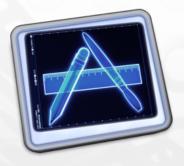
iPhone Application Programming Lecture 8: Instruments

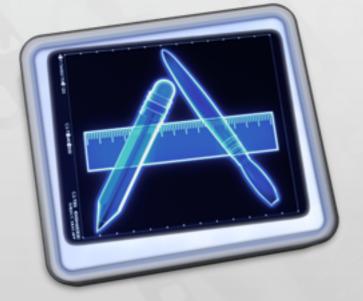
Moritz Wittenhagen Media Computing Group RWTH Aachen University

Winter Semester 2013/2014

http://hci.rwth-aachen.de/iphone







Instruments





Where is the problem?

```
int main(int argc, const char * argv[])
{
    @autoreleasepool
    {
        for(int i = 0; i < HUGE_VAL; i++)
        {
            NSArray *array = @[];
            [[array retain] autorelease];
        }
    }
    return 0;
}</pre>
```





Memory

How much is used? When is it allocated / freed?

• CPU Time

Where is it spent?

How is it distributed between multiple CPUs?

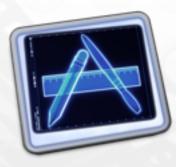


έA7



How is my algorithm doing things? Not: Is it doing them correctly?





Track view Data view

Extended details

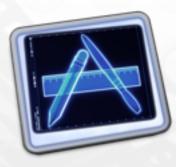
	Instruments15	2
000		O Investore Combol
II 🕘 🗇 🔳 Exciting Bu	ffet : 0 0 0 0 00:00:41 0	Q- Involves Symbol
Record Target	Inspection Range View Library	Search
All Cores	All Processes All Threads	
Instruments	00.00	Extended Detail
6		
🕨 🦣 Time Profiler 🛛 👔		18.0 start
<u> </u>		18.0 main
		18.0 UIApplicationMain
🕨 🛛 🙀 👔 Render Timer 👘		18.0 GSEventRun
	Stadk Depth	18.0 GSEventRunModal
		18.0 CFRunLoopRuninMode
		18.0 CFRunLoopRunSpecific
		18.0CFRunLoopRun
		10.0CFRunLoopDoSource1
		10.0CFRUNLOOP_IS_CALLING
🔯 Time Profiler 🔶 🖨	E Call Tree Call Tree	10.0 PurpleEventCallback
Sample Perspective	Running Time v Self Symbol Name	9.0 _UIApplicationHandleEvent
All Sample Counts	1919.0ms 99.0% 0.0 O b_dispatch_worker_thread2 libdispatch_sim.dylib	9.0 -[UIApplication sendEvent:]
Running Sample Counts	18.0ms 0.9% 0.0 🔤 >start Exciting Buffet Of Improvability 🕤	8.0 -[UIWindow sendEvent:]
V Call Tree		8.0 -[UIWindow _sendTouchesFo
Separate by Thread		8.0 -[UIControl touchesEnded:wi
Invert Call Tree		7.0 -[UIControl(Internal) _sendA
Hide Missing Symbols		7.0 -[UIControl sendAction:to:fo
Hide System Libraries		7.0 -[UIApplication sendAction:t
Show Obj-C Only		7.0 -[UIApplication sendAction:t
Flatten Recursion		7.0 -[NSObject performSelector:
Top Functions Call Tree Constraints		7.0 -[UITabBar(Static) _buttonUp:]
Specific Data Mining		6.0 -[UIControl sendActionsForC
✓ libsystem_c.dylib Charge \$		6.0 -[UIControl(Internal)_sendA
C monystem_coupled counge 1		6.0 -[UIControl sendAction:to:fo
		6.0 -[UIApplication sendAction:t
		6.0 -[UIApplication sendAction:t
		6.0 -[NSObject performSelector:
Symbol Library Restore		6.0 -[UITabBar _sendAction:with
		6.0 -[UIApplication sendAction:t
		6.0 -[UIApplication sendAction:t
		6.0 -[NSObject performSelector:
		6.0 -[UITabBarController_tabBar
		6.0 -[UITabBarController setSele
		6.0 -[UITabBarController_setSel
		6.0 -[UITabBarController transiti
		6.0 -[UITabBarController transiti
		4.0 -[UIViewController beginApp
		=
		==



Track view Data view

Extended details

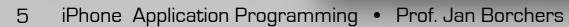
000	Landau and Mr.	
000	Instruments15	¥*
(II) () (Exciting Bu	ffet : () () () () () () () () () () () () ()	Q- Involves Symbol
Record Target	Inspection Range Run 1 of 1 View Library	Search
All Cores	All Processes All Threads	
Instruments	00:00	Extended Detail
8		▼ Heaviest Stack Trace - Q.
🕨 🦚 Time Profiler 🛛 👔		18.0 start
		18.0 main
		18.0 UIApplicationMain
Render Timer 🕧		18.0 GSEventRun
	Stadk Depth	18.0 GSEventRunModal
		18.0 CFRunLoopRunInMode
		18.0 CFRunLoopRunSpecific
		10.0 _CFRunLoopDoSource1
		10.0CFRUNLOOP_IS_CALLING
		10.0 PurpleEventCallback
🔄 Time Profiler 🗢 🗢		9.0 _UIApplicationHandleEvent
	Running Timev Self Symbol Name	9.0 -[UIApplication sendEvent:]
All Sample Counts	1919.0ms 99.0% 0.0 E_dispatch_worker_thread2 libdispatch_sim.dylib	8.0 -[UfWindow sendEvent:]
Running Sample Times	18.0ms 0.9% 0.0 💼 🕨 start Exciting Buffet Of Improvability 🔿	8.0 -[UlWindow_sendTouchesFo
▼ Call Tree		8.0 -[UIControl touchesEnded.wi
Separate by Thread		7.0 -[UIControl(Internal)_sendA
Invert Call Tree Hide Missing Symbols		7.0 -[UIControl sendAction:to:fo
Hide System Libraries		7.0 -[UIApplication sendAction:t
Show Obj-C Only		7.0 -[UIApplication sendAction:t
Flatten Recursion		7.0 -[NSObject performSelector
Top Functions		7.0 -[UITabBar(Static)_buttonUp:]
Call Tree Constraints		6.0 -[UIControl sendActionsForC
V Specific Data Mining		6.0 -[UIControl(Internal)_sendA
Slibsystem_c.dylib Charge :		6.0 -[UIControl sendAction.to:fo
		6.0 -[UIApplication sendAction:t
		6.0 -[UIApplication sendAction:t
		6.0 -[NSObject performSelector
Symbol Library Restore		6.0 -[UITabBar_sendAction:with
		6.0 -[UIApplication sendAction:t
		6.0 -[UIApplication sendAction:t
		6.0 -[NSObject performSelector
		6.0 -[UITabBarController_tabBar
		6.0 -[UITabBarController setSele
		6.0 -[UITabBarController_setSel
		6.0 -[UITabBarController transiti
		6.0 -[UITabBarController transiti
		4.0 -[UIViewController beginApp
		TTA A DATE OF A DATE



