mdevcon 2014

ADVANCED DEBUGGING

with **XCode 5 & Friends**

KiGi

Instructor:

Kendall Helmstetter Gelner

@kendalldevdiary

Kendall.Gelner@KiGiSoftware.com

materials found:

https://github.com/KiGi/AdvancedDebuggingCode

WHO AM I

- Full-Time iPhone developer since release of SDK.
- Independent consultant for iOS development.
- Over two decades programming experience from corporate to end user.
- Over 50k Reputation on StackOverflow

THINGS OF NOTE

- All files included on GitHub:

  https://github.com/KiGi/AdvancedDebuggingCode

- XCode tips, tricks and discoveries on twitter

  @kendalldevdiary

- Also make sure to watch WWDC videos 413 (Advance Debugging With LLDB) & 404 (Debugging With XCode 5) - this talk builds on and adds to many ideas there.

ON DEBUGGING

"most people who are good at debugging are also good programmers, but a minority of good programmers are good at debugging."
--http://programmingliterate.tumblr.com

"It is difficult not to wonder whether that combination of elements which produces a machine for labor does not create also a soul of sorts, a dull resentful metallic will, which can rebel at times."
--Pearl S. Buck

"Once a new technology rolls over you, if you're not part of the steamroller, you're part of the road."
--Stewart Brand

THE PHILOSOPHY

Debugging is about giving you many paths of visibility into your application that gather enough information for inspiration to strike.

THREE SECTIONS

- Simulation, Instruments
- LLDB and deep inspection/dynamic manipulation.
- Third party tools, Core Data debugging at end.

Demos for each tool, brief Hands-on labs between sections!

DEBUGGING APP

- Some use of ARC, some not (FontList and AFNetworking)
- Buggy for a reason, careful using code from it!

SIMULATOR & INSTRUMENTS

SIMULATOR FEATURES

- The simulator has some helpful debugging features added over time:

  Slow Animation

  Graphics Blending

  Location

SIMULATING LOCATION

- You can set simulated location in the simulator
- You can have XCode run through a GPX file simulating location.
- XCode can also help you make a GPX file.
- You can use the Automator instrument to more accurately simulate location (includes direction and speed).

INSTRUMENT: XCODE DEBUGGER

- Simplified/Summarized view of data you would get from many Instruments.

- Found in sidebar above stack trace when running.

- Click on CPU/Memory for full page of info.

- Can launch full Instrument from summary panels, but use normal profile launch also.

INSTRUMENT:
INSTRUMENT BASICS

- Small **i** can reveal many interesting settings.
- Also note options in left side bar.
- Can always stack more instruments in.
- Can add flags, narrow regions of interest.

INSTRUMENT: TIME PROFILER

- Great way to check why an app is busy - or why it's not.
- Helps you focus on worst performance first.
- Check out Core view to ensure proper use of cores on device.
- XCode 5 simpler Profiler view can act as guideline for what needs Time Profiler.
- Record Waiting Threads can show causes for pauses.

INSTRUMENT: OBJECT ALLOC (MEMORY)

- Instrument used for memory diff (heapshot).
- Usually good to look for highest level memory user.
- XCode 5 simpler Profiler view can illuminate potential memory problems while running, scan it as you run.

LEAKS VS CLOGS

- ARC prevents common memory issues, but not all!
- Leaks are like a murder mystery.
- You can also "clog" memory with memory you didn't know you were still using.
- With ARC, mostly use Object Alloc instrument with Heapshot.

INSTRUMENT: CORE ANIMATION

- Measurable UI performance on device.
- Useful visual inspection of hidden graphics attributes.
- Limited examination of structure for third party apps.

INSTRUMENT: CUSTOM (DTRACE)

- DTrace can be complex, but there are easy paths to use:

  Build New Instrument->(ObjC type, full method signature)

- Great when stacked with other instruments.

- Great for quick call flow information.

- Custom instruments stored in:

  ~/Library/Application Support/Instruments/PlugIns/Instruments

INSTRUMENT: ZOMBIE

- For specialized use, in hunting dow source of crash.
- Not as needed with ARC.
- Fire it up, wait for crash and examine retain/release info.
- A new mystery - who released when they should not have? Or, who maintained a reference that went bad?
- Consider using weak references to fix.

