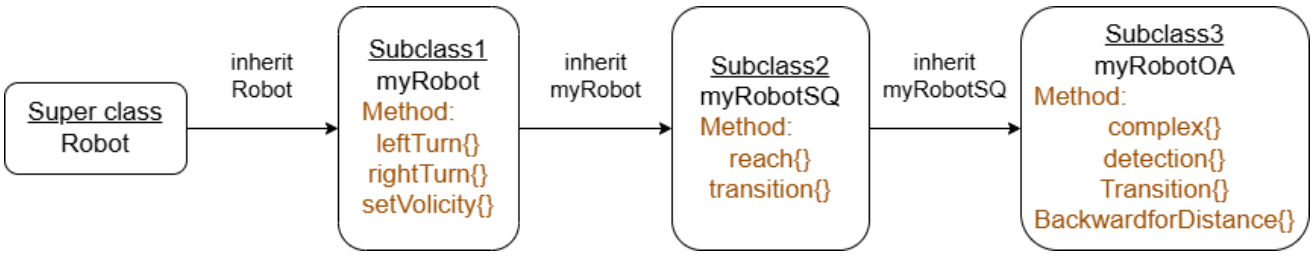
Robot

111323046 徐綺筑

1. Object-Oriented Programming: Subclass and Inheritance

- The project is implemented using:



myRobot for basic motion; myRobotSQ for square navigation; myRobotOA for Obstacle avoiding

- Subclass method

(a) Sensor feedback in OA

Input: proximity sensor's value, output: center of mass

getPositionSensor: get the value returned by 8 proximity sensors on E-puck.

Complex: take the value of each sensor as input, the output is the angular position of the center of mass calculated using:

$$\theta = \frac{M_0\theta_0 + M_1\theta_1 + \cdots + M_7\theta_7}{M_0 + M_1 + \cdots + M_7}$$

Thus, a method that transforms ps Value into the position of the center of mass can be got by:

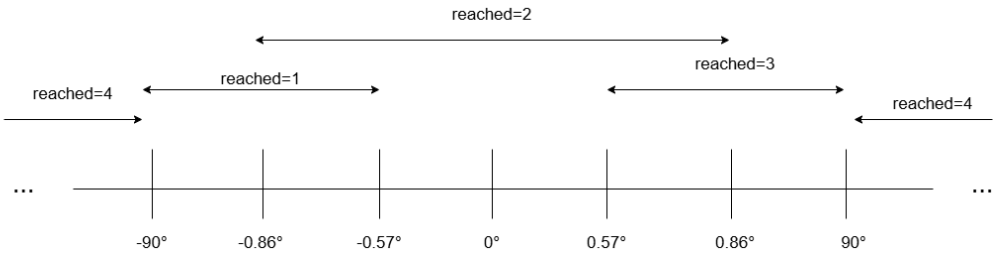
```
DistanceSeensor *ps[8]
void getPositionSensor(double *psValue){
// Get the value of the 8 proximity sensors on the E-puck.
}
```

```
// Compute center of mass
void Complex(double *psValue){
    getPositionSensor(psValue)
        // calculate mass value of each sensor and sum them up
        // complex number is used where real:value of sensor;
          imaginary:angular position of sensor
        // outputs the angular position of center of mass
        // It indicates the average direction of the robot's surrounding obstacles..
}
```

(b) obstacle detection

input: center of mass, output: reached condition

detection: define 4 kinds of reached condition based on the angular position of the center of mass: (1) obstacle on the left, (2) on the right, (3) in the middle and (4) no obstacle in front. How to define the reached is shown below.



If the centroid angle is within 0.86, it is defined as 2, and so on.

(c) State transition.

Input symbols:

U_1: obstacle on the left

U_2: obstacle in the middle

U_3: obstacle on the right

U_4: no obstacle in front

State:

S9: Go Forward

S10: Turn Right

S11: Turn Left

S12: 50% random choosing turning

S13: 10% random choosing turning

2. Finite-State Machines

- State diagram

First, do the square navigation: symbol A represents the reach condition whether reach the distance required.