

CS4092 Special Topics in Computing I

Lecture 1

Welcome

- Cathal Hoare
- hoare@cs.ucc.ie
- Moodle Module: CS4092.2018 - self enrolment, no enrolment key
- Other resources as we encounter them (Apple documents, github, etc)

Plagiarism

1. Plagiarism is presenting someone else's work as your own. It is a violation of UCC Policy and there are strict and severe penalties.
2. You must read and comply with the UCC Policy on Plagiarism www.ucc.ie/en/exams/procedures-regulations/
3. The Policy applies to *all* work submitted, including software.
4. You can expect that your work will be checked for evidence of plagiarism or collusion.
5. In some circumstances it may be acceptable to reuse a small amount of work by others, but *only* if you provide explicit acknowledgement and justification.
6. If in doubt ask your module lecturer *prior* to submission. Better safe than sorry!

Whens and Wheres

- Lectures:
 - Monday 1200-1300 G.15 WGB
 - Friday 1100-1200 WGB 106
- Labs:
 - To be announced - none this week

Assessment

- End of Year Exam at end of Semester 1 - worth 80% of the mark
- 3 Assignments worth 2 x 6% and 1 x 8%
- A sample paper and solution will be provided
- We may substitute tutorials in lieu of labs on occasion
- Attendance will be kept

What You Will Learn

- Principles of developing mobile applications
 - Design - design robust, performant, usable applications
 - Development - use language features and tools to develop our applications
 - Testing - use language features and tools (and devices - simulators and real) to test our application
 - Deployment - use Apple Developer portal to support ad-hoc testing and also submit one of our apps to the Apple Store approval process.

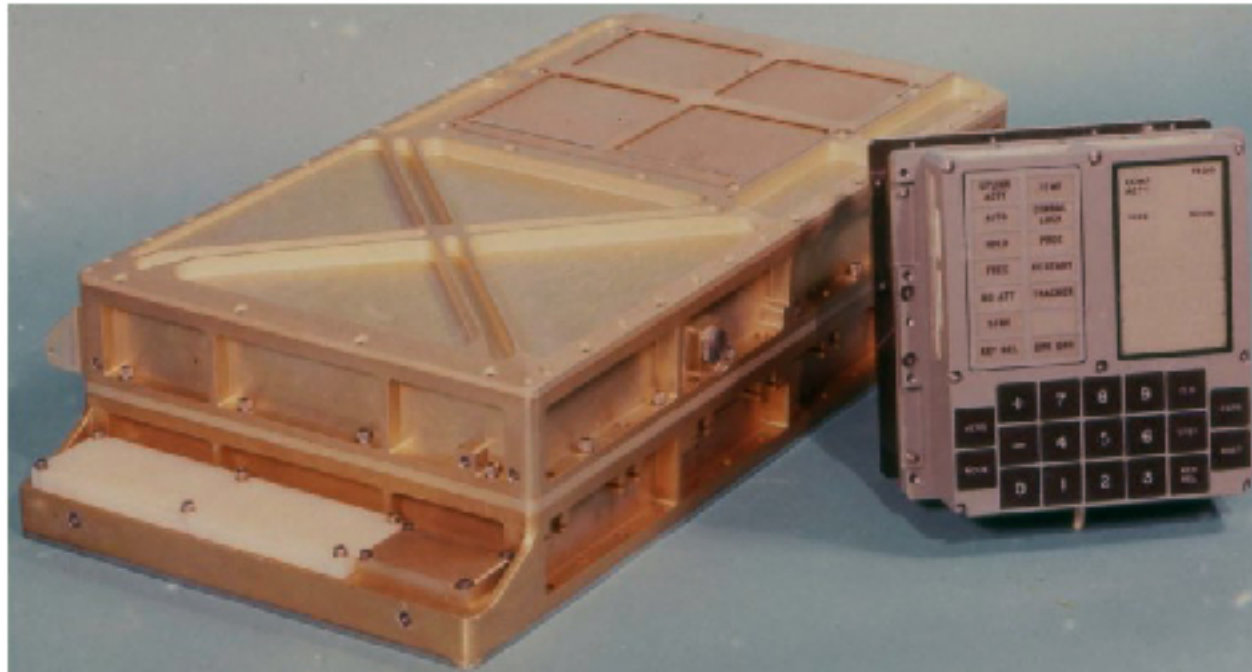
What You Will Learn

- We will do all of this through the iOS Platform using features of:
 - XCode and other development tools
 - Apple Developers Portal
- This is not a 'learn a language' module - we will learn what we need of Swift as we go.

Topics will include...

