

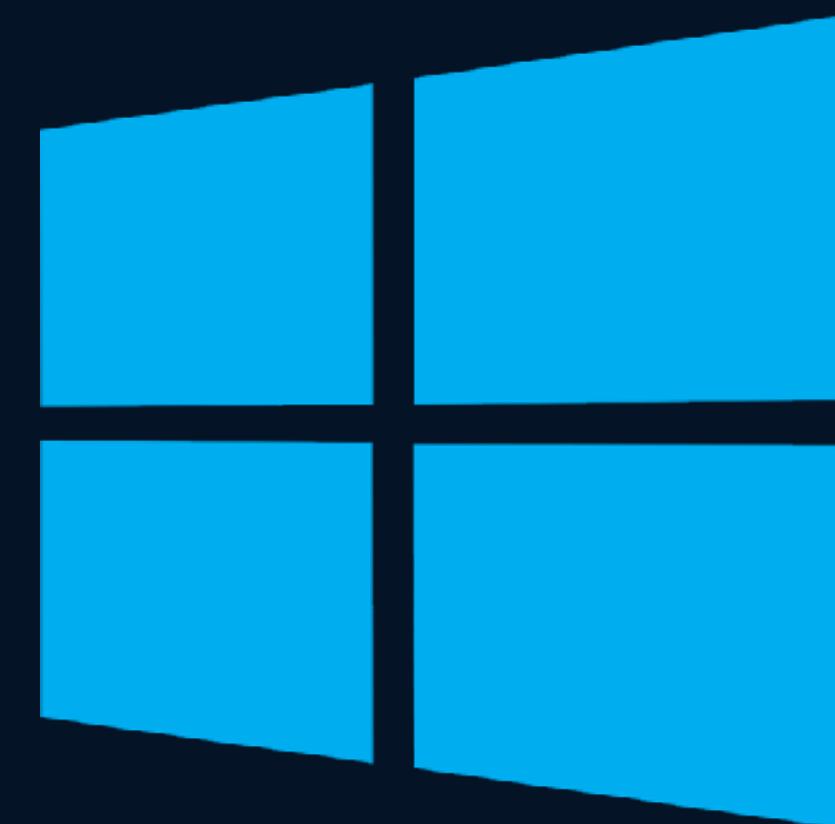


Please download Xcode! hackbca.com/ios
While installing, ensure you have administrator access.

Download our sound and image files at hackbca.com/ios - you'll be using them in this workshop.

