

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 THỰC HIỆN :LÊ TUẤN ANH

MÃ SỐ SINH VIÊN :K225480106001

LỚP :K58KTP

GIÁO VIÊN GIẢNG DẠY :NGUYỄN VĂN HUY

BÀI TẬP KẾT THÚC MÔN HỌC

MÔN HỌC : LẬP TRÌNH PYTHON

BỘ MÔN: CÔNG NGHỆ THÔNG TIN

Sinh Viên: Lê Tuấn Anh

mssv: k225480106001

Lớp: K58KTP

Ngành: Kỹ Thuật Phần Mềm

Giáo Viên Giảng Dạy: Nguyễn Văn Huy

Ngày Giao Đề Tài: 20/05/2025

Ngày Hoàn Thành: 01/06/2025

ĐỀ TÀI: *Game hangman GUI*

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)

THÁI NGUYÊN - 2025

**TRƯỜNG ĐHKTCN
KHOA ĐIỆN TỬ**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

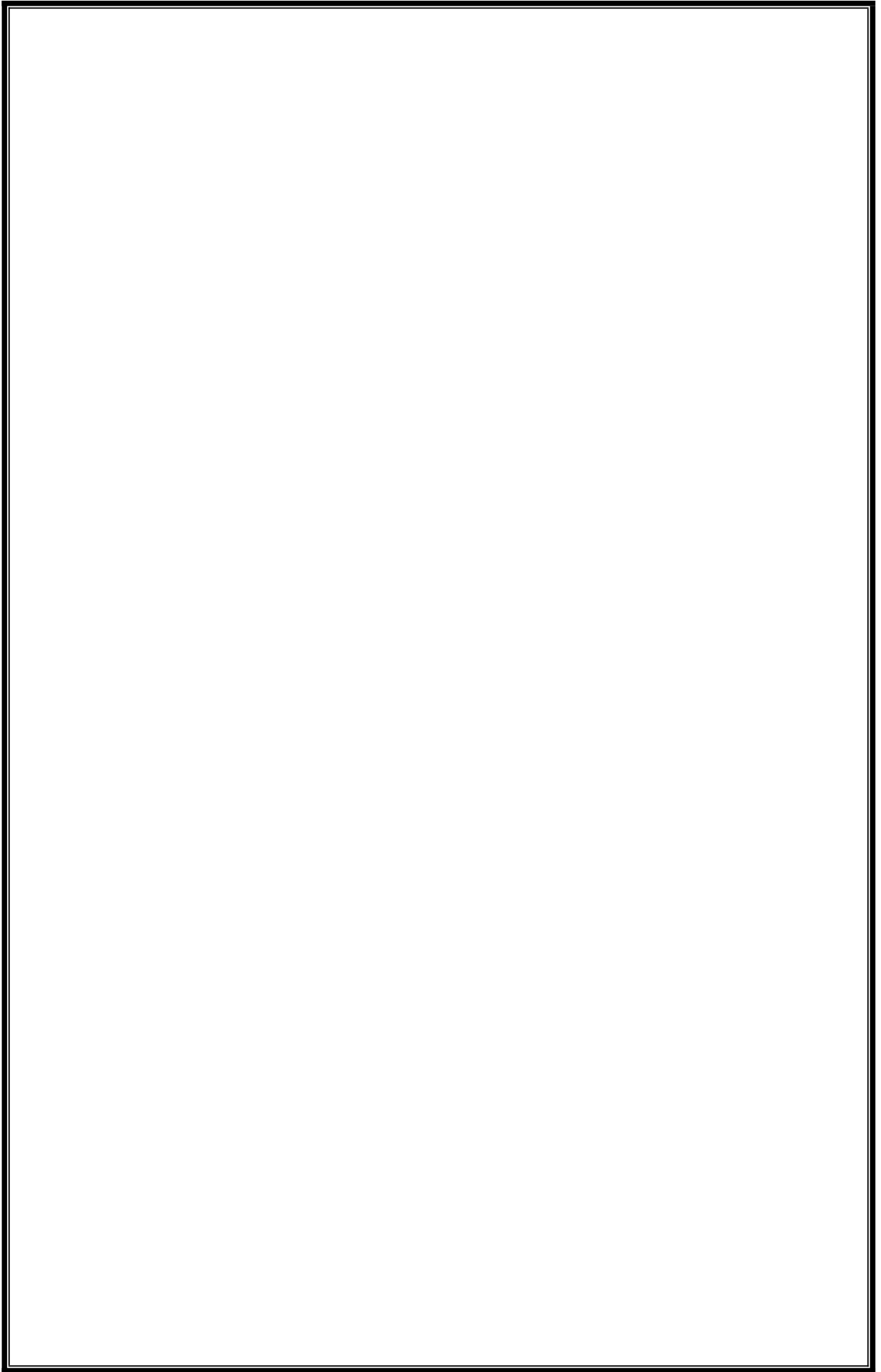
.....

Xếp Loại.....Điểm:.....

