



用户指南

AWS DeepRacer 学生



AWS DeepRacer 学生: 用户指南

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

什么是 AWS DeepRacer Student ?	1
您对 AWS AI & ML 奖学金计划感兴趣吗?	1
您是首次使用 AWS DeepRacer Student 的用户吗?	1
您有兴趣参加 AWS DeepRacer Student League 吗?	2
什么是 AWS AI & ML 奖学金计划?	3
AWS AI & ML 奖学金计划中的奖项	3
如何报名参加 AWS AI & ML 奖学金计划并获得资格预审?	4
有关 AWS AI & ML 奖学金计划和 Udacity 平台的常见问题 (FAQ)	4
什么是 AWS 玩家账号?	7
为支持的服务创建 AWS 玩家账户	7
删除 AWS 玩家账号	7
开始使用	8
先决条件	8
第 1 步: 注册 AWS DeepRacer Student	8
第 2 步: 完成 AWS DeepRacer Student 的注册	9
第 3 步: (可选) 阅读并选择参加 AWS AI & ML 奖学金申请	10
第 4 步: 更新个人资料	10
第 5 步: 从主页浏览 AWS DeepRacer Student	11
培训强化学习模型	13
第 1 步: 使用 AWS DeepRacer Student 训练强化学习模型	13
第 2 步: 命名模型	13
第 3 步: 选择赛道	13
第 4 步: 选择算法	14
第 5 步: 自定义您的奖励函数	14
第 6 步: 选择时长并将您的模型提交到排行榜	15
第 7 步: 在排行榜上查看模型的表现	16
第 8 步: 使用克隆改进模型	16
第 9 步: (可选) 下载模型	17
加入竞赛	18
参加学生联赛竞赛	18
参加学生社区竞赛	18
自定义奖励函数	20
编辑 Python 代码以自定义您的奖励函数	20
奖励函数输入参数	20

all_wheels_on_track	22
closest_waypoints	24
closest_objects	25
distance_from_center	25
heading	27
is_crashed	27
is_left_of_center	28
is_offtrack	28
is_reversed	28
objects_distance	28
objects_heading	29
objects_left_of_center	29
objects_location	29
objects_speed	29
progress	29
speed	30
steering_angle	30
steps	32
track_length	32
track_width	33
x, y	34
waypoints	35
安全性	36
数据保护	36
在 AWS DeepRacer Student 门户中捕获的数据	37
AWS DeepRacer Student 门户中的静态加密	37
AWS DeepRacer Student门户中的传输加密	37
身份和访问管理	38
合规性验证	38
故障恢复能力	38
基础设施安全性	39
常见问题疑难解答。	40
为什么我的 AWS DeepRacer 学生账户被自动注销？	40
如何选择退出 AWS AI & ML 奖学金计划？	40
我无法删除我的 AWS DeepRacer 学生账户	40
我在下拉列表中找到我的学校名称	41

我无法继续训练我的模型	41
我收到“该电子邮件已注册账户”的错误消息	41
我使用了 Gmail 账户进行注册，但找不到我的验证码	41
配额	42
账户删除	43
文档历史记录	45
.....	xlvi

什么是 AWS DeepRacer Student ?

AWS DeepRacer Student 是全球高中和大学入学的学习者培养机器学习 (ML) 技能的地方。它提供教材、可选的 AWS AI & ML 奖学金计划，还为 AWS DeepRacer Student League 提供训练和测试强化学习 (RL) 模型的机会。请参阅 [???](#) 部分的主题了解如何开始。

AWS DeepRacer Student 的功能

- 主页 — 查找有关即将举行的活动、练习训练 RL 模型的详细信息、访问 ML 教育内容并跟踪您的模型训练时间。您还可以从左侧导航窗格中管理您的 AWS DeepRacer Student 资料和账户信息。
- 学习 — 访问由 AWS 专家开发的 ML 内容，包括视频。没有经验的学生可以从易于理解、自定进度的材料中学习 ML 的基础知识。
- 练习 — 选择跟踪、算法和奖励函数来创建 RL 模型。或者，在指导下走查奖励函数 Python 代码，然后选择对其进行自定义。使用 AWS DeepRacer 服务在模拟的 3D 赛车环境中训练您的模型。克隆您表现最好的模型并迭代其奖励函数，攀升 AWS DeepRacer Student League 排行榜。
- 竞争 — 将您的模型提交给每月的虚拟竞赛，以获得成就并有机会赢取奖品。各地区顶尖的学生赛车手将有机会参加 Student World Championships。要了解更多信息，请参阅[条款和条件](#)。所有学生每月有 10 个小时为 AWS DeepRacer Student League 训练 RL 模型。只有使用 AWS DeepRacer Student 训练过的模型才能在 AWS DeepRacer Student League 中使用。

主题

- [您对 AWS AI & ML 奖学金计划感兴趣吗？](#)
- [您是首次使用 AWS DeepRacer Student 的用户吗？](#)
- [您有兴趣参加 AWS DeepRacer Student League 吗？](#)

您对 AWS AI & ML 奖学金计划感兴趣吗？

如果您自认自己在技术领域服务欠缺或代表性不足，请选择加入 AWS AI & ML 奖学金计划。要详细了解资格条件、申请方式以及奖项，请参阅 [什么是 AWS AI & ML 奖学金计划？](#)

您是首次使用 AWS DeepRacer Student 的用户吗？

如果您是首次接触 AWS DeepRacer Student 的用户，我们建议您先阅读以下部分：

- [AWS DeepRacer Student 入门](#)

- [参加 AWS DeepRacer Student 竞赛](#)
- [在 AWS DeepRacer Student 中训练强化学习模型](#)

您有兴趣参加 AWS DeepRacer Student League 吗？

AWS DeepRacer Student League 是一项由 AWS 赞助的比赛，学生可以在 3 月至 9 月期间参加每月一次的虚拟比赛，在国家和地区级别的比赛中竞争，并获取资格参加 10 月的 Student World Championships。学生有机会获得奖励和奖品。有关更多信息，请参阅[条款和条件](#)。

要参加第一场比赛，请参阅 [参加 AWS DeepRacer Student 竞赛](#)。

什么是 AWS AI & ML 奖学金计划？

AWS AI & ML 奖学金计划作为 AWS DeepRacer Student 的一部分启动，旨在通过为成功的申请者提供获得最多两个 Udacity Nanodegree 的机会，增加人工智能 (AI) 和机器学习 (ML) 领域的多样性。Udacity Nanodegree 是一门大规模的开放式在线课程 (MOOC)，旨在缩小学习和职业目标之间的差距。如需了解更多详情，请参阅 Udacity 支持文档中的[什么是 Nanodegree 计划？](#)。申请成功的人还将有机会参加独家活动并获得指导，以帮助他们进一步发展自己的职业生涯。有关更多信息，请参阅[AWS AI & ML 奖学金计划中的奖项](#)。

该奖学金面向在科技领域服务欠缺和代表性不足的人。申请人必须年满 16 岁，并且目前就读于高中、大学或社区学院。有关如何进行资格预审的更多信息，请参阅[如何报名参加 AWS AI & ML 奖学金计划并获得资格预审？](#)

参加 AWS AI & ML 奖学金计划是免费的。

AWS AI & ML 奖学金计划于 2022 年 4 月 11 日正式启动。

AWS AI & ML 奖学金计划采用组队的方式进行。每年选出两组 1,000 名学生（共有 2,000 名学生）。第一组学生的申请需于 2024 年 5 月 31 日前提交给 Udacity，第二组学生的申请需于 2024 年 9 月 30 日前提交。

AWS AI & ML 奖学金计划中的奖项

关于 AWS AI & ML 奖学金计划，AWS 正在与 Udacity 合作。

每年，AWS AI & ML 奖学金计划将在全球范围内提供 2,500 项 Udacity Nanodegree 奖学金，分配给两个不同的群组。所有学生都将获得以下奖励：

- 免费参加使用 Python 进行人工智能编程的 Nanodegree 课程。本课程教授人工智能技术入门所需的基础技能，并培养编程、线性代数和神经网络方面的技能。
- 在一周的办公时间内，与 Udacity 讲师一起帮助回答有关课程内容的问题。此外，学生还可以参加由 Udacity 讲师主持的每周案例研究练习。
- 获得独一的指导和职业资源，包括行业专家的会议、简历编辑指导和面试技巧。

此外，《使用 Python 进行人工智能编程》课程（基于课程评估的表现）中排名前 500 的学生将获得更高级的 Nanodegree，该学位涵盖了深度学习的基础知识，并为您提供成为机器学习工程师所需的技能。

