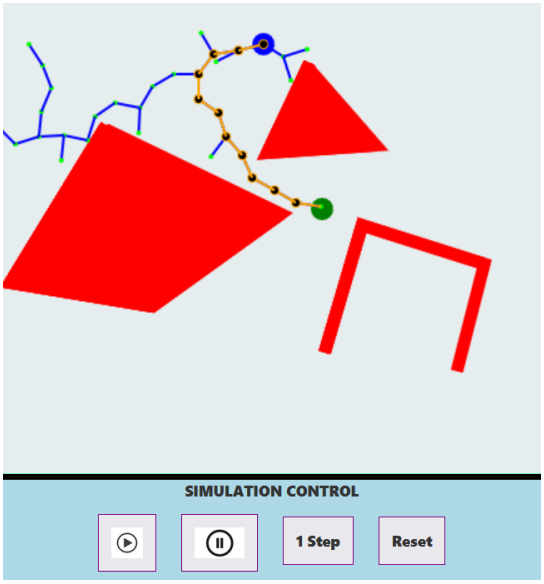
Interactive Robotics Education Tool

Purpose

Our goal is to create a demonstration tool that supports robotics education by visualizing different aspects of robotics algorithms and concepts, e.g., robot kinematics, motion planning, and controls. Target audiences include university students, advanced high school students, and community members interested in learning about how robots work.
We already have a basic version of this tool, which was developed by a great COMP523 team last year, but we are still missing some key components.

Existing Project

The existing client-based web application shows a robot in a 2D environment, lets the user make changes to this environment and visualizes the steps and functionality of different robot algorithms. The building blocks of the robot environment and the user interface already exist and are the same for many different robotics topics and algorithms.



The figure above shows a screenshot of the current application. A typical 2D robot planning scenario consists of a rectangular area that contains the current robot position (blue), several obstacles (red), and a goal (green), all of which can be modified by the user in the setup phase. In the simulation phase, the execution of a motion planning algorithm (RRT in this case) is visualized step by step. This helps the user understand how changes to the environment affect the algorithm's performance. The user is able to pause, rewind and adjust the execution speed. The system currently supports the RRT motion planning algorithm as well as several robot kinematics models (bicycle, tricycle and differential drive).

The existing code is publicly available via a BSD-style license and can be found on github. At our request the previous team also created extensive documentation both for users and for developers interested in expanding the project (hopefully, that's you!), which can be found here. There even is a video walking you through installing dependencies and providing an overview of the code structure.

Deliverables

We would like the team to expand our existing web application with the following elements:

- Bug Algorithms: This is a class of robot motion planning algorithms for robots that have no knowledge about their environment except the goal they want to reach. There are 3 variants of this algorithm varying the robot's decision points when to head straight for the goal and when to follow an obstacle it encounters. The algorithms are short (<10 lines) and do not require any prerequisite knowledge. Similar to the existing planning algorithm, users should be able to move the start and goal position and to draw obstacles.
- PID Controller: many tasks require robots to sense their environment and to react appropriately to the sensor input, correcting their current course of action if they are no longer on track to their goal. A control strategy decides how aggressive this correction should be. In the past, we have been using this simulation of an inverted pendulum in class, but it is much too complicated and goes too deep into underlying physics. Instead, we would like to have a very simple simulation of a robot following a line and correcting if it steers too much to the left or right. Users should be able to draw the line and to adjust the parameters of the controller in the UI to experience how these changes influence the robot's behavior. Again, no prior knowledge is required.
- Login portal: We would like to be able to track who and for how long is using which part of this application. For instance, in a classroom setting, it would be very interesting to see if using the application leads to better understanding of specific algorithms.
- Documentation: The previous group did a great job documenting the current state of the project. We would like you to expand this documentation to include the new features you are developing and to add any information that you wish had been provided to you when you started working on the project.
- Deployment: The project currently 'lives' only in a github repository. To make it accessible, we would like to run it on our website similar to this example.

Prerequisite Knowledge

No prior robotics knowledge is required. An undergraduate data structures class covering graphs (e.g., COMP 210) provides all necessary background knowledge, and all equations and/or pseudocode for the robotics algorithms will be provided. If the team is interested in deeper understanding of a particular algorithm to enable a better performing web application, we would be happy to provide a short tutorial.