# RoboCup 2021 Junior soccer simulation moyasirobo

# Team Description Paper

Japan Nagano-nct
Horiuchi Yushi 19327@g.nagano-nct.ac.jp
Tabei Tomoki 19421@g.nagano-nct.ac.jp

## Abstract

RoboCupJunior Simulation Challenge is Soccer robot simulation.
Run a program written in python with webots and compete with enemies for scores.
We wrote the program separately for the keeper and the attacker.
The keeper defends the goal and the attacker goes to score the goal.
It was very difficult because it is my first experience.

## Introduction

Horiuchi, in left side has never challenged RoboCupJunior.

But Tabei, in right side has participated in 2020.

Basically, Horiuchi gave an idea and Tabei make it happen by wright program. But both of us writing program.

## Our programs

Usually, we only improved.

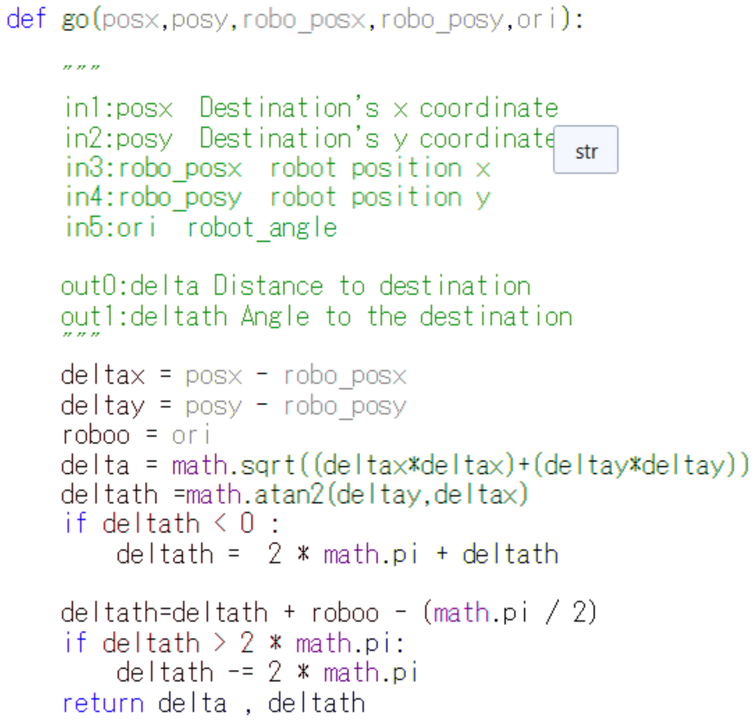
First, we wrote program that robots only chase down the ball.

But it is easy to goal for enemy robots, so we made Goalkeeper. And we wrote program that robots don't own goal. However, our team robots didn't goal. So, we wrote that attack program.

(1) base program- Chase the ball

we wrote this

In utils



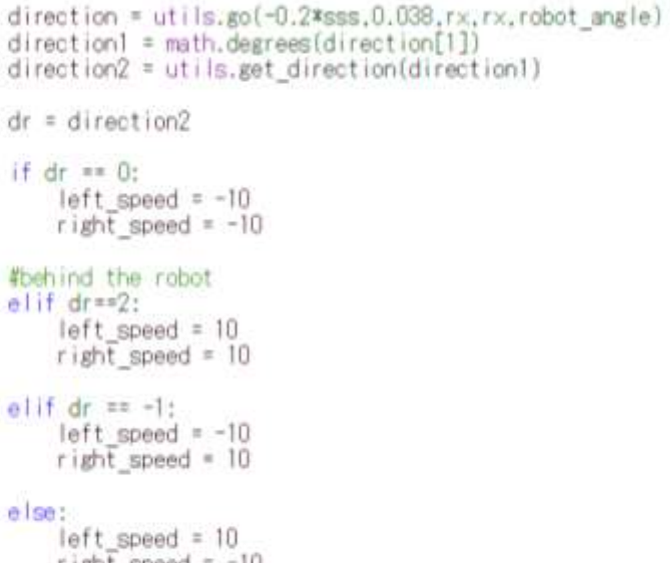
```
def go(posx,posy,robo_posx,robo_posy,ori):

    """
    in1:posx  Destination's x coordinate
    in2:posy  Destination's y coordinate
    in3:robo_posx  robot position x
    in4:robo_posy  robot position y
    in5:ori  robot_angle

    out0:delta Distance to destination
    out1:deltath Angle to the destination
    """
    deltax = posx - robo_posx
    deltay = posy - robo_posy
    roboo = ori
    delta = math.sqrt((deltax*deltax)+(deltay*deltay))
    deltath =math.atan2(deltay,deltax)
    if deltath < 0 :
        deltath =  2 * math.pi + deltath

    deltath=deltath + roboo - (math.pi / 2)
    if deltath > 2 * math.pi:
        deltath -= 2 * math.pi
    return delta , deltath
```

