

# VEX CODE VR

Student Name: 6546E VEX

Class: Robotics Club (MQUAQQ)

Assignment:

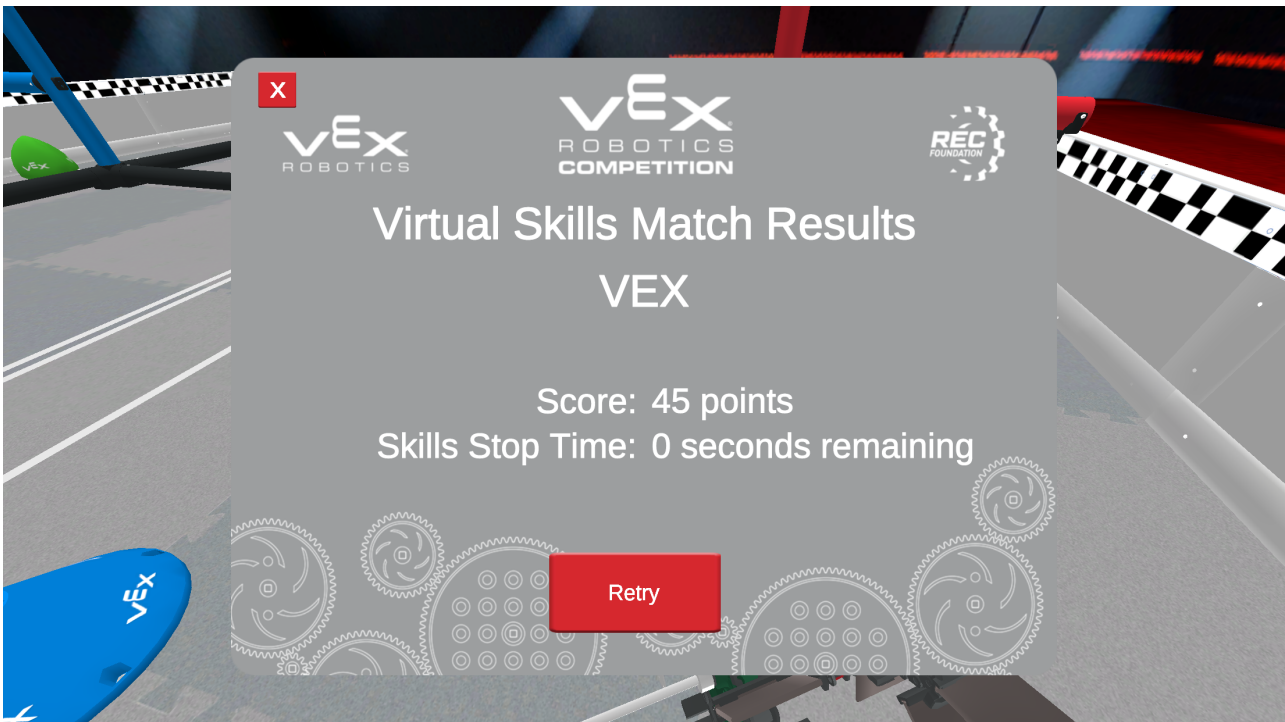
Notes:

