

Controller #1



On controller 1, we have all of the driving controls. We use the d-pad up for forward, d-pad down for backward, d-pad left for strafing left, and d-pad right for strafing right. We used the d-pad because it made more sense than the ABXY buttons because the d-pad has clear directions. Our methods are designed with

the intention to be reusable and so that the power can vary. We used this to create a turbo boost that can be triggered with A and Y because they were the next best option after the d-pad considering that our robot only goes forward, backward, right and left but nowhere in between so having 4 buttons makes more sense than using the joysticks. The final buttons used on controller 1 are the bumpers, we use these for turning because they are on both sides and don't feel like they are for strafing so they felt like the best choice. We were able to use some methods from last year that had already been written just by changing the power. This was very helpful for the start of the season testing.

Controller #2



On controller 2, we have controls for everything that scores points. We have the lifer/grabber and we have the servos which we can use to drag the foundation. The lifter is given a specified power for when the buttons are pressed but when nothing is being pressed it is always adjusting the power so that it stays in place and doesn't fall. When

the grabber is opened it is given a power and then turned off, but if the grabber is closed then the power is set until it is opened again so that it can grab the bricks. We experimented using encoders to open and close the grabber but there was the chance that the brick would stop it from ever reaching the goal and it would fall into an infinite loop so we decided against it. We also have controls for the servos on the back of our robot which are mainly just used for autonomous since we added a plate to the back of our robot because we can now push the foundation in endgame but we can also pull it if we need to.

The Code

There is a hierarchy in the code that makes it so that if two buttons are pressed, one is prioritized. We did this to keep the robot running. The order is: Forward, Backward, Left, Right, Turn Left, Turn Right. It is in that order because that is the order that we expect to be most helpful and the order in which we plan to use them. We felt the direction we would be moving the most is forward then back and so on.

We developed some methods for autonomous that used encoders to get to drive or strafe the position that we wanted. We were able to find the encoder values to get to these positions by writing the data to the text file while we drove the robot around in teleop and then just inputting them into the methods we wrote. Each method for driving checks to see if the front left motor has reached the target so all of the calculations were made for one wheel. This made the code much shorter but increased the margin of error because the wheels were not consistent with each other even at the same power, we also used the same methods that we used in teleop for giving the motors power because of how easy it was to add with little writing of new code. To get around this, we multiplied some of the powers by 0.9 in some directions so that each motor would have a closer output and we would have less drift from the robot overall. We also have two methods for grabbing the foundation with servos. They set the servos to go to the positions that we found to be down and up to grab and release the foundation respectively.