Control Award Submission Form

Team # 5661	Team Name: Wolves Robotics 1

Autonomous objectives:
Positioning and orientating to identify the team prop through pre-defined spline paths. Following identification, positioning to place the purple pixel on the corresponding spike mark. Afterwards, following a pre-defined path to the corresponding backdrop section to place the golden pixel. Finally, parking off to the side of the board to give room for alliance member.

Sensors used:
The sensors equipped on our bot include the Logitech C920 Webcam, 2 GoBilda Odometry Pods, and the Rev Hub Internal IMU. The Logitech C920 Webcam is used to identify backdrop april tags. Lastly, both odometry pods and the IMU are used for localization and path following in autonomous.

Key algorithms:
An OpenCV pipeline is used to determine prop location by converting the camera input to YCrCb and comparing average color density in the respective prop regions to determine where the prop is by color. PID algorithm for controlling robot heading, path correction, and path following. Box collision detection by separate axis theorem for collision avoidance with backdrop. AprilTag detection for correcting robot localization in autonomous due to dead wheel inconsistencies.

Driver controlled enhancements:
Automatic robot orientation and automated retraction of linear slides when placing pixels on the backdrop for rapid placement. Collision avoidance with backdrop to avoid knocking pixels off. Different driving modes: "Driving" and "Placing," for more intuitive robot control and autonomous management of the linear slides and the claw.

Engineering portfolio references:
In the engineering portfolio, references are located under Software Processes where Autonomous and TeleOp are described more. Under the Autonomous section it describes the many options to choose from when detecting the prop are what the benefits of each are. OpenCV is the main utility we use to create a simple prop detection algorithm by comparing average color regions. In the Engineering Process section, it's also described how we had to control the robot arm given the design

Using OpenCV, once the robot determines where the prop is, move to either the "C" path for the center, "L" path for the left, or "R" path for the right.

Once the robot reaches the "P" point, it pushes out the pixel stored in the intake and proceeds to the board to place a pixel on it.