

# Duckietown: an Innovative Way to Teach Autonomy

Jacopo Tani, Liam Paull, Maria Zuber, Daniela Rus,  
Jonathan How, John Leonard, and Andrea Censi.

Massachusetts Institute of Technology, Cambridge, 02139, MA, USA,  
jtani@mit.edu,  
Project website: <http://duckietown.mit.edu>

**Abstract.** Teaching robotics is challenging because it is a multidisciplinary, rapidly evolving and experimental discipline that integrates cutting-edge hardware and software. This paper describes the course design and first implementation of *Duckietown*, a vehicle autonomy class that experiments with teaching innovations in addition to leveraging modern educational theory for improving student learning. We provide a robot to every student, thanks to a minimalist platform design, to maximize active learning; and introduce a role-play aspect to increase team spirit, by modeling the entire class as a fictional start-up (Duckietown Engineering Co.). The course formulation leverages backward design by formalizing intended learning outcomes (ILOs) enabling students to appreciate the challenges of: (a) heterogeneous disciplines converging in the design of a minimal self-driving car, (b) integrating subsystems to create complex system behaviors, and (c) allocating constrained computational resources. Students learn how to assemble, program, test and operate a self-driving car (*Duckiebot*) in a model urban environment (*Duckietown*), as well as how to implement and document new features in the system. Traditional course assessment tools are complemented by a full scale demonstration to the general public. The “duckie” theme was chosen to give a gender-neutral, friendly identity to the robots so as to improve student involvement and outreach possibilities. All of the teaching materials and code is released online in the hope that other institutions will adopt the platform and continue to evolve and improve it, so to keep pace with the fast evolution of the field.

**Keywords:** Duckietown, Autonomous Vehicles, Educational Robotics, Active Learning, Constructive Alignment, Backwards Design

## 1 Introduction

*Autonomous cars* are a quickly maturing robotics application being researched and developed by all major automotive companies. The need for autonomy engineers drives academia to develop training methods that effectively meet the teaching challenges of this developing field.

Teaching autonomy is challenging due to factors involving *what* is taught, *how* it is taught and with what *resources*. Robotics is an interdisciplinary field: a student approaching the design, construction and operation of a robot will require theoretical and practical knowledge of mathematics, physics, controls, computer vision, as well as mechanical, software and electrical engineering. Additional challenges arise from the intimate blending of hardware and software required to make a robot work. Sensors and actuators are constantly improving and algorithmic approaches to solving specific tasks, such as path planning, localization, image processing, and many others are active areas of research and are continually evolving.

The strong experimental component of robotics invalidates the traditional “frontal” approach to teaching where students are passively fed notions. Modern educational theory has identified alternative strategies that increase the students’ long-term knowledge retention and transfer to practical problems [1–4]. *Backwards design* [5], *constructive alignment* [6] and *active learning* [1], are focused on student learning rather than the instructor teaching. Learning is more effective when students actively engage in activities designed to reach the intended learning outcomes (ILOs) [1, 2]. In particular, generation Y (or “Millennials”) do not respond well to classical teaching methods [7]. Team-oriented and more informal approaches to learning achieve better outcomes, and students find new teaching methods that incorporate multimedia and social networking more appealing [8].

In this paper we describe *Duckietown*, our approach to address the new educational demands of autonomy. *Duckietown* is an engaging and affordable robotics teaching, outreach and research platform. It was taught for the first time during Spring 2016 at the Massachusetts Institute of Technology. We first introduce the *Duckietown* platform, comprising of *Duckiebots* and *Duckietowns* (Sec. 2), highlighting their modularity and minimality. We then detail the course design (Sec. 3) with attention to the *intended learning outcomes*, *teaching and learning activities* and *assessment tools*. We then describe an innovative way of creating a broader team spirit among instructors and students by introducing a role-playing aspect to the course (Sec. 4). Finally we describe the results of the first implementation of *Duckietown* as a vehicle autonomy class (Sec. 5), followed by concluding remarks (Sec. 6).

We have released all teaching and other materials as open-source on the website [duckietown.mit.edu](http://duckietown.mit.edu), in the hope that in the future this could become a standardized platform for high school, undergraduate, and graduate level robotics education.

## 2 The Duckietown Platform

The *Duckietown* platform [9] comprises self-driving vehicles (*Duckiebots*) and model urban environments (*Duckietowns*).

The **Duckiebot** is a minimal platform consisting of mostly off-the-shelf components. *Computation* is performed on-board on a Raspberry PI 2 (RPi2), strong of 4 900 MHz ARM cores and 1GB of RAM. *Actuation* is provided through two DC motor controlled wheels, included with the frame in the chassis kit. Wheel odometry is not used for estimation; therefore, the chassis is the most fungible part of the



**Fig. 1.** *Duckiebots* are minimal autonomy platforms designed to operate in *Duckietowns* of arbitrary topology.

robot and can be replaced by any setup that uses two DC motors in differential drive configuration. The Adafruit DC motor board is designed to attach directly to the RPi2. *Sensing* is provided exclusively through a high definition camera. The RPi2 supports a custom-designed camera via a dedicated parallel connection. For setup, debug, and optional manual control, *communication* to the Duckiebot is managed through WiFi. For large deployments a solution to network saturation is an onboard access point on each Duckiebot. These mobile hotspots create a 5GHz network, are powered independently, and connect to the RPi2 through ethernet. Moreover, a wireless joystick can be used for making manual control more convenient. Duckiebots can also be equipped with RGB LEDs to enable global behaviors that require inter-vehicle communication.