Track view Data view

Extended details

	Instrum	ents15	2
(II) (C) Exciting But	ffet : 0 0 0 00:0	10:41 0	Q- Involves Symbol
Record Target	4 Run	1 of 1 > View Library	Search
All Cores	all Processes	All Threads :	
Instruments	00.00	U01:00	Extended Detail
			V Heaviest Stack Trace &- Q
Fime Profiler 🕐			18.0 main
			18.0 UIApplicationMain
A CONTRACTOR OF			18.0 GSEventRun
Render Timer 🕧			18.0 GSEventRunModal
	Stack Denta		18.0 CFRunLoopRunInMode
			18.0 CFRunLoopRunSpecific
			18.0 _CFRunLoopRun
			10.0CFRunLoopDoSource1
			10.0CFRUNLOOP_IS_CALLING
			10.0 PurpleEventCallback
😧 Time Profiler 🔶	E Call Tree Call Tree		9.0 _UIApplicationHandleEvent
Sample Perspective	Running Time ¥ Self Symbol Name		9.0 -[UIApplication sendEvent:]
 All Sample Counts 	1919.0ms 99.0% 0.0 O ▶_dispatch_worker_thread2		8.0 -[UIWindow sendEvent:]
Running Sample Times	18.0ms 0.9% 0.0 Start Exciting Buffet Of Impre	ovability 📀	8.0 -[UIWindow_sendTouchesFo
V Call Tree			8.0 -[UIControl touchesEnded:wi
Separate by Thread Invert Call Tree			7.0 -(UIControl(Internal)_sendA
Hide Missing Symbols			7.0 -[UIControl sendAction.to:fo
Hide System Libraries			7.0 -[UIApplication sendAction:t
Show Obj-C Only			7.0 -[UIApplication sendAction:t
Flatten Recursion			7.0 -[NSObject performSelector:
Top Functions			7.0 -[UITabBar(Static) _buttonUp:]
▶ Call Tree Constraints			6.0 -[UIControl sendActionsForC
▼ Specific Data Mining			6.0 -[UIControl(Internal)_sendA
libsystem_c.dylib Charge :			6.0 -[UIControl sendAction:to:fo
			6.0 -[UIApplication sendAction:t
			6.0 -[UIApplication sendAction:t
			6.0 -[NSObject performSelector
Symbol Library Restore			6.0 -[UITabBar_sendAction:with
			6.0 -[UIApplication sendAction:t
			6.0 -[UIApplication sendAction:t
			6.0 -[NSObject performSelector:
			6.0 -[UITabBarController_tabBar
			6.0 -[UITabBarController setSele
			6.0 -[UITabBarController_setSel
			6.0 -[UITabBarController transiti
			6.0 -[UITabBarController transiti
			4.0 -[UIViewController beginApp

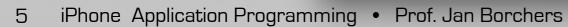




Track view Data view

Extended details

000	Instruments15	,
II C Exciting Buffet : Record Target		Q- Involves Symbol Search
(□ Ξ) All Cores	All Processes All Threads	(III)
Instruments 00:00	\	Extended Detail
Time Profiler	Grack Deskin	 ▼ Heaviest Stack Trace ☆- Q. 18.0 start 18.0 main 18.0 UIApplicationMain 18.0 GSEventRun 18.0 GSEventRunModal 18.0 CFRunLoopRunInMode 18.0 CFRunLoopRunSpecific
		18.0CFRunLoopRun 10.0CFRunLoopDoSource1 10.0CFRUNLOOP_IS_CALLING 10.0CFRUNLOOP_IS_CALLING
🔄 Time Profiler 🔹 😑 Call 1	Tree 🗢 Call Tree	10.0 PurpleEventCallback
-	g Time y Self Symbol Name	9.0 _UIApplicationHandleEvent 9.0 -[UIApplication sendEvent:]
O All Sample Counts 1919.0ms		8.0 -[UlWindow sendEvent.]
	is 0.9% 0.0 🛅 🕨 start Exciting Buffet Of Improvability 🔿	8.0 -[UIWindow_sendTouchesFo
▼ Call Tree		3.0 -[UIControl touchesEnded.wi
Separate by Thread		7.0 -[UIControl(Internal)_sendA
Invert Call Tree Hide Missing Symbols		7.0 -[UIControl sendAction:to:fo
Hide System Libraries		7.0 -[UIApplication sendAction:t
Show Obj-C Only		7.0 -[UIApplication sendAction:t
Flatten Recursion		7.0 –[NSObject performSelector:
Top Functions		7.0 -[UITabBar(Static) _buttonUp:]
Call Tree Constraints Constraints		6.0 -[UIControl sendActionsForC
✓ Specific Data Mining ✓ libsystem_c.dylib Charge :		6.0 -[UIControl(internal)_sendA
a nosystem_c.upito charge		6.0 -[UIControl sendAction:to:fo
		6.0 -[UIApplication sendAction:t
		6.0 -[UIApplication sendAction:t
		6.0 -[NSObject performSelector:
Symbol Library Restore		6.0 -[UITabBar _sendAction:with
		6.0 -[UIApplication sendAction:t
		6.0 -[UIApplication sendAction:t
		6.0 -[NSObject performSelector:
		6.0 -[UITabBarController_tabBar
		6.0 -[UITabBarController setSele
		6.0 -[UITabBarController_setSel
		6.0 -[UITabBarController transiti
		6.0 -[UITabBarController transiti
		4.0 -[UIViewController beginApp
		=

