

**CSE Software Listing 2015-2016**

Both Windows and Mac platforms are supported.

**Firewall:**

An AWS server is being provided by PLTW. Students will access the server using the SSH protocol. School firewalls and proxy servers will need to be configured to allow SSH on port 2222 to pltw.org and on port 22 to picoctf.com.

**Software for Student/Teacher Machines:**

**Scratch™ 2 Offline Editor**

Scratch uses an in-browser interface as of version 2.0. The offline editor will be advantageous if Internet access is spotty at the beginning of semester.

<http://scratch.mit.edu/scratch2download/>

***Python*® Enthought Basic Canopy distribution**

<http://www.enthought.com/products>

Register for the Academic version at <https://www.enthought.com/products/canopy/academic/> to get Canopy Basic.

Once installed, use Canopy's package manager to install additional libraries: basemap, lxml, requests, and scipy.

**Filezilla Client**

<https://filezilla-project.org/>

**Firefox**

<http://www.mozilla.org/en-US/firefox/new/>

**Firebug Firefox Add-on**

<http://getfirebug.com/>

**FireSSH Firefox Add-on**

<https://addons.mozilla.org/en-US/firefox/addon/firessh/>

**Google Chrome™**

<https://www.google.com/intl/en_us/chrome/browser/>

**Java™**

<http://java.com/en/download/index.jsp>

**Flash**

<http://get2.adobe.com/flashplayer/>

**Adobe® Reader®**

<http://get.adobe.com/reader/>

**Adobe® AIR**

<http://get.adobe.com/air/>

**App Inventor AIStarter**

The AI starter provides support for the emulator option for work with MIT App Inventor.

<http://appinventor.mit.edu/explore/ai2/setup-emulator.html>

**Face-L Facile Face Labelling Software**

Mac version 1.0.1; Windows version 1.0.0. A known bug opens an error dialog which can be ignored when the Windows version is launched.

<http://sourceforge.net/projects/pyvision/files/FaceL/>

**Light-Bot-Lite**

<http://www.light-bot.com/light-bot-lite.swf>

(Download by opening game at this URL in Firefox and then selecting **Firefox > Save page** and save as a Shockwave Flash Object.)

**NetLogo**

<http://ccl.northwestern.edu/netlogo/download.shtml>

**GitHub Client**

<http://windows.github.com>or <http://mac.github.com/>

**MEGA**

<http://www.megasoftware.net/>

**Notepad++(Windows) or TextWrangler(Mac)**

[http://notepad-plus-plus.org/ (Windows) or](http://notepad-plus-plus.org/%20(Windows)%20or%20)[http://www.barebones.com/products/textwrangler/download.html (Mac)](http://www.barebones.com/products/textwrangler/download.html%20(Mac))

**PuTTy (Windows) or iTerm2 (Mac)**

These SSH clients are used instead of FireSSH for the picoCTF competition in Project 2.3.4. In the rest of the course, the SSH client FireSSH is used as a SSH client.

<http://www.putty.org/>

**http://iterm2.com**

**HxD (Windows) or HexFiend (Mac)**

A hex editor is used in the advanced portions of the picoCTF competition in Project 2.3.4.

<http://mh-nexus.de/en/hxd/>

<http://ridiculousfish.com/hexfiend/>

**7-zip**

<http://www.7-zip.org>

XaoS

<http://fractalfoundation.org/resources/fractal-software/>

**Software for Android devices:**

**MIT AI2 Companion**

<https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3>

**Color Chooser Stage 1**

Available in Activity 1.2.2 sourceFiles.zip

**Wireless Network Configuration**

A wireless connection between the Android device and the laptop or desktop is strongly recommended. See <http://appinventor.mit.edu/explore/support/explain-wifi-connection.html>. This configuration allows students to test their Android app as they create it on the laptop or desktop machine, where they access App Inventor through a web browser.

The curriculum provides an alternative method for working with App Inventor by using email or a USB storage device configuration for transferring programs to the Android device. This method is much less satisfying for students but requires less IT support to establish. This method, described in the CSE curriculum, is not described on the App Inventor web site and is distinct from and less desirable than the third best option described on App Inventor's website as "live testing with USB cable."

For the preferred method using "live programming by wireless connection," the Android device and the computer must be on the wireless network, in the same subnet, without restrictions on connections between network devices. Wireless access also must be without password authentication, though authentication can be implemented by whitelisting devices. If security is in place preventing this connection over existing WiFi, a wireless router can provide a new access point limited to the classroom set of computers and devices.