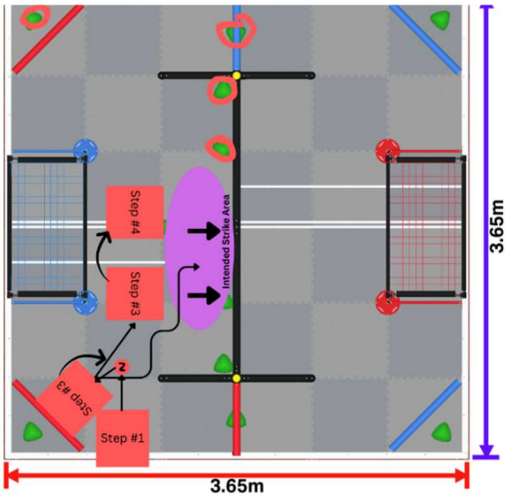
7481B Notes While Coding VR Skills

1. The arm was successfully able to drop down and spin out the triball.
2. The robot needs to move further to get to the Triball.
3. The arm can get the triball.
4. The robot needs to have a slight adjustment made to the distance it moves.
5. The robot can push a triball over the field.
6. The robot can now grab them consecutively, but I need to make it move less each time.
7. The code works. Now I must make the robot push all triballs over the long barrier.
8. The robot will not push far enough to get triballs over the long barrier.
9. The robot will now push some triballs over the long barrier. I am going to try to make all of them get in.
10. The robot gets four out of every five triballs in the scoring zone.
11. There is a one-second delay to make the robot put the triballs in a more precise position.
12. The one second is too long and we do not finish in time so it will be .75 seconds.
13. The new delay works, so I will now add a new section to the code to use the preload.
14. The preload code turns in the wrong direction.
15. The preload code turns in the right direction but does not drive far enough.
16. The preload code turns correctly and drives correctly, but the arm needs to be closer to the ground.
17. The arm will not move closer to the ground because of a limit, so I will make the intake push the triball further
18. The intake needs to push further
19. The strategy works
20. I forgot to make the robot drive back to the position it was coded to be in before starting the rest of the code, so I will make sure that the code has
21. The arm goes too far up and does not grab the triball, so it needs not to move up as much
22. The arm still moves up too high
23. The arm moves upward 25 degrees and can still pick up the triball, but it is either too high or not high enough
24. The arm was too high, as moving only 20 degrees fixed the issue
25. The robot is now going closer to the barrier to make sure that the triballs go over
26. The robot does go closer, but the robot needs to wait longer before going to the next triball
27. The robot needs to wait for 85 Msec, instead of 80 Msec
28. The robot needs to wait for 90 Msec, instead of 85 Msec
29. The robot needs to wait for 95 Msec, instead of 90 Msec
30. The robot has to wait one second now. I was hoping to save some time, but 1 second is necessary
31. The robot will now go forward for every triball, to have the highest chance of getting the triball to the other side of the field
32. This match, everything went right. The alliance triball was scored in the goal, and all triballs successfully went over the long barrier. I am submitting the final score of 25.

Those notes show how a coder thinks and feels while he or she is programming a robot. The entire process is hard, especially for the coder, which makes the satisfaction at the end even more enjoyable because you worked hard to get to where you are. My name is Clayton Sharp, and I Come from the team 7481B. I am the team's main programmer, as well as a designer, data analyst, strategist, and archivist. I did half of the programming on our team's robot and the striker virtual robot. I also had some help from Paul Kuruvilla, who is a strategist, Drive coach, Builder, Logistics, Game Manual Expert, Timekeeper, Researcher, Developer, and archivist. He helped make the pseudo-code (Left) and the picture to show the code on a field (right).

