

# A Strategy For Great Work

Stories and the lessons learned from them

Session 237

Ken Kocienda

Experimental Designer

Trying to do great work

# 2012: Basics + Habits

## Tactical

Today  
Strategic

Tactics are the actual means used to gain an objective, while strategy is the overall campaign plan, which may involve complex operational patterns, activity, and decision-making that lead to tactical execution.

Wikipedia

[http://en.wikipedia.org/wiki/Tactic\\_\(method\)](http://en.wikipedia.org/wiki/Tactic_(method))

# Strategy begets tactics

Ideas which inspire and inform my everyday work

# HOW TO MAKE A GREAT PROJECT IN 21 DAYS

It isn't that simple

My stories







# My stories

Extract out the lessons from the stories

Lesson + lesson + lesson...

*My strategy*

8 stories

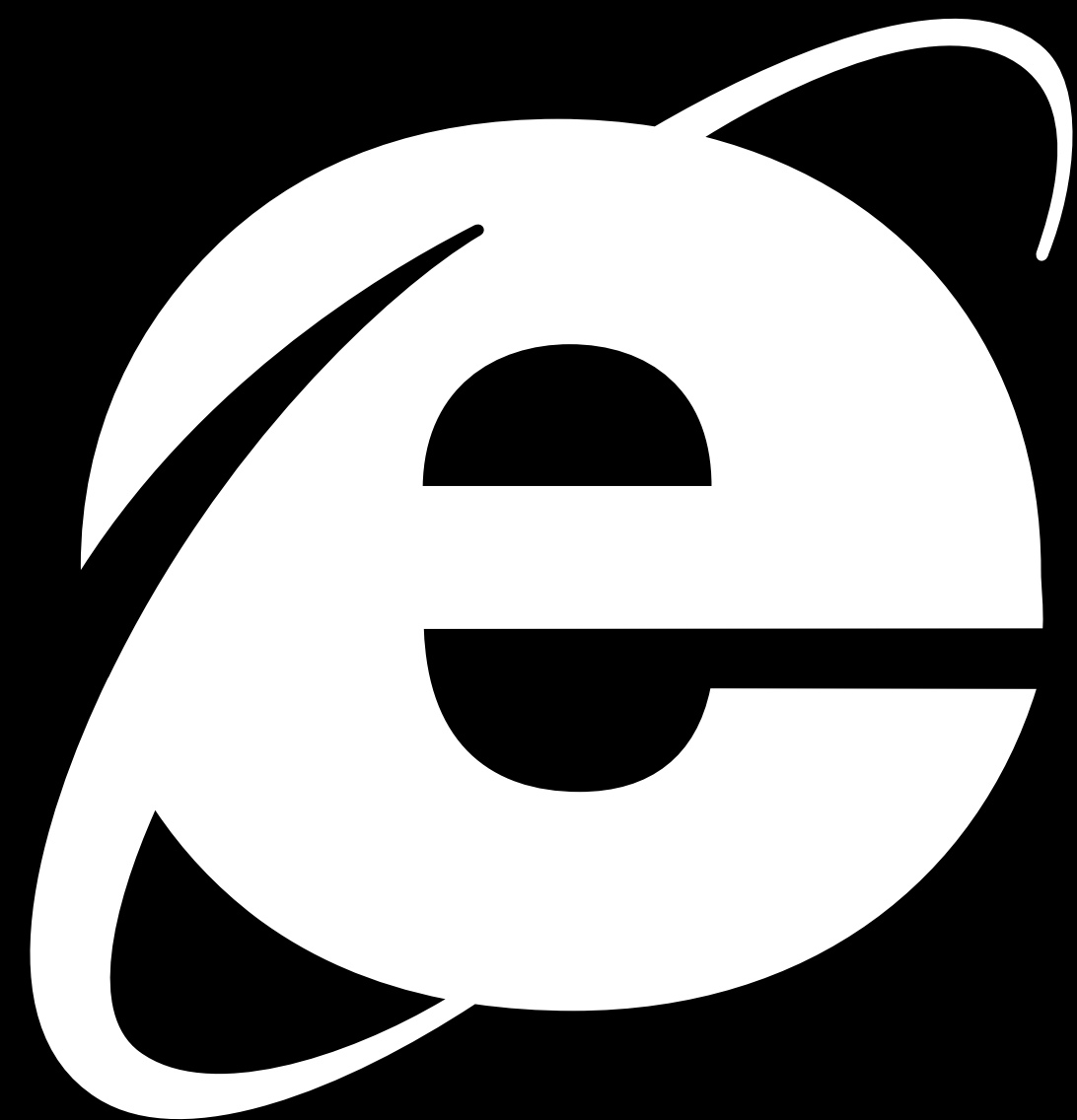
11 lessons

# #1. Like A Crystal Ball

Know a good idea when you see it

I joined Apple in June 2001

Mac OS X 10.0 released March 2001



Apple web browser  
...and an Obj-C web framework, too!



# Options

MSIE, iCab, Opera, Mozilla, Gnome, KHTML, scratch?

# Investigation

Open source

# Six weeks

Test results... but no demo

# New engineer

“What have you been doing?”

Two days later...

Demo!

# KHTML running on a Mac

Konqueror in an X window

Amazing!

# The demo was like a crystal ball

We saw that KHTML was our future



Know a good idea when you see it

I looked like a bumbling idiot... I didn't care

# Like A Crystal Ball

Know a good idea when you see it



**Make Demos  
& Prototypes!**

# Like A Crystal Ball

Know a good idea when you see it

## #2. The Black Obelisk

Don't try to solve every problem at once

# KHTML!

What do we do now?

