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

Sprint 3 Planning Document

Sprint Overview

The primary goal of Sprint 3 will be to finish integrating all components and to perform integration testing of the entire system. It will also include several tasks adding new, required functionality, such as recognition of the racetrack boundaries. The end result will be a fully-functional racing system.

Scrum Master: Anthony Goeckner
Weekly meetings will be held on Wednesday and Friday evenings, with additional meetings on Saturdays or Sundays as needed.

Risks:
- Hardware failure/breakage during testing.
- Difficulty creating initialization vectors for Kalman Filter
- Difficulty tuning the computer vision system to work with different lighting systems.
- Some user stories require significant research and may be very time consuming.

Current Sprint Detail

Functional Requirements

1. **As a developer, I would like to add functionality to start, stop, and pause a race.**
    a. Tasks:
        i. Write functions to connect the currently existing buttons.
        ii. Write a start function to start a new race.
        iii. Write a stop function to permanently stop a race.
        iv. Write a pause function to temporarily stop a race.
        v. Write a resume function to resume a paused race.
        vi. Test the start/stop/pause/resume functionality and buttons under various race conditions.
    b. Acceptance Criteria:
        i. Clicking the “Start Race” button must start the race, causing cars to begin to move.

   ii. Clicking the “Pause Race” button should temporarily stop a race. This stops the cars but the cars should resume from their current positions once the resume button is clicked.
   iii. Clicking the “Resume Race” button should restart a previously paused race. Cars should continue from their paused positions, and the race will continue as normal.
   iv. Clicking the “Stop Race” button must immediately halt the race, stopping the cars. This ends the race permanently, and cars must be reset before a new race can begin.
 c. Assigned To: Zach Perry
 d. Workload: 10 hours

2. **As a developer, I would like to determine a start and finish line on the racetrack.**
   a. Tasks:
      i. Create a virtual or physical line across the track
      ii. Map the coordinates of the line in with the track boundaries
      iii. Use OpenCV to outline the start line on the camera feed
      iv. Test the creation of a start/finish line.
      v. Test recognition and mapping of start line
   b. Acceptance Criteria:
      i. Given a start line, OpenCV will be able to detect and map the line
      ii. Given a start line, the Camera feed will display, if virtual, the line on the camera feed
      iii. Given a start line the a track and a start line, the two are differentiated by the computer vision system
   c. Assigned To: Harold Smith
   d. Workload: 10 hours

3. **As a developer, I would like to determine the number of laps each vehicle has travelled and display them in the UI.**
   a. Tasks:
      i. Initialize each new car’s statistical values
      ii. Track each individual car and detect car passing through start line
      iii. Update and store new statistical values for the individual car
      iv. Send updated values to the UI
      v. Test lap counting capabilities with multiple cars.
      vi. Test lap counting when the car crosses different points of the start/finish line.
   b. Acceptance Criteria:

i. Given a new car, the car will have statistical values initialized to zero or None
ii. Given a car passing the start line, the counter for laps and lap times will be updated for the individual car
iii. Given a car passing the start line, the updated values for the individual car will be sent to the UI
iv. Given a set of updated values, the UI will update the values for the individual car in the “Leaderboard” frame

c. Assigned To: Harold Smith
d. Workload: 5 hours

4. As a researcher, I would like to implement acceleration in straight sections of the track.

a. Tasks:
i. Change the guidance algorithms to increase speed of the cars when in a straight section.
ii. Change the guidance algorithms to return the cars to a normal speed and slow down at the end of a corner.
iii. Change the guidance algorithms to prevent on-track collisions and offs due to increased speed.
iv. Test new guidance algorithms.

b. Acceptance Criteria:
i. Given a car enters a straight section of track, it should accelerate out of the previous corner before slowing down for the next corner.
ii. Given the cars are moving around the track, the cars should not crash or go off course.
iii. Given the cars are moving around the track, the cars should not accelerate into a car in front of them.

c. Assigned To: Zach Perry
d. Workload: 10 hours

5. As a developer, I would like to change the speed control from discrete to continuous.

a. Tasks:
i. Use a continuous instead of a discrete throttle setting.
ii. Change the speed parameter from an int to a float in the networking code.
iii. Test to make sure that no functionality or control has been lost.

b. Acceptance Criteria:

i. Given an equivalent throttle level, cars must respond the same with the new system as they did with the previous system.
ii. All function calls must operate the same as before, error free.
iii. Given a value between -1.0 and 1.0, the system must set the throttle to that percentage of its maximum forward and backward (negative) speeds.

c. Assigned To: Zach Perry
d. Workload: 5 hours

6. **As a developer, I would like to improve the CV Vehicle recognition system so that it functions better in different lighting conditions.**
   a. Tasks:
      i. Perform testing in different lighting environments to determine computer vision failure modes.
      ii. Make adjustments to CV tuning parameters to compensate for different lighting conditions.
      iii. Research methods to automatically adjust CV parameter tuning based on environment.
      iv. Test adjustments to ensure that the computer vision system works reliably in environments commonly found in the Lawson Computer Science Building.
   b. Acceptance Criteria:
      i. Given multiple reasonable lighting environments (in LWSN), the computer vision system is robust enough to function in any of them.
      ii. Given a proper lighting environment, the computer vision system is able to capture and track moving vehicles in all locations on the track.
      iii. Given a proper lighting environment, the computer vision system tracks moving vehicles with minimal “drops”, or brief periods of time when vehicles cannot be detected.
   c. Assigned To: Anthony Goeckner
   d. Workload: 20 hours