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

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)



MỤC LỤC

LỜI CẢM ƠN	1
LỜI CAM ĐOAN	3
DANH MỤC HÌNH ẢNH	4
LỜI MỞ ĐẦU	5
CHƯƠNG 1: GIỚI THIỆU ĐẦU BÀI	6
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	8
2.1 ngôn ngữ lập trình python	8
2.2 thư viện tikinter	10
2.3 cấu trúc dữ liệu cơ bản	10
2.4 lập trình hướng đối tượng(OOP)	11
2.5 xử lý tệp tin và ngoại lệ	12
CHƯƠNG 3: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH	13
3.1 sơ đồ khối hệ thống và biểu đồ phân cấp chức năng	13
3.2 sơ đồ các thuật toán chính	16
3.3 cấu trúc dữ liệu	19
3.4 chương trình	20
CHƯƠNG 4: THỰC NGHIỆM CHƯƠNG TRÌNH	22
4.1 hình ảnh	22
4.2 kết luận	27
TÀI LIỆU THAM KHẢO	29

LỜI CẢM ƠN

Trước tiên, em xin bày tỏ lòng biết ơn sâu sắc đến thầy Nguyễn Văn Huy, người đã tận tình giảng dạy, hướng dẫn và hỗ trợ em trong suốt quá trình học tập và thực hiện bài tập. Những ý kiến đóng góp quý báu và sự tận tâm của Thầy đã giúp em hoàn thành đề tài một cách tốt nhất. Em cũng xin gửi lời cảm ơn đến các Thầy Cô trong khoa Điện Tử và Trường Đại học Kỹ Thuật Công Nghiệp đã tạo điều kiện thuận lợi về kiến thức, cơ sở vật chất và môi trường học tập để em có thể nghiên cứu và hoàn thành bài tập này.

LỜI CAM ĐOAN

Em xin cam đoan rằng đề tài “Game Hangman GUI” này là kết quả học tập và làm việc nghiêm túc của cá nhân em, Lê Tuấn Anh dưới sự hướng dẫn của thầy Nguyễn Văn Huy nội dung báo cáo, các kết quả thực nghiệm, các đoạn mã chương trình và hình ảnh được trình bày trong tài liệu đều do tôi trực tiếp thực hiện. Trong quá trình hoàn thành báo cáo, tuyệt đối không sao chép, đạo văn hay sử dụng trái phép bất kỳ phần nội dung nào từ các tài liệu, sản phẩm của người khác mà không ghi nguồn. Tôi hoàn toàn chịu trách nhiệm trước nhà trường, giảng viên hướng dẫn và pháp luật về tính trung thực và bản quyền của toàn bộ nội dung trong báo cáo này. Nếu phát hiện có bất kỳ hành vi gian lận, sao chép không trung thực nào, tôi xin cam kết chấp nhận mọi hình thức xử lý theo quy định.

SINH VIÊN KÝ TÊN

DANH MỤC HÌNH ẢNH

H1.1 Sơ đồ khối Thuật toán load_words

H1.2 Sơ đồ khối Thuật toán reset_game

H1.3 Sơ đồ khối Thuật toán process_guess

H1.4 Sơ đồ khối Thuật toán draw_hangman

h1.5 hình ảnh khi bắt đầu vào game

h1.6 hình ảnh khi đoán đúng chữ

h1.7 hình ảnh khi chữ cái được đoán trước đó rồi

h1.8 hình ảnh khi đoán sai chữ

h1.9 hình ảnh khi game over

h2.1 hình ảnh khi thắng game

LỜI MỞ ĐẦU

Trong thời đại công nghệ phát triển mạnh mẽ, việc học lập trình không chỉ dừng lại ở việc nắm vững cú pháp ngôn ngữ mà còn cần biết cách vận dụng các kiến thức đó để xây dựng các ứng dụng thực tiễn, có tính tương tác và hấp dẫn. Một trong những phương pháp hiệu quả để rèn luyện tư duy lập trình và kỹ năng xử lý thuật toán là thông qua việc xây dựng các trò chơi đơn giản.

Đồ án này được thực hiện nhằm mục tiêu áp dụng kiến thức lập trình Python và thư viện Tkinter để xây dựng trò chơi Hangman – một trò chơi đoán chữ truyền thống, nơi người chơi cần phán đoán một từ khóa bí mật bằng cách nhập từng chữ cái. Mỗi lần đoán sai, một phần cơ thể của nhân vật sẽ được vẽ lên giá treo. Trò chơi mang tính giáo dục và giải trí cao, đồng thời tạo cơ hội để sinh viên tiếp cận với các khái niệm quan trọng như quản lý giao diện người dùng (GUI), xử lý tệp dữ liệu, quản lý trạng thái chương trình và kiểm soát luồng điều kiện trong Python.

CHƯƠNG 1: GIỚI THIỆU ĐẦU BÀI

Trò chơi đoán chữ (Hangman) là một trò chơi trí tuệ đơn giản nhưng hấp dẫn, giúp người chơi rèn luyện khả năng tư duy và vốn từ vựng. Trong trò chơi này, người chơi cần đoán một từ bí mật bằng cách nhập từng chữ cái. Mỗi lần đoán sai sẽ khiến nhân vật "bị treo cổ" dần theo từng bước, và trò chơi kết thúc khi người chơi đoán đúng toàn bộ từ hoặc phạm sai quá số lần cho phép.