Open source

Part of the KDE desktop

We only wanted KHTML  
...and KJS too

# Untangle KHTML from KDE

About 100 source files



# Brought them over to the Mac

Started compiling one file at a time

# Stubbed in missing functions

One at a time

# Painstakingly slow work

Everything compiled after six weeks

# Nothing linked!

Another six weeks

# Crashes

Lots of crashes

# A Mac program that used KHTML

The lights came on

<http://www.yahoo.com>



# Don't try to solve every problem at once

We had no idea how KHTML worked... we didn't care



# The Black Obelisk

Don't try to solve every problem at once

# #3: The Hardest Problem

Find smart friends and listen to them

# Fall 2003

Add editing to WebKit

# Use WebKit to edit HTML mail

Make it look and work like a text editor

# Arrow navigation

The hardest problem I ever worked on

|Hi *there!*

**Hi *there!***|

**Hi *there!***

`<div><b>Hello <i>there!|</i></b></div>`



**Hi *there!***

`<div><b>Hello <i>there!</i></b></div>`

**Hi *there!***

`<div><b>Hello <i>there!</i></b></div>`

**Hi *there!***

```
<div><b>Hello <i>there!</i></b></div>
```

# Complex model/view relationship

Model changes, but view doesn't change

# DOM positions

```
<div><b>Hello <i>there!</i></b></div>
```

# Equivalent DOM positions

`<div><b>Hello <i>there!</i></b></div>`

# Right Arrow Algorithm

```
DOMPosition pos(GetInsertionPointPosition());  
pos = EquivalentDownstreamPosition(pos);  
pos = GetNextDOMPosition(pos);  
SetInsertionPointPosition(pos);
```

# Right Arrow Algorithm

```
DOMPosition pos(GetInsertionPointPosition());  
pos = EquivalentDownstreamPosition(pos);  
pos = GetNextDOMPosition(pos);  
SetInsertionPointPosition(pos);
```

These functions must work perfectly



# Right Arrow Algorithm

```
DOMPosition pos(GetInsertionPointPosition());  
pos = EquivalentDownstreamPosition(pos);  
pos = GetNextDOMPosition(pos);  
SetInsertionPointPosition(pos);
```

Already at the end of the document?

# Right Arrow Algorithm

```
DOMPosition pos(GetInsertionPointPosition());  
pos = EquivalentDownstreamPosition(pos);  
pos = GetNextDOMPosition(pos);  
SetInsertionPointPosition(pos);
```

What if you want to move by word?

