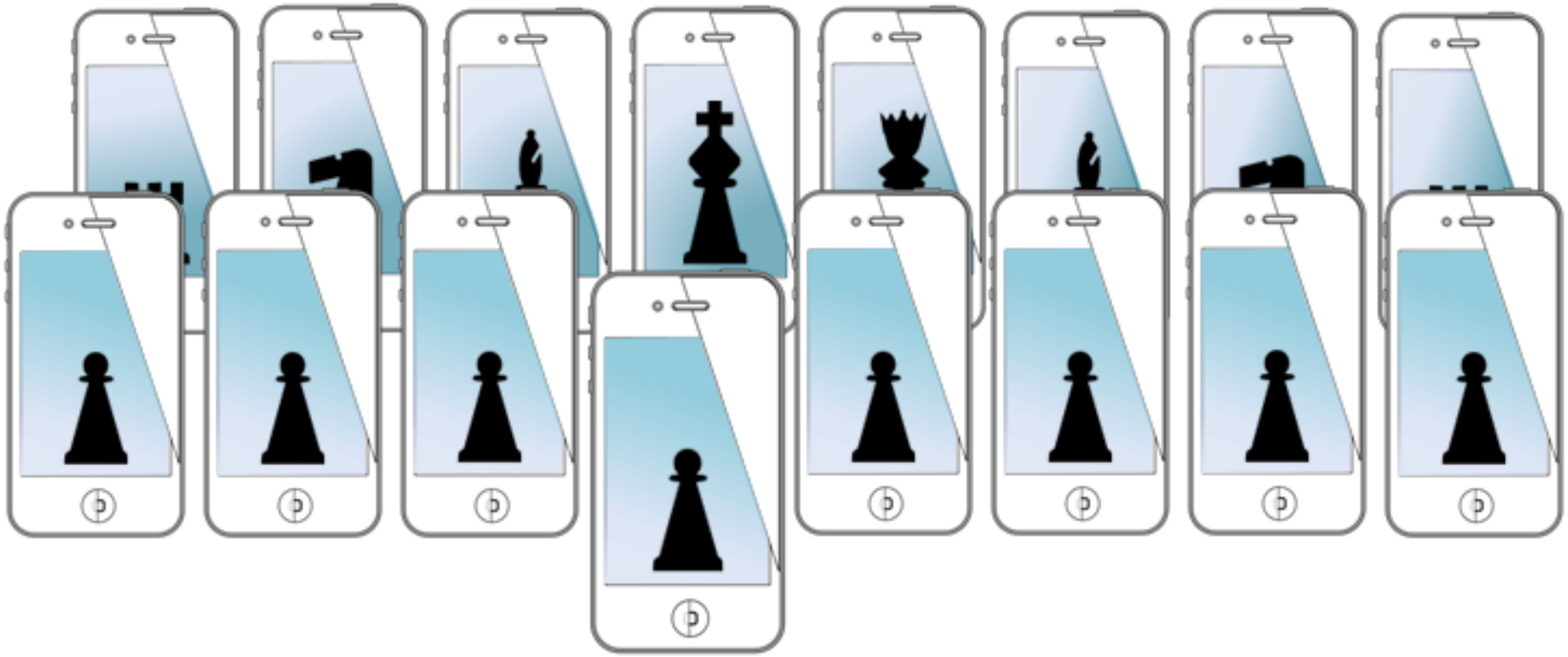
MOBILE SENSING & LEARNING



CS5323 & 7323

Mobile Sensing and Learning

course introduction

Eric C. Larson, Lyle School of Engineering,
Department of Computer Science, Southern Methodist University

agenda

- class logistics
- introductions
- what is this mobile sensing course?
    - and what this course is not…
    - course goals
    - syllabus (i.e., how to do well)
        - hardware, lab, grading, MOD
- iOS development platforms

course logistics

- lecture: in class, zoom, recorded
- lab: this class has no lab!
- office hours: Monday 3:30-5PM (Caruth 451, zoom?)
- we will use canvas for managing the course
- and GitHub for managing code:
  - https://github.com/SMU-MSLC
- Zoom étiquette

