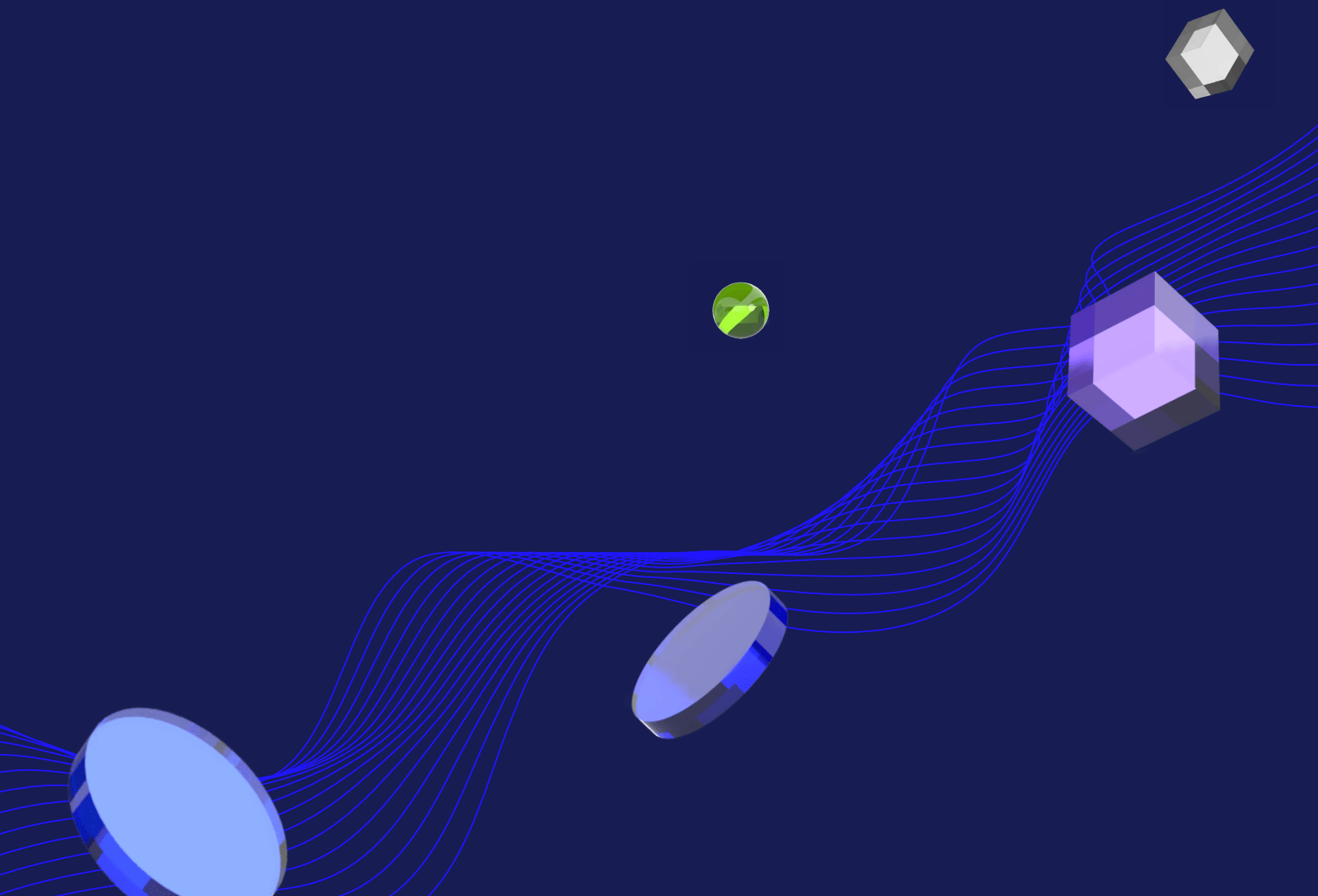




SCHOOL OF AUTONOMOUS SYSTEMS

Robotics Software Engineer

Nanodegree Program Syllabus



Overview

This program was built in collaboration with robotics engineers to ensure you learn the skills necessary for success in the field. Learn probabilistic robotics algorithms such as localization, mapping, simultaneous localization and mapping (SLAM), path planning, and navigation, and implement them using C++, the Robot Operating System (ROS), and Gazebo. Build five interactive projects using C++ and ROS, and use them to demonstrate probabilistic robotics algorithms skills.

