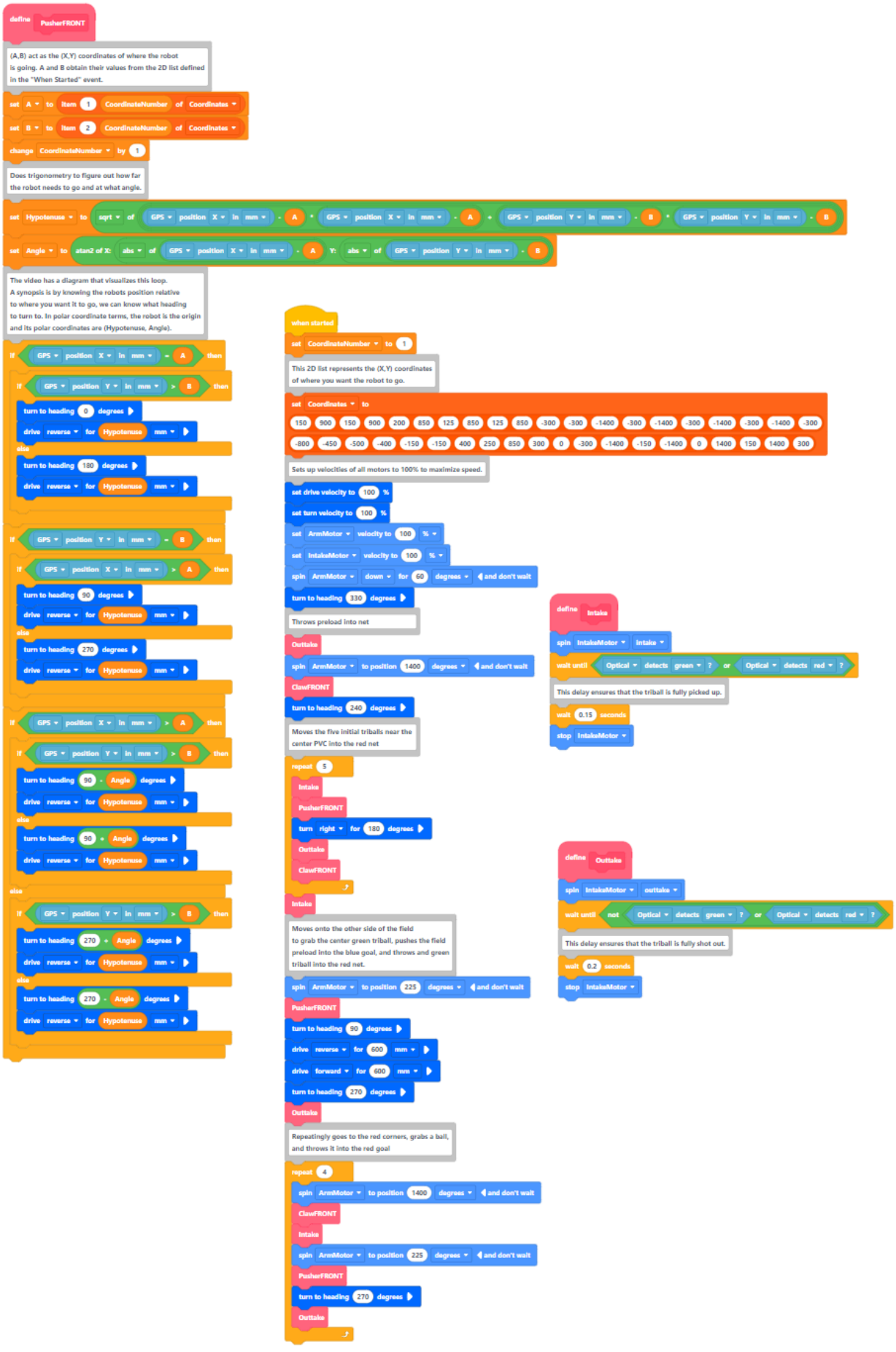
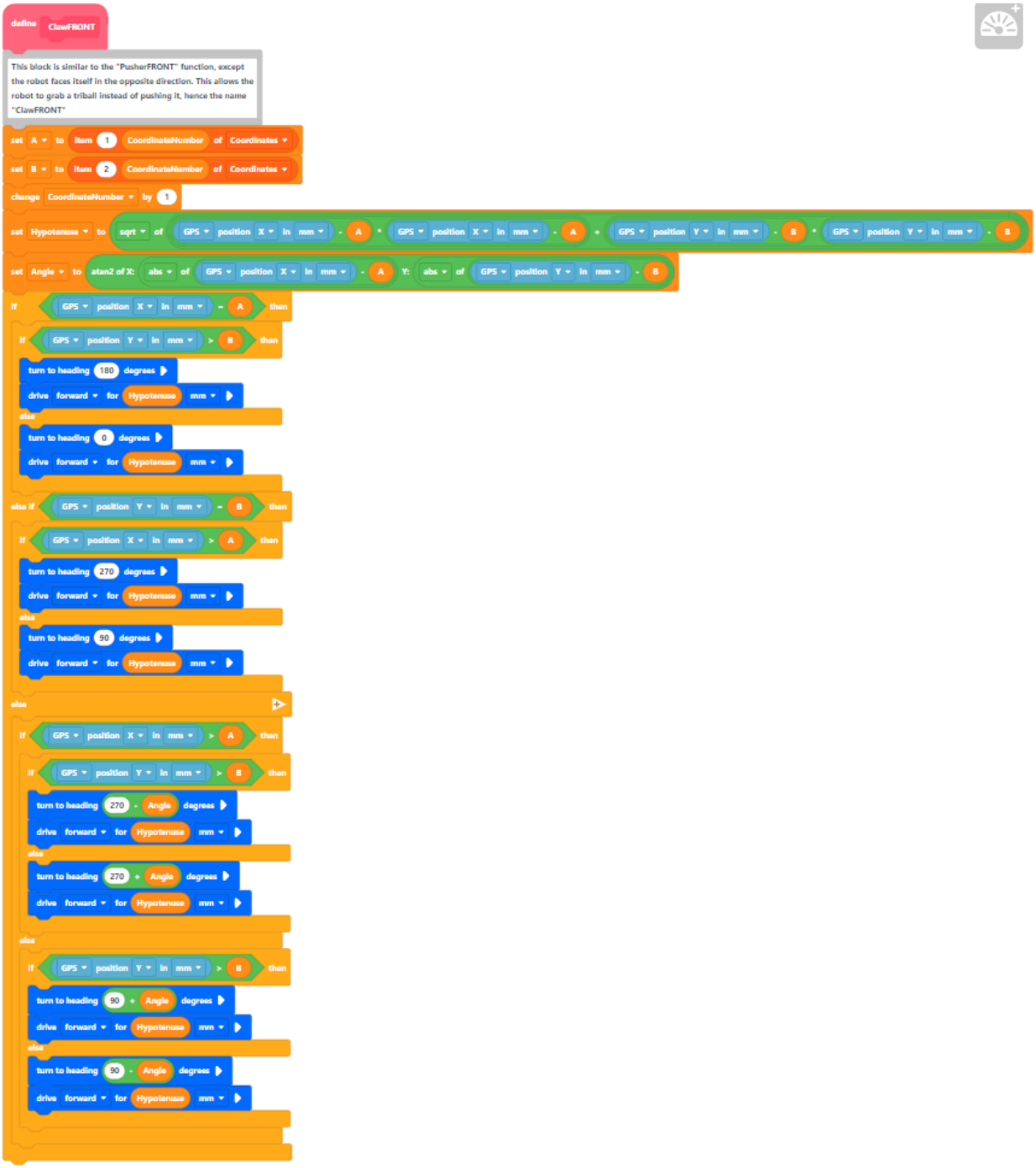
C3POE - 38535B - 2024
VEX VR Online Challenge

