CP31 –
Scope and Requirements Document

Updated: 2020-09-07 11:25 PM

Below is the scope document for the project involving Computer Vision tasks and improving the Donkey Car simulator. Contained in this document is the client expectations for the semester.

[Initial Project Outline]

Contents

Scope & Requirements

This project is broken into two challenge and problem components that are closely related.

Challenge & Problem 1: Simulator improvements.

Challenge & Problem 2: Sign, Path & Object Detection using only Computer Vision (OpenCV)

The goal of this project is for students to leave with an excellent understanding on simulation in Unity and computer vision (OpenCV) technologies. Students will also gain valuable project management, industry and professional experience.

Scope and Requirements Document
USYD Capstone 2020

High-Level Scope Summary

- **Milestone 1 – Simulator Additions (Week 3)**
    1. Add two new track layouts (outlined in email)
    2. Add objects to map
- **Milestone 2 – Simulator Enhancements (Week 9)**
    1. Improve and documented new fast way to import new track layouts
    2. Add up to two additional (four total) track layouts to simulator and document process
    3. Further improvements and documenting for placing objects (signs, traffic lights, objects)
- **Milestone 3 – Computer Vision minor sign detection (Week 6)**
    1. Improve existing solution (provided code)
    2. Research and Document improvements, show results
    3. Demo on sample data and in simulator (different environments)
    4. Add extra objects for detection not in original solution
- **Milestone 4 – Computer Vision major detection (Week 9)**
    1. All outlined signs detected and corresponding action response
    2. Demo on sample data and in simulator (different environments)
    3. *Advanced*: Line following and Lane Detection, with response from object actions
    4. *Advanced*: Lanes of different colours (white, yellow, etc) based on environment
    5. *Advanced*: No Lanes solution / detection so vehicle still follows a path (e.g. a footpath)
- **Milestone 5 – Completed Solution with Demo and Documentation (Week 12)**
    1. Usage documentation (as per each Milestone)
    2. Demo of working solution in simulator
    3. Demo of working solution on real-world track (TBD)
    4. Deliverables as per 'Hand-over deliverables' section

Summary of Simulator Improvements

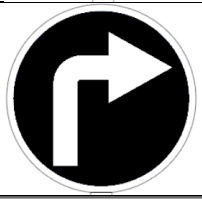
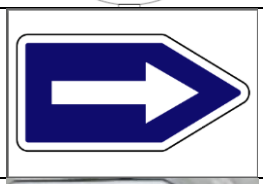
Milestones 1 & 2 expanded.

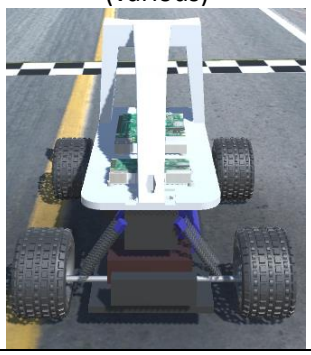
- **New Tracks & Environments:**
    1. Imported Track 1 (Robotics Masters Simple)
    2. Imported Track 2 (Robotics Masters Challenge Course)
    3. Imported Track 3 (One-way Circuit) – by automatic generation
    4. Imported Track 4 (Albert Park) – by automatic generation
- **Objects**:
    1. All signs (in table) modelled in the simulator and able to be placed at set locations
    2. New Donkey Car from RoboCarStore modelled in simulator *(Advanced – if time permits)*
- **Features:**
    1. Automatic track generation from PNG or JPG image for flat tracks
- **Extended**:
    1. Students welcome to improve any other aspect of the Unity Simulator that they feel is required.

Signs & Objects

Some of the signs in this table will need high-resolution copies made for modelling in simulators. All objects below are expected to be modelled in the simulator.

For the purposes of consistency, the dimensions of all the signs are locked to 100 mm by 100mm areas (measured to the edge of the frame for round signs, and on the height axis for non-square ones). All signs are mounted on black poles.

