Development Aspects in Mobile Computing

CSCI3310

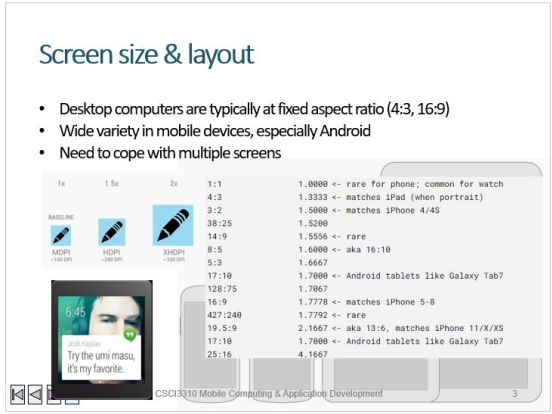
Mobile Computing & Application Development

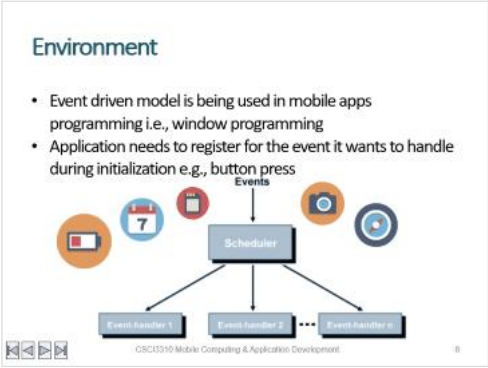
	Desktop PC	Mobile devices
Computational power	Much better	Limited
User	Multi-users	Strictly personalized
Usage	Fixed location	Anywhere
Energy Awareness	Practically no	Severe

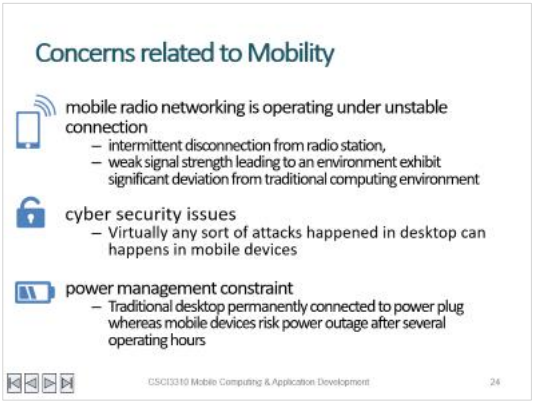
Issues in Development

The following issues need be addressed in development as compared with traditional desktop app

- Heterogenous **Screen-size** & layout
- **User Input** technology – gesture, voice, various sensors available
- Resource-scare Execution **Environment**
- Wireless **Connectivity** – WiFi, Cellular, Bluetooth, NFC etc
- Heterogenous **Development Platforms**

