Parrot Mambo Swift Playground Command Reference

Take-off / Landing

```
takeOff()
land()
```

Dependencies:
Maybe you'll put a little wait between takeOff() and land() in order to make the drone hover in the air for a few seconds.

Setting the Speed

Sets the speed for the move command.

```
droneSpeed = Int
```

Moving Along X- and Y-Axis

Simple Movement

```
move(direction: MoveDirection. [ down | up | left | right | forward |
backward ], duration: Int)
```

Complex Movement

```
move(pitch: Int, roll: Int, gaz: Int, yaw: Int, duration: Int)
pitch: MoveDirection. [ forward | backward ] (forward > 0, backward <0)
roll: MoveDirection. [ left | right ] (right > 0, left < 0)
gaz: MoveDirection. [ up | down ] (up > 0, down < 0)
yaw: TurnDirection. [ left | right ] (right > 0, left < 0)
```

Conditional Move

```
move(direction: MoveDirection. [ down | up | left | right | forward |
backward ])

stopMoving()
```

Dependencies:
The Mambo has to `takeOff()` first. At the end it needs to `land()`.
Moves at the speed that is set by the droneSpeed variable.
The conditional `move()` has to be ended by the `stopMoving()` command.

Turning Along the Z-Axis

```
turn(direction: TurnDirection. [ left | right ], angle: Int)
```

angle: 0 - 180 (If angle is greater than 180, the Mambo will 180°.)

Dependencies:
The Mambo has to `takeOff()` first. At the end it needs to `land()`.

Flip Along X- and Y-Axis

```
flip(direction: FlipDirection. [ front | back | right | left ])
```

Dependencies:
The Mambo has to `takeOff()` first. At the end it needs to `land()`.
Battery has to be charged at least 20%.
No accessories are mounted.

Operating the Grabber

```
openGrabber()
closeGrabber()
```

Dependencies:
The grabber is mounted and connected to the Mambo.

Operating the Camera

```
takePicture()
```

Dependencies:
The Mambo has to `takeOff()` first. At the end it needs to `land()`.

Events

In the File
`Parrot Education.playgroundbook/Content/Sources/motionDetector.swift`
the return values or the `waitNextMotionEvent()` are defined as an enum.

```
waitNextMotionEvent() -> MotionEvent
```

```
MotionEvent = [ .tiltForward | .tiltBackward | .tiltRight | .tiltLeft
| .shakeUp | .shakeDown]
```

Parrot Mambo Swift Playground Tutorial

Preface

Limiting your creativity / setting rails

It is good to know that Swift Playgrounds sometimes limits you to a subset of the possible solutions to a problem. It can happen that you are able to program syntactically correct answer but the Swift Playground just gives you an error message. For example it would be totally fine to take a picture even if the Mambo hasn't taken off. But you get an error message telling you, that the Mambo needs the `takeOff()` command first. This is not because of a bad programmer or a bug, but to make you stick to the task and limit distraction.

@@@X- , Y- and Z- Axis
@@@Which Mambo Bundle

Chapter 1
1.2 - TakeoffLand

1.2.1 - Basic Version

```
takeOff()
wait(4)
land()
```

1.3 - UpDown

1.2.1 - Basic Version

```
takeOff()
move(direction: MoveDirection.up, duration: 1)
wait(1)
move(direction: MoveDirection.down, duration: 2)
land()
```

1.4 - LeftRight

1.4.1 - Basic Version

```
takeOff()
move(direction: MoveDirection.left, duration: 1)
wait(1)
move(direction: MoveDirection.right, duration: 1)
wait(1)
land()
```

1.5 - Challenge: Vertical Square

1.5.1 - Basic Version

```
func verticalSquare() {
    move(direction: MoveDirection.left, duration: 1)
    move(direction: MoveDirection.up, duration: 1)
    move(direction: MoveDirection.right, duration: 1)
    move(direction: MoveDirection.down, duration: 1)
}

takeOff()
verticalSquare()
land()
```

1.6 - ForwardBackward

1.6.1 - Basic Version

```
takeOff()
move(direction: MoveDirection.forward, duration: 1)
move(direction: MoveDirection.backward, duration: 1)
land()
```

1.7 - Rotation

1.7.1 - Basic Version

```
takeOff()
turn(direction: TurnDirection.right, angle: 90)
turn(direction: TurnDirection.left, angle: 90)
land()
```

