

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ
Bộ môn: Công nghệ Thông tin.

BÀI TẬP KẾT THÚC MÔN HỌC

MÔN HỌC
LẬP TRÌNH PYTHON

Sinh viên: Nguyễn Tiến Đức.

Lớp: K58KTP.01

Giáo viên GIẢNG DẠY: TS. Nguyễn Văn Huy

Link GitHub:

https://github.com/NguyenTienDuc2004/bai_tap_ket_thuc_mon_hoc_python



**TRƯỜNG
ĐHKTCN**

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN TỬ

Độc lập - Tự do - Hạnh phúc

BÀI TẬP KẾT THÚC MÔN HỌC

MÔN HỌC: LẬP TRÌNH PYTHON

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

Sinh viên: Nguyễn Tiến Đức

Lớp: K58KTP.01 Ngành: Kỹ Thuật Máy Tính

Giáo viên hướng dẫn: TS.Nguyễn Văn Huy

Ngày giao đề Ngày hoàn thành

Tên đề tài : Game hangman GUI

Yêu cầu :Đọc file,xử lý ngoại lệ khi không mở được file

Cập nhật GUI: label từ ẩn,Label số lần sai,Canvas để vẽ

Kết thúc thắng hoặc thua với MessageBox

Chơi lại (nút “Chơi lại”)

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

Thái Nguyên, ngày....tháng....năm 20....

GIÁO VIÊN HƯỚNG DẪN

(Ký ghi rõ họ tên)

MỤC LỤC

Chương 1. Giới thiệu đầu bài.....	5
Chương 2. Cơ sở lý thuyết.....	8
2.1. Thư viện Tkinter	8
2.2. Cấu trúc dữ liệu cơ bản	8
2.3. Lập trình hướng đối tượng (OOP)	9
2.4. Xử lý tệp tin và Ngoại lệ.....	9
Chương 3. Thiết kế và xây dựng chương trình.....	10
3.1. Sơ đồ khối hệ thống và Biểu đồ phân cấp chức năng.....	11
3.2. Sơ đồ khối các thuật toán chính.....	12
3.3. Cấu trúc dữ liệu.....	15
3.4. Chương trình	15
Chương 4 .Thực nghiệm chương trình	Error! Bookmark not defined.
4.1. Chương trình đã làm được	16
4.2. Kết luận.....	20
Kết luận và Hướng Phát Triển cho Trò Chơi Hangman.....	20
Kết luận	20
Hướng Phát Triển.....	20

Lời mở đầu

Trong thời đại công nghệ phát triển mạnh mẽ, việc ứng dụng lập trình để xây dựng các trò chơi giải trí không chỉ giúp người học rèn luyện kỹ năng tư duy logic mà còn tạo ra những sản phẩm có tính tương tác cao và hấp dẫn. Trong khuôn khổ môn học lập trình Python, em đã lựa chọn xây dựng trò chơi **Hangman (Treo cổ)** – một trò chơi đoán chữ quen thuộc, mang tính giải trí và giáo dục cao.

Chương trình được phát triển bằng ngôn ngữ Python kết hợp với thư viện **Tkinter** để tạo giao diện người dùng thân thiện. Qua dự án này, em không chỉ rèn luyện kỹ năng lập trình cơ bản mà còn hiểu rõ hơn về cách tổ chức mã lệnh, xử lý sự kiện, và quản lý trạng thái trò chơi trong môi trường đồ họa.

Báo cáo này sẽ trình bày quá trình xây dựng trò chơi Hangman, từ ý tưởng, thiết kế đến hiện thực hóa bằng mã nguồn Python, cùng với những kết quả đạt được và hướng phát triển tiếp theo.

Chương 1. Giới thiệu đầu bài

Bài tập yêu cầu xây dựng một trò chơi Hangman (Người Treo Cổ) sử dụng giao diện đồ họa (GUI) được phát triển với thư viện Tkinter của Python. Người chơi sẽ tham gia vào quá trình đoán các chữ cái để tìm ra một từ bí mật đã được lưu trữ sẵn trong một tệp tin văn bản có tên words.txt.