**Duckietowns** are structured urban environments in which Duckiebots are designed to operate. A Duckietown floor is made by *roads*, and above the floor are *signals* (i.e., *signs* and *traffic lights*). These layers enable:

- *modularity* that allows for construction of road topologies of variable complexity;
- stringent *appearance guidelines* that ensure consistency;
- *low-cost* components for universal adoption;
- *scalability* of functionality that provides teaching and research pathways;
- opportunities for *customization* that enhance student ownership.

*Floor Layer - Roads:* The road level of a Duckietown is built using interlocking foam tiles. Three tile types exist: (i) straight roads; (ii) 90° turns and (iii) intersections. Intersections must be abutted on all sides by straight roads, and may be (iii-a) T junctions (3 way) or (iii-b) crossroads (4 way).

*Signal Layer - Signs:* are present in three types: (i) traffic signs, (ii) street names, and (iii) localization signs. All three sign types utilize AprilTags [10] to convey information to the Duckiebots. The use of AprilTags in unison to images/text allows for a progression of functionality and programming skill sets. Such scaling in difficulty and functionality are a cornerstone of Duckietown as an educational and research tool. The appearance and location of signs is formalized [9] to ensure recognition by Duckiebots.

*Signal Layer - Traffic Lights:* Traffic lights are designed as a “car without a chassis” and therefore utilize the same hardware and software as the Duckiebot. They produce signals (i.e., stop and pass) encoded in the blinking frequencies of the LEDs, which are

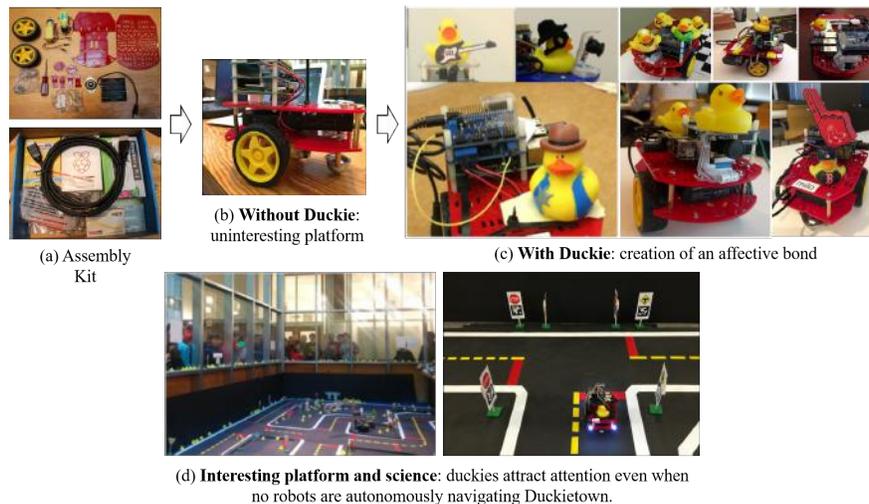
more robustly detected than color-based coding. This architecture provides the opportunity to network the traffic lights and scale the problem to include investigation of real time traffic monitoring, smart infrastructure, and traffic flow optimization.

If the rules for city construction [9] are followed, a Duckiebot will navigate Duckietowns of arbitrary topology by a sequence of: (a) navigating one or more straight tiles until a red strip appears; (b) waiting for a coordination signal; (c) executing an open-loop motion; and (d) re-localizing on a straight tile.

## 2.1 Why Duckies? A friendlier image for autonomous vehicles

*Duckies* are entirely non-functional, purely decorative, yet essential. Robots are typically thought of as dangerous, strong, fast, aggressive and unpredictable. We designed Duckiebots instead to be safe, weak and slow. Moreover, through the use of the *duckie* theme, they are perceived as curious, friendly and fun. The Duckietown in our laboratory attracts plenty of attention, even when there are no robots (Fig. 2d).

We also designed Duckiebots so not to suffer from preconceived gender stereotypes by using a *gender-neutral approach* in the choice of colors, symbols and themes. Humans are very sensitive to gender cues, even in inanimate machines like robots. Previous work has explored the role of perceived robot “gender” during human-robot interaction [11]. For example, robots are considered “male” if their function is dangerous. “Male” robots are less appealing to prospective female roboticists [12].



**Fig. 2.** Duckies make the difference between a boring, cheap platform (b) and an object with which the students can form an emotional bond (c).

