Jameson B., Chris L., Megan O., Alex W.

Team Silver Bullet

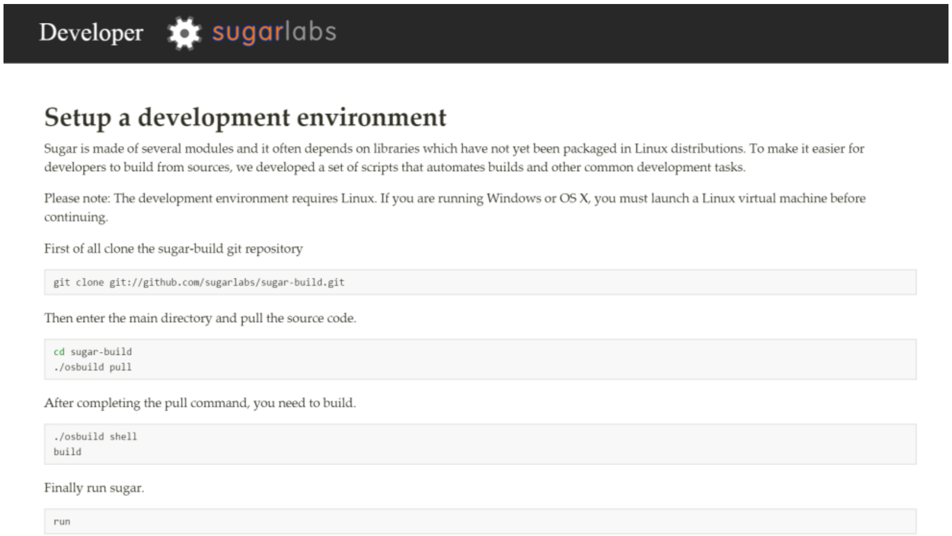
Dr. Jim Bowring

CS 362

15 September 2015

Deliverable #1

Team Silver Bullet consists of Jameson Burroughs, Christopher Lewis, Megan O'Neal, and Alex Wray. We chose Sugar Labs [wiki.sugarlabs.org] as our open source project to compile, test, and document. In this deliverable, we have successfully cloned our project to our Ubuntu virtual machines. There were a few steps in completing this task.



Setup a development environment

Sugar is made of several modules and it often depends on libraries which have not yet been packaged in Linux distributions. To make it easier for developers to build from sources, we developed a set of scripts that automates builds and other common development tasks.

Please note: The development environment requires Linux. If you are running Windows or OS X, you must launch a Linux virtual machine before continuing.

First of all clone the sugar-build git repository

```
git clone git://github.com/sugarlabs/sugar-build.git
```

Then enter the main directory and pull the source code.

```
cd sugar-build
./osbuild pull
```

After completing the pull command, you need to build.

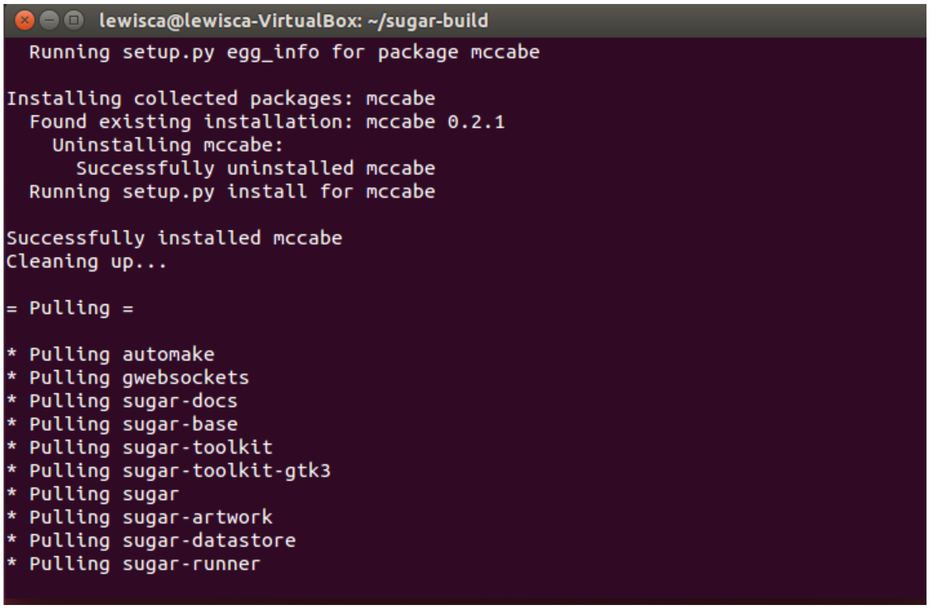
```
./osbuild shell
build
```

Finally run sugar.

