Debugging

Debugging

“Xcode’s debugger (called LLDB) is the fundamental tool that will help you find these bugs and fix them.

Xcode Console

Buggy project

run

See output:

2016-08-24 12:52:38.463 Buggy[1961:47078] -[Buggy.ViewController buttonTapped:]:
unrecognized selector sent to instance 0x7ff6db708870”

Debugging

```
@IBAction func buttonTapped(_ sender: UIButton) {
    print("Called buttonTapped(_:)")
}
```

Caveman Debugging

caveman debugging

strategically placing print() calls in your code to verify that functions and methods are being called (and called in the proper sequence) and to log variable values to the console to keep an eye on important data.

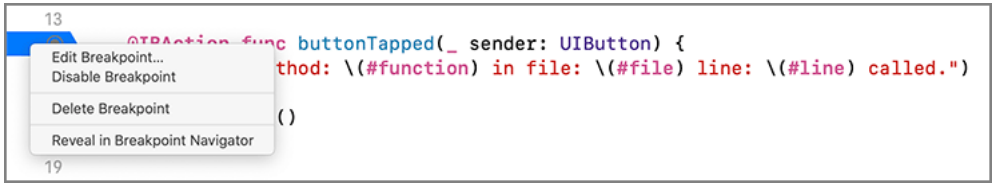
```
“@IBAction func buttonTapped(_ sender: UISwitch) {
    print("Called buttonTapped(_:)")
    // Log the control state:
    print("Is control on? \(sender.isOn)")
}
```

Breakpoints

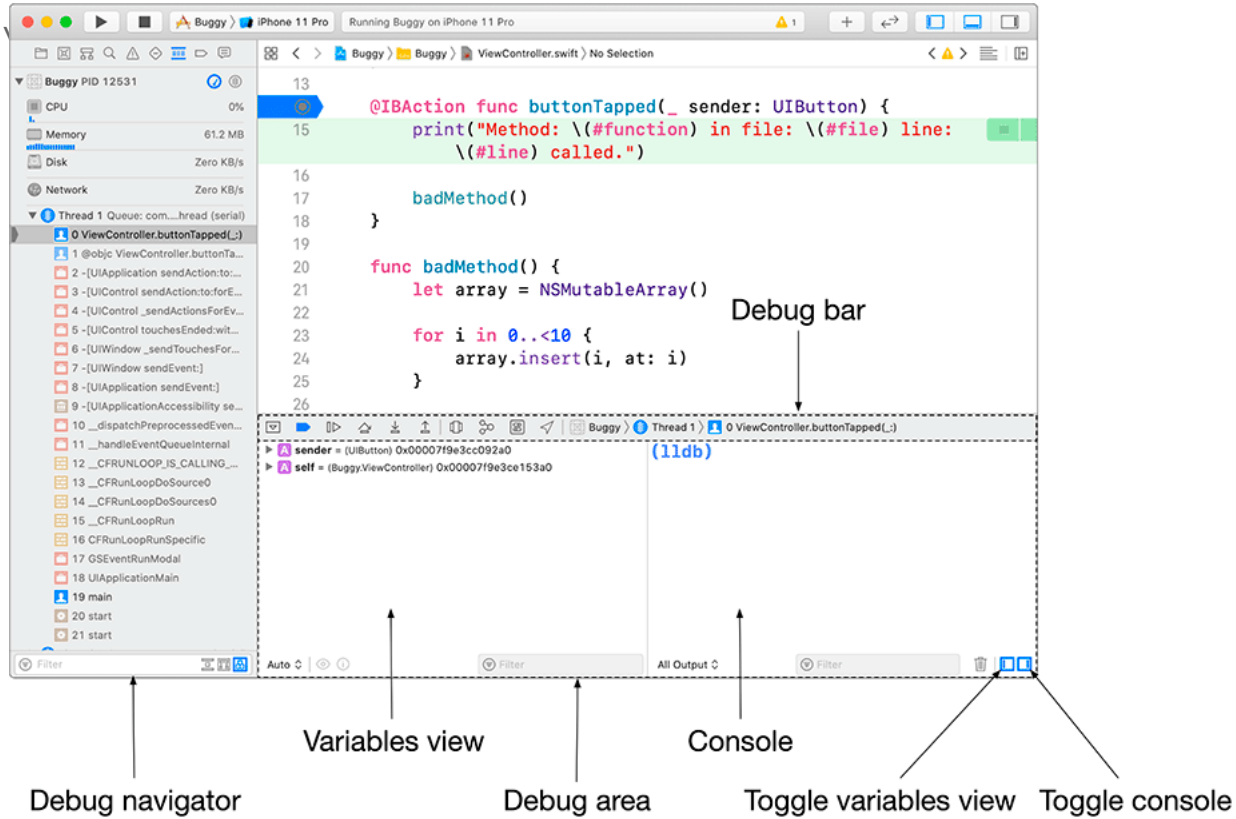
ViewController.swift

```
13
14      @IBAction func buttonTapped(_ sender: UIButton) {
15          print("Method: \(#function) in file: \(#file) line: \(#line) called.")
16
17          badMethod()
18      }
19
```