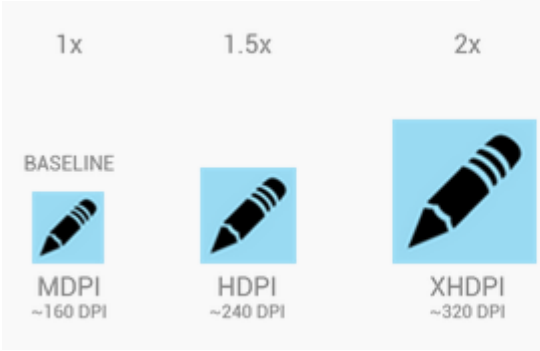


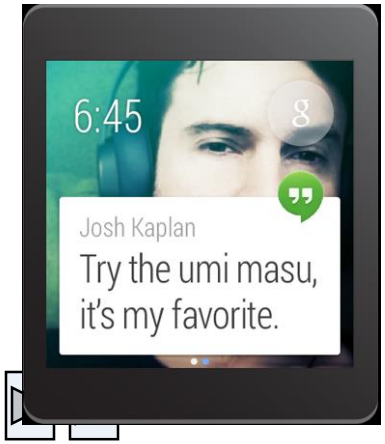




Screen size & layout

- Desktop computers are typically at fixed aspect ratio (4:3, 16:9)
- Wide variety in mobile devices, especially Android
- Need to cope with multiple screens






```
1:1          1.0000 <- rare for phone; common for watch
4:3          1.3333 <- matches iPad (when portrait)
3:2          1.5000 <- matches iPhone 4/4S
38:25        1.5200
14:9         1.5556 <- rare
8:5          1.6000 <- aka 16:10
5:3          1.6667
17:10        1.7000 <- Android tablets like Galaxy Tab7
128:75       1.7067
16:9         1.7778 <- matches iPhone 5-8
427:240      1.7792 <- rare
19.5:9       2.1667 <- aka 13:6, matches iPhone 11/X/XS
17:10        1.7000 <- Android tablets like Galaxy Tab7
25:16        4.1667
```

Screen size & layout

- Mobile device screen size is also small (3 – 10 in) when compared with desktop (19-24 in)

- Implications

Amount of data displayed is **significantly lesser**

Only "**right**" data should be displayed

User also need to **know current position** in the whole data navigation process

User Input

Finger Gesture is being pioneering used in mobile apps to replace keyboard

More advanced tactile end user input mechanisms proved so popular that they being **retrofitted** into traditional desktop

Camera & **sensors** input are also unique input in app

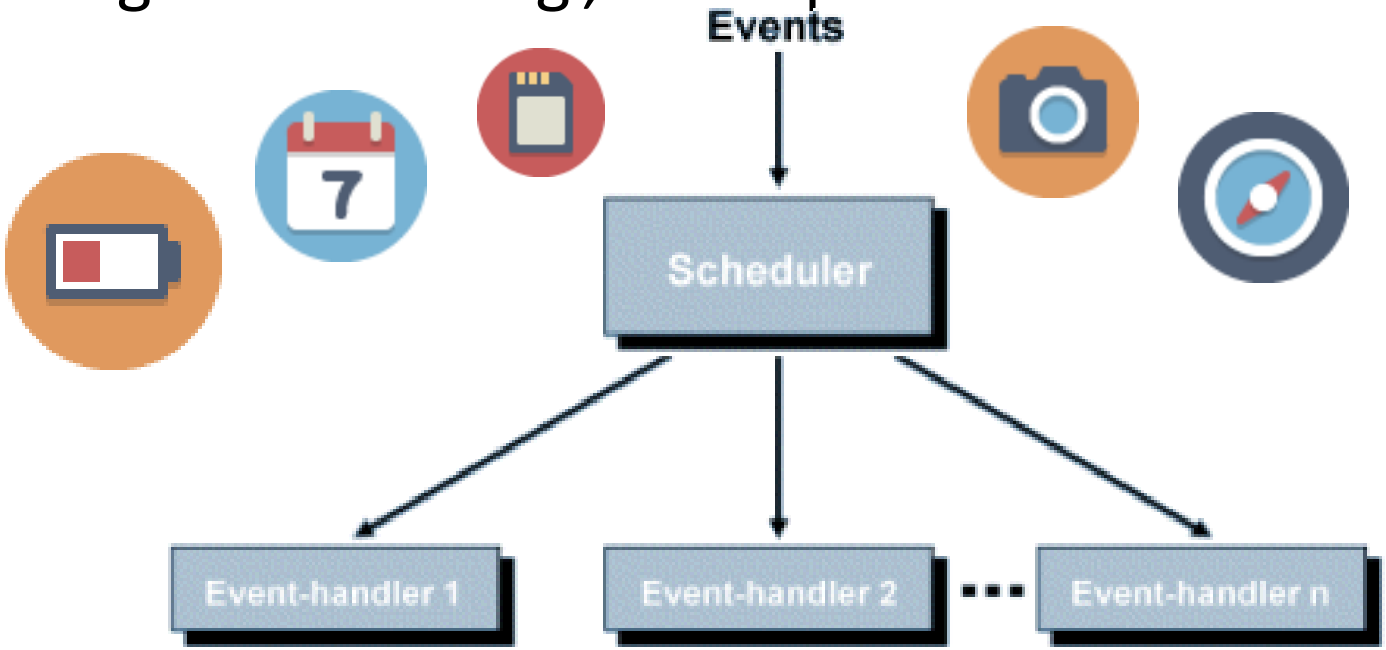
Sensors

New generation of built-in sensors broaden further the application domain e.g., VR/AR, eHealth etc.