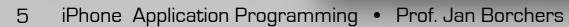




Track view Data view

Extended details

000	Instruments15	2
II 💿 🗇 🔳 Exciting Bu	d Bunlofl b	Q- Involves Symbol
Record Targe	t Inspection Range View Library	Search
All Cores		
Instruments	00.00	
Time Profiler		Heaviest Stack Trace &- Q 18.0 start 18.0 main 18.0 UIApplicationMain 18.0 G5EventRun 18.0 G5EventRun 18.0 G5EventRunModal 18.0 CFRunLoopRunInMode
		18.0 CFRunLoopRunSpecific 18.0CFRunLoopRun 10.0CFRunLoopDoSource1 10.0CFRUNLOOP_JS_CALLING 10.0 PurpleEventCallback
🧟 Time Profiler 🗧 🖨	E Call Tree + Call Tree	9.0 _UIApplicationHandleEvent
Sample Perspective	Running Time y Self Symbol Name	9.0 -[UIApplication sendEvent:]
O All Sample Counts	1919.0ms 99.0% 0.0 E_dispatch_worker_thread2 libdispatch_sim.dylib	8.0 -[UtWindow sendEvent:]
Running Sample Times	18.0ms 0.9% 0.0 🛅 🕨 start Exciting Buffet Of Improvability 🔿	8.0 -[UIWindow_sendTouchesFo
Call Tree Separate by Thread		8.0 -[UIControl touchesEnded:wi
Invert Call Tree		7.0 -[UIControl(Internal)_sendA
Hide Missing Symbols		7.0 -[UIControl sendAction.to:fo
Hide System Libraries		7.0 -[UIApplication sendAction:t
Show Obj-C Only		7.0 -[UIApplication sendAction:t
Flatten Recursion		7.0 -[NSObject performSelector:
Top Functions		7.0 -[UITabBar(Static) _buttonUp:]
Call Tree Constraints		6.0 -[UIControl sendActionsForC
Specific Data Mining		6.0 -[UIControl(Internal)_sendA
libsystem_c.dylib Charge :		6.0 -[UIControl sendAction.to:fo
		6.0 -[UIApplication sendAction:t
		6.0 -[UIApplication sendAction:t
		6.0 -[NSObject performSelector:
Symbol Library Restore		6.0 -[UITabBar_sendAction:with
		6.0 -[UIApplication sendAction:t
		6.0 -[UIApplication sendAction:t
		6.0 -[NSObject performSelector:
		6.0 -[UITabBarController_tabBar
		6.0 -[UITabBarController setSele
		6.0 -[UITabBarController_setSel
		6.0 -[UITabBarController transiti
		6.0 -[UITabBarController transiti
		4.0 -[UIViewController beginApp
		The part of the second
		==





Strategies (Example)

Instruments

Instruments	00:00	\square
Fime Profiler		

• Threads

Threads	00:00		∇
Exciting Buffet Of Improvabi		1.	**
Exciting Buffet Of Improvabi			
Exciting Buffet Of Improvabi			

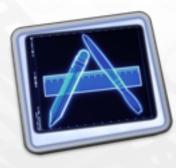
• CPUs

CPUs	00:00	00:00		
Core 0 2.8 GHz Dual Core CPU				
Core 1 2.8 GHz Dual Core CPU		1		

