*Break into POGIL teams of 4 and assign each team member one of the following roles.*

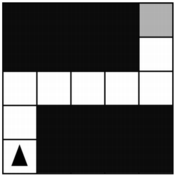
|  |  |  |
| --- | --- | --- |
| **Student Name** | **Role** | **Responsibility** |
|  | Facilitator | Reads the questions aloud, keeps track of time and makes sure everyone contributes appropriately. |
|  | Spokesperson | Talks to the instructor and other teams. |
|  | Quality Control | Records all answers & questions, and provides team’s reflection to team and instructor. |
|  | Process Analyst | Considers how the team could work and learn more effectively. |

Algorithms: Solving a Maze

The problem below is similar to a type of AP CSP exam question. Consider a robot that can follow the simple sequence commands below:

* **MOVE\_FORWARD** : The robot moves 1 square forward in the direction it is facing.
* **ROTATE\_RIGHT** : The robot turns right 90 degrees, staying in the same square but facing right.
* **ROTATE\_LEFT** : The robot turns left 90 degrees, staying in the same square but facing left.
* **CAN\_MOVE( *direction* )** : This command can be used with 4 possible directions: **left, right, forward,** and **backward**. It returns true if there is an open square in the specified direction from the square that the robot is in.

Let's put our robot in the maze below. The robot is represented as a black triangle and is initially facing up. It can only move forward to a white square. It cannot move onto the black squares or move beyond the edge of the grid.



Answer the following questions with your POGIL team:

1. For the robot in the maze above, is CAN\_MOVE(forward) true? Is CAN\_MOVE(right) true?
2. (Portfolio) Write an algorithm using the 4 commands above to navigate the robot through the maze to reach the gray square. You can pretend that one of you is the robot and walk through your algorithm with your fingers on the maze. Are there commands that are repeated in your algorithm? Circle them.
3. (Portfolio) Let's replace the repeated commands with a repetition control structure. The following command can be used to repeat a block of commands:

REPEAT n times

Commands

Rewrite your algorithm above using Repeat n times control structures (substituting in a

number for n) instead of repeating the MOVE\_FORWARD command many times.

4.) A more general way to navigate a maze is with the following algorithm which uses GoalReached to test if the robot has reached the gray square.

REPEAT UNTIL GoalReached

{

IF (CAN\_MOVE forward)

MOVE\_FORWARD

IF (CAN\_MOVE left)

ROTATE\_LEFT

IF (CAN\_MOVE right)

ROTATE\_RIGHT

}

* 1. Which part(s) of the algorithm above are selection control structures? Circle them.
  2. Which part of the algorithm above is a repetition control structure? Circle it. Remember a control structure can consist of multiple statements.
  3. Does the algorithm solve the maze above and navigate the robot to the goal, the gray square? How many times does it need to run through the loop?
  4. (Portfolio) Can you come up with a maze that this algorithm will not be able to solve? Describe or take a photo of your drawing of such a maze for your portfolio.

5.) (Portfolio) Write an algorithm for washing a stack of 10 items that are cups and saucers mixed together, where the rule is that the cups are washed in hot water and the saucers in cold water. Use simple commands like hot\_wash and cold\_wash. You may also use the control structures IF and REPEAT n times. Identify the parts of your algorithm that are examples of Sequence, Selection, and Repetition.