## 3 Course Design

In designing this course we adhere to the practice of *backwards design*, where the final educational objectives are considered first and condensed in a list of intended learning outcomes (ILO) [13]. The teaching and learning activities (TLAs) are then crafted to align with the ILOs but also be actively engaging for the students. Active learning eases

understanding, long-term retention and transfer of knowledge from the classroom to the practical applications. Assessment tools were distributed throughout the course to allow for feedback exchange between students and staff in order to improve their convergence to the ILOs. The course has the final objective of making students:

- appreciate how *heterogenous disciplines* such as control theory, machine learning, computer vision, and artificial intelligence are integrated into a complex system;
- design and implement subsystems that work well together (problem of *integration*);
- analyze the constraints imposed by limited on board resources (e.g., computation, sensing, actuation) and operation environments, and design trade-offs to maximize performance while minimizing shared resources use (problem of *co-design*).

### 3.1 Intended Learning Outcomes

Assuming as students prerequisites: the ability to program, minimal Linux knowledge, and an undergraduate-level class in perception or control; we divide the ILOs in four categories: *operational tools*, *development methods*, *autonomy (perception, control, coordination)*, and *dissemination/documentation*.

By the end of the course, students are able to:

1. Use **operational tools** to:
  - (**OP-Construction**) Build a Duckiebot, given the hardware and a set of instructions. Includes hardware assembly;
  - (**OP-Configuration**) Configure the software and network as well as calibrate their actuators and sensors. Includes Linux and Robotic Operating System (ROS);
  - (**OP-Operation**) Demonstrate correct operation of their Duckiebot.
2. Master **system development methods**, such as:
  - (**DEV-ROS**) Develop ROS software modules and integrate them in the system;
  - (**DEV-Tools**) Utilize *standard tools* for software development (e.g., source code repositories; branching and merging);
  - (**DEV-Test**) Utilize the best practices of system development, including *test-driven* and *data-driven* development;
  - (**DEV-Group**) Familiarize with the dynamics of open-source development, including the challenges of integrating independently-developed functionalities.
3. Implement new features in a system demonstrating **autonomy** capabilities, such as:
  - (**AU-Perception**) Image processing, Bayesian filtering, localization;
  - (**AU-Control**) Navigation, control, coordination;
  - (**AU-Behaviors**) Integrate perception and control into complex behaviors.
4. Achieve effective **dissemination** by:
  - (**DIS-Explain**) Explaining design choices and trade-offs;
  - (**DIS-Document**) Documenting their work, by creating step-by-step instruction sets to enable future users to reproduce their results;
  - (**DIS-Demonstrate**) Disseminating the results of their work through a demonstration to the general public.

**Table 1.** Overview of the course.

Phase	Activities	Educational Intent	Addressed ILOs
I: Autonomy Tutorial	Course and Staff introductions	High level overview	OP-Construction, Configuration
	Baseline system development	Bring all up to speed	OP-Construction, Configuration, Operation DEV-ROS, Tools
II: Team Projects	Team-projects I: Features development	Creation of fundamental capabilities	OP-Configuration, Operation DEV-ROS, Tools, Test, Group
	Team-projects II: Behaviors development	Creation of complex interactions	DIS-Explain, Document, Demonstrate
III: Public Demonstration	Demo: Open House	Dissemination, Assessment	OP-Configuration, Operation DIS-Demonstrate
	Wrap up of demo	Documentation Formalization	DIS-Explain, Document

### 3.2 Teaching and Learning Activities

The course structure (Table 1) is designed to gradually introduce TLAs that facilitate achievement of the ILOs. TLAs are a mix of traditional *lectures* and *active learning tasks*. Lectures (Table 2) cover topics that introduce autonomy and more specific content that informs students in designing features and global behaviors of Duckiebots. Staff provide hands-on mentorship in laboratory sessions. Additional active learning tasks include *group-based projects*, *presentations* to the class and *public demonstrations*.

The course is divided in **three phases** (Table 1): *Phase I*: is the *autonomy tutorial*, where TLAs are focused on classroom lectures and laboratory sessions; *Phase II*: is mostly characterized by *team-based* and *project-based* work; in *Phase III*: finally students integrate findings of the previous phase to deliver a public demonstration.

**Phase I – Autonomy tutorial** This phase provides a tutorial in autonomy and bringing all personnel (students and staff alike) up to speed on the basic operational tools necessary to contribute to the project.

Core activities such as the setup of computers and accounts for shared resources, assembly of Duckiebots and creation of ROS modules (homework) are addressed during this phase. During this phase the following TLAs take place: (a) classroom lectures; (b) laboratory sessions; and (c) homework.

During *classroom activities* the lectures (Table 2) focus on the theoretical foundations of the practical applications of vehicle autonomy. Throughout this phase, the following topics are presented: robotics architecture (ROS) [14], image formation process [15], camera calibration [16], minimal sufficient representations for visual tasks, nonlinear filtering [17], robust localization and mapping [18], and motion planning [19].

