

# CS101 Homework #1

( Save a lost child in a maze )

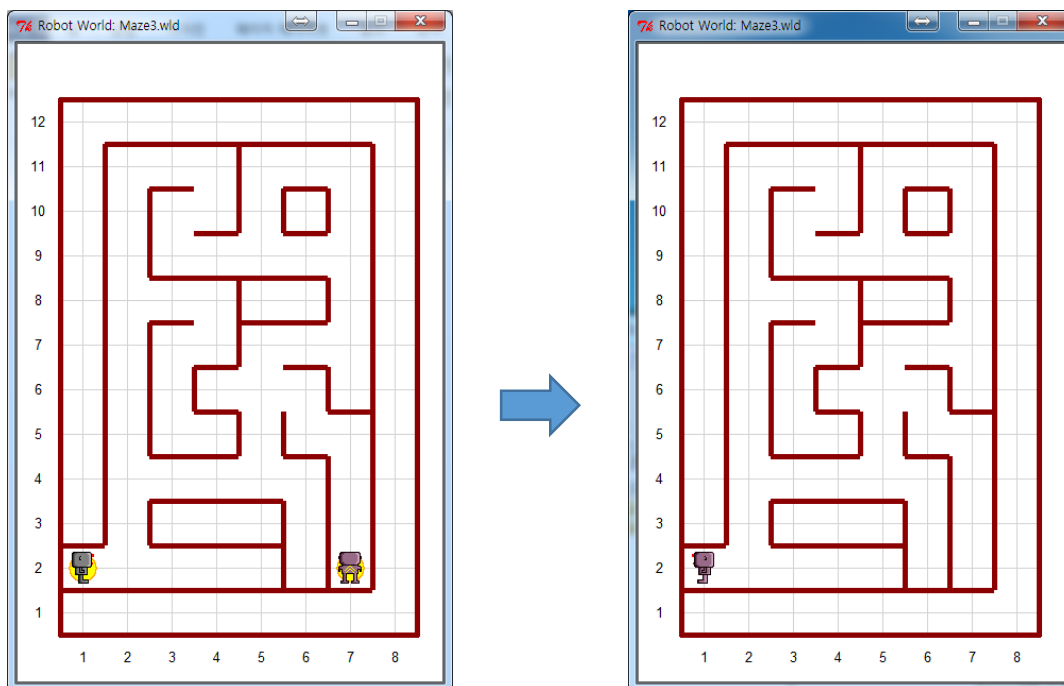
Due date: Sunday, Sept. 27, 2015 (Until 23:59)

Delayed-due date: Wednesday, Sept. 30, 2015 (Until 23:59)

**Please read the homework description carefully** and make sure that your program meets all the requirements stated. The homework is an individual task. You can discuss the problem with your friends but you should not program together. **You will get an F on the entire course if your homework includes any plagiarism.**

## Goal

Ami gets lost in a maze, so Hubo has to bring Ami from the maze. Let Hubo find Ami and escape from the maze together with Ami.



## Requirements

You should implement two parts in the given template code, finding Ami and returning to the exit of the maze. First, Hubo explores the maze to find Ami. At this point, Hubo has to drop beepers to indicate the route to the exit of the maze. After finding Ami, Hubo returns to the exit with Ami through the indicated route.

**Refer to the videos provided in order to help you understand this homework and remember that they are just examples.** It means that you will not be penalized as long as it works well and satisfies the following requirements, even if your program does not work exactly same as the provided sample videos. **So please keep focusing on meet the following requirements.**