Tính năng của chương trình:

- Đọc dữ liệu từ tệp tin: Chương trình có khả năng đọc danh sách các từ từ tệp tin words.txt.
- Xử lý ngoại lệ: Chương trình được thiết kế để xử lý các trường hợp lỗi có thể xảy ra trong quá trình đọc tệp, ví dụ như tệp không tồn tại hoặc tệp bị trống.
- Cập nhật giao diện người dùng (GUI): Giao diện sẽ hiển thị trạng thái hiện tại của từ cần đoán (các chữ cái đã đoán đúng sẽ hiển thị, các chữ cái chưa đoán sẽ là dấu gạch dưới). Giao diện cũng hiển thị số lần người chơi đã đoán sai và vẽ hình người treo cổ tương ứng với số lần sai đó trên một Canvas.
- Thông báo kết quả: Khi trò chơi kết thúc (người chơi đoán đúng toàn bộ từ hoặc đoán sai quá số lần quy định), chương trình sẽ hiển thị thông báo kết quả (thắng hoặc thua) thông qua hộp thoại Messagebox.
- Chức năng chơi lại: Cung cấp nút "Chơi Lại" để người chơi có thể bắt đầu một ván mới bất cứ lúc nào.
- Kiểm tra đầu vào: Đảm bảo người chơi chỉ nhập một chữ cái hợp lệ cho mỗi lần đoán.

Thách thức:

- Đọc và xử lý tệp tin đúng định dạng, bao gồm cả việc xử lý các trường hợp lỗi khi tệp không tồn tại hoặc trống.
- Quản lý trạng thái của trò chơi (từ cần đoán, các chữ cái đã đoán, số lần sai).
- Cập nhật giao diện GUI một cách linh hoạt dựa trên tương tác của người chơi.
- Vẽ hình người treo cổ từng bước trên Canvas dựa trên số lần đoán sai.
- Xử lý sự kiện từ người dùng (nhập chữ cái, nhấn nút).

Kiến thức vận dụng:

Để hoàn thành bài tập này, các kiến thức đã được vận dụng bao gồm:

- Lập trình hướng đối tượng (OOP) trong Python để cấu trúc chương trình thành các lớp và đối tượng (ví dụ: lớp HangmanGame).
- Làm việc với thư viện Tkinter để xây dựng giao diện đồ họa, bao gồm các widget như Label, Entry, Button, Canvas, và MessageBox.
- Thao tác với tệp tin trong Python để đọc dữ liệu từ words.txt.
- Xử lý ngoại lệ (Exception Handling) trong Python để đảm bảo chương trình chạy ổn định ngay cả khi có lỗi liên quan đến tệp tin.
- Sử dụng các cấu trúc dữ liệu cơ bản như chuỗi (string), tập hợp (set), và danh sách (list).
- Áp dụng logic điều kiện và vòng lặp để quản lý luồng chơi và kiểm tra kết quả.
- Sử dụng mô-đun random để chọn ngẫu nhiên từ cần đoán.

Chương 2. Cơ sở lý thuyết

Chương này sẽ trình bày các khái niệm và cấu trúc dữ liệu cơ bản trong lập trình Python cùng với thư viện Tkinter, là nền tảng để xây dựng trò chơi Hangman.

2.1. Thư viện Tkinter

Tkinter là thư viện chuẩn của Python để tạo ra giao diện người dùng đồ họa (GUI). Nó cung cấp các công cụ (gọi là "widget") để xây dựng cửa sổ, nút bấm, nhãn hiển thị, ô nhập liệu, vùng vẽ, v.v. Trong chương trình này, Tkinter được sử dụng để:

- Tạo cửa sổ chính của trò chơi (tk.Tk()).
- Hiển thị trạng thái từ cần đoán (tk.Label).
- Hiển thị số lần đoán sai (tk.Label).
- Tạo vùng vẽ hình người treo cổ (tk.Canvas).
- Tạo ô để người chơi nhập chữ cái đoán (tk.Entry).
- Tạo nút bấm để gửi lượt đoán và chơi lại (tk.Button).
- Hiển thị thông báo kết quả trò chơi (tkinter.messagebox).
- Xử lý các sự kiện tương tác của người dùng (nhập liệu vào Entry, nhấn Button).

