



Introduction to Mobile App Development

Dmytro Zubov, PhD

dmytro.zubov@ucentralasia.org



Introduction to Mobile App Development

Mobile Apps are a new paradigm of application development

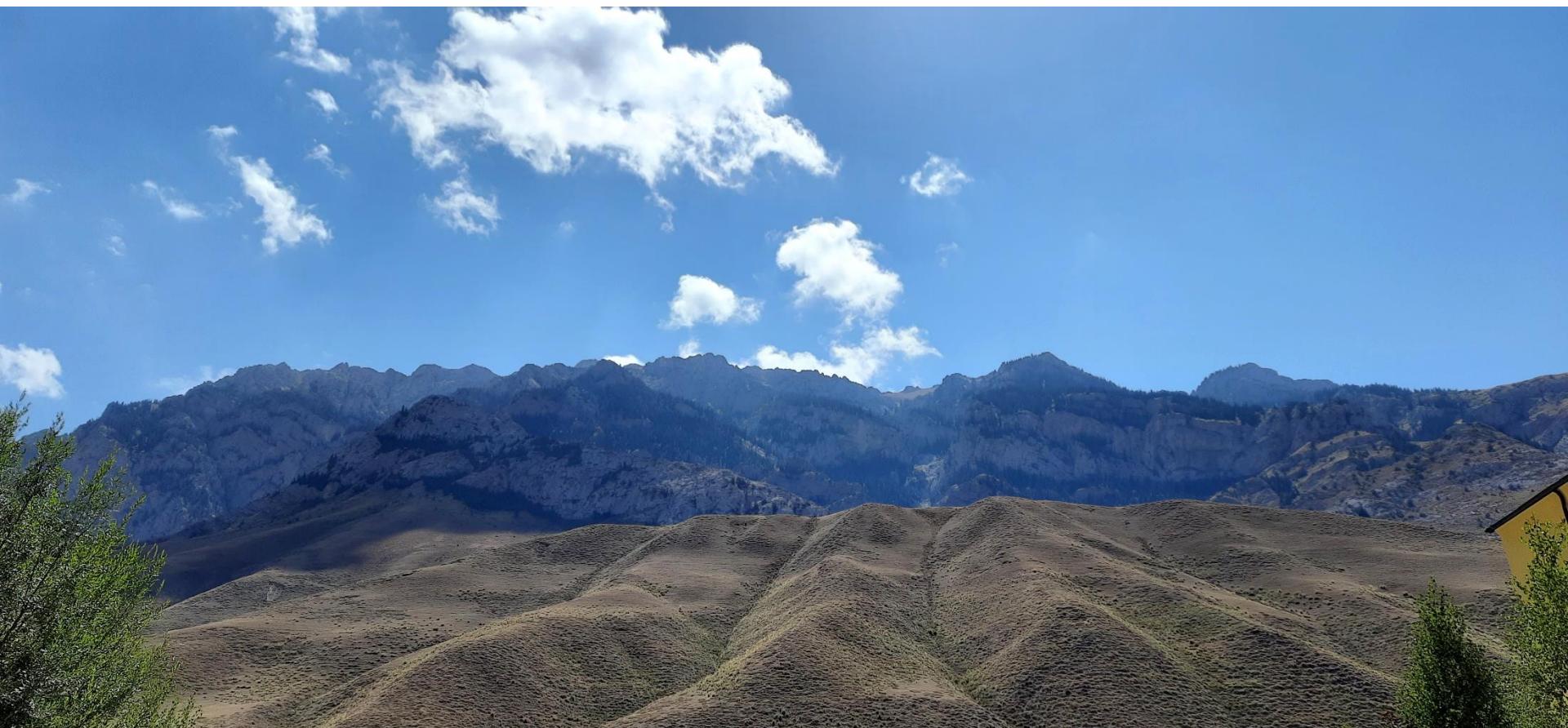


What we gonna discuss today?

- Outline of the course
- Syllabus
- Course project
- Intro to Mobile App development
- HTML5 offline web apps
- Present yourself in 5 minutes: background, interests, ideas for the course project and the final project (3rd period this week?)
- An example of a simple Android app development? (if we have enough time...)



Naryn, 1:36 pm, August 29, 2022



Outline of the course

- Intro to Mobile App Development
- Intro to Java programming language
- History of Android OS and free software we need for our practical work
- Creating an Android app and analyzing its structure
- Java programming language fundamentals
- Java and OOP
- Java concurrency utilities
- Starting Another Activity in Java Android App
- Adding Firebase to Android app
- Google ARCore Software Development Kit for Android Apps
- Intro to Android Game Development
- Intro to Android CameraX with Java



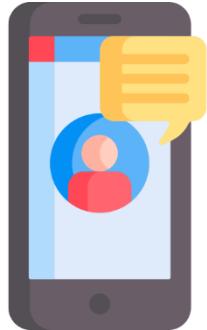
Other ideas ? :)

References



- Course materials in Moodle and Ms Teams
- Abdul-Rahman Mawlood-Yunis. Android for Java Programmers. Springer Publishing, 2022
- Jeff Friesen. Learn Java for Android Development. 3rd Ed., Apress Publishing, 2014
- Cay S. Horstmann. Big Java. 4th Ed., John Wiley & Sons Publishing, 2010

What is Mobile Application?