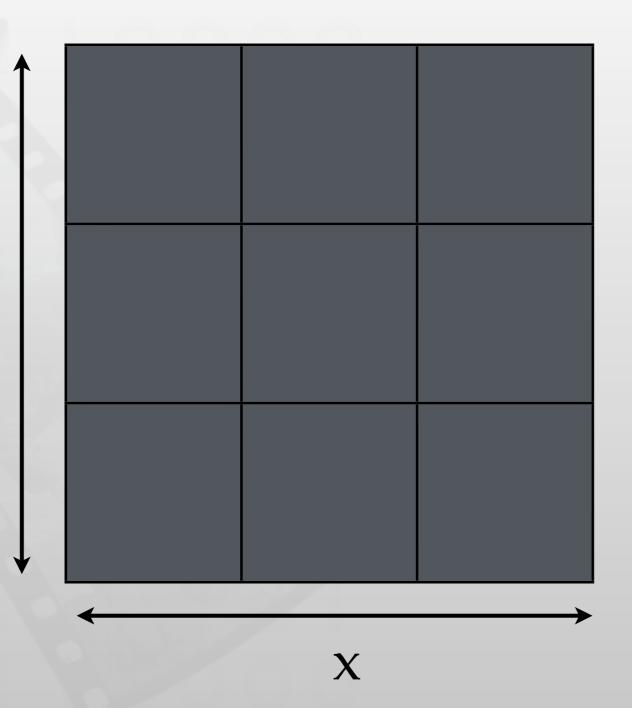




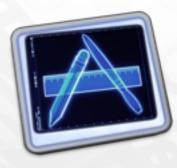
Demo Project



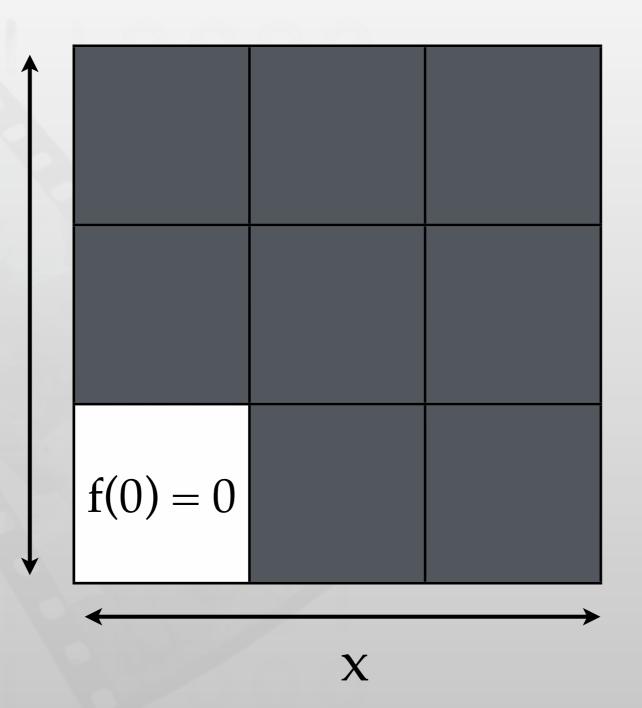
f(x) = x



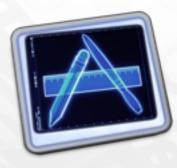




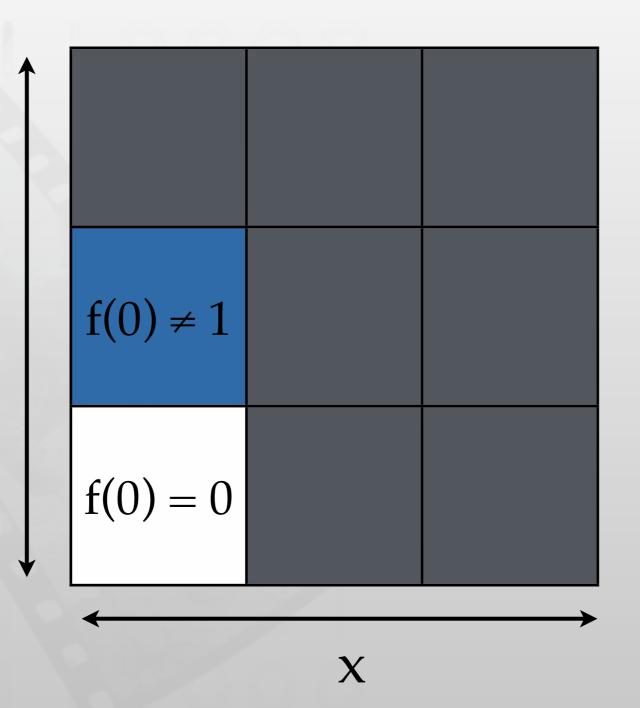
f(x) = x







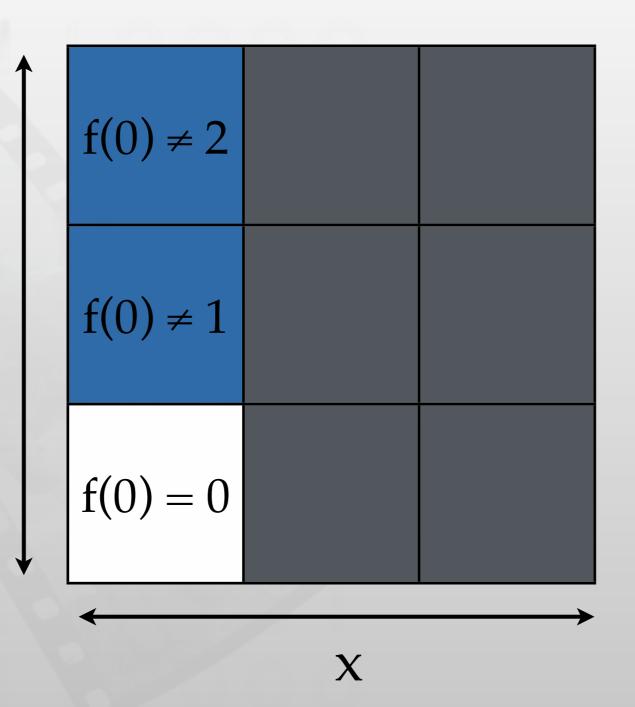
f(x) = x







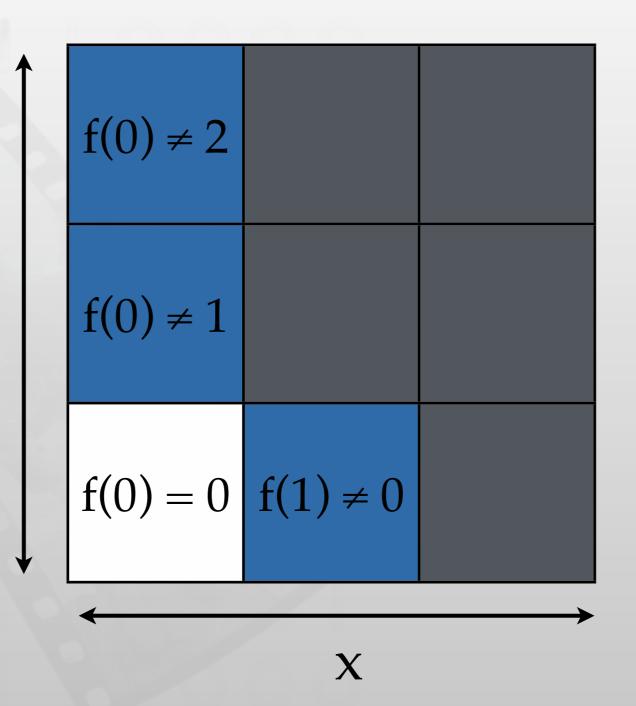
f(x) = x







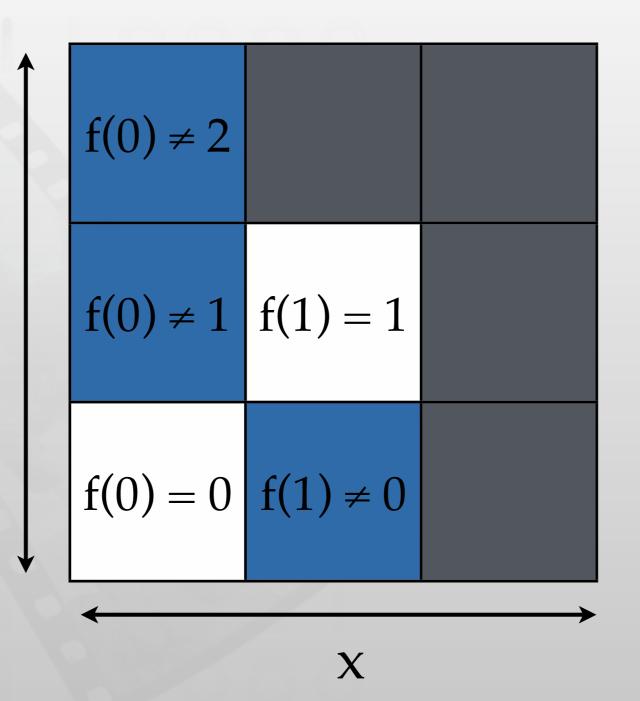
f(x) = x







f(x) = x







f(x) = x