2.2. Cấu trúc dữ liệu cơ bản

Chương trình sử dụng các cấu trúc dữ liệu sẵn có trong Python để quản lý thông tin của trò chơi:

- **List (Danh sách):**
 - Được sử dụng để lưu trữ danh sách các từ có thể dùng cho trò chơi, đọc từ tệp words.txt (self.words).
 - Các từ trong danh sách có thể được truy cập theo chỉ mục hoặc lặp qua.
- **Set (Tập hợp):**
 - Được sử dụng để lưu trữ các chữ cái mà người chơi đã đoán (self.guessed_letters).
 - Set cung cấp khả năng kiểm tra sự tồn tại của một phần tử (guess in self.guessed_letters) và thêm phần tử mới (self.guessed_letters.add(guess)) một cách hiệu quả, đặc biệt hữu ích khi cần kiểm tra xem một chữ cái đã được đoán hay chưa mà không quan tâm đến thứ tự hoặc lặp lại.

- **String (Chuỗi):**

- Biểu diễn từ bí mật (self.word).
- Biểu diễn các chữ cái do người chơi nhập vào (guess).
- Được sử dụng để tạo ra chuỗi hiển thị trạng thái từ với các dấu gạch dưới cho chữ cái chưa đoán (self.display_word()).

2.3. Lập trình hướng đối tượng (OOP)

Chương trình được xây dựng dựa trên mô hình lập trình hướng đối tượng bằng cách định nghĩa lớp HangmanGame.

- **Class (HangmanGame):** Là một khuôn mẫu (blueprint) để tạo ra các đối tượng quản lý trò chơi Hangman.
- **Object:** Một thể hiện (instance) của lớp HangmanGame (game = HangmanGame(root)).
- **Attributes (Thuộc tính):** Các biến lưu trữ trạng thái của đối tượng, ví dụ: self.root, self.words, self.word, self.guessed_letters, self.errors, self.game_over.
- **Methods (Phương thức):** Các hàm được định nghĩa trong lớp, biểu diễn hành vi của đối tượng, ví dụ: __init__ (hàm khởi tạo), load_words, reset_game, process_guess, draw_hangman, display_word.

Việc sử dụng OOP giúp tổ chức mã nguồn một cách rõ ràng, dễ quản lý, bảo trì và mở rộng.

2.4. Xử lý tệp tin và Ngoại lệ

- **Đọc tệp tin:** Chương trình mở và đọc nội dung của tệp words.txt để lấy danh sách các từ. Thao tác này được thực hiện trong phương thức load_words sử dụng hàm open() và khối with.
- **Xử lý ngoại lệ (Exception Handling):** Để đảm bảo chương trình không bị dừng đột ngột khi có lỗi xảy ra trong quá trình đọc tệp (ví dụ: tệp không tồn tại, lỗi quyền truy cập), khối try...except được sử dụng.
 - try: Chứa đoạn mã có khả năng gây ra lỗi (ví dụ: mở tệp).
 - except FileNotFoundError: Bắt lỗi khi tệp không tìm thấy.
 - except Exception as e: Bắt các lỗi khác có thể xảy ra trong quá trình đọc tệp.
 - Trong các khối except, chương trình hiển thị thông báo lỗi cho người dùng bằng messagebox.showerror và thoát

chương trình một cách an toàn nếu không thể tiếp tục (sys.exit(1)).

Những cơ sở lý thuyết này tạo nên cấu trúc và chức năng cốt lõi của chương trình game Hangman GUI.

Chương 3. Thiết kế và xây dựng chương trình

Chương này trình bày chi tiết về cấu trúc, thiết kế và các thành phần chính của chương trình game Hangman GUI.