I was stuck

I asked for help

Very smart people on my team

Explained the problem

# DOMPositions: too low-level

More powerful abstraction

# VisiblePosition

A position where the insertion point can blink

# visible\_units

Words, lines, and documents

# Abstractions organized work

Each problem had a place for its solution to live



Find smart friends and listen to them

Don't stay blocked

# The Hardest Problem

Find smart friends and listen to them

# #4: QWERTY

Work should explain itself

# Summer 2005

## iPhone development

# Joined to work on WebKit

Soon transitioned to keyboards

**App**

**Keyboard**

**App**

**Keyboard?**

# Blank slate

We didn't have plastic keys



# We could experiment

We came up with many ideas

# Challenges

Small targets are hard to tap

# Challenges

Tap targets get covered by your finger

Bigger keys

Solve both problems

# An example

The keyboard keyboard





AFG	BGK	CHK	DIL	EJM
HIJ	LMN	NOQ	NPR	OPS
QRS	QRS	TUW	TVW	UVX
TUV	WXY	XZ	YZ	YZ

T |

AFG  
HIJ  
QRS  
TUV

BGK  
LMN  
QRS  
WXY

CHK  
NOQ  
TUV  
XZ

DIL  
NPR  
TVW  
YZ

EJM  
OPS  
UVX  
YZ



TH

AFG  
HIJ  
QRS  
TUV

BGK  
LMN  
QRS  
WXY

CHK  
NOQ  
TUV  
XZ

DIL  
NPR  
TVW  
YZ

EJM  
OPS  
UVX  
YZ

THE |

AFG  
HIJ  
QRS  
TUV

BGK  
LMN  
QRS  
WXY

CHK  
NOQ  
TUV  
XZ

DIL  
NPR  
TVW  
YZ

EJM  
OPS  
UVX  
YZ



# Keys are big

Easy to press the correct key



# Learning curve

First time experience is a mystery

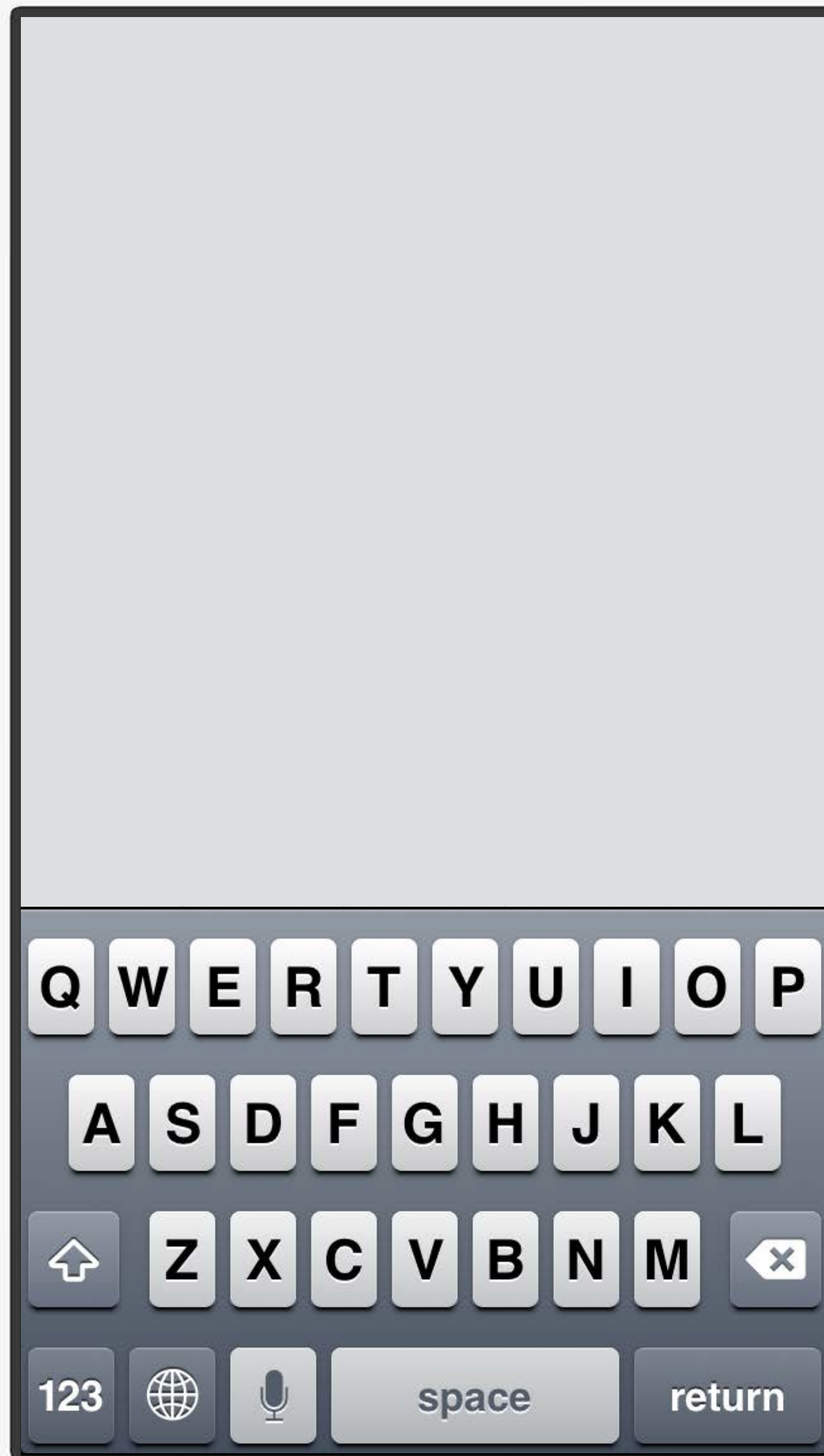
# Tutorial

Teach people how to use the keyboard

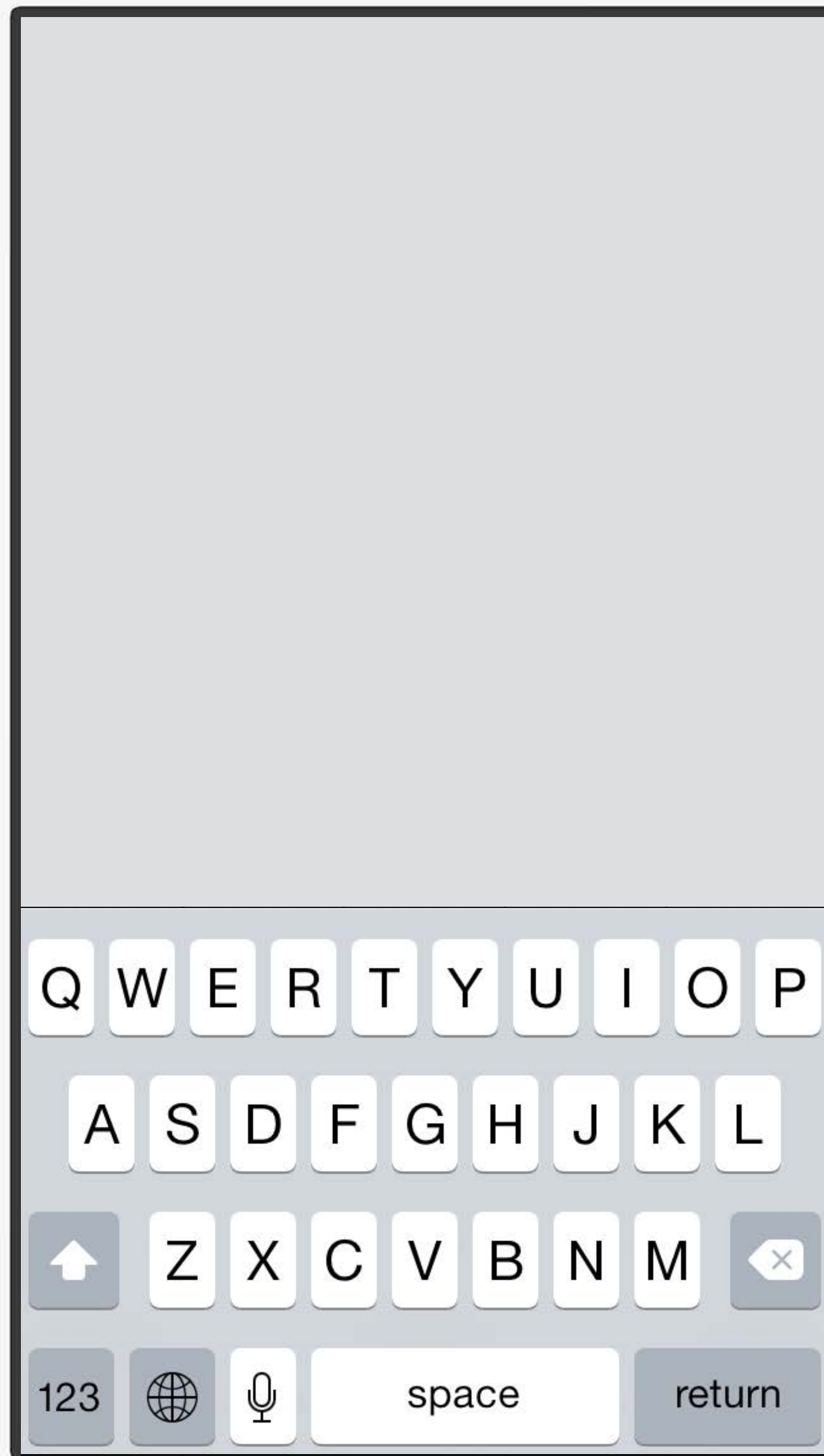
# After each experiment...

