

1

COS 470 - MOBILE DEVELOPMENT

---

# MOBILE DEVICES

Stephen Houser <[houser@maine.edu](mailto:houser@maine.edu)>

---

---

---

---

---

---

---

---

2

COS 470 MOBILE DEVELOPMENT

---

## MOBILE DEVICES

- ▶ History & Evolution
- ▶ Design Philosophy
- ▶ Mobile Architecture



---

---

---

---

---

---

---

---

3

COS 470 MOBILE DEVELOPMENT

---

## MOBILE DEVICE EVOLUTION

### Evolution of the Mobile Phone



Motorola 8900X-2    Nokia 2146    Nokia 3210    Nokia 6210    Ericsson T39    Alcatel OT511    Samsung E250    Apple iPhone    BlackBerry Curve 8900    Samsung Galaxy S2    Samsung Galaxy S4    Sony Xperia Z Ultra

---

---

---

---

---

---

---

---

COS 470 MOBILE DEVELOPMENT

## MOBILE DEVICE EVOLUTION



**iPhone X**

<b>DISPLAY</b>	Type	Super AMOLED capacitive touchscreen, 16M colors
	Size	5.8 inches, 1125 x 2436 pixels
	Multitouch	Yes
<b>PROCESSOR</b>	OS	iOS 11
	Chipset	Apple A11 Bionic
	CPU	Hexa-core 2x Monsoon + 4x Mistral
	GPU	Apple GPU (three-core graphics)
<b>CAMERA</b>	Primary	Dual 12 MP, phase detection
	Secondary	Geo-tagging, simultaneous 4K video recording, 2160p@24/30/60fps
<b>BATTERY &amp; MEMORY</b>	Battery	Non-removable Li-Ion battery
	Talk time	Up to 21 h (3G)
	Internal	64/256 GB, 3 GB RAM
	External	No

64GB ~~\$99,000\*~~  
\* Approx Price

256GB ~~\$1,02,000\*~~  
\* Approx Price

SAGMart.com

4

---

---

---

---

---

---

---

---

---

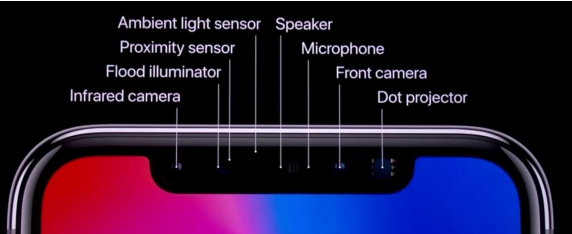
---

---

---

COS 470 MOBILE DEVELOPMENT

## MOBILE DEVICE EVOLUTION



Ambient light sensor  
Proximity sensor  
Flood illuminator  
Infrared camera  
Speaker  
Microphone  
Front camera  
Dot projector

5

---

---

---

---

---

---

---

---

---

---

---

---

COS 470 MOBILE DEVELOPMENT

## MOBILE DEVICE EVOLUTION

### LG G5 specs overview

<b>4-core, 2.2 GHz</b>	<b>2800 mAh</b>	
CPU: Qualcomm Snapdragon 820		
GPU: Adreno 530		
RAM: 4 GB RAM	<b>LTE</b>	
STORAGE: 32 GB	HSPA, HSUPA, UMTS, EDGE, GPRS	
	NFC, Wi-Fi 802.11, Bluetooth 4.2	
 <p>2.91" (73.9 mm) 5.38" (136.6 mm) 5.61 oz (159 g) 0.29" (7.3 mm) thickness 5.3" (134.1 mm) 1440 x 2560 pixels 577 ppi, IPS LCD</p>	<b>16 megapixels</b>	
	REAR: F1.8 aperture, sensor 1/2.6"	
	Front: 8 Megapixels Video capture: 3840x2160 (4K)	
	<b>Android (6.0)</b>	

6

---

---

---

---

---

---

---

---

---

---

---

---

**MOBILE DEVICE EVOLUTION**



---

---

---

---

---

---

---

---

---

---

**MOBILE DEVICE EVOLUTION**

- ▶ Better batteries - longer lasting
- ▶ Smaller components, displays
- ▶ More powerful processors and graphics
- ▶ Cameras!

---

---

---

---

---

---

---

---

---

---

**APPLICATIONS**

- ▶ Who's going to write all these applications?
  - ▶ Wireless Application Protocol (WAP)
  - ▶ Effectively stripped down HTML
  - ▶ Java Applications as well
- ▶ Smart (programmable) Phones
  - ▶ Microsoft, Apple, Google

---

---

---

---

---

---

---

---

---

---

### PLATFORMS

- ▶ Apple iOS
  - ▶ iPod + Phone + Web Browser
  - ▶ First widely used "app store"
- ▶ Google Android
  - ▶ Operating System, not hardware
  - ▶ Open Source, no developer fees, alternate "stores"
- ▶ Microsoft
  - ▶ "Windows Everywhere", never really caught on.