3.1. Sơ đồ khối hệ thống và Biểu đồ phân cấp chức năng

Sơ đồ khối hệ thống (Mô tả):

Chương trình có thể được hình dung gồm các khối chính sau, tương tác với nhau:

- **Khối Quản lý Giao diện (GUI Manager):** Sử dụng thư viện Tkinter để tạo và quản lý tất cả các thành phần giao diện (cửa sổ, nhãn, ô nhập, canvas, nút bấm). Nó nhận sự kiện từ người dùng (nhập liệu, nhấn nút) và hiển thị thông tin trò chơi.
- **Khối Logic Trò chơi (Game Logic):** Chứa các thuật toán và biến trạng thái quản lý luật chơi. Nó lưu trữ từ bí mật, các chữ cái đã đoán, số lần sai, và trạng thái kết thúc trò chơi. Khối này nhận dữ liệu đoán từ Khối Quản lý Giao diện và xử lý nó.
- **Khối Xử lý Dữ liệu (Data Handler):** Có nhiệm vụ đọc danh sách từ từ tệp words.txt. Nó xử lý các lỗi liên quan đến tệp tin và cung cấp danh sách từ cho Khối Logic Trò chơi.
- **Khối Vẽ (Drawing Module):** Liên kết với Khối Logic Trò chơi để nhận thông tin về số lần sai hiện tại và sử dụng Khối Quản lý Giao diện (Canvas) để vẽ các bộ phận của hình người treo cổ tương ứng.

Quan hệ giữa các khối:

- Khối Quản lý Giao diện giao tiếp với người dùng và gửi đầu vào (chữ cái đoán) tới Khối Logic Trò chơi. Nó nhận thông tin hiển thị (từ ẩn, số lần sai, yêu cầu vẽ) từ Khối Logic Trò chơi và Khối Vẽ để cập nhật hiển thị.
- Khối Logic Trò chơi sử dụng Khối Xử lý Dữ liệu để tải từ ban đầu, xử lý các lượt đoán, cập nhật trạng thái và thông báo kết quả hiển thị qua Khối Quản lý Giao diện.
- Khối Vẽ nhận trạng thái lỗi từ Khối Logic Trò chơi và sử dụng các công cụ vẽ của Khối Quản lý Giao diện để vẽ hình.

Biểu đồ phân cấp chức năng (Mô tả):

Các chức năng chính của chương trình được phân cấp như sau:

- **Quản lý Trò chơi (Game Management)**

- Khởi tạo Trò chơi (Initialization)
 - Tải danh sách từ
 - Thiết lập trạng thái ban đầu
 - Xây dựng giao diện GUI
- Thiết lập lại Trò chơi (Reset Game)
 - Chọn từ mới
 - Đặt lại trạng thái (số lỗi, chữ cái đã đoán)
 - Cập nhật lại GUI ban đầu
- Xử lý Lượt đoán (Process Guess)
 - Nhận đầu vào từ người dùng
 - Kiểm tra tính hợp lệ của đầu vào
 - Cập nhật trạng thái đoán (chữ cái đã đoán, số lỗi)
 - Kiểm tra điều kiện thắng/thua
- Kết thúc Trò chơi (End Game)
 - Hiện thị kết quả (thắng/thua)
 - Vô hiệu hóa nhập liệu/nút đoán
- **Hiện thị (Display)**
 - Hiện thị trạng thái từ (Display Word State)
 - Hiện thị số lần sai (Display Error Count)
 - Vẽ hình người treo cổ (Draw Hangman)
 - Hiện thị thông báo (Show Messages)
- **Xử lý Dữ liệu (Data Handling)**
 - Đọc tệp từ (Read Word File)
 - Xử lý lỗi đọc tệp (Handle File Errors)

3.2. Sơ đồ khối các thuật toán chính

Thuật toán load_words:

- **Chức năng:** Đọc và chuẩn bị danh sách từ từ tệp.
- **Đầu vào:** Đường dẫn tệp words.txt.
- **Các khối/Bước:**
 1. Mở tệp words.txt ở chế độ đọc ("r") với mã hóa utf-8.
 2. Đọc tất cả các dòng trong tệp.
 3. Lặp qua từng dòng: loại bỏ khoảng trắng thừa ở đầu/cuối dòng, chuyển thành chữ in hoa.
 4. Lưu các dòng đã xử lý vào một danh sách.

