This guide is based on the files that I use the most. My knowledge is mostly oriented around the functionality of the game, rather than the mechanics of how it works or the styling/visuals. I can definitely try to answer questions outside of what I know; it just might take longer than if we consulted Julian, who wrote the code and knows the project better than me.

Installation and setup

Currently, there are two working versions of the game:

* The version on the master branch, which you can access (assuming you have git installed on your machine) by running the command

git clone <https://github.com/thatpaiguy/Wordification>

Make sure to periodically run the command git pull to pull in any changes that might be made to the branch if you want to see the latest version. You may also have to install other dependencies to get the game running.

To run the game **on the master branch**, navigate to matching\_game (cd matching\_game) and then run the command python wordification\_driver.py.

If you get an error, it likely means you don’t have dependencies (python3, pip, pygame, etc.) installed. I would recommend reading the error messages and installing the packages it tells you are missing until it runs correctly. Most can be installed using some version of the command sudo apt-get install python3, for example. However, I think some (pygame, if I am remembering correctly) require you to install pip and have special installation instructions. I think a good strategy for getting this working is Googling the error message you get, Googling how to install the package that it says is missing, and running the game until it works. If this doesn’t work for you, let me know and we’ll work on it.

For now, I think it would probably be best for you to work off of the master branch, and then we can merge our changes later. I would also recommend you install a text editor like Atom if you do not already have one!

Assuming you have been able to run the game, you should see sorting and matching options for three levels on the master branch and just sorting options if you choose to run my branch. You should be able to play the game to completion; let me know if you see a results page and if your sound works when you run it from master when you get the chance!

Important Files

* datafiles/
  + - game2words.txt
      * This file contains the words used in the game. The format is

A B C

* + - * A: word
      * B: the number associated with the vowel **sound**
      * C: the vowel itself

The master branch version of this file does not have any words that I have added this semester, but the samyuc branch does.

* + matching\_game/
    - game.py
      * Game class defined here
    - *ignore the main.py, Julian says we don’t need to worry about this right now*
    - sorting.py
      * SortingGame class defined here, inherits Game class
      * initBoardVariables method
        + This method builds the board for the chosen letter and sounds, the master only has this functionality for “i”, the samyuc branch has all five letters
    - wordification.py
      * Same as the driver file below, except this requires a login. I have been modifying the driver file below and working without the login. I’ll ask Julian for login details soon and see how this changes things.
    - wordification\_driver.py
      * Builds the game, calling the appropriate Game class
* Website/
  + I haven’t checked out this directory yet, so I don’t know where this stands yet. It appears that a Django website is already in place, based on the site\_notes… file. Typically, to run Django websites locally, you create a virtual environment and run the command python3 manage.py runserver through there.
  + Here is [more information](https://docs.djangoproject.com/en/3.0/topics/install/) on how to get started on that. It would be good if you could figure out where this stands by reading through Julian’s notes/getting in touch with him. I’ll also look into this when I have time.