We always came back to QWERTY

AFG	BGK	CHK	DIL	EJM
HIJ	LMN	NOQ	NPR	OPS
QRS	QRS	TUW	TVW	UVX
TUV	WXY	XZ	YZ	YZ







# Work should explain itself

People already know how to use QWERTY

# QWERTY

Work should explain itself

# #5: Every Word On Every Keystroke

Choose the simplest thing which might work

# iPhone keyboard

Do the work to make QWERTY possible

# You want to type quickly

The keyboard software needs to help

# Autocorrection

What you meant—not what you did





*Conditional random fields (CRFs)*

*Hidden Markov model (HMM)*

*Maximum-entropy Markov model (MEMM)*

*Bayesian statistics*

*Stochastic programming*

# Large statistical models

Not possible

???

I last studied math in the 11th grade

# Simple

Because math is hard

# Type the letter 'i'

i

Q W E R T Y U I O P

A S D F G H J K L

Z X C V B N M

# Type the letter 'i'

i

Q W E R T Y U I O P

A S D F G H J K L

Z X C V B N M

# Type the letter 't'

i

Q W E R **T** Y U I O P

A S D F G H J K L

Z X C V B N M

# Type the letter 't'

it

Q W E R T Y U I O P

A S D F G H J K L

Z X C V B N M



# Type the letter 't'

it

Q W E R T Y U I O P

A S D F G H J K L

Z X C V B N M

it?

ig?

if?

ir?

of?

og?

or?

ot?

oy?

uf?

ug?

uy?

...

# Speculation

Did you mean this? Did you mean that? Did you...

# Many dictionary lookups

Need for speed

# Multithreaded?

I am scared of multithreaded code

# Simple

Because I can understand simple

# Searching a Dictionary

...  
Baryshnikov  
baryta  
basal  
basalt  
basaltic  
bascule  
bascules  
base  
baseball  
baseballs  
baseband  
baseboard  
baseboards  
based  
Basel  
...

- Memory map word list
- Iterate list to find line ends
- Store them in a pointer array
- Binary search
- Fast!

# Custom C++ string class

Wrap any string in the memory mapped area

# Check dictionary membership?

Best measured in microseconds



# Made speculation cheap

Made autocorrection possible

# Type the letter 't'

it

Q W E R T Y U I O P

A S D F G H J K L

Z X C V B N M

it?

ig?

if?

ir?

of?

og?

or?

ot?

oy?

uf?

ug?

uy?

...

# Fast

Only stored the string for the current word

Re-ran entire algorithm...

...on every keystroke

Searched the whole dictionary...

...on every keystroke

Choose the simplest thing which might work

Because simple is often speedy

# Every Word On Every Keystroke

Choose the simplest thing which might work

#6: We only need one of these, right?

Only show your best work



2009

Developing the iPad

# iPad needed a keyboard

In landscape, close to the size of a full-size keyboard

# Overlap With Standard Keyboard



# Overlap With Standard Keyboard



# Overlap With Standard Keyboard



# Type comfortably

Autocorrection could help

Maybe people would like more keys?

We had the room for them













Best of both worlds!

Cool!

