Autobot Racing
Team 21
Anthony Goeckner, Ben Huemann, Zach Perry, Harold Smith

Sprint 1 Planning Document

Sprint Overview

In this sprint, we hope to accomplish building the framework of the backend, begin development of main UI, receive hardware, create interfaces with the hardware, and start testing the computer vision, in addition to performing basic modelling of a simple control system.

Scrum Master: Harold Smith
Weekly Meetings will be held on Wednesday evenings

Risks:

- Hardware is damaged
- Hardware is not what was expected
- Cannot interface between software and cars
- Cannot interface between software and camera

Current Sprint Detail

Functional

1. **As a developer, I would like to provide an API allowing researchers to perform control-systems research on autonomous vehicles.**
    a. Tasks:
        i. Create APIs for use by researchers to allow the use of researcher-designed control systems.
        ii. Test the associated APIs.
    b. Acceptance Criteria:
        i. An API is created which allows for multiple control systems to be used simultaneously, each on a different vehicle.
        ii. The framework provides an interface to the camera to allow for computer vision.
        iii. Given a certain specification for the structure of a control system, a researcher can create a control system that will interact successfully with the framework.
    c. Assigned To: Harold Smith + Ben Huemann

   d. Workload: 16 hours

2. **As a developer, I would like a communication system between the PC and multiple Raspberry Pis**
   a. Tasks:
      i. Create messages data type that is robust in format to allow for easy modification
      ii. Create a way to allow PC to communicate to all Raspberry Pis
      iii. Create a way to allow PC to communicate to individual Raspberry Pis
      iv. Create a way to allow for Raspberry Pis to communicate with the PC
      v. Test message passing of each function for correctness
      vi. Test message passing of each function for desired timing
   b. Acceptance Criteria
      i. Given data on the PC when sent as a message to all Raspberry Pis then the same message will be received on every pi
      ii. Given data on the PC when sent as a message to individual Raspberry Pis then the same message will be received only on that pi
      iii. Given a reply on the Raspberry Pi when sent as a message to the PC then the same message will be received on the PC
   c. Assigned to: Ben Huemann
   d. Workload: 12 hours
3. **As a developer, I would like to control the entire system using a PC or web-based interface.**
   a. Tasks
      i. Pick the library to use for developing the UI
      ii. Draw out rough outline of desired UI and necessary components
      iii. Begin coding the UI
      iv. Test the functionality of each component in the UI
   b. Acceptance Criteria
      i. A rough sketch of the main UI was drawn including all of the necessary components
      ii. Half, or more, of the desired components were implemented
      iii. Given a UI component when a user uses the component then the component will give the desired functionality
   c. Assigned to: Harold Smith
   d. Workload: 20+ hours
4. **As a developer, I would like to have an efficient message passing system.**
   a. Tasks
      i. Create message passing functions for communication between classes
      ii. Test message passing of each function for correctness

iii. Test message passing of each function for desired timing

b. Acceptance Criteria

i. Given every class when trying to communicate between classes then each class should have the functionality to pass messages to one another

ii. Given a well formed message when sent then the message should be received at the destination with complete correction

iii. Given a well formed message when sent then the message should be received in a desired time

iv. Given a high volume of data traffic through the message-passing system, all messages will be delivered reliably and in a timely fashion.

c. Assigned To: Harold Smith

d. Workload: 7+ hours

5. **As a developer, I would like to model a basic control system for autonomous driving.**
   - **a.** Tasks
     i. Create a basic proportional controller to maintain correct heading.
     ii. Create ability to induce error in simulated inputs.
     iii. Apply simulated inputs to control system to determine response.
     iv. Test output of control system for expected responses.
   - **b.** Acceptance Criteria
     i. Given some simulated error, control system is able to maintain a desirable course.
     ii. Given a small amount of error, the control system will display a minimal amount of overshoot.
     iii. Given a modeled control system, the output is easily comprehensible.
   - **c.** Assigned To: Anthony Goeckner
   - **d.** Workload: 7 hours
6. **As a developer, I would like to create a basic simulation program to model control system behavior.**
   - **a.** Tasks
     i. Display simulated position and orientation of vehicle.
     ii. Display desired course of vehicle.
     iii. Display actual course of vehicle.
     iv. Display a simulated course or navigable area.
     v. Test simulation program with a variety to control algorithms to determine correct operation.
   - **b.** Acceptance Criteria
     i. Given a simulation, the position and orientation of the vehicle are accurately displayed.

ii. Given a simulation, the desired course of the vehicle is accurately displayed.
iii. Given a simulation, the actual course of the vehicle is accurately displayed.
iv. Given a simulation, the navigable area of the track is accurately displayed.

