**Bonus Feature:**

- A GUI implementation.

- an on-screen keyboard indicating which word is hit/present and miss. Which makes user easier to make good decisions during the game.

**Decision / trade-off made** **during development:**

**Decision making of choosing programming language:**

I conducted some research after reviewing the assignment requirements and chose to use Python for development. Python offers strong compatibility with GUI frameworks and provides tools that simplify building a server/client game.

While I’m not an expert in Python, I’ve used AI assistance during development. That said, I’ve made sure to fully understand the program’s logic—because without that, debugging and making future modifications would be difficult. I want to emphasize that I’m using AI responsibly and not relying on it blindly or “abusing” it

**Decision making during development:**

**Words list**

At first, I was thinking of fetching a word list through a database through API or something. But I realize that words aren’t changing often, so I downloaded a text file containing 5-letters to make it as a source of word list.

**Coding**

Some AI-generated codes are powerful, and (at least for me) it is hard to debug and make modifications. So, I would rather write it more straight foward. Given the following example, commanded one is AI-generated code.一張含有 文字, 多媒體軟體, 軟體, 字型 的圖片

AI 產生的內容可能不正確。

**Debugging**

Debugging the normal Wordle logic was straightforward, I printed the word selected by the program to the console and verified whether the feedback matched the expected logic.

However, debugging the “host cheating” feature was time-consuming to trace how the program selects a word after the user inputs theirs. To manage this, I followed the provided example closely to ensure the core logic of the “host cheating” mechanism was implemented correctly.

As a trade-off, I chose not to test this feature in a fully comprehensive way, focusing instead on confirming its foundational behavior.

**Assignment requirements**

After developing the core logic of the game, I re-read the assignment requirements and noticed that the “maximum number of rounds before game over” should be configurable. So, I added a command-line argument that allows users to set the round limit themselves. This reminded me it is import to ensure a product aligns with user requirements especially in a business context.