LAB TIME

10 MIN

- Explore Core Animation effects on your favorite app
- Use time profiler and explore time used when scrolling debugging app (or your own)!
- Change location in simulator directly and from XCode.
- Create a custom instrument to track awakeFromNib, try various controls.

ADVANCED LOGGING

BETTER LOGGING

- NSLog always outputs, even for customers.
- Evaluating data for NSLog can slow things down.
- Instead - consider DNSLog, defining away for AppStore builds and altering where debugging goes.
- debugDescription can also be implemented for extra data.
- Consider using blocks for expensive log calculations

ADVANCED LOGGING

- Consider using blocks for expensive log calculations.
- Local notification logging allows for background logging.
- Can route NSLog through things like TestFlight TFLog.

DTRACE LOGGING

- Script to examine call trees without instruments.
- Knowledge of DTrace could open a lot of probing possibilities.
- Note as with DTrace instrument, only works in sim (simulator apps have own PID).
- More info on DTrace:

  http://blog.bignerdranch.com/1907-hooked-on-dtrace-part-1/

LLDB

LLDB - EXPRESSIONS

- Expressions compile real code to run in app.
- Access with "expr" command (GDB used "p" or "po" to evaluate expressions, so can LLDB)
- Define debugger global variables with "$" prefix (use id for complex types)
- Use with continued breakpoints for real-time patching!
- thread return <EXPR> to return from method with expression.

LLDB - USEFUL COMMANDS

- po "recursiveDescription" works on any view to dump view hierarchy.

- if you crash is in objc_msgsend, you can at least find the method name called with:

```
p (char *)$ecx (simulator)

p (char *)$r1 (device)
```

LLDB - USEFUL COMMANDS

- Set property watchpoints (max two on device) with

```
watchpoint set variable self->_myProp
```

- thread until <linenumber> for quick jump (hover stopped working?)

- If LLDB does not "see" a class, try using NSClassFromString.

BREAKPOINTS

CONDITIONAL

- For any expression used in a breakpoint condition, make sure the LLDB command line accepts it first.
- Conditional breakpoints can use any expression.
- With an auto-continue breakpoint, you can add small blocks of expr code to try adding logic on the fly.
- Note you may have to cast return values for every call.
- Can add condition on LLDB defined variable.
- If the condition does not work it will stop every time (now with message!).

DEBUGGER AS AUDIO PROBE

- Covered in older WWDC video, but there is a better why...
- Understand relations between calls in an API, or the flow of data in your app.
- Also, speech (numbers only...)!
- You can use your own sounds placed in ~/Library/Sounds

DATA FORMATTERS

- Use to provide better view on your local variables.
- Can define in XCode variable window:

```
TheTextValueIs is {(NSString*)[$var size]}:s
```

- Also possible to use python summaries. Not same format as XCode.
- Summary Details: **http://lldb.llvm.org/varformats.html**

DATA FORMATTERS

- Can find examples of system formatters (in XCode 4.5+):

XCode.app/Contents/SharedFrameworks/
LLDB.framework/

Resources/Python/lldb

/formatters/objc

DATA FORMATTERS

- Can create custom python summaries
- Easier than it sounds
- Examples at

  http://llvm.org/svn/llvm-project/lldb/trunk/examples/customization/bin-utils/

  Demo!

CUSTOM QUICKLOOK

- You can create quick look for your own classes!
- Return either NSAttributedString or UIImage!
- Add method **-debugQuickLookObject** (returns **ID**)!
- … but only in current beta XCode.
- QuickLook now also shows data formatter summary strings

LAB TIME

15 MIN

- Try setting a debugger variable in one breakpoint and viewing it a different breakpoint.

- Use EXPR on a continue breakpoint in cellForRow to strip out extra text on flickr user name output.

- Add an audio breakpoint to numberOfSections, cellForRow & heightForRow in KGFlickrListVC , and hear how they are called. Is there an issue?