Jimmy Tsonis, Matthew Yam

Team # 38535B

Glenview, Illinois


define PusherFRONT
(A,B) act as the (X,Y) coordinates of where the robot is going. A and B obtain their values from the 2D list defined in the "When Started" event.
set A to item 1 CoordinateNumber of Coordinates
set B to item 2 CoordinateNumber of Coordinates
change CoordinateNumber by 1
Does trigonometry to figure out how far the robot needs to go and at what angle.
set Hypotenuse to sqrt of GPS position X in mm - A * GPS position X in mm - A + GPS position Y in mm - B * GPS position Y in mm - B
set Angle to atan2 of X: abs of GPS position X in mm - A Y: abs of GPS position Y in mm - B
The video has a diagram that visualizes this loop. A synopsis is by knowing the robots position relative to where you want it to go, we can know what heading to turn to. In polar coordinate terms, the robot is the origin and its polar coordinates are (Hypotenuse, Angle).
if GPS position X in mm < A then
if GPS position Y in mm > B then
turn to heading 0 degrees
drive reverse for Hypotenuse mm
else
turn to heading 180 degrees
drive reverse for Hypotenuse mm
if GPS position Y in mm < B then
if GPS position X in mm > A then
turn to heading 90 degrees
drive reverse for Hypotenuse mm
else
turn to heading 270 degrees
drive reverse for Hypotenuse mm
if GPS position X in mm > A then
if GPS position Y in mm > B then
turn to heading 90 - Angle degrees
drive reverse for Hypotenuse mm
else
turn to heading 90 + Angle degrees
drive reverse for Hypotenuse mm
else
if GPS position Y in mm > B then
turn to heading 270 + Angle degrees
drive reverse for Hypotenuse mm
else
turn to heading 270 - Angle degrees
drive reverse for Hypotenuse mm
when started
set CoordinateNumber to 1
This 2D list represents the (X,Y) coordinates of where you want the robot to go.
set Coordinates to
150 900 150 900 200 850 125 850 125 850 -300 -300 -1400 -300 -1400 -300 -1400 -300 -1400 -300
-800 -450 -500 -400 -150 -150 400 250 850 300 0 -300 -1400 -150 -1400 0 1400 150 1400 300
Sets up velocities of all motors to 100% to maximize speed.
set drive velocity to 100 %
set turn velocity to 100 %
set ArmMotor velocity to 100 %
set IntakeMotor velocity to 100 %
spin ArmMotor down for 60 degrees and don't wait
turn to heading 330 degrees
Throws preload into net
Outtake
spin ArmMotor to position 1400 degrees and don't wait
ClawFRONT
turn to heading 240 degrees
Moves the five initial triballs near the center PVC into the red net
repeat 5
Intake
PusherFRONT
turn right for 180 degrees
Outtake
ClawFRONT
Intake
Moves onto the other side of the field to grab the center green triball, pushes the field preload into the blue goal, and throws and green triball into the red net.
spin ArmMotor to position 225 degrees and don't wait
PusherFRONT
turn to heading 90 degrees
drive reverse for 600 mm
drive forward for 600 mm
turn to heading 270 degrees
Outtake
Repeatingly goes to the red corners, grabs a ball, and throws it into the red goal
repeat 4
spin ArmMotor to position 1400 degrees and don't wait
ClawFRONT
Intake
spin ArmMotor to position 225 degrees and don't wait
PusherFRONT
turn to heading 270 degrees
Outtake
define Intake
spin IntakeMotor intake
wait until Optical detects green ? or Optical detects red ?
This delay ensures that the triball is fully picked up.
wait 0.15 seconds
stop IntakeMotor
define Outtake
spin IntakeMotor outtake
wait until not Optical detects green ? or Optical detects red ?
This delay ensures that the triball is fully shot out.
wait 0.2 seconds
stop IntakeMotor


define ClawFRONT
This block is similar to the "PusherFRONT" function, except the robot faces itself in the opposite direction. This allows the robot to grab a triball instead of pushing it, hence the name "ClawFRONT"
set A to Item 1 CoordinateNumber of Coordinates
set B to Item 2 CoordinateNumber of Coordinates
change CoordinateNumber by 1
set Hypotenuse to sqrt of GPS position X in mm - A * GPS position X in mm - A + GPS position Y in mm - B * GPS position Y in mm - B
set Angle to atan2 of X: abs of GPS position X in mm - A Y: abs of GPS position Y in mm - B
If GPS position X in mm = A then
If GPS position Y in mm > B then
turn to heading 180 degrees
drive forward for Hypotenuse mm
else
turn to heading 0 degrees
drive forward for Hypotenuse mm
else if GPS position Y in mm = B then
If GPS position X in mm > A then
turn to heading 270 degrees
drive forward for Hypotenuse mm
else
turn to heading 90 degrees
drive forward for Hypotenuse mm
else
If GPS position X in mm > A then
If GPS position Y in mm > B then
turn to heading 270 - Angle degrees
drive forward for Hypotenuse mm
else
turn to heading 270 + Angle degrees
drive forward for Hypotenuse mm
else
If GPS position Y in mm > B then
turn to heading 90 + Angle degrees
drive forward for Hypotenuse mm
else
turn to heading 90 - Angle degrees
drive forward for Hypotenuse mm