Intro to Self-Driving Cars Syllabus

Contact Info

While going through the program, if you have questions about anything, you can reach us at . For help from Udacity Mentors and your peers visit the Udacity Classroom.

Nanodegree Program Info

Version: 1.0.0

Length of Program: 70 Days*

** This is a self-paced program and the length is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. Actual hours may vary.*

Part 1: Orientation

Welcome to the Intro to Self-Driving Cars Nanodegree program! In this section you'll get a sneak peak of the classroom, meet the team, and learn about the services provided. Then you'll take a readiness assessment and check out some learning resources to help you make the most out of your experience.

Part 2: Bayesian Thinking

Learn the framework that underlies a self-driving car's understanding of itself and the world around it, and to see the world the way a self-driving car does.

Project: Joy Ride

A quick introduction to controlling a (simulated) car with code. Parts 1 and 2 will show you how to control gas and steering and in part 3 you'll program a car to parallel park.

Supporting Lessons

Lesson	Summary
Introduction	A brief introduction to Bayesian Thinking from Sebastian.

Part 3: Working with Matrices

This course will focus on two tools which are vital to self-driving car engineers: object oriented programming and linear algebra.

Project:

An introduction to the amazing tools and algorithms you'll learn in this lesson.

Part 4: C++ Basics

This course is the first step in a rewarding journey towards C++ expertise. The goal is translation: get a program written in Python, and translate it into C++.

Project: Translate Python to C++

Apply your knowledge of C++ syntax by translating the Histogram Filter code from the first course into C++.

Supporting Lessons

Lesson	Summary
C++ Getting Started	The differences between C++ and Python and how to write C++ code.
C++ Vectors	To program matrix algebra operations and translate your Python code, you will need to use C++ Vectors. These vectors are similar to Python lists, but the syntax can be somewhat tricky.
Practical C++	Learn how to write C++ code on your own computer and compile it into a executable program without running into too many compilation errors.
C++ Object Oriented Programming	Learn the syntax of C++ object oriented programming as well as some of the additional OOP features provided by the language.
Python and C++ Speed	In this lesson, we'll compare the execution times of C++ and Python programs.

Part 5: Performance Programming in C++

Explore how to write good code that runs correctly. We'll focus primarily on low level features of C++, but we'll discuss other best practices as well.

Part 6: Navigating Data Structures

Algorithmic thinking is a skill you'll refine throughout your career. In this course you'll focus on frequently used data structures and algorithms.

Project: Implement Route Planner

In this lesson you will actually implement a Google-maps style routing algorithm using A star search.

Supporting Lessons

Lesson	Summary
How to Solve Problems	In this lesson you'll solve a hard problem with the help of Dave Evans and you'll learn a systematic approach to solving hard computer programming problems as you do.
Data Structures	The list isn't the only structure for storing data! In this lesson you'll learn about sets, dictionaries and other Python data structures.
The Search Problem	When programming a car to drive itself you run into problems. Many of these are "search" problems. In this lesson you'll learn what search problems are and several algorithms for solving them.

Part 7: Vehicle Motion and Control

This course is a crash course in two branches of mathematics which are crucial to self driving cars: calculus and trigonometry. You will learn how a self driving car uses various motion sensors to help it understand its own motion. At the end of this course you will use raw sensor data (which give information about distance driven, acceleration, and rotation rates) to reconstruct a vehicle's trajectory through space.

Part 8: Computer Vision and Machine Learning

In this course you'll learn how a computer sees an image, and how we can use machine learning to teach a computer to identify images programmatically.

Project: Traffic Light Classifier

Build a classification pipeline that takes in an image of a traffic and outputs a label that classifies the image as a: red, green, or yellow traffic light.

Supporting Lessons

Lesson	Summary
Computer Vision and Classification	Students will learn how to program an image classifier using computer vision techniques. Along the way you'll learn about machine learning, color transformation, feature extraction, and more!

Part 9: Graduation!

Congratulations! You're ready to graduate. Learn how you can continue your Udacity journey by enrolling in a Career-Ready Nanodegree Program

Udacity

Generated Mon May 6 21:33:22 PDT 2019