5. Kiểm tra xem danh sách có từ nào không.
 6. **Đầu ra:** Trả về danh sách các từ nếu thành công và danh sách không rỗng. Trả về None nếu tệp không tìm thấy (FileNotFoundError), có lỗi đọc khác (Exception), hoặc danh sách từ bị rỗng.
- **Quan hệ đầu vào ra:** Đầu vào là đường dẫn tệp, đầu ra là danh sách từ hoặc tín hiệu lỗi (None).

Thuật toán reset_game:

- **Chức năng:** Khởi tạo hoặc đặt lại trạng thái trò chơi về ban đầu.
- **Đầu vào:** Không có (sử dụng thuộc tính self.words).
- **Các khối/Bước:**
 1. Kiểm tra xem danh sách từ (self.words) có tồn tại và không rỗng không. Nếu không, hiển thị lỗi và thoát.
 2. Chọn ngẫu nhiên một từ từ self.words và gán cho self.word.
 3. Khởi tạo tập hợp rỗng cho self.guessed_letters.
 4. Đặt số lỗi self.errors về 0.
 5. Đặt cờ self.game_over về False.
 6. Nếu giao diện đã được tạo: Cập nhật hiển thị của nhãn từ ẩn và nhãn số lỗi, xóa hình vẽ cũ trên canvas và vẽ lại giá treo cổ ban đầu, xóa nội dung trong ô nhập liệu, kích hoạt lại ô nhập liệu và nút đoán.
- **Đầu ra:** Không có (cập nhật trạng thái nội bộ và giao diện).
- **Quan hệ đầu vào ra:** Phụ thuộc vào self.words, cập nhật nhiều thuộc tính trạng thái và giao diện.

Thuật toán process_guess:

- **Chức năng:** Xử lý lượt đoán chữ cái của người chơi.
- **Đầu vào:** Sự kiện (ví dụ: nhấn Enter, nhấn nút "Đoán") hoặc gọi trực tiếp. Lấy dữ liệu từ ô nhập liệu (self.entry).
- **Các khối/Bước:**
 1. Kiểm tra nếu game đã kết thúc (self.game_over). Nếu có, bỏ qua xử lý.
 2. Lấy chữ cái từ ô nhập liệu, loại bỏ khoảng trắng, chuyển thành chữ in hoa.
 3. Xóa nội dung trong ô nhập liệu.
 4. Kiểm tra tính hợp lệ của chữ cái đoán (không rỗng, chỉ có 1 ký tự, là chữ cái). Nếu không hợp lệ, hiển thị cảnh báo.

5. Kiểm tra xem chữ cái đã được đoán trước đó chưa. Nếu có, hiển thị thông báo.
6. Thêm chữ cái vào tập hợp `self.guessed_letters`.
7. Kiểm tra xem chữ cái đoán có trong `self.word` không. Nếu không, tăng `self.errors` lên 1.
8. Cập nhật hiển thị của nhân từ ẩn và nhân số lỗi.
9. Vẽ lại hình người treo cổ dựa trên số lỗi mới.
10. Kiểm tra điều kiện thắng (tất cả chữ cái trong `self.word` đều nằm trong `self.guessed_letters`). Nếu thắng, hiển thị thông báo thắng, đặt `self.game_over = True`, vô hiệu hóa ô nhập liệu và nút đoán.
11. Kiểm tra điều kiện thua (`self.errors >= 6`). Nếu thua, hiển thị thông báo thua (kèm theo từ bí mật), đặt `self.game_over = True`, vô hiệu hóa ô nhập liệu và nút đoán.

- **Đầu ra:** Không có (cập nhật trạng thái nội bộ và giao diện, hiển thị thông báo).
- **Quan hệ đầu vào ra:** Đầu vào từ ô nhập liệu và sự kiện, xử lý logic, cập nhật trạng thái và giao diện, kiểm tra kết quả.