- Software application designed to run on mobile device
- It needs a mobile OS to run the app
- It is developed to run on small devices such as smartphones, tablets
- Mobile Apps can be either Native apps or Web based apps or can be both (Hybrid apps)

Syllabus



- Syllabus
- Course project:
 - work in groups of up to 3 students (pls explain why you need 3 people)
 - discussion of the project progress every Friday(+/-)
 - pls provide the topic of the course project next week
- Tutorials
 - work in groups of up to 3 students
 - report: Android project files + written document OR 5-minute video
- Attendance: all time-slots, self-check on Moodle first 5 minutes

Advantages of Web Apps

- Universal access
 - Browsers are everywhere
 - Any device on the network can access content: PCs, Macs, Linux, Android, iPhone, Blackberry, etc.
- Automatic “updates”
 - Content comes from server, so it is never out of date
- Well-established tools and methodologies
 - In multiple languages: Java, PHP, .NET, Ruby/Rails, CGI, etc.



Disadvantages of Web Apps



- Few and weak GUI controls
 - Textfield, text area, button, checkbox, radio, list box, combo box, and some other tools from toolbox
 - No direct drawing (except for HTML5 Canvas)
- Inefficient communication
 - HTTP is a weak protocol
- Hard to write
 - Requires knowledge of many technologies: Java, HTML, HTTP, CSS, JavaScript, XML
- Designed for large displays with mouse
 - So harder to use on small phone displays with touch screen
- Limited access to local resources of smartphone

Advantages of Mobile Apps

- Many GUI controls
 - Textfield, text area, button, checkbox, radio, list box, combo box, clock, calendar, date picker, dialog box, image gallery, etc.
- Comparable to options in desktop programming
 - Supports direct drawing
- Animated games like Angry Birds possible
- Can interact with local resources
 - Can read files (e.g., contacts list), have local database, access GPS, initiate phone calls, get input from microphone, create voice output, read screen orientation, etc.



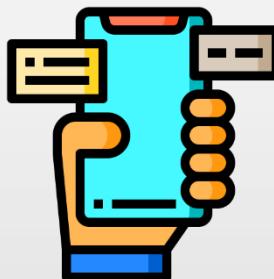
Advantages of Mobile Apps (cont.)

- Efficient communication
 - Can use any networking protocols we want
- Easier to write
 - Requires knowledge of one language only: Java (Kotlin) for Android and Objective-C (Swift) for iPhone
- Designed for small displays with touch screen
 - So, many apps and GUI controls are optimized for this environment

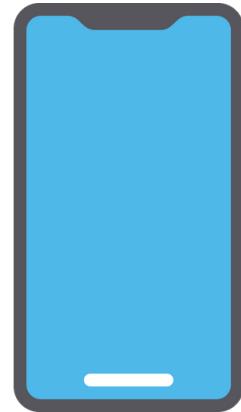


Disadvantages of Mobile Apps

- No universal access
 - Apps must be installed one at a time on each phone
 - An Android app cannot run on iPhone, Blackberry, PC, Mac, or Linux
- Difficult to manage updates
 - User must intervene to get latest versions



What is Mobile OS



- Mobile operating system (mobile OS) is the operating system that controls a smart phone, tablet, PDA, or other mobile device
- Modern mobile operating systems combine the features of a personal computer operating system with
 - Touch screen
 - WiFi
 - Camera
 - Speech recognition
 - Music player
 - Personal digital assistant (PDA)
 - Bluetooth
 - GPS mobile navigation
 - Voice recorder
 - Near field communication
 - Other features...

Comparison of mobile OSs, Aug 31, 2022

https://en.wikipedia.org/wiki/Comparison_of_mobile_operating_systems

Feature	Android	iOS	Tizen	Sailfish OS	Ubuntu Touch	Mobian ^[1]	Plasma Mobile	PureOS	PostmarketOS	KaiOS	HarmonyOS
Developed by	Google, Open Handset Alliance	Apple Inc.	Linux Foundation, Tizen Association, Samsung, Intel	Sailfish Alliance, Mer, Jolla and Sailfish community contributors	UBports and Ubuntu community contributors (previously Canonical Ltd.)	Debian on Mobile Team	KDE and Blue Systems	Purism	PostmarketOS community	KaiOS Technologies Inc.(TCL)	Huawei
Market share ^{[2][3]}	71.90%	27.33%	0.22%	N/A	N/A	N/A	N/A	N/A	N/A	0.14%	4%
License	Base system is Free and open-source. Closed-source drivers are often needed for hardware support.	Proprietary, open source kernel and core	Partial; both proprietary and open-source components, assorted licenses	Free and open-source, but the UI and the SDK are proprietary and closed source	Free and open-source, mainly the GPL ^[4]	Free and open-source, mainly the GPL	Free and open-source, mainly the GPL	Free and open-source, mainly the GPL	Free and open-source, GPL	Proprietary except for open source kernel patches (formerly the MPL 2.0 B2G OS)	Proprietary except for open-source components
Current version	12L	15.6.1	4.0.0.7	4.3.0.15	16.04 OTA-22	20210516	5.24.3	10.0	21.12 Service Pack 3	2.6.0	3.0.0.76
Development version	13	16.0	6.0 M2	Unknown	Unknown	20220612	Unknown	Unknown	N/A	N/A	3.0.0
Current version release dates	March 7, 2022; 5 months ago	August 17, 2022; 11 days ago	May 30, 2020; 2 years ago ^[5]	February 16, 2022; 6 months ago ^[6]	February 18, 2022; 6 months ago	May 16, 2021	March 8, 2022; 5 months ago	October 2, 2021; 10 months ago	March 13, 2022; 5 months ago		July 27, 2022; 32 days ago

Comparison of mobile OSs, Aug 31, 2022 (cont.)



