Project Plan

2.1 Project management/Tracking procedures

Hybrid agile/waterfall project management

Justification: Our project elements are very much in lock step. We need to get the car to drive before we can get it to steer. We need to get an app up and running before we can connect it to the car. We feel that a waterfall approach will fit our project well since we plan on working in order on the tasks that need to be completed

Project Tracking: Github

2.2 Task decomposition

Task Name	Super Task	Requires
Rabbit		
Physical System	Rabbit	● Select Microcontrollers ● Configure MCs to connect to one another ● Configure the MC system to connect to the Rabbit ● Decode PWM signals to build a vocabulary to communicate with the Rabbit ● Create mount for additional components
Onboard control/interface	Rabbit	● Physical System ● Bluetooth link to app ● Starting mechanism(Sound, Bluetooth, etc.) ● Batterylife (car to app)
Computer Vision	Rabbit	● Physical System ● Onboard control/interface ● Determine input source ● Line processing model ● Figure out steering algorithm
App		

Rabbit control	App	● Physical System, onboard control/interface ● Calculations to determine speed to set the rabbit ● BT connection to MC System
User interface	App	● UX Design ● Basic Parameters for workouts

- Rabbit
    - Physical Systems
    - Computer Vision
    - Onboard control/interface
- App
    - Rabbit Control
    - User interface

2.3 PROJECT PROPOSED MILESTONES,METRICS, AND EVALUATION CRITERIA

- Select/buy Micro Controllers
- Buy RC Car
- Select/buy sensors
- Integrate MCs into Rabbit
- Run simple driving commands from the MC (drive forward, *maybe* turn)
- Develop base level app
- Integrate app and Rabbit so app generated instructions can make the Rabbit move
- Develop Computer Vision
- Test computer vision decision making
- Integrate computer vision so the car can make it around the track on it’s own
- Improvements and efficiencies

2.4 PROJECT TIMELINE/SCHEDULE

October - Order project equipment/components, Integrate MCs into Rabbit, Start working on Basic UX for app, Research/start computer vision, research vehicle platform (mounting components)

November - Get the car to drive using the MCs, connect app to MCs and send simple instructions

December - Integrate Computer vision to MCs, Car can drive around the track by itself

January - Continuation of December, catch up on any backlog work

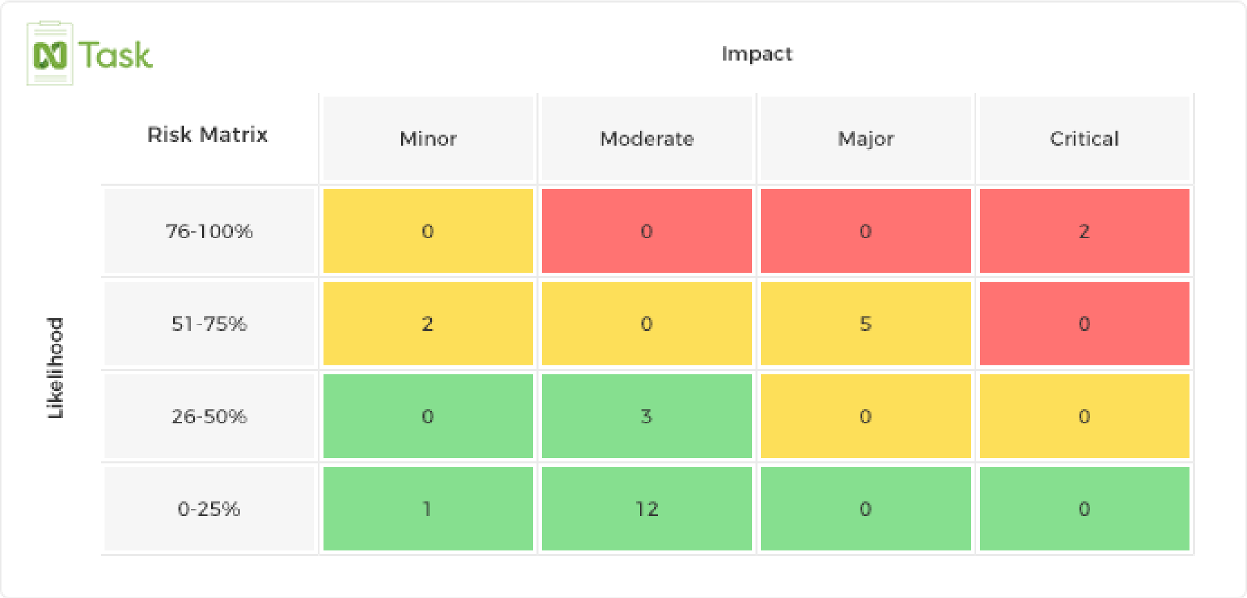
February - Polish existing features, projecting pace line, talk to Track Team