Đề tài này tập trung vào việc xây dựng một ứng dụng trò chơi đoán chữ sử dụng ngôn ngữ lập trình Python với giao diện đồ họa người dùng (GUI) bằng thư viện tkinter. Trò chơi được thiết kế với các chức năng cơ bản như: khởi tạo trò chơi, xử lý lượt đoán, thiết lập lại trò chơi, và kết thúc trò chơi. Danh sách từ sẽ được nạp từ một tệp văn bản, đảm bảo tính linh hoạt và mở rộng.

Thông qua bài toán này, người học không chỉ thực hành các kiến thức lập trình Python như xử lý chuỗi, quản lý trạng thái, và quản lý lỗi, mà còn học được cách xây dựng giao diện trực quan và thân thiện với người dùng.

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.

Tóm tắt chương 1

Trình bày tổng quan về trò chơi Hangman, mục tiêu của đề tài và các tính năng chính như: đọc từ tệp, xử lý ngoại lệ, cập nhật giao diện, xử lý tương tác người dùng và hiển thị kết quả. Đồng thời nêu rõ những thách thức trong quá

trình phát triển và các kiến thức lập trình Python, xử lý tệp và GUI đã được áp dụng để xây dựng chương trình.

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 Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình thông dịch, đa mục đích, hướng đối tượng và bậc cao. Với cú pháp rõ ràng, ngắn gọn, Python rất phổ biến trong giáo dục cũng như thực tiễn phát triển phần mềm.

Trong dự án trò chơi Hangman, Python được lựa chọn vì các lý do sau:

Cú pháp đơn giản, dễ học:

Python sử dụng cú pháp tự nhiên giống tiếng Anh, giúp người mới học nhanh chóng tiếp cận. Ví dụ, không cần dấu ngoặc nhọn `{}` để xác định khối lệnh, thay vào đó là thụt đầu dòng rõ ràng.

Hỗ trợ đa dạng kiểu dữ liệu:

Python cung cấp nhiều kiểu dữ liệu linh hoạt như str (chuỗi), list (danh sách), dict (từ điển), rất hữu ích khi thao tác với danh sách từ, các chữ cái đã đoán, trạng thái của trò chơi,...

Thư viện tiêu chuẩn phong phú:

Trong trò chơi này, Python sử dụng thư viện **Tkinter** để tạo giao diện đồ họa GUI. Đây là thư viện GUI mặc định của Python, không cần cài đặt thêm, dễ sử dụng và đủ đáp ứng các yêu cầu cơ bản của trò chơi.

Hỗ trợ lập trình hướng đối tượng và hướng hàm:

Python hỗ trợ cả hai phong cách lập trình:

Trong đề tài này, chủ yếu áp dụng **lập trình hướng hàm**: chia bài toán thành các hàm xử lý riêng như `load_words()`, `reset_game()`, `process_guess()`, v.v.

Nếu mở rộng thêm, có thể sử dụng lập trình hướng đối tượng để quản lý tốt hơn trạng thái trò chơi, người chơi, v.v.

Khả năng xử lý tệp mạnh mẽ:

Python cung cấp các hàm tích hợp sẵn để đọc, ghi và xử lý tệp dễ dàng. Trong bài này, chương trình đọc danh sách từ từ một tệp `words.txt` và xử lý để làm nguồn dữ liệu cho trò chơi.

2.2. 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).

2.3. 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()).

2.4. 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.

2.5. 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.

Tóm tắt chương 2

trình bày các kiến thức nền tảng phục vụ cho việc xây dựng trò chơi Hangman bằng ngôn ngữ lập trình Python. Python được lựa chọn nhờ cú pháp đơn giản, dễ đọc, dễ học và khả năng hỗ trợ tốt cho lập trình hướng hàm lẫn hướng đối tượng. Trong chương trình, các kiểu dữ liệu như list (danh sách), set (tập hợp) và string (chuỗi) được sử dụng để quản lý từ khóa, chữ cái đã đoán và trạng thái trò chơi. Thư viện Tkinter – thư viện GUI tiêu chuẩn của Python – được dùng để tạo giao diện đồ họa gồm cửa sổ trò chơi, vùng vẽ hình, ô nhập liệu, nhãn hiển thị và các nút điều khiển. Việc tổ chức chương trình theo mô hình hướng đối tượng với lớp HangmanGame giúp mã

nguồn rõ ràng, dễ mở rộng và bảo trì. Ngoài ra, chương trình cũng vận dụng cơ chế xử lý ngoại lệ để đảm bảo hoạt động ổn định, đặc biệt là khi thao tác với tệp văn bản chứa danh sách từ (words.txt). Tất cả các yếu tố trên tạo thành cơ sở lý thuyết vững chắc để triển khai một ứng dụng trò chơi Hangman có giao diện thân thiện, tương tác linh hoạt và xử lý lỗi hiệu quả.