Feature	Android	iOS	Tizen	Sailfish OS	Ubuntu Touch	Mobian ^[1]	Plasma Mobile	PureOS	PostmarketOS	KaiOS	HarmonyOS
OS family	Modified Linux kernel based	Darwin	Linux (based on a combination of Linux MeeGo and Samsung Bada)	Linux	Linux (based on Ubuntu)	Linux (based on Debian)	Linux (mainly based on KDE neon)	Linux (based on Debian)	Linux (based on Alpine Linux)	Firefox OS / Open Web (based on Linux kernel)	Unix-like (based on Linux kernel multi-kernel layer)
Supported CPU architecture	ARM (32-bit ARMv7-A and 64-bit ARMv8-A only), x86, x86-64 ^[7]	64-bit ARMv8-A only	ARM, x86, x86-64	ARM, x86-64	ARM, x86-64	64-bit ARM, RISC-V, x86, x86-64 and LoongArch	ARM	ARM, ?	Convergence operating system	ARM	64-bit ARM, x86, x86-64
Programmed in	C, C++, Java, Kotlin	C, C++, Objective-C, Swift	C++, Xamarin.Forms (.NET C#, F#, VB ^[8])	C++, QML, Python	Apps: HTML5, QML, Go, JavaScript, C++ System: C, C++, QML	C, C++	C++, QML		Python install tool and shell script packages	HTML5, JavaScript	C, C++, Java (until HMOS 3.0), JavaScript, eTS and Cangjie ^[9]

Comparison of mobile OSs, Sept 2, 2021

https://en.wikipedia.org/wiki/Comparison_of_mobile_operating_systems

Feature	Android	iOS	Tizen	Sailfish OS	Ubuntu Touch	Plasma Mobile	PureOS	PostmarketOS	KaiOS
Developed by	Google, Open Handset Alliance	Apple Inc.	Linux Foundation, Tizen Association, Samsung, Intel	Sailfish Alliance, Mer, Jolla and Sailfish community contributors	UBports and Ubuntu community contributors (previously Canonical Ltd.)	KDE and Blue Systems	Purism	PostmarketOS community	KaiOS Technologies Inc. (TCL)
Market share ^{[1][2]}	71.90%	27.33%	0.22%	N/A	N/A	N/A	N/A	N/A	0.14%
License	Base system is Free and open-source, but usually bundled with proprietary apps and drivers, which provide an increasing amount of the functionality. ^[3]	Proprietary, open source kernel and core	Partial; both proprietary and open-source components, assorted licenses	Free and open-source, but the UI and the SDK are proprietary and closed source	Free and open-source, mainly the GPL ^[4]	Free and open-source, mainly the GPL	Free and open-source, mainly the GPL	Free and open-source, GPL	Proprietary except for open source kernel patches (formerly the MPL 2.0 B2G OS)
Current version	11	14.6	4.0.0.7	4.1.0.24	16.04 OTA-18	5.16.4	9.0	alpha	2.6.0
Development version	12	15.0	5.5 M1	Unknown	Unknown	Unknown	Unknown	N/A	
Current version release dates	September 8, 2020; 11 months ago	May 3, 2021; 4 months ago ^[5]	May 30, 2020; 15 months ago ^[6]	May 27, 2021; 3 months ago ^[6]	July 14, 2021; 51 days ago ^[7]	July 30, 2019; 2 years ago ^[8]	26 September 2019; 23 months ago ^[9]		

Comparison of mobile OSs, Sept 2, 2021 (cont.)



Feature	Android	iOS	Tizen	Sailfish OS	Ubuntu Touch	Plasma Mobile	PureOS	PostmarketOS	KaiOS
OS family	Modified Linux kernel based	Darwin	Linux (based on Combination of Linux MeeGo and Samsung Bada)	Linux	Linux (based on Ubuntu)	Linux (mainly based on KDE neon)	Linux (based on Debian)	Linux (based on Alpine Linux)	Firefox OS / Open Web (based on Linux kernel)
Supported CPU architecture	ARM (32-bit ARMv7-A and 64-bit ARMv8-A only), x86, x86-64 ^[10]	64-bit ARMv8-A only	ARM, x86, x86-64	ARM, x86-64	ARM, x86-64	ARM, ?	Convergence operating system	ARM, ARM64, x86, x86-64	ARM
Programmed in	C, C++, Java, Kotlin	C, C++, Objective-C, Swift	C++, Xamarin.Forms (.NET C#, F#, VB ^[11])	C++, QML, Python	Apps: HTML5, QML, Go, JavaScript, C++ System: C, C++, QML	C++, QML		Python install tool and shell script packages	HTML5, JavaScript