iOS Development with SwiftUI

Anthony Li
Room 138

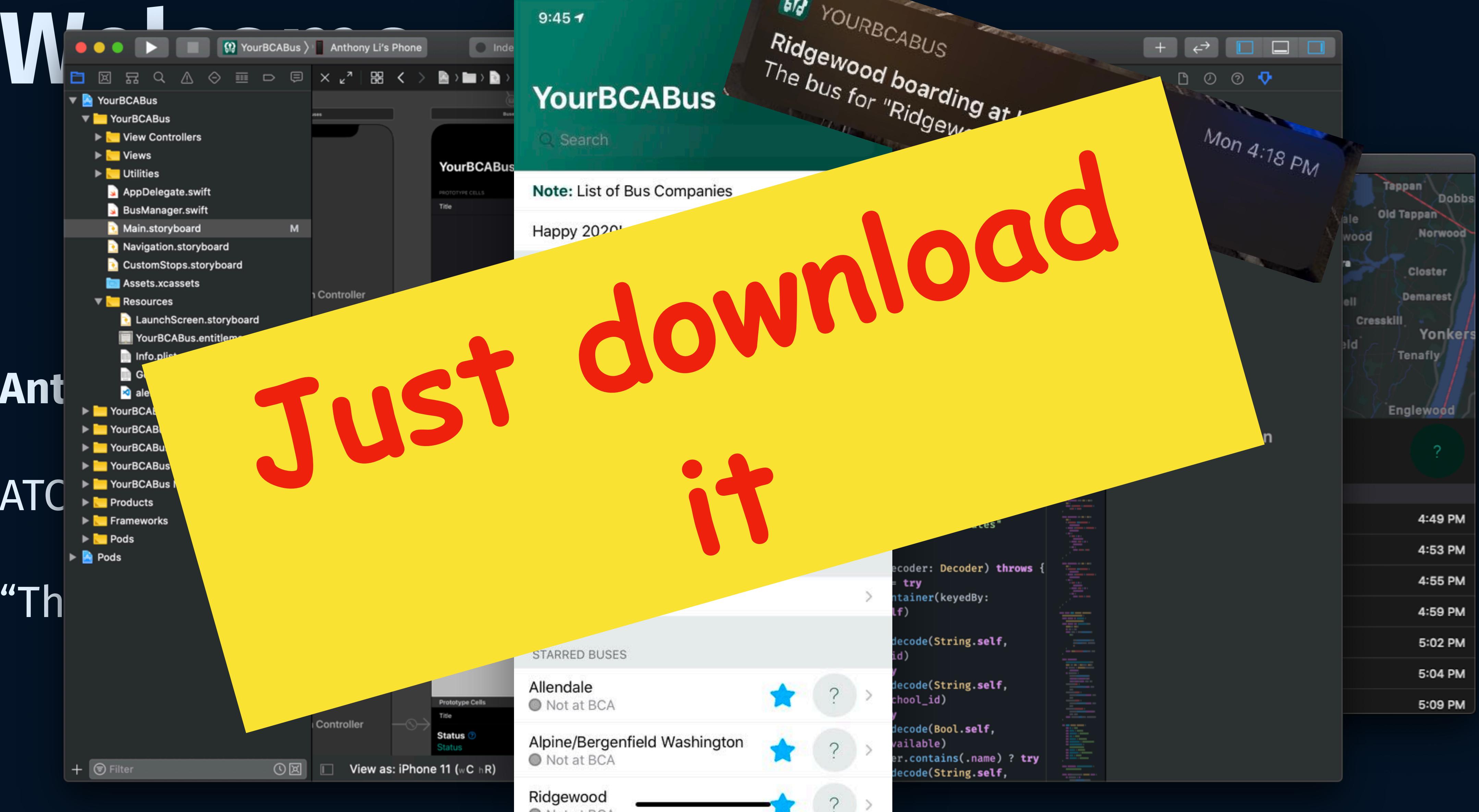


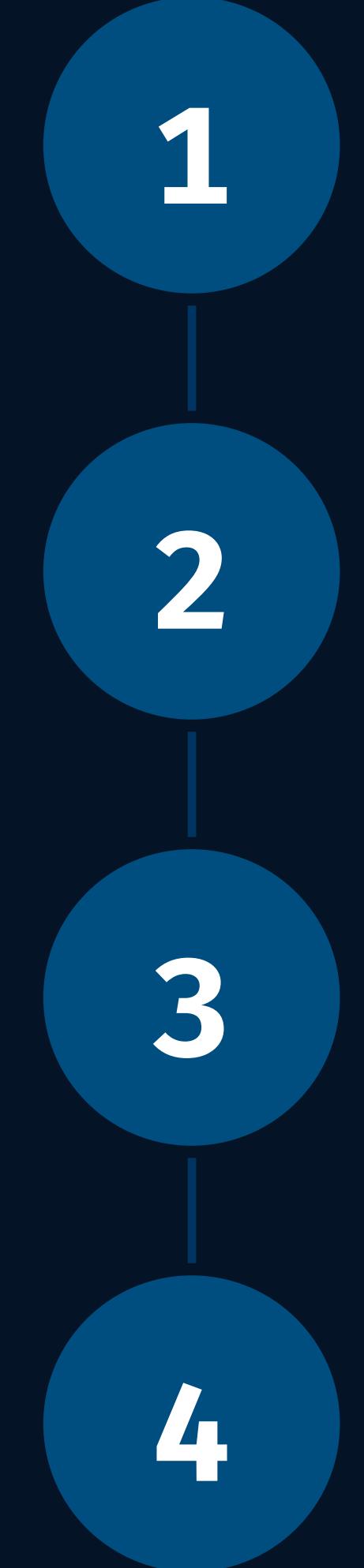
Link



Just download
it

Ant
ATC
“Th



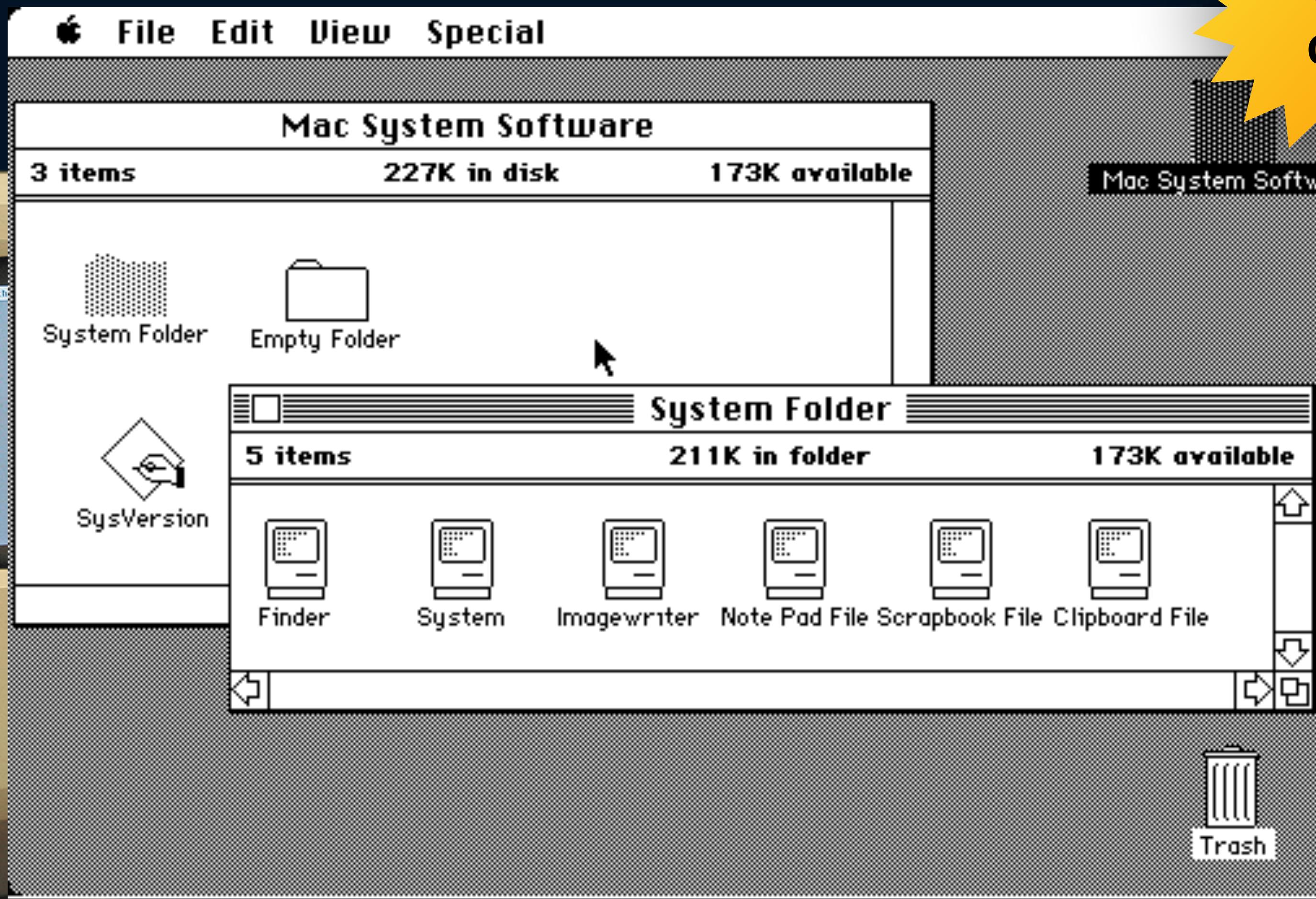
- 
- 1 History
 - 2 Introduction to Swift
 - 3 Duck Clicker
 - 4 hackBCA Schedule Viewer

History

- 13.8 billion years ago, there was a Big Bang.