1.8 - Challenge

1.8.1 - Basic Version

```
takeOff()
move(direction: MoveDirection.forward, duration: 1)
turn(direction: TurnDirection.right, angle: 90)
move(direction: MoveDirection.forward, duration: 1)
turn(direction: TurnDirection.right, angle: 90)
move(direction: MoveDirection.forward, duration: 1)
turn(direction: TurnDirection.right, angle: 90)
move(direction: MoveDirection.forward, duration: 1)
turn(direction: TurnDirection.right, angle: 90)
land()
```

1.8.2 - using a for-loop

```
func horizontalSquare() {
    for n in 1...4 {
        move(direction: MoveDirection.forward, duration: 1)
        turn(direction: TurnDirection.right, angle: 90)
    }

}

takeOff()
horizontalSquare()
land()
```

1.9 - SpeedTimeDistance

1.9.1 - Basic Version

```
takeOff()
move(direction: MoveDirection.up, duration: 1)
droneSpeed = 10
move(direction: MoveDirection.forward, duration: 2)
move(direction: MoveDirection.backward, duration: 2)
land()
```

1.10 - Flips

1.10.1 - Basic Version

```
takeOff()
```

```
flip(direction: FlipDirection.front)
flip(direction: FlipDirection.back)
flip(direction: FlipDirection.right)
flip(direction: FlipDirection.left)
land()
```

2.1 - Complex Moves

2.1.1 - Basic Version

```
takeOff()
move(pitch: 0, roll: 0, gaz: 0, yaw: -20, duration: 2)
land()
```

2.2 - Spiral

2.1.1 - Basic Version

```
func twisterUp() {
    //#-editable-code Add commands to your function
    move(pitch: 20, roll: 20, gaz: 10, yaw: 80, duration: 5)
    //#-end-editable-code
}

func twisterDown() {
    //#-editable-code Add commands to your function
    move(pitch: 20, roll: -20, gaz: -10, yaw: 80, duration: 5)
    //#-end-editable-code
}

func verticalTwister() {
    //#-editable-code Add commands to your function
    twisterUp()
    //turn(direction: TurnDirection.right, angle: 180)
    twisterDown()
    //#-end-editable-code
}

//#-editable-code Tap to enter code
takeOff()
//move(direction: MoveDirection.up, duration: 2)
verticalTwister()
land()
//#-end-editable-code
```

2.3 - Conditional Move

2.1.1 - Basic Version

```
takeOff()
droneSpeed = 50
move(direction: MoveDirection.forward)
wait(2)
stopMoving()
move(direction: MoveDirection.backward)
wait(2)
stopMoving()
land()
```

2.4 - Grabber

2.1.1 - Basic Version

```
takeOff()
openGrabber()
wait(2)
closeGrabber()
wait(2)
land()
```

2.5 - Camera

2.1.1 - Basic Version

```
takeOff()
takePicture()
land()
```

2.6 - Challenge

2.1.1 - Basic Version

```
func pictureRow(){
    for _ in 1...4 {
        move(direction: MoveDirection.forward, duration: 1)
        //takePicture()
    }
}
func pictureSquare(){
    for _ in 1...4{
        //takePicture()
        pictureRow()
        move(direction: MoveDirection.right, duration: 1)
        move(direction: MoveDirection.backward, duration: 2)
    }
    move(direction: MoveDirection.right, duration: 2)
}

takeOff()
droneSpeed = 50
move(direction: MoveDirection.up, duration: 1)
```

```
pictureSquare()
land()
```

2.7 - Gyroscope

2.1.1 - Basic Version

```
takeOff()
// run forever
while true {
    let event = waitNextMotionEvent()

    switch event {
    case .tiltForward:
        move(direction: MoveDirection.up, duration: 1)
    case .tiltBackward:
        move(direction: MoveDirection.down, duration: 1)
    default:
        break
    }
}
```

2.8 - iPad-Remote Control

2.1.1 - Basic Version

```
takeOff()
// run forever
while true {
    let event = waitNextMotionEvent()

    switch event {
    case .tiltForward:
        move(direction: MoveDirection.forward, duration: 1)
    case .tiltBackward:
        move(direction: MoveDirection.backward, duration: 1)
    case .tiltLeft:
        move(direction: MoveDirection.left, duration: 1)
    case .tiltRight:
        move(direction: MoveDirection.right, duration: 1)
    case .shakeUp:
        move(direction: MoveDirection.up, duration: 1)
    case .shakeDown:
        move(direction: MoveDirection.down, duration: 1)
    default:
        break
    }
}
```