Comparison of mobile OSs, Sept 2, 2020



Feature	Android	iOS	Tizen	Sailfish OS	Ubuntu Touch	Plasma Mobile	PureOS	PostmarketOS	KaiOS
Developed by	Google, Open Handset Alliance	Apple Inc.	Linux Foundation, Tizen Association, Samsung, Intel	Sailfish Alliance, Mer, Jolla and Sailfish community contributors	UBports and Ubuntu community contributors (previously Canonical Ltd.)	KDE and Blue Systems	Purism	PostmarketOS community	KaiOS Technologies Inc., Google ^{[1][2]}
Market share ^[3]	86.1%	13.7%	Unknown	Unknown	Unknown	Unknown	Unknown	N/A	?
Current version	10	13.6	4.0.0.7	3.3.0.16	16.04 OTA-11	5.16.4	9.0	alpha	2.6.0
Development version	11	14.0	5.5 M1	Unknown	Unknown	Unknown	Unknown	N/A	
Current version release dates	September 3, 2019; 11 months ago	July 15, 2020; 48 days ago	November 16, 2018; 21 months ago	April 28, 2020; 4 months ago ^[4]	May 13, 2020; 3 months ago ^[5]	July 30, 2019; 13 months ago ^[6]	26 September 2019; 11 months ago ^[7]		
License	Base system is Free and open-source, but usually bundled with proprietary apps and drivers, which provide an increasing amount of the functionality. ^[8]	Proprietary	Partial; both proprietary and open-source components, assorted licenses	Free and open-source, but the UI and the SDK are proprietary and closed source	Free and open-source, mainly the GPL ^[9]	Free and open-source, mainly the GPL	Free and open-source, mainly the GPL	Free and open-source, GPL	Proprietary except for open source kernel patches (formerly the MPL 2.0 B2G OS)

Comparison of mobile OSs, Sept 2, 2020 (cont.)



Feature	Android	iOS	Tizen	Sailfish OS	Ubuntu Touch	Plasma Mobile	PureOS	PostmarketOS	KaiOS
OS family	Modified Linux kernel based	Darwin	Linux (based on Combination of Linux MeeGo and Samsung Bada)	Linux	Linux (based on Ubuntu)	Linux	Linux (based on Debian)	Linux (based on Alpine Linux)	Unix-based
Supported CPU architecture	ARM (32-bit ARMv7-A and 64-bit ARMv8-A only), x86, x86-64 ^[10]	64-bit ARMv8-A only	ARM, x86, x86-64	ARM, x86-64	ARM, x86-64	ARM, ?	Convergence operating system	ARM, ARM64, x86, x86-64	ARM
Programmed in	C, C++, Java, Kotlin	C, C++, Objective-C, Swift	C++, Xamarin.Forms (.NET C#, F#, VB ^[11])	C++, QML, Python	Apps: HTML5, QML, Go, JavaScript, C++ System: C, C++, QML	C++, QML		Python install tool and shell script packages	



Mobile OS Challenges

- Limited resources
 - Memory - CPU
 - Power - Screen resolution
- Connectivity constraints
 - Bad network connectivity
 - Roaming scenarios :(
- Usability constraints
 - User interfaces
 - User input /Data output
- Security
- Scalability
- Wide verity of devices

- Different constraints of mobile operator
 - Power management
 - Data safety in case of power loss is an issue
 - Compact
 - Not much room for UI and the application itself
 - Reuse of common components important
 - Reliability
 - “Always on”



Difference in Mobile and Desktop based applications



- **Screen size**

- The screen size of mobile devices these days typically ranges between 4.5-6.5 inches as compared to that of a desktop which is normally 15-17 inches
- Due to this dramatic change in the screen sizes the mobile web application design is quite challenging

- **Graphics**

- Mobile graphical display varies from the desktop

- **Layout of websites**

- The layout of the mobile app is mostly vertical in contrast to the horizontal layout of desktop websites
- Buttons and bars are mostly in use for mobile apps since it's quite easy to tap and access through fingers on mobiles
- So, the design of website must be user friendly and in accordance with the size and color scheme of the mobile screen to avoid its distortion

Difference in Mobile and Desktop based applications (cont.)



- Navigational limitation

- There are two types of navigations possible on websites – global navigation and contextual navigation
- Navigation on the desktop is quite complex and multilayered in contrast to the flat structure of mobile devices
- While a desktop makes use of both types of navigations, mobile is limited to just global navigation with almost total absence of contextual navigation
- Global navigation remains consistent across a site while contextual navigation shows the changes depending on where a user is on a site
- So, absence of contextual navigation makes the user unaware of its whereabouts on the site
- Thus, the navigation on mobile sites is kept simple and uncomplicated
- The content on mobile sites must be placed such that user need not dig deep to get its work done

Difference in Mobile and Desktop based applications (cont.)



