1Template Guidelines

- Before you begin, make sure you've downloaded the latest fonts. DOWNLOAD FONTS HERE
- *Save this template to your computer to create a new presentation* – always begin new projects with this template to ensure you have the most recent version
- *Only send PDF versions of presentations* if you are sending to anyone outside of CapTech

TEMPLATE SECTIONS:

Title Slides

This section provides the cover and agenda slides

- Update 'Presentation title'
- Update 'Prepared For'
- Update month/year if necessary

Guidelines

This section provides examples of each slide layout & helpful tips

- Review all slides in the section
- Add content as necessary to complete presentation.
- Remove the slides that are not going to be used

Content & Assets

This section should be used to add the presentation content

- Add additional content as necessary to complete presentation – all marketing approved slides can be found on the Marketing Portal Site

Thank You

This section allows up to 3 contacts

- Add the name(s) of contact(s)

Appendix

Add additional supporting slides here

- Remove the section if appendix is not applicable

CapTech

Multi-tasking & Adaptive UI

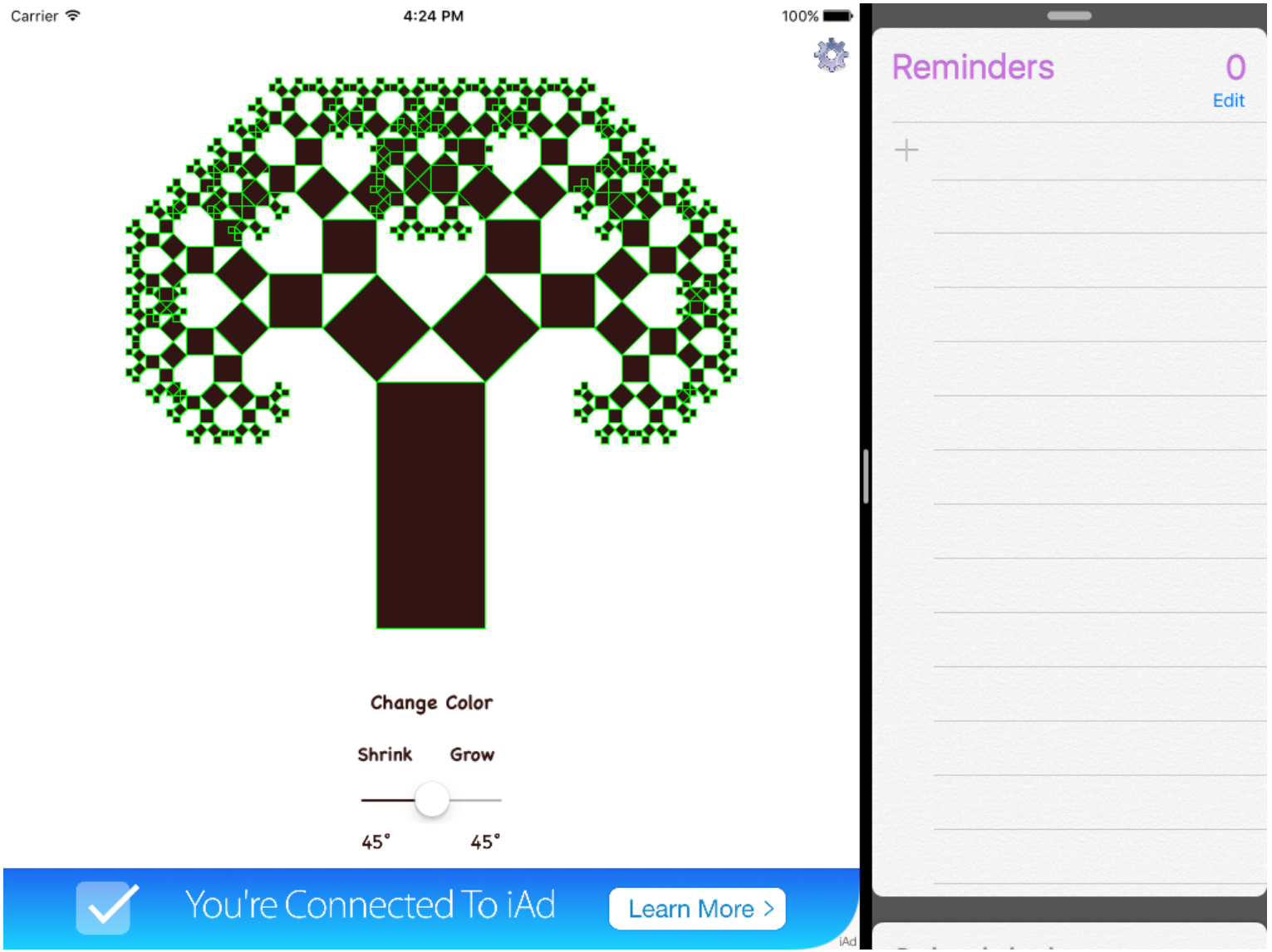
Orlando iOS Meetup – January 2016

Others Talk,
We Listen.