The topics treated in class are then explored through implementation in *laboratory sessions*. Each student *and* staff member (Fig. 3) is provided with a *Duckiebox* with their own robot components and instructions on how to assemble it. *Everyone* has to be able to operate their Duckiebot and learn how to develop features on it. Students alternate between working individually and with the support of senior staff and peers to

**Table 2.** Course Lectures.

<b>Phase I: Autonomy Tutorial</b>	<b>Phase II: Team Projects</b>	<b>Addressed ILOs</b>
Image formation process, Camera calibration, Lane detection and filtering, Nonlinear Filtering	Target tracking and intersection estimation, Object detection, Imperfect data analysis, State estimation from motion blur	AU-Perception
Robust localization and mapping, Motion Planning	SLAM, Global localization, Navigation, Visual inertial navigation	AU-Control
	Autonomy architectures, Safety & correctness, Advanced safety and formal methods	AU-Behaviors
Software architecture (ROS)		OP-Construction DEV-ROS

problem-solve their way through assembling, calibrating and testing their Duckiebots. While students follow instructions prepared by the mentors, they comment and modify them when better or alternative methods are found through experimentation or individual expertise. The contributions of students to the documentation remain beyond the course for the benefit of future Duckietown users through the open source paradigm.

The *Homework* consists of reproducing, customizing and implementing software modules with the aim of preparing for future system development. The first phase ends with an autonomous navigation test, where each Duckiebot drives around a loop to pass.

**Phase II — Team projects** This phase includes two independent iterations of feature development. In each iteration, students are required to form teams of 2-5. Each team has 1-2 mentors. Mentors are typically senior students and staff.

At the beginning of each iteration, students are provided with project-oriented intended outcomes with technical specifications (e.g. the maximum allowable latency is 50ms). The specifications for different teams might be inter-functional and inter-team communication and cooperation are necessary.

Each team is then free to *self-organize* to achieve the intended result. Examples of project-driven TLAs include: (a) design meetings; (b) project-based work (individual or not: team is self-organized); (c) inter-group communication/cooperation; and (d) presentation of design document to class.

At the end of each iteration, groups document their design choices in an analysis document, write step-by-step instructions for reproducing their results, prepare a demonstration of the developed capabilities and present the project to the class.

The two group-based iterations differ in the hierarchy of the developed functions. During the first iteration, groups develop *basic features* and during the second, *global level behaviors* that involve integrating different basic features. Table 3 contains the project topics and examples of potential future development directions.

Classroom activities support this phase by focusing on more advanced material than in Phase I (Table 2). The topics introduced are: navigation [20], safety and correctness (intersections navigation) [21], advanced safety and formal methods [22], multi-vehicle coordination [23], simultaneous localization and mapping (SLAM) [24], estimation

**Table 3.** Team projects during the two iterations of Phase II and examples of future development.

<b>Teams 1st iteration: Features development</b>	<b>Teams 2nd iteration: Behaviors development</b>	<b>Examples of future development</b>
Illumination invariance	Parallel autonomy	Manipulation
LED detector, traffic lights	Object/Vehicle avoidance/following	Inter-bot wireless communication
Odometry calibration from sensor measurements	Traffic light coordination	Model-based control
Lane filtering and control	Stop sign coordination	Vehicle passing
Vehicle detector	Localization and mission planning	Smart infrastructure
AprilTags detection	Robust illumination invariance	Optimal co-design
Local object detection and avoidance		Mobility on demand
Bumpers and shells design and manufacturing	Bumpers and shells design and manufacturing	Safety guarantees

from motion blur [25], distributed systems [26], and visual inertial navigation [27].

**Phase III – Public Demonstrations** In this phase groups showcase their work during a public demonstration event. Interacting with the general public is an opportunity for students to act as teachers and appreciate the relevance of their work through the general public’s feedback. During this phase the following TLAs take place: (a) demo planning; (b) demo day; and (c) demo documentation.

On the day of the demo, students (and mentors) are in charge of integrating and deploying the previously developed functions and achieved behaviors to the general public. Every group demonstrates their own work by building an independent Duckietown, which they have to design and build to specification. After having prepared documentation for dissemination (demo-specific posters, informative brochures, flyers, ...) students operate their stations by continuously running the demonstration, often interactive, while explaining their work. After the event, students archive their material and provide step-by-step instructions for demo reproduction.

### 3.3 Assessment tools

The achievement of the ILOs is assessed mainly through *peer evaluation* and there is no formal test or examination students need to take throughout the course. This works as an incentive for teamwork and general involvement allowing for a more holistic evaluation of the student’s learning and contributions. Grades are assigned through a weighted average of problem sets (30%), participation (35%) and projects (35%). Projects grades were further specified by: team-based grading (70%) and individual grading (30%).

Homework modules in Phase I are graded by the instructors. Instructors are also responsible for providing grades for students’ involvement, productivity and quality of work during the project-based work of Phase II. Projects require a *demo*, a *presentation* to the class, and a *project report*. During the presentations students evaluate and leave feedback for each other through online anonymous forms.

