

[David Silver](https://medium.com/@dsilver829?source=post_header_lockup) - I love self-driving cars and I work on them at @Udacity! Learn to build self-driving cars with us: <https://udacity.com/drive> | Oct 5, 2016

原文发表在 Medium, 点击[此处第1、2学期](https://medium.com/self-driving-cars/term-1-in-depth-on-udacitys-self-driving-car-curriculum-ffcf46af0c08#.kmdgomrst)和[此处](https://medium.com/udacity/term-3-in-depth-on-udacitys-self-driving-car-curriculum-15d03e45d7ea)查看原文。

# **优达学城**

# **无人驾驶车工程师纳米学位课程计划**



我们努力将这门课程打造为全球最好的无人驾驶汽车工程师培训课程。整个课程共分为 3 个学期每学期 3 个月你大约需要 9 个月时间完成整个纳米学位项目。以下是课程的详细内容

### **第一学期**

#### **简介**

1. 与讲师见面 包括  [Sebastian Thrun](https://en.wikipedia.org/wiki/Sebastian_Thrun)、[Ryan Keenan](https://www.linkedin.com/in/ryan-keenan-980759a6) 和我自己。了解无人驾驶车的组成系统以及此纳米学位项目的授课内容和授课形式。

**2. 实战项目检测车道线**

从视频流中获取的图像检测高速公路车道线。使用 OpenCV 图像分析技术包括霍夫变换Hough transforms、Canny 边缘检测、颜色选择切图ROI, region of interest selection灰度处理高斯模糊等方法来识别车道线。

#### **深度学习**

1. 机器学习回顾机器学习的基础知识包括回归和分类。

2. 神经网络了解感知器、激活函数和基本神经网络。使用 Python 语言搭建你自己的神经网络。

3. 逻辑分类器Logistic Classifier研究如何使用机器学习训练逻辑分类器。在 TensorFlow 系统中实现逻辑分类器。

4. 优化Optimization研究分类器性能优化技术包括验证和测试集、梯度下降、动量和学习速率等。

5. 修正线性单元Rectified Linear Units评估激活函数及它们对性能的影响。

6. 正则化Regularization学习丢失数据dropout等技术以避免过拟合训练数据。

7. 卷积神经网络CNN研究卷积神经网络的构建模块包括过滤器、步幅stride和池化pooling。

**8. 实战项目交通标志识别和分类**

实现和训练卷积神经网络对交通标志进行分类。使用验证集、池化和丢弃数据技术选择神经网络架构并提高性能。

9. Keras在 Keras 中构建一个多层卷积神经网络。比较 Keras 的简单性与 TensorFlow 的灵活性。

10. 迁移学习Transfer Learning微调已经经过训练的神经网络来解决你自己的问题。研究 AlexNet、VGG、GoogLeNet 和 ResNet 等标准神经网络。