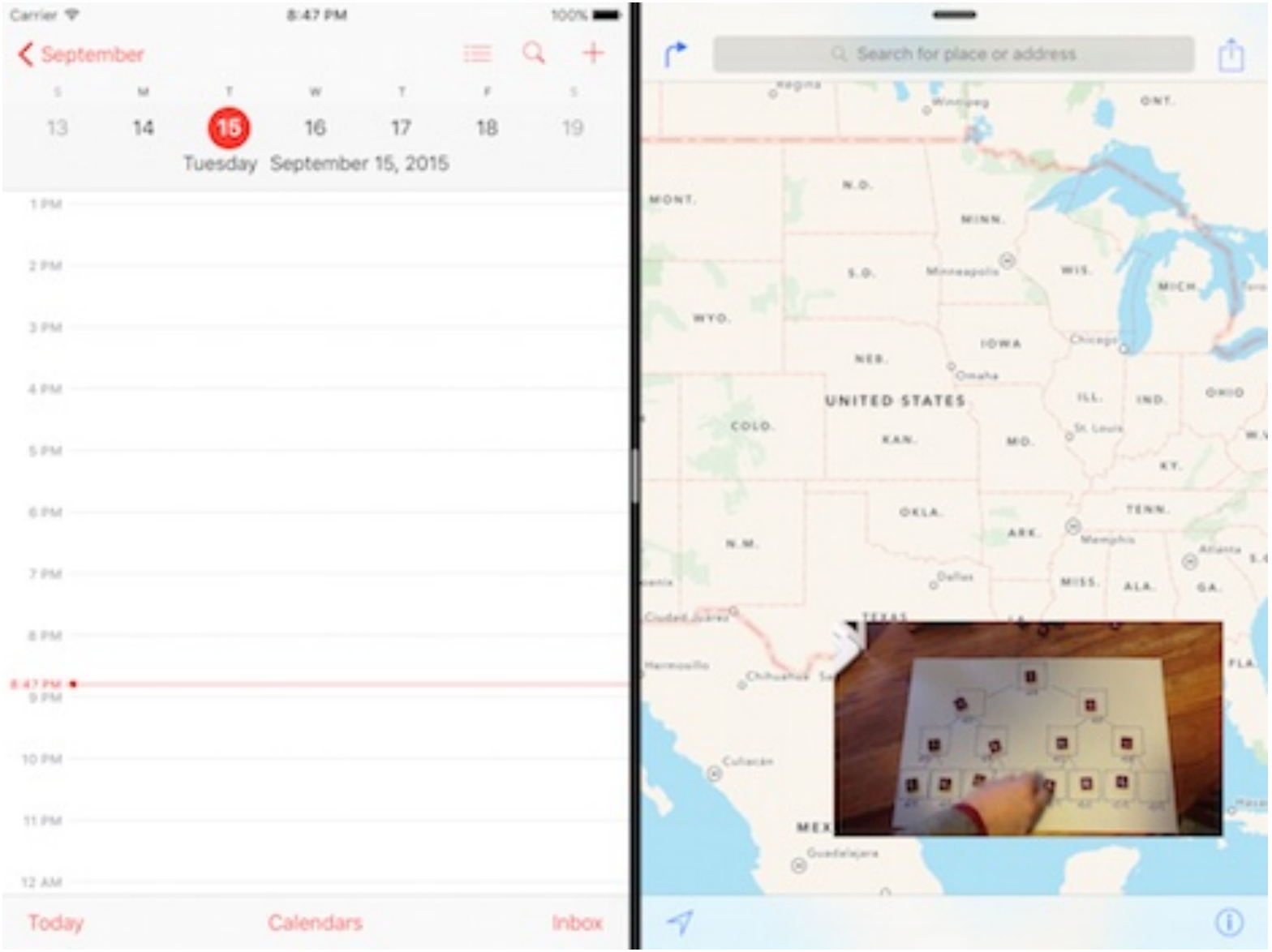
3iOS Devices with Multi-tasking capability

	Slide Over	PIP	Split View
iPad Air	✔	✔	
iPad mini2	✔	✔	
iPad mini3	✔	✔	
iPad Air2	✔	✔	✔
iPad mini 4	✔	✔	✔
iPad Pro	✔	✔	✔

4Demo – Multi-Tasking



5 Split View with PIP



6 Requires FullScreen?

It is recommended that all apps are Universal and allow for Multi-tasking.

