

## Workshop Prerequisites

### Intro to musical programming with Chuck

*Bonnie Eisenman*

Prior to the workshop, attendees should install ChuckK (available for Windows, Mac, and Linux):  
<http://chuck.cs.princeton.edu/release/>

### Coda Lisa: Collaborative art in the browser

*Rico Huijbers*

A laptop and a (modern) browser will be sufficient.

### Breakin: making a breakout clone in F#

*Andrea Magnorsky*

- A windows machine (a VM on OSX has worked for many people). If you need a VM you can [get one for free](#).
- An editor of your choice, if you install Visual Studio 2013 [Community Edition](#) it comes with F# installed :). Alternatively install F# with instructions from [here](#). It is recommended that you install the F# Power Tools extension as well :D
- If you didn't install Visual Studio there are other options [here](#).
- A mouse is a good idea. We have an editor for the game engine where clicking things is necessary.

Once you have that, just clone [JoyOfCoding2015](#) (it is a big repo and will have all the game engine binaries).

If you want to try out your setup and or get familiar with F#, you can try completing the [F# Koans](#).

### Hack your language with Rascal

*Tijs van der Storm & Jouke Stoel*

Prerequisites:

- You have Java 8 JDK installed and setup as default JDK (point the JAVA\_HOME environment variable to your JDK 8 directory).
- You have a Git client installed

Setup:

Download [Eclipse Luna](#)

Select the Eclipse Luna download option for your operating system

### *Give Eclipse some more memory to work with*

This can be done by adding / changing the '-Xmx' property in the eclipse.ini file. Set it to something like 2GB (-Xmx2G) depending on how much your machine can spare. The eclipse.ini file is located in your \$ECLIPSE\_HOME directory. In case of OSX; it can be found under eclipse.app/Contents/MacOS

### *Install Rascal*

- Start Eclipse
- Select Help -> Install New Software
- Make sure that the tick for "Contact all update sites during install to find required software" is enabled.
- Type: <http://update.rascal-mpl.org/unstable/> in the "Work with" edit box and hit enter.
- Select the feature Rascal (only the Rascal Metaprogramming Language is needed).
- Select Next (several times) and accept the software license. The process may take a few minutes!
- Once these features have been installed, restart Eclipse.

After restarting Eclipse you can open up a Rascal perspective by selecting Window -> Open perspective -> Other. Select Rascal from the list.

If you run into trouble while installing please visit:  
<http://www.rascal-mpl.org/help/troubleshooting.html>

## Property based testing hands-on

Marc Evers, Willem van den Ende, Rob Westgeest

For this workshop, we assume you are somewhat familiar with either JavaScript or Haskell. We've kept the Javascript one with as little prerequisites as possible. We've tried to do the same with the Haskell one, but it assumes you have at least done some exercises with Haskell before.

If you can, clone our Github repository <https://github.com/qwaneu/property-based-tutorial> so you have the installation instructions and starting project for one of the many options with you on the day. Either bring a laptop or pair up with someone who has a laptop.

If for some reason you can't clone our Github repository, read the instructions for working browser-based with Cloud9 IDE (for Javascript) or Fp Complete. These are also the least intrusive ones to get going. The other two options are working with Javascript or Haskell on your laptop. Either through locally installing Node.js or Haskell (so please build the projects beforehand, they are in the Github repository as well) or cloning our Docker development environment and SSH-ing into it.

For detailed instructions see: <https://github.com/qwaneu/property-based-tutorial>. Feedback on the instructions are welcome, we'll keep updating them.