- **Camera**, **accelerometer, magnetometer, gyroscope** all brought more applications possibilities
- With inclusion of **bio-sensors**, even tighter coupling of mobile devices with our daily lives expected

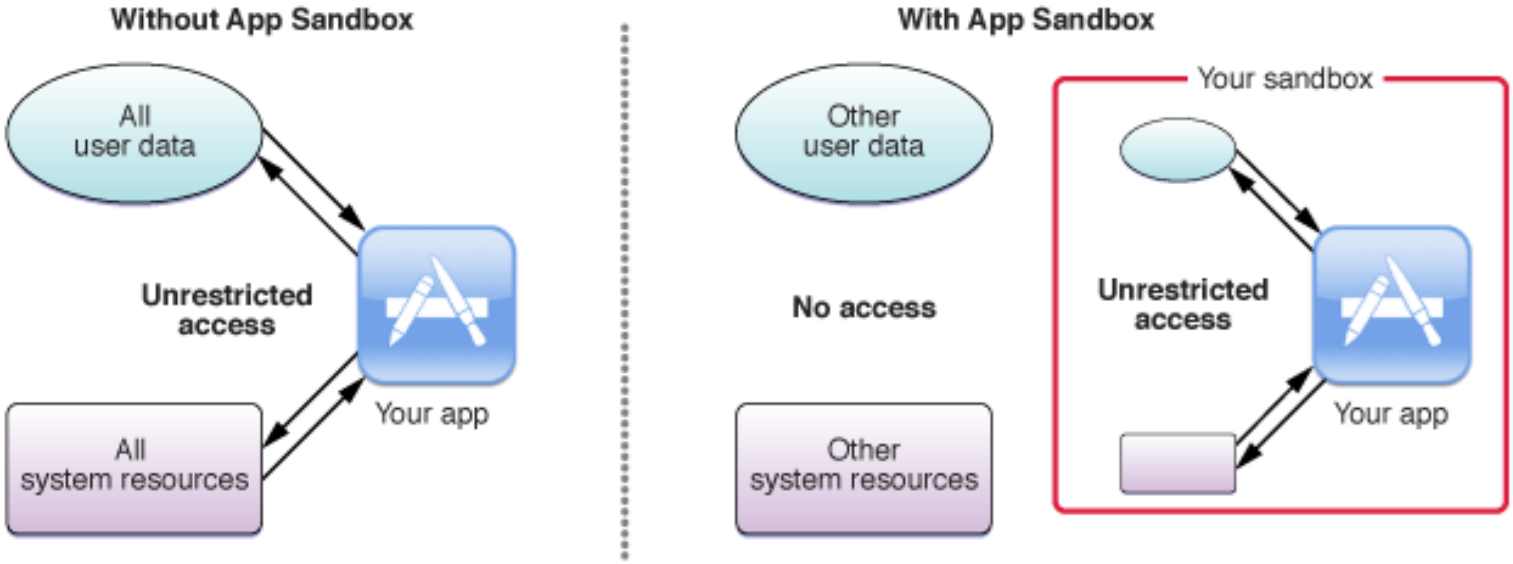
Environment

- Event driven model is being used in mobile apps programming i.e., window programming
- Application needs to register for the event it wants to handle during initialization e.g., button press



