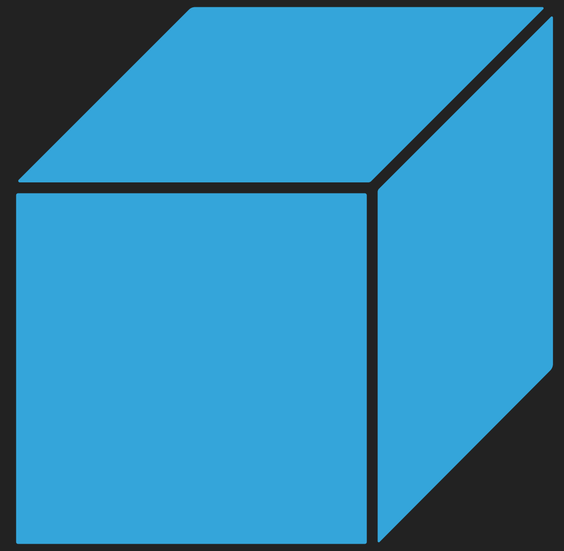
ANGELO CAMMALLERI

REVERSE ENGINEERING UIKIT AS AN APP DEVELOPER

WHY REVERSE ENGINEERING

- The iOS eco system revolves around blackboxes
- Behavior can be undocumented
- Bugs in system frameworks can occur
- Peaking inside the box can give valuable insights



HOW TO DO IT

- No need to be a 10x developer to use reverse engineering
- Modern tools assist in the process
- Hopper Disassembler will be used in this talk
- Try it out for free, you just can’t save your investigation in Hopper

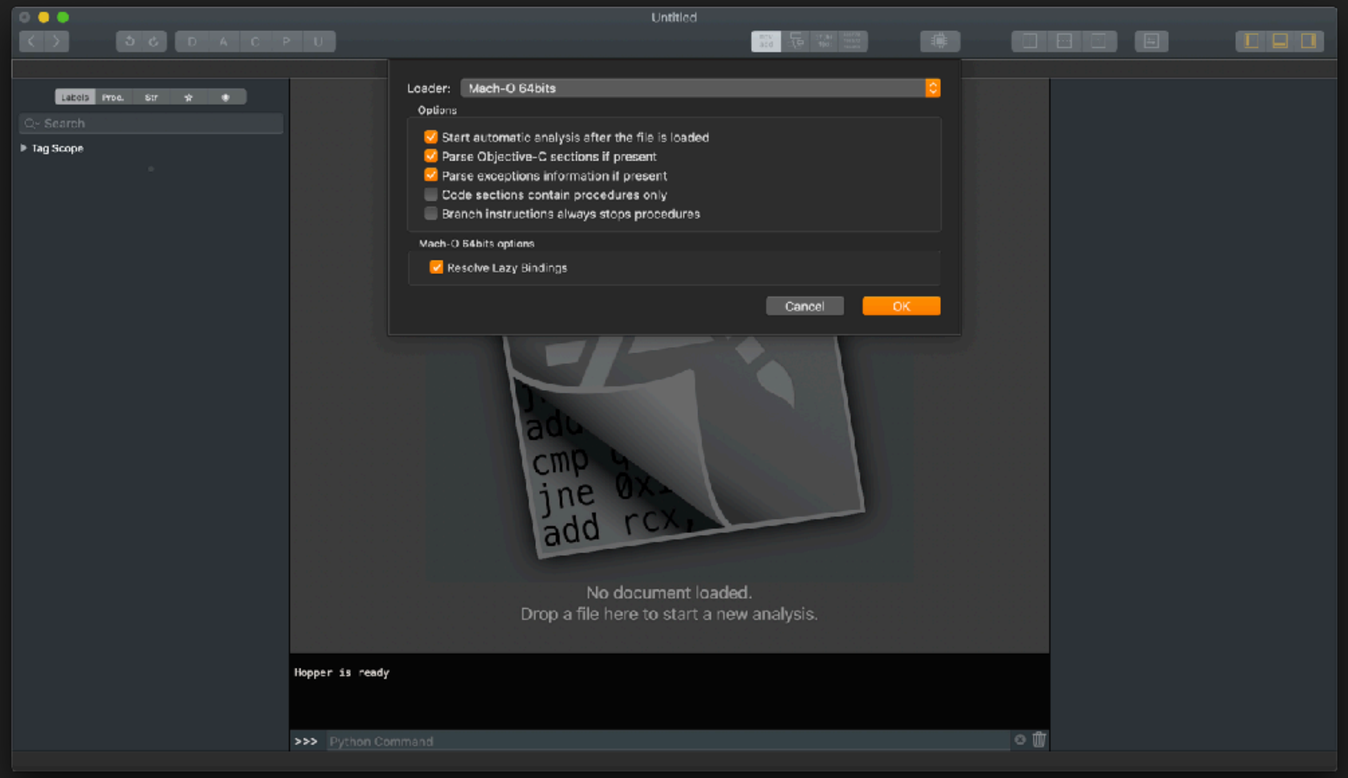
OPENING THE BOX

- To reverse engineer UIKit we need it's binary
- The binary location can change between iOS and Xcode versions
- Location for Xcode 11:

```
/APPLICATIONS/XCODE.APP/CONTENTS/DEVELOPER/
PLATFORMS/IPHONEOS.PLATFORM/LIBRARY/DEVELOPER/
CORESIMULATOR/PROFILES/RUNTIMES/IOS.SIMRUNTIME/
CONTENTS/RESOURCES/RUNTIMEROOT/SYSTEM/LIBRARY/
PRIVATEFRAMEWORKS/UIKITCORE.FRAMEWORK/UIKITCORE
```