Psuedo-Code Page

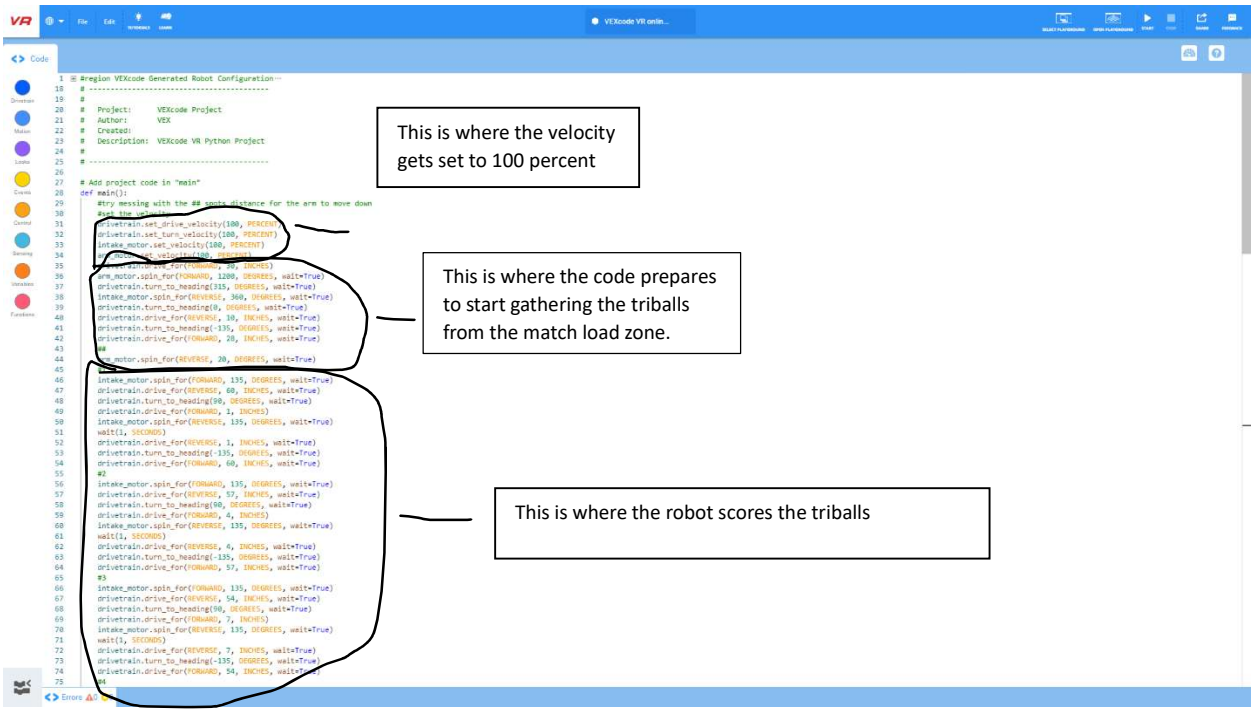
Step #1: Drive forward
Step #2: Pivot -135°, and drive forward to the middle of the Match Load Zone bar
Step #3 (CycleX10): Grab a Triball from the zone, turn -180°, strike it out, and then turn 180° back to start (don't do the last part on 10th Triball)
Step #4: Move over to the intended strike area and drive up the Triballs (to push them)
Step #5: Turn -90°, drive forward, turn 90°, and do the same thing
Step #6: If time allows, push over the circled Triballs into the Red Zone

