Team Music Makers

Demo: Iteration # 1

Describe what you were able to demo vs what was envisioned and record any problems that arose

We were able to demo multiple parts of our project to our IA mentor. In particular, we demonstrated that our Emotion Detection class works to accurately calculate the sentiment of any given text into either a positive or negative category (i.e. “happy” or “sad”). The initial problem noted in this program was that its functions were not defined in a class. This was revised later last week using the concept of object-oriented design, which eliminated many unnecessary functions and gave way to more efficient code. The further steps for this class are to include more emotions, which will be completed in the next iteration.

We also presented the music generator successfully take an emotion string and produce a MIDI file. Due to the MIDI format, we haven’t find a way to play this format on a web and we were unable to find a python module to convert MIDI to other formats. Therefore, we set the app to directly download the MIDI file. We bypassed the format issue in this iteration, and we will find a solution in the next iteration.

We also were able to show an initial login screen for our application; however, data for each user has not been taken into account into the database. For now, any username or password entered will take the user to the next screen. We showed our IA the initial design for out next page, which displays a “happy” and “sad” emoticon for user to select one of two emotions. The text box for users to enter the text will be included in the next iteration.

Show your updated Trello board and GitHub to TAs and provide links in the document you submit

Trello
Github
We have showed both the above links to our TA. Please check the links themselves for continuous updates. This week, we have mostly been working on testing in regards to the backend. We also added an appropriate README to the repo.

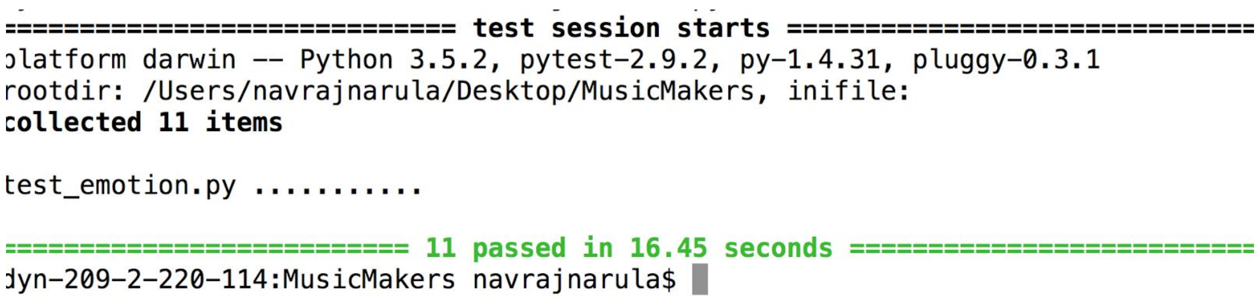
Test Cases and types of testing performed

Emotion Detection:

The Emotion Detector class takes in any text and encodes it according to utf-8 conventions. By means of natural language processing, it will then score a text as either more positive or negative depending on previously defined positive and negative words. If the percentage of positive words is higher than the percentage of negative words, “happy” will be the string passed on to the

Music Generator class. If not, this string will read “sad.” In the case that the percentages are equal to one another, the string “happy” will returned.

We performed unit testing in regards to the Emotion Detector class. It tests strings of words, asserting whether or not those values will be positive or negative (i.e. “happy” or “sad”). Empty inputs were also considered, which were considered invalid inputs. Upon this case, no string is sent to the Music Generator class. You can review some examples of our test here, which considers the positive class.



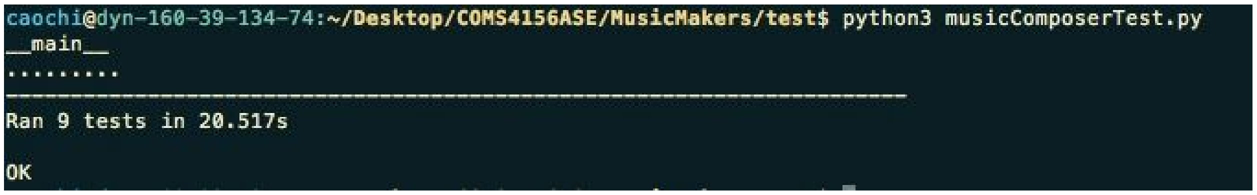
```
============================== test session starts ===============================
platform darwin -- Python 3.5.2, pytest-2.9.2, py-1.4.31, pluggy-0.3.1
rootdir: /Users/navrajnarula/Desktop/MusicMakers, inifile:
collected 11 items

test_emotion.py ...........

=========================== 11 passed in 16.45 seconds ===========================
dyn-209-2-220-114:MusicMakers navrajnarula$
```

Music Generation:

The music generator take a string, an emotion, as an input and a MIDI file as an output. Currently, the valid emotion is “happy” and “sad”, regardless of uppercase or lowercase. It does not accept any other emotion words or other strings.