Thuật toán `draw_hangman`:

- **Chức năng:** Vẽ hình người treo cổ trên canvas dựa trên số lần đoán sai.
- **Đầu vào:** Số lỗi hiện tại (`self.errors`).
- **Các khối/Bước:**
 1. Xóa tất cả các hình vẽ hiện có trên canvas.
 2. Vẽ các bộ phận cố định của giá treo cổ (đế, cột, thanh ngang, dây thừng).
 3. Kiểm tra giá trị của `self.errors` và vẽ thêm các bộ phận của người treo cổ tương ứng:
 - `errors > 0`: Vẽ đầu.
 - `errors > 1`: Vẽ thân.
 - `errors > 2`: Vẽ tay trái.
 - `errors > 3`: Vẽ tay phải.
 - `errors > 4`: Vẽ chân trái.
 - `errors > 5`: Vẽ chân phải.
- **Đầu ra:** Không có (cập nhật hình ảnh trên canvas).
- **Quan hệ đầu vào ra:** Đầu vào là số lỗi, đầu ra là các lệnh vẽ lên canvas.

3.3. Cấu trúc dữ liệu

Các cấu trúc dữ liệu chính được sử dụng để lưu trữ thông tin trong chương trình bao gồm các thuộc tính của lớp HangmanGame:

- `self.root`: Đối tượng cửa sổ gốc của Tkinter. Kiểu dữ liệu: Đối tượng `tk.Tk`. Lưu trữ tham chiếu đến cửa sổ chính.
- `self.words`: Danh sách các từ đọc từ tệp `words.txt`. Kiểu dữ liệu: list chứa các chuỗi (str). Mỗi phần tử là một từ tiềm năng cho trò chơi.
- `self.word`: Từ bí mật được chọn ngẫu nhiên cho ván hiện tại. Kiểu dữ liệu: str.
- `self.guessed_letters`: Tập hợp các chữ cái mà người chơi đã đoán trong ván hiện tại. Kiểu dữ liệu: set chứa các chuỗi (str). Set giúp việc kiểm tra sự tồn tại của một chữ cái nhanh chóng và tránh lưu trữ lặp lại.
- `self.errors`: Số lần người chơi đoán sai trong ván hiện tại. Kiểu dữ liệu: int. Tăng lên mỗi khi đoán sai và quyết định việc vẽ thêm bộ phận người treo cổ.
- `self.game_over`: Cờ báo hiệu trạng thái trò chơi đã kết thúc hay chưa. Kiểu dữ liệu: bool (True hoặc False).

Ngoài ra, các widget của Tkinter như `self.word_label`, `self.error_label`, `self.canvas`, `self.entry`, `self.guess_button`, `self.replay_button` cũng là các đối tượng lưu trữ trạng thái và cấu hình hiển thị của chúng.

3.4. Chương trình

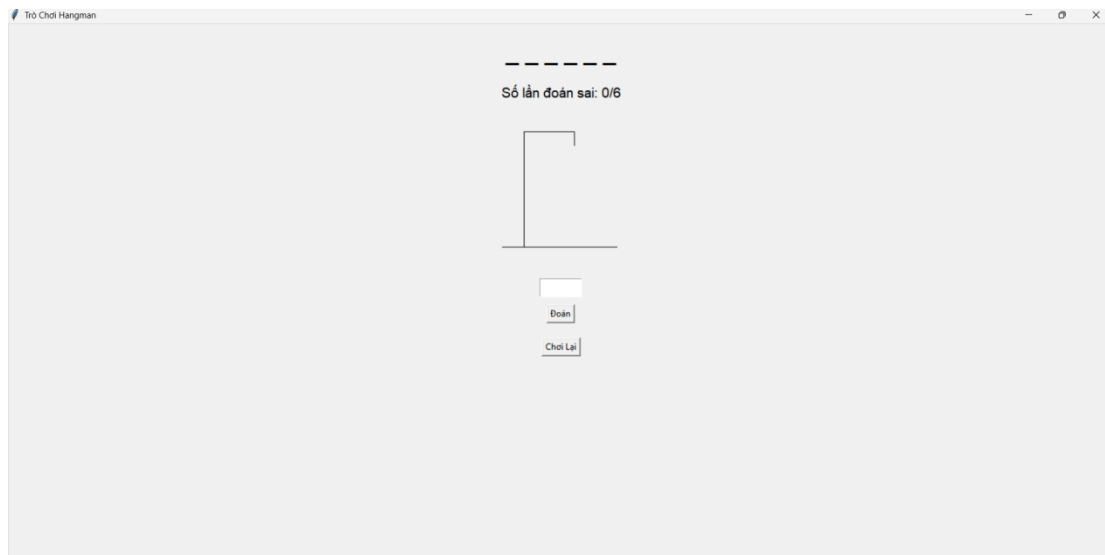
Chương trình chính được tổ chức trong lớp HangmanGame với các phương thức (hàm thành viên) sau:

