

```

#region VEXcode Generated Robot Configuration
import math
import random
from vexcode_viqc import *

# Brain should be defined by default
brain=Brain()

drivetrain = Drivetrain("drivetrain", 0)
intake_bumper = Bumper("IntakeBumper", 3)
front_optical = Optical("FrontOptical", 4)
intake_motor_group = Motor("IntakeMotorGroup", 5)
arm_motor_group = Motor("ArmMotorGroup", 6)
front_distance = Distance("FrontDistance", 9)

#endregion VEXcode Generated Robot Configuration
# -----
#
# Project:      Pink Skink VR Skills
# Author:      Jon Wilson
# Created:     Sept
# Description: This is a program that I used to learn how to program
# my physical robot better. I was introduced to variables when writing
# functions. I needed to pass variables, or parameters to my functions
# so that I had more control over my robot while also being able to type
# less. I also learned to use some sensors: the distance sensor and the
# bumper switch.
# -----

# Turns the intake on until it picks up a block
# Parameter needed, or variable that is passed to this function,
# is the distance needed in MM to reach the cube
def collectCube(distance):
    intake_motor_group.spin(FORWARD)
    drivetrain.drive_for(FORWARD, distance, MM)
    if intake_bumper.pressing():
        intake_motor_group.stop()

# Turns and goes to drop a cube in the goal. It then goes back to the
# original position.
# Parameter needed, or variable that is passed to this function,
# is the turn needed in degrees

```

```

def dumpInGoal(turn):
    arm_motor_group.spin_to_position(375, DEGREES, wait=False)
    drivetrain.turn_to_heading(turn, DEGREES)
    drivetrain.drive(FORWARD)
    distanceBack = front_distance.object_distance(MM)
    while front_distance.object_distance(MM) > 30:
        wait(5, MSEC)
    drivetrain.stop()
    intake_motor_group.spin(REVERSE)
    if not intake_bumper.pressing():
        intake_motor_group.stop()
    wait(1, SECONDS)
    drivetrain.drive_for(REVERSE, distanceBack, MM)
    arm_motor_group.spin_to_position(0,DEGREES, wait=False)
    drivetrain.turn_to_heading(0, DEGREES)

#This function knocks down the red. The robot will start from the point
#after scoring the second purple. It will then knock down all three reds
def reds():
    drivetrain.turn_to_heading(-20,DEGREES)
    drivetrain.drive_for(FORWARD, 460, MM)
    drivetrain.turn_to_heading(90,DEGREES)
    drivetrain.drive_for(FORWARD, 500, MM)
    drivetrain.turn_to_heading(50,DEGREES)

#This function goes forward, lifts the arm to be able to go over the
#bar to partially park, and turn to park
def park():
    drivetrain.drive_for(FORWARD, 650, MM)
    arm_motor_group.spin_to_position(200, DEGREES, wait=False)
    drivetrain.turn_to_heading(90,DEGREES)
    drivetrain.drive_for(FORWARD, 450, MM)

# Add project code in "main"
def main():
    # initializing the program and setting motor velocity
    # and torque for drivebase and arm
    drivetrain.set_drive_velocity(100, PERCENT)
    drivetrain.set_turn_velocity(100, PERCENT)
    arm_motor_group.set_velocity(100, PERCENT)
    intake_motor_group.set_velocity(100, PERCENT)
    # initializing the arm position to be 0 degrees and heading
    # to be 0 degrees
    arm_motor_group.set_position(0,DEGREES)
    drivetrain.set_heading(0, DEGREES)

    #setting variables distance and turn. These variables are set

```

```

#to 125 for distance because that is how far the robot needs to
#travel in MM to the first green cube and it must turn to heading
#125 to the left.
distance = 125
turn = -125

#getting the first 4 green cubes. There was a pattern that the
#robot needed to travel further to get each block and turn more.
#trial and error showed the difference is 10.
for repeat_count in range(4):
    collectCube(distance)
    dumpInGoal(turn)
    distance = distance + 10
    turn = turn -10

#get the first purple block
drivetrain.turn_to_heading(40,DEGREES)
collectCube(300)
drivetrain.turn_to_heading(-45, DEGREES, wait=False)
dumpInGoal(-38)

#get the second purple block
drivetrain.drive_for(REVERSE, 300, MM)
drivetrain.turn_to_heading(90, DEGREES)
collectCube(300)
dumpInGoal(-40)

#knock down the reds
reds()

#get some green cubes for the final goal
drivetrain.turn_to_heading(145,DEGREES)
collectCube(600)
dumpInGoal(130)
drivetrain.turn_to_heading(180,DEGREES)
collectCube(100)
dumpInGoal(125)

#go to park
park()

# VR threads – Do not delete
vr_thread(main)

```