EECS 1012: Introduction to Computer Science

November 28, 2016

Advanced material

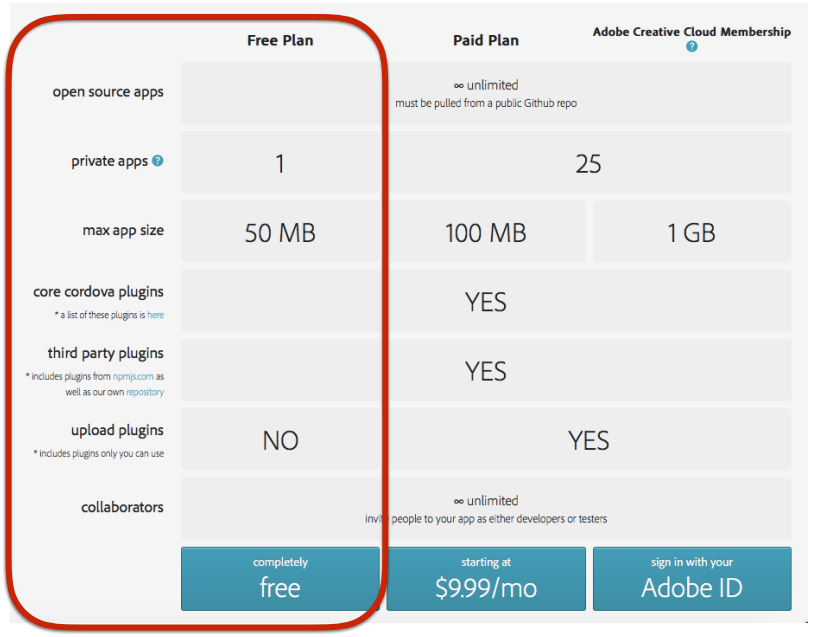
NOT ON LAST TEST

From html2apk to an app

- There are many ways of building a stand-alone application for Android/IOS/other platforms based on html/js/css.
- HTML2APK does most of this, generically
  - It was designed to have very low overhead
- There exist commercial applications that let you do this too, both using the cloud and your own hardware to do the heavy lifting.

Phonegap build

- An adobe-based technology for doing this
- It basically does what you did with html2apk, but rather than having a generic application for any code you write, it packages your code with the application code in html2apk into a unique application.



	Free Plan	Paid Plan	Adobe Creative Cloud Membership
open source apps		∞ unlimited must be pulled from a public Github repo	
private apps	1	25	
max app size	50 MB	100 MB	1 GB
core cordova plugins * a list of these plugins is here		YES	
third party plugins * includes plugins from npmjs.com as well as our own repository		YES	
upload plugins * includes plugins only you can use	NO	YES	
collaborators		∞ unlimited invite people to your app as either developers or testers	
	completely free	starting at $9.99/mo	sign in with your Adobe ID

So before starting

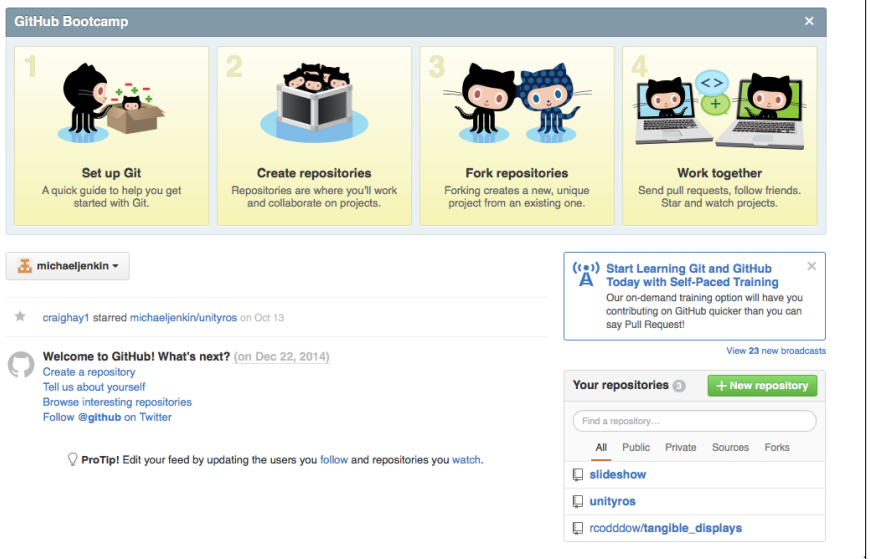
- You need
  - A debugged app
  - A github account
  - An Android (or IOS or …) device. We will concentrate on the Android device
  - A QR code reader on the device
    - Many of these out there

