Challenge: Change all elements of the world to create your own puzzle!

In addition to placing new blocks, stairs, and portals, you can also add gems and switches.

> Adding gems and switches
>
> Just like when you add a block, you use the `place` method on `world` to place gems and switches.
>
> ```
> world.place(Gem(), atColumn: 2, row: 3)
> world.place(Switch(), atColumn: 3, row: 4)
> ```

The shortcut bar contains the methods available on the `world` instance. Use these methods to create your own puzzle—add your own characters, experts, gems, portals, and more! Be creative, and have fun!

```
let ash = Character()
let totalGems = 2
let totalSwitches = 2
var gemCounter = 0
var switchCounter = 0

func addGems() {
    world.place(Gem(), atColumn: 7, row: 6)
    world.place(Gem(), atColumn: 0, row: 0)
}

func addSwtiches() {
    world.place(Switch(), atColumn: 0, row: 6)
    world.place(Switch(), atColumn: 7, row: 0)
}

func addBlocks() {
    world.place(Block(), atColumn: 7, row: 4)
    world.place(Block(), atColumn: 7, row: 4)
    world.place(Block(), atColumn: 7, row: 3)
    world.place(Block(), atColumn: 7, row: 2)
    world.place(Block(), atColumn: 7, row: 1)
    world.place(Block(), atColumn: 7, row: 0)
```

```swift
    world.place(Block(), atColumn: 5, row: 0)
    world.place(Block(), atColumn: 6, row: 0)
    world.place(Block(), atColumn: 3, row: 6)
    world.place(Block(), atColumn: 0, row: 6)

}

func addStairs() {
    world.place(Stair(), facing: east, atColumn: 1, row: 0)
    world.place(Stair(), facing: north, atColumn: 0, row: 2)
    world.place(Stair(), facing: south, atColumn: 0, row: 3)
    world.place(Stair(), facing: east, atColumn: 1, row: 6)
    world.place(Stair(), facing: west, atColumn: 3, row: 6)
    world.place(Stair(), facing: west, atColumn: 4, row: 6)
    world.place(Stair(), facing: south, atColumn: 7, row: 4)
    world.place(Stair(), facing: south, atColumn: 7, row: 3)
}

func addCharacter() {
    world.place(ash, facing: south, atColumn: 3, row: 2)
}

addGems()
addBlocks()
addSwtiches()
addStairs()
addCharacter()
while (gemCounter < totalGems) && (switchCounter < totalSwitches) {
    if ash.isOnGem {
        ash.collectGem()
        gemCounter += 1
    }else if ash.isOnClosedSwitch {
        ash.toggleSwitch()
        switchCounter += 1
    }else if ash.isBlocked {
        ash.turnLeft()
    }else {
        ash.moveForward()
    }
}
```