Program information



Estimated Time

4 months at 10hrs/week*



Skill Level

Advanced



Prerequisites

Learners must have knowledge of:

- Any object-oriented programming language, preferably C++
- Intermediate probability
- Intermediate calculus
- Intermediate linear algebra
- Basic Linux command lines



Required Hardware/Software

Learners need access to Robot Operating System (ROS) and Gazebo. They will code primarily with C++. These platforms and languages are freely available. We will provide learners with a GPU-enabled Linux Workspace that runs in the browser, but internet connection is required. Optionally, learners can install the Linux image on a Virtual Machine on their browser.

*The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 5-10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

Gazebo World

Learn how to simulate a robotic environment with Gazebo, the most common simulation engine used by roboticists around the world.



Course Project

Build My World

Learners will use their new techniques to build their first environment. Key skills demonstrated include:

- Launching a Gazebo environment
- Designing in Gazebo

Lesson 1

Introduction to Gazebo

- Work with the Gazebo simulator to build new environments and deploy assets.

ROS Essentials

Discover how ROS provides a flexible and unified software environment for developing robots in a modular and reusable manner. Learn how to manage existing ROS packages within a project and how to write ROS Nodes in C++.



Course Project

Go Chase It!

Demonstrate proficiency with ROS, C++, and Gazebo by building a ball-chasing robot. Learners will first design a robot inside Gazebo, house it in the world they have built in the Build My World project, and code a C++ node in ROS to chase yellow balls. Key skills demonstrated include:

- Building Catkin Workspaces
- ROS node creation
- ROS node communication
- Using additional ROS packages
- Gazebo world integration
- Additional C++ practice
- RViz Integration

Lesson 1

Introduction to ROS

- Obtain an architectural overview of the Robot Operating System Framework.

Lesson 2

Packages & Catkin Workspaces

- Learn the ROS workspace structure, essential command line utilities, and how to manage software packages within a project.

Lesson 3

Write ROS Nodes

- Write ROS nodes in C++.

Course 3

Localization

Learn how Gaussian filters can be used to estimate noisy sensor readings and how to estimate a robot's position relative to a known map of the environment with Monte Carlo Localization (MCL).



Course Project

Where Am I?

Learners will interface their own mobile robot with the Adaptive Monte Carlo Localization algorithm in ROS to estimate their robot's position as it travels through a predefined set of waypoints. They'll also tune different parameters to increase the localization efficiency of the robot. Key skills demonstrated include:

- Implementation of Adaptive Monte Carlo Localization in ROS
- Understanding of tuning parameters required

Lesson 1

Introduction to Localization

- Learn what it means to localize and the challenges behind it.

Lesson 2

Kalman Filters

- Learn the Kalman Filter and its importance in estimating noisy data.

Lesson 3

Lab: Kalman Filters

- Implement an Extended Kalman Filter package with ROS to estimate the position of a robot.

Lesson 4

Monte Carlo Localization

- Learn the MCL (Monte Carlo Localization) algorithm to localize robots.

Lesson 5

Build MCL in C++

- Code the MCL algorithm in C++.

Course 4

Mapping and SLAM

Learn how to create a Simultaneous Localization and Mapping (SLAM) implementation with ROS packages and C++. Learners will achieve this by combining mapping algorithms with what they learned in the localization lessons.



Course Project

Map My World

Learners will interface their robot with an RTAB Map ROS package to localize it and build 2D and 3D maps of their environment. They must put all the pieces together properly to launch the robot and then teleop it to map its environment. Key skills demonstrated include:

- SLAM implementation with ROS/Gazebo
- ROS debugging tools (rqt and roswtf)

Lesson 1

Introduction to Mapping & SLAM

- Learn the mapping and SLAM concepts, as well as the algorithms.

