Created by Liz Gooch, Maggie Zick, and William Frazier

Naval Postgraduate School Computer Science Dept

20200316 CS3021 AY2021 Winter

CodeNames Cheater App

This is a Python program to assist with playing the card game CodeNames.

*** Important system requirements for operation. ***

This program requires Python 3.9 and runs best from the command line.

The following need to be installed:

PySimpleGUI

scipy

unittest

The following files must be present in the directory:

CodeNames.py

FIFO.py

LinkedList.py

MyDictionary.py

CodeNamesCheaterApp.py

TestCodenames.py

codenames.txt

glove.25k.50d.txt

*** How to play CodeNames. ***

This program is intended to assist the "Spymaster" on a CodeNames team.

Games usually consist of two teams, each with one "Spymaster."

25 CodeNames words are dealt for all participants to see. Three cards are winning words for Team A, three different cards are winning words for Team B, and one game over card is shared between the teams. Only the "Spymasters" are aware of these seven cards. The "Spymasters" are tasked with providing clue words to their team members. Their objective is to get their partners to pick their winning words, avoid their opponents' winning words, and avoid the game over word. The CodeNames Cheater App will provide the "Spymaster" with the five best clue words that are semantically close to winning words and far from the game over word.

*** How to use the program. ***

After ensuring the required libraries are installed, from the
command line, run CodeNamesCheaterApp.py using an updated version
of Python 3.9.

The program may take a few seconds to appear. Once the GUI appears, the program is ready to run. Enter three valid CodeNames "good" words for your team. If you do not have a set of CodeNames cards, the codenames.txt file included in this directory can be used to view all 400 options to enter.

If you enter a word not included in the 400 CodeNames words, the GUI will inform you and ask you to re-enter a valid word. The GUI is not case sensitive.

Click "get clues" and the GUI will output the top 5 clue words with their
rank and "goodness factor."

The program will provide clues indefinitely. To terminate, click the "X" in
the top right corner of the GUI.

*** How does the program work? ***

The CodeNames Cheater App imports 400 CodeNames words from the codenames.txt
file into a linked list dictionary data structure for reliability and quick
hashing of words. A Python dictionary is created with 25,000 clue words each
with an array of 50 dimensions. For each of the three "good" words and one
"bad" word, the "goodness" of all 25,000 clue words are calculated and sorted.
The top five clues are inserted into a FIFO queue data structure to be output
into the GUI.

references: https://jsomers.net/glove-codenames/
https://nlp.stanford.edu/projects/glove/

*** Required and optional techniques****

Two data structures are MyDictionary and FIFO based on Linked list. (see CodeNames.py, lines 15 and 19)

CodeNames, our app, has its own class with three attributes (see CodeNames.py, line 13). Additionally, CodeNames class consists of multiple functions such as a function that reads in the words from the games cards (see CodeNames, line 41) and defines the spatial distance between words (see CodeNames, line 87).

Our basic object function is the Class Codenames within which we created class attributes, docstrings, and setters (CodeNames, lines 8-37.)

For functional programming, we avoid imperative, step-by-step programming (see the many functions in CodeNames.py) and try to use built methods such as type testing (CodeNames, line 137) or try-expect statements (CodeNames, line 61-67.)

We include a file of unit tests called TestCodenames.py.

We demonstrate exception handling in our read functions (CodeNames, lines 41-67).

We did not find a way to include basic inheritance.

Grab bag – we used GitLab to share and work on code.

We used a second library – scipy.