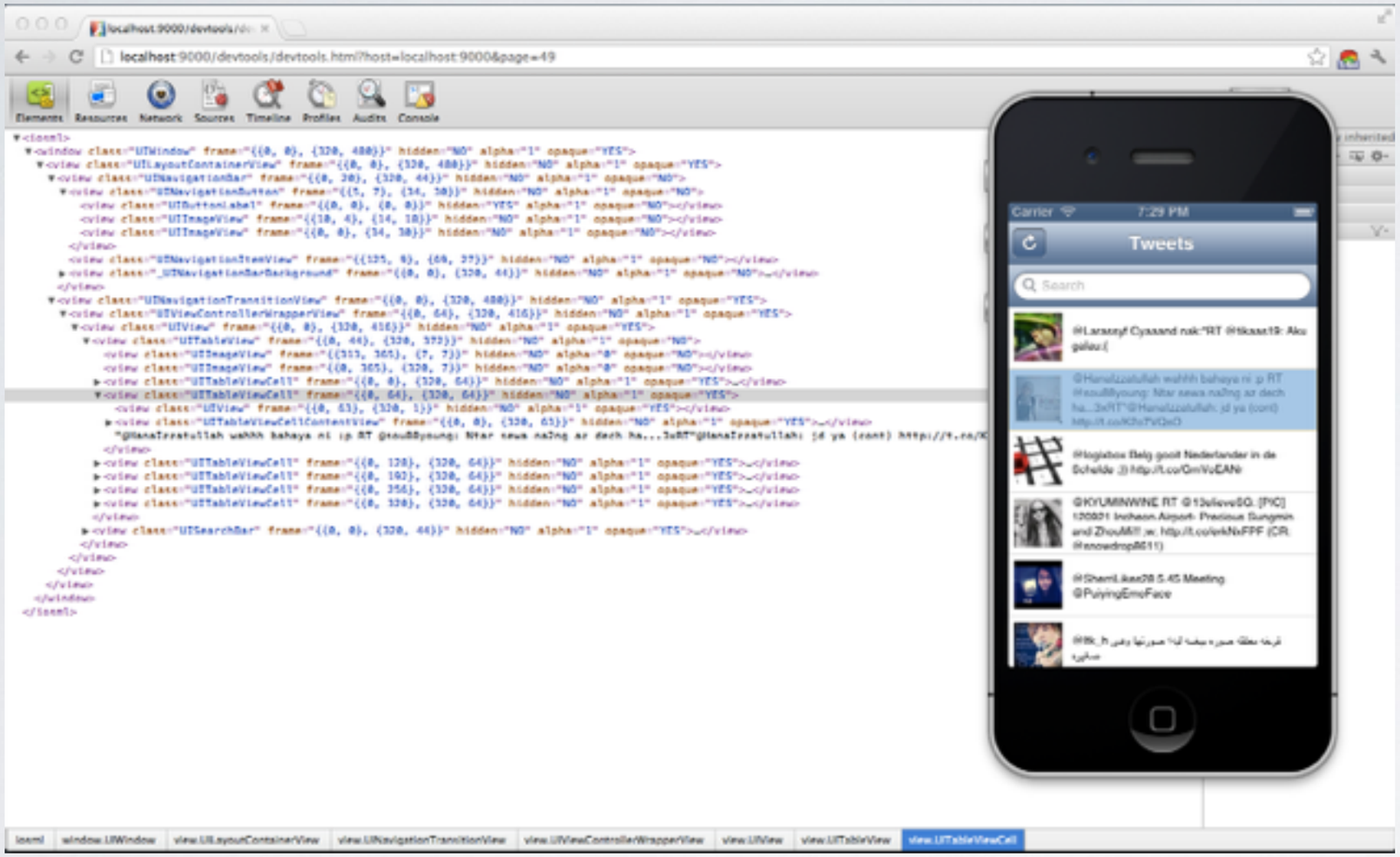
- Successfully have the debugger stop at a cell when the tag count is greater than 10, and speak the index.

THE LARGER WORLD

PONY DEBUGGER

- Included in your "Extra" folder
- Use XCConfig file to configure addition for projects.
- Install XCode command line tools.
- Install & run the server (PonyDebuggerQuickstart.txt)
- Then add code to attach to Pony server.
- Experiment, some things may not work.