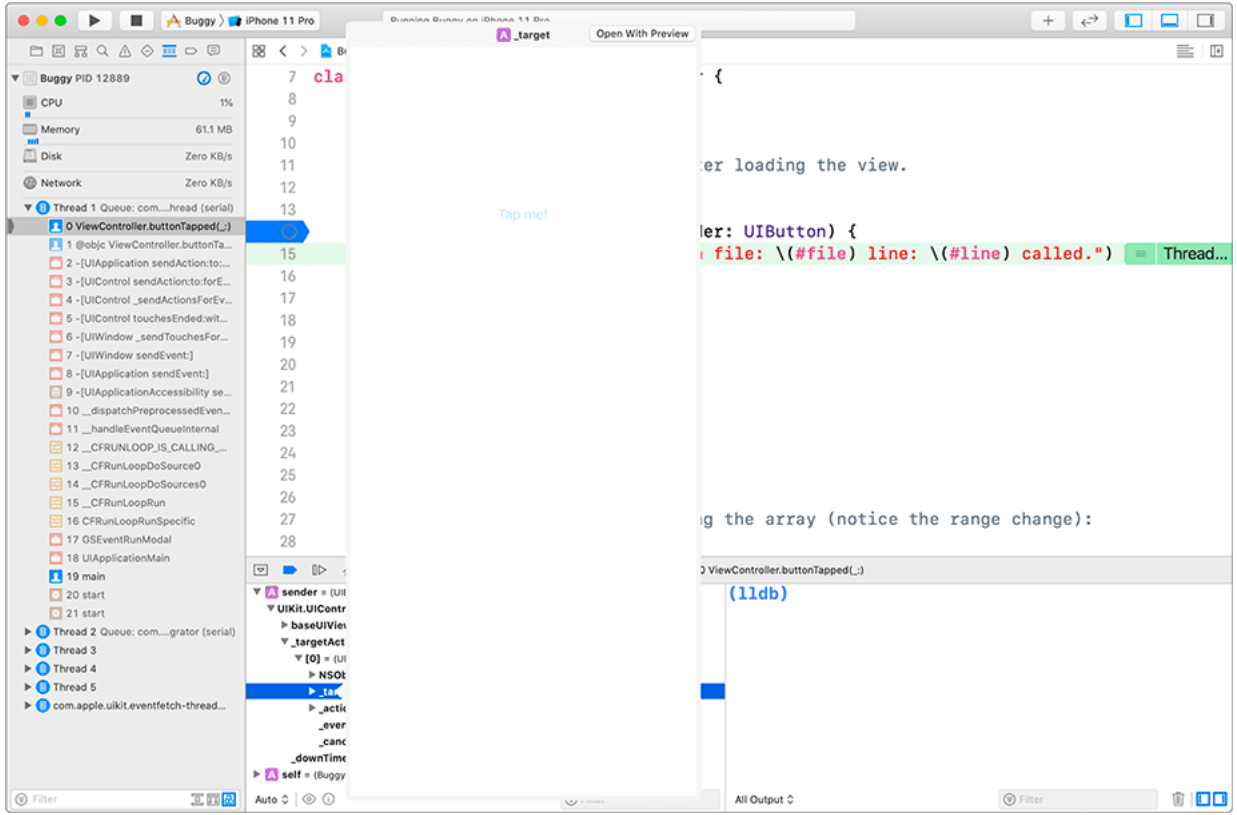
```
13
14      @IBAction func buttonTapped(_ sender: UIButton) {
15          print("Method: \(#function) in file: \(#file) line: \(#line) called.")
16
17          badMethod()
18      }
19
```



