

# World Building

## World Building: Uniting Worlds

```
let block1 = Block()
```

```
world.place(block1, atColumn: 3, row: 3)
```

```
while !isOnClosedSwitch {
```

```
  moveForward()
```

```
  if isBlocked {
```

```
    turnLeft()
```

```
    if isBlocked {
```

```
      turnRight()
```

```
      turnRight()
```

```
    }
```

```
  }
```

```
}
```

```
toggleSwitch()
```



# World Building

## World Building: Connect and Solve

```
let block1 = Block()
```

```
let block2 = Block()
```

```
let block3 = Block()
```

```
let block4 = Block()
```

```
let block5 = Block()
```

```
world.place(block1, atColumn: 2, row: 2)
```

```
world.place(block2, atColumn: 2, row: 2)
```

```
world.place(block3, atColumn: 4, row: 2)
```

```
world.place(block4, atColumn: 6, row: 2)
```

```
world.place(block5, atColumn: 6, row: 2)
```

```
func crossBridge() {  
    turnRight()  
    move(distance: 4)  
    collectGem()  
    turnLeft()  
    turnLeft()  
    move(distance: 4)  
    turnRight()  
}
```

```
for i in 1..3 {  
    move(distance: 2)  
    toggleSwitch()  
    crossBridge()  
}
```



# World Building

## World Building: Connect and Solve

```
let block1 = Block()
```

```
let block2 = Block()
```

```
let block3 = Block()
```

```
let block4 = Block()
```

```
let block5 = Block()
```

```
world.place(block1, atColumn: 2, row: 2)
```

```
world.place(block2, atColumn: 2, row: 2)
```

```
world.place(block3, atColumn: 4, row: 2)
```

```
world.place(block4, atColumn: 6, row: 2)
```

```
world.place(block5, atColumn: 6, row: 2)
```

```
func crossBridge() {  
    turnRight()  
    move(distance: 4)  
    collectGem()  
    turnLeft()  
    turnLeft()  
    move(distance: 4)  
    turnRight()  
}
```

```
for i in 1..3 {  
    move(distance: 2)  
    toggleSwitch()  
    crossBridge()  
}
```



## World Building: Making Your Own Portals

```
let greenPortal = Portal(color: .green)
world.place(greenPortal, atStartColumn: 1,
startRow: 5, atEndColumn: 5, endRow: 1)
```

```
var gemCounter = 0
while gemCounter < 8 {
    moveForward()
    if gemCounter == 4 {
        turnLeft()
        turnLeft()
    } else {
        turnLeft()
    }
    moveForward()
    collectGem()
    gemCounter = gemCounter + 1
    turnLeft()
    turnLeft()
}
```

## World Building: Reach for the Stairs

N/A



## World Building: Making Your Own Portals

```
let greenPortal = Portal(color: .green)
world.place(greenPortal, atStartColumn: 1,
startRow: 5, atEndColumn: 5, endRow: 1)
```

```
var gemCounter = 0
while gemCounter < 8 {
    moveForward()
    if gemCounter == 4 {
        turnLeft()
        turnLeft()
    } else {
        turnLeft()
    }
    moveForward()
    collectGem()
    gemCounter = gemCounter + 1
    turnLeft()
    turnLeft()
}
```

## World Building: Reach for the Stairs

N/A



## World Building: Floating Islands

```
world.place(Stair(), facing: south, atColumn: 3,  
row: 1)
```

```
world.place(Stair(), facing: south, atColumn: 3,  
row: 3)
```

```
world.place(Stair(), facing: west, atColumn: 1,  
row: 4)
```

```
world.place(Stair(), facing: west, atColumn: 1,  
row: 6)
```

```
world.place(Stair(), facing: east, atColumn: 5,  
row: 6)
```

```
world.place(Stair(), facing: north, atColumn: 2,  
row: 7)
```

```
world.place(Stair(), facing: north, atColumn: 4,  
row: 7)
```

```
var switchCounter = 0
```

```
move(distance: 4)
```

```
turnLeft()
```

```
move(distance: 3)
```

```
turnRight()
```

```
while switchCounter < 9 {
```

```
    while !isBlocked {
```

```
        toggleSwitch()
```

```
        moveForward()
```

```
    }
```

```
    turnRight()
```

```
    move(distance: 2)
```

```
    turnLeft()
```

```
    move(distance: 2)
```

```
    turnRight()
```

```
}
```



## World Building: Floating Islands

```
world.place(Stair(), facing: south, atColumn: 3,  
row: 1)
```

```
world.place(Stair(), facing: south, atColumn: 3,  
row: 3)
```

```
world.place(Stair(), facing: west, atColumn: 1,  
row: 4)
```

```
world.place(Stair(), facing: west, atColumn: 1,  
row: 6)
```

```
world.place(Stair(), facing: east, atColumn: 5,  
row: 6)
```

```
world.place(Stair(), facing: north, atColumn: 2,  
row: 7)
```

```
world.place(Stair(), facing: north, atColumn: 4,  
row: 7)
```

```
var switchCounter = 0
```

```
move(distance: 4)
```

```
turnLeft()
```

```
move(distance: 3)
```

```
turnRight()
```

```
while switchCounter < 9 {
```

```
    while !isBlocked {
```

```
        toggleSwitch()
```

```
        moveForward()
```

```
    }
```

```
    turnRight()
```

```
    move(distance: 2)
```

```
    turnLeft()
```

```
    move(distance: 2)
```

```
    turnRight()
```

```
}
```



**World Building: Build a Loop**

N/A

**World Building: Building It All**

N/A

**World Building: Create Your World**

N/A

**\*Some puzzles may have multiple solutions**



**World Building: Build a Loop**

N/A

**World Building: Building It All**

N/A

**World Building: Create Your World**

N/A

**\*Some puzzles may have multiple solutions**