7. **As a developer, I would like to determine the outline of the track using only a computer vision system.**
   a. Tasks:
      i. Use OpenCV “contour” feature to determine the layout of the track.
      ii. From the edges of the contour, generate a simpler polygon with a minimal number of vertices.

iii. Determine the navigable areas visible to the camera by finding both the inner and the outer boundaries of the track.
iv. Test the system in a variety of lighting conditions
v. Test the system in a variety of track configurations.

b. Acceptance Criteria:
i. Given a track with an inner and an outer boundary, the computer vision system will determine the navigable area.
ii. Computer vision will only be used to identify the track during system initialization, at which time the layout will be stored for future use.
iii. Given multiple reasonable lighting environments (in LWSN), the computer vision system is robust enough to function in any of them.

c. Assigned To: Harold Smith
d. Workload: 25 hours

8. As a developer, I would like to integrate Kalman filtering with the computer vision system.

a. Tasks:
i. Determine a Kalman filter library to use.
ii. Determine and populate all initialization vectors of Kalman filter object
iii. Integrate filter input with position and heading data.
iv. Integrate filter output with existing navigation, guidance, and control systems.
v. Test integration of the completed system.

b. Acceptance Criteria:
i. Given noisy input position/heading data, the Kalman filter outputs a steady, reliable estimation of the true position.
ii. Given accurate input position/heading data, the Kalman filter output should not vary significantly.
iii. Given real time input position/heading data, the Kalman filter should provide a continuous stream of normalized position data

c. Assigned To: Anthony Goeckner, Ben Huemann
d. Workload: 20 hours (10 per person)

9. As a developer, I would like to add power on/off functionality to RC controller

a. Tasks:
i. Initialize GPIO pins used to send power to controller
ii. Wire up RC controller to GPIO pins
iii. Power up controller in RCDriver initialization subroutine
iv. Power down controller in RCDriver deinitialization subroutine

v. Test GPIO sends power to controller
vi. Test controller is initialized after digital potentiometer

b. Acceptance Criteria:
i. Given GPIO is set to HIGH, the controller should power on
ii. Given GPIO is set to LOW, the controller should power off
iii. Given RCDriver is initialized, digital potentiometer should be set before controller is powered on

c. Assigned To: Ben Huemann

d. Workload: 15 hours

10. As a developer, I would to implement the Integral portion of a PID controller

a. Tasks
i. Research possible ways that the integral portion of a PID controller is implemented.
ii. Implement the Integral calculation and add to the existing proportional calculation.
iii. Test PID controller with computer vision input.

b. Acceptance Criteria
i. Given some error, control system is able to maintain a desirable course.
ii. Given some error and a correction for that error, the control system will display a minimal amount of overshoot.
iii. Given input, normalized output should be better tuned than the just the stand-alone proportional mode.

c. Assigned to: Anthony Goeckner, Ben Huemann

d. Workload: 20 hours (10 per person)

11. As a developer, I would like to create better vehicle tags and enclosures for the control boards.

a. Tasks:
i. Create new vehicle tags with different identification colors
ii. Create enclosures for the control boards
iii. Test that vehicle tags are correctly identified by the existing computer vision system.
iv. Test that vehicle tags are durable and do not shift position or orientation while driving.

b. Acceptance Criteria:
i. Tags must be durable.
ii. Tags must be visible to the CV system.
1. They must reflect light evenly.

2. They must each be individually colored for each car.

iii. Control enclosures must protect the boards while in storage and use.

c. Assigned To: Zach Perry

d. Workload: 10 hours

12. As a developer, I would like to perform component-level and overall integration testing of the completed system.

a. Tasks:

i. Test integration between each individual pair of components.
1. CV -> Kalman Filter
2. Kalman Filter -> Navigation, Guidance, Control
3. Navigation, Guidance, Control -> Network
4. Network -> Pi
5. Pi -> Transmitter

ii. Perform overall integration testing of the entire system.
1. This involves racing vehicles around the track in normal operating conditions.

b. Acceptance Criteria:

i. All individual component integrations are tested successfully.

ii. Overall integration testing is performed successfully.

iii. The overall integration test is only considered a success if all component integration tests are successful.

iv. The overall integration test must be successful.

c. Assigned To: Anthony Goeckner, Ben Huemann, Zach Perry, Harold Smith

d. Workload: 20 hours (5 per person)

Non-Functional Requirements

There are no non-functional tasks to complete for this sprint.

Backlog

Functional Requirements:

As a researcher...

- I would like to implement immobile obstacle avoidance. (if time allows)
- I would like to implement moving object avoidance. (if time allows)

As a developer...

- 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)

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 foreseeable 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.

Sprint 3 Workload Distribution

Team Member	**Hours**
Anthony Goeckner	45
Ben Huemann	40
Zach Perry	40
Hal Smith	45