Environment

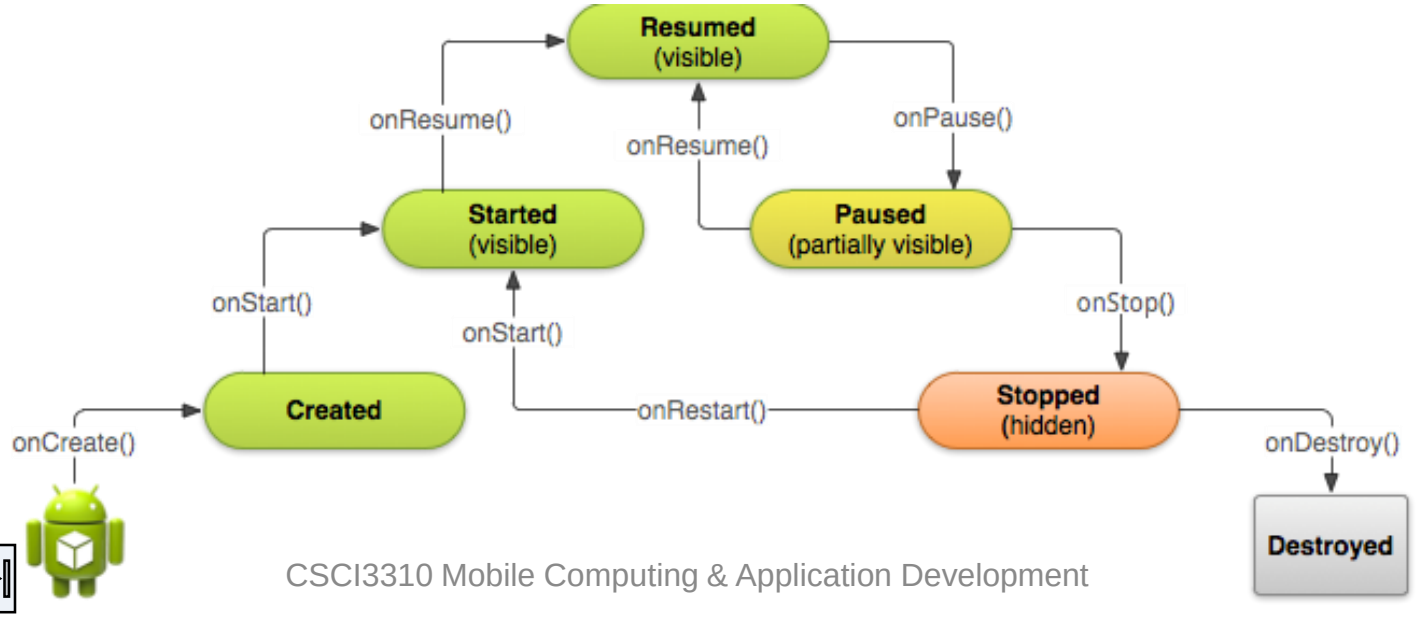
- To ensure maximum security, the environment an app is running is in general considered as a **sandbox**



- Each app can only access assets within designated locations i.e., limited access to files, memory, network resources, etc
  - Need to describe how the app interact with the system (access **permissions**)