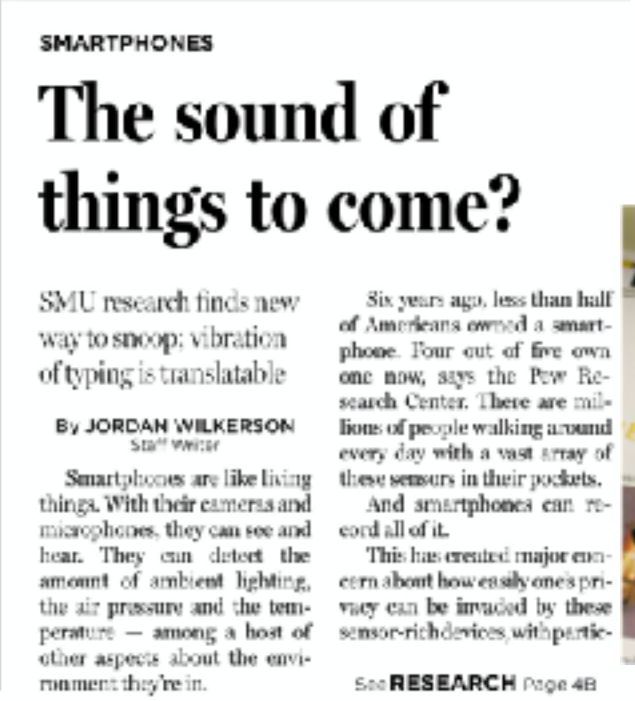
introductions

- education
  - undergrad and masters from Oklahoma State
  - PhD from the university of Washington, Seattle
- research
  - signal, image, and video processing (mobile)
    - how can combining DSP, machine learning, and sensing make seamless computing?
  - security
    - smartphone side channels
  - mobile health
    - moving outside the clinic: how mobile sensing can help patients and doctors
  - sustainability
    - how technology can increase awareness

http://eclarson.com

Phyn

Smart Water Assistant



SMARTPHONES

The sound of things to come?

SMU research finds new way to snoop: vibration of typing is translatable

By JORDAN WILKERSON
Staff Writer

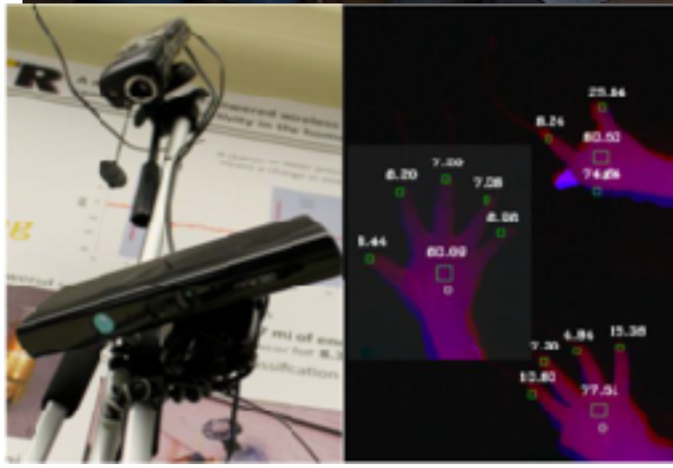
Smartphones are like living things. With their cameras and microphones, they can see and hear. They can detect the amount of ambient lighting, the air pressure and the temperature — among a host of other aspects about the environment they're in.

Six years ago, less than half of Americans owned a smartphone. Four out of five own one now, says the Pew Research Center. There are millions of people walking around every day with a vast array of these sensors in their pockets.

And smartphones can record all of it.

