2 Project Plan

2.1 *Project Management/Tracking Procedures*

Which of agile, waterfall or waterfall+agile project management style are you adopting. Justify it with respect to the project goals.

Agile + a little bit of waterfall. We plan on trying to work on code in parallel, however due to the fact that some code will depend on others, we will need to waterfall into our end goal

What will your group use to track progress throughout the course of this and the next semester. This could include Git, Github, Trello, Slack or any other tools helpful in project management.

Git, Slack, discord.

2.2 Task Decomposition

In order to solve the problem at hand, it helps to decompose it into multiple tasks and subtasks and to understand interdependence among tasks. This step might be useful even if you adopt agile methodology. If you are agile, you can also provide a linear progression of completed requirements aligned with your sprints for the entire project.

- **Learn the skills and concepts we need for this project**
  - **Finish Coursera modules**
  - **Finish AWS module**
- **Create Algorithm and Simulation testing**
  - **Create code for virtual simulation**
  - **Develop model for testing**
  - **Test in virtual simulation**
  - **Debug/revise code to improve performance**
  - **Tune any PID feedback loops**
- **Test physical car**
  - **Build track**
  - **Put together deep racer with sensors**
  - **Test on track**
- **Modify sensors/car for Embedded Systems lab**
  - **Change sensors to fit Embedded Systems lab goals**
  - **Develop rough code for 288 lab to use as test/example**

2.3 *Project Proposed Milestones, Metrics, and Evaluation Criteria*

What are some key milestones in your proposed project? It may be helpful to develop these milestones for each task and subtask from 2.2. How do you measure progress on a given task? These metrics,

preferably quantifiable, should be developed for each task. The milestones should be stated in terms of these metrics: Machine learning algorithm XYZ will classify with 80% accuracy; the pattern recognition logic on FPGA will recognize a pattern every 1 ms (at 1K patterns/sec throughput). ML accuracy target might go up to 90% from 80%.

In an agile development process, these milestones can be refined with successive iterations/sprints (perhaps a subset of your requirements applicable to those sprint).

Milestones:

- Learn Skills and concepts
    - Fully complete all work and videos for the Coursera Introduction to Embedded Machine Learning course (12 hours/person total)
    - Fully complete all work and videos for Amazons DeepRacer AWS training (10 hours/person total)
- Begin Work on Algorithm
    - Develop all base functions predicted to be required for simulation (Hardware controls, controls/PID logic, etc.) by 20 group hours
    - Create first trained model by 40 hours
    - Have simulation tested and any visual issues analyzed by 45 hours
    - Write up first debug report and changes to be implemented by hour 45
    - Have car complete a full lap without going out of bounds virtually by hour 60
    - Get time for lap down to 1:30 by hour 75
    - Have function ready to put into physical car by hour 80
- Test Physical Car
    - Get confirmation on track location by hour 5
    - Build track by hour 10
    - Have car complete full lap by hour 15
- Modify Sensors/car
    - Research sensors and choose which would benefit both teaching students and the performance of the car by hour 5
    - Develop interfacing code for sensors by hour 20
    - Complete a full lap with new sensors by hour 30

2.4 PROJECT TIMELINE/SCHEDULE

• A realistic, well-planned schedule is an essential component of every well-planned project

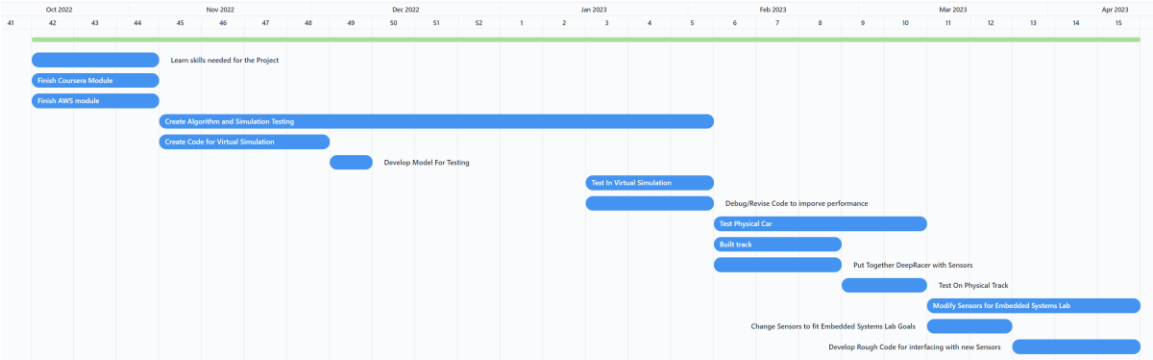
• Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity

• A detailed schedule is needed as a part of the plan:

– Start with a Gantt chart showing the tasks (that you developed in 2.2) and associated subtasks versus the proposed project calendar (including both 491 and 492 semesters). The Gantt chart shall be referenced and summarized in the text.

– Annotate the Gantt chart with when each project deliverable will be delivered

• Project schedule/Gantt chart can be adapted to Agile or Waterfall development model. For agile, a sprint schedule with specific technical milestones/requirements/targets will work.



2.5 Risks And Risk Management/Mitigation

Consider for each task what risks exist (certain performance target may not be met; certain tool may not work as expected) and assign an educated guess of probability for that risk. For any risk factor with a probability exceeding 0.5, develop a risk mitigation plan. Can you eliminate that task and add another task or set of tasks that might cost more? Can you buy something off-the-shelf from the market to achieve that functionality? Can you try an alternative tool, technology, algorithm, or board?

Agile project can associate risks and risk mitigation with each sprint. The main risk of this project is not getting any of the deliverables done, which should not happen.

- **5% chance, can be mitigated by following milestones closely and keeping team accountable**

Possibility that sensors/robot may not be as useful as ones already used in 288 to teach development techniques.

- **3% chance, can be mitigated by researching sensors in advance of testing them**

Possibility that we might not be able to successfully add sensors to the DeepRacer bot

- **20% chance, will still be able to fall back on default sensors**

Possibility that the robot breaks while testing or sensors become damaged.

- **10% chance, will lose time dealing with Amazon returns/Customer support**

2.6 Personnel Effort Requirements

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be the projected effort in total number of person-hours required to perform the task.

Task	*Estimated hours*	*Explanation*
Initial Learning	*22*4 = 88*	*Coursera Course has about 12 hours of work listed and AWS course has about 10 hours listed*
Algorithm & Virtual Testing	*20*4 = 80*	*This process will involve writing the machine learning algorithm and testing it virtually via a simulation provided by AWS. It will also involve debugging and making the algorithm more efficient. We believe each team member will spend about 20 hours completing this task.*
Physical Testing	*20*	*Once the bot has been virtually tested, everything should transfer over to the physical model, leaving little or no issues to be resolved in Physical testing. We believe all members will spend about 20 hours in total completing the testing since there should be no issues.*
Modify Sensors/Car	*35*	*For this task we will have to do all our own research on sensors and the process of adding them to the DeepRacer car as well as developing code to in the data. Once the sensors are set up we will need to modify our previous code to account for the new sensors.*

2.7 Other Resource Requirements

Identify the other resources aside from financial (such as parts and materials) required to complete the project.

Different sensors for additional testing with outside modules

DeepRacer robot + Sensors (Possibly the official kit and some extra sensors like on the CyBots)

Access to paid AWS model training

Supplies for track to test (tape, large open space, obstacles such as pipes and rocks)