所有完成奖学金先决条件并提交 Udacity 申请的学生都可以免费参加由 AWS 和 Udacity 共同创建的生成式人工智能新课程和 AWS 一起介绍生成式人工智能。这门为期一个月的课程是一项可选资源，旨在帮助学生进一步提高生成式人工智能领域的技能。该课程适用于所有提交奖学金申请的学生，而 Nanodegree 奖学金仅适用于奖学金获得者。

如何报名参加 AWS AI & ML 奖学金计划并获得资格预审？

AWS AI & ML 奖学金计划面向 16 岁或以上服务欠缺和代表性不足的学生。代表性不足和服务欠缺的学生包括（但不限于）女性、残疾人、有色人种（黑人、拉丁裔和土著人）以及 LGBTQ+ 群体的成员。

首先，学生需要在 AWS DeepRacer Student 上注册并创建一个账户。注册时，勾选复选框即可选择加入 AWS AI & ML 奖学金计划。设置账户后，您会在主页上看到一个用于跟踪您的奖学金申请资格预审任务的磁贴。学生需要在 AWS DeepRacer Student 中完成两个先决条件才能提交奖学金计划的申请。

通过 AWS AI & ML 奖学金资格预审的学生将获得一个唯一的访问码，他们可以在 Udacity 网站上兑换，以获取 Udacity Nanodegree 奖学金申请表。通过资格预审的学生不一定能获得 Udacity Nanodegree 奖学金。由 Udacity 决定哪些通过资格预审的学生将获得 Udacity Nanodegree 奖学金。

奖学金资格预审基于两个标准：

复习课程并通过评估

要进行资格预审，您必须在所有必需的评估中取得 80% 或更高的分数。每项评估都基于不同的章节，您可以在 AWS DeepRacer Student 导航窗格的学习部分中找到。奖学金申请将于 2024 年 2 月 1 日开始。

达到最短的单圈时间

从 2 月到 9 月，每个月，AWS DeepRacer League 都会推出新的排行榜，其中包含一条新的轨迹。要通过奖学金资格预审，您需要在任何一场比赛中达到低于两分钟的单圈时间。

学生完成两个先决条件后，他们会收到来自 AWS 的一封电子邮件，其中包含唯一的访问码和 Udacity Nanodegree 奖学金申请链接。要获得奖学金资格，学生需要完整填写申请表。申请完成后，Udacity 会联系学生，提供报名参加和 AWS 一起介绍生成式人工智能选修课的步骤，该课程将向所有申请人提供。Udacity 联系被选中获得 Udacity Nanodegree 奖学金的申请人。

有关 AWS AI & ML 奖学金计划和 Udacity 平台的常见问题 (FAQ)

AWS AI & ML 奖学金的官方条款和条件是什么？

要查看官方条款和条件，请参阅 [AWS AI & ML 奖学金计划官方条款和条件](#)。

这个奖学金是给谁的？

这项 AWS AI & ML 奖学金计划面向全球 16 岁或以上服务欠缺和代表性不足的学生。代表性不足和服务欠缺的学生包括（但不限于）女性、残疾人、有色人种（黑人、拉丁裔和土著人）以及 LGBTQ+ 群体的成员。

什么是 Nanodegree？

Nanodegree 是一项基于技能的在线教育计划，旨在帮助缩小学习和职业技能之间的差距。

如果我通过 AWS AI & ML 奖学金资格预审，我是否会自动获得 Udacity Nanodegree 奖学金？

不是。预先获得 AWS AI & ML 奖学金资格的学生可以获得 Udacity Nanodegree 奖学金的申请机会。由 Udacity 决定哪些通过资格预审的学生将获得 Udacity Nanodegree 奖学金。

在被选中获得 AWS AI & ML 奖学金之前，我是否需要提供入学证明？

您可能需要提供入学证明（例如大学成绩单）才能获得 AWS AI & ML 奖学金。

在申请 AWS AI & ML 奖学金计划时，如何表明我目前正在读高中？

要表明您已就读高中，请参阅 AWS DeepRacer Student 用户指南中的 [第 4 步：更新个人资料](#)。

如何保护我的个人身份信息 (PII) 数据？

您的数据在 AWS 云端受到保护。有关更多信息，请参阅 [AWS 隐私通知](#)。

申请 AWS AI & ML 奖学金有年龄要求吗？

是。要申请 AWS AI & ML 奖学金计划，报名时必须年满 16 岁。

如果我获得 AWS AI & ML 奖学金，我将如何收到通知？

申请截止日期过后，您将收到一封来自 Udacity 的电子邮件，其中涉及您的 Nanodegree 奖学金申请状态。

如果我获得 AWS AI & ML 奖学金，我何时会收到通知？

Udacity 将联系申请人并告知奖学金状态。被选中获得奖学金的学生将获得有关重要日期的信息。

如何访问 Udacity 上的 AWS 生成式人工智能课程？

学生需要通过 AWS DeepRacer Student 的资格预审，并在 Udacity 上提交奖学金申请。Udacity 会在学生提交完整的申请后与他们联系，并向他们提供报名参加和 AWS 一起介绍生成式人工智能课程的步骤。

如果我入选 AWS AI & ML 奖学金计划会怎样？

收到您已获得 AWS AI & ML 奖学金的通知后，Udacity 注册团队的代表将通过您在完成申请时提供给 Udacity 的电子邮件与您联系。

如果我获得了 Udacity Nanodegree 奖学金，我预计会在 Nanodegree 学习上花多少时间？

学生应该预计每周花大约 10 个小时进行 Udacity Nanodegree 学习。

对英语水平有要求吗？

尽管不需要英语水平，但学习材料是用英语写的。要取得成功，学生应该具备良好的英语阅读和写作能力。

如果我在种族/性别列表中找不到我认同的选项怎么办？

对于两者，您都可以使用选项我更喜欢自我认同（选择键入）。

什么是 AWS 玩家账号？

AWS 玩家账号是由 AWS 创建的，适用于 AWS DeepRacer 多用户和 AWS DeepRacer 学生的托管身份解决方案。您的 AWS 玩家账号持有在每项 AWS 服务中创建的所有资源。

为支持的服务创建 AWS 玩家账号

当你为[AWS DeepRacer 多用户](#)或[AWS DeepRacer 学生](#)创建账户时，你会自动创建一个 AWS 玩家账号。当您在这些服务中使用不同的功能时，新的资源会自动添加到您的 AWS 玩家账号中。要开始使用 AWS DeepRacer 多用户和 AWS DeepRacer 学生版，请使用以下链接。

创建 AWS DeepRacer 学生账号

要使用 AWS DeepRacer Student，请先创建一个帐户。要了解如何创建帐户，请参阅《AWS DeepRacer 学生用户指南》[第 1 步：注册 AWS DeepRacer Student](#)中的。

使用 AWS DeepRacer 多用户在一个账户下赞助多个参与者。

AWS DeepRacer 多用户模式支持两种不同的用户配置文件，即管理员和参与者。两者都有不同的设置要求。要开始使用，请参阅《AWS DeepRacer 开发人员指南》中的[多用户模式](#)。

删除 AWS 玩家账号

如果您删除 AWS 玩家帐户，则会立即失去对所有支持的服务的访问权限。这包括您获得的所有成就（徽章、积分、头像等）。

删除您的 AWS 玩家账号并不会删除您的 AWS 账号。如果您还想删除您的 AWS 账户，请使用[关闭 AWS 账户](#)中概述的步骤。

如果您使用 AWS 玩家账号创建了 AWS DeepRacer 多用户赛事，则无法删除您的 AWS 玩家账号。这是为了确保您所创建事件的参与者的体验不被破坏。要详细了解管理员如何在 AWS DeepRacer 多用户模式下创建事件，请使用以下主题。

使用 AWS DeepRacer 多用户模式设置事件（管理员）

要了解如何使用多用户模式创建事件，请参阅《AWS DeepRacer 开发者指南》中的[设置多用户模式（管理员）](#)。

AWS 除了在服务账号中创建的 AWS 资源外，玩家账号无权访问任何其他资源。服务账号中的任何 AWS Identity and Access Management 策略和关联资源仅限于所需的资源。

AWS DeepRacer Student 入门

利用本教程开始使用 AWS DeepRacer Student。本教程介绍如何登录 AWS DeepRacer Student、更新您的个人资料、选择加入 AWS AI & ML 奖学金申请、开始参加机器学习 (ML) 和强化学习 (RL) 方面的免费课程以及如何创建 AWS DeepRacer 模型。如果您在创建账户时没有选择加入奖学金，则可以在之后更新个人资料时选择加入。

