##
Project Proposal Update (see highlighted area)
Jiawei Vincent Mai
AndrewID: Jiaweim
Recitation: I
##

Project Description:

/ LEXIGRAPH /
[a more ambitious goal]
a vocabulary building application that utilizes lexical relationships between words to aid in learning new vocabulary, based on their contextual relevance to users.

[a more realistic goal]:
whether you are trying to tackle TOEFL, IELTS, SAT or GRE vocabulary, it is often daunting to imbibe a large sum vocabulary. This App aid in learning new words through clustering through lexical relationships.

Competitive Analysis:

[Google knowledge graph] - inspiration/ precedent
Google Knowledge Graph is Google's own relational data base of the worlds knowledge. It's currently implemented as an info box adjacent to the search result, which displays relevant information and enhances search.

[Graph words online thesaurus] - inspiration/ precedent
A graph visualization of lexical relationship of words in WordNet, a lexical database of English built by Princeton University.

[Oxford American Dictionary & Theausurus] - similar products
An app that integrates a dictionary with a theasurus to help those who need to build vocabulary or is in search of the right word. Its key components include "the Oxford dictionary", "special topics", "favorite words", "recently searched", "word of the day".

[Vocabulary.com]
- similar products
A adaptive vocabulary builder that personalizes learning through algorithmic analysis of user's reponses in relation to other users within its data base; and learning reinforcement through quizs and games.

[GRE Vocabulary Builder-Test Prep] - similar products
A GRE vocabulary test prep that organizes vocabulary by difficulty levels and allow users to track progress and master GRE words through quizes and games.

Taking inspiration from the precedent, LEXIGRAPH proposes an alternative approach to vocabulary building. By leveraging the latent lexical relation between newly acquired vocabulary, and the relavance between new and farmiliar concepts, the app aims at improving efficiency and effectiveness of vocabulary building.

Structural Plan:

/GUI/

[splash]
display at launch:
- word recommendation
based on users knowledge:
- recently learned
- frequently searched
- known majority of synynom neighbors

[fixed menu]
root directory for getting to various modes
- most recent # most recent search
- most frequent # most frequently searched words
- lexigraph # lexigraph visualization and customization
- search # full-screen word search
- pair # learning reinforcement by synonym/antonym pairs
- writer # (optional, provide word suggestions)

[search]
full screen word search

[dictionary]
plain vanilla dictionary result
- pronunciation
- definition(s)

[lexigraph]
lexical graph result/building
- interactive graphical display of target words and its neighboring synonyms
- users can highlight synonyms that are relevant to their knowledge

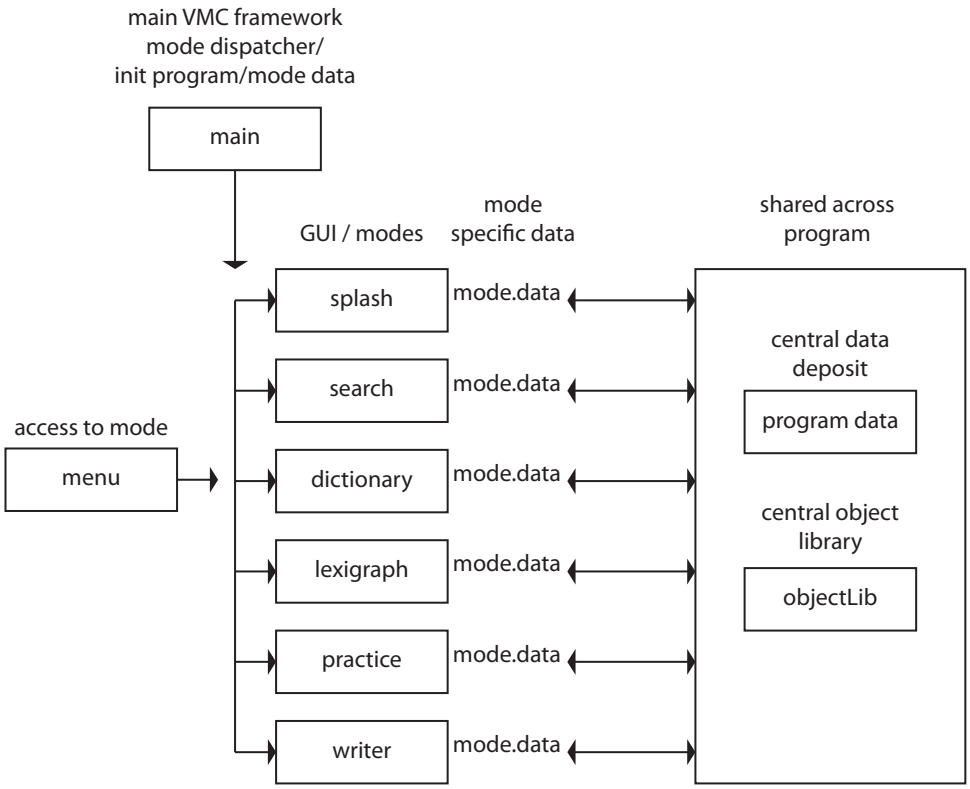
- (optional) users can add new topics to

/backend/

[lexicalgraph]
- main algorithm crawling through lexical data base (WordNet or thesaurus) and builds customize word objects
- vocabulary level estimation:
  implementing a cluster algorithm to guage ballpark of vocabulary level based on user's wordList; words within the estimated range is highlighted for future wordsearch to help facilitate more affective absorbtion.

[user profile management]
- tracks user learning progress

currently the project is organized as shown in the following diagram:



Algorithmic Plan

The key algorithm to implement for the project is the lexical graph builder, which stores all new vocabulary and their respective synonym

groups in a graph data structure. The main algorithm not only crawls through the WordNet lexical database but does it with consideration to the difficulty level of the target vocabulary's neighboring word group. Users can specify whether a lexical synonym set contains words that are more, or less relevant to their prior knowledge by highlighting or de-highlighting each synonym. The user wordList will be updated, and user's vocabulary level will be updated in realtime using a cluster algorithm. Moreover, the lexicalgraph also allows users to add topical edges to existing graph structure and tailor it to their own needs. The main functions within the lexical graph builder includes:

```
def getVocabRange(wordList):
    # returns a list of ranges of estimated user vocabulary
      level by user's wordList (generated based on user's highlight of
      known synonyms)
def getCentrality(word):
    # returns the degree of connectivity between a word and user's
      vocabulary level
def getTopicCluster(topic):
    # returns words by user defined topic clusters
def findPath(rootWord, targetWord)
    # returns the path between root and target words
def synsCompLevel(lexigraph)
    # returns a sorted list of synonym set by % mastered
```

Timeline Plan:

Nov.24	--------	functional demo of core algorithm(s), submit MVP Plan
Nov.28	--------	functional demo of GUI (TP2 due)
Dec.03	--------	MVP due,
Dec.06	--------	Term Project due, submit all required documents

Version Control Plan:

code is backed up and version controlled using google drive desktop

Module List:

NLTK	----------	NLP python module
Tkinter	----------	python built in GUI module

##