At the end of the course, students and staff answer a questionnaire about the value of every other member, including the mentors, for each team they have participated in.

The questionnaire asks for open comments and a numerical evaluation of their “open source” participation. This score takes into account diverse contributions to the project, e.g.: (a) improving and (b) documenting the software provided; (c) adding functionality; (d) taking logs; (e) helping other members of the team; (f) running demos.

Interactions among students and mentors are chronicled in the open source platforms used to manage the documentation and code, facilitating objective evaluation.

#### 4 The “Duckietown Engineering” Role-play

Teamwork is fundamental for accomplishing complex tasks and robotics is no exception. While Phase II of the course (Sec. 3.2) enables cooperation in the class and labs, we implemented a role-play aspect to Duckietown to ensure that teamwork was an integral part of the Duckietown learning experience. When enrolling in the course, students



Designation	Titles	Roles	People
Executive	Board Members	Invited lectures	Principal Investigators (PIs)
	Advisors	Invited lectures	PI, Postdoctoral Researchers
Staff	Chief Technical Staff	Instructors, head of daily operations	PI, Postdoctoral Researchers
	Engineers	Mentors	Postdoctoral Researchers
	Human Resources	Accounts, general assistance	Non technical
	Media	Recording of classes and events	Non technical
	Assistants	City building, inventory management	Undergraduates
Students	International Operations	Enabling implementations abroad	Non-MIT affiliates
	Engineers in Training	Students	(Under) Graduates

**Fig. 3.** *Duckietown Engineering Co.*: a fictitious company created to provide a broader sense of team spirit. A “Duckietie” dress code was enforced for on-duty staff.

automatically join a fictitious company, *Duckietown Engineering Co.*, with the title of “autonomy engineers in training”. The *Duckietown Engineering* brand includes a logo,

a website with an official roster, as well as LinkedIn and Facebook pages. The staff are the “senior management” and the supervising professors are “board members” or “advisors” (Fig. 3). This role-play serves the functions of setting a whimsical mood, while communicating a clear message: the *team* includes the *whole company*; students, staff and board directors working towards the common objective of *learning robotics*.

## 5 Feedback from the first implementation (MIT, Spring 2016)

Duckietown was first offered as a graduate-level class, officially known as “MIT 2.166: *Autonomous Vehicles*”. It has exceeded our expectations because of:

- the large number of student applications (67) show the appeal of concept and topic;
- the proactivity of students and staff, who over-delivered in effort and results;
- the production of over fifty functional, standardized robots;
- the detailed documentation formalizing every step for reproducing the results;
- the media coverage and public interest (53k views on YouTube at time of writing).

### 5.1 Teaching staff

This first edition of Duckietown counted 34 staff, 13 of which were post-doctoral associates (Postdocs). Postdocs were recruited both before and during the course and performed most of the teaching activities. By volunteering their time and competences they gained: (a) *teaching experience*, by lecturing the class on their field of expertise; (b) *leadership experience*, by mentoring student teams in the development of features, (c) opportunity for *scientific publications*, enabled by having access to a complex experimental setup, and (d) the opportunity to gain detailed knowledge of a *customizable* and *open source* course that can potentially be used in their future academic careers.

### 5.2 Student Demographics

Applications were solicited, through a questionnaire, for 12 spots. We received 67 applications from students with backgrounds mainly in mechanical engineering and computer science departments. In addition to standard questions about educational background and previous experiences, each response needed to contain two essays: “Why would you like to take this course?” and “What can you contribute to Duckietown?”. Some of the responses are shown in Fig. 4 and highlight the students’ enthusiasm for joining the team. Out of 27 accepted students (1 dropped the course), eight were women (30%), which is significantly higher than the proportion of female graduate students in the Electrical Engineering and Computer Science Department at MIT (20%).

### 5.3 Deliverables

The course produced results that highlight the successful achievement of the ILOs. Students and mentors defined, developed and deployed the *functionalities* of the Duckiebots. This was accomplished during the group based *projects* (Table 3) in Phase II (Sec. 3.2). For a technical description of the projects we refer the interested reader to [9]. All these contributions were merged back into the base system, providing the behaviors that are now available to the broader community. The most obvious outcome of the course was the number of “functional” Duckiebots and compliant Duckietown(s) that were produced, along with the documentation for reproducing these results.

- [...] In short please please please let me take this class! Quack!

- [...] **This class is really the best class I've ever seen** [...] All the things about the Duckies are cool as it shows that you've had (and will have) a lot of fun creating this class. But I was even more impressed by what is actually going to be thought and developed over the span of this semester.

- [...] **This is the most excited I've been for a class at MIT!** [...]"

- [...] **I think that this class will be very challenging and exciting. I would be also really glad to contribute to a class that will be widespread in the next terms.**" [...]

- [...] **It looks really amazing!!!!** - the talk was super cool and I think this class will be awesome. [...]

- [...] Looking at the Facebook page it seems like this course has had a ton of effort and passion already poured into it. **I would love to be a part of it and to make it work!**" [...]