主题

- [先决条件](#)
- [第 1 步：注册 AWS DeepRacer Student](#)
- [第 2 步：完成 AWS DeepRacer Student 的注册](#)
- [第 3 步：\(可选 \) 阅读并选择参加 AWS AI & ML 奖学金申请](#)
- [第 4 步：更新个人资料](#)
- [第 5 步：从主页浏览 AWS DeepRacer Student](#)

先决条件

要访问 AWS DeepRacer Student 并参与其中，您需要：

- 年满 16 岁、目前就读于高中、社区学院或大学。
- 或者成为高中、大学或社区学院学生的教育工作者或活动组织者。
- 请使用有效的电子邮件地址。

第 1 步：注册 AWS DeepRacer Student

您可以使用本步骤中提供的网址注册 AWS DeepRacer Student。注册时，您就会创建一个 AWS 玩家账户。此账户可用于某些其他 AWS 服务。如果您已有 AWS 玩家账户，则可以将该账户与 AWS DeepRacer Student 结合使用。

1. 打开 <http://deepracerstudent.com/> 登录页面。
2. 选择开始使用。
3. 在登录页面上，如果您还没有 AWS 玩家账户，请选择注册。

Note

如果您已有 AWS 玩家帐户，请在此处输入信息。有关使用 AWS 操作的更多信息，请参阅 [什么是 AWS 玩家账号？](#)。

4. 在注册页面输入以下信息：
 - 电子邮件地址
 - Password
5. 选择注册 系统将向您指定的地址发送一封带有确认码的电子邮件。
6. 在出现的弹出窗口中，输入验证码，然后选择验证。
7. 在 AWS DeepRacer Student 登录页面上，输入您的电子邮件地址和密码，然后选择登录。
8. 在欢迎来到 AWS DeepRacer Student 弹出窗口中，选择完成注册。

Note

您可以选择我稍后再做。如果您想稍后注册，请选择暂时注销。

第 2 步：完成 AWS DeepRacer Student 的注册

完成本节以创建您的 AWS DeepRacer Student 账户。除非另有说明，否则所有字段均为必填字段。

1. 填写添加您的个人信息以创建您的 AWS DeepRacer Student 账户部分中的字段以创建您的账户。
2. 选中该复选框以证明您是在高中、大学或社区学院就读的学生。

Note

如果是高中生，则执行以下操作：

- 对于学校，选择其他。然后，在输入学校名称字段中添加您的高中名称。
- 对于当前或潜在专业，请选择未决定或从列表中选择可能的潜在专业。

第 3 步：(可选) 阅读并选择参加 AWS AI & ML 奖学金申请

阅读有关 AWS AI & ML 奖学金以及该奖学金对象的信息，请参阅您是否想申请参加 AWS AI & ML 奖学金计划？部分。

1. 如果您符合 AWS AI & ML 奖学金的标准，则可以选择加入该奖学金申请。选中复选框以确认您想申请获得奖学金。

Note

如果您不想申请获得奖学金或尚未决定，请取消选中该复选框，然后选择其中一项我稍后再做。暂时注销。以立即退出，或者提交以继续而不选择加入。您也可以在更新个人资料时选择加入。

2. (可选) 使用下拉列表在选择性别和选择种族 (仅限美国参与者) 字段中输入您的信息。
3. 选择提交。

第 4 步：更新个人资料

要更新您的个人资料，请使用您的个人资料页面。您也可以选择加入 AWS AI & ML 奖学金计划。如果您因在 AWS DeepRacer Student League 中的表现而获得成就，则可以通过我的个人资料页面在社交媒体上分享这些成就。

更新您的用户资料


1. 在 AWS DeepRacer Student 左侧导航窗格中，选择您的个人资料。
2. 在您的个人资料页面上，在赛车手姓名中，选择更改您的赛车手姓名。

Note

您的赛车手姓名可以包含 2 到 24 个字符。允许使用字母、数字和连字符 (-)。

3. 在赛车手姓名模式中，输入您的赛车手姓名并选择保存。如果决定不想更改赛车手姓名，请选择取消。
4. 在您的个人资料信息部分，您可以更改以下字段：
 - 名称
 - 赛车手姓名

- 学校名称
- 专业名称
- 毕业年份
- 居住国

 Note

必须选择居住国才能参加 AWS DeepRacer Student League 虚拟竞赛并获得奖品。一旦提交模型参加比赛，则在当前赛季您的居住国就被锁定了。

5. (可选) 在你想申请参加 AWS AI & ML 奖学金计划吗？部分，您可以查看有关奖学金以及如何申请的信息。
 - a. 选中该复选框以选择加入奖学金计划申请。
 - b. (可选) 在选择性别和选择种族字段中输入您的信息。
 - c. 选择提交。

第 5 步：从主页浏览 AWS DeepRacer Student

AWS DeepRacer Student 主页是探索 DeepRacer Student 所有功能最理想的出发点。在主页页面上，可以执行以下操作：

开始学习机器学习 (ML) 的基础知识

您可以使用 AWS DeepRacer Student 的学习部分提供的免费课程。这款强大的课程可帮助您在 AWS DeepRacer Student 的机器学习之旅打下基础。

使用您的机器学习知识进行练习

花些时间使用学习课程后，您就可以创建和训练 AWS DeepRacer 模型了。有关更多信息，请参阅[在 AWS DeepRacer Student 中训练强化学习模型](#)。

参加 AWS DeepRacer Student 竞赛

当您完成第一个 AWS DeepRacer 模型的训练后，就可以准备好参加比赛了。如果您参加学生联赛的月度比赛，请前往公开赛部分并选择进入比赛。完成比赛后，请查看您的模型在排行榜上的表现以及进行更改和改进模型所需的数据。参加每月的虚拟比赛，您将获得积分。这些积分决定了您的全国和地区赛季排名。每月排名前三的赛车手将获得奖牌和奖品。有关更多信息，请参阅[参加 AWS DeepRacer Student 竞赛](#)。

查看模型训练时间

训练和克隆模型时，会占用一部分免费模型训练时数。您可以在主页上查看剩余的模型训练时间。

查看 AWS DeepRacer Student 赛车日历

查看赛车日历并开始为比赛日做计划。

了解 AWS AI & ML 奖学金

您可以了解 AWS AI & ML 奖学金，以及您可以做的准备。有关更多信息，请参阅[什么是 AWS AI & ML 奖学金计划？](#)。

查看其他资源

您可以找到其他能帮助您探索 AWS DeepRacer Student 的资源，例如 Discord 频道和 AWS DeepRacer 网站。这些资源可帮助您与社区的赛车手和粉丝联系，分享技巧和见解。

在 AWS DeepRacer Student 中训练强化学习模型

本演练演示了如何在 AWS DeepRacer Student 中训练您的第一个模型。还为您提供一些有用的提示，以帮助您充分利用自己的经验并加快学习进度。

第 1 步：使用 AWS DeepRacer Student 训练强化学习模型

在 AWS DeepRacer Student 中开始您的旅程，学习在哪里可以找到创建模型按钮，然后开始训练您的第一个模型。请记住，创建和训练模型是一个迭代过程。尝试不同的算法和奖励函数，以获得最佳结果。

培训强化学习模型

1. 在 AWS DeepRacer Student 主页中，选择创建模型。或者，在左侧导航窗格中导航到您的模型。在模型页面的您的模型中，选择创建模型。
2. 在概述页面中，阅读有关如何训练强化模型的信息。本页中说明了流程中的每个步骤。完成阅读后，选择下一步。

第 2 步：命名模型

为您的模型命名。最好为模型指定唯一的名称，以便在需要改进和克隆模型时快速找到各个模型。例如，您可能需要使用命名约定来命名模型，例如：*yourinitials-date-version*。

为您的模型命名

1. 在为您的模型命名页面上，在模型名称字段中输入名称。

Note

当您开始训练模型时，模型的名称即固定且不可更改。

2. 选择 Next (下一步)。

第 3 步：选择赛道

选择您的模拟赛道。赛道作为环境，为您的赛车提供数据。如果您选择非常复杂的赛道，则您的赛车需要更长的总训练时间，并且您使用的奖励函数也更加复杂。

选择您的赛道 (环境)

1. 在选择赛道页面上，选择一条赛道作为赛车的训练环境。
2. 选择 Next (下一步) 。

第 4 步：选择算法

AWS DeepRacer Student 有两种训练算法可供选择。不同的算法以不同方式最大限度地提高奖励。要获得最佳的 AWS DeepRacer Student 体验，请尝试使用这两种算法。有关算法的更多信息，请参阅 [AWS DeepRacer 训练算法](#)。