- `__init__(self, root)`: Phương thức khởi tạo của lớp. Được gọi khi tạo một đối tượng HangmanGame. Nhiệm vụ chính là thiết lập cửa sổ Tkinter, tải từ, khởi tạo trạng thái trò chơi ban đầu, và xây dựng giao diện người dùng bằng cách tạo và đóng gói các widget. Nó cũng liên kết các sự kiện (nhấn nút, nhấn Enter) với các phương thức xử lý.
- `load_words(self)`: Phương thức đọc danh sách từ từ tệp `words.txt`. Chứa logic mở tệp, đọc, xử lý dòng và xử lý các ngoại lệ liên quan đến tệp tin. Trả về danh sách từ hoặc None.
- `reset_game(self)`: Phương thức đặt lại toàn bộ trạng thái của trò chơi để bắt đầu một ván mới. Chọn từ mới, đặt lại số lỗi và các chữ cái đã đoán, cập nhật lại giao diện về trạng thái ban đầu.

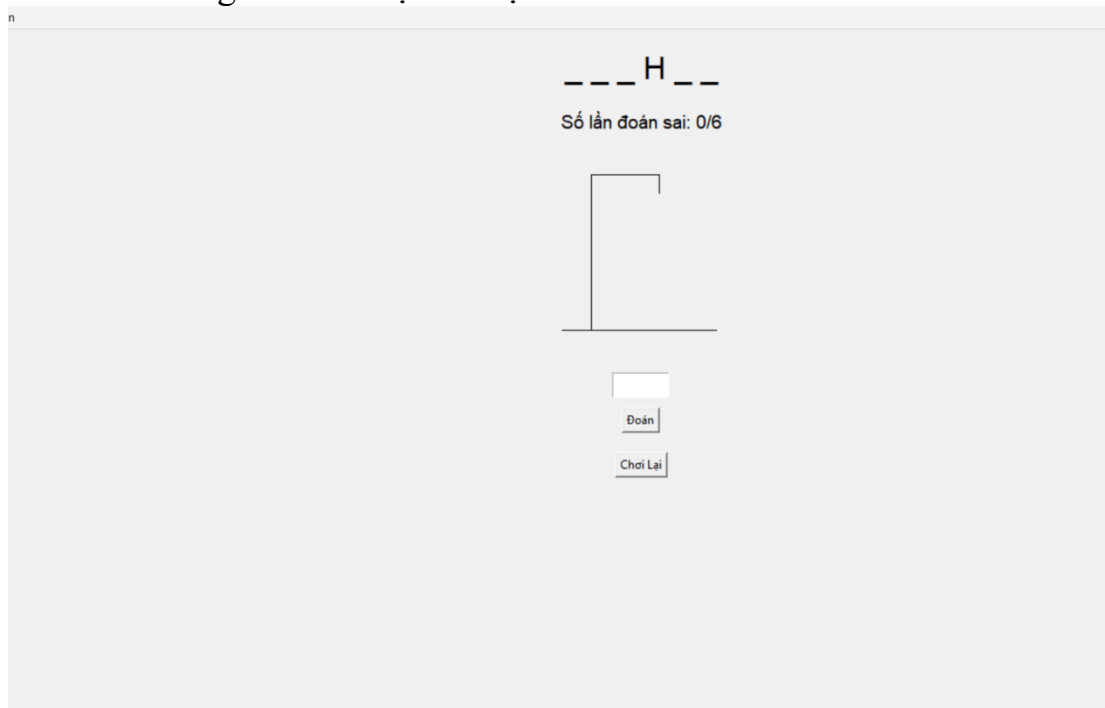
- `display_word(self)`: Phương thức tạo ra chuỗi hiển thị trạng thái hiện tại của từ bí mật, với các chữ cái đã đoán hiện ra và các chữ cái chưa đoán là dấu gạch dưới.
- `draw_hangman(self)`: Phương thức vẽ hình người treo cổ trên canvas. Dựa vào giá trị của `self.errors` để quyết định sẽ vẽ thêm bộ phận nào của người treo cổ.
- `process_guess(self, event=None)`: Phương thức xử lý logic khi người chơi thực hiện một lượt đoán (nhập chữ cái và nhấn Enter hoặc nút "Đoán"). Nó kiểm tra đầu vào, cập nhật trạng thái trò chơi, và kiểm tra điều kiện thắng/thua.
- `if __name__ == "__main__":`: Khối mã này chỉ chạy khi script được thực thi trực tiếp. Nó tạo cửa sổ gốc Tkinter (`root`), tạo một thể hiện của lớp `HangmanGame` (`game`), và bắt đầu vòng lặp sự kiện chính của Tkinter (`root.mainloop()`) để cửa sổ hiển thị và phản hồi tương tác của người dùng.