**11. 实战项目行为克隆**

架构和训练一个深度神经网络以在模拟器中驾驶汽车。收集你自己的训练数据并用它来在测试跑道上克隆你自己的驾驶行为模式。

#### **计算机视觉**

1. 摄像头学习摄像头的物理知识以及如何校准、反失真和转换图像角度。

2. 车道检测研究可以应对弯道、恶劣天气和不同照明情况的先进车道检测技术。

**3. 实战项目高级车道检测**

在各种条件下检测车道线包括不断变化的路面、弯道和变化的照明条件。使用 OpenCV、滤波器、多项式拟合和样条曲线spline实现摄像头校准和变换。

4. 支持向量机SVM实现支持向量机并应用于图像分类。

5. 决策树实现决策树并应用于图像分类。

6. 定向梯度直方图Histogram of Oriented Gradients实现定向梯度的直方图并将其应用于图像分类。

7. 深度神经网络比较支持向量机、决策树、定向梯度直方图和深度神经网络的分类性能。

8. 车辆跟踪学习如何将图像分类技术应用于车辆跟踪以及使用基本过滤器随时间变化整合车辆位置。

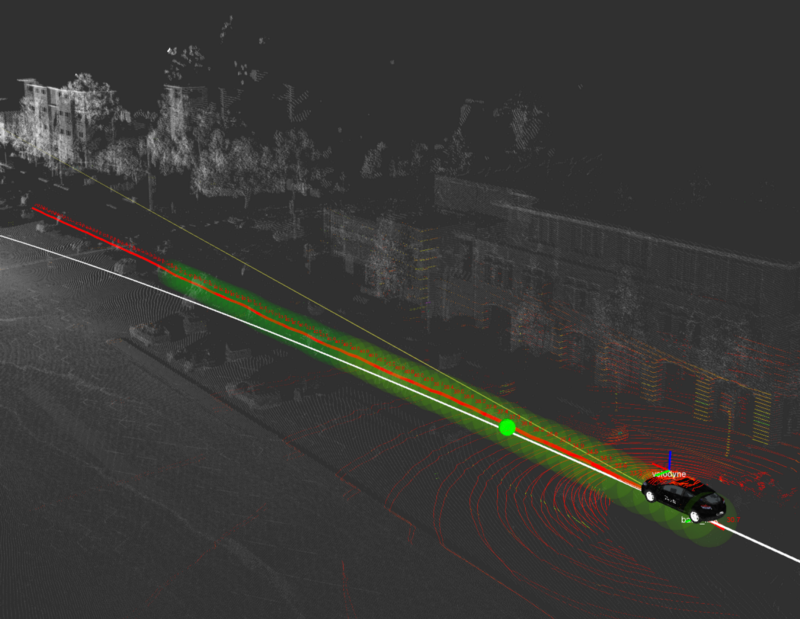
**9. 实战项目车辆跟踪**

使用图像分类器例如 SVM、决策树、HOG 和 DNN跟踪摄像头图像中的车辆。应用过滤器融合位置数据。

### **第二学期**

#### **传感器融合Sensor Fusion**

我们将本学期划分成了数个模块每个模块都是由一系列有所侧重的课程组成。这个传感器融合模块是由我们与我们在梅赛德斯-奔驰的合作伙伴共同完成的。[梅赛德斯-奔驰](http://mbrdna.com/)的团队实力超群。他们是世界级的汽车工程师将无人驾驶技术应用到一些世界上最好的车辆上。他们也是[优达学城的招聘合作伙伴](http://blog.udacity.com/2016/10/new-hiring-partners-self-driving-car-engineer-nanodegree.html)这表示我们一起开发的课程特别侧重于培养与提升那些他们希望雇佣的人才