选择训练算法

1. 在选择算法类型页面上，选择算法类型。有两种算法类型可供选择：
 - Proximal Policy Optimization (PPO)。这种稳定但需要大量数据的算法在训练迭代之间执行一致。
 - Soft Actor Critic (SAC)。这种不稳定但数据效率高的算法在训练迭代之间可能执行不一致。
2. 选择 Next (下一步) 。

第 5 步：自定义您的奖励函数

奖励函数是强化学习的核心。用它来激励您的赛车 (代理) 在探索赛道 (环境) 时采取特定的行动。就像您鼓励和阻止宠物的某些行为一样，您可以使用这个工具来鼓励您的赛车尽快完成一圈，阻止它驶出赛道和曲折行驶。

训练第一个模型时，您可能需要使用默认的奖励函数示例。准备好对模型进行实验和优化时，可以通过在代码编辑器中编辑代码来自定义奖励函数。有关自定义奖励函数的详细信息，请参阅 [自定义奖励函数](#)。

要自定义您的奖励函数

1. 在自定义奖励函数页面上，选择一个奖励函数示例。有 3 种奖励函数示例可供您自定义：
 - 沿着中心线。当您的赛车自动尽可能靠近赛道中心线驾驶时，给予奖励。
 - 保持在边界内。当您的赛车自动驾驶且所有四个车轮都保持在赛道边界内时，给予奖励。
 - 防止曲折。当您的赛车保持在中心线附近时，给予奖励。如果您的赛车使用高转向角度或偏离赛道，则会受到处罚。

Note

如果您不想自定义奖励函数，请选择下一步。

2. (可选) 修改奖励函数代码。

- 选择一个奖励函数示例，然后选择 演示此代码。
- 对于代码的每个部分，您可以通过选择 + 来显示包含解释性文本的弹出式文本框来查看更多信息。在每个弹出窗口中选择下一步，继续完成代码演练。要退出弹出式文本框，请选择角落的 X。要退出演练，请选择完成。

Note

您可以通过选择使用默认代码来选择不编辑奖励函数示例代码。

- (可选) 通过选择奖励函数示例并选择编辑示例代码来编辑奖励函数示例代码。编辑代码并选择验证以检查您的代码。如果您的代码无法验证，或者您想将代码重置为其原始状态，请选择重置。

3. 选择 Next (下一步) 。

第 6 步：选择时长并将您的模型提交到排行榜

模型训练的持续时间会影响其性能。在训练的早期阶段进行试验时，您应该从此参数的较小的值开始，然后逐渐延长训练的时间。

在训练模型的这一步中，训练过的模型将提交到排行榜。您可以通过取消选中该复选框来选择退出。

选择时长并将模型提交到排行榜

1. 在选择持续时间页面上，在选择模型训练时长中选择时间。
2. 在模型描述字段中，为您的模型输入有用的描述，这将有助于您记住所做的选择。

Tip

最好添加有关模型的信息，例如奖励函数和算法的当前选择和修改，以及您对模型性能的假设。

3. 选中该复选框可在训练完成后自动将您的模型提交到 AWS DeepRacer Student 排行榜。或者，您可以通过取消选中该复选框来选择不提交您的模型。

Tip

我们建议您将模型提交到排行榜。提交模型可以帮助您了解您的模型与其他模型的对比情况，并为您提供反馈，以便您可以改进模型。

4. 选择训练您的模型。
5. 在初始化模型训练弹出窗口中，选择确定。
6. 在训练配置页面上，您可以查看模型的训练状态和配置。训练状态为进行中时，您还可以观看模型在所选赛道上训练的视频。观看视频可以帮助你获得宝贵的见解，用这些见解来改进模型。

第 7 步：在排行榜上查看模型的表现

训练完模型并将其提交到排行榜后，您可以查看其表现。

查看模型的表现

1. 在左侧导航窗格中，导航到竞赛并展开。选择一个赛季。在排行榜页面上，您的模型和排名显示在一个分区中。该页面还包括一个排行榜部分，其中包含已提交的模型列表、比赛详情和比赛详情部分。
2. 在显示排行榜的页面中，在带有您的个人资料的部分，选择观看视频以观看视频了解您的模型表现。

第 8 步：使用克隆改进模型

在您训练完模型并有选择地将模型提交到排行榜之后，您可以对其进行克隆以对其进行改进。通过使用先前训练过的模型作为新模型的起点，克隆模型可以节省步骤并提高训练效率。

克隆和改进模型

1. 在 AWS DeepRacer Student，在左侧导航窗格中导航到您的模型。
2. 在您的模型页面上，选择一个模型并选择克隆。
3. 在命名您的模型字段中，为您的克隆模型提供一个新名称，然后选择下一步。

4. 在自定义奖励函数页面上，自定义奖励函数，然后选择下一步。有关自定义奖励函数的详细信息，请参阅 [第 5 步：自定义您的奖励函数](#)。
5. 在选择持续时间页面中，在选择模型训练时长字段中输入时间，在模型描述字段中输入描述，然后选中复选框将克隆的模型提交到排行榜。
6. 选择训练您的模型。您的训练已初始化。此时将显示训练配置页面，其中包含有关您的克隆模型的信息。训练状态为进行中时，您还可以观看模型在所选赛道上训练的视频。
7. 继续克隆和修改您的预训练模型，以在排行榜上获得最佳表现。

第 9 步：（可选）下载模型

训练模型并有选择地将其提交到排行榜后，您可能需要将其下载到 AWS DeepRacer 实体设备上以备将来使用。您的模型将另存为 `.tar.gz` 文件。

下载一个模型

1. 在 AWS DeepRacer Student，在左侧导航窗格中导航到您的模型。
2. 在您的模型页面上，选择一个模型并选择下载。
3. 在浏览器中跟踪模型下载的进度。下载模型后，您可以将其保存到本地硬盘或其他首选存储设备中。

要了解有关使用 AWS DeepRacer 设备的更多信息，请参阅 AWS DeepRacer 指南中的 [操作您的 AWS DeepRacer 车辆](#)。

参加 AWS DeepRacer Student 竞赛

在模拟中成功训练和评估您的模型后，通过参加竞赛将您模型的性能与其他赛车手的模型进行比较。竞赛是一种很好的方式，让您可以获取模型反馈、赢取奖项和奖品、与其他 AWS DeepRacer Student 社区成员进行虚拟交流、了解学习和提高技能的机会，还能享受乐趣。有两种类型的学生竞赛：学生联赛和社区竞赛。

学生联赛竞赛是每月一次的虚拟竞赛，所有学生都可以参加。学生社区竞赛是由教育工作者或活动组织者在 AWS 控制台中创建的私人竞赛，学生需要获得邀请才能参加。

本节讨论如何参加 AWS DeepRacer Student 学生联赛竞赛和学生社区竞赛。

参加学生联赛竞赛

在本节中，了解如何将您的模型提交给 AWS DeepRacer Student 学生联赛竞赛。您可以通过直接向学生排行榜提交经过训练的模型来参加每个月的竞赛。有关模型训练的更多信息，请参阅 [在 AWS DeepRacer Student 中训练强化学习模型](#)。

要参加学生联赛竞赛

1. 在 AWS DeepRacer Student 的左侧导航窗格中，展开竞赛，然后选择学生联赛。
2. 该页面显示比赛详情和排行榜。在公开赛部分，选择进入比赛提交模型，或者如果您尚未创建模型，请选择创建模型。有关更多信息，请参阅 [在 AWS DeepRacer Student 中训练强化学习模型](#)。
3. 在选择参赛模型部分，使用下拉列表在选择模型字段中选择模型。
4. 选择参加比赛以提交您的模型。
5. 根据赛车标准成功评估您的模型后，请查看排行榜，了解您的模型与其他参赛者的排名情况。
6. （可选）选择观看以观看视频了解您的赛车表现。
7. 选择再次比赛以进入其他模型。每场 AWS DeepRacer Student League 竞赛的参赛名额为 50。

参加学生社区竞赛

在本节中，了解如何将您的模型提交给 AWS DeepRacer Student 学生社区竞赛。您可以通过电子邮件收到教育工作者或活动组织者的邀请链接，参加学生社区竞赛。

要参加学生社区竞赛

1. 前往邀请链接并登录您的 AWS DeepRacer 学生账户。
2. 登录后，选择进入比赛按钮。
3. 在选择参赛模型下拉列表中，选择要在社区竞赛中使用的模型。
4. 选择进入比赛提交您的模型。
5. 根据赛车标准成功评估您的模型后，请查看排行榜，了解您的模型与其他参赛者的模型排名情况。
6. (可选) 选择观看以观看视频了解您的赛车表现。
7. 选择再次比赛以进入其他模型。