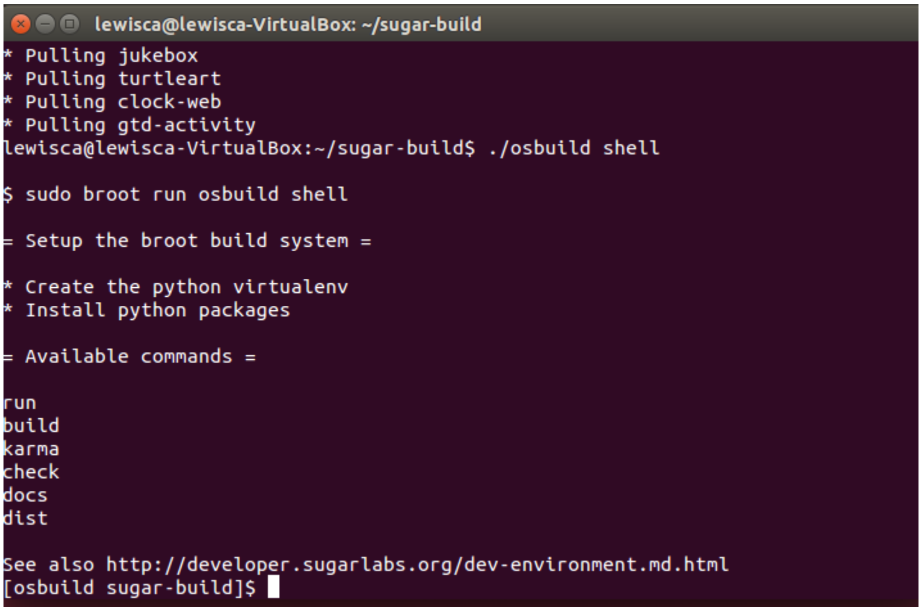
```
run
```

The wiki provided for developers by Sugar Labs provided us with the following steps.

After running './osbuild pull', the project proceeded to download.



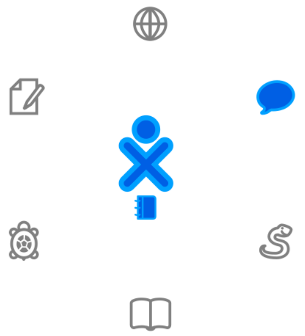
```
  Running setup.py egg_info for package mccabe

Installing collected packages: mccabe
  Found existing installation: mccabe 0.2.1
    Uninstalling mccabe:
      Successfully uninstalled mccabe
  Running setup.py install for mccabe

Successfully installed mccabe
Cleaning up...

= Pulling =

* Pulling automake
* Pulling gwebsockets
* Pulling sugar-docs
* Pulling sugar-base
* Pulling sugar-toolkit
* Pulling sugar-toolkit-gtk3
* Pulling sugar
* Pulling sugar-artwork
* Pulling sugar-datastore
* Pulling sugar-runner
```



```
lewisca@lewisca-VirtualBox: ~/sugar-build
* Pulling jukebox
* Pulling turtleart
* Pulling clock-web
* Pulling gtd-activity
lewisca@lewisca-VirtualBox:~/sugar-build$ ./osbuild shell

$ sudo broot run osbuild shell

= Setup the broot build system =

* Create the python virtualenv
* Install python packages

= Available commands =

run
build
karma
check
docs
dist

See also http://developer.sugarlabs.org/dev-environment.md.html
[osbuild sugar-build]$ 
```

After the downloads completed, we built the shell.

We ran the project from sugar's shell, and were greeted with the completed, running introduction page to Sugar Labs! The main menu was not too much farther past this.



Our experiences so far have been rather pleasant. Once we figured out how to clone the code to our Ubuntu machines, compiling the code and running it was quite simple. From the main menu, pictured above and to the left, we have, starting at the 12 o'clock spot and going clockwise, we have: web, chat, Pippy (a python editor), read, TurtleBlocks (a visual drag and drop style programming teaching tool), and write. Each one works as expected, and each is quite testable with its limitations. There is a more fleshed out version of Sugar Labs on their website in the form of an .iso file, which is undesirable for our team's purpose. We plan on working with the program we have downloaded as-is. There are more than enough possible test cases to analyze.

We found existing test files and ran them in the osbuild shell of sugar. We currently do not know exactly what they do or how to interpret the results. We do know that the python test file we have run multiple times tests the applications within Sugar Labs, so it's a good framework to start out with. We have also yet to select an activity within Sugar Labs to test yet, however our team will accomplish this task quite soon.