1. **Your program has to work with the given five mazes.** You will get partial points even if your program works on some of the mazes, so do your best! If you have no idea, please read *Hints* section in the below.
2. **You can assume the followings:**
  - 1) There is no wide street.
  - 2) The positions of the exit and Ami are fixed to (1, 2) and (7, 2), respectively. In addition, the positions are always the end of streets (`hubo.right_is_clear() == False`, `hubo.front_is_clear() == False`, and `hubo.left_is_clear() == False`).
3. **You can use only what you have learnt until the 3<sup>rd</sup> lecture.** Note the followings:
  - 1) You CANNOT use function parameters and return values
  - 2) You CANNOT use *break* or *continue*.
4. **Note that your source code has to contain appropriate comments.** Please note the followings:
  - 1) The title of the program and the author information must be written at the head of your source code.
  - 2) A brief step-by-step description of algorithm should be included at appropriate positions.
  - 3) It is highly recommended to add comments to improve the readability of your source code. If your source code contains only a few comments, you will get some penalties.
5. **You must also submit a report.** There is no strict format, but only *doc*, *docx*, and *pdf* files are allowed for your report. Then, the report should include the followings:
  - 1) Summary or explanation about the overall algorithm
  - 2) Screen shots of your program to show processes and results of your program execution well.
  - 3) Describe about what you learnt and what you felt while doing this homework.

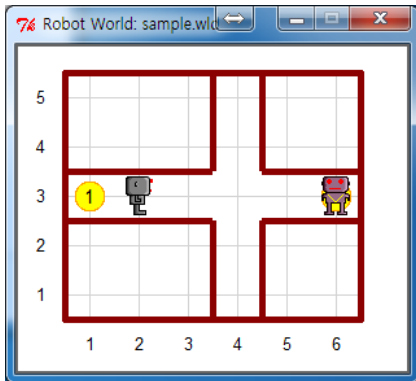
## **Hints**

We explain the ‘**Follow right wall**’ function first, and then we provide two hints to help your implementation. The hints include how to indicate the route and to go along the indicated route. We recommend to implement this homework by using the ‘Follow right wall’ function first and then to modify the function. If it is hard for you, please implement this homework using only the ‘Follow right wall’ function (without the indicated route). Then, you will get partial points.

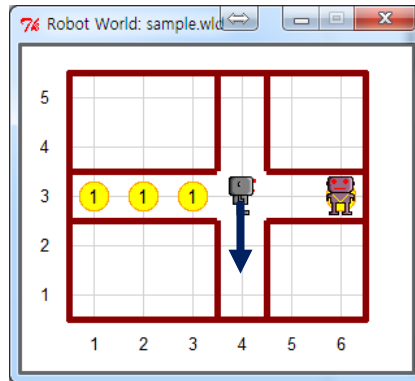
### **1. The ‘Follow right wall’ function**

- In the ‘Follow right wall’ function on the ‘Robot Lecture Note’, Hubo can move one step with following the right wall. Then, Hubo can explore the world by moving each one step until Hubo reaches the beeper.
- In this homework, Hubo and Ami drop beepers to indicate their positions at first, and then Hubo moves one step as shown in Fig. 1-(a). These are already typed in the given template code. After that, Hubo explores the maze until Hubo reaches Ami’s position. When Hubo meets Ami, Hubo will be on a beeper and all directions (right, front, and left) will not be clear as shown in Fig. 1-(e). This is the terminal condition of the function, which is also typed in the given template code.
- In the ‘Follow right wall’ function of the ‘Robot Lecture Note’, there are three cases with priorities as the

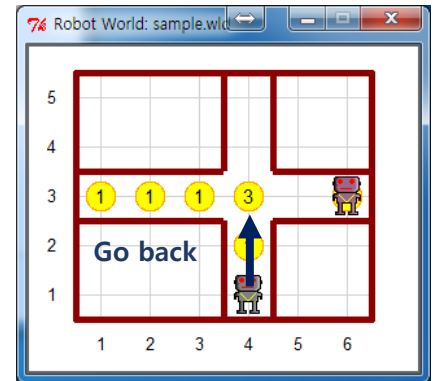
followings; 1) right is clear, 2) front is clear, and 3) otherwise. However, in this homework, we recommend to classify them into four cases as followings; 1) right is clear, 2) front is clear, 3) left is clear, and 4) otherwise.