Exceptions for some apps such as full screen games or ones where the primary purpose uses the camera can use the UIRequiresFullscreen key in the Info.plist.

What if an apps primary purpose isn't camera centric but the app does use full-screen preview to capture camera image?

7UIInterfaceOrientation

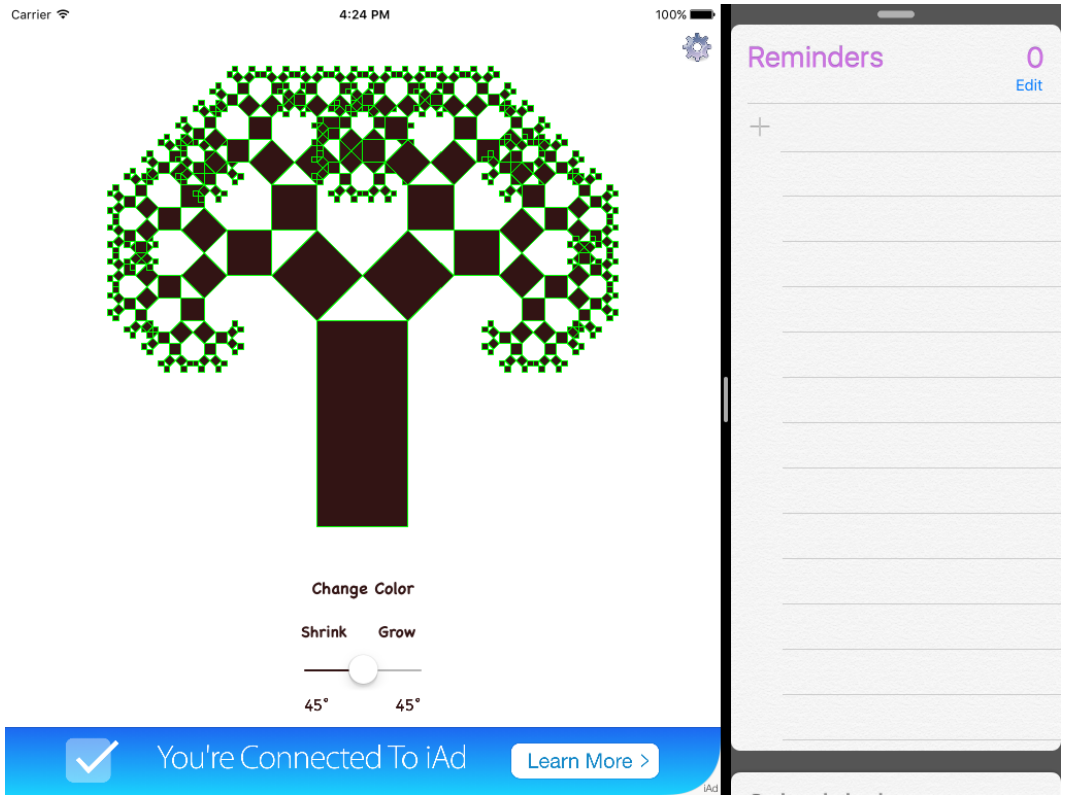
The meanings of orientation Landscape & Portrait can be misleading.

Below the device is in Landscape orientation, but the Reminders App is being shown in a Portrait style.

Unless supporting

iOS < 8.0

UIInterfaceOrientation should be seen as deprecated.



8 Alternatives to UIInterface Orientation

Use TraitCollection:

Contains both a horizontal and vertical size class as .Regular or .Compact

Use UIWindow size:

Compare the width to the height to see if your app is currently showing in a more horizontal or vertical orientation.

Responding to Size Changes & Rotation

Rotation to/from interface orientation methods were all deprecated starting in iOS8.

Just like with rotations, the user has total control over size changes. App can't cause or prevent size change; just need to respond to it.

Use same code for handling size change & rotation

10 Invocation Order of Rotation – Size methods

1. willTransitionToTraitCollection
2. viewWillTransitionToSize
3. traitCollectionDidChange
4. animateAlongsideTransition
5. Completion block/closure of animateAlongsideTransition

Don't do a lot of work in the first 2 methods. If work takes too long app will be terminated.