1984



OG
GUI

osh

1984

Do you want to sell
sugar water for the rest of
your life, or do you want to
come with me and change the
world?



Steve Jobs



John Sculley

1985



®

btw ur fired now



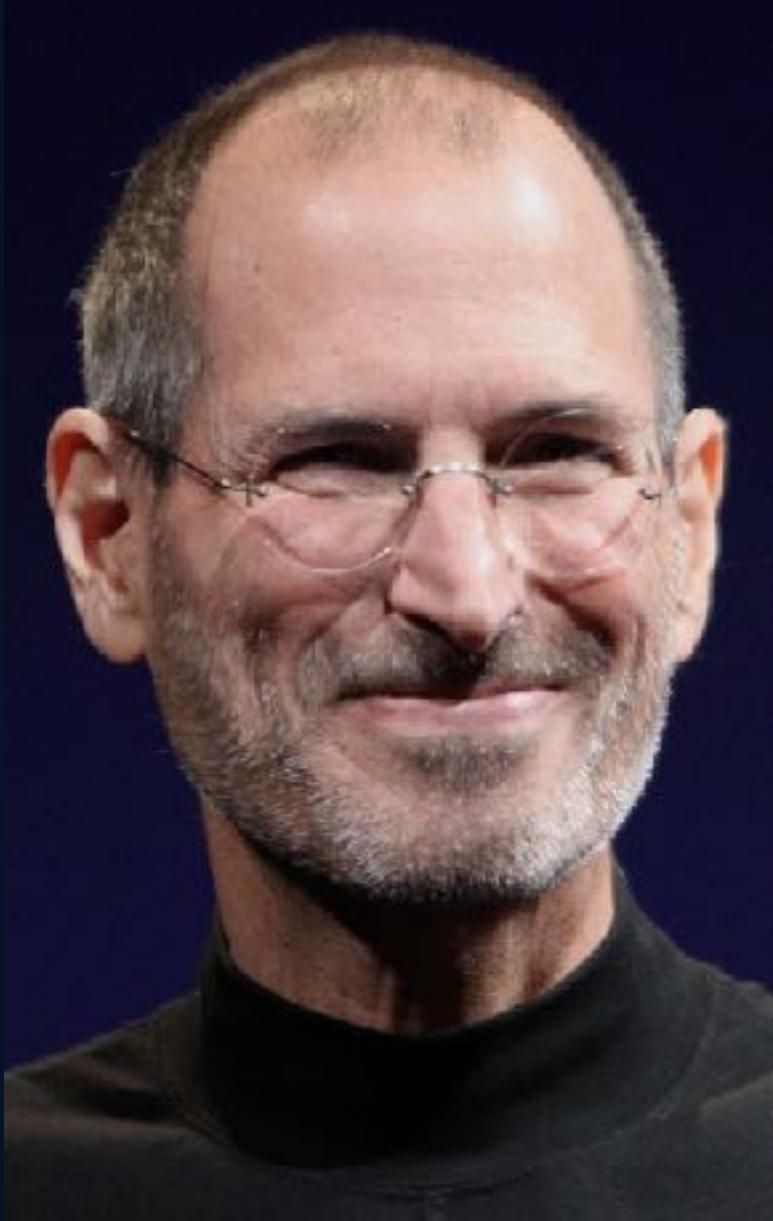
Steve Jobs