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):

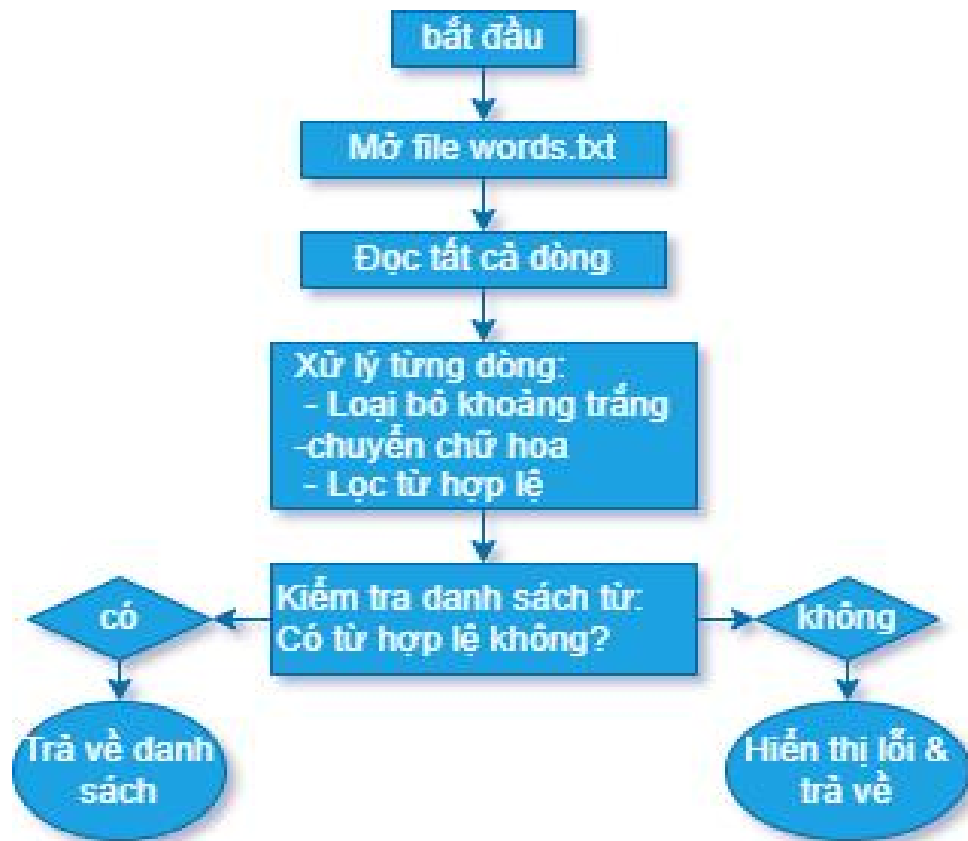
Các chức năng chính của chương trình được phân cấp như sau:

- **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 Lướt 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)
 - Hiển thị kết quả (thắng/thua)
 - Vô hiệu hóa nhập liệu/nút đoán
- **Hiển thị (Display)**
 - Hiển thị trạng thái từ (Display Word State)
 - Hiển thị số lần sai (Display Error Count)
 - Vẽ hình người treo cổ (Draw Hangman)
 - Hiển 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)

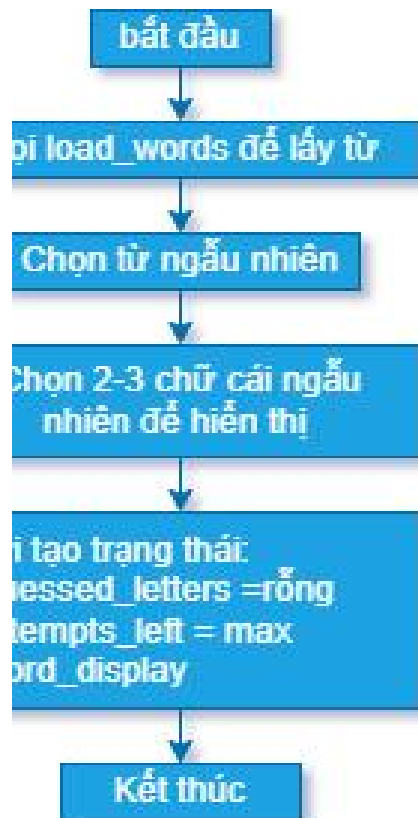
3.2. Sơ đồ khối các thuật toán chính

Thuật toán load_words (đọc danh sách từ)



