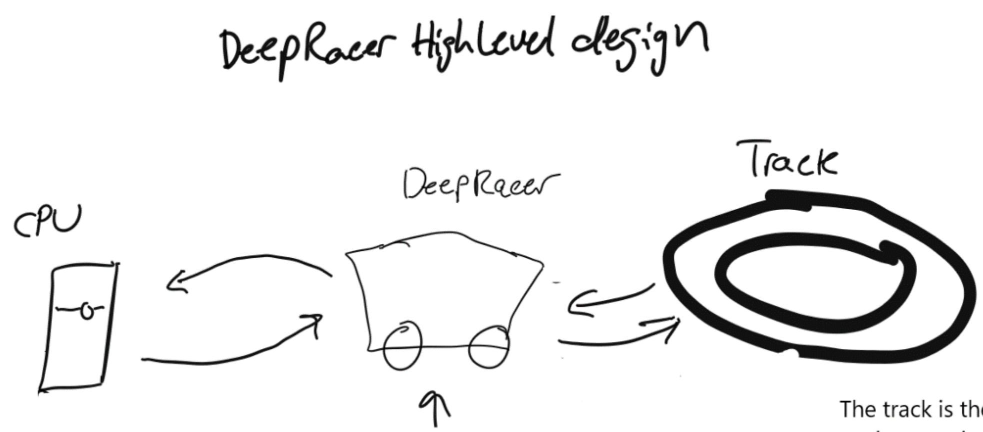
4.3 Proposed Design

4.3.1 Overview

Our goal is to design and implement a machine learning algorithm for the AWS DeepRacer robot. The goal will be for us to have a design that will be able to compete in the AWS DeepRacer League. The DeepRacer runs on a 1/16th scale version of a real racetrack and can be simulated using AWS software or ran on a real track. Another purpose of this project is to figure out how we can apply machine learning for embedded systems to CPR E 288 at here in the ECPE department. By learning about machine learning through a ready-made interface like the DeepRacer we can hope to figure out how to apply it to the CyBots used in class. This should be possible because almost any microprocessor is capable of machine learning. This is because we can use almost any microcontroller to collect data via sensors and send it to a over a network to do the real calculations. After being processed, the computer can give updated data back to the microcontroller in order to make decisions later. This design is expressed in the diagram below. We also hope to start a club at ISU for the DeepRacer and give undergraduates a chance to learn about machine learning and share knowledge with other students.



The CPU will collect data from the DeepRacer and update the AI based on that data via our algorithm. The CPU will either be an AWS server or a decently powerful computer on the ISU network.

DeepRacer collects data on the track and it's preformance via sensors (gyro, accel, camera) and then sends it back to the main processing unit. Based on this, it may change how it runs the track in order to achieve faster times or stay on the road more accurately.

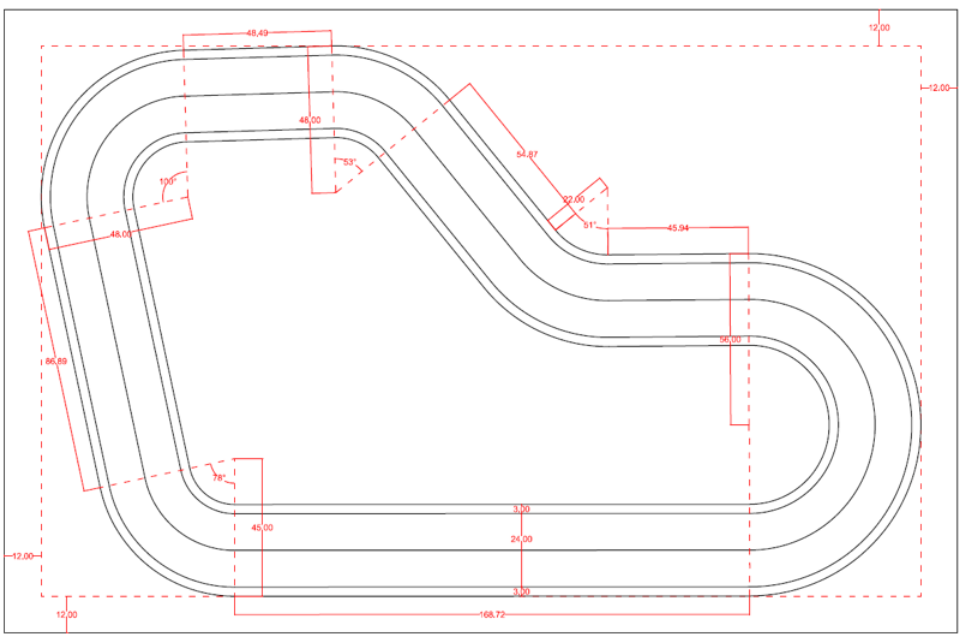
The track is the enviorment in which the DeepRacer collects data and made descisions based on it's algorithm.