sure i guess



John Sculley

1985

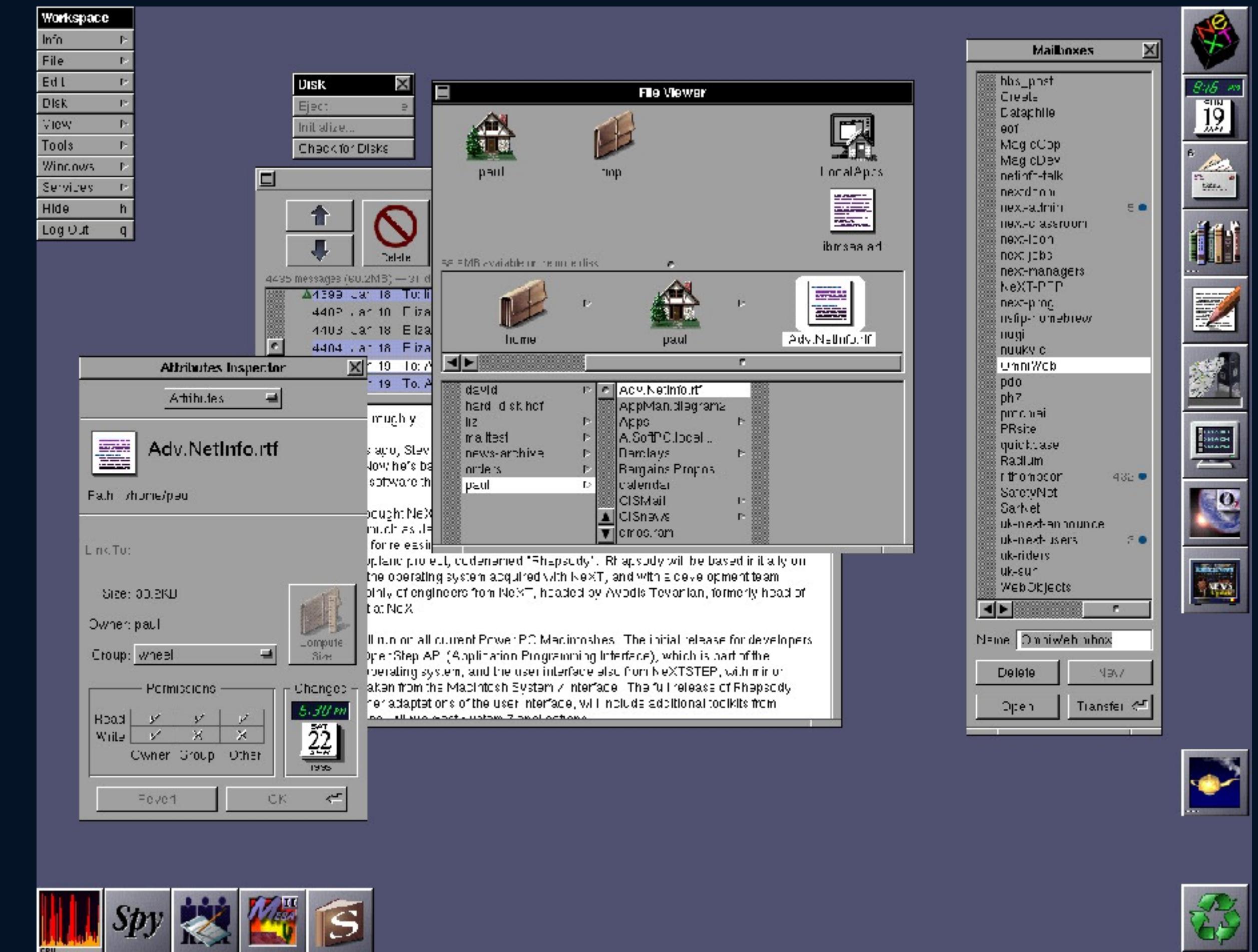
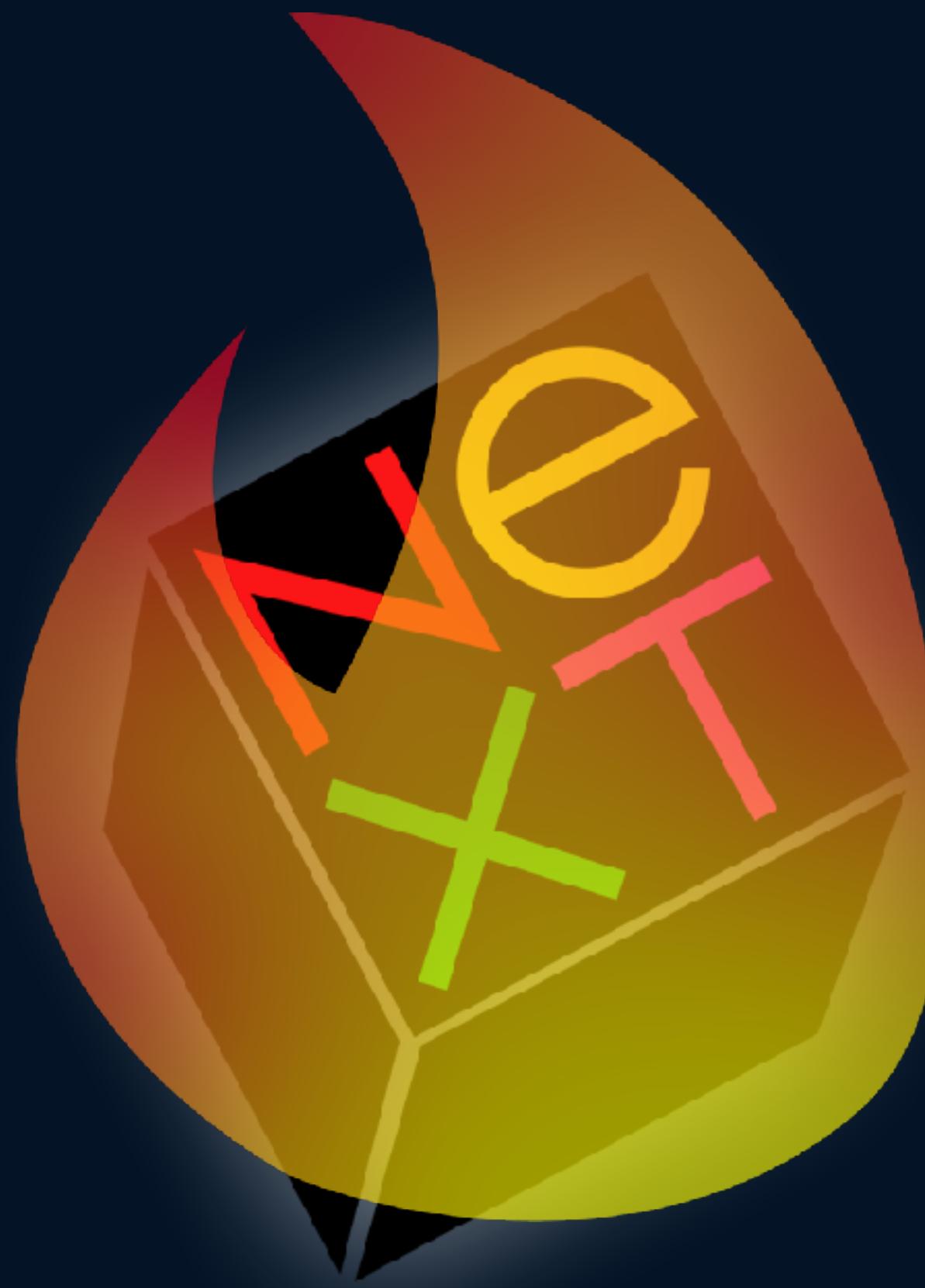


Steve Jobs



- Unix-based GUI!
- Object-oriented programming!
- Drag-and-drop app building!
- First computer to host a web server!

ONLY \$6,500!