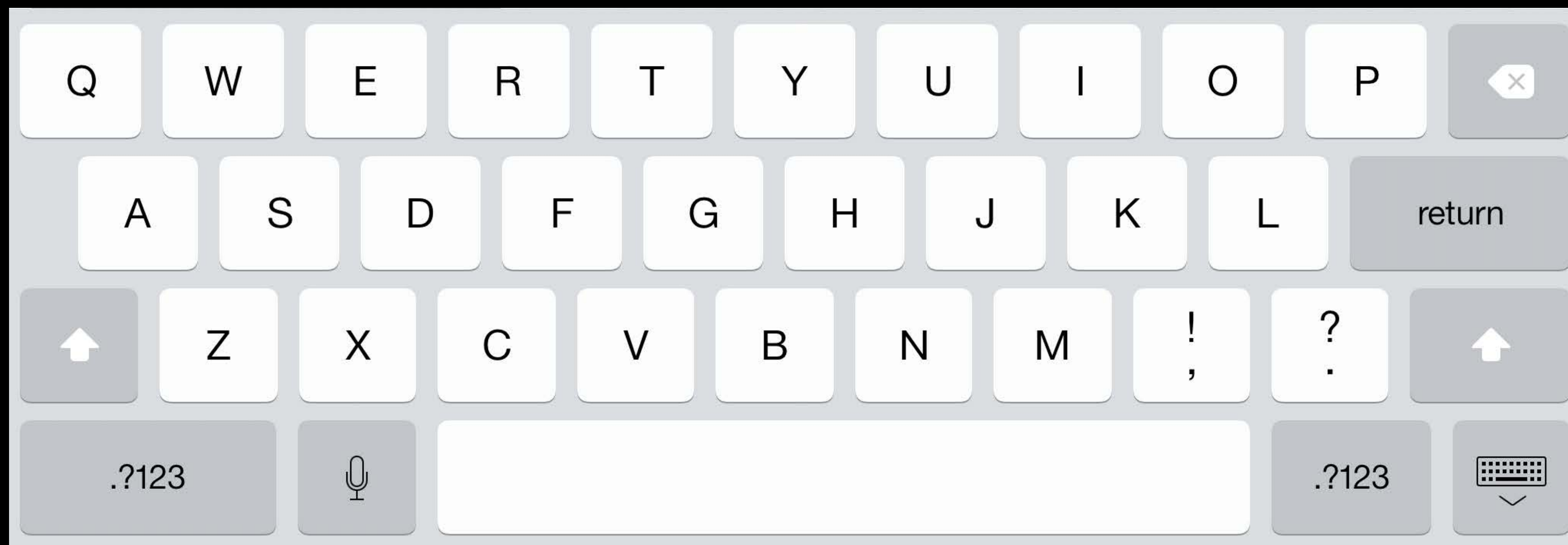
# Demo for Steve

*Gulp!*

“We only need one of these, right?”







Only show your best work

Don't show people your second-best work

We only need one of these, right?

Only show your best work

# #7: Let's Try It 2% Darker

Iterating quickly leads to better work

# 2012

Developing iOS 7

# A new look for iOS

How do you make all those choices?

# Whac-A-Mole

Every decision affects every other decision

# Required a lot of tuning

Iterate, iterate, iterate...



One design iteration

# One design iteration

Demo to designer

# One design iteration

Demo to designer

Take notes

# One design iteration

Demo to designer

Take notes

Go away to make changes

# One design iteration

Demo to designer

Take notes

Go away to make changes

Schedule next demo

# One design iteration

Demo to designer

Take notes

Go away to make changes

Schedule next demo

Next demo to designer

# Hours or days

Can we go faster?