This has created major concern about how easily one's privacy can be invaded by these sensor-rich devices, with partic-

See RESEARCH Page 4B



introductions (if time)

- me
- about you:
  - name (what you go by)
  - grad/undergrad
  - department
  - **something true or false**

what is this course (and not)

- mobile sensing
  - activity recognition **some, yes!**
  - audio analysis **yes!**
  - vision analysis **yes!**
- machine learning **yes! treated as black box**
- microcontroller communication **no, not anymore**
- general iOS development **some basic skills**
- animation and graphics **no, except to display data**
- user interface design **some, all apps rely on user**

learning to learn

- for what we don’t cover: take the free Stanford iOS course!
- prerequisite: model based coding
  - because you will learn at least one new language:
    - objective-c
    - swift

    ← mostly these, **prefer swift**

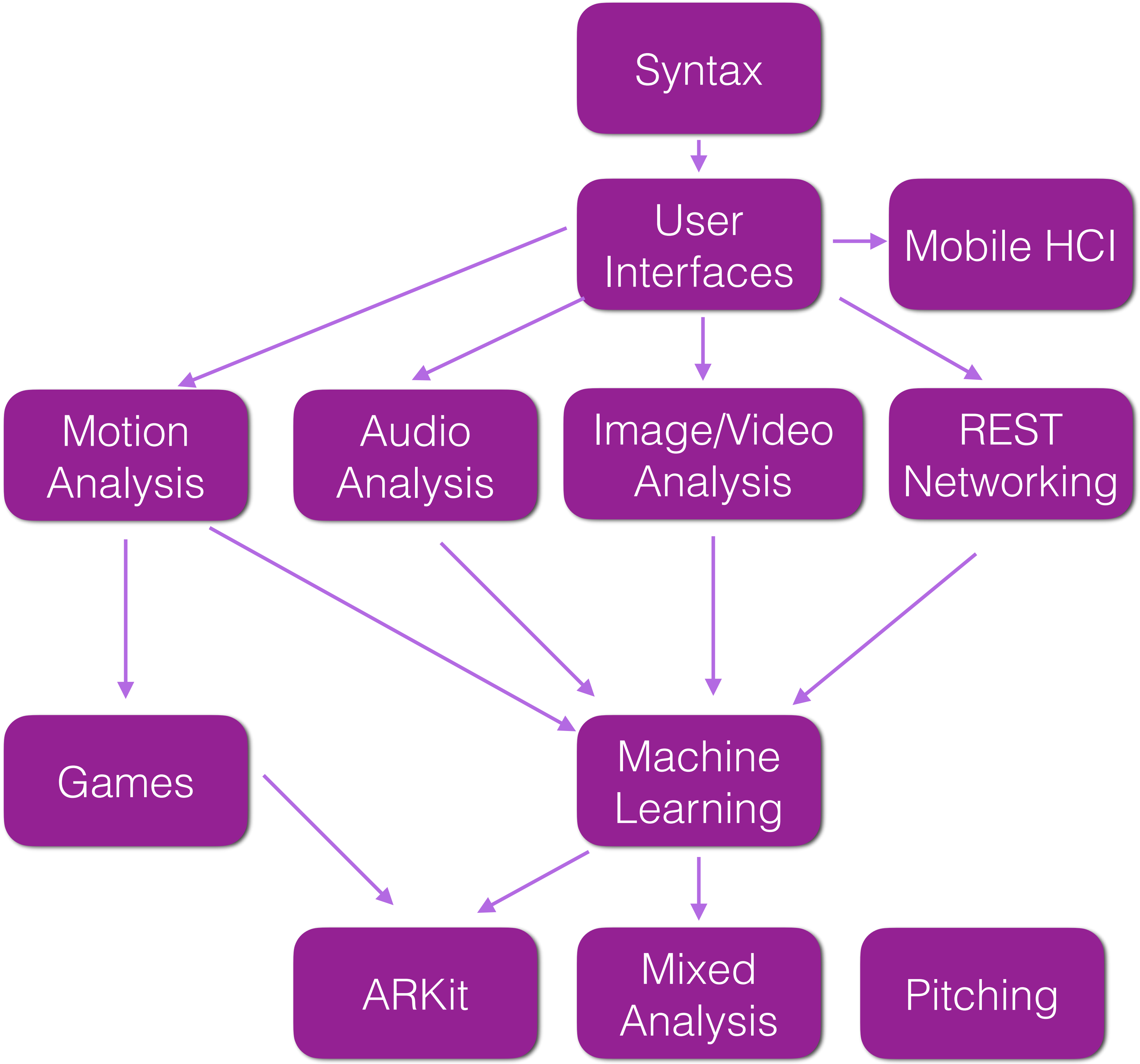
    - python

    ← for REST serving and ML training

    - c++
    - objective c++

    ← only for OpenCV

class overview



course goals

- exposure to iOS development, **MVCs** (not MVVM, SwiftUI)
- understand how to **use embedded sensors**
- **exposure** to **machine learning** for mobile sensors
  - use of built-in ML in iOS via coreML
- real time analysis of data streams
    - applications in health, education, security, etc.
- **present** and **pitch** applications

how to do well

- come to class or watch videos
- **start the app assignments early, with your team**
- iterate and test your apps
- use good coding practices, lazy instantiation, recycle classes, use comments

syllabus

- attendance
    - highly recommended, but you can watch video if needed
    - video of classes through Panopto (published after class)
- hardware is needed to develop apps
    - need a team formed (do this before the end of the week)
        - teams are expected to work remotely together
    - iPhones available for checkout, Xcode in library
    - preferable (required?) to use your own Mac
- Now let’s head over to canvas

syllabus (via canvas)

- grading
- flipped assignments
- final projects
- MOD

before next class

- look at canvas and GitHub repository (clone first repository)
- get a team together (groups of 1, 2, or 3, no exceptions)
  - contribute **equally**, **everyone** codes, **everyone** designs
  - **pick good members** with different skills than you
  - take turns **coding**
  - use the lab time for coding together
  - **you can change teams throughout semester**
- all assignments are already posted for the semester and all flipped module videos