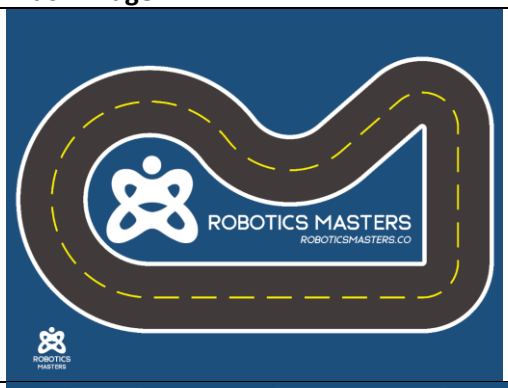
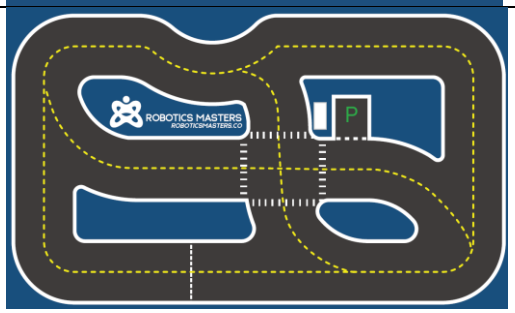
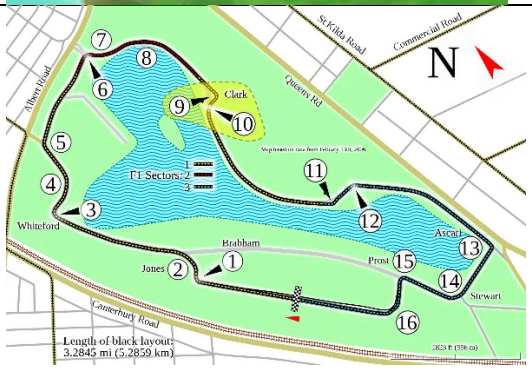
Description	Image (Sample)	Actions
Stop		Stop the car at the white line (if present) for 3 seconds
Traffic Lights (Red, Amber, Green) Sample data provided to students.		**Red**: Stop the car at the white line and wait for green light **Green**: Go or continue **Amber**: Open interpretation. Either choose continue or stop
Turn Blue round Left & Right		**Turn** the car in the direction of the sign immediately
Turn Black round		**Turn** the car in the direction of the sign immediately
Turn Blue Arrow Left & Right		**Turn** the car in the direction of the sign immediately
Turn white on black Left & Right		**Turn** the car in the direction of the sign immediately
Park Green		**Park Routine**: Stop the car (subject to change later)

Park White		**Park Routine**: Stop the car (subject to change later)
Park Yellow		**Park Routine**: Stop the car (subject to change later)
Orange Traffic Cone		**Collision Avoidance**: Avoid the traffic cone by going left or right. Do not colid
Speed Sign (Advanced) (read out / report speed as integer) 5 10 25 40 50		**No Action**: Report the detected speed in the console (print)
Other cars in the simulator (Advanced)	(various) 	**No Action**: Report the detected speed in the console (print)
Lane following (Advanced)	Yellow Lines White Lines White Lines with Yellow Lanes	**Lane Detection** Stay between a left or right lane. Stay on the track.

* Advanced are for when students have completed and demonstrated all other signs in the simulator and real-world environment. Students are still requested to model the speed signs as part of the simulator improvement, even if they don't detect them.

Tracks

For the purposes of this unit, we are adding in extra tracks to the simulator. They are outlined and described below. Some tracks still do not have the correct picture associated with them. However, example screenshots are provided.

Track Name	Track Image	Description
Robotics Masters – Circuit		Simple Circuit with white edge lines and a dotted yellow centre lane line.
Robotics Masters – Challenge Course		Challenge course that includes multiple turns and decision making. Designed to have signs placed throughout the course. White edge lines with yellow dotted lanes. This course must include blue walls around the exterior that are 1.0 meters high.
Baby Park circuit - only single turns, three lanes	(yet to be drawn) 	This circuit is designed for testing the augmentation function in donkey car. The circuit consists of only a single direction (right/left) turn circuit with two straights. The circuit must have three lanes (two white edges, two yellow dashed lines).
Albert Park – F1 Race track in Melbourne (Advanced)		Model a 1:10 scale of the Melbourne F1 Track. Include white lines on the edge of the course and a yellow lane marker. NOTE: This should be modelled using the simulator improvement for importing tracks.