1997



Steve Jobs

1997

Apple buys NeXT.





®



Mac OS X



**AppKit
Foundation**

UNIX

2007



iPhone OS



**AppKit “UIKit”
Foundation**

UNIX

2014

```
if (myDelegate != nil) {  
    if ([myDelegate respondsToSelector:  
        @selector(scrollViewDidScroll:)]) {  
        [myDelegate scrollViewDidScroll:myScrollView];  
    }  
}
```



```
myDelegate?.scrollViewDidScroll?(myScrollView)
```

Swift

Objective-C

2019



```
tfunc updateVisibleCells() {
    for indexPath in tableView.indexPathsForVisibleRows ?? [] {
        if let cell = tableView.cellForRow(at: indexPath) as? ThemeCell {
            cell.theme = model.items[indexPath.row]
        }
    }
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let identifier: String = "Cell"
    let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath) as! ThemeCell
    cell.theme = model.items[indexPath.row]
    return cell
}

override func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    model.selection = model.items[indexPath.row]
    tableView.deselectRow(at: indexPath, animated: false)
}
```

UIKit



```
struct ContentView : View {
    @State var model = Themes.listModel

    var body: some View {
        List(model.items, action: model.selectItem) { item in
            Image(item.image)
                .VStack(alignment: .leading) {
                    Text(item.title)
                    Text(item.subtitle).color(.gray)
                }
        }
    }
}
```

SwiftUI

iOS

Your Apps

UIKit

SwiftUI

Foundation

Quartz

Objective-C

Swift

UNIX

- 
- 1 History
 - 2 Introduction to Swift
 - 3 Duck Clicker
 - 4 hackBCA Schedule Viewer

1

History

2

Introduction to Swift

3

Duck Clicker

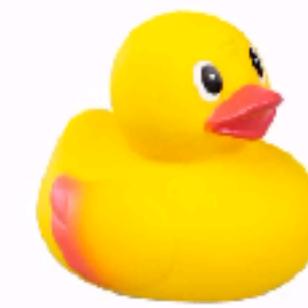
4

hackBCA Schedule Viewer

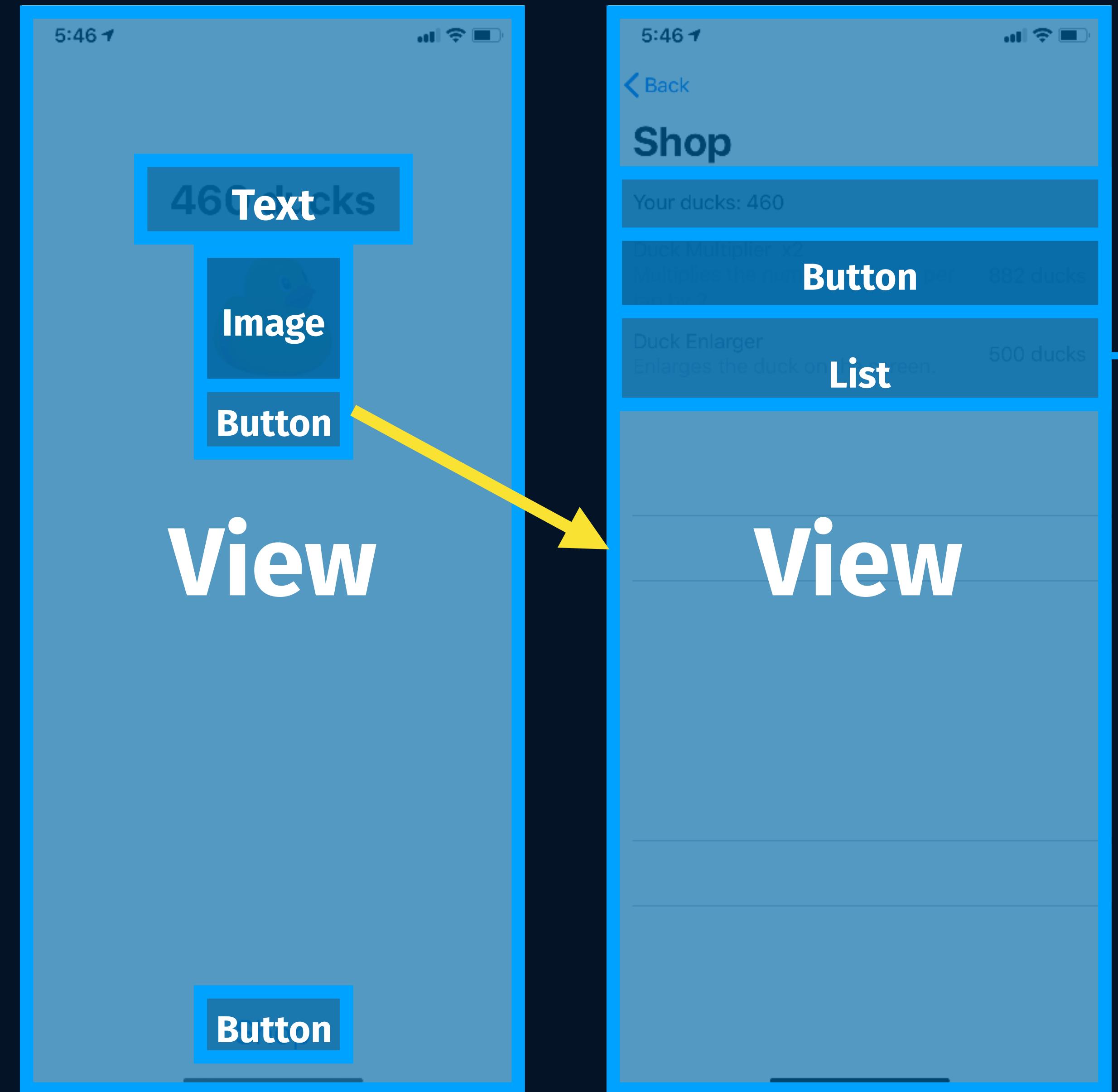
4:41 ↗



608 ducks



[Shop](#)



