

## **Mini-Manual**

Welcome USER,

Thank you for using our program. If you are unaware of what this program is, this program solves boards from the game, *Boggle*, as well as gives you the option to play the game itself. This program solves the boards through an optimization of finding if a word is valid or not with a dictionary. In short, the optimization is caused by creating a set with all the possible prefixes of words.

### **What is Boggle?**

Boggle is a game where you are given a board, usually a 4 x 4, where each cube contains a letter. The goal of the game is to create the most words by connecting letters to adjacent letters. A player can connect letters diagonally but they are not allowed to use the same cube twice for a word. Boggle is often played as a single-player game but you can also play against others to see who can find the most words or get the most points.

For our game, we used the following point system:

3 or 4 letter word = 1 point

5 letter word = 2 points

6 letter word = 3 points

7 letter word = 5 points

8 or more letter word = 11 points

### **File Input**

If you wish to input for the program to read a Boggle Board. You must enter a text file of the following format:

R C

L L L L

L L L L

L L L L

L L L L

Where R is the number of rows in the board, C is the number of columns in the board, and L is a letter that is capitalized. In the example above, R = 4 and C = 4. The spacing format must also be the same as the example. After inputting a file or playing a game, you will have the option to:

1. View the valid words of the inputted file or the board from the game
2. View the board of the inputted file or the board from the game
3. View the maximum points possible of the inputted file or the board from the game

# Design guide

## Boggle Board Solver

To evaluate the board, we use DFS. We create all possible combinations of possible moves from the directions starting from a root position. After creating all possible trees, we check in a dictionary if it's a valid word, and save that word. To optimize this process, we end a branch if the letters from the root to the leaf compose a word that is not in a prefix set. The branch gets terminated if it's not a possible prefix to build a word.

## Boggle Game

As for the game, we used Tkinter for the GUI. Nothing complicated here, just applying a Python GUI toolkit.

Here are some sources that we have consulted:

[https://youtube.com/playlist?list=PLCC34OHNcOtoC6GglhF3ncJ5rLwQrLGnV&si=M\\_eEFUTIWK-CKOEa](https://youtube.com/playlist?list=PLCC34OHNcOtoC6GglhF3ncJ5rLwQrLGnV&si=M_eEFUTIWK-CKOEa).

<https://www.geeksforgeeks.org/python-tkinter-text-widget/>.

<https://docs.python.org/3/library/tkinter.html>.

<https://realpython.com/python-gui-tkinter/>.