激光雷达点云

下面是传感器融合模块每节课程的内容

1. 传感器

2. 传感器融合模块的第一课涵盖了在无人驾驶汽车上的两种最重要的传感器——雷达与光学雷达背后的物理

3. 卡尔曼滤波器Kalman filter

4. 卡尔曼滤波器是将数据融合的重要工具运用 Python 实现这些滤波器并融合单一传感器实时数据

5. C++ Primer

6. 复习 C++ 的关键概念并实现第二学期的项目

7. 项目用 C++ 实现扩展卡尔曼滤波器

8. 无人驾驶工程师运用扩展卡尔曼滤波将多个传感器的测量数据融合进一个非线性模型。成功建立一个卡尔曼滤波器会令你的潜在雇主印象深刻

9. 无迹卡尔曼滤波器

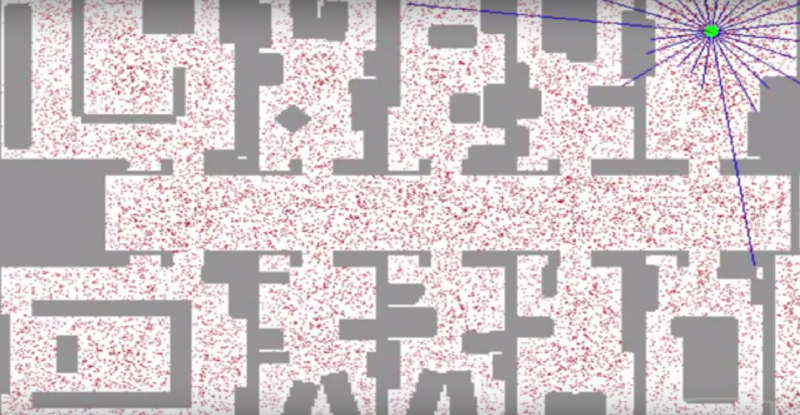
10. 无迹卡尔曼滤波器Unscented Kalman filter用数学上的复杂的方法结合传感器的数据。在许多场合下无迹卡尔曼滤波器表现的要比扩展卡尔曼滤波器好。这是传感器融合工程师用于真实的无人驾驶汽车的传感器融合方法

11. 实战项目行人追踪

12. 融合含有噪声的雷达与激光雷达数据以追踪一名行人的行动轨迹。

#### 定位Localization

这一模块也是由我们在[梅赛德斯-奔驰](http://mbrdna.com/)的合作伙伴完成的他们将前沿的定位技术应用于他们自己的无人驾驶汽车我们将一起向学生展示如何运用并实现每个定位工程师都需要掌握的基本算法。



粒子滤波

定位模块由以下课程组成

1. 运动 (motion)

2. 学习运动和概率将如何影响你对你所在位置的认识。

3. 马尔科夫定位

4. 运用一个简单的贝叶斯滤波器定位一辆在简单环境中的车辆。

5. 自运动

6. 学习车辆移动的基本模型其中包括自行车模型。运用不同传感器所提供的数据估计汽车的位置。

7. 粒子滤波

8. 运用一种概率抽样技术即粒子滤波来定位处于复杂环境中的车辆

9. 高性能粒子滤波

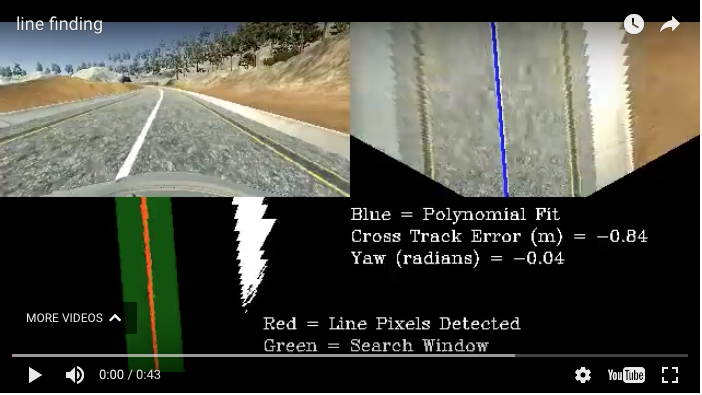
10. 用 C++ 实现粒子滤波

11. 项目定位车辆

12. 实现粒子滤波并利用现实世界中的数据定位一辆丢失的车辆

#### **控制**

这一模块是有我们与 [Uber 先进技术团队](https://www.uber.com/info/atg/)Uber Advanced Technologies Group合作的。Uber 是无人驾驶领域进展最快的公司之一。在美国他们已经在多个地点测试了他们的无人驾驶车并且他们很乐于为大家介绍无人驾驶汽车所用到的核心控制算法。Uber 先进技术团队也是优达学城的招聘合作伙伴因此如果你想在 Uber 工作请关注他们的课程。



优达学城模拟器中的一个控制器

此模块包括以下课程

1. 控制

2. 学习控制系统是如何让车辆准确地行驶

3. PID 控制比例微分积分控制

4. 实现经典的闭环控制器——比例微分积分控制系统

5. 线性二次型调节器

6. 实现一个更加复杂的控制算法使车辆在一个充满噪声的环境中保持稳定

7. 项目保持车道

8. 实现一个控制器并保持车辆行驶在线内。额外的挑战运用计算机视觉技术识别车道线并评估航迹误差。

### **第三学期**

This term is three months long, and features a different module each month.