Chương 4 Thực nghiệm chương trình

4.1. Chương trình đã làm được



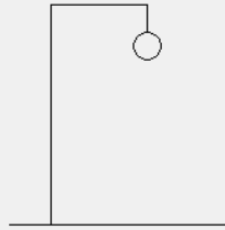
Nếu đoán đúng thì hiển thị các mục có chứa chữ có sẵn



Còn nếu đoán sai thì vẽ 1 đường dưới thanh treo

_ _ _ H _ _

Số lần đoán sai: 1/6



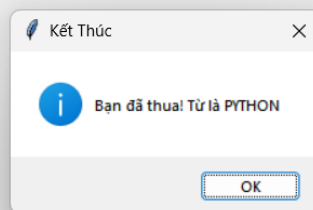
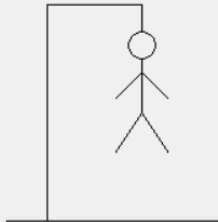
Đoán

Chơi Lại

Cứ thế cứ thế nếu sai hết thì báo đã đoán sai và đây là kết cục

P _ _ H _ _

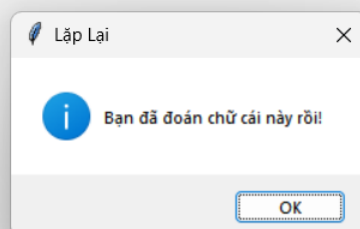
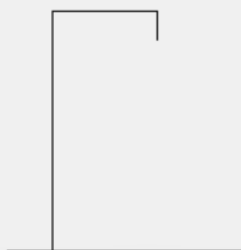
Số lần đoán sai: 6/6



Nếu đoán trùng thì báo đoán trùng

P _ _ _ _ _

Số lần đoán sai: 0/6



4.2. Kết luận

Kết luận và Hướng Phát Triển cho Trò Chơi Hangman

Kết luận

Trò chơi Hangman đã được triển khai thành công với giao diện đơn giản, cho phép người chơi đoán chữ và theo dõi số lần đoán sai (0/6). Hiện tại, trò chơi hoạt động tốt với cơ chế cơ bản, hiển thị trạng thái của người treo cổ và cập nhật số lần đoán sai.

Hướng Phát Triển

Thêm từ điển từ vựng phong phú hơn với các cấp độ khó.

Cải thiện giao diện đồ họa bằng cách sử dụng thư viện như Pygame hoặc Tkinter.

Thêm tính năng lưu điểm cao nhất và hiển thị bảng xếp hạng.

Tích hợp âm thanh khi đoán đúng/sai hoặc khi trò chơi kết thúc.

Cho phép người chơi nhập từ mới để thách thức người khác.