Playground: VRC Virtual Skills - Over Under

Project Name: VEXcode Project

Project Type: Python

Date: Wed Jan 31 2024



```

1 #region VEXcode Generated Robot Configuration
2 import math
3 import random
4
5 from vexcode_vrc import *
6 from vexcode_vrc.events import get_Task_func
7
8 from math import sqrt
9 from math import degrees
10 from math import atan
11 from math import pow
12 from math import pi
13
14 brain=Brain()
15
16 drivetrain = Drivetrain("drivetrain", 0)
17 arm_motor = Motor("ArmMotor", 3)
18 rotation = Rotation("Rotation", 7)
19 intake_motor = Motor("IntakeMotor", 8)
20 optical = Optical("Optical", 11)
21 gps = GPS("GPS", 20)
22
23 #endregion VEXcode Generated Robot Configuration
24
25
26
27 ##FUNCTIONS
28 #SHOOTING
29 def shooting(): #function to set arm direction and shoot
30     intake_motor.stop() #stops the intake motor just in case it's still in
    taking
31     robotAngle = drivetrain.heading(DEGREES) #finds the robots position (an
    gle)
32     shootAngle = abs(robotAngle - 90) #figures out the robots angle relativ
    e to the goal
33
34     if(shootAngle < 90): #checks if the angle is less than 90 (robot is fac
    ing the goal)
35         arm_motor.spin(FORWARD) #arm spins forward to face to goal
36         intake_motor.spin(REVERSE) #shoot the triball out
37     else: #checks if the angle is greater than 90 (the robo
    t is turned away from the goal)
38         arm_motor.spin(REVERSE) #the arm will flip to face the goal
39         intake_motor.spin(REVERSE) #shoots the triball out
40 #SHOOTING
41
42
43 #ARM RESET
44 def armPos(direction): #function to reset the arms position WITHOUT SHOOTNG
45     if(direction == True): #if direction is true
46         arm_motor.spin(FORWARD) #then the arm will spin to face the front
    of the robot
47     else: #if direction is false
48         arm_motor.spin(REVERSE) #then the arm will spin to face the back o
    f the robot
49 #ARM RESET
50

```

```

51
52 #INTAKE & OUTTAKE
53 def intake(inOrOut):           #function to intake and outtake triball
54     if(inOrOut == True):       #if inOrOut is True, then INTAKE
55         intake_motor.spin(FORWARD) #motor spins forward to intake triball
56         wait(0.3, SECONDS)       #give enough time to intake
57         intake_motor.stop()      #stop intake motor
58     else:                       #if inOrOut is False, then OUTTAKE
59         while(optical.is_near_object() == True): #while there is an object
in the intake
60             intake_motor.spin(REVERSE) #spin reverse to outtake the triball
61             wait(5, MSEC)
62             intake_motor.stop()      #when there is no longer anything in the intake, stop the motor
63
64 #INTAKE & OUTTAKE
65
66 #MOVEMENT
67
68 def movement(xPositionFinal, yPositionFinal):
69     xPositionInitial = gps.x_position(MM) #Take the current position of the robot (X coordinate)
70     yPositionInitial = gps.y_position(MM) #Take the current position of the robot (Y coordinate)
71     xChange = xPositionFinal - xPositionInitial #find difference between current and final position (X coordinate)
72     yChange = yPositionFinal - yPositionInitial #find difference between current and final position (Y coordinate)
73     moveDistance = sqrt(pow(xChange, 2) + pow(yChange, 2)) #distance needed to travel a direct path
74
75     #find the angle the robot must turn to get to the final position
76     if (xChange > 0 and yChange > 0):
77         moveAngle = 90 - degrees(atan(yChange / xChange))
78     elif (xChange < 0 and yChange > 0):
79         moveAngle = 270 + degrees(atan(yChange / abs(xChange)))
80     elif (xChange < 0 and yChange < 0):
81         moveAngle = 270 - degrees(atan(yChange / xChange))
82     elif (xChange > 0 and yChange < 0):
83         moveAngle = 90 + degrees(atan(abs(yChange) / xChange))
84     elif (xChange == 0 and yChange > 0):
85         moveAngle = 0
86     elif (xChange > 0 and yChange == 0):
87         moveAngle = 90
88     elif (xChange == 0 and yChange < 0):
89         moveAngle = 180
90     elif (xChange < 0 and yChange == 0):
91         moveAngle = 270
92
93     drivetrain.turn_to_heading(moveAngle, DEGREES) #Turn the robot in the direction of the final destination
94     drivetrain.drive_for(FORWARD, moveDistance, MM) #drive there using the fastest route
95
96 #MOVEMENT
97
98 #MOVE BACKWARDS