Client Expectations

General

This project is broken into two major components each with two milestones. This scope document will outline the expected delivery of each milestone as per the *High-Level Scope Summary* list found below. Students are also expected to do lots of research to ensure the best solutions are found, with research recorded in the appropriate repository and format. Documentation must be included with all work completed in a format that is legible and understandable.

The client reserves the right to amend the scope of the project throughout the semester based on team progress and unforeseen challenges that may arise. The client reserves the right to communicate problems with tutors to be raised with the course coordinator to take academic action if deemed essential for lack of progress or not meeting expectations.

Contribution

The client (Robotics Masters) expects that students meet and exceed the scope and requirements outlined in this document to achieve the best grade in the course. Each individual student in the team are expected to contribute to the project. Contribution and time expected per a week on this project is approximately 15+ hours per week for each student, excluding meetings. That is a total contribution of 60 to 90+ hours per week for each team.

Contribution to the project will be tracked through Bitbucket (there should be constant daily commits being made by all team members), communication (activity on Discord Server) and through all updates provided to the client. This information will be used at the end of the semester when it comes to evaluating everyone's contribution and final grade for the capstone unit.

Meetings

Each team is expected to attend two Zoom meetings per week. The times have now been sent to everyone.

Each team member is expected to submit and present a 1-minute summary of their contribution for each week on Monday/Tuesday meetings. These videos should be uploaded to YouTube as an unlisted video, presented during each Monday/Tuesday meeting and shared to the client after the meeting. Feedback will be given after each individual video during the meeting.

Any technical questions that arise should be asked during these meetings.

Each team is to present and submit the plan for next 'sprint' period at the end of each meeting. This should contain a list of tasks that each team member is working on for the next time period and outline what is expected to be delivered at the next meeting.

Communication

Email – the client prefers email communication for official documents, scope questions and communicating with tutors. The client will also respond to technical and scope questions via email, however, would prefer that the Discord Server (Sydney 2020) is used.

Discord – the client has selected Discord and email as the preferred tool to use throughout the semester for questions and notices. Teams will also be collaborating/discussing/sharing ideas with other teams run by Ben Sand (another client). There is a total of 15 capstone teams taking part in

this initiative. This kind of collaboration has been done before and it has been very positive for students and opens several different opportunities, such as meeting tech influencers, established developers and other technology specialists.

Events

Students are expected to participate in the following activities throughout the semester as part of the core component of this project and to ensure understanding of the project

Virtual Racing League – These events are held throughout the year. The event consists of teams training a CNN model using the simulator and racing for the fastest lap time. The client expects teams to participate in *at least one* of these events to gain a better understanding of how the Donkey Car Platform operates. The dates for the races throughout the semester are; next race is 25th September 2020 and then in Late October / Early November.

Given the time zone difference with the US and if the Donkey Car Community decides to run this event at 3:00 AM like the normally do, the client will set up their own Virtual Racing League (pending testing) and will announce the dates for a particular weekend. This event is to ensure that teams have a good understanding of the simulator and Donkey Car environment.

Venture Café - Students will be required to participate and present at two Venture Café sessions throughout the semester. It is an online meeting community where different creators, start-up owners meet and discuss what they are doing with their own projects. It works like a 'drop-in session' where people may only join for a short amount of time or could stay for longer. Teams will be sharing your progress and discoveries through this platform. Teams are welcome to attend Venture Cafe on any Thursday to get a feel for the platform and chat with other technology people.

Hand-over Deliverables

Documentation – All documentation is to be created and written in Markdown, then build with Readthedocs for portability and usability.

Jupyter Notebooks – All final code that involves testing a system (e.g. detecting a turning sign) should be published and submitted as a Jupyter notebook. Existing code can be converted over the last two weeks of the project.

Libraries – The client will outline with the team how the final Python library is to be structured. The teams are expected to refactor the code base to this structure throughout the project for testing in different environments throughout the semester.

Reference Material

All reference materials will be provided by the client to teams via Discord or email.

End of Document