自定义奖励函数

创建奖励函数就像设计激励计划一样。参数是可用于制定激励计划的值。

不同的激励策略会导致不同的车辆行为。若要让车辆更快地行驶，请尝试在赛车花费太长时间才能完成一圈或偏离赛道时奖励负值。为了避免车辆出现曲线驾驶模式，请尝试设定一个转向角度的范围限制，并对车辆在直线赛道上较少激进转向时给予奖励。

您可以使用航点（沿着赛道中心线以及外缘和内边缘放置的编号标记）来帮助您将某些驾驶行为与赛道的特定特征（例如直道和曲线）相关联。

设计有效的奖励函数是一个创造性和迭代性的过程。尝试不同的策略，混合搭配参数，最重要的是，玩得开心！

主题

- [编辑 Python 代码以自定义您的奖励函数](#)
- [AWS DeepRacer 奖励函数的输入参数](#)

编辑 Python 代码以自定义您的奖励函数

在 AWS DeepRacer Student 中，您可编辑示例奖励函数，为你的模型制定自定义的赛车策略。

要自定义您的奖励函数

1. 在 AWS DeepRacer Student 创建模型体验的第 5 步：自定义奖励函数页面上，选择一个奖励函数示例。
2. 使用奖励函数示例选择器下方的代码编辑器，使用 Python 代码自定义奖励函数的输入参数。
3. 选择验证以检查您的代码是否有效。或者，选择重置重新开始。
4. 完成更改后，选择下一步。

使用 [AWS DeepRacer 奖励函数的输入参数](#) 了解每个参数。在奖励函数示例中查看如何使用不同的参数。

AWS DeepRacer 奖励函数的输入参数

AWS DeepRacer 奖励函数将作为变量传递的字典对象 `params` 作为输入。

```
def reward_function(params) :

    reward = ...

    return float(reward)
```

params 词典对象包含以下键/值对：

```
{
    "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
```

```
}
```

使用以下参考来更好地理解 AWS DeepRacer 输入参数。

all_wheels_on_track

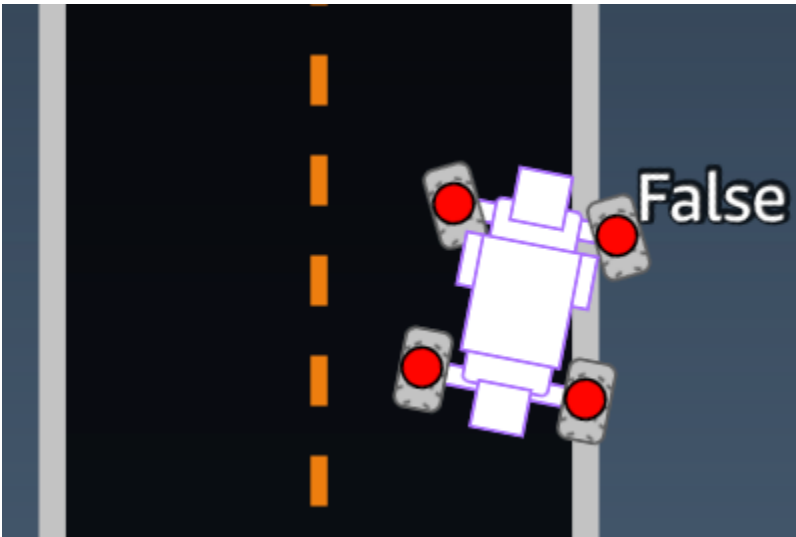
类型 : Boolean

范围 : (True:False)

一个 Boolean 标记，指示代理是在赛道上还是偏离赛道。如果车辆的任一车轮位于赛道边界外，则将该代理视为偏离赛道 (False)。如果车辆的四个车轮都在内部和外部赛道边界内，则将车辆视为在赛道上 (True)。下图显示了代理在赛道上。



下图显示了由于两个轮子在赛道边界之外而未进入赛道的代理。



示例：使用 `all_wheels_on_track` 参数的奖励函数

```
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

类型 : [int, int]

范围 : [(0:Max-1),(1:Max-1)]

最接近代理当前位置 (x, y) 的两个相邻 waypoint 的从零开始的索引。距离是根据与代理中心的欧氏距离来测量的。第一个元素指代理后面最近的路点，第二个元素指代理前面最近的路点。Max 是路点列表的长度。在[waypoints](#)的图示中，closest_waypoints 将为 [16, 17]。

以下示例奖励函数演示如何使用 waypoints、closest_waypoints 和 heading 来计算即时奖励。

AWS DeepRacer 支持以下 Python 库 : math、random、numpy、scipy 和 shapely。要使用其中一个，请在函数定义 def reward_function(params) 之前添加一个导入语句 import *supported library*。

示例 : 使用 *closest_waypoints* 参数的奖励函数。

```
# 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

类型：[int, int]

范围：[(0:len(object_locations)-1), (0:len(object_locations)-1)]

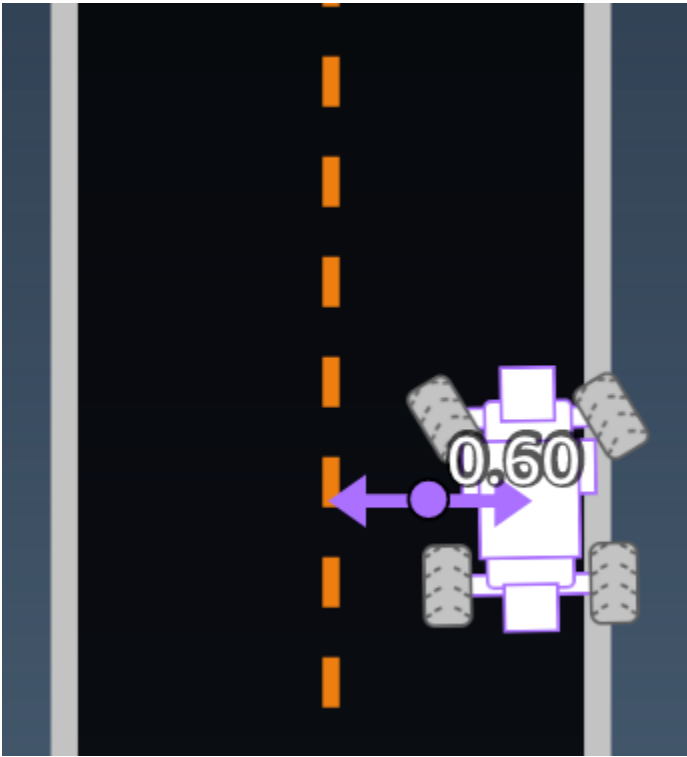
最接近代理当前位置 (x, y) 的两个物体的从零开始的索引。第一个索引指代理后面最近的物体，第二个索引指代理前面最近的物体。如果只有一个物体，则两个索引都为 0。

distance_from_center

类型：float

范围：0:~track_width/2

代理中心和赛道中心之间的位移 (以米为单位)。当代理的任一车轮位于赛道边界外时可观察到的最大位移, 并且根据赛道边界的宽度, 它可以略小于或大于 `track_width` 的一半。