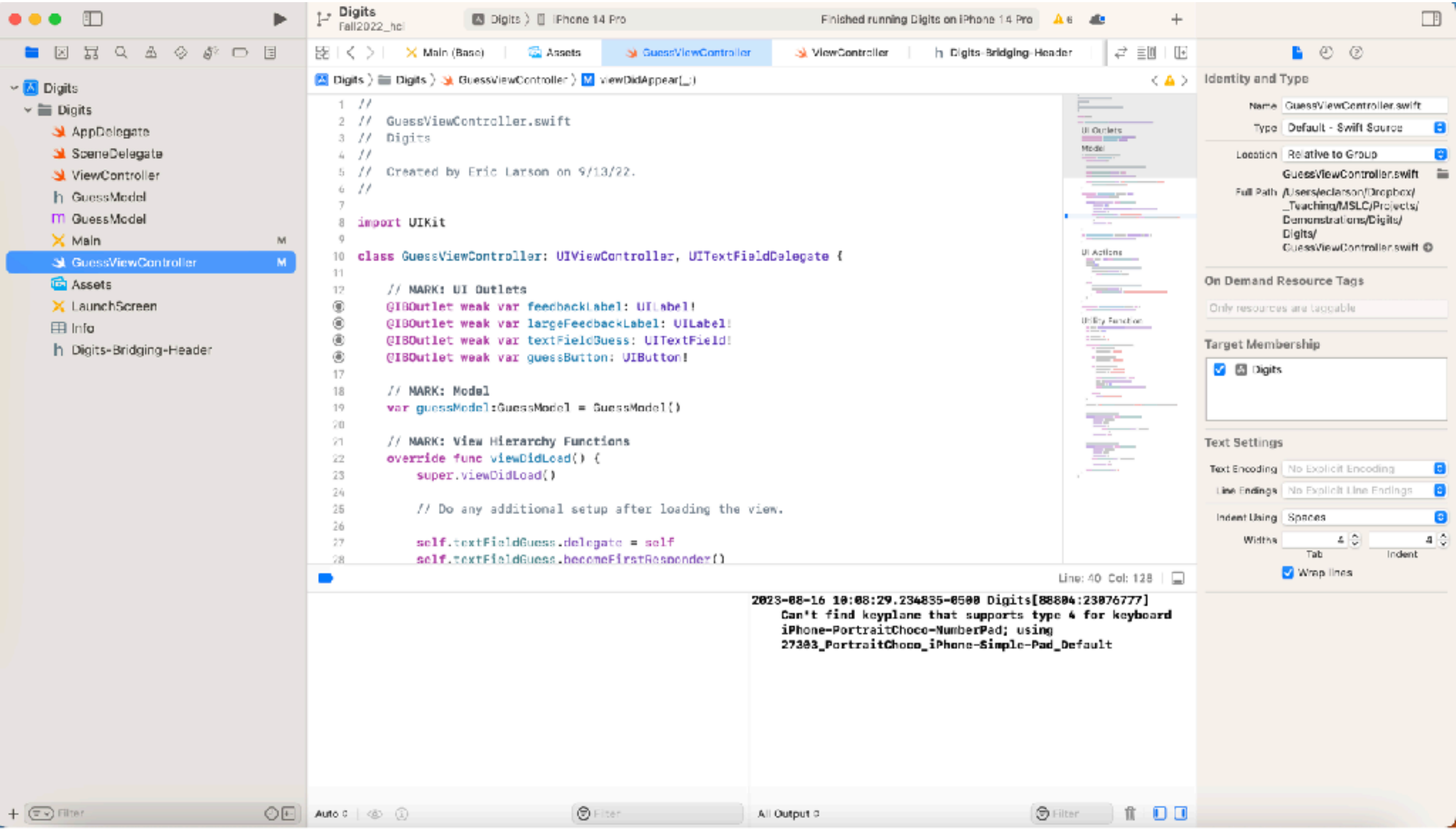
developing in iOS

- **cross platform**
  - React native, flutter, Xamarin (now MAUI), and others…
  - do not always support the latest hardware capability
  - using some phone sensors can be a pain (or slow)
  - works on many different phones
- **native (what we will be using)**
  - Xcode with objective-c and swift
  - limited to Apple, but free
  - packages are well supported, but has learning curve…
  - limited to ONLY iOS (and other Apple OS's)

Xcode

- best shown by example
- a very powerful and complex IDE
- user interface design through **SwiftUI** or **UIKit**
    - we will use UIKit exclusively (easier to get started)
    - storyboards are graphical layout editors that connect to classes for interaction
    - UIKit requires knowledge of auto layout for different sized screens

Xcode overview



```
//
//  GuessViewController.swift
//  Digits
//
//  Created by Eric Larson on 9/13/22.
//

import UIKit

class GuessViewController: UIViewController, UITextFieldDelegate {

    // MARK: UI Outlets
    @IBOutlet weak var feedbackLabel: UILabel!
    @IBOutlet weak var largeFeedbackLabel: UILabel!
    @IBOutlet weak var textFieldGuess: UITextField!
    @IBOutlet weak var guessButton: UIButton!

    // MARK: Model
    var guessModel:GuessModel = GuessModel()

    // MARK: View Hierarchy Functions
    override func viewDidLoad() {
        super.viewDidLoad()

        // Do any additional setup after loading the view.

        self.textFieldGuess.delegate = self
        self.textFieldGuess.becomeFirstResponder()
```

```
2023-08-16 10:08:29.234835-0500 Digits[88804:23076777]
    Can't find keyplane that supports type 4 for keyboard
    iPhone-PortraitChoco-NumberPad; using
    27303_PortraitChoco_iPhone-Simple-Pad_Default
```

Identity and Type

Name: GuessViewController.swift
Type: Default - Swift Source
Location: Relative to Group
GuessViewController.swift
Full Path: /Users/eclarson/Dropbox/_Teaching/MSLC/Projects/Demonstrations/Digits/Digits/GuessViewController.swift