The first month focuses on path planning, which is basically the brains of a self-driving car. This is how the vehicle decides where to go and how to get there.

The second month presents an opportunity to specialize with an elective; this is your chance to delve deeply into a particular topic, and emerge with a unique degree of expertise that could prove to be a key competitive differentiator when you enter the job market. We want your profile to stand out to prospective employers, and specialization is a great way to achieve this.

The final month is truly an *Only At Udacity* experience. In this System Integration Module, you will get to put your code on Udacity’s very own self-driving car! You’ll get to work with a team of students to test out your skills in the real world. We know firsthand from our hiring partners in the autonomous vehicle space that this one of the things they value most in Udacity candidates; the combination of software skills and real-world experience.

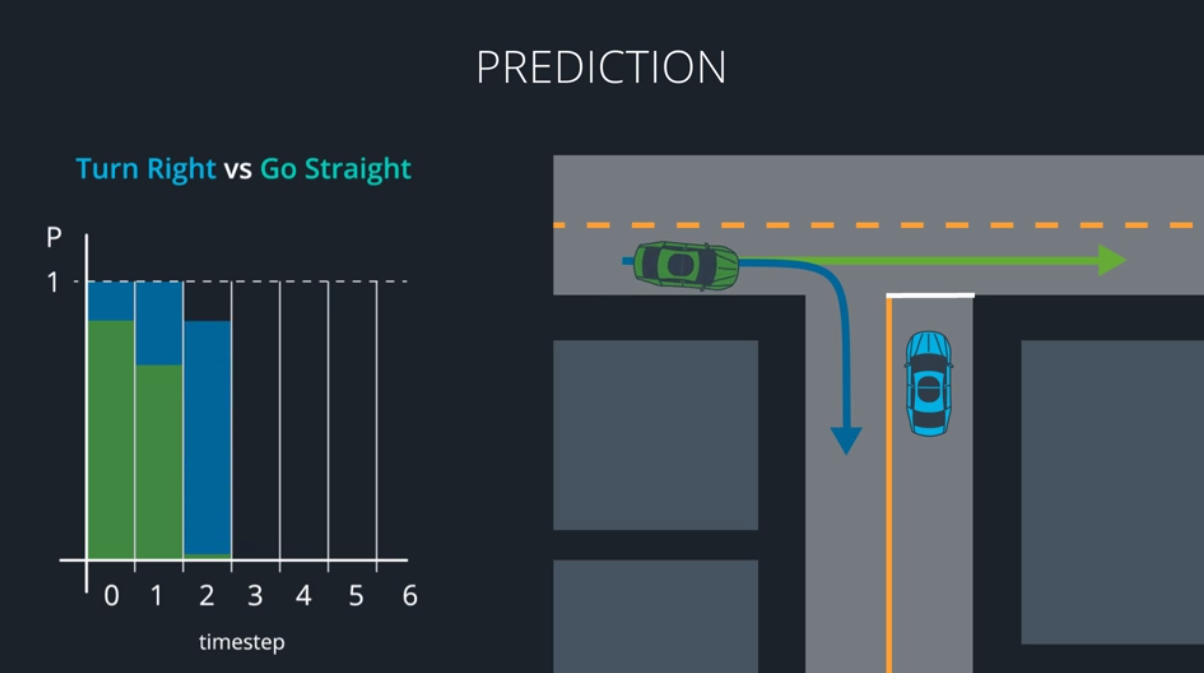
**Month 1: Path Planning**



Path planning is the brains of a self-driving car. It’s how a vehicle decides how to get where it’s going, both at the macro and micro levels. You’ll learn about three core components of path planning: environmental prediction, behavioral planning, and trajectory generation.