示例：使用 `distance_from_center` 参数的奖励函数

```
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

类型：float

范围：-180:+180

代理相对于坐标系 x 轴的前进方向（以度为单位）。



示例：使用 *heading* 参数的奖励函数

有关更多信息，请参阅[closest_waypoints](#)。

is_crashed

类型：Boolean

范围：(True:False)

一个 Boolean 标记，用于指示代理的最终状态是撞向另一个物体 (True) ，还是不是 (False)

is_left_of_center

类型：Boolean

范围：[True : False]

Boolean 标志，用于指示代理是位于赛道中心左边 (True) 还是不在赛道中心左边 (False)。

is_offtrack

类型：Boolean

范围：(True:False)

Boolean 标志，用于指示代理的四个车轮已驶出赛道的内部或外部边界 (True) 与否 (False)。

is_reversed

类型：Boolean

范围：[True:False]

Boolean 标志，用于指示代理是顺时针 (True) 还是逆时针 (False) 行驶。

此参数在您针对每个过程改变方向时使用。

objects_distance

类型：[float, ...]

范围：[(0:track_length), ...]

环境中物体之间相对于起始线的距离列表。第 i 个元素测量沿赛道中心线第 i 个物体与起始线之间的距离。

Note

$\text{abs} | (\text{var1}) - (\text{var2}) |$ = 汽车与物体的接近程度, WHEN $\text{var1} = [\text{"objects_distance"}][\text{index}]$ and $\text{var2} = \text{params}[\text{"progress"}] * \text{params}[\text{"track_length"}]$

要获取车辆前方最近物体和车辆后方最近物体的索引，请使用 `closest_objects` 参数。

objects_heading

类型：[float, ...]

范围：[(-180:180), ...]

物体前进方向 (以度为单位) 列表。第 i 个元素测量第 i 个物体的前进方向。静止物体的标题为 0。对于自动程序车辆，相应元素的值是机器人汽车的前进角度。

objects_left_of_center

类型：[Boolean, ...]

范围：[True|False, ...]

Boolean 标记的列表 第 i 个元素值指示第 i 个物体位于赛道中心的左侧 (True) 还是右侧 (False)。

objects_location

类型：[(x,y), ...]

范围：[(0:N,0:N), ...]

此参数存储所有物体位置。每个位置都是 [\(x,y\)](#) 的元组。

列表的大小等于赛道上的物体数。列出的物体包括静止的障碍物和移动的机器人赛车。

objects_speed

类型：[float, ...]

范围：[(0:12.0), ...]

赛道上物体的速度 (米/秒) 列表。对于静止物体，其速度为 0。对于自动程序车辆，值为您在训练中设置的速度。

progress

类型：float

范围：0:100

赛道完成百分比。

示例：使用 *progress* 参数的奖励函数

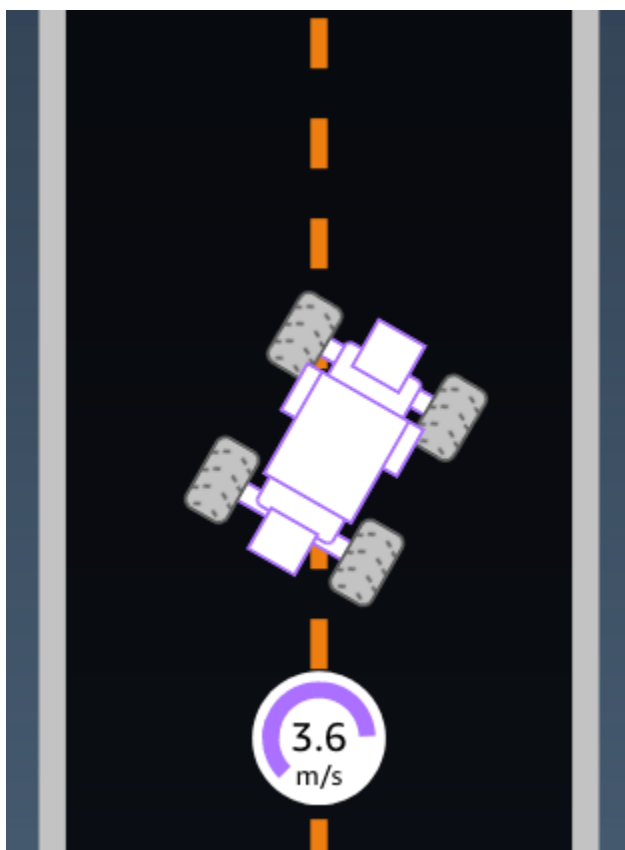
有关更多信息，请参阅[步骤](#)。

speed

类型：float

范围：0.0:5.0

观察到的代理速度，以米/秒 (m/s) 为单位。



示例：使用 *speed* 参数的奖励函数

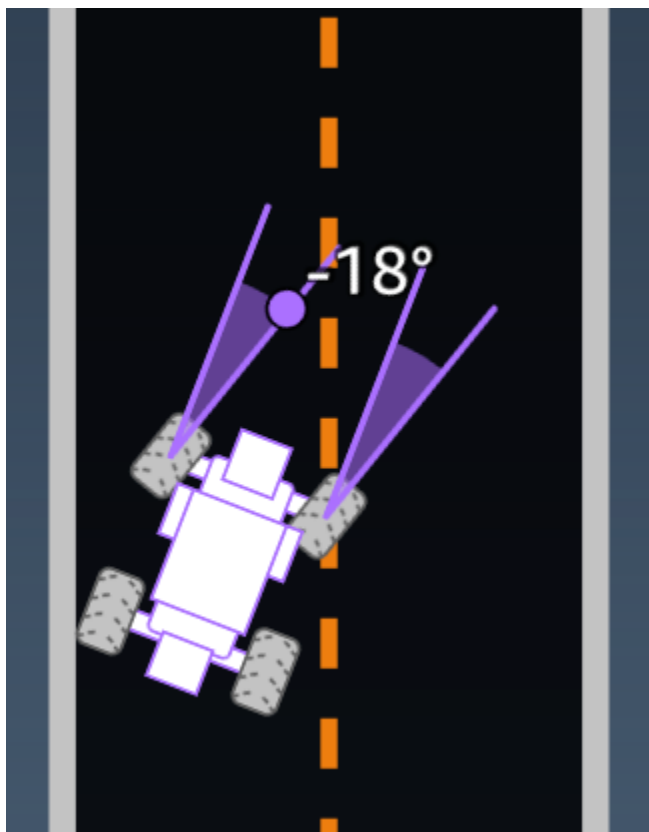
有关更多信息，请参阅[all_wheels_on_track](#)。

steering_angle

类型：float

范围：-30:30

前轮与代理中心线之间的转向角 (以度为单位)。负号 (-) 表示向右转向，正号 (+) 表示向左转向。代理的中心线不一定与赛道中心线平行，如下图所示。



示例：使用 *steering_angle* 参数的奖励函数

```
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

类型：int

范围：0:N_{step}

完成的步骤数。一个步骤对应于代理使用当前策略完成的一个观察操作序列。

示例：使用 *steps* 参数的奖励函数

```
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

类型：float

范围： $[0:L_{\max}]$

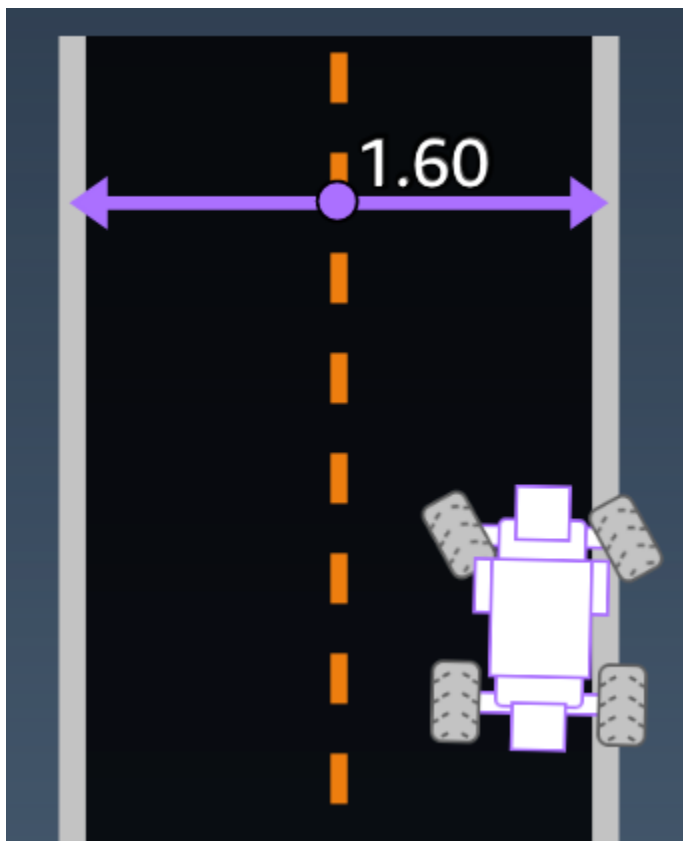
赛道长度 (以米为单位)。 L_{\max} is track-dependent.

track_width

类型：float

范围： $0:D_{\text{track}}$

赛道宽度 (以米为单位)。



示例：使用 *track_width* 参数的奖励函数