1

History

2

Introduction to Swift

3

Duck Clicker

4

hackBCA Schedule Viewer

Schedule

Registration	14:00 >	Gym
Opening Ceremony	15:30 >	Auditorium
Hacking begins	16:30 >	Gym
Intro to Web Development	17:00 >	Room 110
Christina Kaddouh		
Do you want to build your...		
Understanding Git Version Control	17:00 >	Room 138
Edward Feng		
Git is a widely-used versi...		
Dinner	18:00 >	
iOS Development with SwiftUI	19:00 >	Room 138
Anthony Li		
Dive into the world of dev...		
Vue.js	19:00 >	Room 138
Aidan Glickman		
Vue is a popular JavaScri...		
Hosting Websites with Github Pages	20:00 >	Room 110
Edward Feng		
After creating a website, y...		

Master

Intro to Web Development

17:00 - 19:30

Christina Kaddouh

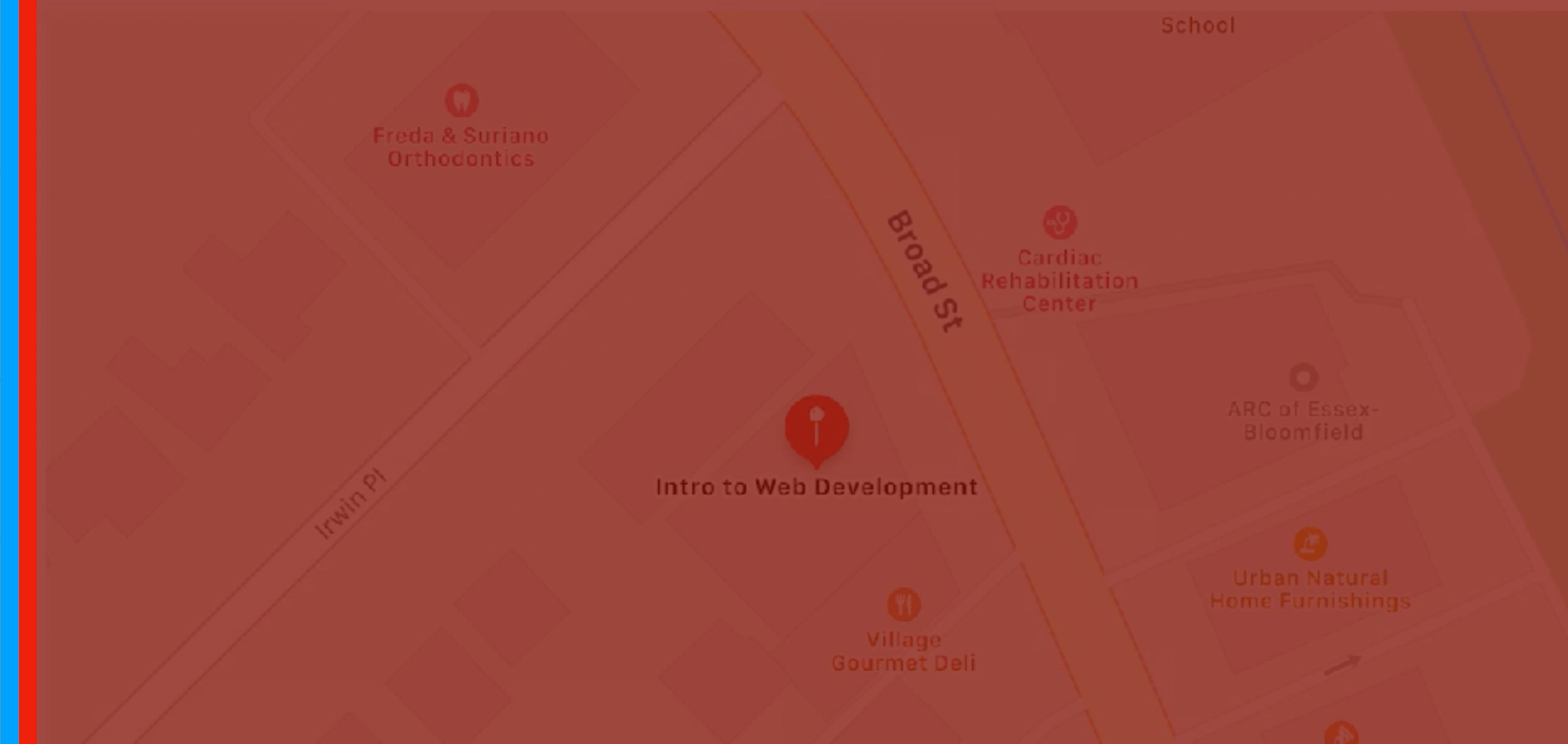
Room 110

Do you want to build your very own website using HTML and CSS, but don't know where to start? In this workshop, we will go into basic markup principles of HTML and CSS for beginners and teach you the skills you would need to create a website. This workshop includes a dinner break.

Requirements

You should have a proper text editor installed (eg. [Atom] (<https://atom.io/>) or [VSCode] (<https://code.visualstudio.com/>)).