PONY DEBUGGER



REVEAL

- Included in your "Extra" folder
- Find at RevealApp.com -Trial version, costs $80 US.
- Shows breakout of views
- Just add framework (embedded in app bundle) and run

NETWORK LINK CONDITIONER

- Download through XCode:

  XCode >Open Developer Tool > More Developer Tools

  Look for "Hardware IO Tools" package (added to "Extra")

- Affects all the apps on the system when enabled!

- Existing settings are OK, but you can make a better version of bad networks...

CHARLES

- Also in "Extra" folder.
- This one costs money, but is worth it (free LONG trial).
- Great reporting of all the traffic details you could wish.
- Set HTTPS hosts to proxy traffic (install Charles local cert)

  http://www.charlesproxy.com/ssl

- Make sure a "curl" or browser call works before you try to code against something or look for bugs.

IPHONE CONFIGURATION UTILITIY

- Download from web (Google for it), also in "Extras" folder
- Can be used to quickly switch to proxy server for traffic examination.
- Can have multiple profiles installed, with only some active.

APPCODE

- Download from web at http://www.jetbrains.com/objc
- No detail here, but worth a look.
- Can switch to XCode and back again, uses same project files.

CYCRYPT

- Runtime examination and changing of application using Javascript.
- From Saurik who brought us MobileSubstrate
- Swizzling framework for apps.
- http://www.cycript.org
- Intro video at: https://www.youtube.com/watch?v=5d1cK0nq4GY

CORE DATA DEBUGGING

- Instruments can help track saves and other operations.
- Also just before a save, look at ManagedObjectContext updatedObjects or insertedObjects to see what will be saved.
- Output from Core Data objects in debugger may truncate strings! Your full string is there.
- Use valueForKey: to get properties out of a Core Data object in the debugger. willAccessValueForKey: triggers fetch, as does "allObjects" on relationships.
- See WWDC2013 session "Core Data Performance Optimiztion and Debugging" (Session 211)

CORE DATA STRUCTURE

- Be careful of deletes, very dangerous.
- isDeleted does not really mean isDeleted.
- Inheritance means flat tables.
- Make everything you can optional, validations dangerous.
- Make use of helper libraries, but don't get too far removed.
- Create new model version for every app store release (if needed).

CORE DATA FILES

- Keep in caches directory if not permanent.
- Auto migrate, destroy if that doesn't work.
- iCloud key/value pretty stable, Core Data documents fiddly.
- You can pre-load databases, copy them into writable directory on app start.
- You can copy DB from the simulator directories to keep different cases around (as long as you have the same version) or transplant databases from device.
- Can query Core Data file with SQLLite tools.

LAB TIME
OPEN

- Install Charles and examine traffic when Flickr is called.

- Install Reveal framework in project and experiment in simulator (or on open network if you dare!).

- Try configuring bad network connection and text Flickr code.

- Try installing a profile on your device to go through the Charles proxy (your IP / port 8888 by default), or set up HTTP proxy on device WiFi connection.

THANKS FOR ATTENDING!

Also consider my XCode5 Plugin talk tomorrow!

Instructor:

Kendall Helmstetter Gelner

@kendalldevdiary

Kendall.Gelner@KiGiSoftware.com

materials found:

https://github.com/KiGi/AdvancedDebuggingCode

RESOURCES

Phone Configuration Utility:
http://support.apple.com/kb/DL1465?viewlocale=en_US&locale=en_US

Mogenerator:
http://rentzsch.github.io/mogenerator/

Pony Debugger:
http://djromero.wuonm.com/pony-debugger/

Java JRE (for Charles):
http://www.oracle.com/technetwork/java/javase/downloads/1880261

Charles Web App:
http://www.charlesproxy.com

Network Link Conditioner Pref Pane:
https://developer.apple.com/downloads/index.action?name=for%20Xcode%20- (Hardware IO Tools link, needs dev login)

Cycrypt:
http://www.cycript.org
Intro video: https://www.youtube.com/watch?v=5d1cK0nq4GY
Dynamic modification of running code.

LLDB:
http://lldb.llvm.org/tutorial.html - tutorial on common commands
http://llvm.org/svn/llvm-project/lldb/trunk/examples/customization/bin-utils/ - python examples
http://stackoverflow.com/questions/7677613/can-anyone-share-a-sample-lldbinit-file - lldbinit examples
http://lldb.llvm.org/varformats.html - LLDB variable formats