$$f(0) \neq 2 \quad f(1) \neq 2 \quad f(2) = 2$$

$$f(0) \neq 1 \quad f(1) = 1 \quad f(2) \neq 1$$

$$f(0) = 0 \quad f(1) \neq 0 \quad f(2) \neq 0$$

$$\longleftrightarrow$$





Simple Optimization Demo





Memory Analysis

- Allocation
 - Monitors memory allocation and reference counting

• Leaks

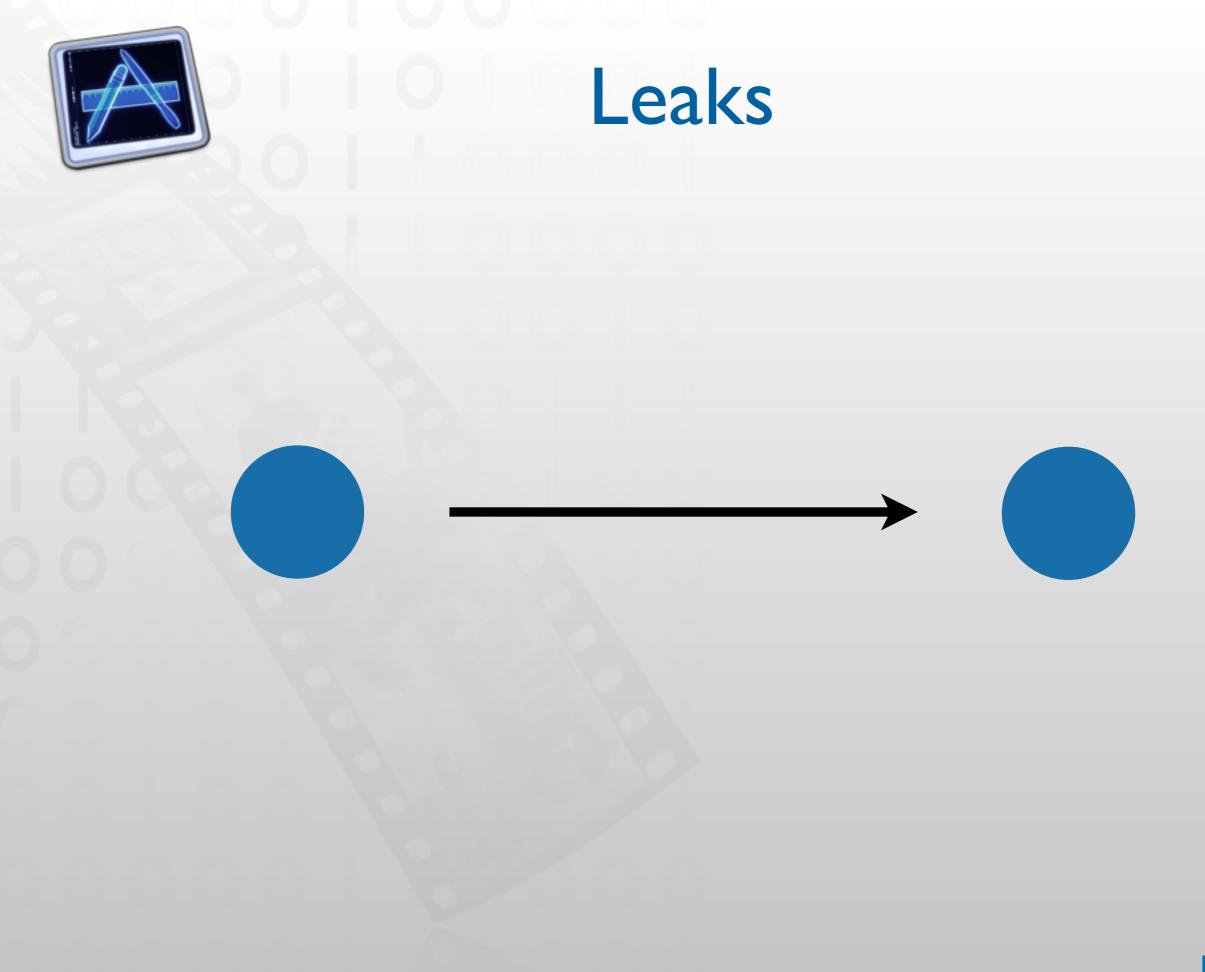
- Checks for inaccessible memory
- Finds retain cycles
- Zombies
 - Checks for freed memory being accessed