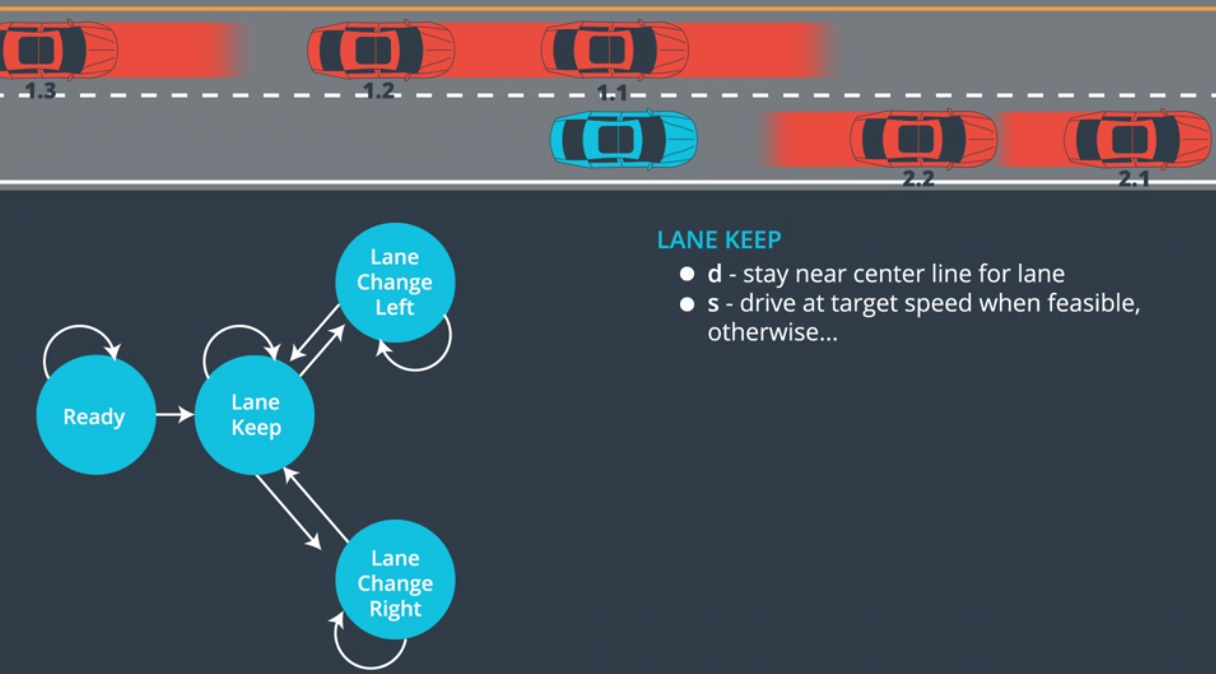
Best of all, this module is taught by our partners at [Mercedes-Benz Research & Development North America](http://mbrdna.com/). Their participation ensures that the module focuses specifically on material job candidates in this field need to know.

**Path Planning Lesson 1: Environmental Prediction**



In the Prediction Lesson, you’ll use model-based, data-driven, and hybrid approaches to predict what other vehicles around you will do next. Model-based approaches decide which of several distinct maneuvers a vehicle might be undertaking. Data-driven approaches use training data to map a vehicle’s behavior to what we’ve seen other vehicles do in the past. Hybrid approaches combine models and data to predict where other vehicles will go next. All of this is crucial for making our own decisions about how to move.

**Path Planning Lesson 2: Behavior Planning**



At each step in time, the path planner must choose a maneuver to perform. In the Behavior Lesson, you’ll build finite-state machines to represent all of the different possible maneuvers your vehicle could choose. Your FSMs might include *accelerate*, *decelerate*, *shift left*, *shift right*, and *continue straight*. You’ll then construct a cost function that assigns a cost to each maneuver, and chooses the lowest-cost option.

**Path Planning Lesson 3: Trajectory Generation**



Trajectory Generation is taught by Emmanuel Boidot, from Mercedes-Benz’s Vehicle Intelligence team.