Lesson 2

Occupancy Grid Mapping

- Map an environment by coding the Occupancy Grid Mapping algorithm with C++.

Lesson 3

Grid-Based FastSLAM

- Simultaneously map an environment and localize a robot relative to the map with the grid-based FastSLAM algorithm.
- Interface a turtlebot with a grid-based FastSLAM package with ROS to map an environment.

Lesson 4

GraphSLAM

- Simultaneously map an environment and localize a robot relative to the map with the GraphSLAM algorithm.

Path Planning and Navigation

Learn different path planning and navigation algorithms. Then, combine SLAM and navigation into a home service robot that can autonomously transport objects in your home.



Course Project

Home Service Robot

In this capstone project, learners will use a SLAM package to autonomously map an environment. Then, they will interface your robot with a path planning and navigation ROS package to move objects within an environment. Key skills demonstrated include:

- Advanced ROS and Gazebo integration
- ROS Navigation stack 7
- Path planning

Lesson 1

Intro to Path Planning & Navigation

- Learn what the lessons in path planning and navigation will cover.

Lesson 2

Classic Path Planning

- Learn a number of classic path planning approaches that can be applied to low-dimensional robotic systems.

Lesson 3

Lab: Path Planning

- Code the BFS and A* algorithms in C++.

Lesson 4

Sample-Based & Probabilistic Path Planning

- Learn about sample-based and probabilistic path planning, and how they can improve on the classic approach.

Course 6

KUKA Path Planning (Optional)



Course Project

KUKA Path Planning

Learners will apply what they have learned about ROS and path planning to search for a path and navigate a KUKA robot through a 2D maze. Key skills demonstrated include:

- Path planning
- Using C++ and Python with external ROS API

Lesson 1

Project Introduction

- Learn the requirements of the project.
-

Lesson 2

Project Details

- Learn the project specifications and how to get started.

Meet your instructors.



Sebastian Thrun

Founder & Executive Chairman at Udacity

As the founder and president of Udacity, Sebastian's mission is to democratize education. He is also the founder of Google X, where he led projects including the Self-Driving Car, Google Glass, and more.



David Silver

Curriculum Lead at Udacity

David Silver leads the Udacity Curriculum Team. Before Udacity, David was a research engineer on the autonomous vehicle team at Ford. He has an MBA from Stanford and a BSE in Computer Science from Princeton.



Karim Chamaa

Course Developer

Karim started his early career as a mechanical engineer. He earned his MS in mechatronics and robotics engineering from NYU. His specialties include kinematics, control, and electronics.