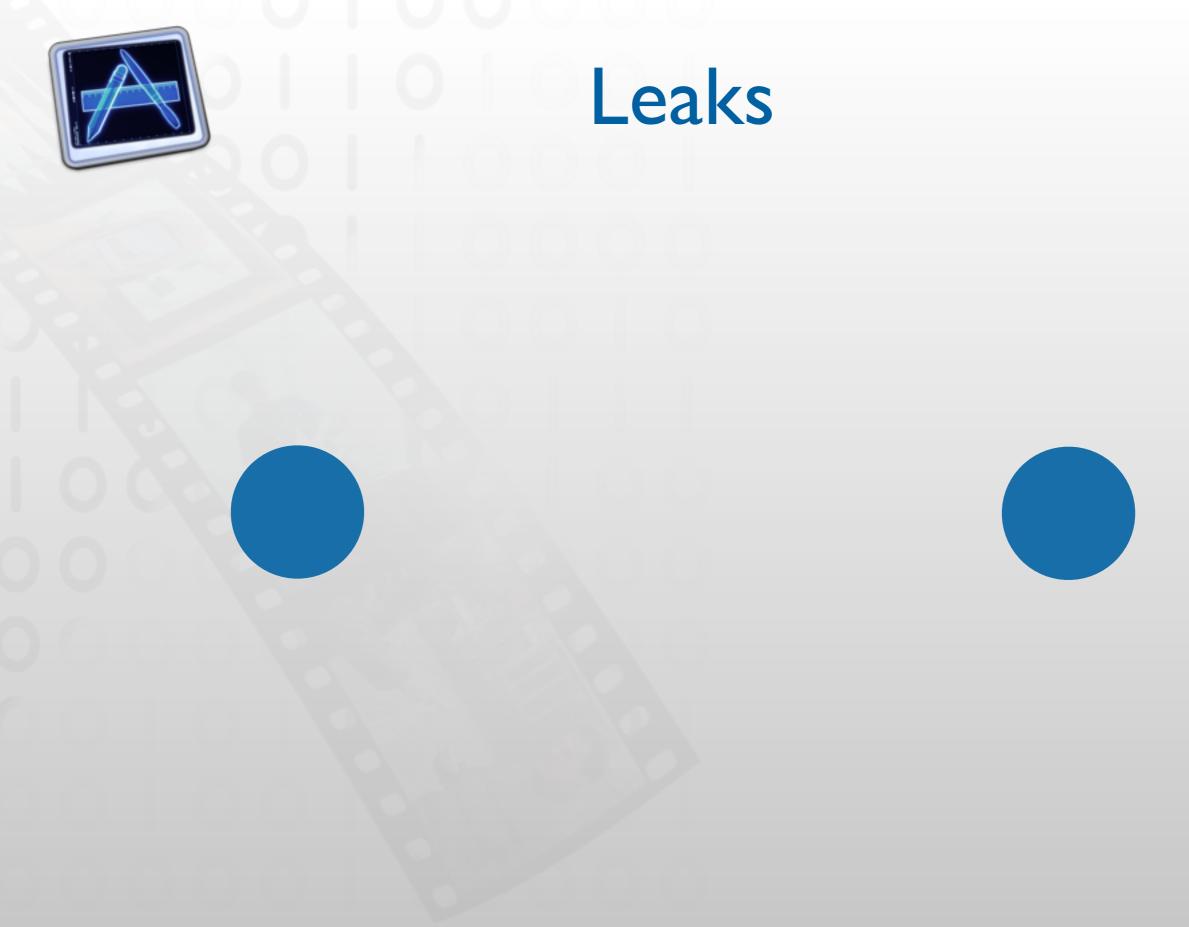










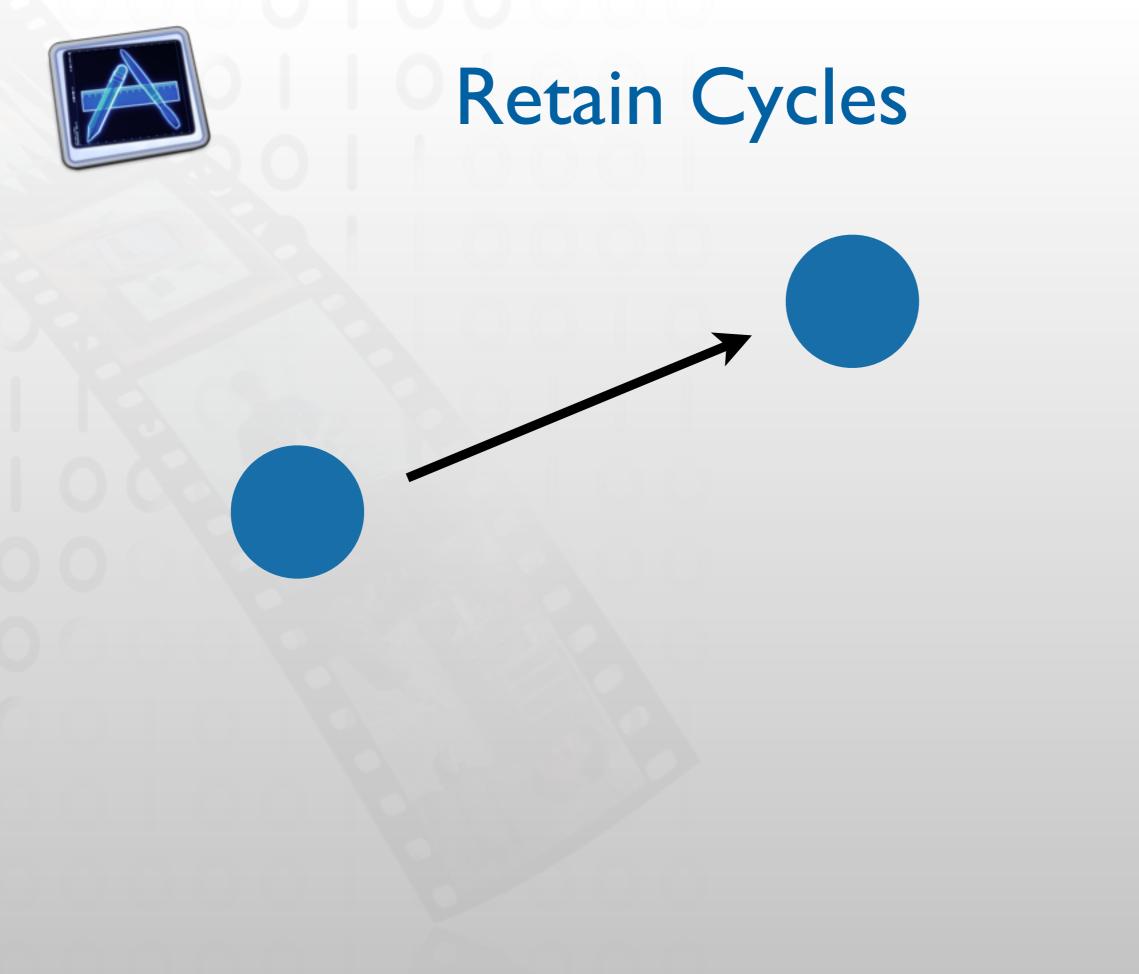




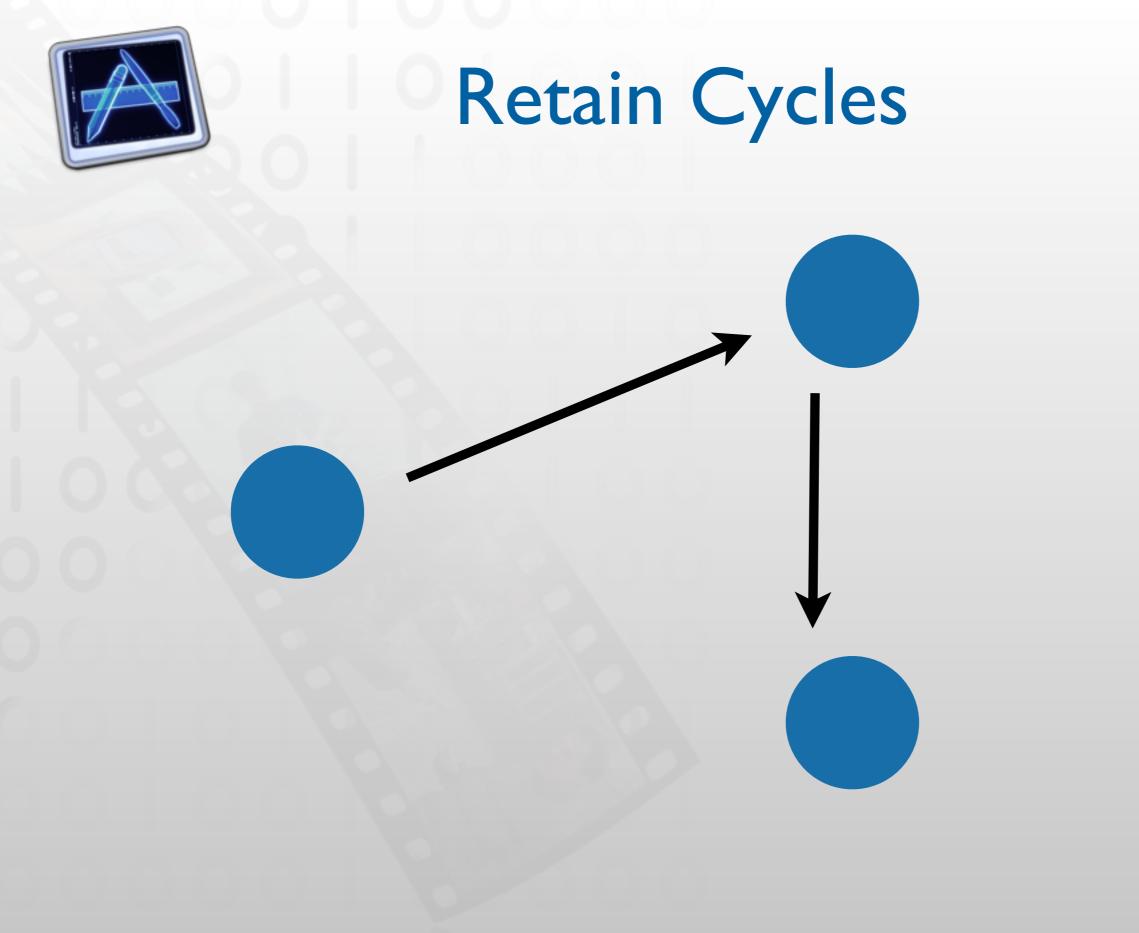


Retain Cycles

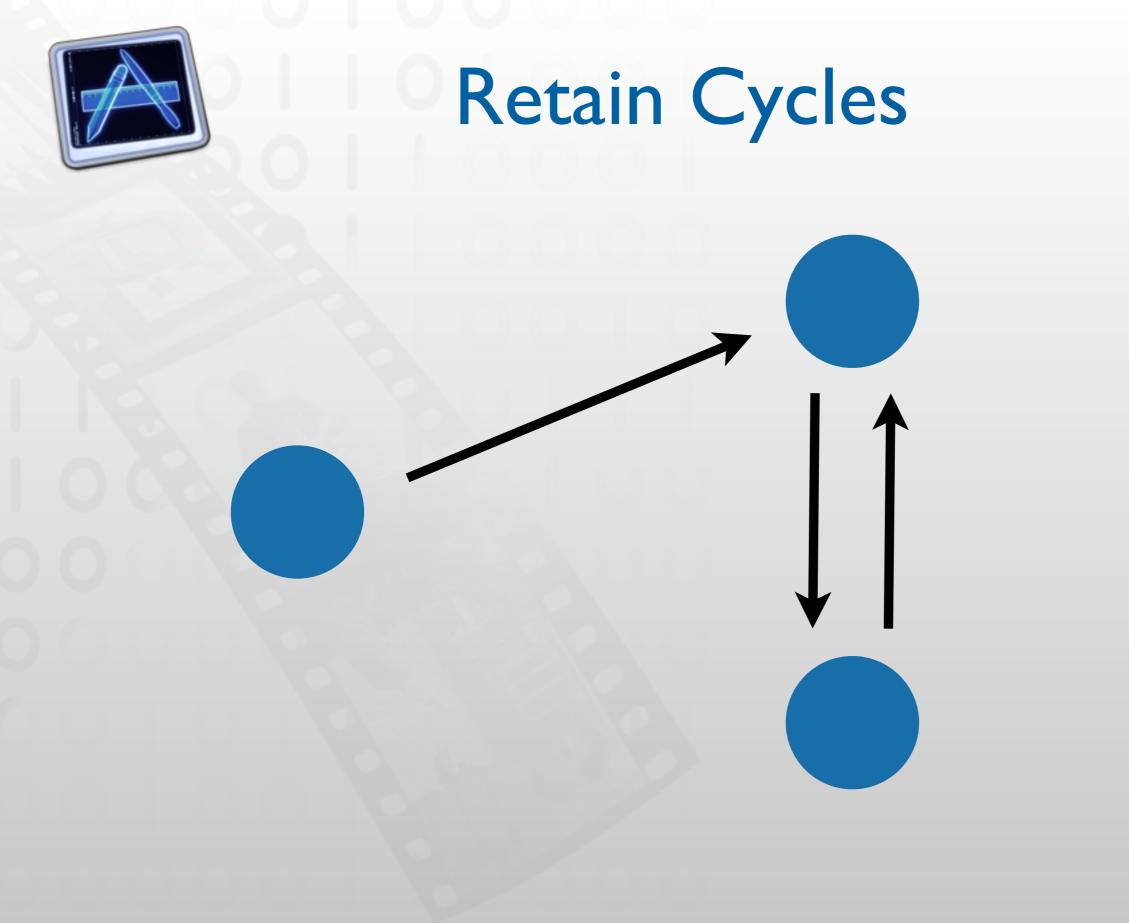








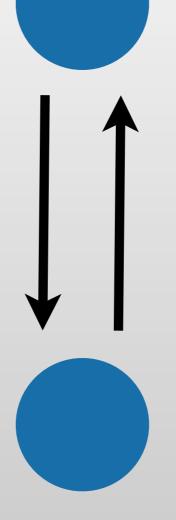
















Allocations & Leaks Demo





Zombies Freed memory being accessed

- "Good" Zombies
 - Obvious crashes
 - You release, system reuses, you try to access
 - Crash (usually EXC_BAD_ACCESS)
- Bad Zombies
 - No crash, or crash at strange location
 - You release, you allocate something, you try to access
 - Weird side-effects







Zombies Demo



21 iPhone Application Programming • Prof. Jan Borchers



- When you are done with a task: Leaks
- Whenever you get strange crashes or inexplicable values: Zombies
- You can use the simulator





Profiling

- Check in regular intervals what the CPU is doing
- Time Profiling
 - Where does the CPU spend time?
 - Distribution of work between threads / CPUs
- System Trace
 - What is the system doing? Thread scheduling Paging System calls









Time Profiling Demo

24 iPhone Application Programming • Prof. Jan Borchers



Using Time Profiling



- When your app seems too slow
 - Identify hotspots
 - Identify opportunities for parallelization
 - Identify parallelization issues (e.g. forced serial execution)
- Use on iOS Device





System Trace Demo



Using System Trace



- When results of Time Profiling are insufficient
 - Excessive context switching
 - Paging issues
 - Find opportunities to group system calls
- Use on iOS Device





f(x) = x



Order of Drawing



	f(0) ≠ 2	f(1) ≠ 2	f(2) = 2
	f(0) ≠ 1	f(1) = 1	f(2) ≠ 1
	f(0) = 0	f(1) ≠ 0	f(2) ≠ 0
<→ X			