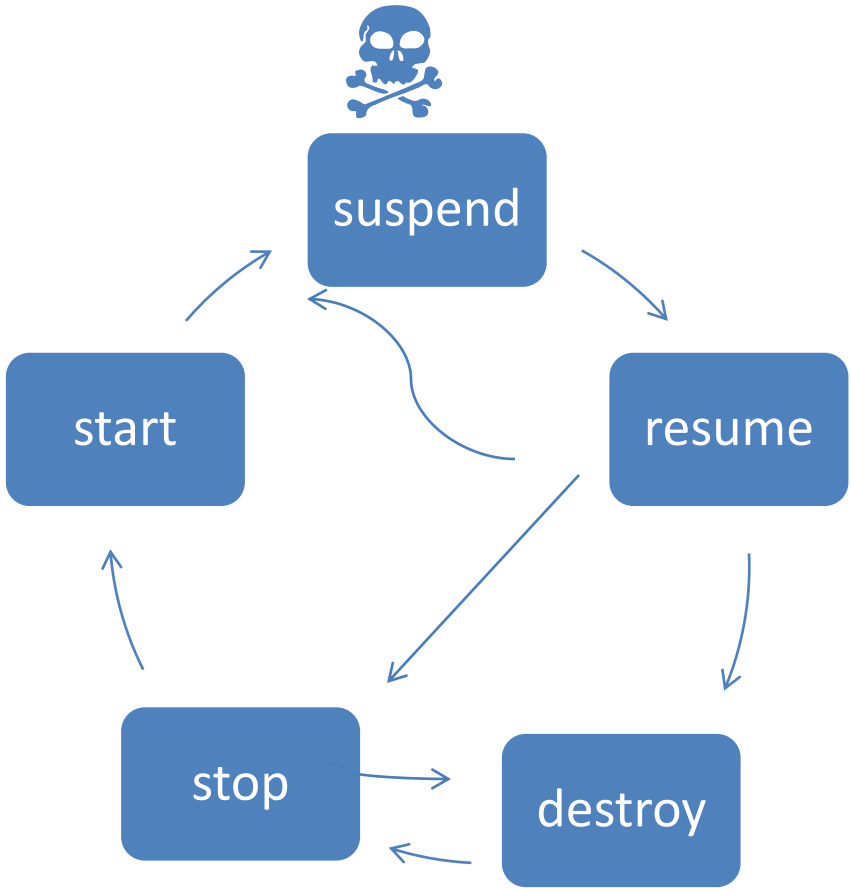
Environment

- From time to time, your app may be suspended by various event e.g., incoming phone call, user pressing home button etc.
- Applications need ways to **save the states** of the app, or doing cleanup
- **Interruption** may occur due to incoming call, network disconnect etc.
- Application suspend/resume thus need be specified also



Development Consideration

- During the lifetime, an app will transit between various stages according to operating condition
- Programmer need to **associate callbacks** to different transitions so that
  - **Does not lose progress** if leave and return to app later
  - **avoid crash** when user receives a call or switch to another app
  - **Won't consume system resources** when not active



Concerns related to Mobility

mobile radio networking is operating under unstable connection

- intermittent disconnection from radio station,
- weak signal strength leading to an environment exhibit significant deviation from traditional computing environment

cyber security issues

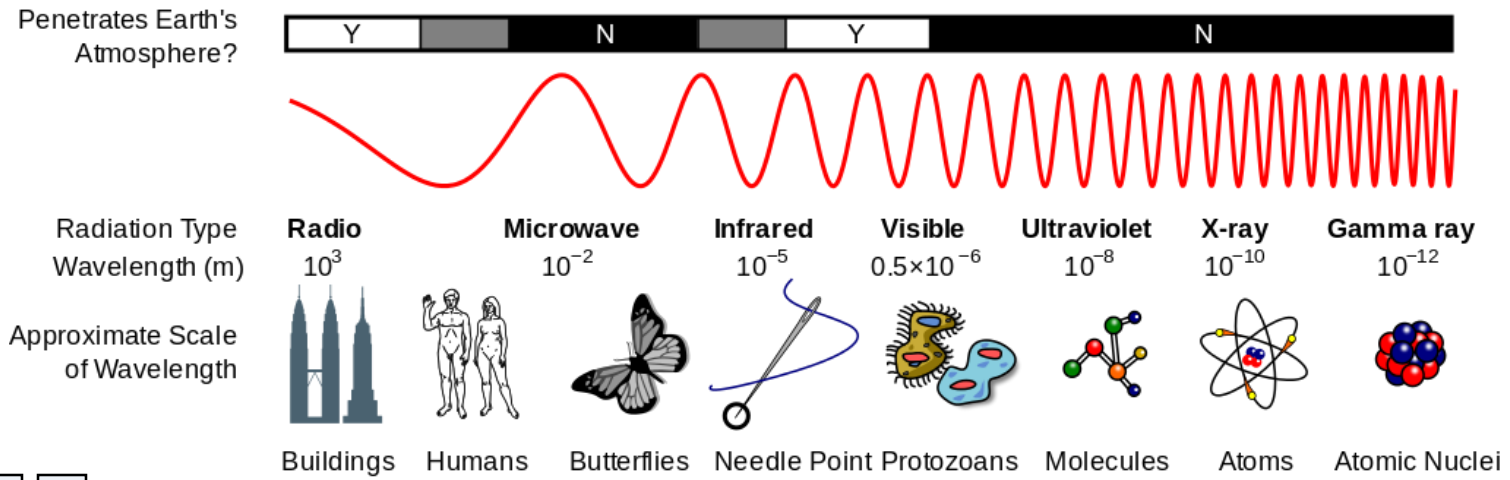
- Virtually any sort of attacks happened in desktop can happens in mobile devices

power management constraint

- Traditional desktop permanently connected to power plug whereas mobile devices risk power outage after several operating hours

Connectivity

- **GPS** for positioning is already a standard
- Standard network connectivity is provided for mid range wireless communication protocols such as **WiFi**
- Cellular connection through **3G**, **4G** and **5G** later
- Also short range **Bluetooth**, **NFC**



Secure Coding in Mobile Apps

How apps developer be prepared to tackle the topmost Mobile security threats?