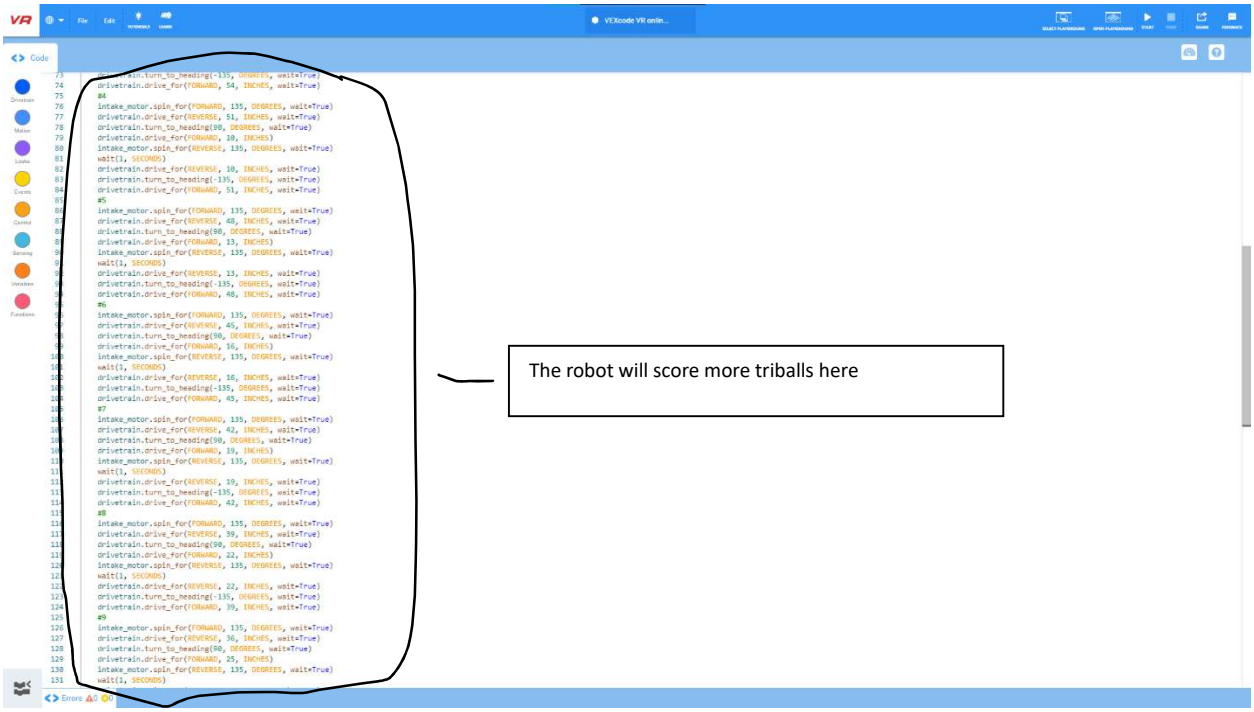


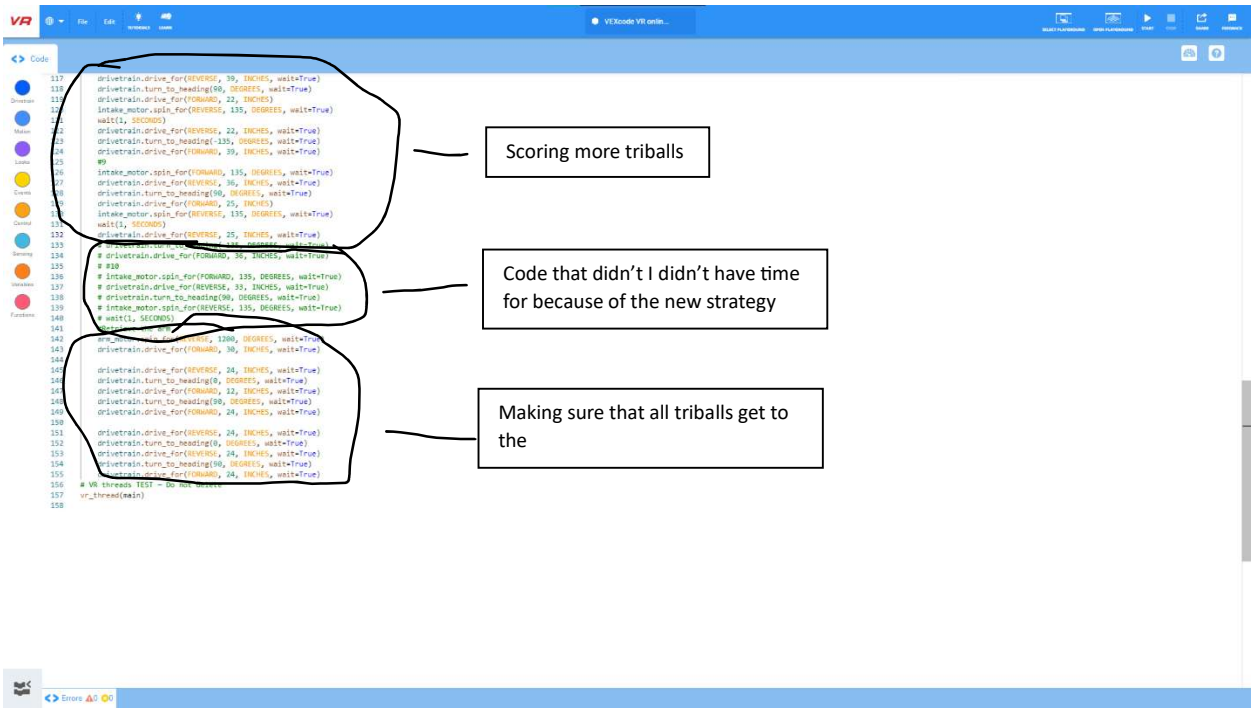
The code is not very complex. The only sensor it uses is the inertial sensor, which is used to make the robot turn more accurately. There are no variables used in the code, to prevent a slight bit of randomness, and no functions are used, because none of the actions can be repeated more than once, so I cannot use functions. The code starts by using a very simple command to set the velocity of every motor on the robot to 100 percent. Then the robot drives forward to the goal and deploys the arm. The intake on the arm then spins and releases the alliance triball into the goal, which scores 5 points, even though it is in the blue goal because it is an alliance triball. That makes it score for its color alliance, no matter which goal it is put in. That scores 5 points. Then the robot will drive over to the match load zone and lift the arm by 20 degrees, to allow the robot to go slightly further over the long barrier, which means that I can have the triballs go completely over the barrier. The robot then grabs the triball with the intake and drives to the long barrier, where it turns in the direction of the long barrier and lets go of the triball. It waits for 1 second after releasing the triball to allow it to move all the way over and then drives back to the match load zone. It will repeat this cycle 8 times and will score 16 points. Then on the last run, it starts normally, but before the robot returns to the match load zone, it will pull the arm back into the robot's starting position. Then it will push all triballs that didn't make it over the barrier to the other side. It pushes one area, moves to another area & pushes that, and then goes past the original area to push a final section of triballs. While doing that or the transporting of the triballs, another triball gets pushed over the barrier and scores 2 points. That all adds up to 25 points, because the first 8 scores 16 points, then the 9th scores 2 points, the stray one that gets pushed scores 2 points, and the alliance triball scores 5 points. The highest score that I know about for the VEXcode Vr online challenge or VEXcode VR in general is 97 points, and this is a pretty good score when compared to that, as some scores are less than 15 points. I started with a high score of 16 with a not-so-reliable code, got to 17 using a new strategy (putting the triball in the goal), and then got down to 7 with a bad strategy. Then on 1/30/24, I was able to get the 25 points I submitted. I used the free version of VEXcode VR in elementary school with the VEX IQ program and blocks to practice my coding skills in the ocean cleanup section. I am extremely grateful that VEXcode VR exists, and that the Python coding language was added to make coding more fun than just dragging blocks from a toolbox. It has helped accelerate my coding skills and made me a better coder, and that made my team like me as a coder.