A debugged app

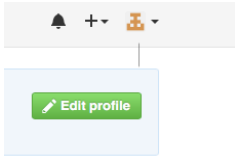
- Conway's game of life from last day
  - Code is on the course web site

Github

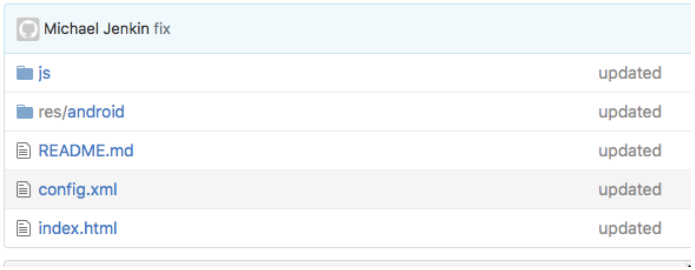
- A free external repository for maintaining a code base
  - Your code is viewable by the public unless you pay money for a private repository.
- Basic concept: you have a local copy on your machine and a master copy on the server.
- You can push versions from your local copy to the server and pull versions from the server to your machine.



So…



- Create an account on github
- Create a repository within your github account
  - You can have more than one repository
- I have already created a ‘game of life’ repository



Link code on your machine to the repository

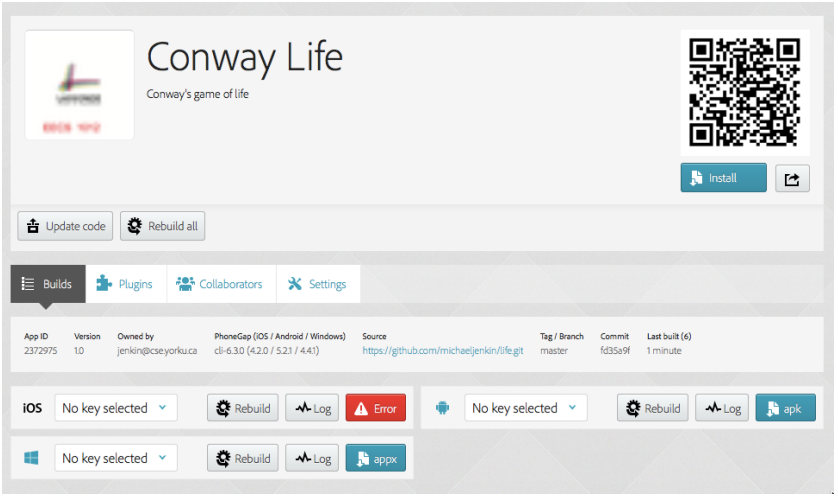
- Basic goal here is to connect your git repository to the current code on your virtual box

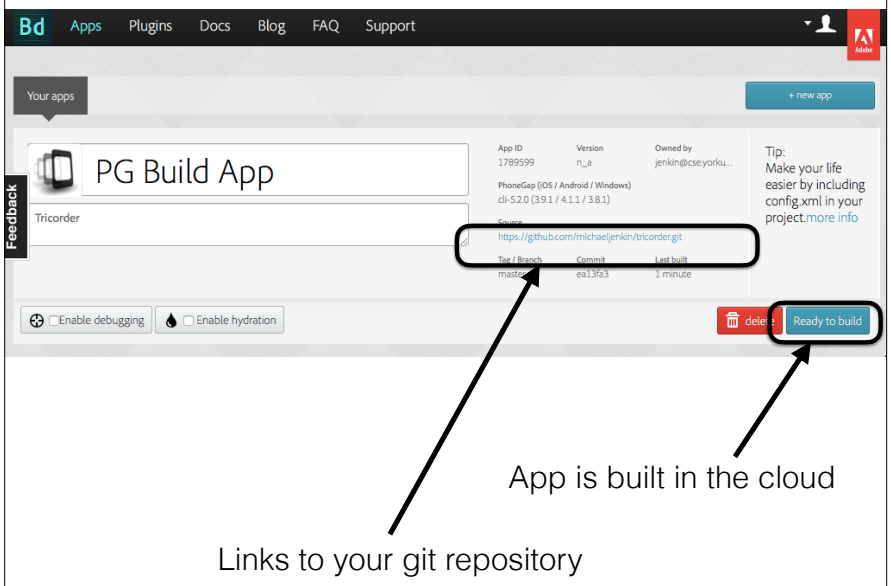
```
git clone https://github.com/michaeljenkin/life.git

this will copy all the files from the repository to you

make changes

get add *
get commit –m “really important updates”
git push origin master
```

(Git also does version control, and other things. We will ignore that here.)

