다.

AWS DeepRacer Student 계정 삭제

AWS DeepRacer Student 포털은 AWS 플레이어 계정에 다음 정보를 저장합니다.

- 이메일 주소
- 암호
- 내 레이서 이름
- 내 Student League 리더보드 순위

수집 데이터에 관한 자세한 내용은 AWS DeepRacer Student 사용 설명서AWS DeepRacer Student의 데이터 보호 항목을 참조하세요.

AWS의 서버에서이 정보를 제거하려면 다음 절차에 따라 AWS DeepRacer Student 포털 계정을 삭제합니다. AWS DeepRacer Student 계정을 삭제하면 AWS 플레이어 계정과 연결된 모든 리소스도 삭제됩니다.

AWS 플레이어 계정에 대한 자세한 내용은 섹션을 참조하세요. AWS 플레이어 계정이란 무엇입니까?

> **Note**
>
> AWS DeepRacer 다중 사용자 이벤트를 생성한 경우 AWS 플레이어 계정을 삭제할 수 없습니다. 자세한 내용은 AWS DeepRacer Student 사용 설명서의 my AWS DeepRacer Student 계정을 삭제할 수 없습니다.항목을 참조하세요.

AWS DeepRacer Student 플레이어 계정을 삭제하려면

> **Important**
>
> AWS DeepRacer Student 계정 삭제는 취소할 수 없는 작업입니다. AWS DeepRacer Student를 삭제하면 AWS 플레이어 계정과 모든 관련 리소스도 삭제됩니다.
> AWS DeepRacer Student 계정을 삭제하면 AWS 플레이어 계정의 리소스가 1년 이내에 서버에서 제거됩니다.

1. AWS DeepRacer Student 랜딩 페이지: https://student.deepracer.com/signIn 엽니다.
2. 메시지가 표시되면 AWS DeepRacer Student 계정에 로그인합니다.

3. 내 계정을 선택합니다.
4. 내 계정페이지에서 내 계정 삭제를 선택합니다.
5. 삭제를 확정하려면 입력란에 '삭제'라고 입력하십시오 아래에 **`Delete`**를 입력합니다.
6. Delete(삭제)를 선택합니다.

계정이 성공적으로 삭제되면 성공적으로 삭제된 계정 메시지가 나타나고 AWS DeepRacer Student 로그인 페이지로 돌아갑니다.

계정을 삭제하려면 AWS 계정 해지에 설명된 단계를 사용합니다 AWS.

당사는 고객이 개인 정보 보호 및 데이터 보안을 중요하게 여긴다는 사실을 잘 알고 있으며, 고객 콘텐츠에 대한 무단 접근이나 공개를 방지하고자 책임감 있고 정교한 기술 및 물리적 제어를 구현합니다. 고객의 신뢰를 유지하도록 지속적으로 노력하고 있습니다. 개인정보 보호 고지 페이지에서 AWS 데이터 개인정보 보호 약정에 대해 자세히 알아볼 수 있습니다.

AWS DeepRacer Student 사용 설명서의 문서 기록

다음 표에서는 AWS DeepRacer Student의 설명서 릴리스를 설명합니다.

변경 사항	설명	날짜
2024년 프로그램 업데이트	2024 AWS AI 및 ML 장학금 프로그램 및 AWS DeepRacer Student League를 참조하는 주제에 대한 업데이트입니다. AWS AI 및 ML 장학금을 신청하는 모든 학생이 이용할 수 있는 새 교육 과정에 관한 정보가 추가되었습니다.	2023년 11월 1일
2023 AWS DeepRacer Student League 업데이트	2023 AWS DeepRacer Student League를 참조하는 여러 주제에 대한 업데이트. 자세한 내용은 이용 약관을 참조하세요.	2023년 3월 1일
최초 릴리스	AWS AI 및 ML 장학금 프로그램 및 AWS DeepRacer Student League에 대한 지원을 포함한 AWS DeepRacer Student 사용 설명서의 최초 릴리스입니다.	2021년 12월 1일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.