March - Start optimizing, speed control, performance testing

April - Fun features, Marketing Design

May - Continuation of April, Documentation, Presentation

2.5 RISKS AND RISK MANAGEMENT/MITIGATION



Task

		Impact			
	Risk Matrix	Minor	Moderate	Major	Critical
Likelihood	76-100%	0	0	0	2
	51-75%	2	0	5	0
	26-50%	0	3	0	0
	0-25%	1	12	0	0

Risk	Likeliness	Impact	Mitigation Strategies
Feature development time exceeds projected timeline	.9	Critical	We have built room for delays into our schedule
App to car communication - Takes too long - Out of Range	.3	Low	Analog kill switch, program that kills the car when it detects a disconnect
Vehicle hardware failure - Delay to hardware development - Lack of platform to develop software on	.2	Moderate	Purchase quality products from reputable suppliers Test product and fix any problems.
Computer Vision reading inaccurate	.7	Moderate	Test computer vision before deploying to RC car

-Potential of Rabbit crashing -Timeline gets pushed back			

2.6 Personal Effort requirements

Task	Time Needed	Individual Contribution
Hardware Setup	12 hours	X3 people
MC Connection	5 hours	X2 people
MC Connect to Rabbit	5 hours	X2 people
Computer Vision	48 hours	X4 people
App(basic UX)	8 hours	X2 people
App integration to Rabbit	4 hours	X4 people
Computer Vision Integration to Rabbit	24 hours	X7 people
Additional Features	Infinity	All

2.7 OTHER RESOURCE REQUIREMENTS

Item	Cost	Link	Description
Track Car	$340	https://traxxas.com/products/models/electric/rustler-4x4?t=features	This RC car is capable of meeting human running speeds and provides a large enough platform for attaching microcontrollers and their necessary sensors.
Camera	$25	https://www.sparkfun.com/products/14028?src=raspberrypi	This is a high resolution camera that should be sufficient for detecting lane lines and can easily be connected to a dedicated camera bus on the Jetson Nano.

Car-Control Microcontroller	$30	https://www.adafruit.com/product/2995	The Feather M0 Bluefruit has a built-in bluetooth LE microprocessor for handling app-to-microcontroller communication. It also is faster than a standard Arduino Uno but can be easily programmed on the same platform.
Computer vision microcontroller	$60	https://www.sparkfun.com/products/17244	Jetson Nanos are power efficient microcomputers capable of computer vision processing with CUDA acceleration.
Sensors - GPS module - IMU - PWM Driver - Hall effect sensor	Price - $26 - $15 - $10 - $2	https://www.seeedstudio.com/Grove-GPS-Module.html https://www.adafruit.com/product/3463 https://www.adafruit.com/product/2928 https://www.adafruit.com/product/158	Various sensors for distance tracking, pwm signal generation, and acceleration tracking.
Vehicle battery & charger	$90	https://traxxas.com/products/parts/batteries/idpowercellbatteries/lipo/2869X-7600mah-74v-2S-25C	Self-explanatory, standard Traxxas battery and charger
Microcontroller batteries			We want to move away from microcontroller batteries, and instead use the same battery from the car to power all systems (currently not set up)