**App Inventor at City Lit – Cribsheet Part 1**

**Getting started at City Lit**

1. Get a notebook from the trolley unless you have brought your own.
2. Switch it on and Logon to the City Lit notebook with your student ID (5-6 numeric digits) and (password) your date of birth in the format ddmmmyyyy. e.g 09jan1970 (month name first three letters and all in lower-case.) If you have forgotten your student ID ask me. If you have forgotten your date of birth ask a relative! If you are doing this correctly and it doesn’t work ask me. This login should always work at City Lit so recording you student ID for your next course is a good idea.
3. Once the notebook is booted (City Lit or personal) you will need to connect to the hidden wi-fi network set up for this course. Click the wifi icon on the screen and select **Unknown Network.** When prompted the network name is **kitkat** and the password is **20142014CDM01.** Check this is connected before moving on. You need this working to connect to App Inventor in the cloud and your Android device (if you have one)
4. Start up a web browser and go to App Inventor in the cloud - **ai2.appinventor.mit.edu.**

Check for any useful news and then click the **Create apps** button (top right). If you are not logged in to a Google account you will be asked for your Google Username and password.

You should now be ready to go with App Inventor where you left it last time you were there.

If you do this on your own network (at home) with your own computer this should work without steps 1,2 and 3) My apologies for the stuff you need to do at City Lit. This is for the protection of our student and networks and is beyond my control.

**Testing your Apps**

You are developing on a PC using Windows (unless you’ve turned up with a Mac or Linux computer) but Android code needs to run on an Android Device (usually a phone or tablet that runs Android OS.) You need to get your app to your device. There are several ways to do this but you may need to install additional code on your computer or android device to make this work.

1. **Connect your Android device to your computer using a USB cable**. ***(Connect /USB)-*** This sounds simple, you possibly do this with your camera, but is difficult to do as your computer needs to load/install a *device driver* (piece of code) that allows it to communicate with your Android device. **You cannot do this at City Lit** as the notebooks are locked down so you can’t install device drivers. Can also be difficult if you buy *no-name* android devices as finding the device driver code on the (usually) Chinese website of the OEM can be a challenge. You might need to go this route at home if you don’t have wi-fi or if the computer you want to use is wired to your internet connection. If you get this working your Android device screen should mirror the designer screen.

1. **Connect by Wi**-**FI Recommended. (Connect/ AI Companion)** Both your computer and your Android device need to be connected to the same WI-FI network. At home this should be simple. At City Lit we need to make sure that both the computer and your Android device are connected to the **KitKat** network as City Lit’s other network will not allow this to happen (for security reasons.)  
     
   To allow your Android device to connect to App Inventor you will need to install **the AI2 Companion app** from Google Play. When you select this option App inventor will display a *QR code* on the screen. Scan this with the AI2 companion app to connect your devices. Your Android device screen should mirror the designer screen.
2. **Run the Android Emulator (Connect/Emulator) –** Runs a virtual (imaginary) Android device on your computer. This is your only option if you don’t have a real Android device but means you can’t test apps that use an android device’s sensors like GPS and orientation. The emulator (a Java program running on your computer) needs to connect to App Inventor in the cloud and need to run a program called AIStarter for this to work. This is installed on the City Lit notebooks but you might need to start it so check the start menu on the PC if App Inventor asks you to start this. If you are using your own computer and want to run the emulator you will need to install AIStarter.  
     
   Notes:

The emulator is a complete, imaginary, Android device and make take 2 mins or so to start as there is lots of code to load. Try not to close the emulator window (minimise it instead) or you’ll need to wait for it to restart again next time.

It should (like the other methods) mirror the contents of your designer screen. If it doesn’t, or there is an emulator error message, go to **Connect/Reset connection** to stop the emulator and then restart it with ***Connect/Emulator.***

All of App inventor, except the Emulator, is in the cloud so MIT can update this whenever they want and you just see the changes when you next use it. As AIstarter is a program on your computer you may get a message that this needs to be updated (as we did last week). If you are lucky enough to select a computer from the trolley that was not used last week you may need to do this for that computer too!

1. The fourth way - If all else fails (and you have an Android device) is to build and Install the app. Select **Build/App (provide QR code for. apk**) and wait for your app to be built. This will show a QR code on your computer which you can then scan with any QR code scanner App (not AiCompanion) that you download from Google play. This will install the app on your device and you will run it like any other app. It will have a default App Inventor icon for now  
     
     
   Notes:  
     
   Your device may tell you that its security settings only allow it to install apps from Google Play. You will need to change this in your device’s settings. The procedure for this varies with different version of Android so Google a solution that works for you.  
     
   You’ll need to delete the app if you get bored with it.

**Creating Apps in App Inventor**

App Inventor has two views; **Design View** and **Blocks View**. Design view is used for selecting and arranging the components that make up your App. Blocks view allow you to link and manipulate blocks of functionality embedded in those components,

The App Inventor ***Designer view*** has four sections:

A tabbed ***Component palette*** containing components you can build you apps with. Components can be visible (used to build the user interface) and invisible (do things but don’t appear on the screen). Use a component by dragging it onto the Viewer window

***Viewer Window*** – used to collect components dragged from the component palette. Show the appearance and location of visible components.

***Components Viewer*** – show the name and relationship between components in your app. Can be used to rename and delete components

**Properties viewer** – show the properties of the currently selected component. You can select a component to edit by clicking on the component in the viewer windows or on the component’s name in the components viewer

**Our usual workflow is to:**

Decide on the components we need – you’ll need to discover where they are and what they can do before you get any good at this.

Drag the first component from the *Component Palette* to the *Viewer* Window.

Give the component a sensible name in the *Component Viewer* – five textboxes called textbox1 … textbox5 are going to cause us delay later in the process

Set the properties of the component, as it will be when the app starts, in the Properties viewer. For example, a label component which displays a piece of text will have a font, font size, colour and the text it displays as some of its properties.

Repeat until you have added and set the starting properties of all your components.

**Finding Components**

Component are found within tabs in the Component Palette. Use components on the **User Interface** and **Layout** tabs to arrange your user interface. These are mainly visual components.

**Media tab** – contains components that allow you to create, find and use sound and video in your apps.

**Drawing and Animation** - Create a *Canvas* on your screen that can be used for animated effects like simple games

**Sensors** – use an Android device’s sensors. Available sensors will vary on the device. Expensive branded Android devices will have all of these. Cheap phones may be missing features like NFC. Can’t test these in the emulator.

**Social** – make/ receive phone calls, send receive texts etc. Use these with text to speech and speech recognizer on the Media tab for autodial/answerphone services etc.

**Storage** – store and retrieve data on your device or in the cloud

**Connectivity** – start another apps on your device, connect to a nearby device using Bluetooth, or use the web to send/receive information.