**Monday, August 21, 2023**

Today I learned about how Senior Research is going to work in general (especially in regard to the weekly journals) and what project ideas do and do not turn into good projects. Better projects have a specific metric for success rather than a qualitative assessment.

**Wednesday, August 23, 2023**

Today, I researched the project I plan to do so I could determine possible approaches for solving it. I want to do a project related to multi-agent car racing, with the constraints of having each agent use the policy learned or created, and also the addition of obstacles onto the track which has not been very common in past research. I also plan on creating a demonstration incorporating my algorithm in which a car follows a walking person while avoiding obstacles. Previous research incorporates [Monte Carlo tree search](https://github.com/travelbureau/f0_icml_code), structuring the problem of path planning to a game tree where each leaf node is a valid trajectory that could be reached from the parent node. Game-theory based approaches also seem to exist, such as [this paper](https://arxiv.org/pdf/2209.07758.pdf) which used regret minimization for planning overtake maneuvers. There are also several implementations of reinforcement learning for the robotic cars I plan on using, including [this](https://github.com/navidmdn/f1tenth_rl) and [this](https://github.com/abhaybd/F1-RL) which provide simple examples of single-agent, no-obstacle driving. Previous approaches which I am already familiar with (researched previously) to solving this simplified self-driving problem include [end-to-end machine learning with CNNs](https://arxiv.org/abs/1604.07316), purely algorithmic algorithms involving the iterative closest point algorithm to generate maps and race on a [pre-computed racelines](https://github.com/f1tenth/ESweek2021_educationclassA3/tree/main/02_Pure_Pursuit), and “reactive” approaches which hand-craft steering angles based on analyzing the LiDAR data. End-to-end models normally take images of the driving surface (road) as input and output a steering and throttle value normalized to (-1, 1). I have used this method in the past, but it requires manually driving the car around to create a dataset and also is not very explainable or robust. Raceline methods are the opposite – once a track is mapped out and a trajectory generated, there are control strategies which can make a car follow this set of waypoints perfectly. The problem with this, of course, is the amount of information required and the fixed nature of the trajectory. A major concern I have with my project is what to use as input: monocular/stereo cameras, lidar or a combination of both could be used. This would change the kind of algorithm I would use as well as the simulation environment and hardware on the real car. For the person following demo I want to do, vision would have to be incorporated but only to track the target. Considering hardware, it seems there are two somewhat complete cars in the SysLab which are in varying levels of disrepair. I plan to take the spinning LiDAR off of one and put it on the other car with the Nvidia TX1 board already on it. I’m not sure how functional the ESC/motor/batteries are, so I will have to figure that out to do any real demos on actual cars (since only simulations are not very interesting).