On Demand Resource Tags

Only resources are taggable

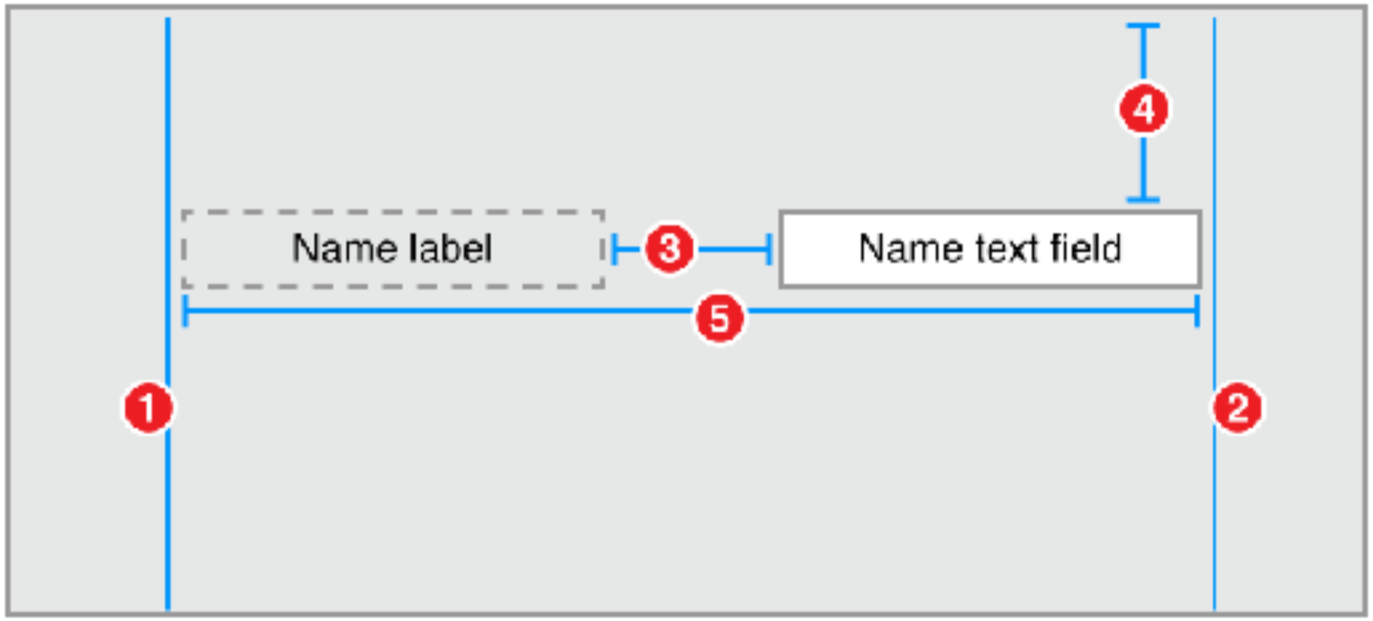
Target Membership

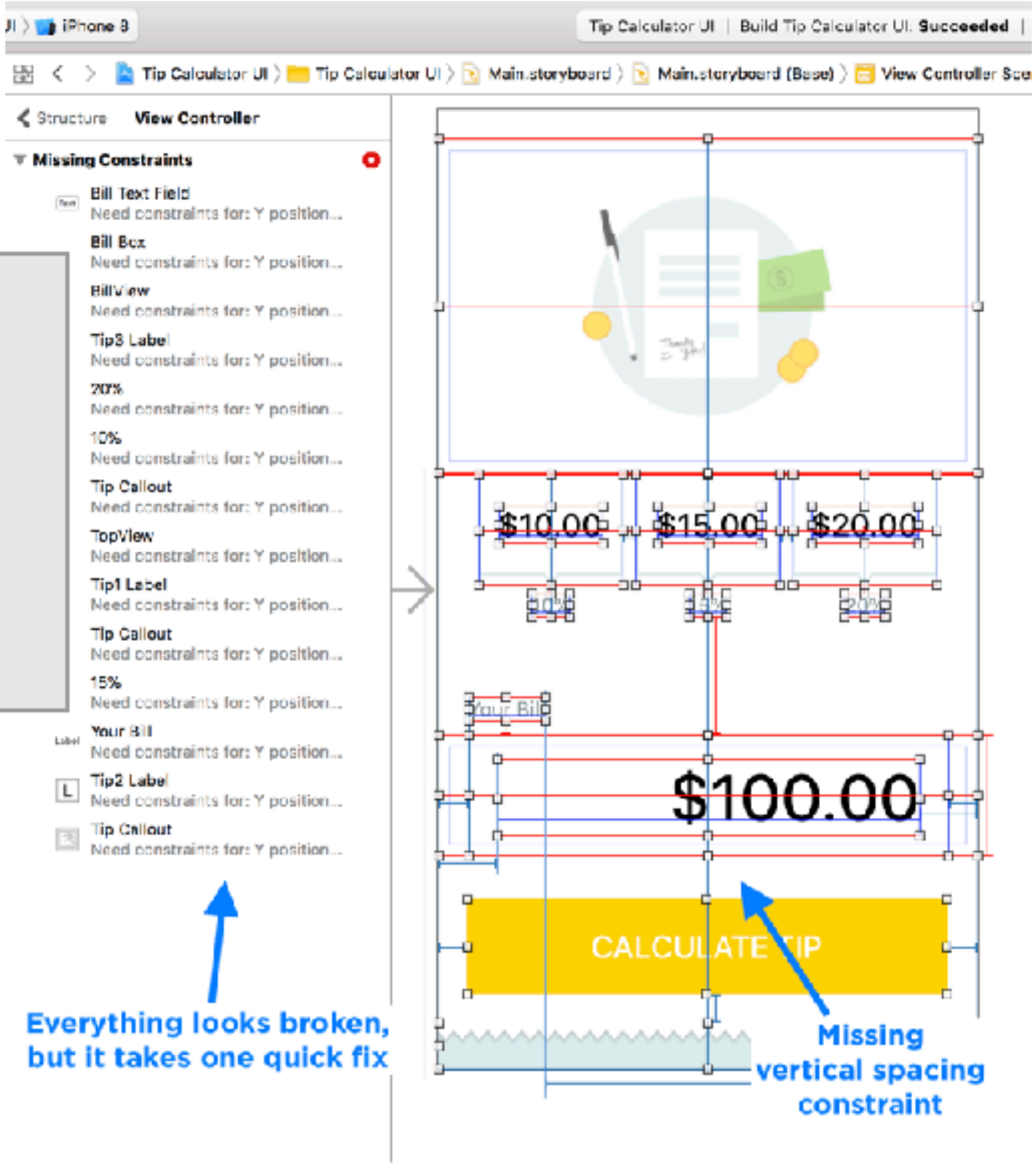
☑ Digits

Text Settings

Text Encoding: No Explicit Encoding
Line Endings: No Explicit Line Endings
Indent Using: Spaces
Widths: 4 Tab, 4 Indent
☑ Wrap lines

auto layout with storyboard





I 〉 iPhone 8
Tip Calculator UI | Build Tip Calculator UI: Succeeded |
Tip Calculator UI 〉 Tip Calculator UI 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 View Controller Sce
Structure View Controller

▼ Missing Constraints

Bill Text Field
Need constraints for: Y position...

Bill Box
Need constraints for: Y position...

BillView
Need constraints for: Y position...

Tip3 Label
Need constraints for: Y position...

20%
Need constraints for: Y position...

10%
Need constraints for: Y position...

Tip Callout
Need constraints for: Y position...

TopView
Need constraints for: Y position...

Tip1 Label
Need constraints for: Y position...

Tip Callout
Need constraints for: Y position...

15%
Need constraints for: Y position...

Your Bill
Need constraints for: Y position...

Tip2 Label
Need constraints for: Y position...