In each robot



```
direction = utils.go(-0.2*sss,0.038,rx,rx,robot_angle)
direction1 = math.degrees(direction[1])
direction2 = utils.get_direction(direction1)

dr = direction2

if dr == 0:
    left_speed = -10
    right_speed = -10

#behind the robot
elif dr==2:
    left_speed = 10
    right_speed = 10

elif dr == -1:
    left_speed = -10
    right_speed = 10

else:
    left_speed = 10
    right_speed = -10
```

We cased whether the ball was in front of, to the right, to the left, or behind the robot.

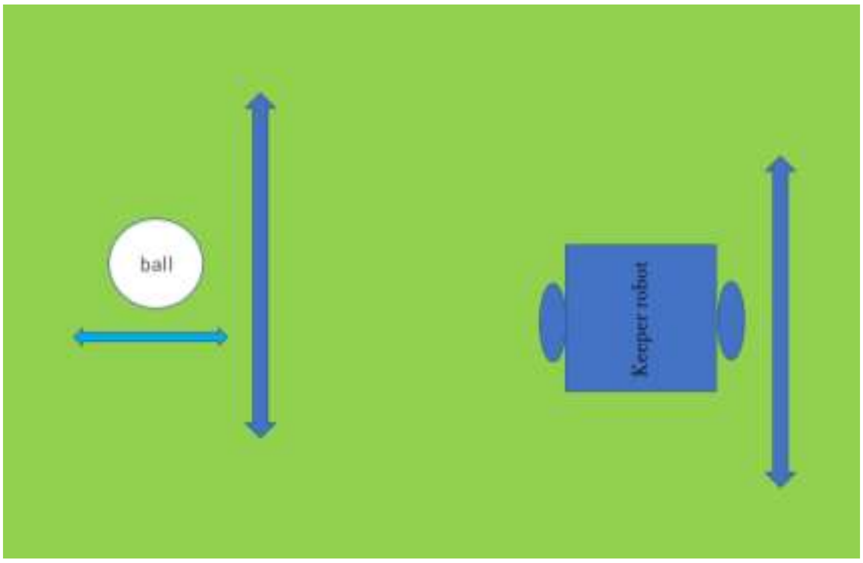
In front, the robot goes straight.