I ran unit testing on the music generator. It tested all the strings, valid and invalid, and examine whether or not a corresponding MIDI file is generated. When a valid string, “happy”, “sad”, “hAppy”, etc., was tested, a corresponding MIDI file was generated; when an invalid string, “happpy”, “ha_ppy”, “exciting”, was tested, no file was generated. The Music composer part works fine.

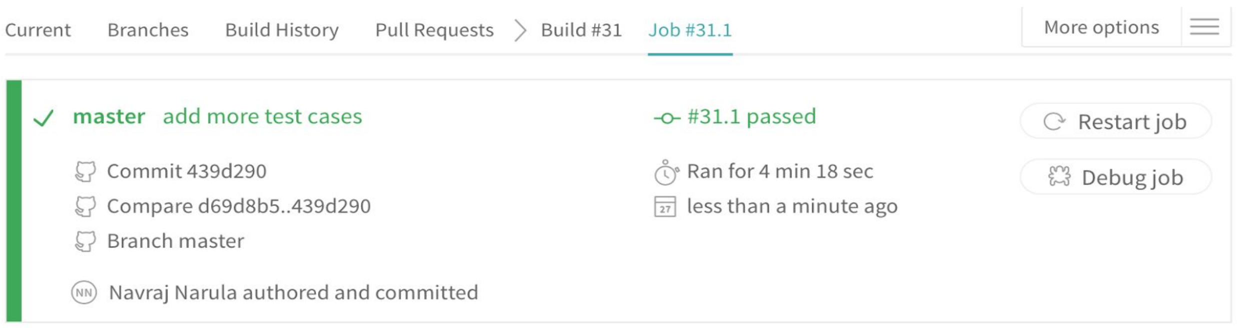
Test detail



```
caochi@dyn-160-39-134-74:~/Desktop/COMS4156ASE/MusicMakers/test$ python3 musicComposerTest.py
__main__
.........
----------------------------------------------------------------------
Ran 9 tests in 20.517s

OK
```

Continuous Integration setup and build

For our continuous integration setup and build, we chose to use Travis CI (similar to our last iteration). We again chose to use our Emotion Detection class. Below is a screenshot of our build successfully passing on 11 test cases.