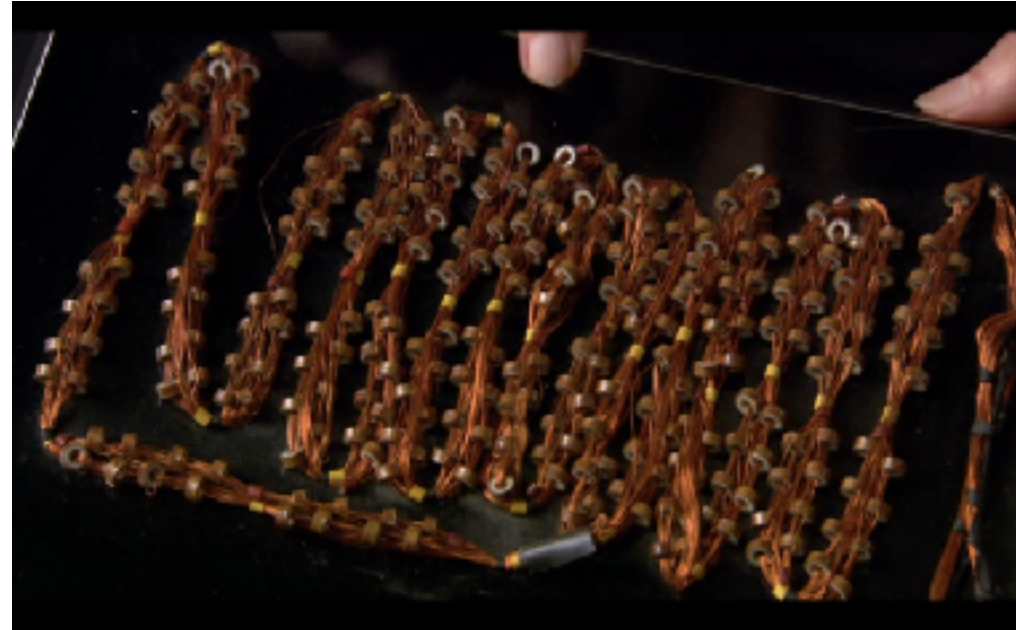
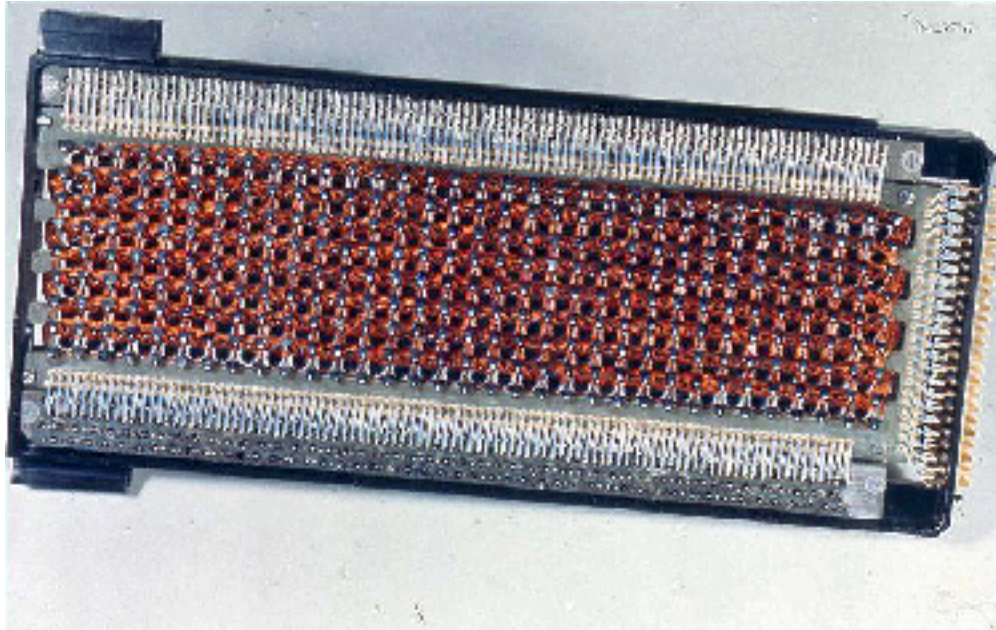
- App Design and Development Tools
- Views and View Controllers
- Segues
- Gestures
- Tables and Table View Controllers
- Making HTTP calls
- Firebase interaction
- Using Facebook to Authenticate
- Maps and Geolocation
- Animations
- 2nd Screens
- The App Store

The Computer that took us to the Moon



- 2.048 MHz, 4kB RAM, 72kB ROM (persistent storage)
- Inputs from inertial guidance system, rendezvous radar, landing radar, ground control and astronaut control inputs and output to spacecraft engine control system
- Landed man on the Moon....

The Computer that took us to the Moon



- ROM was 'written' by weaving 'rope memory'
- A magnetic doughnut and a wire represented a bit - a 1 was returned if the wire passed through the doughnut, a 0 otherwise.
- Development involved weaving. Debugging probably involved cursing.

iPhone 7s



- 2.3 MHz quad-core processor, separate GPU
- 2GB RAM, up to 256GB persistent storage
- An iPhone hasn't been to the Moon....

What else...

- Malleable Interface
- Sensor Rich
- Communications Rich
- Computing Rich
- We can add custom software
- So its uses are bounded by our imagination!

Two Starting Points

- A new app - a clean sheet:
 - We may or may not receive a clear set of requirements
 - If not we should help the app owner develop this list - we can storyboard and then prototype for them
- Modifying an app:
 - Where an app already exists, the owner will have a better idea about app development and what is possible. There may be data to help us decide on a development direction.

Lets start with a clean sheet

- We should emphasis some key points to the owner
 - We represent all stakeholders - the developer (us), the owner and the end user
 - We should also remember that there is a 4th...Apple

Basic Expectations of an App

Support key features such as Uniform system appearance,

Apps are expected to:

1. Provide privacy, polite execution
2. Behave as expected for particular events and controls
3. Follow well-defined execution paths
4. Should run well in multitasking environment
5. Be ready to respond, but also not unduly use resources such as battery

.

Spanner App

- This app will help plan, remind and document the maintenance schedule of your car
- It will be implemented as a classic List-Detail app where an initial table lists all instances of maintenance both planned and historic
- The table view can be filtered to show either planned or historic items
- Tapping on one of these will reveal a detailed view of the maintenance item
- It will be possible to add new items from the table view and also to delete items
- A profile of the car containing its vital details will also be available
- Items will be stored remotely on a server. It will be possible to receive notifications for planned items.
- As we learn new topics, we will add other features to this list.

Next Time...

- Design our first app and implement its data model...