---

---

---

---

---

---

---

---

---

---

### APPLE IOS DEVELOPMENT

- ▶ iPhones and iPads, rumors of inclusion in macOS
- ▶ Objective-C or Swift for native development
- ▶ Requires macOS to bundle for distribution\*
- ▶ Tight security controls on app distribution
- ▶ Subject to Apple's approval process
- ▶ Here lies the most money ([App Annie](#) 2015)

---

---

---

---

---

---

---

---

---

---

### ANDROID DEVELOPMENT

- ▶ Thousands of devices from multiple manufacturers
  - ▶ varying degrees of "upgradeability"
- ▶ Java using Android Studio or native ARM (C/C++)
- ▶ Open Source
- ▶ Google Play store is \$25 one-time fee
- ▶ Alternate "stores" and alternate distribution channels

---

---

---

---

---

---

---

---

---

---

13

### HYBRID DEVELOPMENT

- ▶ Apache Cordova (open source)
  - ▶ HTML, CSS, & JS
- ▶ Xamarin
  - ▶ C#
  - ▶ Part of Visual Studio (Microsoft)
- ▶ React Native (open source, facebook)
  - ▶ HTML, JS, & Native UI elements

---

---

---

---

---

---

---

---

---

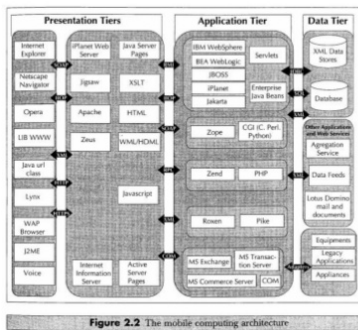
---

---

---

14

### DESIGN PHILOSOPHY



- ▶ Both follow 3-tier
- ▶ Presentation (View)
- ▶ Logic/Application (Controller)
- ▶ Data (Model)

---

---

---

---

---

---

---

---

---

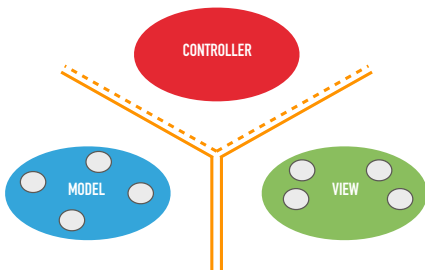
---

---

---

15

### DESIGN PHILOSOPHY




---

---

---

---

---

---

---

---

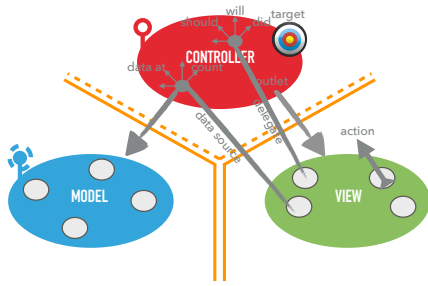
---

---

---

---

### DESIGN PHILOSOPHY



16

---

---

---

---

---

---

---

---

---

---

### 3-TIER ARCHITECTURE

- ▶ 3-Tier applies to many design scenarios
  - ▶ Keep presentation lightweight, independent of rest
  - ▶ Keep logic separate and encapsulated, easy to modify
  - ▶ Keep data separate. NEVER build based on current values
- ▶ How does it apply to websites and desktop applications?

17

---

---

---

---

---

---

---

---

---

---

### 3-TIER ARCHITECTURE

- ▶ Controller
  - ▶ Role is a "traffic cop"
  - ▶ Takes requests from user and turns into method calls
  - ▶ Finds the right model to use
  - ▶ Finds the right view to use
  - ▶ Returns all that back to the user

18

---

---

---

---

---

---

---

---

---

---

### 3-TIER ARCHITECTURE

- ▶ Model
  - ▶ Representation of the data
  - ▶ May be linked to a database or remote "server"
    - ▶ Can be much more complicated than that!
  - ▶ Contains relationship rules (1-many, many-many)

---

---

---

---

---

---

---

---

---

---

### 3-TIER ARCHITECTURE

- ▶ View
  - ▶ Template to be populated with data from the model
  - ▶ All UI components go here
  - ▶ May have internal state data (non-persistent)

---

---

---

---

---

---

---

---

---

---

### CLIENT SERVER

- ▶ Client/Server Architecture
  - ▶ Connect local application (views) with large backend data and services
  - ▶ How much should be done...
    - ▶ Locally (on device)?
    - ▶ Remote (in cloud)?

---

---

---

---

---

---

---

---

---

---

### CLIENT SERVER

- ▶ Thick Client
  - ▶ business logic and some data on device.
  - ▶ "Off the grid" capable
- ▶ Thin Client
  - ▶ Business logic and data in the cloud
  - ▶ Must be connected
- ▶ Which is better? When?

---

---

---

---

---

---

---

---

---

---

### MOBILE DEVICES

- ▶ Thick Client
  - ▶ business logic and some data on device.
  - ▶ "Off the grid" capable
- ▶ Thin Client
  - ▶ Business logic and data in the cloud
  - ▶ Must be connected
- ▶ Which is better? When?

---

---

---

---

---

---

---

---

---

---

### ARCHITECTURE

- ✓ History & Evolution
- ✓ Design Philosophy
- ✓ Mobile Architecture

---

---

---

---

---

---

---

---

---

---