Jailbreaking

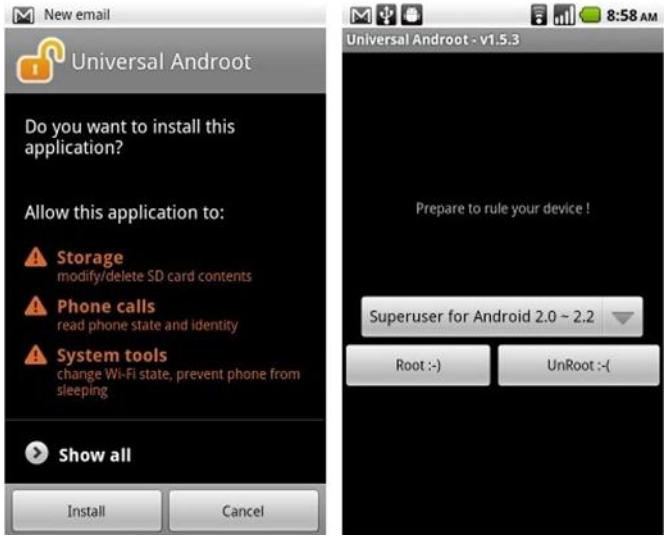
- Process of **removing limitations imposed by Apple** on devices running iOS
- Allows user gaining root access

- Thus downloading apps & other utility through channels other than App Store or iTune

- Someone viewed it as expressing opposition to Apple's censorship over apps distribution

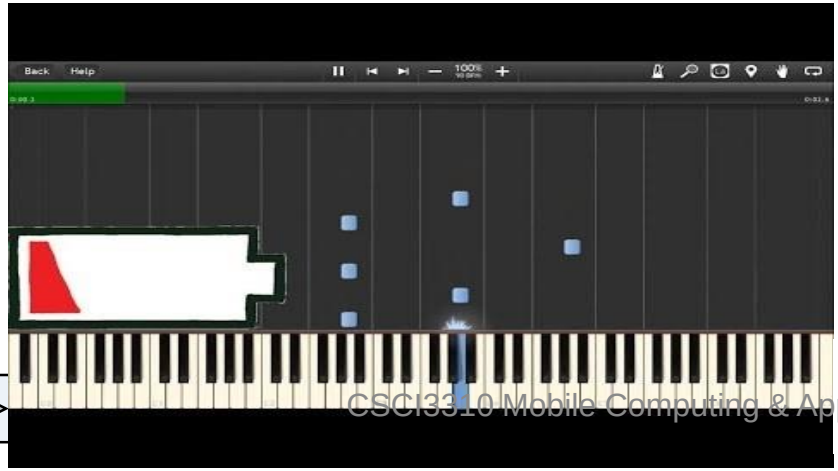
Android Root

- Root in Android gives you full control over the system
- Overclocking, altering of system files, manipulating sensors are possible
- Void warranty of the device also



Reducing Power Consumption

- Energy saving
  - Current smartphone design can only operate in order of 10 hours
  - For particular apps e.g. **location based app**, energy draining is more severe
  - **Network connection** will **drain out power** faster
  - Issues to better design the app working so as to preserve more energy and improve user experience



MOBILE PHONE LOW BATTERY SOUNDS IN SYNTHESIA

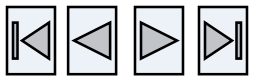
12,115,762 views

Sześcian

Published on Jan 21, 2018

Development Smartphone Apps

- Consists of 2 or 3 (practically 2) major groups at this moment
  - iOS
  - Android
  - Others such as Windows Mobile, etc



Developing platforms

Web Apps

- Load within a browser
- Universal and flexible, but slower and limited
- Language: JavaScript, HTML5, CSS, etc.