Detail



Settings

Search



Anthony Li

Apple ID, iCloud, iTunes & App Store

Update Apple ID Settings

1



Airplane Mode



Wi-Fi

Freigen-Student



Bluetooth

On



VPN



Notifications



Sounds



Do Not Disturb



Screen Time

General

About



Software Update



AirDrop



Handoff



Multitasking & Dock



iPad Storage



Background App Refresh



Date & Time



Keyboard



Fonts

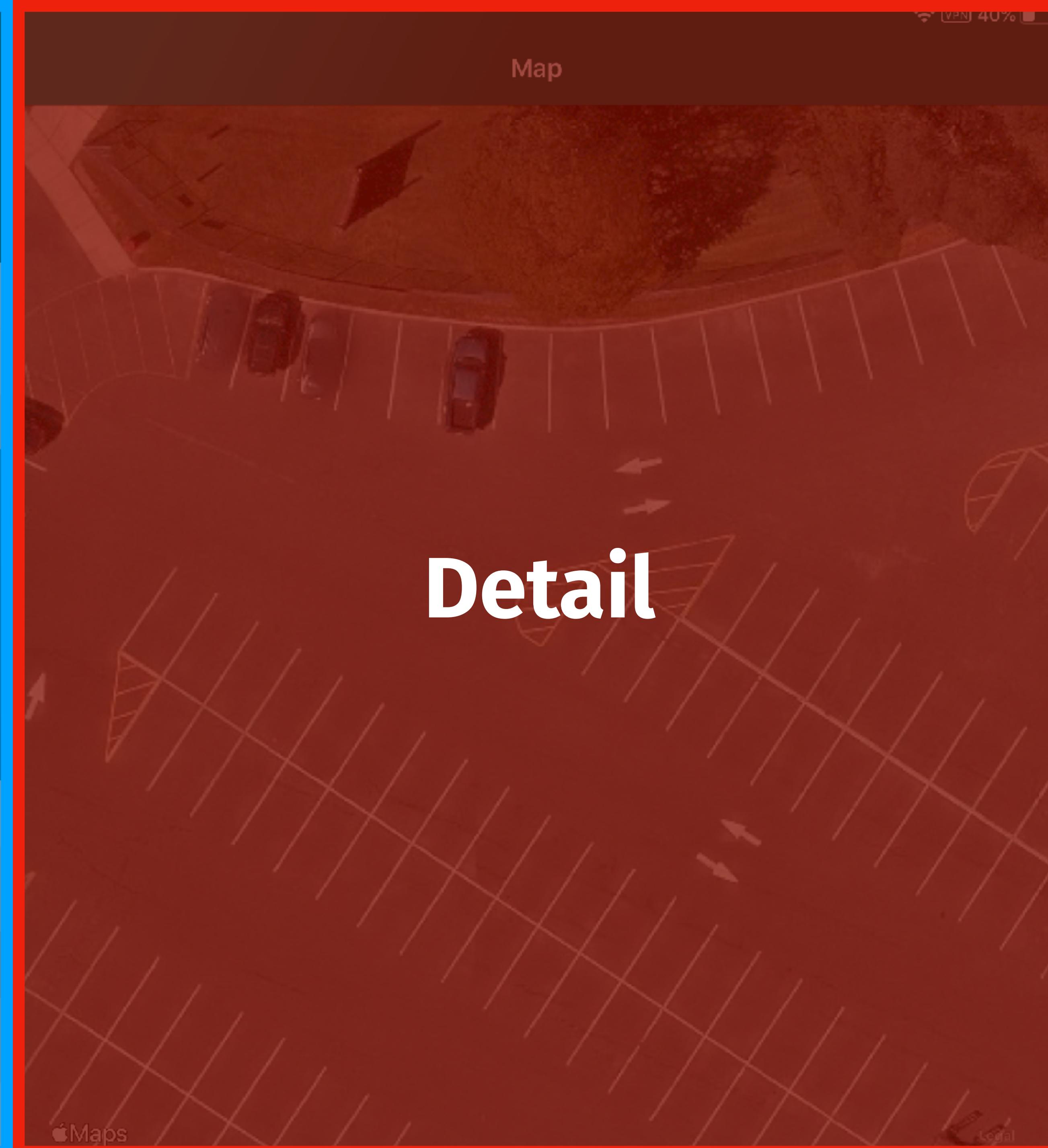
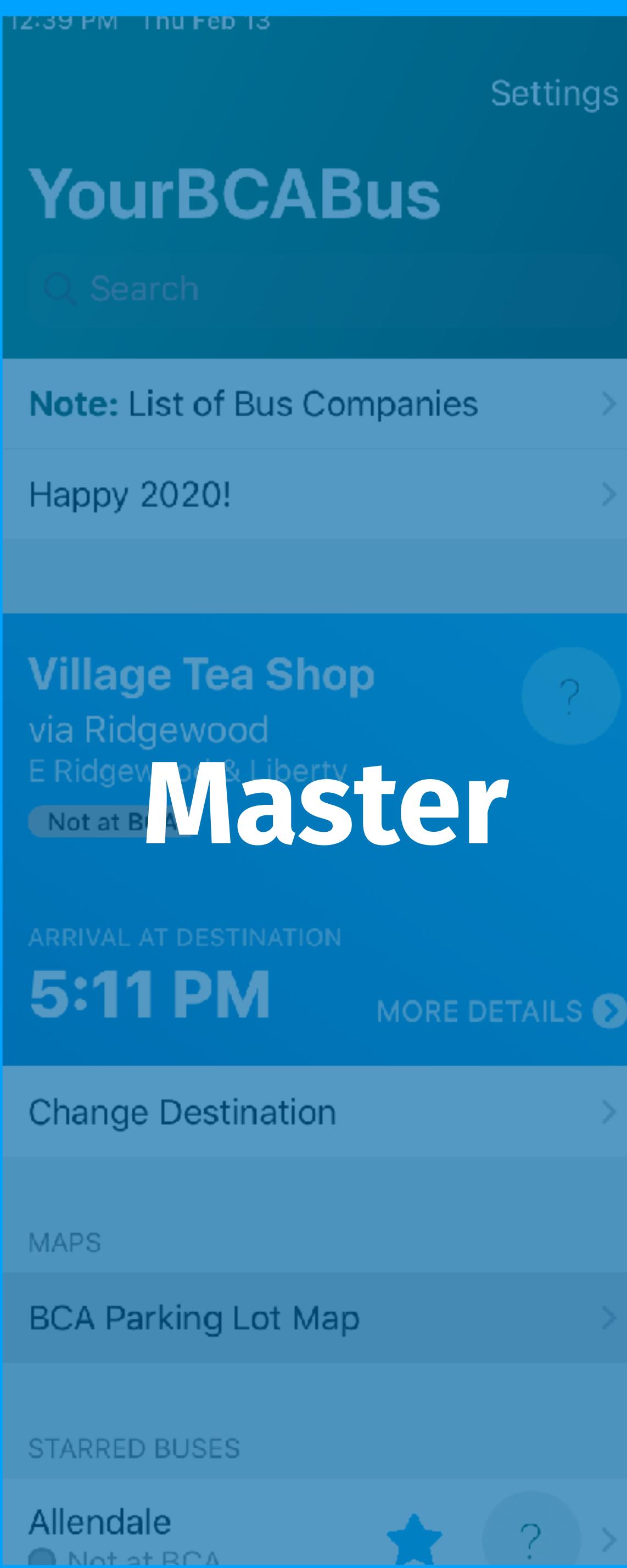