- [...] **This class is one rare chance** to explore - with a scientific approach - the state of the art in autonomous vehicles transversally, including hardware design, perception, planning, control. More importantly, it is a chance to hit the current technological boundaries in these fields, using a real system. The latter is very inspirational to guide my future research [...]"

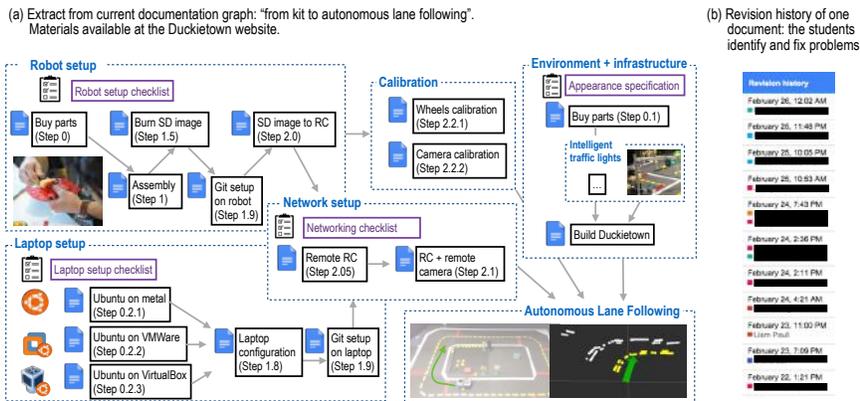
- [...] I am very interested in taking this course because, while I have extensive experience in mechanics and electronics and some experience with robotics and controls, I don't have much experience with uniting these in decision-making autonomous systems. **I can tell the class will be very enjoyable given the excitement and enthusiasm displayed by the staff.**"

- [...] The class is founded upon the philosophy that the best way to learn something is by using it to build cool stuff. Along those lines, I think Duckietown qualifies as "cool stuff". With regards to 6.S08, and hopefully 2.166, **I'm excited to help mold a new educational avenue for MIT students.**

**Fig. 4.** Some extracts from the answers to the question: “Why would you like to take this course?”

**Documentation/Reproducibility** Cooperative development of a complex system requires thorough documentation, an aspect often insufficiently stressed during engineering academic training. Students were required to document, through step-by-step instructions based on a given template, every contribution to the project. Fig. 5 shows, as an example, the overview of the autonomy tutorial instructions. Each step in the process links to open source documentation containing the implementation details, common challenges and troubleshooting options. Moreover, all the documentation being openly available allows for students to dynamically correct and improve it.

Based on these documents, two high school students managed to reproduce the robot behaviors from scratch with minimal supervision.



**Fig. 5.** (a) From kit to autonomous lane following in a dozen steps. (b) Extract of revision history for an instruction document: 12 people have contributed to this file in one week. The names of students are censored for privacy.

**Public Demonstration** Another measurable outcome of the course was the public demonstration, given in occasion of the MIT Open House Day. The installation area covered by Duckietown was roughly 330 m<sup>2</sup> (3600 sqft.) and the class constructed eight different Duckietowns (Fig. 6), deployed tens of Duckiebots and engaged over two thousand people over the course of a day. In each Duckietown area teams showcased a specific global behavior or functionality developed.

The public walked a path through the installation being gradually introduced to increasing levels of abstraction, from the mechanical design of a Duckiebot to “parallel autonomy” [28] and intersection coordination of multiple robots. At each station students presented posters describing the underlying mechanisms, ensured the continuous and correct operation of their demonstrations and answered questions from the public, effectively disseminating their work. Mentors oversaw the students’ efforts to ensure smooth operations.



**Fig. 6.** Overview of the demo day Duckietown installation layout. Over one thousand duckies were taken by the public on that day.

Public feedback was unanimously positive, and over one thousand souvenir duckies were taken by the end of the day.

#### 5.4 Reaching Underserved Demographics

##### **Experiment: Gender perception of Duckiebots during Duckietown 2016**

**Setup:** The students were asked to choose any name for their robot. We deliberately never mentioned any of our thoughts about gender and stereotypes reported above.

**Overview:** The data shows evidence of self-identification: if the name is not neutral, there is a strong correlation between the student gender identity and the perceived robot gender. Further data will be accumulated in future years to further test this hypothesis (we have not included names from staff operated Duckiebots).

**Details:** The students are eight women and 18 men. The names chosen by women are in *italic*. These are the Duckiebots with a **masculine name**: *Duckula*, Rex, Neptunus, Nikola, Ernie, Milo, Bill, Charles, Ayrton, Morty. These are the Duckiebots with a **feminine name**: *Julie*, *Lily*, *Ada*. These are the **neutral names**: *Oreo*, *Thing*, *Magitek*, *Cookie*, Quackmobile, Duckmobile, Amadobot, Penguin, Redrover, Lebeast, Pipquack, Setlist, Starducks, Maserati.