c. Assigned To**:** Anthony Goeckner
d. Workload**:** 3 hours

7. **As a developer, I would like to begin creation of a computer vision system capable of tracking vehicles as they navigate a racecourse.**
   a. Tasks
      i. Map a coordinate system onto the image given by the camera.
      ii. Test recognizing individual vehicles.
      iii. Determine the orientation of each vehicle.
      iv. Test recognizing the outline of the course.
   b. Acceptance Criteria
      i. Given multiple vehicles in view of the camera, each can be uniquely identified.
      ii. Given a moving vehicle, it can be identified as accurately as a vehicle sitting still.
      iii. Given the computer vision system and a standard web-camera, a high image sampling rate can be achieved.
      iv. Given an image containing vehicles, each vehicle can be precisely located on the internal coordinate system.
   c. Assigned To: Anthony Goeckner
   d. Workload: 20 hours
8. **As a developer, I would like to ensure the computer vision software is able to easily identify individual cars and the track boundaries.**
   a. Tasks:
      i. Test computer vision against multiple cars (without track)
      ii. Test computer vision against track (without cars)
      iii. Test computer vision against cars and track
   b. Acceptance Criteria:
      i. Given cars are tested independently, when tested for computer vision functionality the cars will be uniquely identified along with their orientation.
      ii. Given track is tested independently, when testing for computer vision functionality the track should be identified along with its boundaries

iii. Given cars are overlaid on top of the track when testing for computer vision functionality, both should be identified independently

c. Assigned to: Ben Huemann
d. Workload: 10 hours

9. **As a developer, I would like to research ways to interface the RF transmitters and the Raspberry Pis.**
    a. Tasks:
        i. Investigate hardware used in RC remote
        ii. Test unmodified remote for current functionality
        iii. Test feasibility of electrical connection
        iv. Create plan of action to connect Pis to remotes
        v. Modify hardware to allow Pi to operate RC cars via software
        vi. Create basic software driver to control remotes via software.
    b. Acceptance Criteria
        i. Successfully attempt to connect Raspberry Pi with RC controller
        ii. Given a Raspberry Pi, when connecting to an RC remote, the Raspberry Pi should have no hardware connection errors
        iii. Given a software-controlled remote, the software should be capable of performing all operations originally available on the remote.
    c. Assigned to: Zach Perry
    d. Workload: 10+ hours

Non-Functional

1. **As a developer, I would like to find, create a list of, and order the hardware required for the project**
    a. Tasks:
        i. Search for hardware (track, car, camera, etc)
        ii. Create a bill of materials
        iii. Contact Victory Soe
        iv. Wait for hardware arrival
        v. Pick up hardware
        vi. Test hardware for functionality
    b. Acceptance Criteria:
        i. Given a list of hardware necessities when we ask Victory for approval then the hardware purchases are approved
        ii. Given hardware list when hardware is purchased then hardware will arrive in a timely manner

iii. Given hardware arrives when the development team arrive then hardware will be received by the team
iv. Given hardware when hardware is used/tested then the hardware will work as expected

c. Assigned to: Zach Perry
d. Workload: ~10 hours, plus time waiting for hardware ordering/arrival

2. **As a developer, I would like add physical identifiers to the cars and track to make them more identifiable to the vision system**
    a. Tasks:
        i. Add an identifier to the top of each car, notecard or paint, depending on what is easier to identify for the computer vision software
        ii. Create designs for each car that are unique
        iii. Ensure design designates orientation of the car
        iv. Clearly mark the edges of the track
    b. Acceptance Criteria:
        i. Given a note card for each car, when attached to the cars they should not hinder movement of the car
        ii. Given note cards are attached to the cars, when cars are moving there should be no shaking/movement of the cards
        iii. Given multiple cars, when attempting to distinguish between them each design is easily distinguishable from the rest
        iv. Given a car, when attempting to find the orientation of the car, the orientation is easily distinguishable based on the note card
        v. If notecards prove too complicated, paint may be used instead
    c. Assigned to: Zach Perry
    d. Workload: 10+ hours

Autobot Racing
Team 21
Anthony Goeckner, Ben Huemann, Zach Perry, Harold Smith

Backlog

Functional Requirements:

As a researcher...

- I would like the cars to navigate a set course.
- I would like to implement autonomous lane-keeping behavior.
- I would like to implement autonomous passing behavior.
- I would like to implement speed-holding behavior.
- I would like to implement automatic braking behavior.
- I would like to implement automatic acceleration behavior.
- I would like to implement immobile obstacle avoidance. (if time allows)
- I would like to implement moving object avoidance. (if time allows)
- I would like to record race statistics, such as lap times.

As a developer...

- I would like to determine the course followed by the cars, including the course around obstacles.
- I would like to interact with an RF controller to send signals to the cars.
- I would like to control a minimum of three cars simultaneously.
- I would like to identify immobile obstacles using computer vision. (if time allows)
- I would like to identify moving obstacles using computer vision. (if time allows)
- I would like to use a single camera, mounted above the course, for all object tracking and localization.

Non-Functional Requirements:

As a developer…

- I would like to have code that is sufficiently documented and well formatted.
- I would like code that is modular and built to accommodate updates in the near or far future.
- I would like the application programming interface (allowing researchers to control cars) to be flexible and well-designed.

As a project owner...

- I would like hardware costs to be reasonable and well-controlled.
- I would like for off-the-shelf hardware to be used in development, in order to decrease construction costs.

Autobot Racing
Team 21
Anthony Goeckner, Ben Huemann, Zach Perry, Harold Smith

Sprint 1 Workload Distribution

Team Member	Hours
Anthony Goeckner	30
Ben Huemann	30
Zach Perry	30
Hal Smith	35