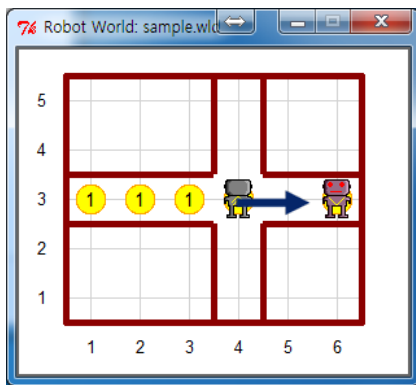
**Fig. 1-(a)**



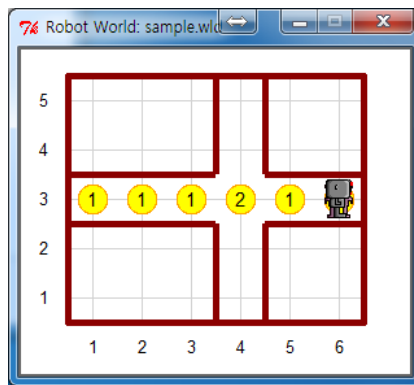
**Fig. 1-(b)**



**Fig. 1-(c)**



**Fig. 1-(d)**



**Fig. 1-(e)**

## 2. Indicating the route

- While exploring the maze to find Ami, Hubo has to indicate the route by using beepers. Hubo drops beepers if Hubo is not on a beeper, and Hubo picks up a beeper if Hubo is on the beeper. Hubo should drop or pick up beepers before moving each step.
- Fig. 1-(a) ~ (e) shows the example of the 'Follow right wall' function with indicating the route. Hubo drops a beeper and moves on every step as shown in Fig. 1-(b) ~ (c). However, when Hubo returns to the junction from the end of the street as shown in Fig. 1-(c), Hubo will be on a beeper. In this case, Hubo picks up the beeper and moves on as shown in Fig. 1-(d). Then, when Hubo meets Ami, the route between the exit and Ami's position will be indicated as shown in Fig. 1-(e).
- The number of dropped beepers on each step should be equal to the number of directions that Hubo can move to. The numbers shown in Fig. 1 indicate the number of dropped beepers. In Fig. 1-(a), Hubo can move only forward, so Hubo drops one beeper. Hubo on the junction (Fig. 1-(b)) drops three beepers because Hubo can move to the right, front, and left. If Hubo is on the T-junction, Hubo will drop two beepers. In other words, Hubo drops one beeper for each direction (right, front, and left) that is clear.

### 3. Returning to the exit through the indicated route

- You can also implement this procedure by modifying the 'Follow right wall' function. First, the condition of the while statement should be modified as *'while hubo.on\_beeper()'* because Hubo will go along the indicated route. This is already typed in the given template code. The other modification is that Hubo has to return to the junction immediately if Hubo is not on a beeper. We explain that by using the following example. First, Hubo and Ami pick up a beeper and move on as shown in Fig. 2-(a) and (b). When they reach the junction, Hubo turns right and moves on by the 'Follow right wall' function (Fig. 2-(c)). However, there is no beeper, so Hubo goes back to the junction immediately. Then, Hubo turns right again to go to the forward direction. Since Hubo is on a beeper (Fig. 2-(d)), Ami picks up the beeper and follows Hubo as shown in Fig. 2-(e). Finally, Hubo and Ami can return to the exit.
- To explain in detail, we explain them with four cases of the 'Follow right wall' function as followings:
  - 1) If right is clear, Hubo turns right and moves on. If Hubo is not on a beeper, Hubo returns to the junction. Otherwise, Ami picks up a beeper and follows Hubo.
  - 2) If front is clear, Hubo moves one step and then checks a beeper. Similarly, Hubo returns to the junction if there is no beeper, and otherwise Ami picks up a beeper and follows Hubo.
  - 3) If left is clear, Hubo turns left and moves on, and then Hubo also checks a beeper like the upper cases.
  - 4) Otherwise, if all directions are not clear, Hubo reaches the exit of mazes, so Hubo just picks up a beeper.