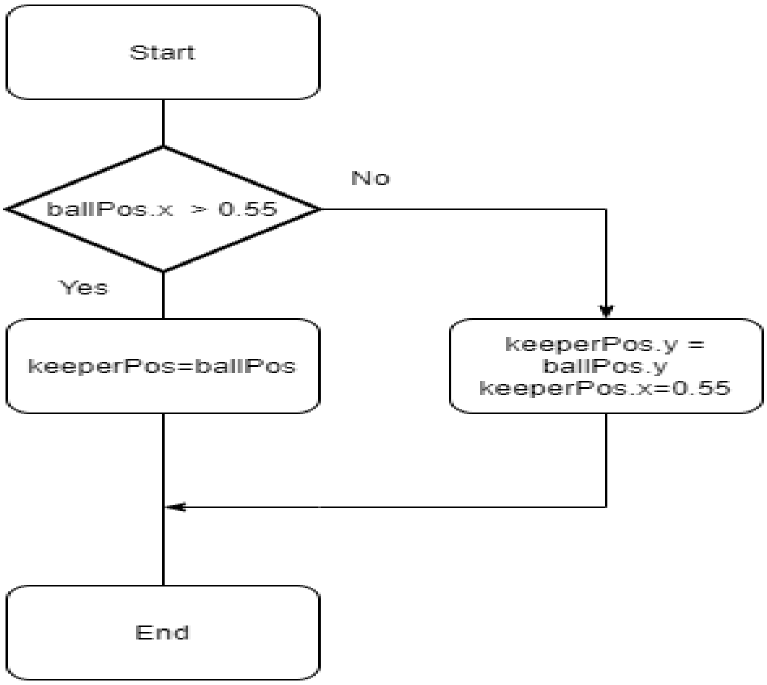
To right, it turns right.

To left, it turns left.

(2) keeper program

Move along the y coordinate of the ball.