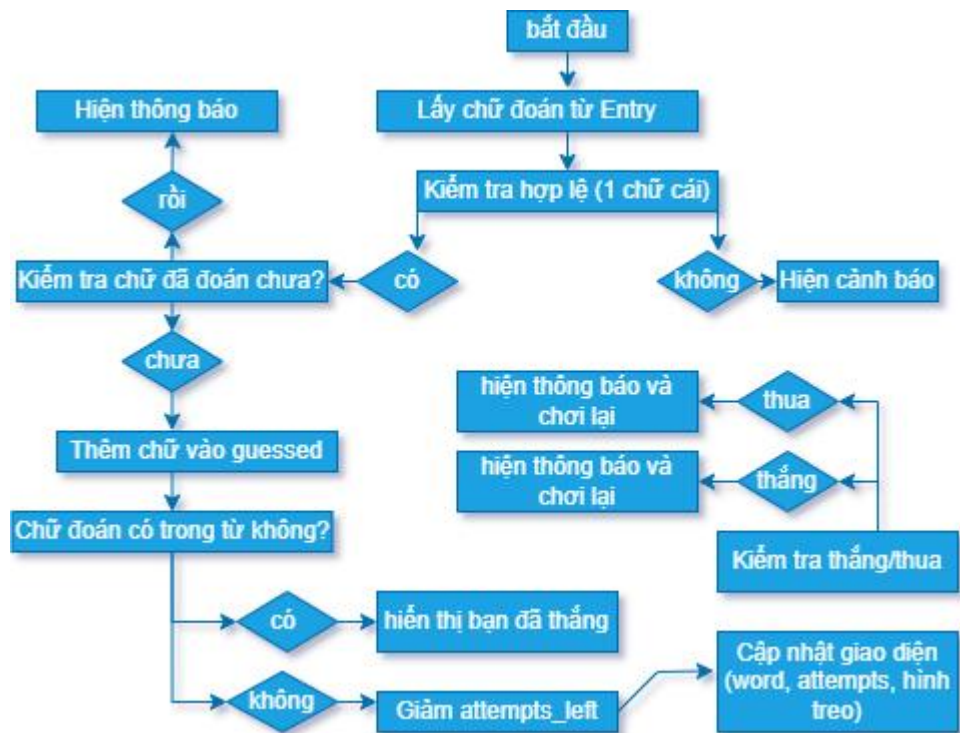
H1.1 Sơ đồ khối Thuật toán load_words

Thuật toán reset_game (khởi tạo lại game)



H1.2 Sơ đồ khối Thuật toán `reset_game`

Thuật toán process_guess (xử lý đoán chữ)



H1.3 Sơ đồ khối Thuật toán process_guess

Thuật toán draw_hangman (vẽ hình treo cổ)

H1.4 Sơ đồ khối Thuật toán draw_hangman

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.

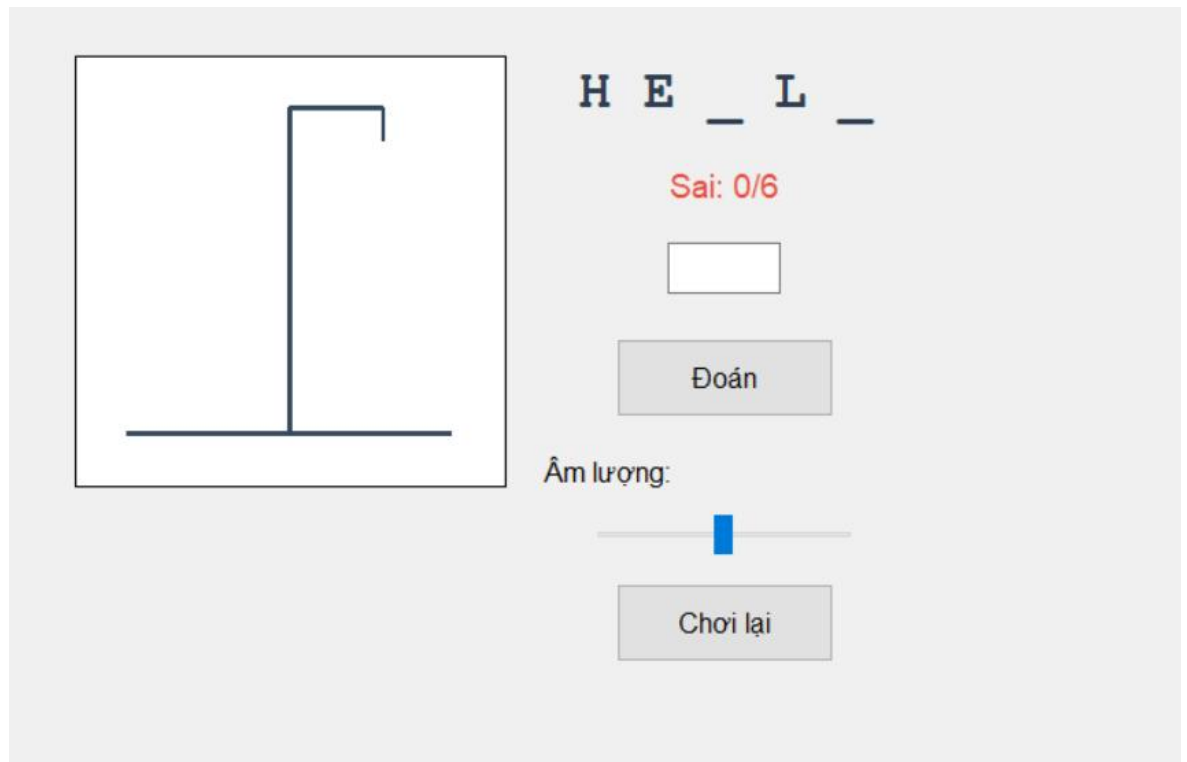
Tóm tắt chương 3

Mô tả chi tiết cấu trúc hệ thống, các sơ đồ chức năng và thuật toán chính như: `load_words`, `reset_game`, `process_guess`, `draw_hangman`. Chương này cũng

làm rõ vai trò của từng thuộc tính và phương thức trong lớp HangmanGame, từ khởi tạo giao diện đến cập nhật trạng thái trò chơi.

CHƯƠNG 4: THỰC NGHIỆM TRƯỜNG TRÌNH

Đầu tiên chúng ta sẽ thấy một giao diện gồm có: cột treo, nút tăng giảm âm lượng, các từ khóa bị ẩn, nút đoán chữ và nút chơi lại.