Hybrid Apps

- Shared code-base in web or other languages and tailored native code for different devices
- E.g., Wrap with Apache Cordova or PhoneGap framework
- The best-of-both-worlds or the botch?

Native Apps

- Provide the best user experience
- Have access to low-level device functionalities
- Language: Objective-C/Swift for iOS; Java/Kotlin for Android

Web Platform

- Smart phones or tablet typically come with web browser pre-installed
- Support for **JavaScript** & **HTML5** is fairly consistent

- Mobile app can be built as a standard web application plus special **style sheets (CSS)** to accommodate mobile form factor
- Disadvantage
  - No access to features on mobile devices such as camera, contact list etc.

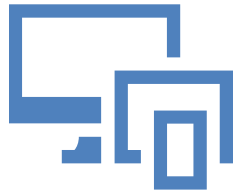
App Development

- There existed some web sites/software that can generate an app for an ordinary user
- Limited to simple photo/video playing
- For serious innovations or applications, you still must know programming in order to translate your idea into real apps

iGenapps : app generation services

MIT APP INVENTOR

Cross-platform SDK



A software Development Kit (SDK) which simplify most of the low-level tasks in writing an application

programmers need **not bother about the platform** specific details to program a smartphone

Abstracting technical details into a general computer

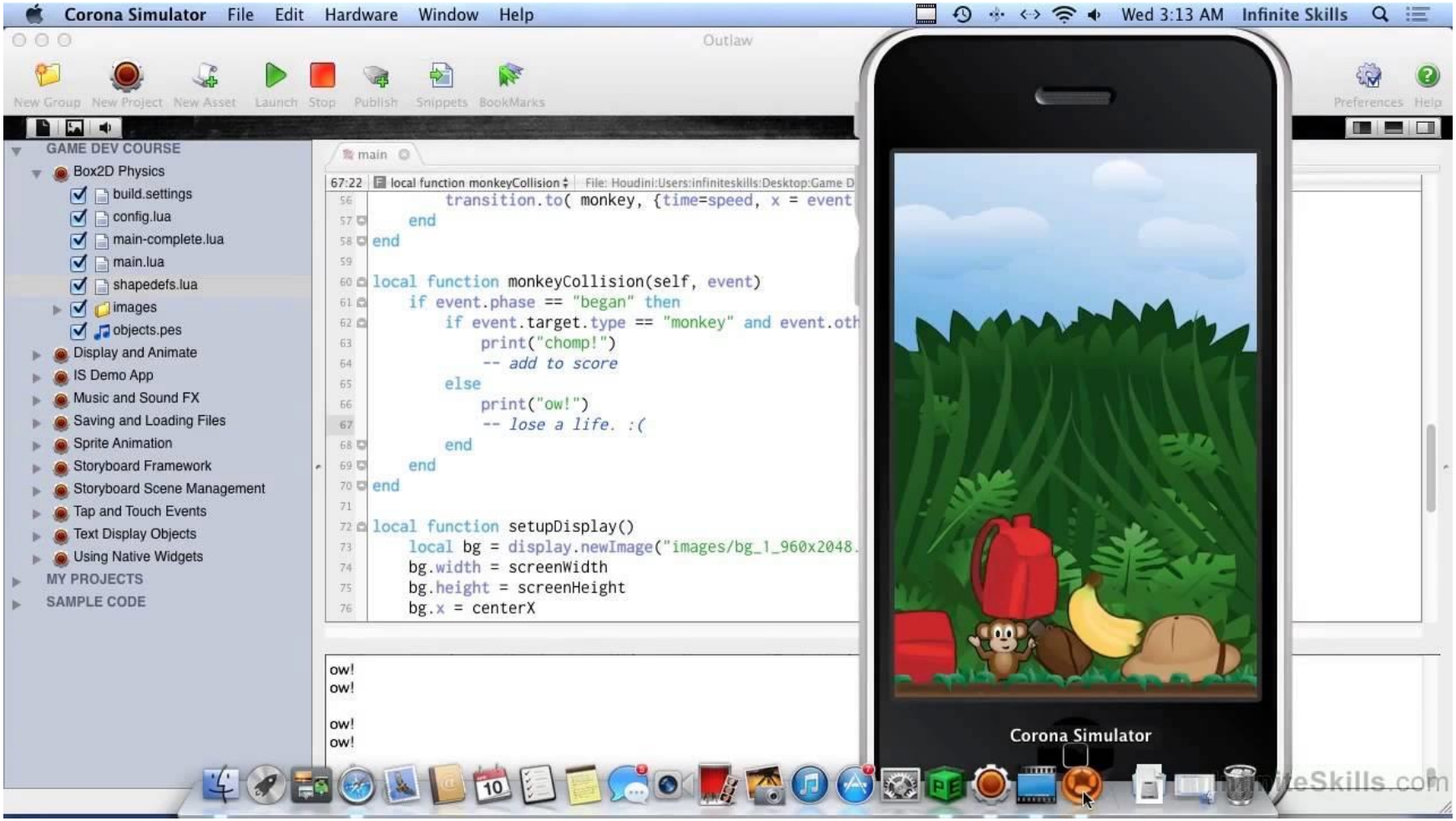
Advantages

Save costs for maintaining different vendor versions of your app

Can **focus on more high level features** rather than digging in reference manual for fine technical details, thus speed up development

more efficient when developing **gaming** apps across different platforms

Advantages


Corona Simulator File Edit Hardware Window Help
Wed 3:13 AM Infinite Skills
Outlaw
New Group New Project New Asset Launch Stop Publish Snippets BookMarks
Preferences Help
GAME DEV COURSE
Box2D Physics
build.settings
config.lua
main-complete.lua
main.lua
shapedefs.lua
images
objects.pes
Display and Animate
IS Demo App
Music and Sound FX
Saving and Loading Files
Sprite Animation
Storyboard Framework
Storyboard Scene Management
Tap and Touch Events