Tip Callout
Need constraints for: Y position...

Everything looks broken, but it takes one quick fix

Missing vertical spacing constraint

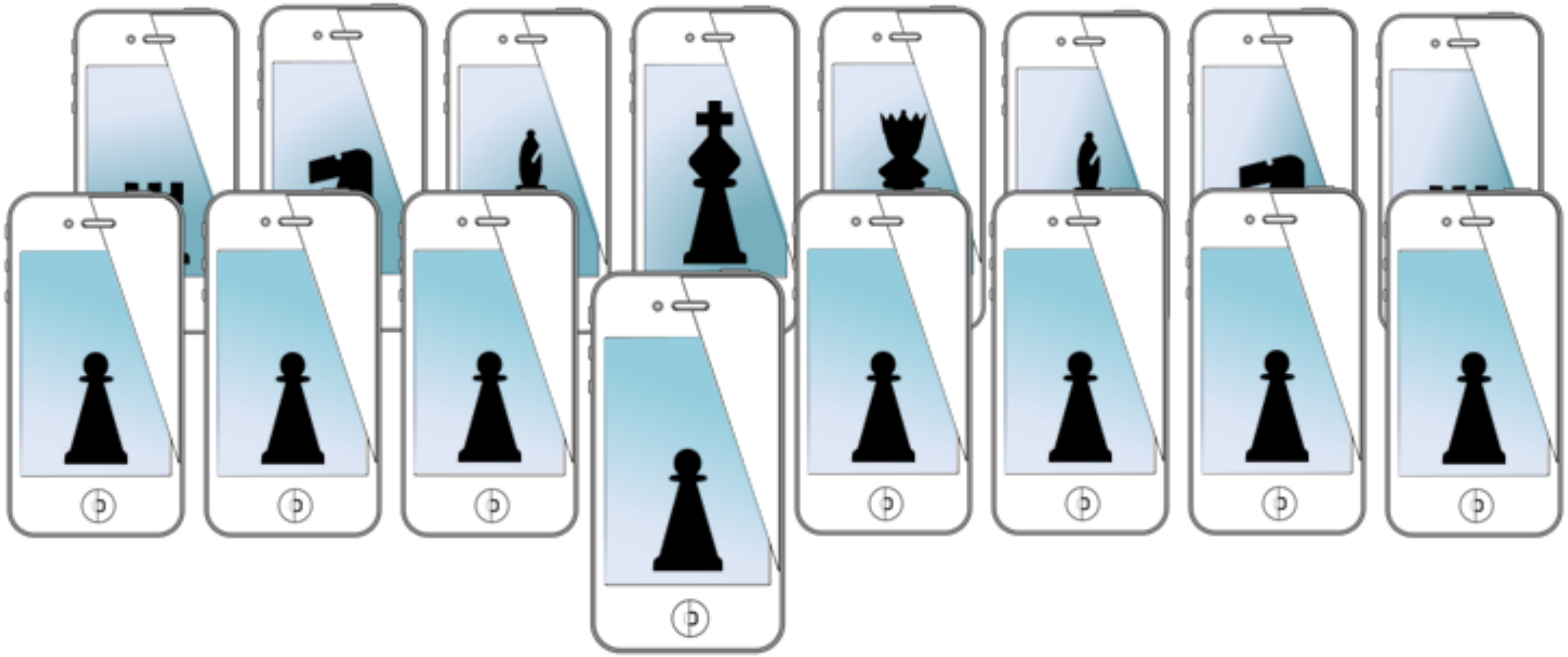
our first app with Xcode

- provides GUI for most git commands
  - commit, branch, push, pull, etc.
- **rarely** is command line needed
- git is great for code but not storyboards
- and some auto layout too!

for next time…

- have teams figured out
- find out how to launch Xcode on your team mac

MOBILE SENSING & LEARNING



CS5323 & 7323

Mobile Sensing and Learning

course introduction

Eric C. Larson, Lyle School of Engineering,
Department of Computer Science, Southern Methodist University