phonegap

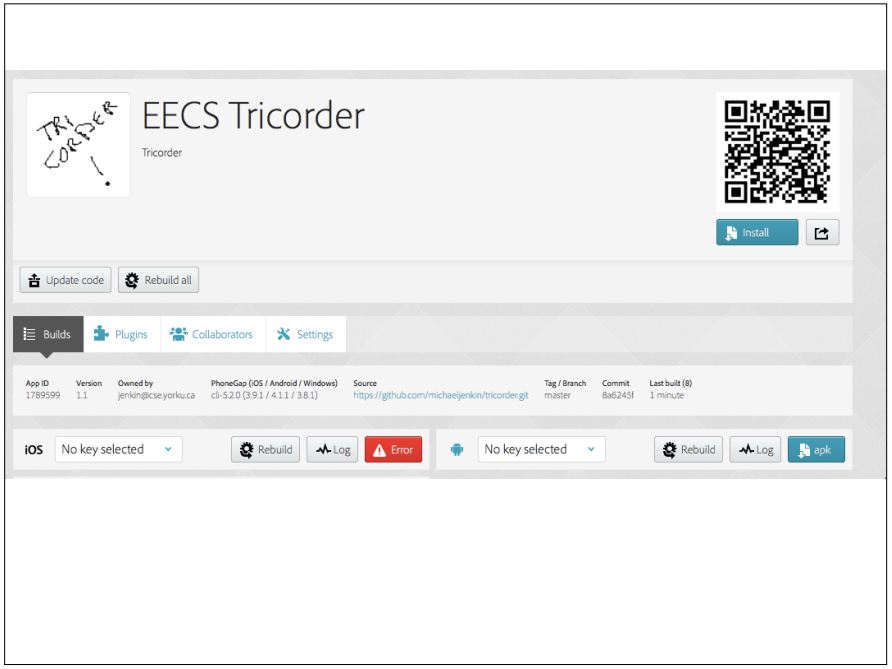
- Does what the html2apk application does in the class, but uses github as the repository
  - Actually makes a unique app for each application
- You will require an build.phonegap.com (adobe) account (also free)





Build for Android

- Download using QR code
- Note - android must accept 3rd party applications
- Note - can easily build for IOS & other platforms
  - For IOS require a provisioning code
  - Commercial versions for Android requires you to sign the application



config.xml

- Provides a way of configuring your application.
- Again, a great example of using a standard representation in your work.

```xml
<?xml version = "1.0" encoding = "UTF-8"?>

<widget xmlns = "http://www.w3.org/ns/widgets"
   xmlns:gap = "http://phonegap.com/ns/1.0" id = "ca.yorku.eecs" version = "1.0">

   <name>Conway Life</name>

   <description>
      Conway's game of life
   </description>

   <author href = "http://www.cse.yorku.ca/~jenkin" email = "jenkin@cse.yorku.ca">
      Michael Jenkin
   </author>

   <preference name = "permissions" value = "none"/>

   <icon src = "res/android/mipmap-ldpi/ic_launcher.png" gap:platform = "android" gap:qualifier = "ldpi" />

   <icon src = "res/android/mipmap-mdpi/ic_launcher.png" gap:platform = "android" gap:qualifier = "mdpi" />

   <icon src = "res/android/mipmap-hdpi/ic_launcher.png" gap:platform = "android" gap:qualifier = "hdpi" />

   <icon src = "res/android/mipmap-xhdpi/ic_launcher.png" gap:platform = "android" gap:qualifier = "xhdpi" />

   <icon src = "res/android/mipmap-xxhdpi/ic_launcher.png" gap:platform = "android" gap:qualifier = "xxhdpi" />

</widget>
```

config.xml

- config.xml file allows you to give the application a name, change its icon, etc.
- It is an xml file (again, an example of using a standard technology elsewhere

```
<preference name="orientation" value="landscape" />
```

Other details…

- Can add splash screens, fix orientation, etc.
- Has added methods to enable access to hardware from JavaScript
  - Uses cordova (standard library) to do this

Icons

- Different screens require different icons in different sizes.
- This is the reality that a phone is not a watch is not a tablet.
- Many ways of creating these, exist standard tools that will do this for you from one standard image

www.makeappicon.com

Yet more details

- To build for some platforms (e.g., IOS) it is necessary to have a key from Apple
- You can build commercial applications with this tool
  - Although that costs a bit
- That being said, you can have a single code base that runs on a range of different platforms

Summary

- In this course you used a proprietary tool (html2apk) to provision JavaScript/HTML/CSS-based applications on an Android device.
- Exists a range of commercial alternatives that you can use to write dedicated apps for hand-held devices.