Goal: Adjust your algorithm to navigate around extra blocks.

This puzzle is similar to the previous one, but additional blocks around the walls prevent your right-hand rule algorithm from working properly. In some situations, you're blocked on the right, but you can't move forward because you're also blocked in the front.

> Right-hand rule
>
> Adjust your algorithm to deal with these situations:
>
>  Blocked on the right and in front.
>
>  Blocked on the right.
>
>  Not blocked.

To fix this, you'll need to tweak your algorithm. The image above shows the three different situations your character will encounter, and the arrows suggest how to respond for each one. Can you modify `navigateAroundWall()` to handle each situation?

1. Use pseudocode to think through how your character should move in the three situations above.
2. Based on your pseudocode, tweak the existing code below, then run it to see what happens.

```
    //Hacky but works

func wall() {
    if isOnGem {
        collectGem()
    }
    if isBlockedRight {
        moveForward()
    }
    if !isBlockedRight {
        turnRight()
        moveForward()
    }
```

```
    if isBlocked && !isBlockedLeft {
        turnLeft()
    }
    if isBlocked && isBlockedLeft && !isBlockedRight {
        turnRight()
    }
    if isBlocked && isBlockedLeft && isBlockedRight {
        turnLeft()
    }    //if !isBlocked && isBlockedLeft &&
     !isBlockedRight{moveForward()}
}

while !isOnClosedSwitch {
    wall()
    if isOnGem {
    collectGem()
    turnLeft()
    turnLeft()
    }
}
toggleSwitch()
```