- Different Expectations of users

- Cell phones are basically communicating devices which nowadays are equipped with more enhanced features like camera, music player, games, etc.
- As compared to the desktop apps, the demands and expectations of a mobile user might be different and expect that too much of resources are not used since other mobile apps must work properly too
- **One unique feature of mobile devices includes GPS services**
- Cell phones make use of GPS to pinpoint locations, get to know their current location through local searches
- GPS feature of personalized search is totally missing on a desktop
- Nowadays even mobile websites are increasingly using GPS feature to render various kinds of services to the mobile users
- User expects to make use of GPS in order to find out nearest available services like navigation

Design Considerations of mobile apps



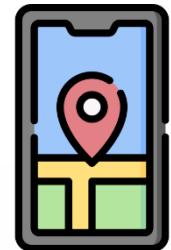
- Content Prioritization

- In comparison to desktop websites, which usually contain a wide range of content and information, mobile sites usually include only the most crucial and time- and location-specific functions and features

- Vertical Instead of Horizontal Navigation

- On desktop websites, horizontal navigation at the top of a page is a widely accepted way of structuring and presenting a site's content. However, vertical navigation dominates on the mobile sites

Design Considerations of mobile apps (cont.)



- Text and Graphics

- On desktop websites, designers use graphics for many different purposes, including promoting, marketing, and navigating. Mobile sites avoid using promotional and marketing graphics and use minimal graphics for navigation.

- Contextual and Global Navigation

- Global navigation is consistent across a site. Contextual navigation depends on where users are on a site. Most mobile sites have global navigation, contextual navigation is rare on mobile sites.

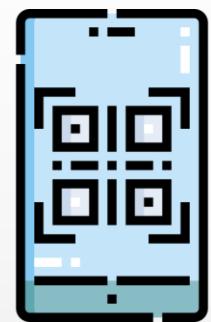
Design Considerations of mobile apps (cont.)



- Integration with Phone Functions
 - Mobile sites offer better integration with phone functions and present marketing opportunities such as facilitating direct orders by phone or sending promotional text messages.
- Localized & Personalized Search
 - Mobile sites can take advantage of technology that automatically detects where users are to present local search results. When users set up their preferences or profile, personalized search results become more relevant to them.

Mobile Development Phases

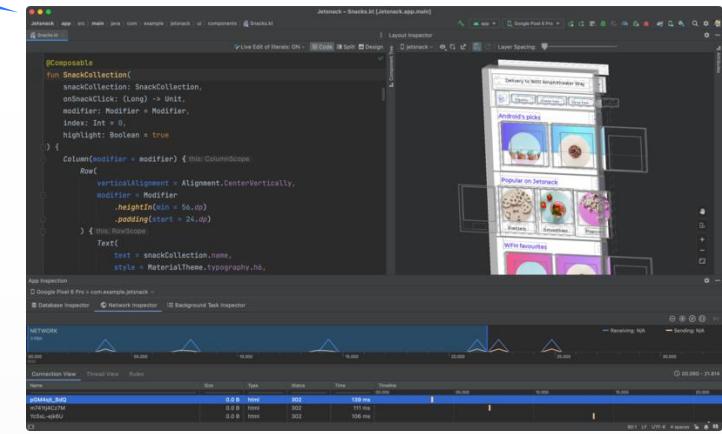
- Market analysis
- ↳ Gathering info for client requirements
- ↳ Analyzing and planning
- ↳ Concept sketching
- ↳ Research & Strategies
- ↳ User Interface Design (front-end)
- ↳ Coding (back-end)
- ↳ Testing



Is that classical approach for the software development?

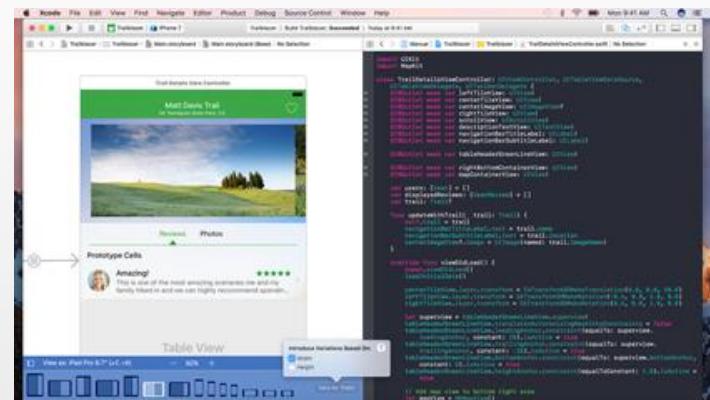
What are IDEs for the development of mobile apps? (as of Aug 31, 2022)

- Android: Android Studio



(May 9, 2022: stable release 2021.2.1 (Chipmunk))

