Robot Competition - Status Report

Lewis Trundle

2469635T

Project Outline

Motivation

Over the last decade, robotic competitions such as, the AWS DeepRacer League have become increasingly popular. However, these competitions can require expensive robots and have always taken place in-person. Creating a new competition where anyone can enter their implementation from all around the world will encourage accessibility in this area and encourage more people to become interested and explore robotics.

Aims

The project has two key deliverables. The first is designing and developing a website for a self-driving competition. The website should outline the rules of the competition. The second is developing code for the robot to be used in the competition. The code should allow for both a controller for the robot, and the ability for the robot to drive itself. It should also run on a web-browser, meaning anyone around the world can use it without having to install anything.

Progress

- Background research conducted on other self-driving robot competitions.
- Co-operatively planned details of the competition, this includes: rules, track design, and details of each round.
- Designed different variations of a race-track and decided upon final version.
- Independently designed and developed a website for the competition.
    - Website was designed using Figma: Robot Competition Website.

    - Website developed using HTML, Bootstrap and Javascript. Is currently being hosted on Github Pages: UofG Cup.
- Created Google Form to accept submissions and automatically upload entrant data to Google Sheet.
- Created Youtube playlist for entrants to upload videos of their robot to.

Problems and Risks

Problems

- One robot stopped working which meant it couldn’t be used to test movement.
- Developing the website was more time-consuming that I would have liked. Additionally, using Github Pages also restricts the ability for dynamic content.
- In general, a heavy workload for other courses this semester has reduced progress on the project. However, this should improve next semester.

Risks

- Overloading the robot with instructions causes it to stop.
    - Mitigation: only send instructions across every couple of seconds.
- Low memory storage of espruino device
    - Mitigation: reduce the code on the robot as much as possible.

Plan

Week 1-2:

Get joystick controller working for robot.

Fix general bugs on website.

Make plan for a user study to test joystick

Week 3-4

Implement line-tracking ability for robot.

Test joystick implementation via a user study and identify areas of improvement.

Week 5-6

Fix any bugs with joystick.

Implement object avoidance for robot.

Week 7-8

Fully test the robot

Write up first draft of dissertation.

Week 8-10

Make final improvements to robot.

Write up final draft of dissertation.