Week 1

- Progress
  - First Senior Project meeting
- Plan
  - Meet with team next week
- Risks
  - Covid
- Needs
  - None

Week 2

- Progress
    - Met with faculty coach
    - Sent request to Kurt for our nitron account
    - Reached out to the project sponsor. Meeting Scheduled for next week
    - Team name chosen
- Plan
    - Meet with product owner
    - Discuss requirements and domain of project
    - Do research on path planning algorithms
    - Converse with the product owner on the research that was done
- Risks
    - Team hasn't grasped what the project entails.
    - Team members have no prior experience with path planners or machine learning
- Needs
    - To meet with project sponsor to be able to ask questions

Week 3

- Progress
    - Met with faculty coach and product owner
    - Did research into MissionPlanner and QGroundControl
    - Investigated website space
    - Trello board created for task tracking
- Plan
    - Get a running simulation in one of the path planning algorithms
    - Choose a methodology to follow
    - Have the website alive with some barebones something on it
- Risks
    - Simulation could be hard to even get running. Team is still somewhat unfamiliar with the software
    - Unknown difficulty on adding some sort of variance or chaos engineering into simulation.
- Needs
    - Knowledge and understanding in the existing path planning solutions

Week 4

- Progress
  - Website is up! Not finished, but it’s usable
  - Project synopsis approved
  - Cursory domain model completed
  - Simulations have been tested and explored
- Plan
  - Look into flight stack codebases to better understand the scope
  - Definitively deciding of a flight stack to continue with
- Risks
  - Poor documentation makes research more difficult
- Needs
  - More information about the flight stacks

Week 5

- Progress
    - Distributing rewards can potentially exist in the form of mission commands
    - Python scripts work! They are very robust and we can do quite a bit with them
    - Everything in mission planner is run through the scheduler, which is in charge of decision for the plane drone, and there is more research to be done here
    - Definitely chosen MissionPlanner
- Plan
    - Continue research on the fronts of rewards, scripts, and decision making
- Risks
    - Mission commands end up being useless
    - Scripts are restricted into the areas they are able to manipulate
    - Implementing the sending of mavlink messages will be easy, but having the code to receive the messages may be difficult.
    - Building an environment to get source code running may be difficult
- Needs
    - More time would be really nice

Week 6

- Progress
    - We have a more robust chaos script that introduces random variances to the simulation
    - We have a working build of ardupilot
- Plan
    - We have a plan to implement rewards and penalties for different sets of coordinates
    - We're planning on using our own simulation mode so that we can change variables as we see fit
- Risks
    - Implementing penalties and rewards may be very heavy and prove much more difficult than intended.
    - Working around the no flight zones may not have an easy alternative
    - Building dev environments may take more time than we would like.
- Needs
    - A better understanding of the relationship between ardupilot and MissionPlanner, because this is relatively poorly documented.

Week 7

- Progress
    - We've been working on getting the dev environment setup, and we have only one developer's running, but hopefully that will allow all of us to get setup.
    - Chaos engineering is now transferred to work with copters.
- Plan
    - We have a plan to implement rewards and penalties for different sets of coordinates
    - We're planning on using our own simulation mode so that we can change variables as we see fit
    - This is a similar plan to last week, because we didn't estimate our goals from last week properly.
- Risks
    - Implementing penalties and rewards may be very heavy and prove much more difficult than intended.
    - The dev environment has the potential to just stop working again, and be very finicky.
- Needs
    - A better understanding of the relationship between ardupilot and MissionPlanner and MAVProxy and SITL, because this is relatively poorly documented.
    - A better understanding of how geofencing works within ardupilot

Week 8

- Progress
  - We have a flight mode!
  - We have geofencing!
  - We have working development environments
- Plan
  - Implement mission commands via mavgen
  - Implement rewards through reading mission commands in a flight mode
  - Attempt to break linearity of waypoints, which would allow us to view each waypoint and figure out which way to go next.
  - Update website
  - Create a plan for deliverables through the end of semester
- Risks
  - Implementing penalties and rewards may be very heavy and prove much more difficult than intended. This has been slightly mitigated through our previous understandings of how we can implement this.
  - Linearity of missions may not be possible to be broken, and may be very heavy to implement.
  - Development in C++ may prove a large learning curve for the team.
  - Development in XML may prove a large learning curve for the team.
- Needs
  - A Better understanding of how already implemented flight modes work (specifically auto mode)
  - Better understanding of how geofencing works within ardupilot