The first 3 methods aren't guaranteed to be called.

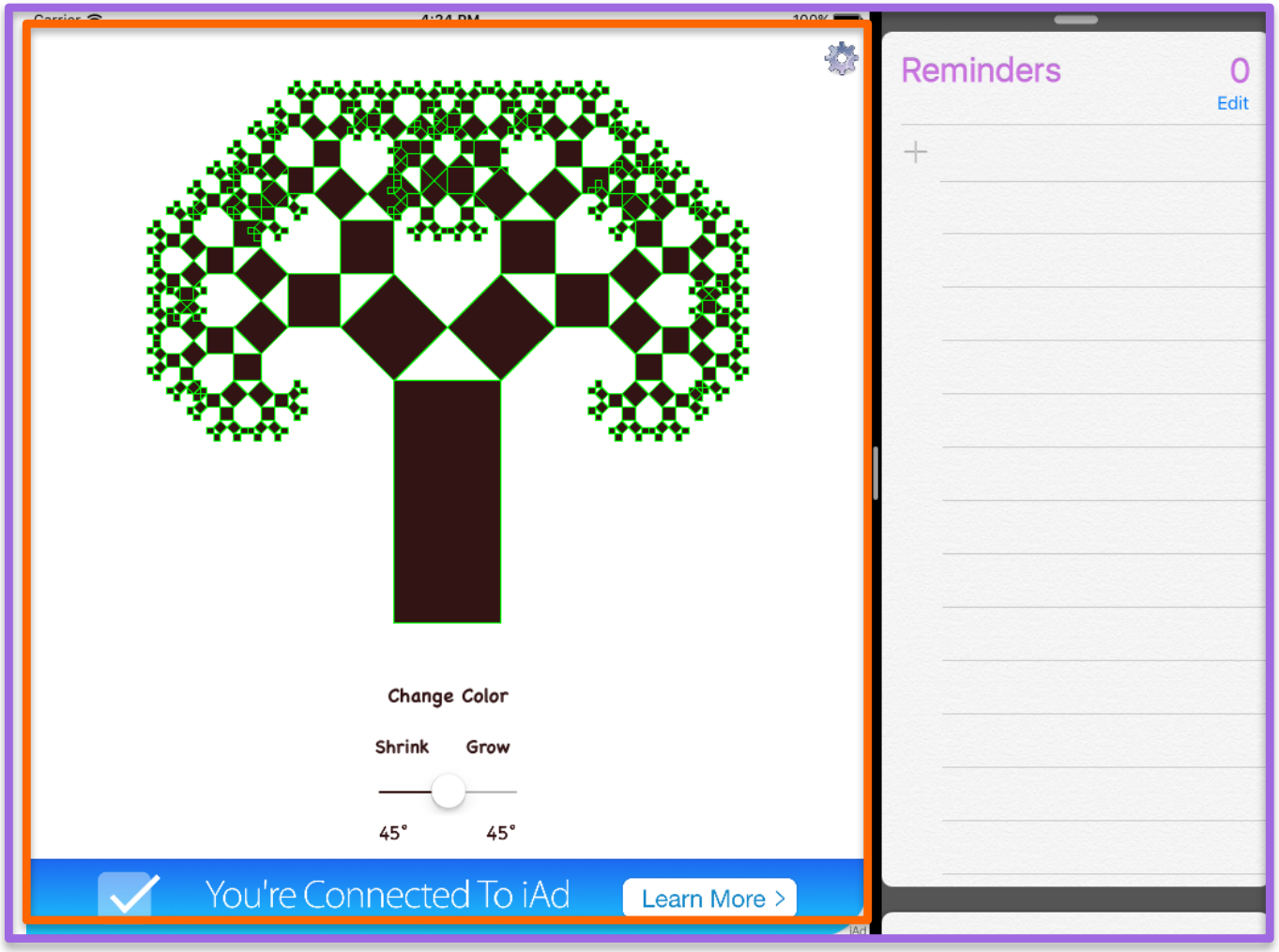
11Example Code:

```
override func willTransitionToTraitCollection(newCollection: UITraitCollection, withTransitionCoordinator
coordinator: UIViewControllerTransitionCoordinator) {
    super.willTransitionToTraitCollection(newCollection, withTransitionCoordinator: coordinator)
    coordinator.animateAlongsideTransition({ (UIViewControllerTransitionCoordinatorContext) -> Void in
        //don't call layoutIfNeeded, it's time consuming and will be done for you anyway
        //call setNeedsLayout
    }) { (UIViewControllerTransitionCoordinatorContext) -> Void in
        //do any time consuming work here
    }
}
```

12 UIWindow - UIScreen

UIWindow

UIScreen



UIWindow - UIScreen

UIWindow(frame: UIScreen.mainScreen.bounds())

Replace with just UIWindow() making it the full size the app is currently using and the window will update automagically during size changes.