In the Trajectory Lesson, you’ll use C++ and the [Eigen](http://eigen.tuxfamily.org/index.php?title=Main_Page) linear algebra library to build candidate trajectories for the vehicle to follow. Some of these trajectories might be unsafe, others might simply be uncomfortable. Your cost function will guide you to the best available trajectory for the vehicle to execute.

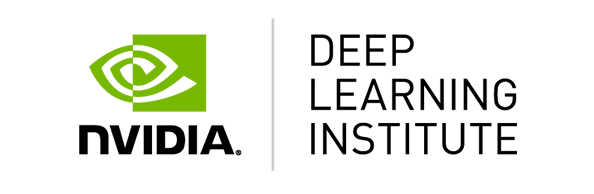
Path Planning Project: Highway Path Planner

Using the newest release of the [Udacity simulator](https://github.com/udacity/self-driving-car-sim), you’ll build your very own path planner and put it to the test on the highway. Tie together your prediction, behavior, and trajectory engines from the previous lessons to create an end-to-end path planner that drives the car in traffic!

**Month 2: Electives**

Term 3 will launch with two electives: Advanced Deep Learning, and Functional Safety. We’ve selected these based on feedback from our hiring partners, and we’re very excited to give students the opportunity to gain deep knowledge in these topics.

Month 2 Elective: Advanced Deep Learning



Udacity has partnered with the [NVIDIA Deep Learning Institute](https://www.nvidia.com/en-us/deep-learning-ai/education/)to build an advanced course on deep learning.

This module covers semantic segmentation, and inference optimization. Both of these topics are active areas of deep learning research.

Semantic segmentation identifies free space on the road at pixel-level granularity, which improves decision-making ability. Inference optimizations accelerate the speed at which neural networks can run, which is crucial for computational-intense models like the semantic segmentation networks you’ll study in this module.

**Advanced Deep Learning Lesson 1: Fully Convolutional Networks**



In this lesson, you’ll build and train fully convolutional networks that output an entire image, instead of just a classification. You’ll implement three special techniques that FCNs use: 1x1 convolutions, upsampling, and skip layers, to train your own FCN models.

**Advanced Deep Learning Lesson 2: Scene Understanding**



In this lesson, you’ll learn the strengths and weaknesses of bounding box networks, like YOLO and Single Shot Detectors. Then you’ll go a step beyond bounding box networks and build your own semantic segmentation networks. You’ll start with canonical models like VGG and ResNet. After removing their final, fully-connected layers, you can add the three special techniques you’ve already practiced: 1x1 convolutions, upsampling, and skip layers. Your result will be an FCN that classifies each road pixel in the image!

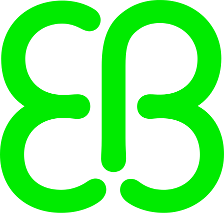
**Advanced Deep Learning Lesson 3: Inference Optimizations**

One of the challenges of semantic segmentation is that it requires a lot of computational power. In this lesson, you’ll learn how to accelerate network performance in production, using techniques such as fusion, quantization, and reduced precision.

**Advanced Deep Learning Project: Semantic Segmentation**

In the project at the end of the Advanced Deep Learning Module, you’ll build a semantic segmentation network to identify free space on the road. You’ll apply your knowledge of fully convolutional networks and their special techniques to create a semantic segmentation model that classifies each pixel of free space on the road. You’ll accelerate the network’s performance using inference optimizations like fusion, quantization, and reduced precision. You’ll be studying and implementing approaches used by top performers in the [KITTI Road Detection Competition!](http://www.cvlibs.net/datasets/kitti/eval_road.php)

Month 2 Elective: Functional Safety



Together with [Elektrobit](https://www.elektrobit.com/), we’ve built a fun and comprehensive Functional Safety Module.

You’ll learn functional safety frameworks to ensure that vehicles is safe, both at the system and component levels.

**Functional Safety Lesson 1: Introduction**



You’ll build a functional safety case with Dheeraj, Stephanie, and Benjamin from Elektrobit.