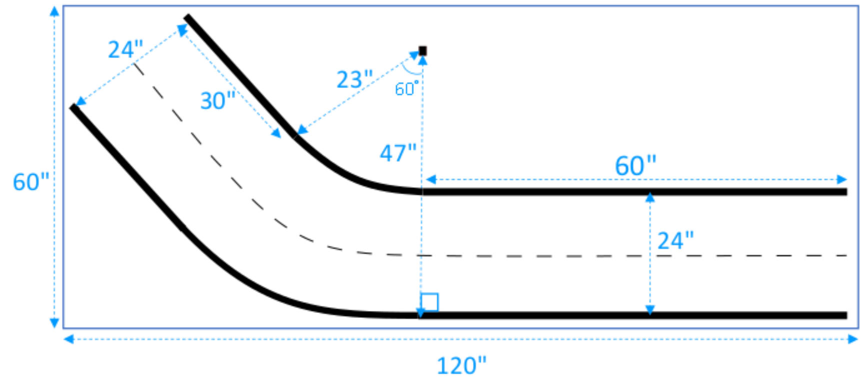
4.3.2 Detailed Design and Visual(s)

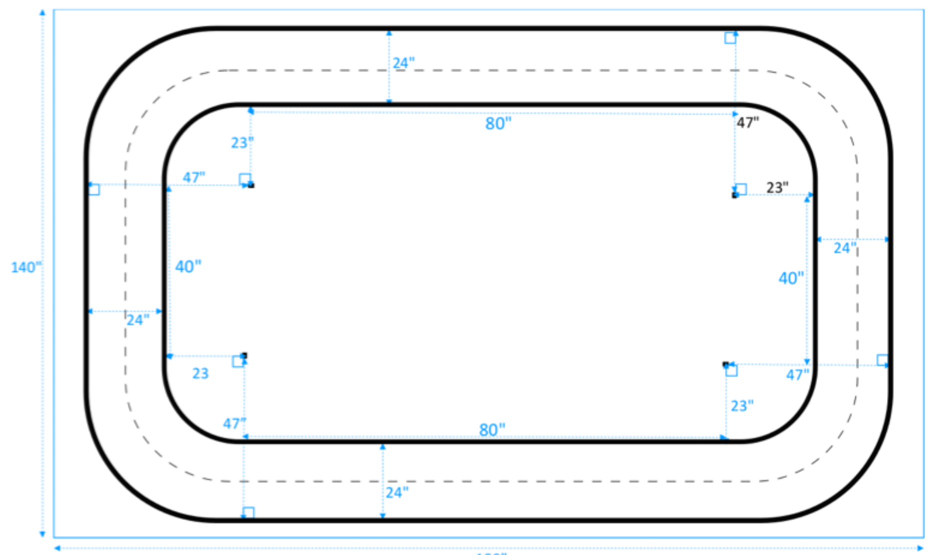
This project will involve a lot of learning as we go, and it is also a software-based project. Because of this, there will not be many physical designs and we do not have a top-level idea of any software yet. We can, however, talk about some platforms and hardware that we plan on using.

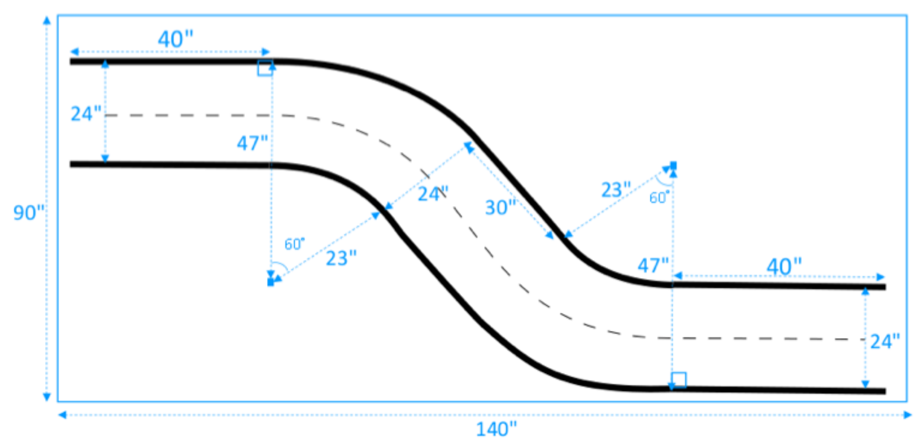
For this project we will utilize DeepRacer hardware manufactured by Amazon. It has the motors and sensors (gyroscope, accelerometer, camera) needed to traverse the track and an Intel processor to process this data and send it to the main processing unit.

We will also need to use a 1/16th scale indoor racetrack. The racetrack can be any material if there is a contrast between the material for the track and the white lines used to represent the bounds. Amazon provides templates for the track on their DeepRacer resources website.









A track will probably be made from a black or green vinyl material and be trinted on with white for the track lines.

Along with this we plan on using the AWS DeepRacer simulator for testing as well as the AWS service itself for cloud computing power. There is a chance that we will acquire a desktop computer with a decent GPU in order to do calculations ourselves since AWS only gives out about 10 hours of computing time for free.