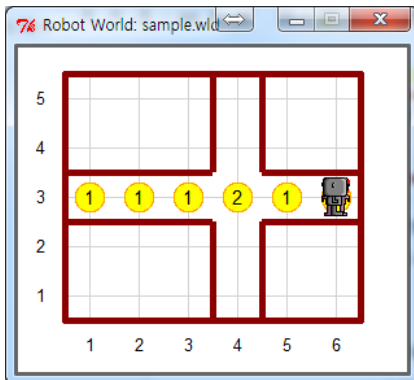


Fig. 2-(a)

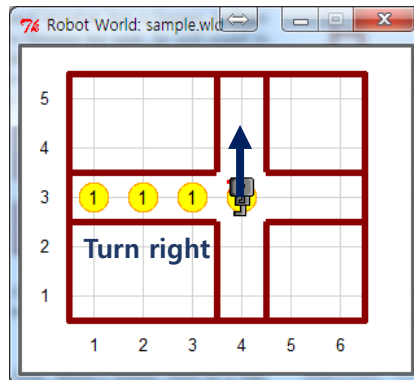


Fig. 2-(b)

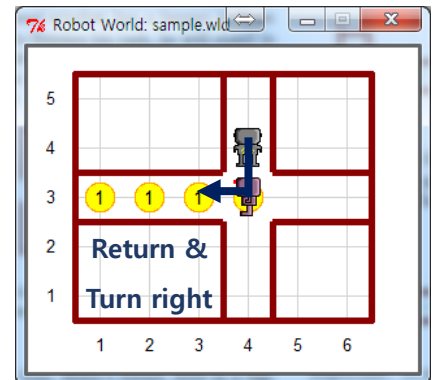


Fig. 2-(c)

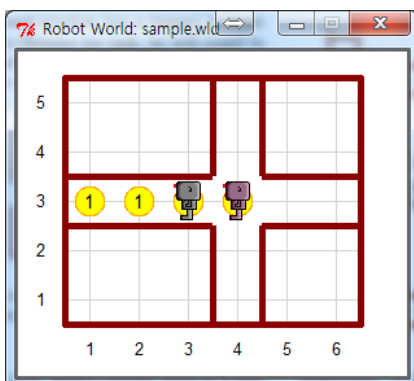


Fig. 2-(d)

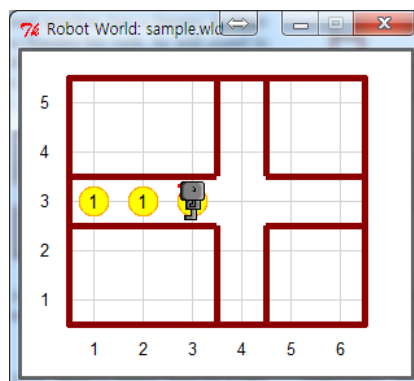


Fig. 2-(e)

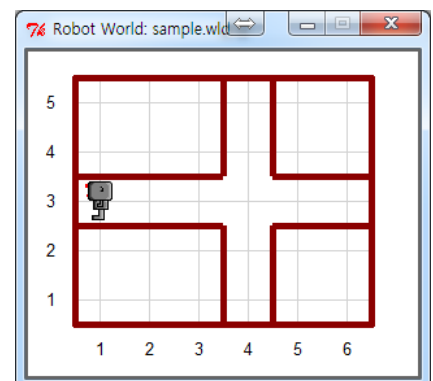


Fig. 2-(f)

## **Submission**

You need to submit the followings:

- The program file: 'HW1\_yourid.py' (e.g.) HW1\_20151234.py
- The homework report: 'HW1\_yourid.doc', 'HW1\_yourid.docx', or 'HW1\_yourid.pdf' (e.g.) HW1\_20151234.doc, HW1\_20151234.docx, HW1\_20151234.pdf

You **MUST** compress the source code and the report together into 'HW1\_yourid.zip' (e.g.), HW1\_20151234.zip and submit the compressed file via the webpage for homework submission.

**If you do not follow the above submission policy, you will get penalty.**