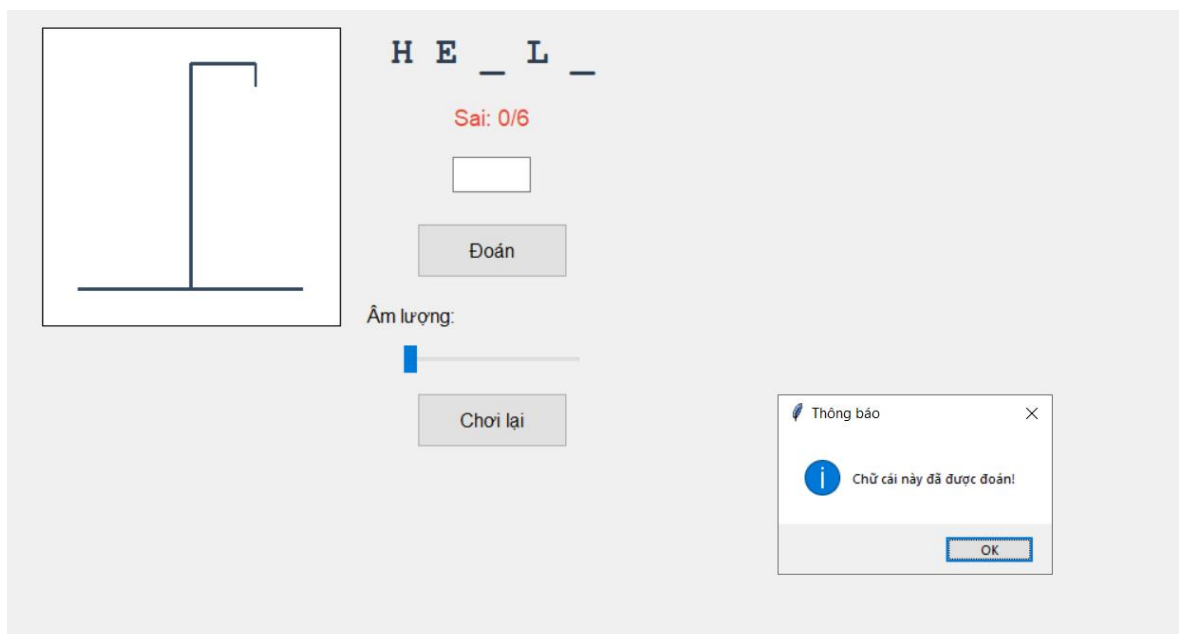
h1.5 hình ảnh khi bắt đầu vào game

Khi chúng ta đoán chữ mà chữ đó có trong từ khóa ở file file words.txt thì chữ đó sẽ được hiển thị lên



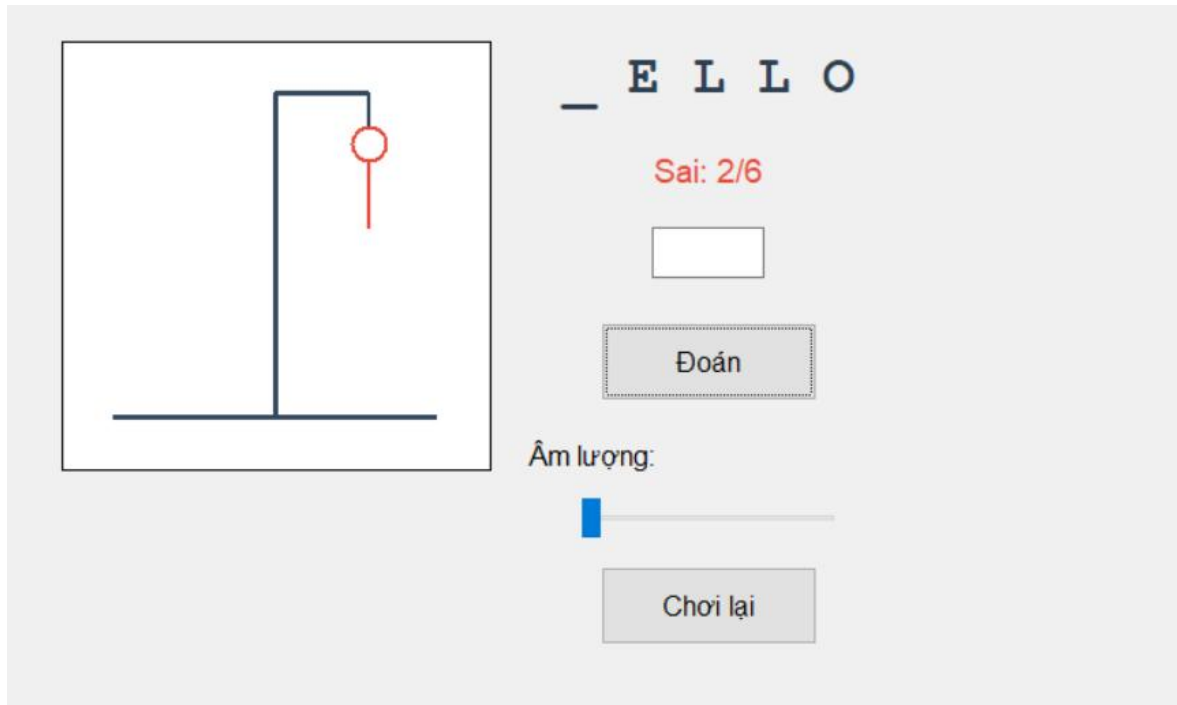
h1.6 hình ảnh khi đoán đúng chữ

Khi chúng ta đoán chữ mà chữ đó đã được đoán ngay trước đó rồi thì hệ thống sẽ báo chữ này đã được đoán