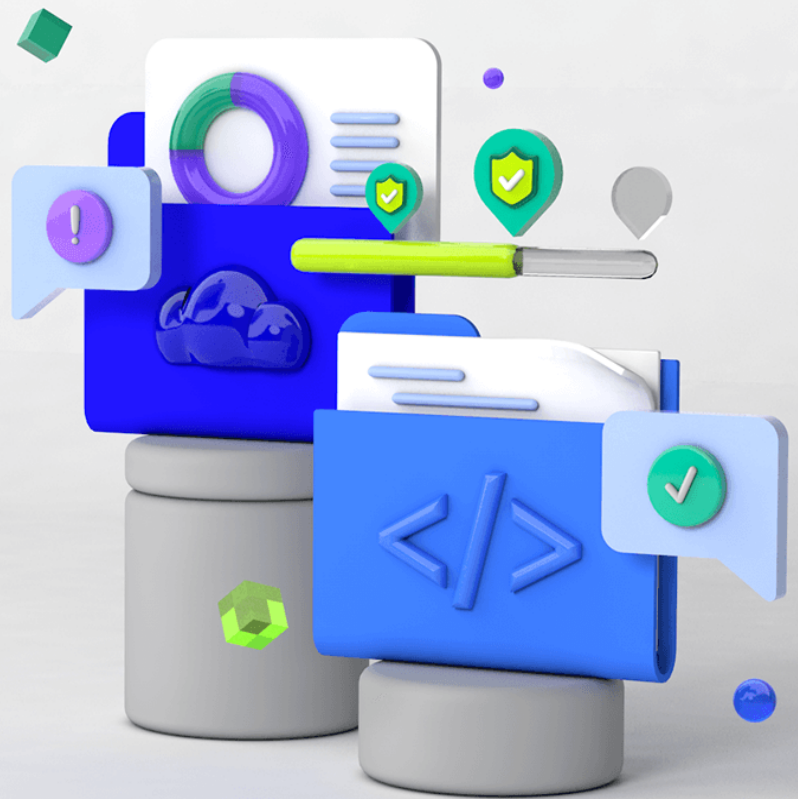


Julia Chernushevich

Course Developer

Julia is an experienced educator and robotics specialist. Her previous work experiences include teaching mechatronics engineering at the University of Waterloo and designing electric vehicles for underground mines.

Udacity's learning experience



Hands-on Projects

Open-ended, experiential projects are designed to reflect actual workplace challenges. They aren't just multiple choice questions or step-by-step guides, but instead require critical thinking.



Knowledge

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover how to solve the challenges that you encounter.



Workspaces

See your code in action. Check the output and quality of your code by running it on interactive workspaces that are integrated into the platform.



Quizzes

Auto-graded quizzes strengthen comprehension. Learners can return to lessons at any time during the course to refresh concepts.



Custom Study Plans

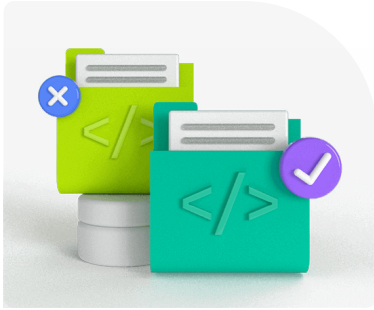
Create a personalized study plan that fits your individual needs. Utilize this plan to keep track of movement toward your overall goal.



Progress Tracker

Take advantage of milestone reminders to stay on schedule and complete your program.

Our proven approach for building job-ready digital skills.



Experienced Project Reviewers

Verify skills mastery.

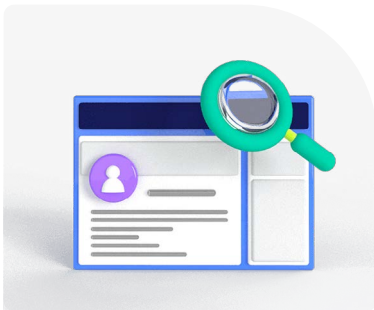
- Personalized project feedback and critique includes line-by-line code review from skilled practitioners with an average turnaround time of 1.1 hours.
- Project review cycle creates a feedback loop with multiple opportunities for improvement—until the concept is mastered.
- Project reviewers leverage industry best practices and provide pro tips.



Technical Mentor Support

24/7 support unblocks learning.

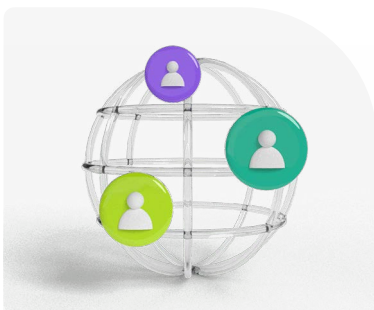
- Learning accelerates as skilled mentors identify areas of achievement and potential for growth.
- Unlimited access to mentors means help arrives when it's needed most.
- 2 hr or less average question response time assures that skills development stays on track.



Personal Career Services

Empower job-readiness.

- Access to a Github portfolio review that can give you an edge by highlighting your strengths, and demonstrating your value to employers.*
- Get help optimizing your LinkedIn and establishing your personal brand so your profile ranks higher in searches by recruiters and hiring managers.



Mentor Network

Highly vetted for effectiveness.

- Mentors must complete a 5-step hiring process to join Udacity's selective network.
- After passing an objective and situational assessment, mentors must demonstrate communication and behavioral fit for a mentorship role.
- Mentors work across more than 30 different industries and often complete a Nanodegree program themselves.

*Applies to select Nanodegree programs only.

Learn more at

www.udacity.com/online-learning-for-individuals →