21M359 Music Composition Software Cheat Sheet

Creating a new system:

Copy and rename template_system.py
Fill in sections

Creating a new module:

Copy and Rename TemplateModule.py
Add **“from <your module’s name> import *”** to modules.py
Fill in generate_output(self)

Instantiate a module:

variable = ModuleName(<name>, <list input ports>, <list output ports>)
ex: A = TransposeUpOne(“TUO1”, [“in0”, “in1”], [“out0”])

Connect two modules:

<module A>.connect(<output port of A>, <module B>, <input port of B>)
ex: A.connect(“out0”, B, “in0”)

Get the input of a port in a module:

self.get_input(<port>)
ex: self.get_input(“in0”)

Set the output of a port in a module:

self.set_output(<output>, <port>)
ex: self.set_output(<output data>, “out0”)

Getting Started with Git

Installing Git: http://git-scm.com/book/en/Getting-Started-Installing-Git

Create a Github account at https://github.com/signup/free

Email your username to cdolan@mit.edu

To checkout the project:

git clone https://github.com/cgdolan/21M359Fall2012.git music_composer

Changed files --(add)--> Staging Area --(commit)--> Local Repo --(push)--> Github Repo

1. **git status**: see what files are changed

2. **git add <filename>**: add an edited file to the staging area

--OR--

git add -A: add all edited files to the staging area

3. **git commit -am "<message>"**: commit staging area changes to the local repo

4. **git pull**: get latest changes from the class repo

Note: you may need to manually resolve conflicts in each file and git commit -a again

5. **git push**: commit local repo changes to the class github repo