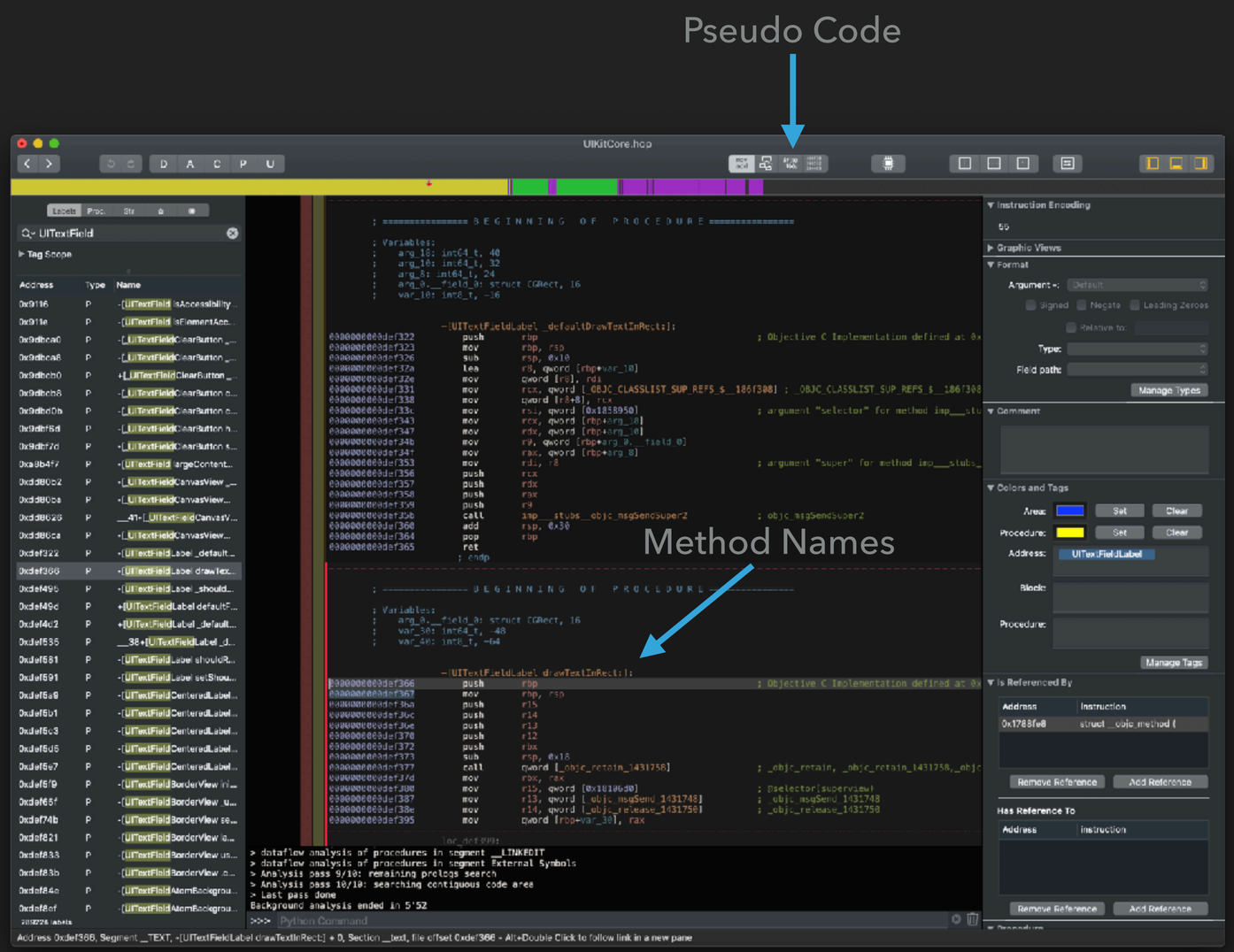
HOPPER

- The UI of Hopper should feel familiar to Xcode users
- After selecting the binary we leave the defaults unchanged
- Lean back while Hopper analyses UIKit 🍹



INSIDE UIKIT

- After the analysis we are greeted by a screen like this
- Assembly can be hard to read
- But we have method names!
- Hopper can transform this into pseudo code with one click



PSUEDO CODE YOUR WAY IN

- In this format everything is instantly more readable
- Ifs can be understood
- We can see calls to other methods

```
/* @class UITextFieldLabel */
-(void)drawTextInRect:(struct CGRect)arg2 {
    var_8 = arg2;
    rax = [self retain];
    rbx = rax;
    var_30 = rax;
    goto loc_def399;

loc_def399:
    r12 = [[rbx superview] retain];
    [rbx release];
    if (r12 == 0x0) goto loc_def432;

loc_def3b8:
    rbx = r12;
    if ((objc_opt_isKindOfClass(r12, objc_opt_class(@class(UITextField))) & 0x1) == 0x0) goto loc_def399;

loc_def3d6:
    rbx = [r12 retain];
    rcx = *(&var_8 + 0x10);
    [r12 _drawTextInRect:var_30 forLabel:rcx];
    [rbx release];
    [rbx release];
    return;

loc_def432:
    rbx = var_30;
    if ([rbx shouldRenderWithoutTextField] != 0x0) {
            var_40 = rbx;
            *(&var_40 + 0x8) = *_OBJC_CLASSLIST_SUP_REFS_$_;
            [[&var_40 super] drawTextInRect:var_8];
            rsp = (rsp - 0x20) + 0x20;
    }
    return;
}
```

DEMO

SOURCES

- “Reverse-Engineering mit Hopper für Fortgelaufene“ by Max Seelemann - https://macoun.de/video2018/gs7.php
- Step by step guide for more in depth reverse engineering with Hopper and Xcode - https://github.com/bartcone/reverse-engineering-blog
- Hopper Disassembler - https://www.hopperapp.com