Text Display Objects
Using Native Widgets
MY PROJECTS
SAMPLE CODE
main
67:22 local function monkeyCollision
transition.to(monkey, {time=speed, x = event
end
end
local function monkeyCollision(self, event)
if event.phase == "began" then
if event.target.type == "monkey" and event.oth
print("chomp!")
-- add to score
else
print("ow!")
-- lose a life. :(
end
end
end
local function setupDisplay()
local bg = display.newImage("images/bg_1_960x2048.
bg.width = screenWidth
bg.height = screenHeight
bg.x = centerX
ow!
ow!
ow!
ow!
Corona Simulator

Drawback

Usually require **licensing fee** to use, add up to the development cost

Programming environment is limited to the framework provided by the middleware e.g., **no low-level access** to the data structures of original platform

Performance hit – relies on high level operations to reduce code complexity. This results in performance hits when many functions are coded in middleware.

Native Development platforms (iOS)

- Including iPhone/iPad
  - Being most influential one
  - Uniform user experience across the whole platform
  - **Require Mac OS** environment for development
  - Mostly use **XCode** integrated development environment (IDE)
  - Need to learn **Objective C** /**Swift**

Native Development platforms (Android)

- Android
  - Most convenient developing platform as it allows **Windows**, **Mac** or **Linux** environment
  - Google Play used to impose **less vetting** than AppStore, but User base also the largest one
  - Needs to cope with **non-uniform user experience** across the entire platform

Google removes malicious Angry Birds apps from Android Market in Jun 2011. Reason: Playstore doesn't vet its apps enough

References

Comparison of smartphones @ wiki

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

"Google's Android OS: Past, Present, and Future". PhoneArena. August 18, 2011. Retrieved March 12, 2017.

http://www.phonearena.com/news/Googles-Android-OS-Past-Present-and-Future_id21273

Iphone wiki

http://en.wikipedia.org/wiki/IPhone

Monetizing Android App

http://www.androidheadlines.com/2013/07/monetizing-and-marketing-your-android-app-in-a-free-to-play-world.html