Language & Region



Dictionary

Detail



iOS

Your Apps

UIKit

SwiftUI

Foundation

Quartz

Objective-C

Swift

UNIX

SwiftUI

The SwiftUI framework provides a declarative, composable way to build user interfaces for iOS, macOS, tvOS, and watchOS. It's designed to make it easier to build complex UIs by letting you focus on what your app looks like rather than how it's built.

Topics

- Essentials**
 - [Learn to Make Apps with SwiftUI](#)
Follow a series of guided tutorials to learn to make apps using SwiftUI and Xcode.
- User Interface**
 - [Views and Controls](#)
Present your content onscreen and handle user interactions.
 - [View Layout and Presentation](#)
Combine views in stacks, generate groups and lists of views dynamically, and define view presentations and hierarchy.
 - [Drawing and Animation](#)
Enhance your views with a wide range of shapes, and shadows, and customize animated transitions between view states.
 - [Framework Integration](#)
Integrate SwiftUI views into existing apps, and embed AppKit, UIKit, and WatchKit views and controllers into SwiftUI view hierarchies.
- Data and Events**
 - [State and Data Flow](#)
Control and respond to the flow of data and changes within your app's models.
 - [Gestures](#)
Define interactions from taps, clicks, and swipes to fine-grained gestures.
- Previews in Xcode**
 - [Previews](#)
Generate dynamic, interactive previews of your custom views.

Topics

Essentials

- [About App Development with UIKit](#)
Learn about the basic support that UIKit and Xcode provide for your iOS and tvOS apps.
- [Protecting the User's Privacy](#)
Secure personal data, and respect user preferences for how data is used.

App Structure

UIKit manages your app's interactions with the system and provides classes for you to manage your app's data and resources.

UI

- [App and Environment](#)
Manage life-cycle events and your app's UI scenes, and get information about traits and the environment in which your app runs.

Documents, Data, and Pasteboard

Organize your app's data and share that data on the pasteboard.

Resource Management

Manage the images, strings, storyboards, and nib files that you use to implement your app's interface.

App Extensions

Extend your app's basic functionality to other parts of the system.

Inter-Process Communication

Share data through Handoff, support universal links into your app's content, and display activity-based services to the user.

Mac Catalyst

Create a version of your iPad app that users can run on a Mac device.

User Interface

Views help you display content onscreen and facilitate user interactions; view controllers help you manage views and the structure of your interface.

Views and Controls

Present your content onscreen and define the interactions allowed with that content.

View Controllers

Manage your interface using view controllers and facilitate navigation around your app's content.

View Layout

Use stack views to lay out the views of your interface automatically. Use Auto Layout when you require precise placement of your views.

Appearance Customization

Add Dark Mode support to your app, customize the appearance of bars, and use

Graphics, Drawing, and Printing

UIKit provides classes and protocols that help you configure your content and render your content.

Images and PDF

Create and manage images, including those that use bitmaps.

Drawing

Configure your app's drawing environment using colors, rendering contexts, and shadows.

Printing

Display the system print panels and manage the printing process.

Text

In addition to text views that make it easy to display text in your app, UIKit provides custom text management and rendering that supports the system's font and text styles.

Text Display and Fonts

Use UIKit views to display text, manage fonts, and check spelling.

Text Storage

Manage text storage, and coordinate the layout of text.

Keyboards and Input

Configure the system keyboard, or create your own keyboard for yourself.

UIKit

Deprecated

Avoid using deprecated classes and protocols in your apps.

Deprecated Symbols

Review symbols that are no longer supported, and find the ones instead.

Structures

```
struct NSDiffableDataSourceSnapshot
```

Classes

```
class UIActivityItemsConfiguration
```

```
class UIKey
```

Beta

Protocols

```
protocol UIActivityItemsConfigurationReading
```

```
protocol UIContextMenuInteractionAnimating
```

Schedule

Registration	14:00	Gym >
Opening Ceremony	15:30	Auditorium >
Hacking begins	16:30	Gym >
Intro to Web Development Christina Kaddouh Do you want to build your...	17:00	Room 110 >
Understanding Git/Version Control Edward Feng Git is a widely-used versi...	17:00	Room 138 >
Dinner	18:00	>
iOS Development with SwiftUI Anthony Li Dive into the world of dev...	19:00	Room 138 >
Vue.js Aidan Glickman Vue is a popular JavaScri...	19:00	Room 138 >
Hosting Websites with Github Pages Edward Feng After creating a website, y...	20:00	Room 110 >

Intro to Web Development

17:00 - 19:30

Christina Kaddouh

Room 110

Do you want to build your very own website using HTML and CSS, but don't know where to start? In this workshop, we will go into basic markup principles of HTML and CSS for beginners and teach you the skills you would need to create a website. This workshop includes a dinner break.

Requirements

You should have a proper text editor installed (eg. [Atom](<https://atom.io/>) or [VSCode](<https://code.visualstudio.com/>)).

