Guide for instructors

Maintainer: Jacopo Tani

Part A
Overview

This section's aim is to provide a step-by-step guide to get started building a new class with Duckietown.

We hope that everything is perfect and easy but if you find yourself needing help please use the links at the bottom of each of the pages to ask questions or look on the online education forums.

This guide is primarily targeted at *graduate level* class preparation. For undergraduate and lower-level classes please stay tuned.

UNIT A-1

How to Use This Guide

There are *two* distinct possible users of this class guide:

- Instructors that want to use Duckietown to build a course on autonomy
- Instructors that want to use Duckietown as part of a class about some aspect of autonomy (e.g. computer vision, control, autonomous vehicles, etc)

1.1. Instructors Building a class on Autonomy

1.2. Instructors Using a Subset of Duckietown within an existing class

UNIT A-2
The Serial Parallel Model

In the “serial-parallel” model, the class is divided into two distinct sections as shown in Figure 2.4.

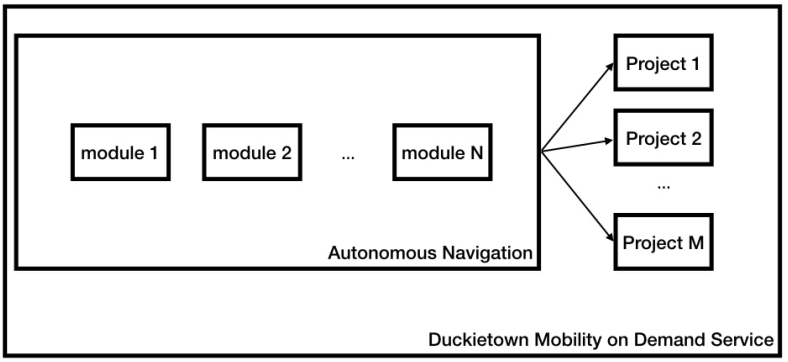

Figure 2.4

Figure 2.2. "Serial-parallel class structure

In the first part is a basic primer on autonomy and contains a high breadth low depth introduction to all topics relevant to autonomy. This includes: autonomy architectures, software architectures, actuation, signal processing, kinematics and dynamics, computer vision and sensing, filtering and estimation, mission and motion planning, control. During this first part, students incrementally build and demonstrate capabilities on the Duckiebot synchronously with the class. Exercises are used to build up the students’ basic skills of programming, computer vision, linear algebra, and others.

The second part of the class is much more free form. Normally, the class projects will be announced at the end of the first part and students will be primarily focused on completing their projects. Lectures become more free-form and are more advanced topics, such as: high-level perception (semantic segmentation, object detection, SLAM, ...), fleet level planning and mobility on demand, advanced control algorithms.

A sample curriculum is given in the table below. Each entry in the table corresponds to a unit. Units may comprise different types of resources. For more details on the available resources for a unit please refer to Unit A-3 - Pedagogical Resources. Generally, for each unit, materials are presented in the form of traditional lecture slides and then supported by interactive python notebook instructionals. Lecture videos and theory material contained in the book is left as a reference for students. Concepts are then reinforced through the exercises and robot demonstrations.

Duckietown Class Schedule : For public

UNIT A-3

Pedagogical Resources

A suite of pedagogical resources are provided that allow the an instructor to select how to present the material through different mechanisms in accordance with current educational principles

3.1. Background Materials

The background materials can be made available to students prior to the start of the class as and can be used to learn theory and build skills that will be useful in the class.

1) Theory Preliminaries

Theoritical background information is contained in the - Preliminaries.

2) Software Reference

Software reference material is contained in the Software reference.

3.2. Lecture Material

There are several different types of resources available at - Learning materials for use in classes and as learning aides for students.

In general, the material is divided into “Units”, where each unit is supported by some or all of the following: slides, written theory, python notebooks, exercises, and robot demonstrations.

1) Slides

Slides are written directly in Markdown in this book. For example ðŸŽ¦ Autonomous Vehicles. For information on how to create your own slides in this format see ðŸŽ¦ Making slides.

Generally, which each slide presentation there is accompanying theory in the book in the same section.

2) Lecture Videos

Video lectures will be posted for each unit. See here for more details soon.

3) Python Notebooks

Many units are also supported by notebook. One option is to present the lecture material and then run through the python notebook to reinforce the theory presented with practical examples. The notebooks use real data logs, either from Duckietown or other sources to demonstrate the concepts presented in abstract form in the slides in a more concrete manner.

3.3. Exercises

Exercises are provided to help students master the material. They can optionally be used as graded homework assignments. Solutions are provided in a separate private repository that you should have access to if you are registered instructor of the course (follow the instructions on here.

3.4. Reproducible Robot Demonstrations

The backbone of the class and the entire project is the Duckiebot platform and the Duckietown environment. It is **required** that you build a Duckietown to support your class. Space can be an issue, but this is really the best part.

The instructions for building the Duckiebot are provided in - Duckiebot operation manual and the instructions for building a conforming Duckietown are given in (unknown ref opmanual_duckietowns/duckietowns)

warning (1 of 1) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link
'#opmanual_duckietowns/duckietowns'.
```

Location not known more precisely.

Created by function `check_if_any_href_is_invalid` in module `mcdp_docs.check_missing_links`.

. The - Duckiebot operation manual contains all the necessary information for assembly, calibration, troubleshooting, and running of the demos. The demos include basic Unit B-4 - Lane following, Unit B-5 - Indefinite Navigation (the Duckiebot reads the road signs and takes a random feasible action), and multi-robot operations such as Unit B-8 - Explicit Coordination, and many others (the list continually grows).

UNIT A-4

Contributing New Materials

If you build new class materials during the course of your class, please consider contributing to the project. Please see - Duckumentation documentation for instructions on how to contribute back your material.