

Project 1

Wordle Solver

1 Introduction

Wordle is a very popular English word-guessing game that was created by Josh Wardle in 2021.¹ The game became a social media phenomenon because it's simple yet highly addictive.



Figure 1: A Wordle Gameplay Screenshot.

You can try it via these links below:

- Wordle - The New York Times (Fig 1): <https://www.nytimes.com/games/wordle/index.html>
- Wordle Game - Play Unlimited (not restricted to one game per day): <https://wordly.org/>

In the *New York Times* version, each day, the game gives you a single **secret 5-letter word** (for example: “REBUS”). You have up to 6 attempts to guess the word, the fewer, the better. After each guess, the game shows color hints:

- **Green:** the letter is correct and in the right position.
- **Yellow:** the letter is in the word, but in the wrong position.
- **Gray:** the letter is not in the word.

¹Wordle - Wikipedia: <https://en.wikipedia.org/wiki/Wordle>

2 Requirements

Before we get started, here are some important things to consider:

- The source code must be written in **Python**.
- You can use supporting libraries, but you must implement the search algorithms **yourself**.
- There is **no restrictions** on how to organize your code. However, it should be structured professionally and clearly.

2.1 Reimplement the Game (10 pts)

Implement a basic, playable version of **Wordle** (Unlimited version). The program **MUST HAVE** a Graphical User Interface (GUI), either a web application or a desktop application is acceptable. The program must be able to:

- Randomly select a secret 5-letter word from a given vocabulary list.
Here are some Wordle dictionaries you can consider:
<https://github.com/tabatkins/wordle-list>
<https://gist.github.com/scholtes/94f3c0303ba6a7768b47583aff36654d>
- Display a 6x5 grid for the player's guesses.
- Validate the player's input (must be a valid 5-letter word from the dictionary).
- Provide color feedback (Green, Yellow, Gray) for each character in the guessed word based on the secret word.
- End the game when the player guesses the word correctly or uses all 6 attempts.

2.2 BFS Solver (10 pts)

- Implement a Wordle solver using **Breadth-First Search**.

2.3 DFS Solver (10 pts)

- Implement a Wordle solver using **Depth-First Search**.
- The Iterative Deepening Search (IDS) algorithm can be used as an alternative if you want.

2.4 UCS Solver (15 pts)

- Implement a Wordle solver using **Uniform-Cost Search**.
- Define and justify your cost function clearly in the report.

2.5 A* Solver (15 pts)

- Implement a Wordle solver using **A* Search** with an **appropriate heuristic** that estimates how close a candidate word is to the target.
- Justify your heuristic and discuss whether it is *admissible* and/or *consistent* in the report.

2.6 Report (30 pts)

Submit a well-formatted PDF report with:

- **Project Planning and Task Distribution:** Document the team member responsibilities, which includes information on each task assigned to team members and the completion rate. E.g., Student A has percentage of completion 90% and the group work has total score of 9.0, then A receives a score of $9.0 * 90\% = 8.1$.

Therefore, it is important to evaluate the contribution of group members fairly.

- **Algorithm Description (10 pts):** Provide a detailed explanation of each search algorithm.
- **Experiments (20 pts):** Assess the search performance via Search Time, Memory Usage, Expanded Nodes and Average number of guesses; with measurements, visualizations (chart) and your insights.
- **References and Appendix.**

2.7 Video (10 pts)

Upload a demo to YouTube and provide the **public link** in the report. The video content includes:

- **Graphical Interface (5 pts):** Demonstrate the GUI, highlighting key visual elements.
- **Feature Presentation (5 pts):** Demonstrate your implemented functionalities concisely. The video should have subtitles or narration to make it easy to follow.

3 Submission

Please follow to the following submission guidelines:

- Your source code and report must be contributed in the form of a compressed file (.zip, .rar) and named according to the format `StudentID1_StudentID2_...`
- If the compressed file is larger than 25MB, upload it to Google Drive and share it via a link.
Absolutely no modifications are allowed after the deadline.

Example details of the directory organization:

```
StudentID1_StudentID2_...
├── Source
│   ├── main.py
│   ├── README.md (how to run source code)
│   ├── requirements.txt (libraries to be installed)
│   └── ...
└── Report.pdf (included demo video URLs)
```

4 Notices

Please pay attention to the following notices:

- This is a **GROUP** assignment.
- Duration: about 3 weeks.
- AI tools are **not restricted**; however, students should use them wisely. Lab instructors have the right to conduct additional oral interviews with random groups to assess their knowledge of the project.
- Any form of plagiarism, dishonesty, or misconduct will result in a grade of zero for the course.

The end.