**Fig. 7.** Gender identification mini-experiment.

We have performed a test by observing what names the students chose for their robots (Fig. 7). The conclusion, from the available sample, is that the Duckiebots tend to be seen as feminine by women, and as masculine by men. Our current interpretation is that this is due to a process of self-identification with the duckie and/or the robot, which is evidence that we have been successful in removing gender cues from the Duckiebots, while making them very relatable. Our empirical observations show that the duckies are perceived particularly well by women. Therefore, although Duckietown is not focused on the recruitment of women, this effort might help towards reaching underrepresented talent from that demographic.

#### 5.5 Student Feedback and Discussion

At the end of the course we asked students to provide feedback, on a 1-7 scale, on three queues related to the perceived quality of teaching. The results are shown in Fig. 8. We

Quality of Teaching	Rating Scale: 1=Strongly Disagree, 4=Neutral, 7=Strongly Agree, N/A=Not Applicable (7 is best)							RESPONSES	MEDIAN	STDEV	
	AVG	1	2	3	4	5	6				7
Stimulated interest	6.1	■	■	■	■	■	■	■	15	7.0	1.49
Displayed thorough knowledge of subject material	6.0	■	■	■	■	■	■	■	15	7.0	1.69
Helped me learn	5.7	■	■	■	■	■	■	■	15	7.0	1.88

**Fig. 8.** Student feedback was overall positive and it highlights opportunities for improvement.

observe that most student were happy with the experience. The most critical feedback was related to the “helped me learn” queue, in particular it highlights a margin for improvement on the integration of the more theoretical aspects of the course in the hands-on activities. We believe this was a consequence of this being the pilot edition of the course. Fundamentally *nothing existed* of the Duckietown platform prior to this course and many structural “one-time” tasks had to be accomplished. For example: defining the communication protocols, the documentation templates, creating appearance specifications of cities, writing the instructions, coordinating staff and students, deciding what software to develop and how, etc. These tasks took a toll on the other aspects of the course. In subsequent editions less time and effort will be necessary for the definition of structural features, in favor harmonization of the theoretical topics underlying vehicle autonomy and the developmental component of the course.

Nonetheless, *we were overwhelmed by the student engagement*. In Duckietown, it was normal for students to fix problems before the instructors were aware of them, extend the system without being prompted to do so, and cultivate a healthy obsession with duckies. Moreover, students have pulled together to help their peers, as was encouraged through the course forum hosted on Slack [29].

We believe that these outcomes were enabled by the “one robot per child” policy, in turn made possible by the affordable platform design. The fact that every student had their own personal hardware to work with cultivated a vastly different dynamic as compared to the case where students are only allowed to use equipment during lab hours. Students brought their Duckiebots home and even on their travels during vacation time. Moreover, students were increasingly motivated by the possibility that their contribution to a regular class could live on beyond the course itself.

We moreover noticed how students put significant effort in creatively customizing their Duckiebot, *de facto* developing an emotional bond with it. We believe this dynamic was enabled by the choice of the duckie theme (Sec. 2.1), in particular for its contrast to the stereotypes on robotics, gender neutrality, and “cuteness” factor.

## 6 Conclusions

Duckietown is a *research* and *outreach* platform in addition to being an *educational* one. We believe that teaching, research and outreach are complementary and enhance learning when considered together.

Duckietown creates significant research opportunities by providing a modular experimental testbed with a multitude of autonomous vehicles. *Research* opportunities attract senior staff to volunteer their time and abilities in the course, enhancing the quality of learning for students while at the same time fostering collaborations.

*Outreach* fights common stereotypes attracting younger generations to robotics in particular and STEM fields in general. Interaction with the general public gives students the opportunity to “teach” themselves, increasing their subject knowledge as well as the awareness of their social role as future engineers and scientists.

From a resource standpoint, we note that the implementation of Duckietown as described required the use of several spaces: a small inventory room to store all the hardware and a laboratory permanently equipped with a Duckietown. An additional room was temporarily equipped with a second Duckietown during phase II of the course, to accommodate the need of students to frequently test their work in progress. From a hardware perspective, all students had their own laptop (per institute policy), with a few additional laptops (*Duckietops*) made available in case of need. All Duckiebot hardware was kept by students throughout the course, and handed back in at the end of it.

Overall, we feel that this paradigm for robotics education is novel and successful. The feedback that we have received from the robotics community, the MIT community, and the public at large has been overwhelmingly positive. It has created enthusiasm and active participation of students and staff, attracted national and international attention from other institutions (National Chia Tung University, Taiwan; Tsinghua University, China and Rensselaer Polytechnic Institute, USA have or are implementing versions of Duckietown aligned with their didactics) and media.

All materials produced during the course are freely available online, under open-source licenses. For pointers to all materials, please see <http://duckietown.mit.edu>.

## Acknowledgments

This work was funded by the National Science foundation through award IIS #1318392 and through the National Robotics Initiative award #1405259. The work was also supported by the Toyota Research Institute and the Ford Motor Company.