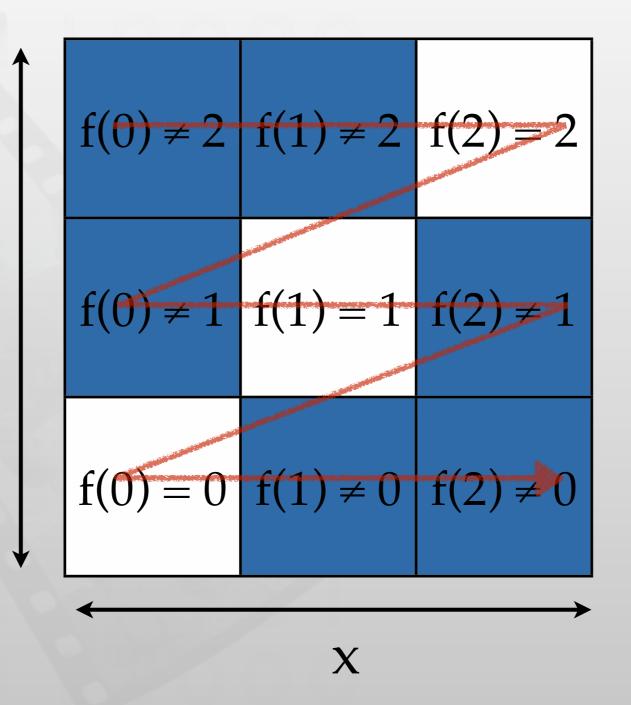


f(x) = x



Order of Drawing





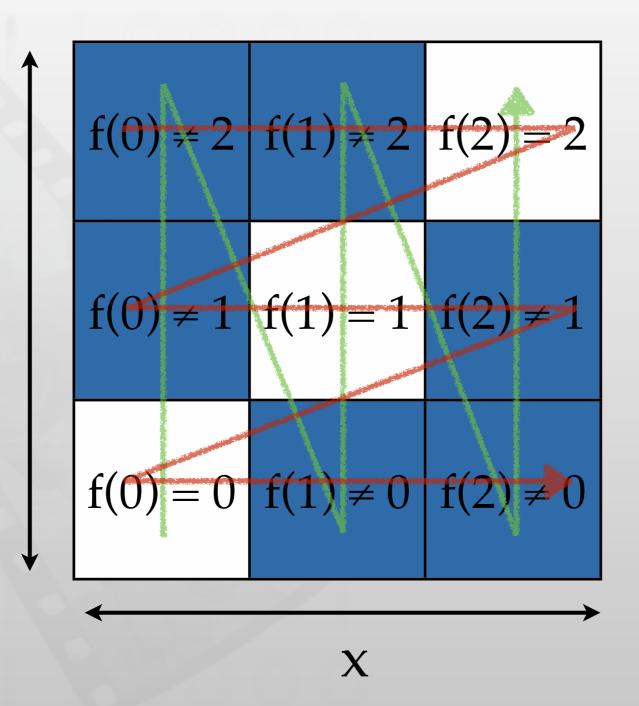




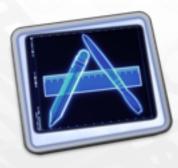
f(x) = x

Order of Drawing





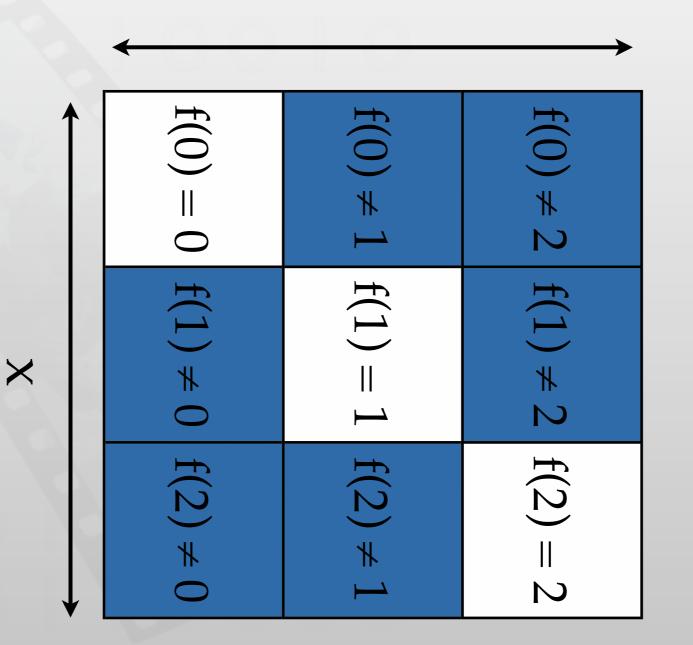




Order of Drawing



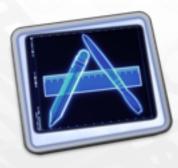




 $\boldsymbol{\triangleleft}$



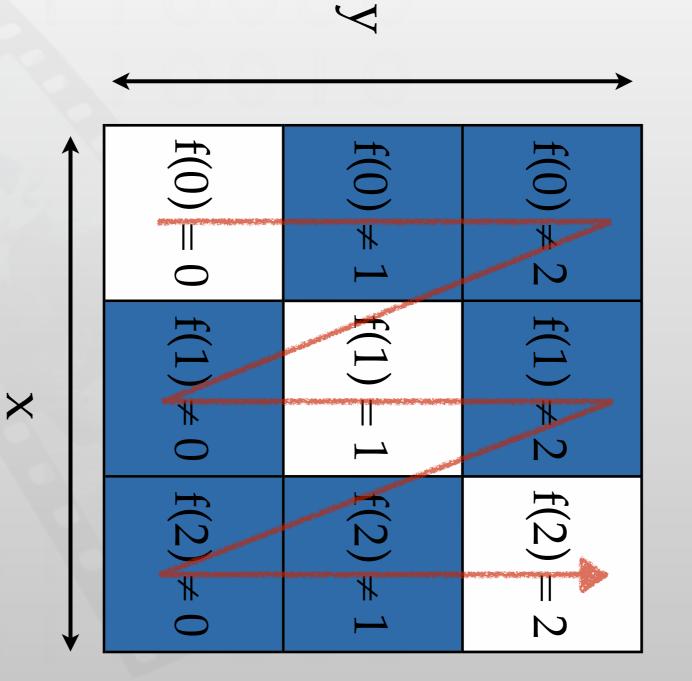
iPhone Application Programming • Prof. Jan Borchers



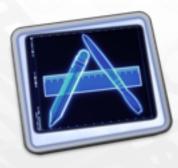
Order of Drawing







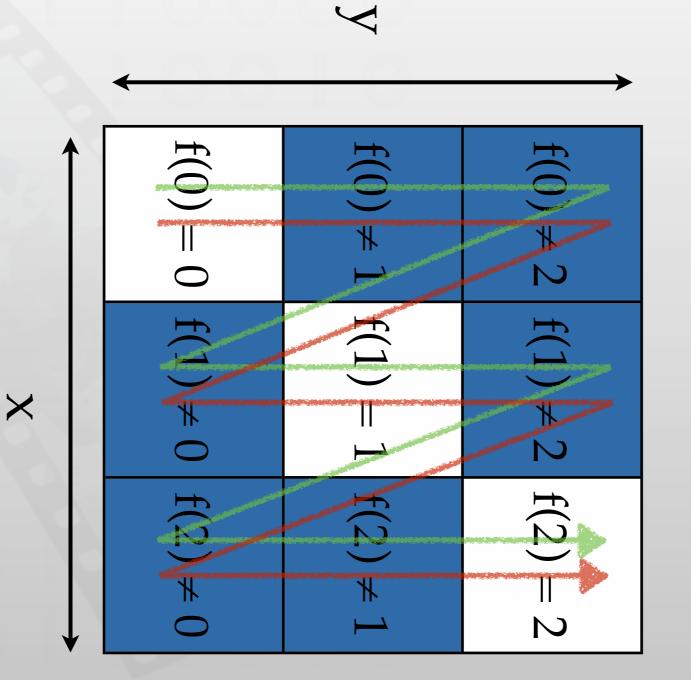




Order of Drawing











Other Instruments

- Energy Diagnostic
- Core Animation
- OpenGL ES
- System Usage
- UI Automation









- General
 - Find bugs at runtime
 - Increase algorithmic efficiency
- Profiling
 - Identify bottlenecks
 - Parallelization
 - Memory Instruments
 - Sanity checks to find leaks and zombies
 - Increase memory efficiency

