Homework 03/10/2023

Q1: Write a class named AdvancedMotorPair. This class has the following features:

1. It inherits the MotorPair class, so it implicitly owns all the methods of MotorPair, such as move().

2. It supports the following functions.

```
#Turn certain degrees to right; the default degree is 90
def turnRight(self, speed=20, degree=90)

#Turn certain degrees to left; the default degree is 90
def turnLeft(self, speed=20, degree=90)

#Move forward on a exactly straight line (using gyro)
#Hint: you need to instantiate the PrimeHub inside the function because you need gyro.
def moveStraight(self, speed=20)

#follow a line based on the binary line following algorithm using the color sensor
#@port: the port number connected to the color sensor
#@color: the color of the line
#@speed: the speed of the driving base
#@seconds: move for an amount of seconds (default setting is 2 seconds)
def binaryLineFollower(self, port='A', color='black', speed=20, seconds=2)

#follow a line based on the proportional line following algorithm using the color sensor
#@port: the port number connected to the color sensor
#@speed: the speed of the driving base
#@seconds: move for an amount of seconds (default setting is 2 seconds)
def proportionalLineFollower(self, port='A', speed=20, seconds=2)
```

The constructor method will have only two parameters port1 and port2, shown as below:
For example, when the user initiates an instance, the code looks like this:

```
motorPair = AdvancedMotorPair('B','C')
```

Then, use the following codes to test the class:

```
hub = PrimeHub()
motorPair = AdvancedMotorPair('B','C')
motorPair.move(5,'cm')
motorPair.turnRight(20, 90)
motorPair.move(5,'cm')
motorPair.turnLeft(20, 90)
motorPair.move(5,'cm')
motorPair.turnRight(20,145)
motorPair.moveStraight(20,2)
motorPair.binaryLineFollower('A', 'black', 20, 2)
motorPair.proportionalLineFollower('A', 20, 2)
```

Hint: I attached the **AdvancedMotorPair** class used in our last class and the slides about binaryLineFollower code and proportionalLineFollower code for your reference/convenience.

Q2: PACMAN game

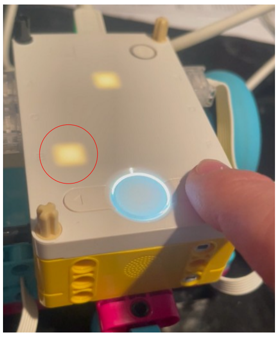
Treat the hub’s 5x5 light matrix as the map of the game.

Use the left button and right button to move one light spot (the pac-man) to eat another light spot (the bean). At the beginning the pac-man appears at the top-left corner of the map, and the bean is located at the centre of the map. Whenever the pac-man hits the bean, the bean will be regenerated at a random location of the map. Then, the player can use the left/right button to move the pac-man again to eat that bean, and so on so forth.

Everytime you click the left and right button to move the pac-man, the hub needs to beep a sound 60 with default sound length 0.2 seconds. When it hits/eats a bean, the hub needs to beep another sound 70 with a longer sound length 0.5 seconds.

There is a demo in the wechat group, please ask your parent to show you the demo.

Move the pac-man



pac-man hits the bean

Hint: As for how to control the pixel on light matrix, please take a look at slide 19 of this lesson: https://primelessons.org/en/PyProgrammingLessons/ListsTuples.pdf