**Sophia Ho**

**AndrewID: swho**

**Recitation: P**

**Conlingo**

**Description**

Conlingo is a game that allows a user to create a constructed language, whether through user inputted preferences or through computer-generated mappings of linguistic rules for the language. The program can also save the syntactic rules and generated lexicon (vocabulary) of the constructed language for later access. This program then allows the user to look through previously generated constructed languages and learn them through mini games.

**Competitive Analysis**

Like Vulgar, Conlingo is a constructed language generator. While Vulgar tries to model a natural language’s quirks and idiosyncrasies with the same types of deviations from logic in sentence structure and word formation, languages produced in Conlingo will be more rigid and will strictly follow syntactical rules. Similar to Vulgar, Conlingo will display the constructed language’s syntactic rules, syllable structure, and phonetic chart. However, unlike Vulgar, Conlingo will allow users to input their desired syllable structure and sentence order. Conlingo will also allow the user to learn the language through games, unlike Vulgar, which only displays the language’s rules.

Similar to Duolingo, Conlingo will teach the user a language through games. However, Conlingo will only teach the user constructed languages and Conlingo can also produce its own constructed languages. The mini games in Conlingo will also be different from the ones in Duolingo in that there will not be a progression from level to level, rather more of an arcade-style experience where the players can score points and try to beat their high score.

Like Babbel, Conlingo will teach the user a language through games. However, Conlingo will only offer constructed languages that have not been created before. These games will also be more point based rather than picture or translation based. Also, unlike Babbel, Conlingo will give the user the option to view the entire vocabulary, sound structure, and sentence structure of the constructed language at once for a more academic approach towards language learning.

**Structural Plan**

* File containing main program
* File containing mini game 1 program
* File containing mini game 2 program
* File containing language creator program (assigns language rules and creates vocabulary mapped to English definitions)
* File containing Menu and Button class (for display purposes)
* File containing all possible phones being used in this program and
* File containing Phone class
* File containing Word class (with subclasses for words with different parts of speech)
* File containing Sentence class
* Directory containing directories of constructed language information
  + Text file containing constructed language information

**Algorithmic Plan**

The most difficult part of this project will be being able to generate different randomized but still structured constructed languages efficiently.

1. Create classes for phones (consonants and vowels), words (with different parts of speech and different conjugation rules), and sentences
2. Assign legal structures by assigning number of consonants allowed in an onset and in a coda and by choosing all the legal consonants and vowels in this language
3. Assign word structures using legal syllables to create root words and derivational affixes
   * Nouns
     + Assign how and whether nouns will be conjugated based on gender, case, plurality
   * Verbs
     + Assign how and whether verbs will be conjugated based on tense and negation
   * Pronouns
     + Assign how and whether pronouns will be conjugated based on gender, case, plurality
   * Adjectives
     + Assign how and whether adjectives will be conjugated based on gender, case, plurality
   * Adverbs
     + Assign how adjectives are modified into adverbs
   * Determiners
     + Assign how and whether determiners will be conjugated based on gender, case, plurality
   * Prepositions
     + Assign
   * Conjunctions
     + Assign
4. Assign sentence structures
   * Decide word order (Subject-Verb-Object, Subject-Object-Verb, Verb-Subject-Object, etc.)
5. Create language by assigning all these rules and storing them in a text file along with a generated vocabulary
6. Access this text file for each language to display rules and play games

**Timeline Plan**

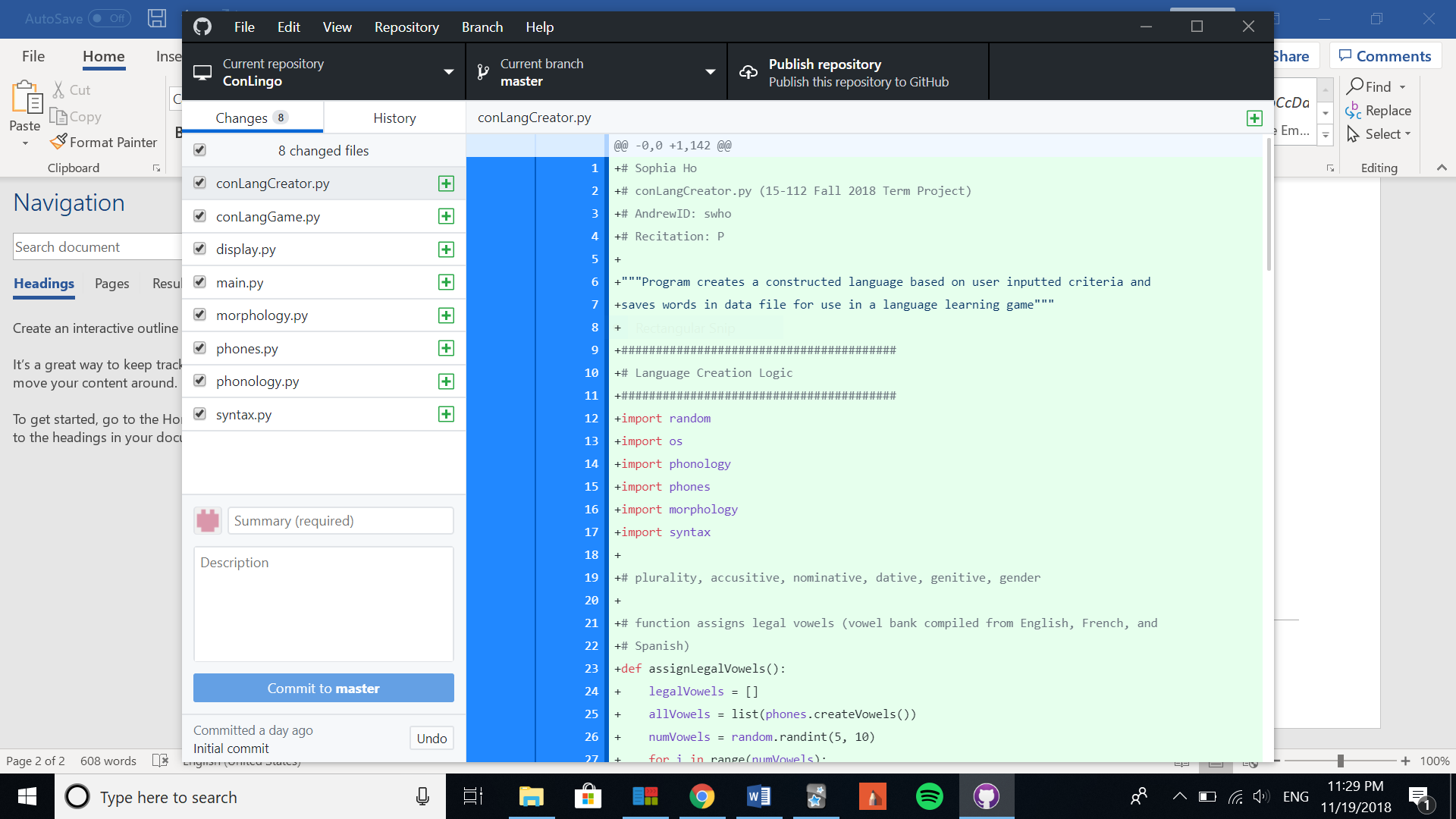
TP1: Phone class, Word class, Sentence class, word generation

TP2: display interface, mini game 1

TP3: mini game 2

**Version Control Plan**

Using Github to back up code.



**Module List**

Tkinter

**TP2 Update**

* Only one mini game
* Focus on language creation aspect