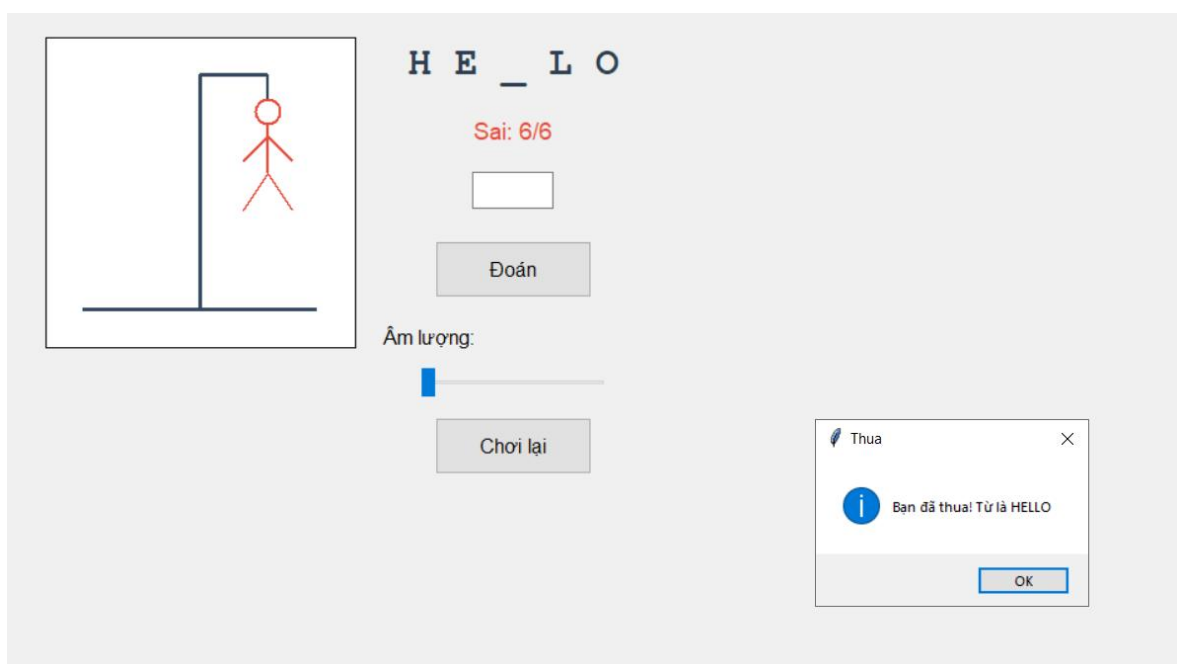
h1.7 hình ảnh khi chữ cái được đoán trước đó rồi

Nếu chúng ta đoán chữ mà không đúng với từ trong từ khóa thì số lần sai sẽ tăng lên 1, và tại vị trí cột treo sẽ hiển thị người treo(người treo sẽ hiển thị từng bộ phận theo số lần đoán sai và số lần đoán sai tối đa là 6 lần)



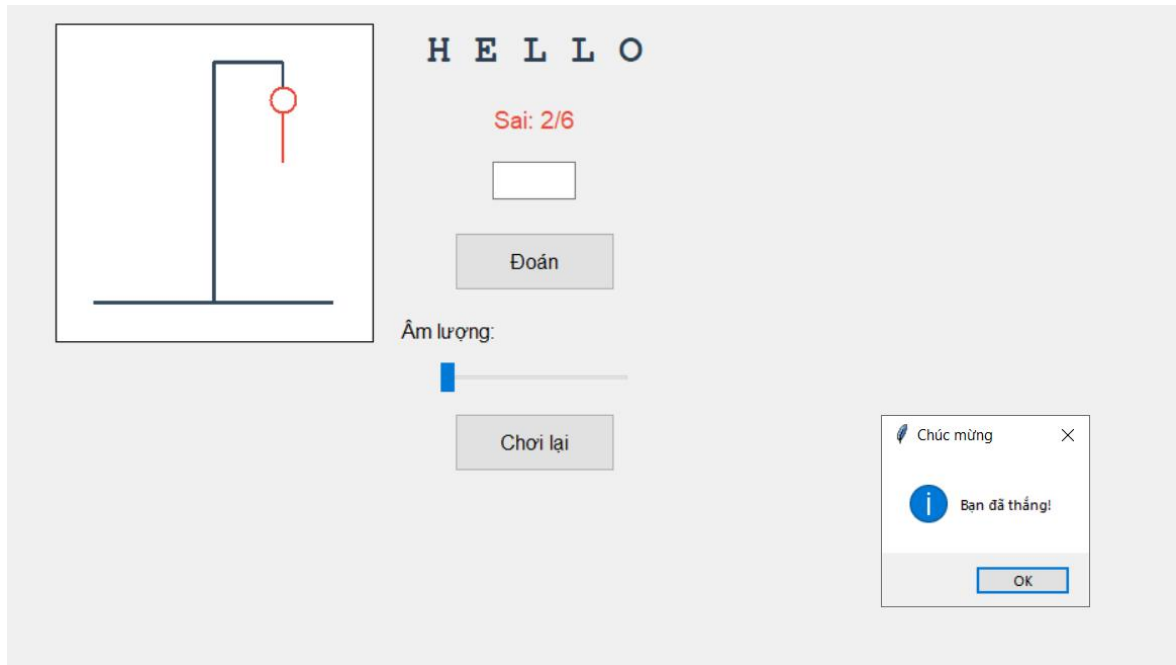
h1.8 hình ảnh khi đoán sai chữ

Khi số lần đoán sai vượt mức 6 lần có nghĩa là bạn đã thua và hệ thống sẽ hiển thị ra từ khóa chính xác