```

```

99  def movementBack(xPositionFinal, yPositionFinal):
100     xPositionInitial = gps.x_position(MM)
101     yPositionInitial = gps.y_position(MM)
102     xChange = xPositionFinal - xPositionInitial
103     yChange = yPositionFinal - yPositionInitial
104     moveDistance = sqrt(pow(xChange, 2) + pow(yChange, 2))
105
106     if (xChange > 0 and yChange > 0):
107         moveAngle = 270 - degrees(atan(yChange / xChange))
108     elif (xChange < 0 and yChange > 0):
109         moveAngle = 90 + degrees(atan(yChange / abs(xChange)))
110     elif (xChange < 0 and yChange < 0):
111         moveAngle = 90 - degrees(atan(yChange / xChange))
112     elif (xChange > 0 and yChange < 0):
113         moveAngle = 270 + degrees(atan(abs(yChange) / xChange))
114     elif (xChange == 0 and yChange > 0):
115         moveAngle = 180
116     elif (xChange > 0 and yChange == 0):
117         moveAngle = 270
118     elif (xChange == 0 and yChange < 0):
119         moveAngle = 0
120     elif (xChange < 0 and yChange == 0):
121         moveAngle = 90
122     drivetrain.turn_to_heading(moveAngle, DEGREES)
123     drivetrain.drive_for(REVERSE, moveDistance, MM)
124
125     #MOVE BACKWARDS
126
127
128
129  def main():
130     #set velocities
131     drivetrain.set_drive_velocity(90,PERCENT)
132     drivetrain.set_turn_velocity(90,PERCENT)
133     intake_motor.set_velocity(120, PERCENT)
134     arm_motor.set_velocity(100, PERCENT)
135
136
137     #shooting first preload in
138     arm_motor.spin(FORWARD) #arm spinning
139     wait(500, MSEC)
140     movement(-900, 0) #go to middle
141     drivetrain.turn_to_heading(270, DEGREES) #turn to face goal
142     arm_motor.stop()
143     intake(False) #out take preload into goal
144
145     #shooting second preload in
146     drivetrain.turn_to_heading(0, DEGREES) #turn to other preload
147     intake(True) #intake
148     drivetrain.turn_to_heading(270, DEGREES) #turn to goal
149     intake(False) #out take into goal
150
151     drivetrain.set_drive_velocity(100,PERCENT)
152     drivetrain.set_turn_velocity(100,PERCENT)
153     #shooting third triball
154     movement(-800, 10)
155     intake(True) #begin intaking

```

```

156 movement(0, 300)
157 shooting() #shoot into goal
158 movement(500, 300)
159 movementBack(-500, 250) #go backwards to get the next ball
160 intake(False) #arm back down
161
162 #shooting fourth triball
163 movement(-300, 140) #align to the triball
164 intake(True) #intake centre triball
165 movement(-150, 150)
166 shooting() #shoot into goal
167 movement(500, 180)
168 intake(False) #outtake to shoot
169
170
171 #shooting fifth triball
172 movementBack(-500, 250)
173 movement(-300, 500)
174 intake(True)
175 movement(-150, 450)
176 shooting()
177 movement(500, 450)
178 intake(False)
179
180 #shooting sixth triball
181 movementBack(-500, 400)
182 movement(-300, 750)
183 intake(True)
184 movementBack(-300, -300)
185 movement(-150, -300)
186 shooting()
187 movement(500, -200)
188 movementBack(-300,-300)
189
190 #shooting seventh triball
191 drivetrain.turn_to_heading(150, DEGREES)
192 intake(True)
193 movement(-150, -300)
194 shooting()
195 movement(500, -300)
196
197 #shooting eighth triball
198 movementBack(-300, -600)
199 movement(-250, -800)
200 intake(True)
201 movement(-250, -400)
202 movement(100, -400)
203 shooting()
204 movement(500, -400)
205 intake(False)
206
207
208 #returning to match load zone
209 movement(-1350, -1450)
210 drivetrain.turn_to_heading(210, DEGREES)
211 intake(True)
212 movement(-150,-200)

```

```
213     shooting()
214     movement(500, -200)
215     intake(False)
216
217     #for loop to shoot match loads into goal
218     for i in range(0,4):
219         movement(-1350, -1450)
220         intake(True)
221         wait(200, MSEC)
222         movement(-150,-200)
223         shooting()
224         movement(500, -200)
225         intake(False)
226
227     vr_thread(main)
228
229
```