

# Lesson 9

## Parameters, Moving Further Forward

```
let expert = Expert()
func move (distance: Int) {
    for i in 1...distance {
        expert.moveForward ()
    }
}
```

```
move(distance: 6)
expert.turnRight()
expert.move(distance: 2)
expert.turnRight()
move(distance: 5)
expert.turnLeft()
move(distance: 5)
expert.turnLeft()
expert.turnLockUp()
expert.turnLockUp()
expert.turnLeft()
```

```
move(distance: 3)
expert.turnRight()
move(distance: 3)
expert.turnRight()
move(distance: 3)
expert.turnLeft()
move(ditance: 2)
expert.turnLeft()
move(distance: 2)
expert.collectGem()
```



# Lesson 9

## Parameters, Generalizing a Function

```
func turnLock(up up: Bool, numberOfTimes: Int) {  
  for _ in 1...numberOfTimes {  
    if up == true {  
      expert.turnLockUp()  
    } else {  
      expert.turnLockDown()  
    }  
  }  
}
```

```
func expertTurnAround() {  
  expert.turnLeft()  
  expert.turnLeft()  
}
```

```
func characterTurnAround() {  
  character.turnLeft()  
  character.turnLeft()  
}
```

```
expert.turnLock(up: true, numberOfTimes: 4)  
expertTurnAround()
```

```
expert.turnLock(up: true, numberOfTimes: 4)  
character.move(distance: 3)  
character.collectGem()  
character.TurnAround()  
character.moveFoward()  
expert.turnLockUp()  
character.turnRight()  
character.moveForward()  
character.collectGem()  
characterTurnAround()  
character.move(distance: 2)  
expert.turnLock(up: false, numberOfTimes: 3)  
character.turnRight()  
character.moveForward()  
character.turnLeft()  
character.moveForward()  
character.collectGem()  
characterTurnAround()  
character.move(distance: 2)  
expertTurnAround()  
expert.turnLockDown()  
character.moveForward()  
character.collectGem()
```



# Lesson 9

## Parameters, Generalizing a Function

```
func turnLock(up up: Bool, numberOfTimes: Int) {  
  for _ in 1...numberOfTimes {  
    if up == true {  
      expert.turnLockUp()  
    } else {  
      expert.turnLockDown()  
    }  
  }  
}
```

```
func expertTurnAround() {  
  expert.turnLeft()  
  expert.turnLeft()  
}
```

```
func characterTurnAround() {  
  character.turnLeft()  
  character.turnLeft()  
}
```

```
expert.turnLock(up: true, numberOfTimes: 4)  
expertTurnAround()
```

```
expert.turnLock(up: true, numberOfTimes: 4)  
character.move(distance: 3)  
character.collectGem()  
character.TurnAround()  
character.moveFoward()  
expert.turnLockUp()  
character.turnRight()  
character.moveForward()  
character.collectGem()  
characterTurnAround()  
character.move(distance: 2)  
expert.turnLock(up: false, numberOfTimes: 3)  
character.turnRight()  
character.moveForward()  
character.turnLeft()  
character.moveForward()  
character.collectGem()  
characterTurnAround()  
character.move(distance: 2)  
expertTurnAround()  
expert.turnLockDown()  
character.moveForward()  
character.collectGem()
```



# Lesson 9

## Parameters, Crank Up and Down

N/A

## Parameters, Placing at a Specific Location

```
let expert = Expert()
world.place(expert, facing: south, atColumn: 2,
row: 6)
```

```
func turnAround() {
    expert.turnLeft()
    expert.turnLeft()
}
```

```
func turnLockCollectGem() {
    expert.turnLeft()
    expert.turnLockUp()
    turnAround()
    expert.moveForward()
    expert.collectGem()
    turnAround()
    expert.moveForward()
    expert.turnRight()
}
```

```
turnLockCollectGem()
expert.move(distance: 4)
turnLockCollectGem()
expert.move(distance: 4)
expert.turnRight()
expert.moveForward()
expert.collectGem()
```



# Lesson 9

## Parameters, Crank Up and Down

N/A

## Parameters, Placing at a Specific Location

```
let expert = Expert()
world.place(expert, facing: south, atColumn: 2,
row: 6)
```

```
func turnAround() {
    expert.turnLeft()
    expert.turnLeft()
}
```

```
func turnLockCollectGem() {
    expert.turnLeft()
    expert.turnLockUp()
    turnAround()
    expert.moveForward()
    expert.collectGem()
    turnAround()
    expert.moveForward()
    expert.turnRight()
}
```

```
turnLockCollectGem()
expert.move(distance: 4)
turnLockCollectGem()
expert.move(distance: 4)
expert.turnRight()
expert.moveForward()
expert.collectGem()
```



# Lesson 9

## Parameters: Place an Expert

N/A

## Parameters: Placing Two Characters

```
let character = Character()
```

```
let expert = Expert()
```

```
world.place(character, facing: north, atColumn: 0, row: 0)
```

```
world.place(expert, facing: north, atColumn: 3, row: 0)
```

```
func collectAndJump() {  
    while !character.isBlocked {  
        character.collectGem()  
        character.jump()  
        character.Jump  
    }  
}
```

```
expert.toggleSwitch()
```

```
expert.turnLock(up: false, numberOfTimes: 3)
```

```
collectAndJump()
```

```
character.turnRight()
```

```
collectAndJump()
```

```
character.turnLeft()
```

```
character.collectGem()
```

```
character.move(distance: 2)
```

```
character.collectGem()
```



# Lesson 9

## Parameters: Place an Expert

N/A

## Parameters: Placing Two Characters

```
let character = Character()
```

```
let expert = Expert()
```

```
world.place(character, facing: north, atColumn: 0, row: 0)
```

```
world.place(expert, facing: north, atColumn: 3, row: 0)
```

```
func collectAndJump() {  
    while !character.isBlocked {  
        character.collectGem()  
        character.jump()  
        character.Jump  
    }  
}
```

```
expert.toggleSwitch()
```

```
expert.turnLock(up: false, numberOfTimes: 3)
```

```
collectAndJump()
```

```
character.turnRight()
```

```
collectAndJump()
```

```
character.turnLeft()
```

```
character.collectGem()
```

```
character.move(distance: 2)
```

```
character.collectGem()
```



# Lesson 9

Parameters: Two Experts

N/A

Parameters: Twin Peaks

N/A

\*Some puzzles may have multiple solutions



# Lesson 9

Parameters: Two Experts

N/A

Parameters: Twin Peaks

N/A

\*Some puzzles may have multiple solutions