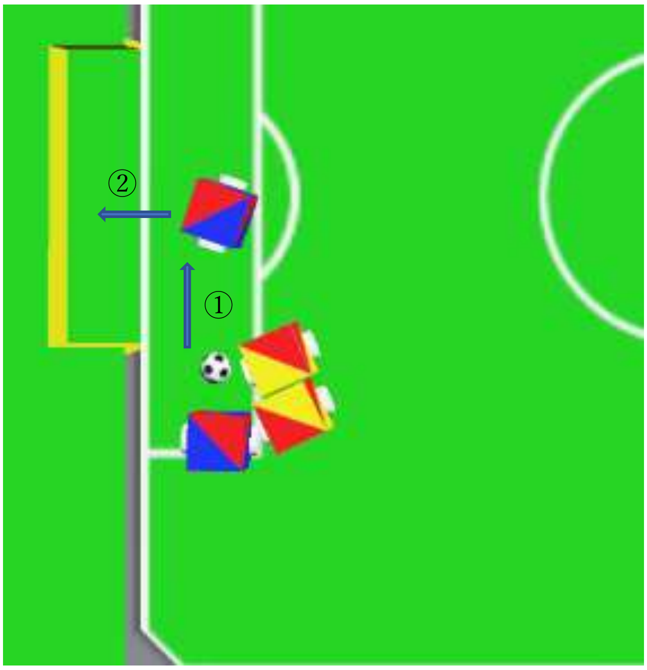


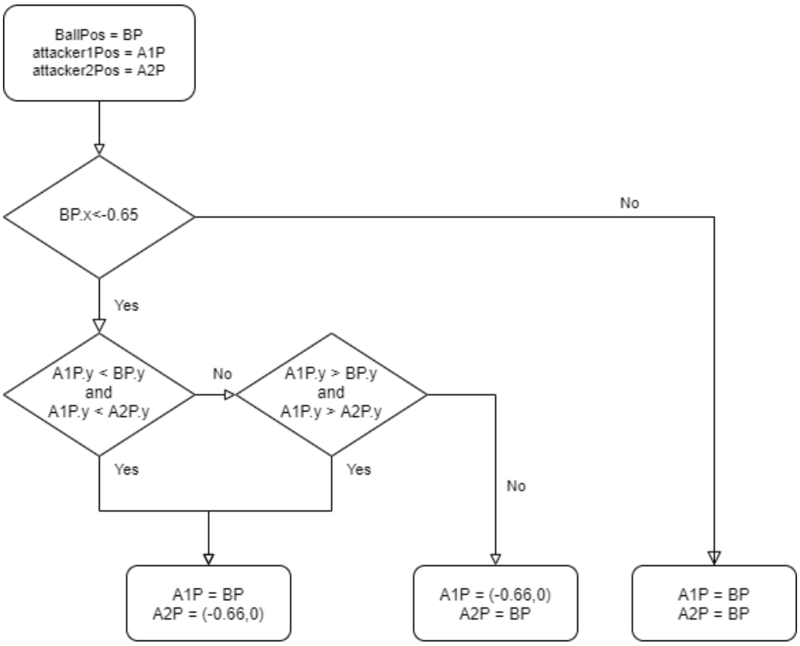


The robot to wait outside the penalty area so that it doesn't teleport when the ball isn't near the goal.

(3) attack program

Robot 1 passes to robot 2 in front of the goal and makes it goal.

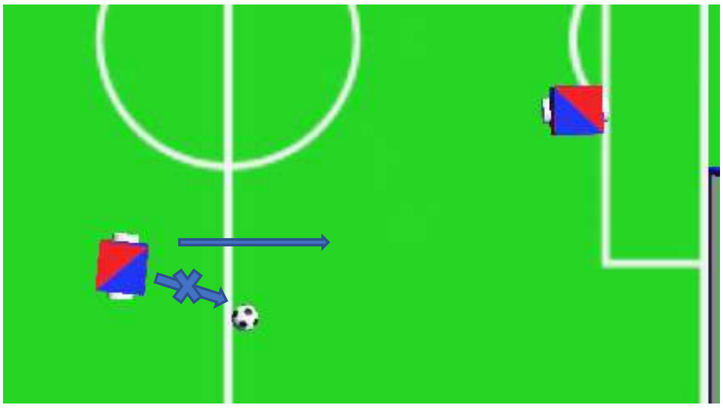




(4) other part

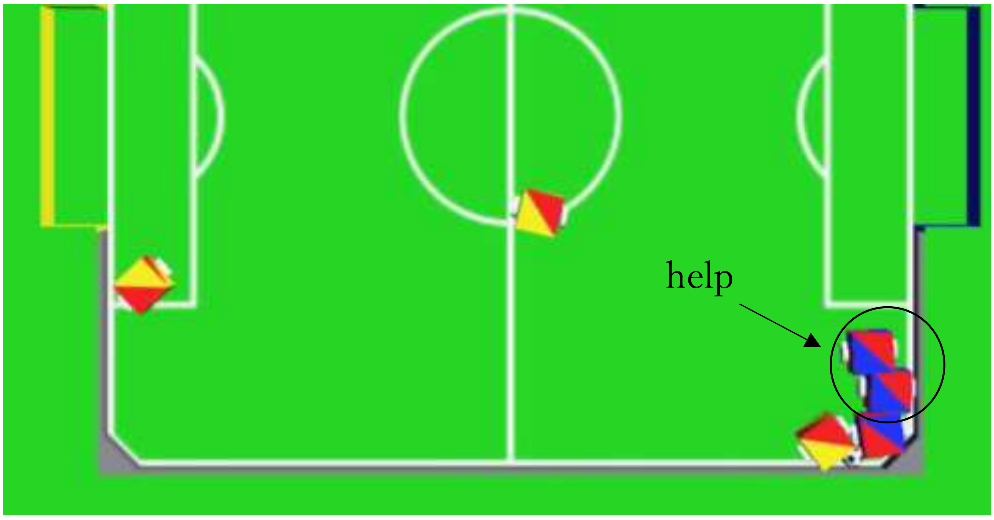
We considered s

If the ball is behind the robot, avoid the ball



And we thought about helping the goalkeeper.

If the goalkeeper and the enemy are in a push each other, attacker robots help it.



## Conclusions and Future Work

We want to use skillful footwork to disturb the enemy and follow the plow to reach the goal.

We didn't have time to research because of various things this time.

We want to increase the variation of offense and want the robot to chase the ball more smoothly.

## References

https://docs.python.org/ja/3/