```
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

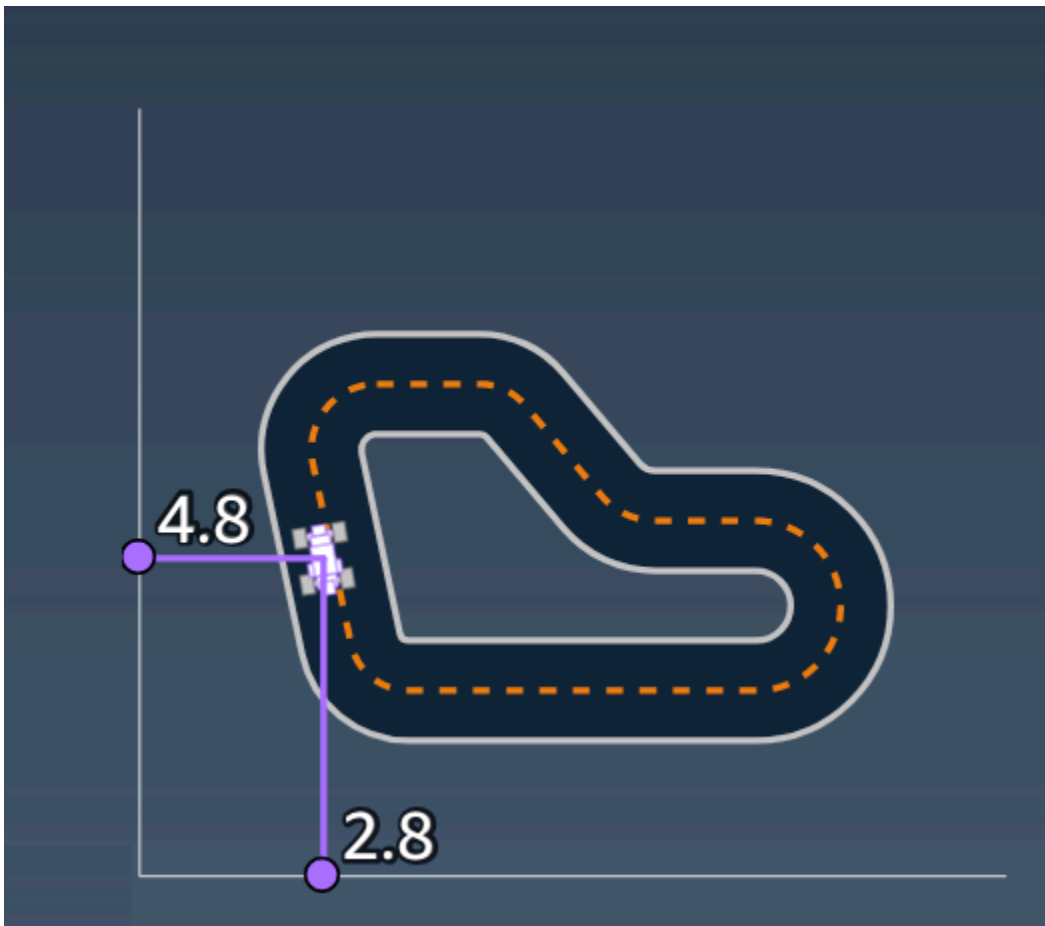
return float(reward)
```

x, y

类型：float

范围：0:N

包含赛道的模拟环境的沿 x 和 y 轴的代理中心位置（以米为单位）。原点位于模拟环境的左下角。

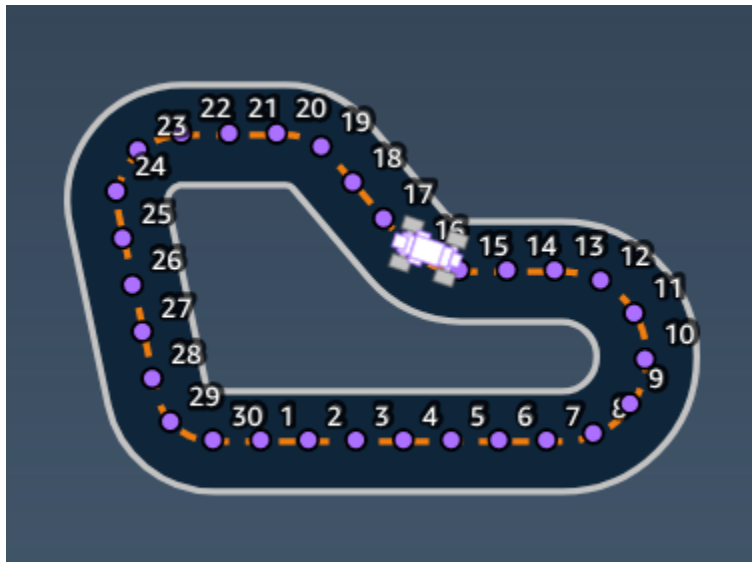


waypoints

类型：`[float, float]` 的 list

范围：`[[xw,0, yw,0] ... [xw,Max-1, yw,Max-1]]`

沿赛道中心排列、取决于赛道的 Max 里程的有序列表。每个里程碑均由 $(x_{w,i}, y_{w,i})$ 坐标描述。对于环形赛道，第一个路径点与最后一个路径点相同。对于直道和其他非环形赛道，第一个路径点与最后一个路径点不同。



示例：使用 `waypoints` 参数的奖励函数

有关更多信息，请参阅[closest_waypoints](#)。

AWS DeepRacer Student 的安全

AWS 十分重视云安全性。作为 AWS 客户，您将从专为满足大多数安全敏感型企业的要求而打造的数据中心和网络架构中受益。

安全性是 AWS 和您的共同责任。[责任共担模式](#) 将其描述为云的安全性和云中的安全性：

- 云的安全性 – AWS 负责保护在 AWS 云中运行 AWS 服务的基础设施。AWS 还向您提供可安全使用的服务。第三方审核员定期测试和验证我们的安全性的有效性，作为 [AWS Compliance Programs](#) 的一部分。要了解适用于 AWS DeepRacer Student 的合规性计划，请参阅 [合规性计划范围内的 Amazon Web Services](#)。
- 云中的安全性 - 您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您的公司的要求以及适用的法律法规。

此文档将帮助您了解如何在使用 AWS DeepRacer Student 时应用责任共担模式。它说明了如何配置 AWS 以实现您的安全性和合规性目标。您还会了解如何使用其它 AWS 服务以帮助您监控和保护 AWS DeepRacer Student 资源。

目录

- [AWS DeepRacer Student 中的数据保护](#)
- [适用于 AWS DeepRacer Student 的 Identity and Access Management](#)
- [AWS DeepRacer Student 的合规性验证](#)
- [AWS DeepRacer Student 的弹性](#)
- [AWS DeepRacer Student 中的基础设施安全](#)

AWS DeepRacer Student 中的数据保护

以下各节说明了 AWS DeepRacer Student 会捕获哪些数据，以及 AWS DeepRacer Student 在哪里使用数据加密来保护您的数据。

当您创建 AWS DeepRacer Student 账户时，您同时会创建一个 AWS 玩家账户。在您的 AWS DeepRacer Student 账户中创建的资源存储在您的 AWS 玩家账户中。有关 AWS 玩家账户的更多详情，请参阅 AWS DeepRacer Student 用户指南中的 [什么是 AWS 玩家账号？](#)。

主题

- [在 AWS DeepRacer Student 门户中捕获的数据](#)
- [AWS DeepRacer Student 门户中的静态加密](#)
- [AWS DeepRacer Student 门户中的传输加密](#)

在 AWS DeepRacer Student 门户中捕获的数据

要使用 AWS DeepRacer Student 门户，所需的数据将存储在您的 AWS 玩家账户中。AWS DeepRacer Student 门户中捕获的数据不用于帮助改进服务。

在 AWS DeepRacer Student 中捕获的数据。

以下是在 AWS DeepRacer Student 中创建并存储在 AWS 玩家账户中的数据摘要。

- 您用于注册帐户的电子邮件地址和密码。
- 您的赛车手名字
- 您在 Student League 排行榜上的排名
- 您经过训练的模型
- 奖励函数代码

AWS DeepRacer Student 门户中的静态加密

默认情况下，AWS DeepRacer Student 门户上捕获的数据是加密的。

AWS 玩家账户使用 Amazon Cognito 来加密和存储用于登录 AWS DeepRacer Student 的电子邮件和密码。有关更多信息，请参阅 [Amazon Cognito 中的数据保护](#)。

在 AWS DeepRacer Student 中捕获的所有其他数据都使用 AWS 自有密钥通过 AWS Key Management Service AES-GCM 和大小为 256 位的密钥在云中进行静态加密。这些数据在 Amazon Simple Storage Service (S3) 和 Amazon DynamoDB 中存储并加密。

AWS DeepRacer Student 门户中的传输加密

您的注册和授权电子邮件地址使用客户端加密进行加密。[在 AWS DeepRacer Student 中捕获的所有其他数据](#)都将从您的账户中复制出来，并在内部 AWS 系统中进行处理。默认情况下，AWS DeepRacer Student 使用通过 HTTPS 的安全连接对传输中数据进行加密。

适用于 AWS DeepRacer Student 的 Identity and Access Management

Amazon Identity and Access Management (IAM) 是一个服务，可以帮助管理员安全地控制对资源的访问。IAM 管理员控制可以AWS通过身份验证（登录）和授权（具有权限）使用资源的人员。DeepRacer Student 不会直接与 IAM 集成来控制用户对 AWS 资源的访问权限。相反，AWS DeepRacer Student 使用由 AWS DeepRacer 管理的经过身份验证的代理 API 来保护用户资源。

AWS DeepRacer Student 的合规性验证

作为多个 AWS 合规性计划的一部分，第三方审计员将评估 AWS 服务的安全性和合规性。

有关特定合规性计划范围内的 AWS 服务列表，请参阅[合规性计划范围内的 Amazon Web Services](#)。有关常规信息，请参阅[AWS 合规性计划](#)。

您可以使用 AWS Artifact 下载第三方审计报告。有关更多信息，请参阅[下载 AWS Artifact 中的报告](#)。

AWS 提供以下资源来帮助实现合规性：

- [安全性与合规性 Quick Start 指南](#) - 这些部署指南讨论了架构注意事项，并提供了在 AWS 上部署基于安全性和合规性的基准环境的步骤。
- [AWS 合规性资源](#) - 此业务手册和指南集合可能适用于您的行业和位置。
- 《AWS Config 开发人员指南》中的[使用规则评估资源](#) - AWS Config；评估您的资源配置对内部实践、行业指南和法规的遵循情况。
- [AWS Security Hub](#) - 此AWS服务提供了AWS中安全状态的全面视图，可帮助您检查是否符合安全行业标准 and 最佳实践。

AWS DeepRacer Student 的弹性