# Settings

One of my smartest friends

^

*very*




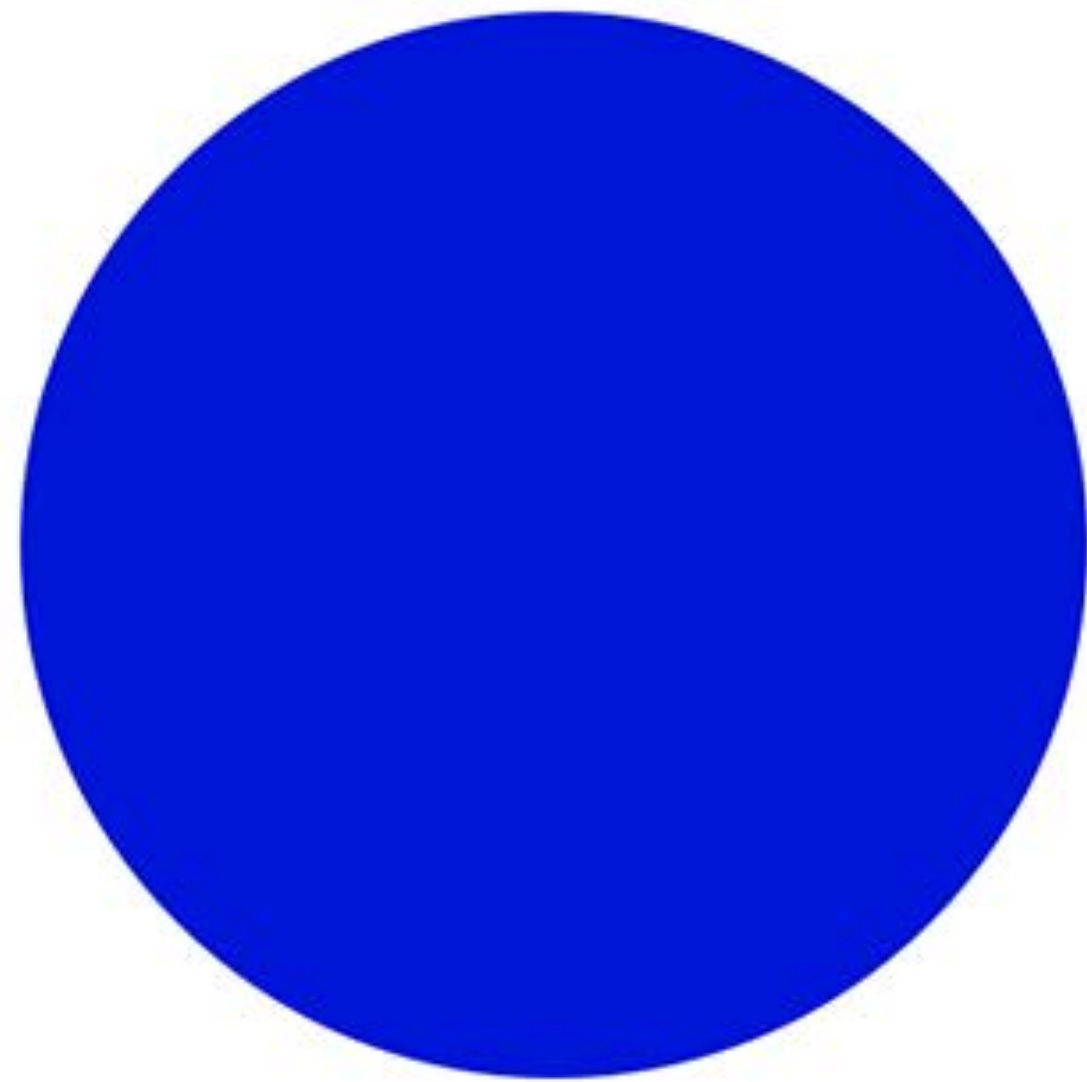
# Tunable values

KVO + save/restore + editing UI



9:41 AM

100% 



Edit



9:41 AM

100%

Done

Reset

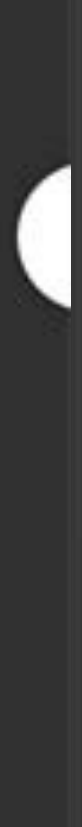
1

1

65%

100%

85%



Count

Points

Hue

Saturation

Brightness

Untitled

Save

Show Saved

Edit



9:41 AM

100%

Done

Reset

3

3

100%

100%

8%



Count

Points

Hue

Saturation

Bright

Untitled


Save

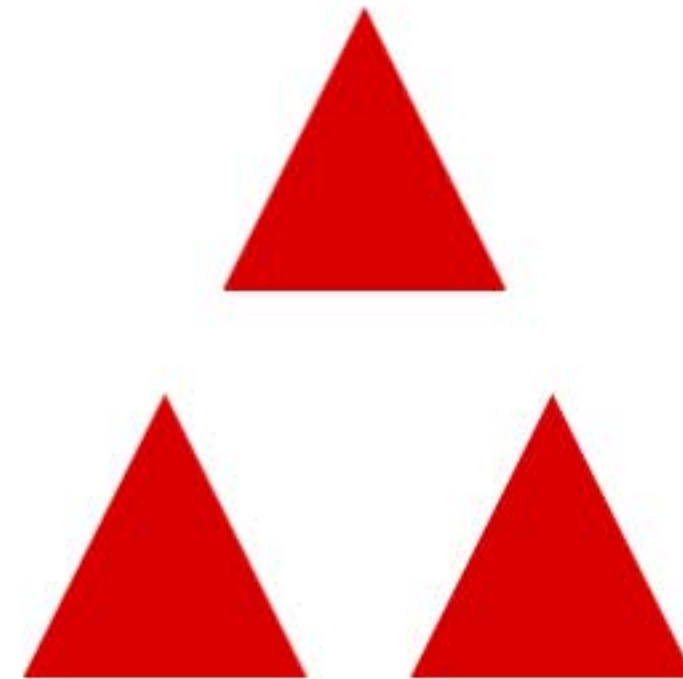
Show Saved

Edit



9:41 AM

100% 



[Edit](#)