In this lesson, Elektrobit’s experts will guide you through the high-level steps that the [ISO 26262](https://www.iso.org/standard/43464.html) standard requires for building a functional safety case. ISO 26262 is the world-recognized standard for automotive functional safety. Understanding the requirements of this standard gets you started on mastering a crucial field of autonomous vehicle development.

**Functional Safety Lesson 2: Safety Plan**

In this lesson, you’ll build a safety plan for a lane-keeping assistance feature. You’ll start with the same template that Elektrobit functional safety managers use, and add the information specific to your feature.

Functional Safety Lesson 3: Hazard Analysis and Risk Assessment

You’ll complete a hazard analysis and risk assessment for the lane-keeping assistance feature. As part of the HARA, you’ll brainstorm how the system might fail, including the operational mode, environmental details, and item usage of each hypothetical scenario. Your HARA will record the issues to monitor in your functional safety analysis.

**Functional Safety Lesson 4: Functional Safety Concept**

For each issue identified in the HARA, you’ll develop a functional safety concept that describes high-level performance requirements.

**Functional Safety Lesson 5: Technical Safety Concept**

You’ll translate high-level functional safety concept requirements into technical safety concept requirements that dictate specific performance parameters. At this point you’ll have concrete constraints for the system.

**Functional Safety Lesson 6: Software and Hardware**

Functional safety includes specific rules on how to implement hardware and software. In this lesson, you’ll learn about spatial, temporal, and communication interference, and how to guard against them. You’ll also review MISRA C++, the most common set of rules for writing C++ for automotive systems.

**Functional Safety Project: Safety Case**

You’ll use the guidance from your lessons to construct an end-to-end safety case for a lane departure warning feature. You’ll begin with the hazard analysis and risk assessment, and create further documentation for functional and technical safety concepts, and finally software and hardware requirements. Analyzing and documenting system safety is critical for autonomous vehicle development. These are skills that often only experienced automotive engineers possess!

**System Integration**

System integration is the final module of the Nanodegree program, and it’s the month where you actually get to put your code on the Udacity Self-Driving Car!

You’ll learn about the software stack that runs on “Carla,” our self-driving vehicle. Over the course of the final month of the program, you will work in teams to integrate software components, and get the car to drive itself around the Udacity test track.

**Vehicle Subsystems**

This lesson walks you through Carla’s key subsystems: sensors, perception, planning, and control. Eventually you’ll need to integrate software modules with these systems so that Carla can navigate the test track.

**ROS and Autoware**

Carla runs on two popular open-source automotive libraries: ROS and Autoware. In this lesson you’ll practice implementing ROS nodes and Autoware modules.

**System Integration**

During the final lesson of the program, you’ll integrate ROS nodes and Autoware modules with Carla’s software development environment. You’ll also learn how to transfer the code to the vehicle, and resolve issues that arise on real hardware, such as latency, dropped messages, and process crashing.

**Project: Carla**

This is the capstone project of the Nanodegree program! You will work with a team of students to integrate the skills you’ve developed over the last nine months. The goal is to build Carla’s software environment to successfully navigate Udacity’s test track.

When you complete Term 3, you will graduate from the program, and earn your Udacity Self-Driving Car Engineer Nanodegree credential. You will be ready to work on an autonomous vehicle team developing groundbreaking self-driving technology, and you will join a rarefied community of professionals who are committed to a world made better through this transformational technology.

See you in class!

#### 

以上这些内容将随着我们不断更新授课大纲而更新因为一门伟大的课程在于不断汲取反馈和加以改进

如果你成功加入了此课程恭喜你我们很高兴能成为你的讲师。

如果我们建议你在加入课程之前先温习几个课题并进行自我评估请务必照做我们很高兴为你传授知识并希望你有一个愉快的学习体验。

如果你尚未申请此课程[立即行动吧](https://cn.udacity.com/drive)期待在课堂上看到你