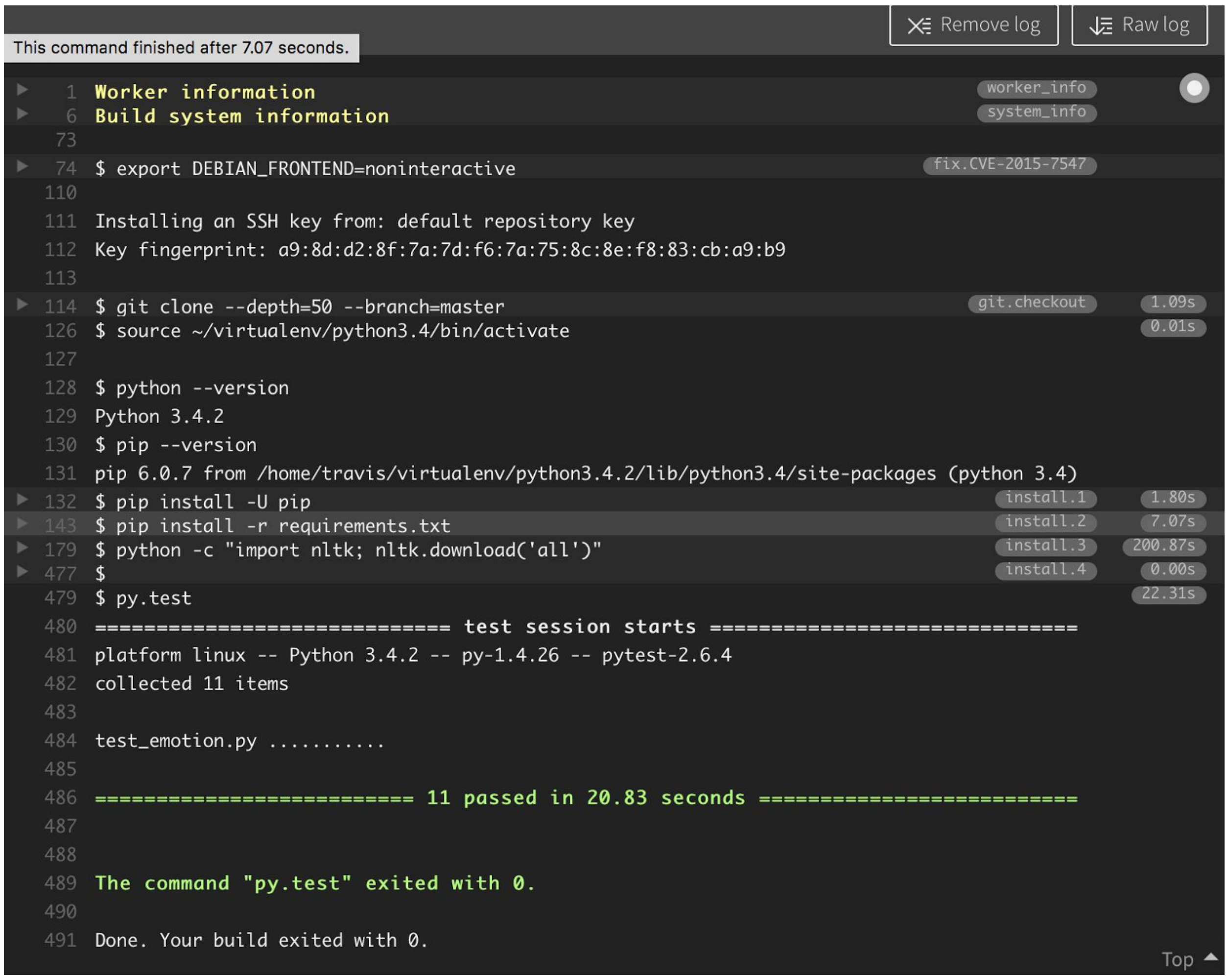


master add more test cases

Commit 439d290
Compare d69d8b5..439d290
Branch master

Navraj Narula authored and committed

#31.1 passed
Ran for 4 min 18 sec
less than a minute ago



This command finished after 7.07 seconds.

```
Worker information                                                  worker_info
Build system information                                            system_info

$ export DEBIAN_FRONTEND=noninteractive                             fix.CVE-2015-7547

Installing an SSH key from: default repository key
Key fingerprint: a9:8d:d2:8f:7a:7d:f6:7a:75:8c:8e:f8:83:cb:a9:b9

$ git clone --depth=50 --branch=master                              git.checkout    1.09s
$ source ~/virtualenv/python3.4/bin/activate                                        0.01s

$ python --version
Python 3.4.2
$ pip --version
pip 6.0.7 from /home/travis/virtualenv/python3.4.2/lib/python3.4/site-packages (python 3.4)
$ pip install -U pip                                                install.1       1.80s
$ pip install -r requirements.txt                                   install.2       7.07s
$ python -c "import nltk; nltk.download('all')"                     install.3     200.87s
$                                                                   install.4       0.00s
$ py.test                                                                          22.31s
============================== test session starts ==============================
platform linux -- Python 3.4.2 -- py-1.4.26 -- pytest-2.6.4
collected 11 items

test_emotion.py ...........

========================== 11 passed in 20.83 seconds ==========================


The command "py.test" exited with 0.

Done. Your build exited with 0.
```

We previously encountered errors related to nltk download and dependency issues. This was resolved by adding this line to our travis.yml file:

```
- python -c "import nltk; nltk.download('all')"
```

Because of this recent error and out fix on it, we are now easily able to incorporate more tests into our next iteration.

Deliverables for next iteration

For our next iteration, we plan to integrate each separate part of our project together. In regards to the backend, we will connect our Music Composer class to our Emotion Detector class. After processing a piece of text, the Emotion Detector class will return a string: either “happy” or “sad,” corresponding to the emotion buttons on the screen. After the Music Composer receives this input, it will query the corresponding pre-processed data, data trained with Hidden Markov Model, to generate music.

We also plan to make our databases management system more robust by including user information and make additions to our front-end design. We hope to also start testing the front-end in the next iteration.

Recommendations from your mentor

Our mentor advised us that we should focus on the User Interface and making it more visibly advanced for the next iteration. She said that using a template would be the best way to go, as opposed to working from scratch. Additionally, she said that we should make sure that our login page is secure, and that we aren’t using a GET request for the username and password.

She also advised that it is important to check the debug log while testing, especially in Travis CI. It is also important to keep directory and directory location in mind.