

Cordle: A Word Ladder Puzzle

Ly Huynh Minh Dang

October 29, 2025

1 Executive Summary

This document summarizes the Word Ladder project: a desktop puzzle game built with Python and Pygame. The original codebase began as a Wordle-style project and evolved into a fully-featured Word Ladder game with a Halloween theme, intelligent hinting, timed challenges, and visual animations.

2 Objectives

- Build a cross-platform desktop Word Ladder game using Python and Pygame.
- Provide playable puzzles across multiple word lengths (4–15 letters).
- Implement an adaptive difficulty system so longer words present shorter transformation paths.
- Integrate a hint system (dictionary lookups + path hints) and persistent personal best time storage.
- Add atmospheric visuals and animations (Halloween theme, bat overlays, pumpkins, corgi mascots).

3 Gameplay and Rules

3.1 Core Rules

1. Each puzzle gives a **start word** and a **target word** of the same length.
2. The player must transform the start into the target by changing exactly one letter per step.
3. Every intermediate word must exist in the supplied word lists (the “word_data/” files).
4. The goal is to reach the target word following a valid chain of English words.

3.2 Difficulty and Path Length Policy

To keep gameplay balanced, the project uses an adaptive path policy:

- Shorter words (4–9 letters) may have longer solution paths to provide interesting puzzles.
- Long words (10+ letters) are deliberately constrained to short paths (typically 2–4 steps) because transforming long words is inherently more difficult.

3.3 Example

CAT -> COT -> COG -> DOG (3 steps)

4 Key Features

4.1 Hint System

- Dictionary Hints: The game can fetch live definitions from an online dictionary API to help players learn unfamiliar words.
- Path-Based Hints: When requested, the system can reveal a next step from the optimal solution path to the target.
- Limited Hints: Hints are rate-limited to encourage puzzle solving rather than hint-spamming.

4.2 Timer and Progress Tracking

- Each puzzle is timed; completion times are recorded to a local JSON file (`best_times.json`).
- Personal bests are shown in the UI so players can try to beat previous records by word length.

4.3 Visuals and Animations

- Halloween theme with light, accessible colors, bat-swarm overlays, and pumpkin decorations.
- Jumping animation: When a submitted word is valid, tile cells animate with a small vertical "jump" as their hint colors reveal.
- Shaking animation: When a submitted word is invalid (not in the dictionary or not one-letter change), the current row shakes horizontally to indicate the error.
- The UI uses a custom Halloween font and themed images (corgi pumpkins and decorative pumpkins) placed symmetrically around the title.

4.4 Controls

- Keyboard: Type letters, **Enter** to submit, **Backspace** to delete.
- Mouse: Use the on-screen virtual keyboard to type or click UI buttons (Hint, Back to Menu, etc.).

5 Implementation Notes

5.1 Code Organization

- `main.py` - game loop and state management
- `game_logic.py` - path generation, validation, dictionary/API integration
- `drawing.py` - all rendering and animation helpers
- `constants.py` - colors, sizes, animation constants
- `word_data/` - word lists by length (`wordlist_04.txt ... wordlist_15.txt`)
- `assets/` - images and fonts used by the theme
- `best_times.json` - auto-generated personal bests

5.2 Animation Implementation (brief)

Animations are implemented in the rendering layer. A simple animation state machine controls two main effects:

- Jumping: a per-tile delayed sine ease that displaces tile Y positions for a short duration when revealing colors.
- Shaking: a horizontal oscillation applied to the entire row for invalid submissions.

6 Installation and Running

1. Ensure Python 3.10+ is installed.
2. Create and activate a virtual environment (recommended):

```
python3 -m venv venv
source venv/bin/activate    # macOS / Linux
```

3. Install dependencies:

```
pip install -r requirements.txt
```

4. Run the game:

```
python main.py
```

5. (Optional) The dictionary API requires an internet connection for live definitions; the game gracefully falls back if offline.

7 Project Link and Demo

- Repository: <https://github.com/DankDaPancake/WordLadder>
- Demo video: <https://youtu.be/MoZUWvpFm5E>

8 Conclusion

This Word Ladder project showcases a compact but polished puzzle experience with an eye toward accessibility and player feedback. The Halloween theming, combined with clear animations and an intelligent hint system, aims to make gameplay both fun and educational. Future work can include additional puzzle generators, multiplayer challenges, and more refined difficulty tuning.