1.1 Requirements & Constraints

List all requirements for your project. Separate your requirements by type, which may include functional requirements (specification), resource requirements, physical requirements, aesthetic requirements, user experiential requirements, economic/market requirements, environmental requirements, UI requirements, and any others relevant to your project. When a requirement is also a quantitative constraint, either separate it into a list of constraints, or annotate at the end of requirement as "**(constraint)**." Ensure your requirements are realistic, specific, reflective or in support of user needs, and comprehensive.

Requirements:

- Physical:
  - Track must be compact to preserve space as well as be able to be stored and rebuilt while still being within the specified width and length requirements. (**Constraint)**
  - Additional features to the car must be supported by the AWS DeepRacer framework and licensed by Amazon. (**Constraint)**
- Resource:
  - Software shall be stored in a method that is easy to access and to update efficiently, with version control.
  - The entire process must be well documented, with the added ability to easily append updates to the design for future use. This means high-level and low-level documentation as well as commented, readable code.
  - Implementation must be lightweight, to run on the DeepRacer hardware (embedded machine learning brings more constraints than regular machine learning that might have access to extremely powerful computers.)
    - Lightweight in terms of ensuring the OS meets its scheduling requirements effectively and predictably. (**Constraint**)
- Functional:
  - Functional algorithm that will teach a robot car to run a track.
  - Ability to work in the simulated environment and on a physical track with DeepRacer robot.
  - Must be able to complete a full lap on a track in the ASW DeepRacer League
- Aesthetic:
  - The DeepRacer can be modified to include additional sensors, however – it must not compromise the design in a method that interferes with the overall performance of the vehicle.

Additional Constraints:

- Documentation must be stored in a method that can be accessed by future students.

1.2 ENGINEERING STANDARDS

What Engineering standards are likely to apply to your project? Some standards might be built into your requirements (Use 802.11 ac Wi-Fi standard) and many others might fall out of design. For each standard listed, also provide a brief justification.

https://standards.ieee.org/ieee/2830/10231/

- With a potential fleet of multiple cars in the educational environment, this would effectively fall into the domain into shared machine learning. As a result, we will need to comply with the IEEE standard.

IEEE SA - IEEE 2050-2018:

- With our project in the domain of a small-scale real-time embedded system with multiple sensors, we will also follow the IEEE standard to ensure that our device behaves properly in unforeseen circumstances – following the ideals of real-time operating systems and scheduling.

WiFi: IEEE 802.11 Wireless LAN | part of Broadband Access: Wireline and Wireless - Alternatives for Internet Services | Wiley Telecom books | IEEE Xplore

- Our bots will also use WI-FI to communicate with our server and to store data. As a result, we will follow the IEEE standard for implementing WI-FI effectively.

Using a coding standard to improve program quality | IEEE Conference Publication | IEEE Xplore

- As code within a group can become very messy, we have decided to follow the process in this document to ensure that our code conforms to a basic standard to ensure readability and reliability.

AWS_DeepRacer_League_2022_Official_Rules_3_1.pdf (awsstatic.com)

- As we plan on racing in the DeepRacer League, we must conform to their rules and specifications for how to design and race our robot, as well as define what our practice track design looks like