4.3.3 Functionality

AWS DeepRacer League/Club: User should be able to join and meet people interested in Machine Learning, gain real-world experience via the physical DeepRacer robot, train the robot physically or through simulation, learn machine learning algorithms to increase bots' performance, compete in competitions virtually and/or physically.

CprE Courses: AWS DeepRacer should be able to be used in CprE courses as an educational tool to incorporate Machine Learning into projects and/or the curriculum. The bot should be able to handle and adapt to the needs of the class (i.e. Add sensors).

4.3.4 Areas of Concern and Development

The current design should be able to compete in competitions, and the outcome of our research on the AWS DeepRacer League should be able to lead to the proposal of a club. However, we will need to obtain a physical version of the DeepRacer bot before we can try testing the functionality of adding sensors not provided by Amazon and if that is an achievable goal. Our immediate plans to develop a solution to this concern is to get a physical DeepRacer as soon as possible to begin testing.

4.4 Technology Considerations

A few technological considerations we must keep in mind is that we are working in real-time when we are looking at this project from a practical standpoint and we are looking at a potentially expensive project in the sense of scalability unless we implement our own software to train our models. Since we are dealing with four different areas, we will divide these up below for organization purposes:

- **Machine Learning**: On the machine learning portion, there are two ways we can store/train the model with which we are developing. We can use the supported AWS channel or move to train our models locally. Both have trade-offs not in the technological sense, but in the time and money sense as seen below.
    - AWS:
        - Strengths: ease of access, no need to maintain infrastructure for the model, saves time, scalability.
        - Weaknesses: pricing and storage
    - Locally Trained:
        - Strengths: budget friendly, easy to transition, could be easy to maintain as the hardware is local.
        - Weaknesses: setting up, maintaining infrastructure, time consuming, would need more machines to scale the project.

- **Embedded Systems**: On the embedded systems front, the issues we face are more through limitations of the actual hardware and the format of which we write our software:
    - Strengths: Could be efficient, we don't have to implement and design our own sensors.
    - Weaknesses: Weaker hardware onboard of the DeepRacer means we will have to be mindful of our software. With a real-time OS that the DeepRacer uses, we will also have to ensure that our machine is predictable with a few guarantees.
- **Software**: With software, we fall into the domain of human errors.
    - Strengths: Simple, quick functions using an embedded systems model (software won't be too big).
    - Weaknesses: Software is prone to human error as we are using a lower-level language than in other applications (C language).
- **Education**: See next section.

4.5 Design Analysis

Overall, our group has been in a research and development cycle, in which we have been collecting more and more information about not only machine learning – but developing the educational tool that we are hoping to achieve here. Lately, we're beginning to transition from the pre-work portion of our project to begin testing a few of the ideas we developed earlier on. Whether it be methods of avoiding costs through AWS, finding easier ways to duplicate this project at a scale large enough for a course, and of course the software side of things. Due to this, we have not built nor implemented – but we have put a few of our ideals to test in these scenarios, expanding into a concept rather than just having the DeepRacer and its track, but being a fully fleshed out idea in which students can come in and learn about the machine learning, the predictable AI, and the processes used in the embedded machine portion of the vehicle.

Since we first dove into this project, we were ultimately looking at it from two different angles, but as mentioned – as we begin to transition into this implementation phase, we've seen four different angles truly develop and poses the uniqueness to our project. We aren't building a vehicle; the vehicle is already built for us. So, then you ask, well what are you building – and we are designing a tool, a product for many users to enjoy and learn with the purpose of education. The next angle we have seen develop within the creases is the efficiency and the cost of the project – is it realist for Iowa State or any other organization to shell out a lot of money to AWS to run this project? No, if this were a student organization, we would be forced to reduce costs – therefore we must develop a way to run this project on a small scale, single computer to allow for low-cost training opportunities. In this respect, our 4-3 design does not mesh well as we initially were planning to go all AWS – however as we have moved away from the idea that "our software is the product" to "our experience is our product" it began shaping the last element we wanted to highlight. The educational grounds, the where is this going to go, how is this going to shape the curriculum possibly, why are professors so interested in seeing what we develop, etc. And the possibilities in this extent are endless, for us as undergraduate students we really do not have the possibility to experiment with machine learning in a feasible way much like we did with embedded systems early on – this implements an extra layer that we have come to find a lot of professors and other organizations are interested to see how scaled down and portable this experience can become.

Since our shift, we found that a lot of our implementation approaches could potentially work, which is great – however in this section we wanted to highlight the importance of this new direction we have discovered as we began to test in a virtual environment with resources available to us. While its true, the vehicle is built and the plans we have for the software are simplistic to the core with material we learned in our embedded systems classes, we have found that the extra layer is having us question our designs from the practical sense of the standard user. We must make the software and the hardware interface with people who may not have our skills in terms of technological literacy, to shift to this perspective and to stay on target – we are beginning to assign roles within our team. As one of our team members mentioned, we aren't in the engineering business – we are in the human business.