- iOS: Xcode

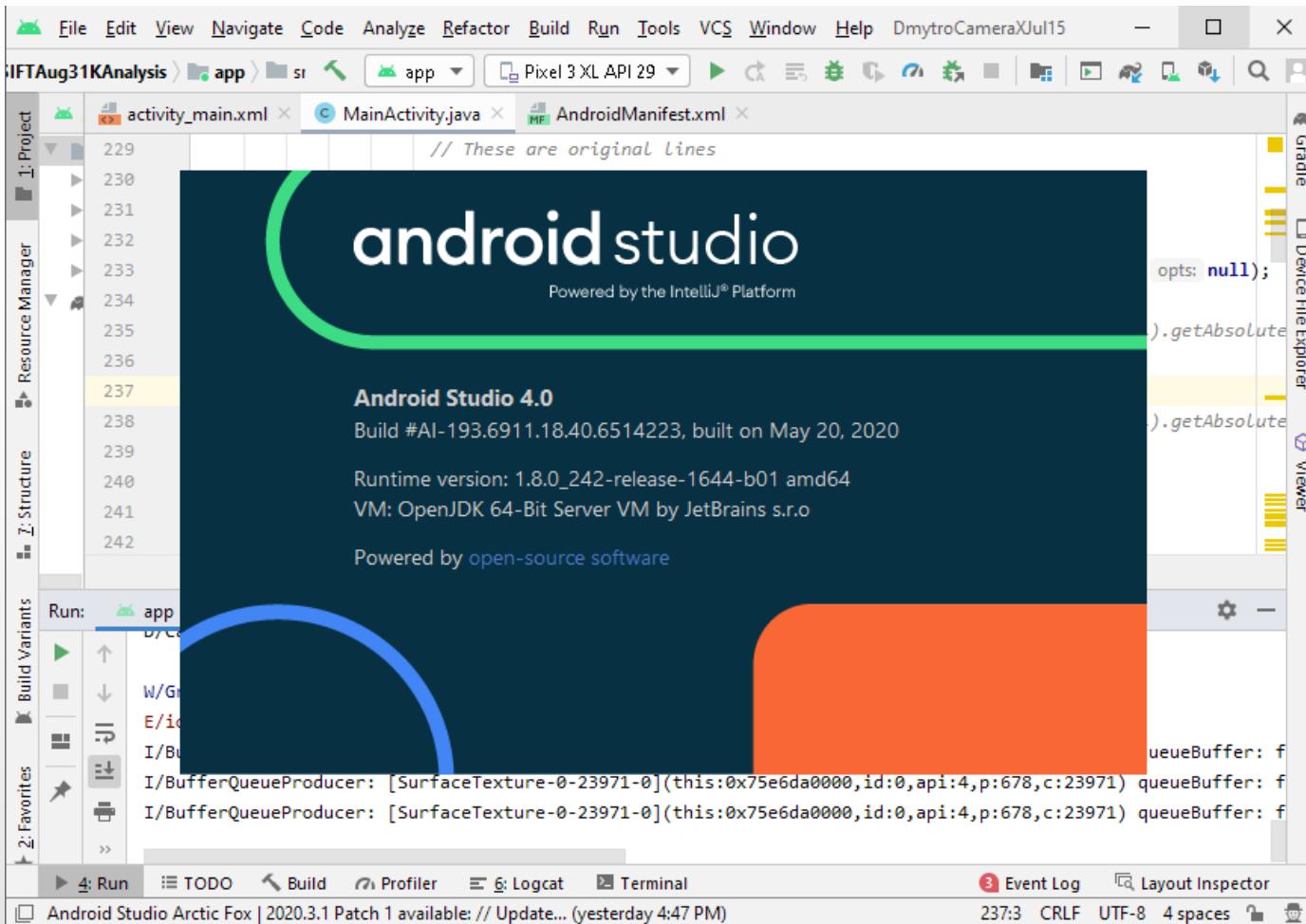


(June 2, 2022: stable release 13.4.1)

What version of Android Studio we gonna use in our course?

<https://developer.android.com/studio/archive>

- Android: Android Studio 4.0



Why? 😊
Why not? 😊

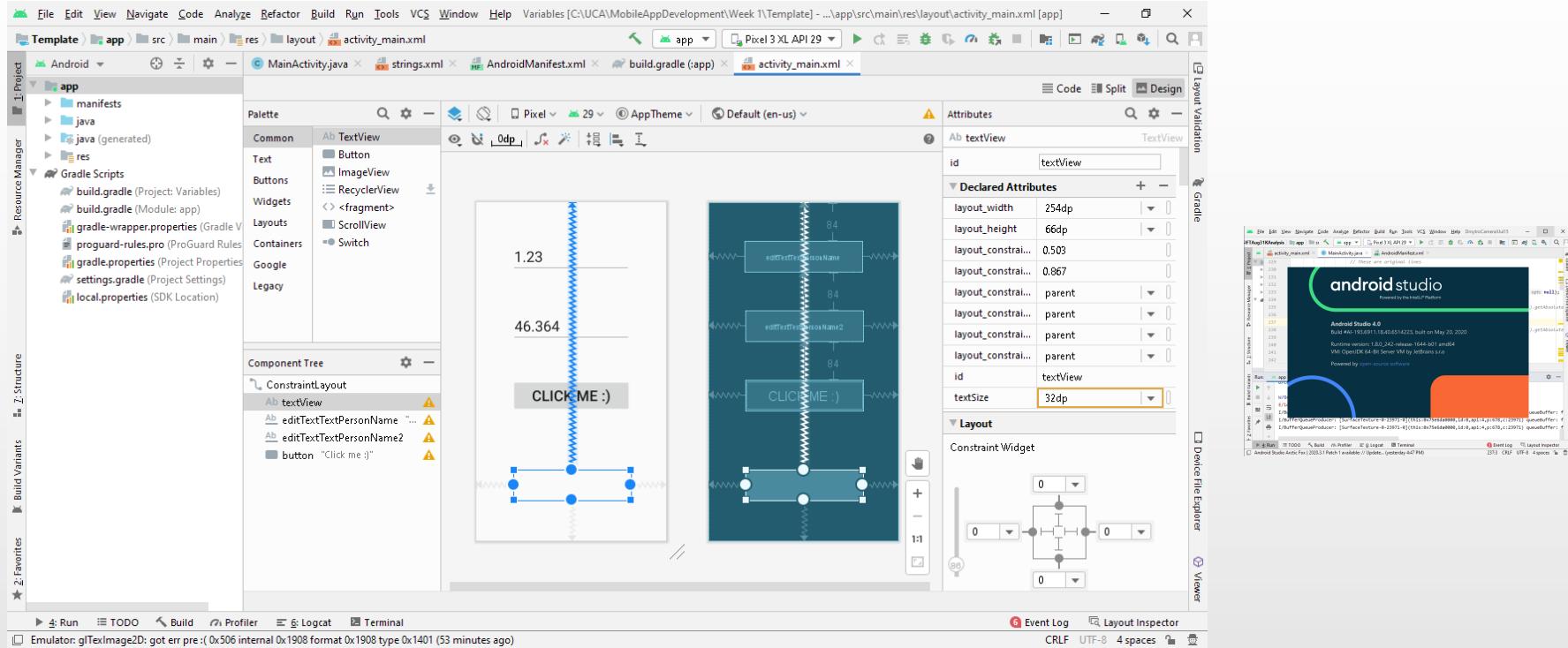
We will work
with Google
ARCore Software
Development Kit
for Android Apps
that works quite
well with
Android Studio
4.0