Breakpoints

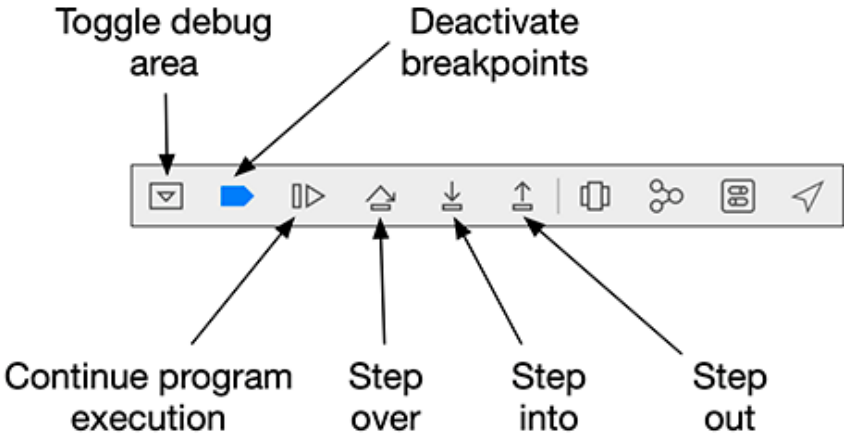


Breakpoints



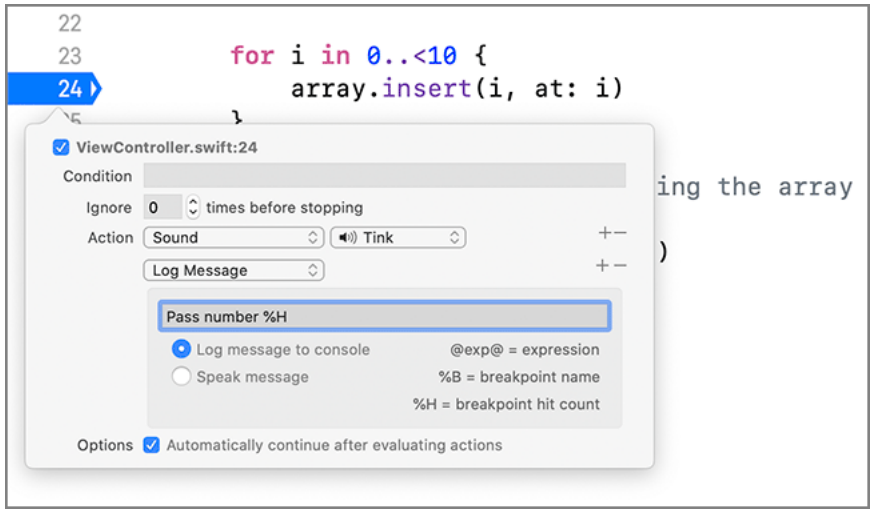
Breakpoints

ViewController.swift



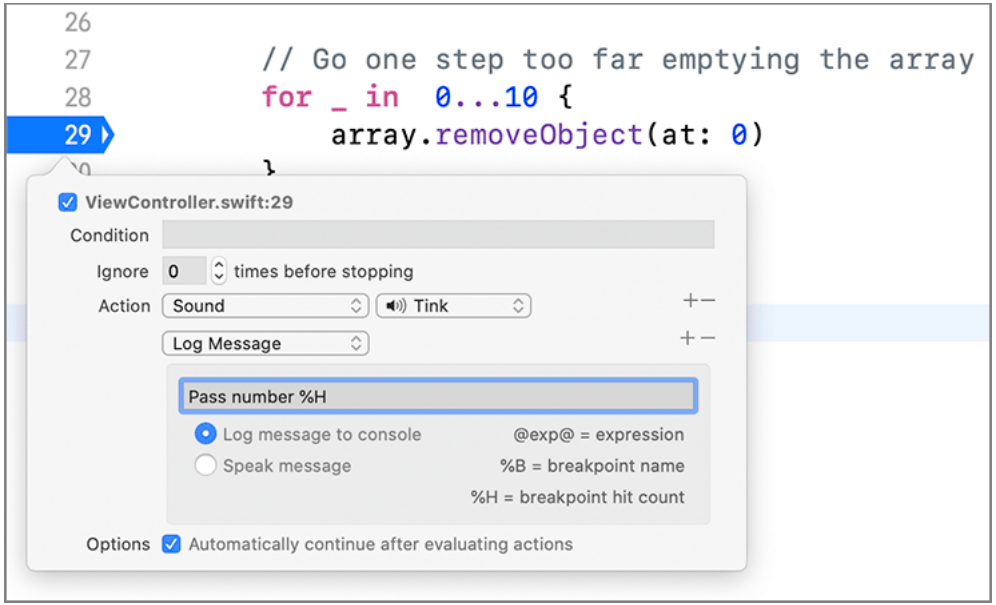
Breakpoints

ViewController.swift



Breakpoints

ViewController.swift



Breakpoints

```
“func badMethod() {
    let array = NSMutableArray()

    for i in 0..<10 {
        array.insert(i, at: i)
    }

    // Go one step too far emptying the array (notice the range change):
    for _ in 0...10 {
    for _ in 0..<10 {
        array.removeObject(at: 0)
    }
}
```