If you want a little more about 7481B, read this paragraph. If not, skip to the next. 7481B is one of 4 teams, 7481A, 7481B, 7481C, and 7481D. The teams are Robot Republic, Redpandamonium, I.R.S., and Robomelons. They are all part of the 7481 or Metallicatz team. Metallicatz is from League City Intermediate (L.C.I.) in the Cleak Creek ISD school district, which is in the Houston metro area in Texas. We have gone to the league nights, Pasadena, Space City, and tussle by the tide competitions, and are about to go to the Region 3 state championship. All LCI teams have at the very least triple triple-qualified for state and are trying to get to worlds. We have received awards for Costume, Innovate, Notebook, Amaze, and Excellence. Some of these, like 2-3 times for Excellence, Metallicatz has gotten multiple times. Our amazing coach is Mrs. Rodgers, who teaches us the basics of robotics, and then only helps us if we need it from her, and teaches us how to handle that situation.

This next part is just there to show the picture version of what the code is doing, just in case you like visualls more than words.




drivetrain.turn_to_heading(-135, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 54, INCHES, wait=True)
#4
intake_motor.spin_for(FORWARD, 135, DEGREES, wait=True)
drivetrain.drive_for(REVERSE, 51, INCHES, wait=True)
drivetrain.turn_to_heading(90, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 10, INCHES)
intake_motor.spin_for(REVERSE, 135, DEGREES, wait=True)
wait(1, SECONDS)
drivetrain.drive_for(REVERSE, 10, INCHES, wait=True)
drivetrain.turn_to_heading(-135, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 51, INCHES, wait=True)
#5
intake_motor.spin_for(FORWARD, 135, DEGREES, wait=True)
drivetrain.drive_for(REVERSE, 48, INCHES, wait=True)
drivetrain.turn_to_heading(90, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 13, INCHES)
intake_motor.spin_for(REVERSE, 135, DEGREES, wait=True)
wait(1, SECONDS)
drivetrain.drive_for(REVERSE, 13, INCHES, wait=True)
drivetrain.turn_to_heading(-135, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 48, INCHES, wait=True)
#6
intake_motor.spin_for(FORWARD, 135, DEGREES, wait=True)
drivetrain.drive_for(REVERSE, 45, INCHES, wait=True)
drivetrain.turn_to_heading(90, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 16, INCHES)
intake_motor.spin_for(REVERSE, 135, DEGREES, wait=True)
wait(1, SECONDS)
drivetrain.drive_for(REVERSE, 16, INCHES, wait=True)
drivetrain.turn_to_heading(-135, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 45, INCHES, wait=True)
#7
intake_motor.spin_for(FORWARD, 135, DEGREES, wait=True)
drivetrain.drive_for(REVERSE, 42, INCHES, wait=True)
drivetrain.turn_to_heading(90, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 19, INCHES)
intake_motor.spin_for(REVERSE, 135, DEGREES, wait=True)
wait(1, SECONDS)
drivetrain.drive_for(REVERSE, 19, INCHES, wait=True)
drivetrain.turn_to_heading(-135, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 42, INCHES, wait=True)
#8
intake_motor.spin_for(FORWARD, 135, DEGREES, wait=True)
drivetrain.drive_for(REVERSE, 39, INCHES, wait=True)
drivetrain.turn_to_heading(90, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 22, INCHES)
intake_motor.spin_for(REVERSE, 135, DEGREES, wait=True)
wait(1, SECONDS)
drivetrain.drive_for(REVERSE, 22, INCHES, wait=True)
drivetrain.turn_to_heading(-135, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 39, INCHES, wait=True)
#9
intake_motor.spin_for(FORWARD, 135, DEGREES, wait=True)
drivetrain.drive_for(REVERSE, 36, INCHES, wait=True)
drivetrain.turn_to_heading(90, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 25, INCHES)
intake_motor.spin_for(REVERSE, 135, DEGREES, wait=True)
wait(1, SECONDS)
The robot will score more triballs here


drivetrain.drive_for(REVERSE, 30, INCHES, wait=True)
drivetrain.turn_to_heading(90, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 22, INCHES)
intake_motor.spin_for(REVERSE, 135, DEGREES, wait=True)
wait(1, SECONDS)
drivetrain.drive_for(REVERSE, 22, INCHES, wait=True)
drivetrain.turn_to_heading(-135, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 39, INCHES, wait=True)
#9
intake_motor.spin_for(FORWARD, 135, DEGREES, wait=True)
drivetrain.drive_for(REVERSE, 36, INCHES, wait=True)
drivetrain.turn_to_heading(90, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 25, INCHES)
intake_motor.spin_for(REVERSE, 135, DEGREES, wait=True)
wait(1, SECONDS)
drivetrain.drive_for(REVERSE, 25, INCHES, wait=True)
Scoring more triballs
drivetrain.drive_for(FORWARD, 36, INCHES, wait=True)
#10
intake_motor.spin_for(FORWARD, 135, DEGREES, wait=True)
drivetrain.drive_for(REVERSE, 33, INCHES, wait=True)
drivetrain.turn_to_heading(90, DEGREES, wait=True)
intake_motor.spin_for(REVERSE, 135, DEGREES, wait=True)
wait(1, SECONDS)
Code that didn't I didn't have time for because of the new strategy
drivetrain.drive_for(REVERSE, 24, INCHES, wait=True)
drivetrain.turn_to_heading(0, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 12, INCHES, wait=True)
drivetrain.turn_to_heading(90, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 24, INCHES, wait=True)
drivetrain.drive_for(REVERSE, 24, INCHES, wait=True)
drivetrain.turn_to_heading(0, DEGREES, wait=True)
drivetrain.drive_for(REVERSE, 24, INCHES, wait=True)
drivetrain.turn_to_heading(90, DEGREES, wait=True)
drivetrain.drive_for(FORWARD, 24, INCHES, wait=True)
Making sure that all triballs get to the