9:41 AM

100%

Done

## GEOMETRY

Count 3

Points 3

## COLOR

Hue  1.00

Saturation  1.00

Brightness  0.85

Alpha  1.00



9:41 AM

100%

Done

## GEOMETRY

Count 1

Points 1

## COLOR

Hue  0.65


Saturation  1.00

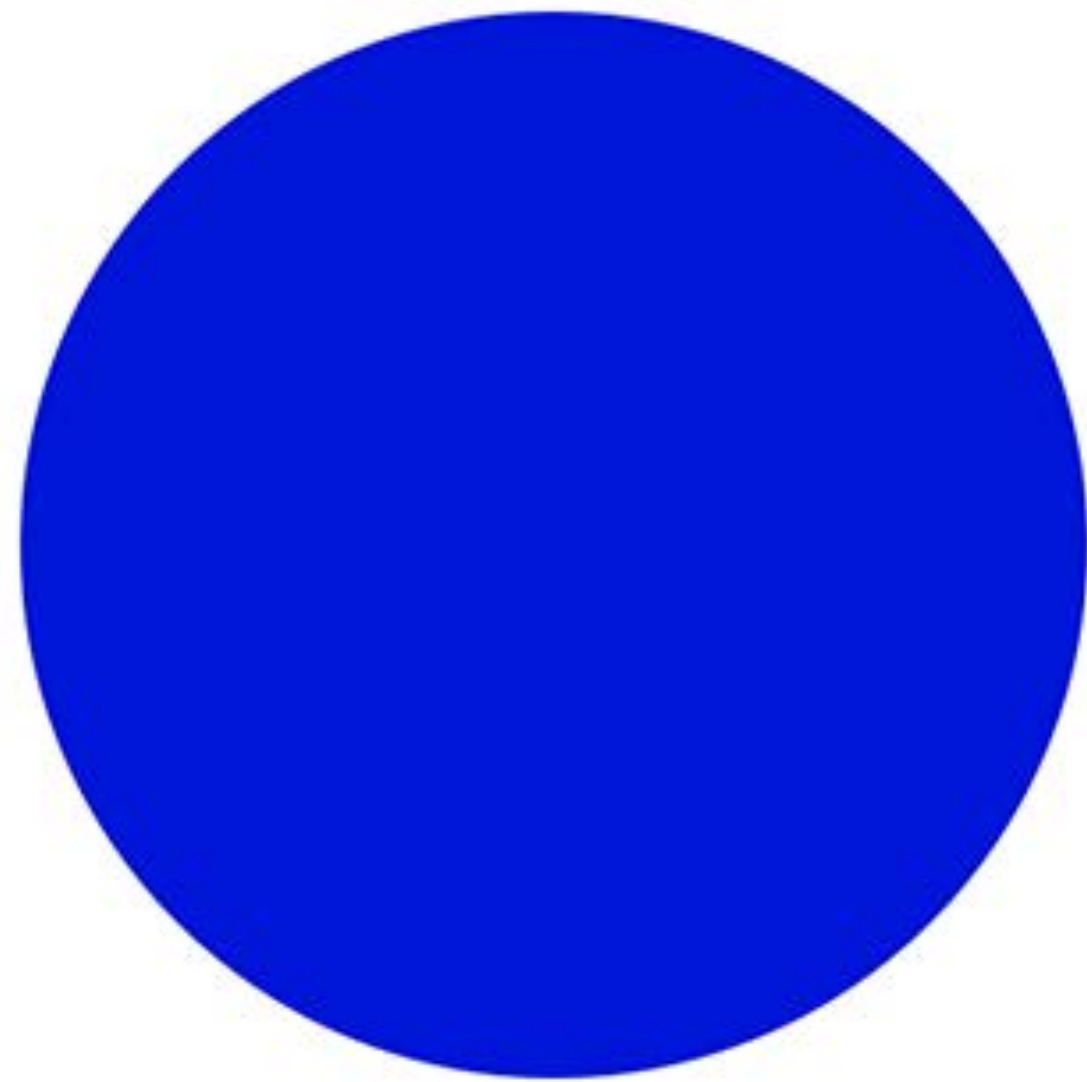
Brightness  0.85

Alpha  1.00



9:41 AM

100% 

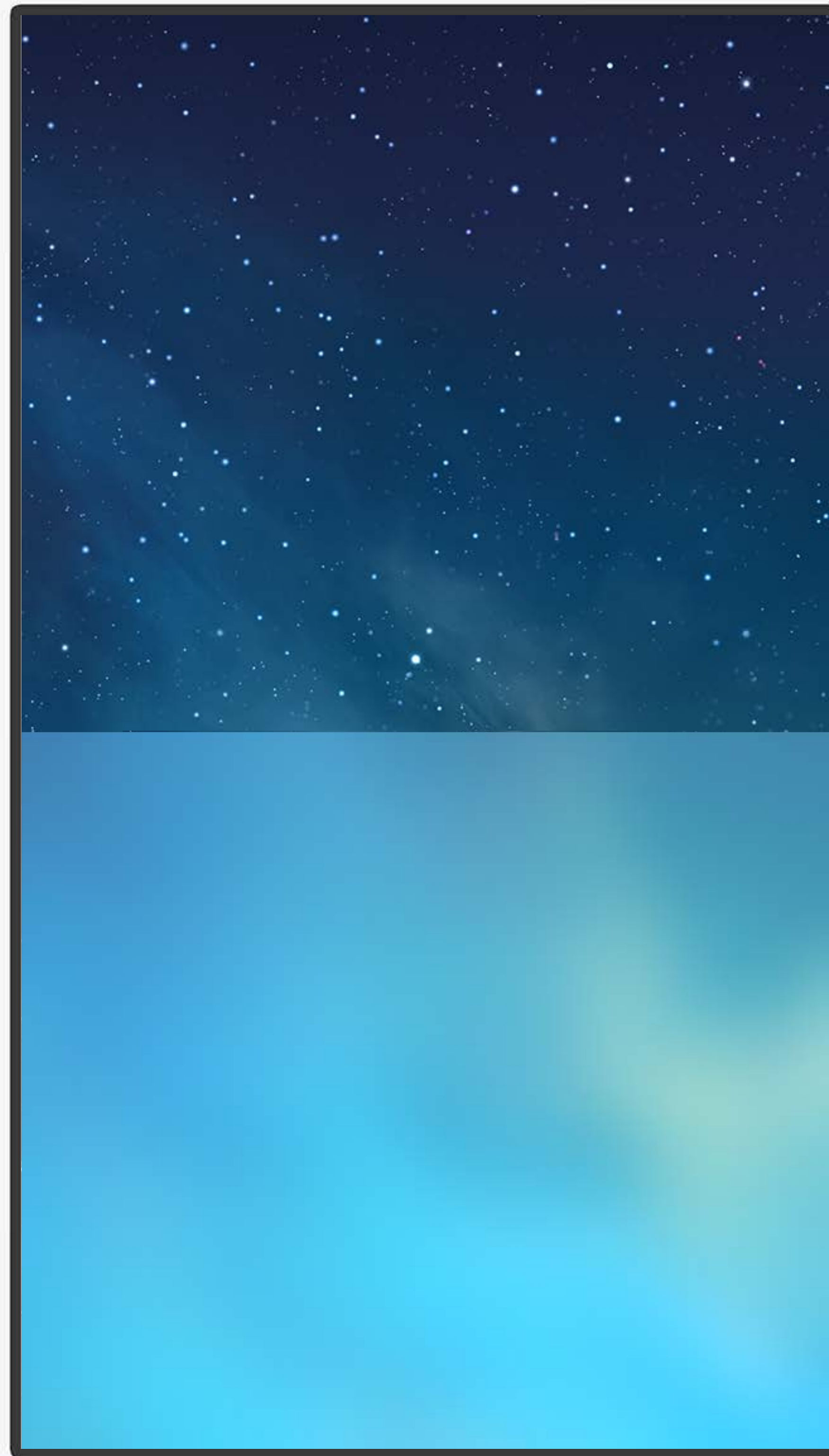


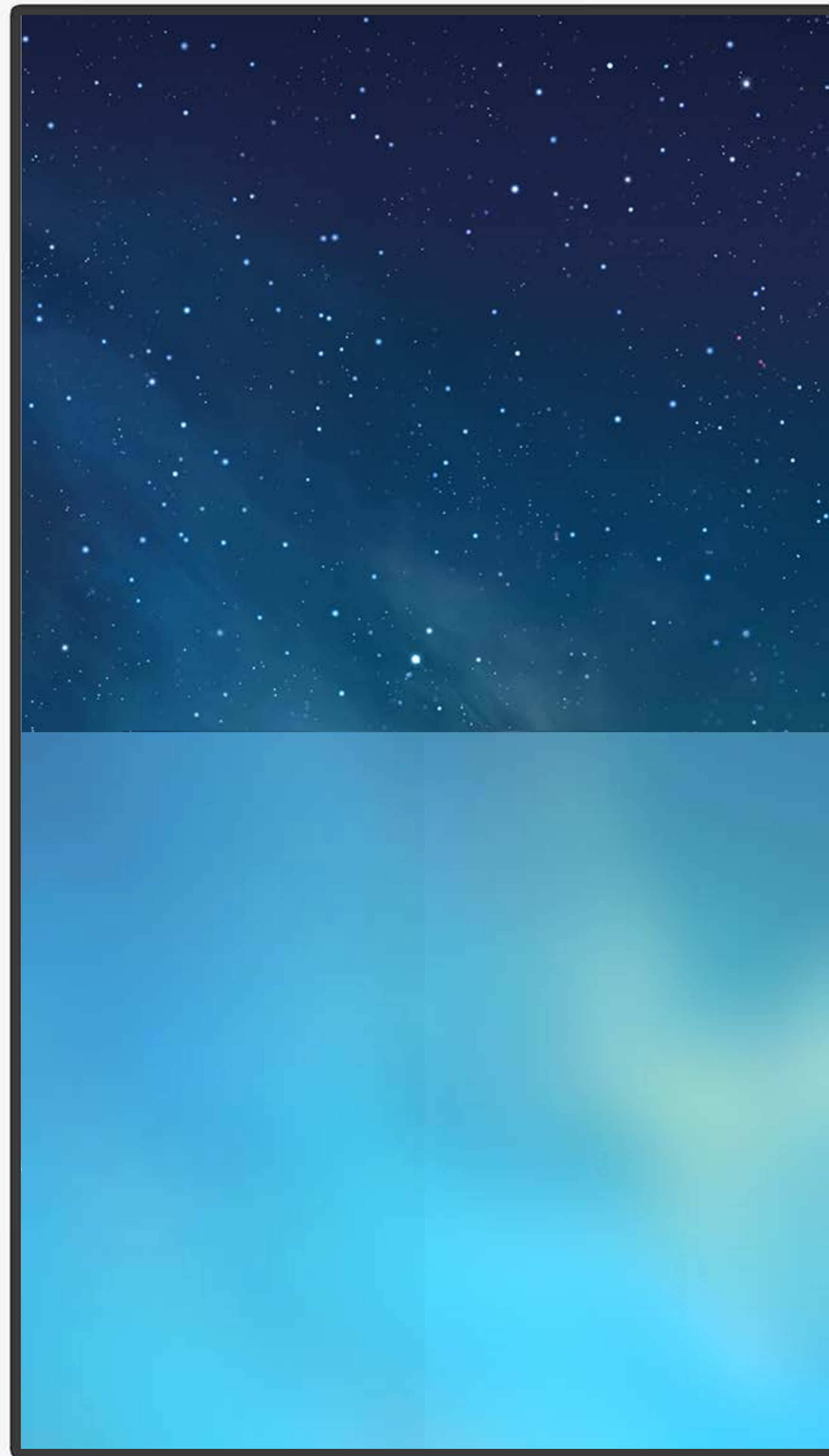
Edit



# Reduce iteration time

Sit with a designer and try many options





# Go back to my desk

Copy new defaults and commit

Cheap and easy to tune

Over and over and over again

# A handful of blur styles

Used consistently throughout iOS 7

# Iterating quickly leads to better work

Direct correlation between quality and ease of iterating

# Let's Try It 2% Darker

Iterating quickly leads to better work



# #8: Rewrite

Three lessons from making this session

# Not the original session

I wrote another set of slides

# Three full rehearsals

I was done way ahead of time!

# One last review

Go over the material in the slides...

# Shot down!

Issues which couldn't be resolved

Rewrite

Three important lessons

There were no raised voices



# There were no raised voices

Be kind to people, but be honest about work

I didn't take it personally

I didn't take it personally

Separate yourself from your work

I needed to get back to work

I needed to get back to work

You're never done

If you do something and it turns out pretty good, then you should go do something else wonderful, not dwell on it too long. Just figure out what's next.

Steve Jobs

# Rewrite

The stories and lessons never end

Stories



Like A Crystal Ball

The Black Obelisk

The Hardest Problem

QWERTY

Every Word On Every Keystroke

We only need one of these, right?

Let's Try It 2% Darker

Rewrite

Know a good idea when you see it  
Don't try to solve every problem at once  
Find smart friends and listen to them  
Work should explain itself  
Choose the simplest thing which might work  
Only show your best work  
Iterating quickly leads to better work  
Be kind to people, but be honest about work  
Separate yourself from your work  
You're never done

My stories

Your stories

