Our development platform is called PlayN and can be found here: **http://playn.io**. Go to **http://playn.io** and click around, reading, if you like, the overview at **http://playn.io/docs/overview.html**. Then go to **http://playn.io/docs/setup.html**.

We can build PlayN games using Eclipse (or some other IDE, but then you are on your own); or we can directly use Apache's Java build system called Maven.

First we will write games that run on our desktops (using the Java back-end built into PlayN). Later, we may want to cross-compile our games to run on Android or iOS (or html5, but with the HTML5 back-end there are limitations). It's easiest to work with these other back-ends via Maven.

Simmons is going to use Maven directly at all times. (Simmons usually works on a Chromebook setup to run Arch Linux. But in class he'll mostly use a Windows laptop running Cygwin. See below.)

We basically need Maven and Git to get started. On Linux and Mac OS X one can just install those packages (see the PlayN setup page).

Windows setup using Cygwin and Maven:

1. First let's install Cygwin.
   - Go to **www.cygwin.com** and download the appropriate version (32- or 64-bit) of setup-x86.exe.
   - Run setup-x86.exe and start the installation.
   - When you get to the Select Packages screen:

     Type git into search. Click on Devel and make sure that git is marked to install. If it is marked Skip, click on the double circle arrow to change the Skip so that the installer will install it.

     Do the same with vim; that is, make sure that it will be installed.
   - Now click Next and wait while Cygwin installs.
   - (Later you run setup-x86.exe again and install other software suites, editors, etc. In fact, you should occasionally run setup-x86.exe even if you don't want to install/uninstall anything, so that it can update your Cygwin installation.)
2. Now let's install Maven under Windows and make sure we can see it from Cygwin.
   - Go to **https://maven.apache.org/guides/getting-started/windows-prerequisites.html**.
   - (If you don't have Java installed on your Windows, then follow the link under prerequisites and install the Java SDK. You likely already have this installed.)
   - On the Maven page, click on downloads on the menu on the left and download the zip file named apache-maven-3.3.9-bin.zip. If you wish, you can download the associated .asc file and the public KEYS and use gpg to verify the signature (something that technically only makes sense doing if you've establish *trust* appropriately).
   - Move apache-maven-3.3.9-bin.zip to, say, C:, i.e. your top Windows directory, and then unzip it. (Look in the unzipped directory. Simmons' has his Maven installation set up so that the top directory is: C:apache-maven-3.3.9 which contains directories bin, etc.)
   - Now add this directory to your Windows Path as follows:

     Click Windows Start Button → Control Panel → System.

     Click Advanced System Settings.

     Click Environment Variables.

     Under User variable for ... highlight the Path variable and click Edit.

To the end of the `Variable value` add exactly `;C:\apache-maven-3.3.9\bin`

- Finally, let's check that `Cygwin` can see `Maven`. Start `Cygwin` by clicking on its icon. At the `Cygwin` command line type: `which mvn`. You should see `mvn` sitting in the directory tree we just added.
- If the last step said that `mvn` wasn't found, try rebooting your computer and/or ask Simmons for help.

3. Now let's get the playN basic demo working.
    - At the Cygwin command prompt type
      ```
      git clone https://github.com/playn/playn-samples.git
      ```
    - Now type
      ```
      cd playn-sample/hello
      ```
    - Then type
      ```
      mvn test -Pjava
      ```
    - Wait for stuff to be downloaded.
    - A window should eventually pop up that you can click in.

Notes:

- You have now set up your Cygwin/Maven based development environment.
- Use the Java back-end, at first. See **http://playn.io/docs/setup.html** for more on cross-compilation.