What version of Android Studio we gonna use in our course?

<https://developer.android.com/studio/archive>

- Android: Android Studio 4.0

- A problem can be starting a new project in Android Studio 4.0 ==> A template project is on the Moodle



What are IDEs for the development of mobile apps? (cont.)

Cross-platform software:

- **Xamarin** is a Microsoft-owned San Francisco-based software company founded in May 2011 by the engineers that created Mono, Xamarin.Android (formerly Mono for Android) and Xamarin.iOS (formerly MonoTouch), which are cross-platform implementations of the Common Language Infrastructure (CLI) and Common Language Specifications (often called Microsoft .NET).

- With a C#-shared codebase, developers can use Xamarin tools to write native Android, iOS, and Windows apps with native user interfaces and share code across multiple platforms, including Windows, macOS, and Linux.



What are IDEs for the development of mobile apps? (cont.)

Cross-platform software:

- **Apache Cordova** enables software programmers to build hybrid web applications for mobile devices using CSS3, HTML5, and JavaScript, instead of relying on platform-specific APIs like those in Android or iOS.

Feature	Android ^[41]	Apple iPhone /iPhone 3G	Apple iPhone 3GS and newer	Bada	BlackBerry 10 and PlayBook OS	BlackBerry OS 4.6-4.7	Firefox OS	Symbian	Tizen	webOS	Ubuntu Touch	Windows Phone
Accelerometer	Yes	Yes	Yes	Yes	Yes	N/A	Yes	Yes	Yes	Yes	Yes	Yes
Camera	Yes	Yes	Yes	Yes	Yes	N/A	Yes	Yes	Yes	Yes	Yes	Yes
Compass	Yes	N/A	Yes	Yes	Yes	N/A	N/A	Yes	N/A	Yes	Yes	Yes
Contacts	Yes	Yes	Yes	Yes	Yes	N/A	Yes	Yes	Yes	N/A	N/A	Yes
File	Yes	Yes	Yes	N/A	Yes	N/A	Yes	N/A	N/A	Yes	N/A	Yes
Geolocation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Media	Yes	Yes	Yes	N/A	Yes	N/A	N/A	N/A	Yes	N/A	Yes	Yes
Network	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Notification (alert, sound, vibration)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Storage	Yes	Yes	Yes	N/A	Yes	N/A	Yes	Yes	Yes	Yes	Yes	Yes



HTML5 offline web apps



- At its simplest, web app is a list of URLs to HTML, CSS, JavaScript, images, or any other kind of resource.
- The home page of the offline web app points to this list, called a manifest file, which is just a text file located elsewhere on the web server.
- A web browser that implements HTML5 offline applications reads the list of URLs from the manifest file, download the resources, cache them locally, and automatically keep the local copies up to date as they change.

HTML5 offline web apps (cont.)



- The Cache Manifest file is a simple text file that lists the resources the browser should cache for offline access.
- To enable the application cache for an app, it is necessary to include the manifest attribute on the document's html tag:

```
<html manifest="example.appcache">  
  ...  
</html>
```

HTML5 offline web apps (cont.)



- A manifest has three distinct sections: CACHE, NETWORK, and Fallback.

- **CACHE:** This is the default section for entries. Files listed under this header (or immediately after the CACHE MANIFEST) will be explicitly cached after their download first time.
- **NETWORK:** Files listed in this section may come from the network if they aren't in the cache, otherwise the network isn't used, even if the user is online. We can white-list specific URLs here, or simply "*", which allows all URLs. Most sites need "*".
- **Fallback:** An optional section specifying fallback pages if a resource is inaccessible. The first URI is the resource, the second is the fallback used if the network request fails or errors. Both URIs must be from the same origin as the manifest file. We can capture specific URLs but also URL prefixes. "images/large/" will capture failures from URLs such as "images/large/whatever/img.jpg".