The upper-left corner of the UIWindow is always (0,0) no matter where it is on screen.

App Resigns Active / Becomes Active

AppDelegate applicationWillResignActive will be called when going into the background. Remember size and state.

When AppDelegate applicationDidBecomeActive is called again, use the remembered state to reset the app back to the last state the user was in.

Shared Resources

Only one app can use the camera. Only one system keyboard will display. There is only one speaker and one microphone.

These resources need to be shared from app to app and you have no control of what the other app is doing.

Demo of keyboard.

CPU, GPU, Memory, IO, Disk Space

All resources are shared. No app has priority over another app.

The CPU and GPU need to be able to display at 60 fps, if two or three apps are sharing the processing and can't run at 60 fps, the lag will be noticeable to the user with the apps stuttering.

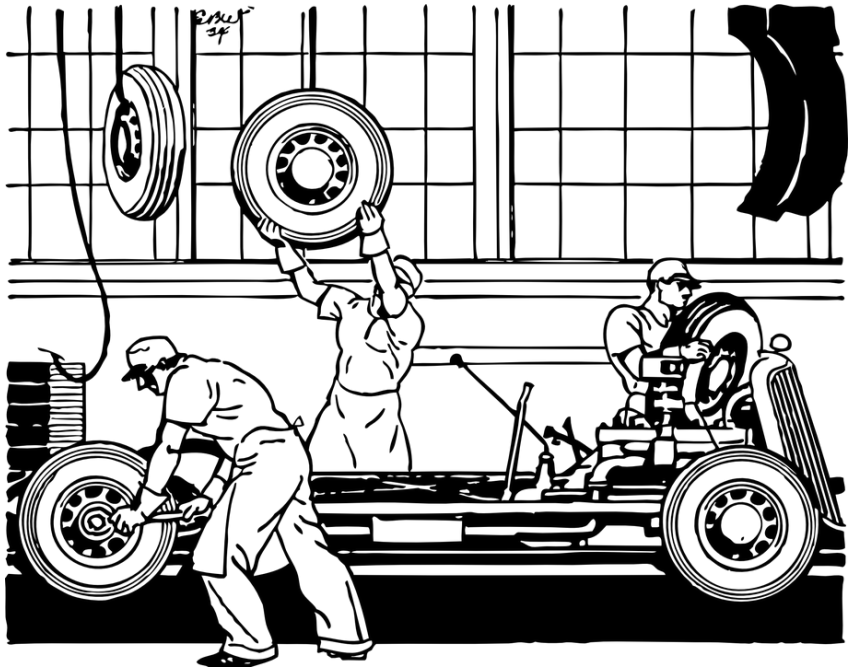
If a device runs out of memory it will terminate an app. Even if yours isn't the one that caused the memory issue, it might still be terminated.

Use Instruments to fine tune memory issues. Make sure algorithms are efficient ~~(Bubble Sort~~)

Do as little work as possible on the main thread.

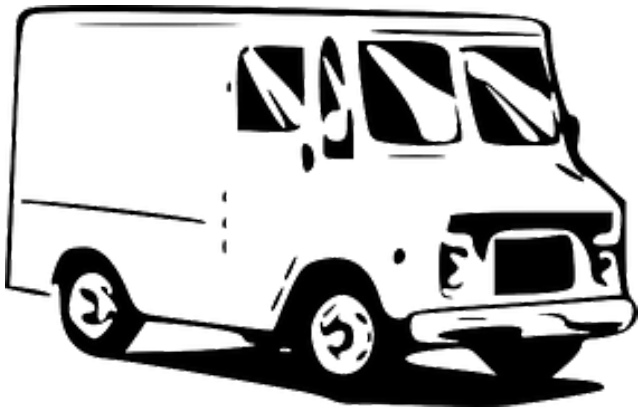
Keeping Memory Footprint Low - Large Inventory Production