## References

1. Freeman, S., Eddy, S.L., McDonough, M., Smith, M.K., Okoroafor, N., Jordt, H., Wenderoth, M.P.: Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences* **111**(23) (2014) 8410–8415
2. Bloom, B.S.: The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher* **13**(6) (1984) 4–16
3. Wieman, C.E.: Large-scale comparison of science teaching methods sends clear message. *Proceedings of the National Academy of Sciences* **111**(23) (2014) 8319–8320
4. Halpern, D.F., Hakel, M.D.: Applying the science of learning to the university and beyond: Teaching for long-term retention and transfer. *Change: The Magazine of Higher Learning* **35**(4) (2003) 36–41
5. Wiggins, G., McTighe, J.: What is backward design? *Understanding by design* (2011) 7–19
6. Biggs, J.: Aligning teaching for constructing learning. *Higher Education Academy* (2003) 1–4
7. Monaco, M., Martin, M.: The millennial student: A new generation of learners. *Athletic Training Education Journal* **2** (apr-jun 2007) 42–46

8. Novotney, A.: Engaging the millennial learner. **41**(3) (Mar. 2010) 60
9. Paull, L., Tani, J., Ahn, H., Alonso-Mora, J., Carlone, L., Cap, M., Chen, Y.F., Choi, C., Dusek, J., Fang, Y., Okuyama, I.F., Hoehener, D., Liu, S.Y., Novitzky, M., Papis, J., Rosman, G., Varricchio, V., Wang, H.C., Yershov, D., Zhao, H., Benjamin, M., Carr, C., Zuber, M., Karaman, S., Frazzoli, E., Del Vecchio, D., Rus, D., How, J., Leonard, J., Censi, A.: Duckietown: an open, inexpensive and flexible platform for autonomy education and research . Submitted to the IEEE International Conference on Robotics and Automation (ICRA) 2017, <http://duckietown.mit.edu/materials.html>, [Accessed 15-Oct.-2016].
10. Olson, E.: Apriltag: A robust and flexible visual fiducial system. In: IEEE Intl. Conf. on Robotics and Automation (ICRA). (May 2011)
11. Siegel, M., Breazeal, C., Norton, M.I.: Persuasive robotics: The influence of robot gender on human behavior. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, Institute of Electrical & Electronics Engineers (IEEE) (oct 2009)
12. Green, T.: Why are too few females in robotics? could it be the robots? (2015)
13. Greensted, C.: Intended learning outcomes. *EFMD Global Focus* **8**(1) (2014) 20–25
14. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA workshop on open source software, Kobe, Japan (2009) 5
15. Sturm, P.: Pinhole camera model. In: *Computer Vision*. Springer (2014) 610–613
16. Zhang, Z.: A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence* **22**(11) (2000) 1330–1334
17. Daum, F.: Nonlinear filters: beyond the kalman filter. *IEEE Aerospace and Electronic Systems Magazine* **20**(8) (2005) 57–69
18. Karlsson, N., Di Bernardo, E., Ostrowski, J., Goncalves, L., Pirjanian, P., Munich, M.E.: The vslam algorithm for robust localization and mapping. In: *Proceedings of the 2005 IEEE international conference on robotics and automation, IEEE* (2005) 24–29
19. Masehian, E., Sedighzadeh, D.: Classic and heuristic approaches in robot motion planning—a chronological review. *World Academy of Science, Engineering and Technology* **29**(1) (2007) 101–106
20. Crowley, J.: Navigation for an intelligent mobile robot. *IEEE Journal on Robotics and Automation* **1**(1) (1985) 31–41
21. Hafner, M.R., Cunningham, D., Caminiti, L., Del Vecchio, D.: Cooperative collision avoidance at intersections: Algorithms and experiments. *IEEE Transactions on Intelligent Transportation Systems* **14**(3) (2013) 1162–1175
22. Duperret, J.M., Hafner, M.R., Del Vecchio, D.: Formal design of a provably safe robotic roundabout system. In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE* (2010) 2006–2011
23. Cao, Y., Yu, W., Ren, W., Chen, G.: An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics* **9**(1) (2013) 427–438
24. Thrun, S., Burgard, W., Fox, D.: *Probabilistic Robotics*. The MIT Press, Cambridge, MA (2005)
25. Tani, J., Mishra, S., Wen, J.T.: Motion blur-based state estimation. *IEEE Transactions on Control Systems Technology* **24**(3) (2016) 1012–1019
26. Tanenbaum, A.S., Van Steen, M.: *Distributed systems*. Prentice-Hall (2007)
27. Bonin-Font, F., Ortiz, A., Oliver, G.: Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems* **53**(3) (2008) 263–296
28. Naser, F., Dorhout, D., Proulx, S., Pendleton, S.D., Andersen, H., Schwarting, W., Paull, L., Alonso-Mora, J., Jr., M.H.A., Karaman, S., Tedrake, R., Leonard, J., Rus, D.: A parallel autonomy research platform. In: *IEEE Conference on Robotics and Automation (ICRA)*. (2017) 1–8 Submitted.
29. Slack: Slack (2016) [Online; accessed 10-October-2016].