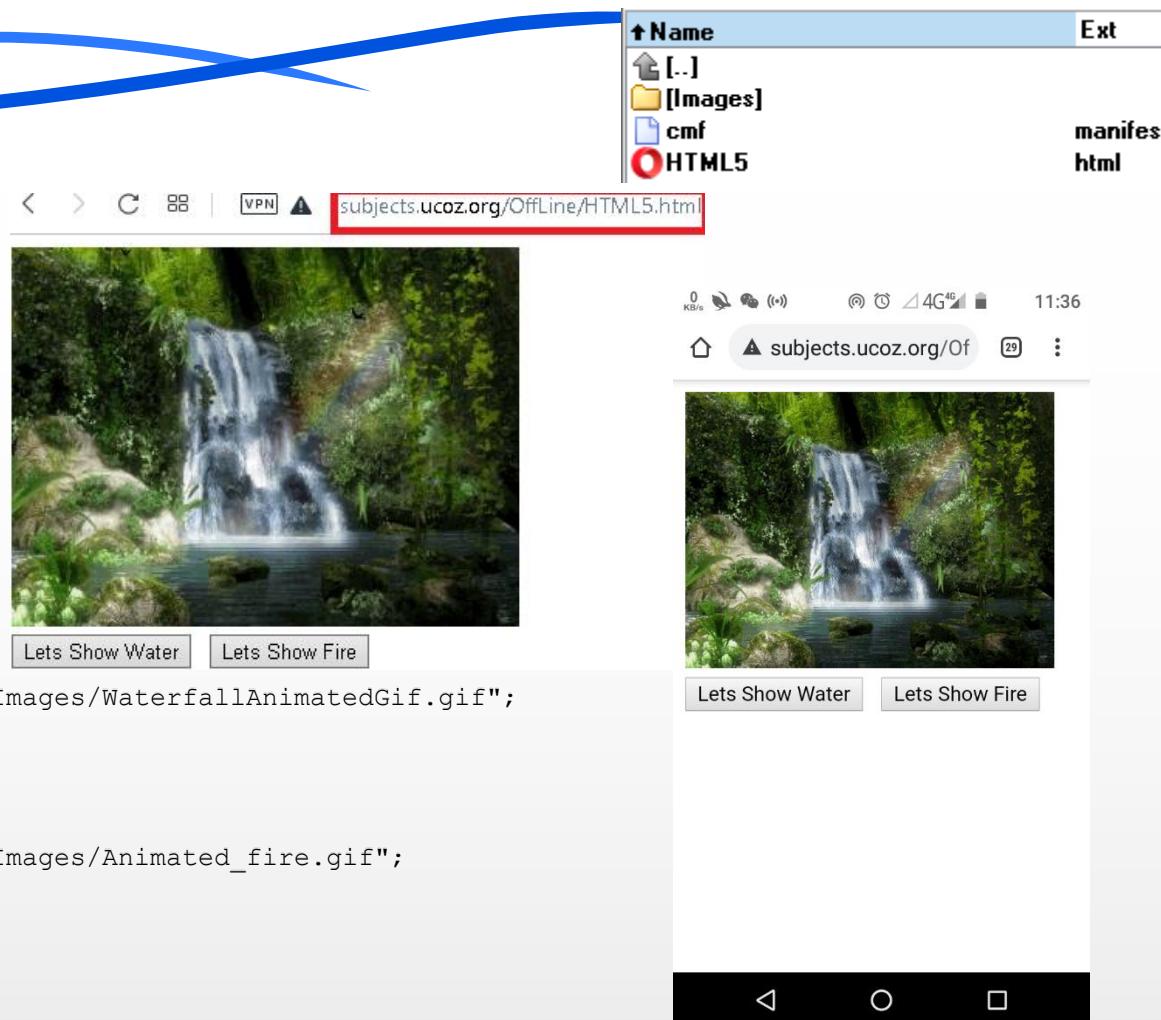
HTML5 offline web apps: An Example

● cmf.manifest

```
CACHE MANIFEST
# Resources to cache - it is a comment
CACHE:
HTML5.html
Images/Animated_fire.gif
Images/WaterfallAnimatedGif.gif
```

● HTML5.html

```
<!DOCTYPE html>
<html manifest="cmf.manifest">
<head>
  <title>Offline Web Application</title>
  <script>
    function changeImage()
    {
      document.images["jsbutton"].src= "Images/WaterfallAnimatedGif.gif";
      return true;
    }
    function changeImageMore()
    {
      document.images["jsbutton"].src= "Images/Animated_fire.gif";
      return true;
    }
  </script>
</head>
<body>
  
  <br><input type=button onclick="changeImage()" value="Lets Show Water"></input>
  &nbsp;&nbsp;<input type=button onclick="changeImageMore()" value="Lets Show Fire"></input>
</body>
</html>
```



Homework



- Installation of Android Studio with virtual devices:
 - Installation of Android Studio 4.0
(<https://developer.android.com/studio/archive>)
(<https://redirector.gvt1.com/edgedl/android/studio/install/4.0.0.16/android-studio-ide-193.6514223-windows.exe>)
 - Installation of Pixel 3 XL or another virtual device(s)

Please start the any app on the virtual device

Do you have any
questions or
comments?





Thank you
for your attention !

In this presentation:

- Some icons were downloaded from flaticon.com and iconscount.com
- Some info was taken from Siva Kumar Reddy Vasipally presentation “Introduction Mobile Application Development”