AWS全球基础设施围绕AWS区域和可用区构建。区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现故障转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

有关AWS区域和可用区的更多信息，请参阅[AWS全球基础设施](#)。

AWS DeepRacer Student 中的基础设施安全

作为一项托管式服务，AWS DeepRacer Student 由 [《亚马逊云科技：安全流程概述》](#) 白皮书中所述的亚马逊云科技 AWS 全球网络安全程序提供保护。

解决常见的 AWS DeepRacer 学生问题

主题

- [为什么我的 AWS DeepRacer 学生账户被自动注销？](#)
- [如何选择退出 AWS AI & ML 奖学金计划？](#)
- [我无法删除我的 AWS DeepRacer 学生账户](#)
- [我在下拉列表找不到我的学校名称](#)
- [我无法继续训练我的模型](#)
- [我收到“该电子邮件已注册账户”的错误消息](#)
- [我使用了 Gmail 账户进行注册，但找不到我的验证码](#)

为什么我的 AWS DeepRacer 学生账户被自动注销？

根据 AWS 安全政策，您将在 30 天后自动退出 AWS DeepRacer 学生账户。

- 要继续使用该服务，请导航到[AWS DeepRacer 学生登录页面](#)，然后使用您的凭据重新登录。

如何选择退出 AWS AI & ML 奖学金计划？

AWS AI & ML 奖学金计划是可选的，面向 16 岁或以上服务欠缺和代表性不足的学生。注册 AWS DeepRacer 学生时，默认情况下，您并未加入 AWS AI 和 ML 奖学金计划。

要参加，您必须先勾选“您想被考虑参加 AWS AI & ML 奖学金计划吗？”中的复选框来选择加入。在您注册 Student 时或之后从“您的个人资料”页面上注册 AWS DeepRacer 学生版块，该页面可从网站的左侧导航窗格中访问。

- 选择加入该计划仅意味着您可以获取申请流程。您也可以选择不申请。

我无法删除我的 AWS DeepRacer 学生账户

如果您无法删除 AWS DeepRacer 学生帐户，请查看是否创建了 AWS DeepRacer 多用户活动。AWS 玩家账户是由为 AWS DeepRacer 多用户和 AWS DeepRacer 学生创建 AWS 的托管身份解决方案。您的 AWS 玩家账户持有在每项 AWS 服务中创建的所有资源。

- 为确保您创建的活动的参与者不会获得中断的体验，如果您的 AWS DeepRacer 学生帐户包含用于 AWS DeepRacer 多用户活动的资源，则无法将其删除。

我在下拉列表中找到我的学校名称

您可能无法在下拉列表中找到所有学校，尤其是高中。

- 如果您的学校不在下拉列表中，请选择其他，然后输入您的学校名称。

我无法继续训练我的模型

您可能已超过每月模型训练时间限制。

- 前往主页，在剩余模型训练时数部分中查看您的已使用训练时间。如果您已超过模型训练时间，请等到您的时间重置后再开始训练。

我收到“该电子邮件已注册账户”的错误消息

当您在 AWS 玩家账户注册页面中输入确认码，并且已经使用相同的电子邮件地址通过 AWS DeepRacer 多用户注册 AWS 玩家账户时，您会收到此错误消息。当您之前使用相同的电子邮件地址注册 AWS DeepRacer 学生账户时，您也会收到此错误。

- 使用您之前创建的凭据[登录AWS DeepRacer 学生登录页面](#)，或者通过选择忘记密码？来请求重置密码？在“密码”字段下。

我使用了 Gmail 账户进行注册，但找不到我的验证码

如果您使用 Gmail 帐户注册了 AWS 玩家帐户，但找不到验证码消息，则可能是该消息已发送到错误的文件夹。

- 登录您的 Gmail 账户，查看您的促销文件夹，查看标题为“您的 AWS 玩家个人资料验证码”的消息。

AWS DeepRacer Student 的配额

每位参加 AWS DeepRacer Student 的学生都会获得 10 小时免费的每月模型训练计算资源和 5GB 的存储空间。

删除您的 AWS DeepRacer 学生账户

AWS DeepRacer 学生门户网站在您的 AWS 玩家账户中存储以下信息：

- 电子邮件地址
- 您的 密码
- 您的赛车手名字
- 您的 Student League 排行榜排名

要了解有关所收集数据的更多信息，请参阅[AWS DeepRacer Student 中的数据保护](#) 《AWS DeepRacer 学生用户指南》。

如果您想从 AWS 的服务器中删除此信息，请使用以下步骤删除您的 AWS DeepRacer 学生门户帐户。删除您的 AWS DeepRacer 学生账号也会删除您的 AWS 玩家账号和所有相关资源。

要了解有关 AWS 玩家账号的更多信息，请参阅 [什么是 AWS 玩家账号？](#)

Note

如果您创建了 AWS DeepRacer 多用户赛事，则无法删除您的 AWS 玩家账号。有关更多详细信息，请参阅[我无法删除我的 AWS DeepRacer 学生账户](#) 《AWS DeepRacer 学生用户指南》。

删除您的 AWS DeepRacer 学生玩家账号

Important

删除 AWS DeepRacer 学生账户的操作无法撤销。当你删除 AWS DeepRacer 学生时，你也是在删除你的 AWS 玩家账号和所有相关资源。

当您删除 AWS DeepRacer 学生账户时，您的 AWS 玩家账户中的资源将在一年内从我们的服务器上删除。

1. 打开 AWS DeepRacer 学生登录页面：<https://student.deepracer.com/signIn>。
2. 如果出现提示，请登录您的 AWS DeepRacer 学生账户。

3. 选择您的账户。
4. 在您的账户页面上，选择删除您的账户。
5. 在要确认删除，请在字段中键入“删除”下，键入 **Delete**。
6. 选择删除。

成功删除您的帐户后，将出现“成功删除帐户”消息，您将返回到 AWS DeepRacer 学生登录页面。

如果您还想删除您的 AWS 账户，请使用[关闭 AWS 账户](#)中概述的步骤。

我们知道客户非常关心隐私和数据安全，因此我们采用可靠且复杂的措施，从技术和物理层面进行控制，防止未经授权访问或披露客户内容。维系客户信任是一项持续的承诺。您可以在我们的隐私声明页面上了解有关 AWS 数据[隐私](#)承诺的更多信息。

AWS DeepRacer Student 用户指南的文档历史记录

下表介绍了 AWS DeepRacer Student 的文档版本。

变更	说明	日期
2024 年计划的更新	有关 2024 年 AWS AI & ML 奖学金计划和 AWS DeepRacer Student League 主题的更新。增加了有关新课程的信息，所有申请 AWS AI & ML 奖学金的学生均可参加。	2023 年 11 月 1 日
2023 年 AWS DeepRacer Student League 的更新	有关 2023 年 AWS DeepRacer Student League 多个主题的更新。有关更多信息，请参阅 条款和条件 。	2023 年 3 月 1 日
首次发布	AWS DeepRacer Student 用户指南的首次发布，包括对 AWS AI & ML 奖学金计划和 AWS DeepRacer Student League 的支持。	2021 年 12 月 1 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。