**Senior Project Proposal**

Michaela McLeod

Paul Sampson

Autonomous Color-Sensing Drone

1. **Objective**

A quadcopter will be fitted with various sensors which communicate wirelessly with a controller on the ground. The quadcopter will fly over a mat made of a patchwork of many different colored squares. Inputting a color on the controller will cause the quadcopter to fly in a search pattern until it finds the relevant square, at which point it will land. The quadcopter can then be instructed to lift off and land on another square, all autonomously. It will also automatically backtrack if it leaves the mat accidentally.

1. **Technologies**

An infrared proximity sensor and a color sensor with a lens will be linked to an Arduino Nano with a radio transmitter. Another Arduino on the ground will be linked to another radio transmitter, along with an array of servos attached to the original remote control. The proximity and color data will be sent to the controller Arduino, which will compare them to both its internal map of the mat and the inputted color. It will use the servos to manipulate the remote and move the quadcopter until a certain condition is met (the outer bounds of the map are passed, the correct color is found, etc.) at which point it will react. The quadcopter will be able to take off and land on its own, as well as detect which square it is on.

1. **What I Want to Learn**

I am excited for our senior project to learn more about robots. I have only spent a little bit of time working with robots while volunteering at Coder Dojo, but I find them very interesting. I also have never used Arduinos. I am excited and ready for these challenges!

1. **Point Scale**

A list of proposed features: 154 points

Detection

* Detect quadcopter height- 3
* Detect color of square below quadcopter- 10
	+ Detect color at close range- 3
	+ Detect color at long range- 7
* Compare detected color to inputted color- 2
* Reorient after flying off mat- 18
	+ Detect square with out-of-bounds color- 3
	+ Retrace last action- 10
	+ After recovery, find current position based on color- 5
* React to low battery- 13
	+ Set upper limit for reasonable throttle levels- 7
	+ Detect when this limit is exceeded- 5

Communication

* Transmit height from drone to controller- 5
* Transmit color from drone to controller- 5

Control

* Move drone horizontally- 16
	+ Left, right, forward, back- 4 for each
* Takeoff- 8
* Fly in grid pattern- 20
* Land- 16
	+ Maintain position laterally after detection of correct color- 10
	+ Decrease thrust until on ground without bouncing or damaging drone- 6
* Take shortest route to inputted color based on internal map- 13
	+ Calculate shortest route dynamically based on current position- 3
	+ Control quadcopter to follow route- 5
	+ React to current square to update route- 5
* Change selected color through button input- 5
	+ Record button input- 2
	+ Signal new selection with LED- 3
* Auto-trim- 20
	+ Detect when trim is off- 10
	+ Auto-adjust until equilibrium- 10
* Calculate velocity - 25
	+ Detect velocities over a certain threshold - 15
	+ Implement counteraction movements for large velocities - 10
* Hover at a constant height – 5
* Making target and grid - 5

**Total = 126**

Michaela McLeod Grading Scale

A: 50+

B: 40-49

C: 30-39

D: 20-29

Paul Sampson Grading Scale

A: 50+

B: 40-49

C: 30-39

D: 20-29