Full warehouse of parts

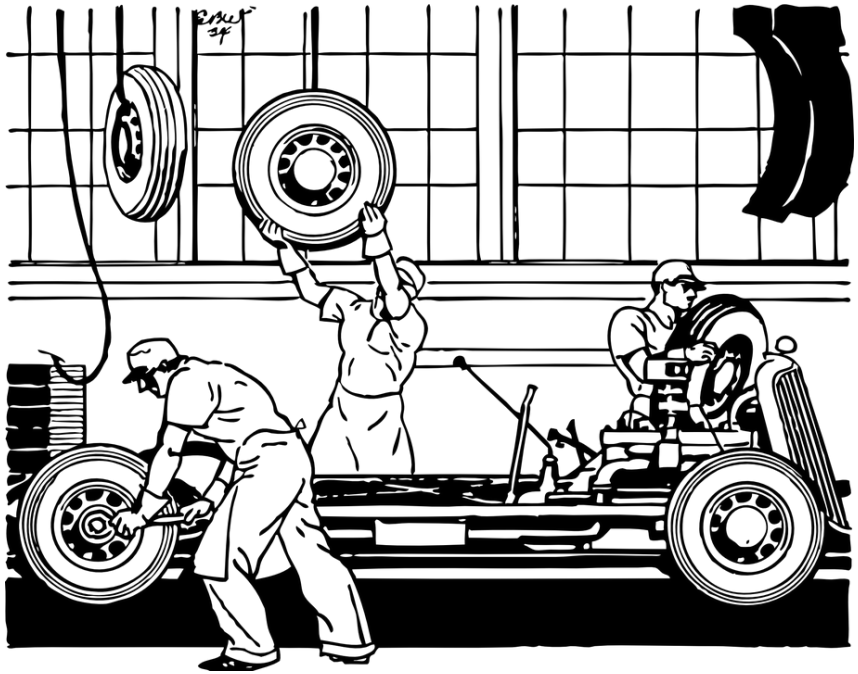


Auto plant

Keeping Memory Footprint Low - Just-in-time Manufacturing



Deliver parts as needed



Auto plant

19 Keeping Memory Footprint Low - Month Transition

◄ March 2016 ►

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

◄ ►

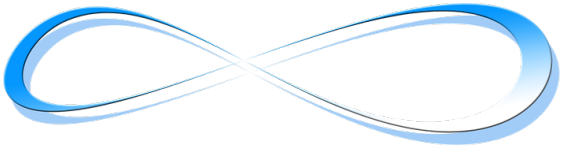
S	M	T	W	T	F	S
2	3	4	5			
9	10	11	12	3	4	5
16	17	18	19	10	11	12
23	24	25	26	17	18	19
30	31			24	25	26

◄ April 2016 ►

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

20Keeping Memory Footprint Low - Not in Memory

21Keeping Memory Footprint Low - Over-full Warehouse



Infinite months

Stack Overflow

Memory Crash

Keeping Memory Footprint Low - Smaller Warehouse

12 months x 7 possible combinations

Whoops, Leap Year, February has another 7

91 different calendars in Memory

Fill up Dictionary of 91 Calendar objects

Keeping Memory Footprint Low - Proof

If it exists for 1,

And if for every value of n, an n + 1 exists,

And if for every value of n, an n - 1 exists.

Then it must exist for all values of n.

24Keeping Memory Footprint Low - How many calendars?

N - 1

N + 1

25 Handling Running out of Memory

Front Image

Back Image

26 Handling Running out of Memory

Front Image

Back Image

Temp

27Handling Running out of Memory

Front Image

Back Image

Temp

Purge

28 Handling Memory Warnings

```
override func didReceiveMemoryWarning() {
}
```

Handle it yourself.

- Clear cached data
- Releasing images,
- Releasing view controllers

OR.....

Adaptive Memory

NSCache – Works just like a Dictionary. Automagically evicts items when under memory pressure.

```
let cache  = NSCache()
cache.setObject(frontImage, forKey: ”tempImage”)
```

30 Testing Memory Pressure

Use Maps with flyover.

Use Instruments.

In the Simulator -> Hardware -> Simulate Memory Warning

Additional Resources

Multi-tasking

https://www.captechconsulting.com/blogs/ios-9-tutorial-series-multi-tasking-with-adaptive-ui

Size-Classes

http://www.captechconsulting.com/blogs/ios-9-tutorial-series---size-classes-preparing-your-apps-with-adaptive-ui

Auto-Layout

https://www.captechconsulting.com/blogs/ios-7-tutorial-series-auto-layout-in-xcode-5