

Goal: Use the right-hand rule to navigate through a maze.

An algorithm that follows the [right-hand rule](#) works well for solving simple puzzles, but did you know that you can also use it to move through more complicated paths, like mazes?

For this puzzle, walk through the maze in your mind. If you always keep your right hand on the wall, you'll eventually reach the last gem. To solve the maze, adjust your algorithm from the previous page to navigate to the gem and collect it.

- 1 Start with the code from the previous page by copying and pasting, or, if you want a challenge, start with all-new code.
 - 2 Adjust your code so that your character follows the right-hand rule through maze until reaching the gem.
-

```
func wall() {
    if isBlockedRight && !isBlocked {
        moveForward()
    }
    if (isBlocked && !isBlockedRight && !isBlockedLeft) {
        turnRight()
        moveForward()
    }
    if (!isBlocked && isBlockedLeft && !isBlockedRight) {
        turnRight()
    }
    if (isBlocked && isBlockedLeft) {
        turnRight()
        for i in 1 ... 3 {
            moveForward()
        }
        turnLeft()
        moveForward()
    }
    if (isBlocked && isBlockedRight && !isBlockedLeft) {
        turnLeft()
    } else {
        moveForward()
    }
}
```

```
    }  
}  
  
// While Loop to Run Function  
while !isOnGem {  
    wall()  
    if isOnGem {  
        collectGem()  
        break  
    }  
}
```