h1.9 hình ảnh khi game over

Khi số lần sai nhỏ hơn 6 và các từ đã được đoán chính xác thì chúng ta sẽ thắng



h2.1 hình ảnh khi thắng game

4.2. Kết luận

Kết luận và Hướng Phát Triển cho Trò Chơi Hangman

Kết luận

Qua quá trình thực hiện đề tài “**Xây dựng trò chơi Hangman sử dụng Python và Tkinter**”, nhóm đã có cơ hội áp dụng và củng cố kiến thức về lập trình Python, giao diện đồ họa (GUI) với Tkinter, cũng như cách tổ chức chương trình theo hướng lập trình hướng đối tượng.

Chương trình mô phỏng trò chơi Hangman với giao diện thân thiện, dễ sử dụng, bao gồm các chức năng cơ bản như: khởi tạo trò chơi, hiển thị từ cần đoán, nhận và xử lý chữ cái người chơi nhập vào, vẽ hình người treo cổ tương ứng với số lần đoán sai, và kết thúc trò chơi khi thắng hoặc thua.

Trong quá trình thực hiện, nhóm cũng gặp một số thách thức như xử lý lỗi khi đọc tệp dữ liệu, cập nhật giao diện theo trạng thái trò chơi, và đảm bảo tính tương tác người dùng một cách mượt mà. Tuy nhiên, bằng việc vận dụng linh hoạt kiến thức về cấu trúc dữ liệu, xử lý chuỗi, thao tác tệp tin và lập trình GUI, nhóm đã hoàn thiện chương trình với đầy đủ chức năng theo yêu cầu.

Đề tài không chỉ giúp nhóm rèn luyện kỹ năng lập trình mà còn nâng cao khả năng làm việc nhóm, tư duy phân tích và giải quyết vấn đề. Trong tương lai, trò chơi có thể được mở rộng thêm với nhiều cấp độ khó, từ vựng phong phú hơn hoặc giao diện sinh động hơn nhằm nâng cao trải nghiệm người dùng.

Hướng Phát Triển

Để nâng cao chất lượng và tính hấp dẫn của trò chơi Hangman, chương trình có thể được mở rộng và cải tiến theo các hướng sau:

Tăng độ phong phú của dữ liệu từ vựng

Mở rộng nguồn dữ liệu từ đơn giản sang nâng cao, phân loại từ theo chủ đề (động vật, nghề nghiệp, đồ vật,...).

Kết nối với API hoặc cơ sở dữ liệu online để cập nhật danh sách từ mới thường xuyên.

Phân cấp độ khó cho trò chơi

Cung cấp các cấp độ như dễ, trung bình, khó bằng cách điều chỉnh độ dài từ hoặc độ phổ biến.

Thêm tính năng chọn cấp độ khi bắt đầu trò chơi.

Giao diện đồ họa nâng cao

Cải tiến GUI với hình ảnh minh họa sinh động thay vì vẽ đơn giản bằng canvas.

Tùy biến giao diện theo phong cách người dùng, bổ sung hiệu ứng âm thanh, hoạt ảnh.

Lưu điểm và thống kê

Thêm tính năng lưu điểm số, số ván thắng/thua.

Hiển thị bảng xếp hạng hoặc thống kê kết quả theo người chơi.

Hỗ trợ nhiều người chơi hoặc chế độ thi đấu

Cho phép 2 người chơi luân phiên đoán từ.

Phát triển chế độ chơi theo lượt hoặc đối kháng qua mạng nội bộ hoặc online.

Tương thích đa nền tảng

Chuyển đổi chương trình sang dạng ứng dụng desktop độc lập (dùng PyInstaller) hoặc phát triển trên nền tảng web/mobile bằng công nghệ phù hợp như Flask hoặc Kivy.

Tóm tắt chương 4

Trình bày kết quả thực nghiệm chương trình thông qua các tình huống cụ thể như: bắt đầu trò chơi, đoán đúng, đoán sai, xử lý khi đoán trùng, thông báo khi thắng/thua. Giao diện đồ họa được minh họa bằng hình ảnh kèm mô tả chức năng của từng thành phần.

TÀI LIỆU THAM KHẢO

Python Software Foundation. (2023). *The Python Language Reference*. Truy cập từ: <https://docs.python.org/3/>

Tkinter Documentation. (2023). *Tkinter — Python interface to Tcl/Tk*. Truy cập từ: <https://docs.python.org/3/library/tkinter.html>

Pygame Community. (2023). *Pygame Documentation*. Truy cập từ: <https://www.pygame.org/docs/>

Weng, Y. (2022). *Learn to Program with Python*. No Starch Press.

Sweigart, A. (2020). *Invent Your Own Computer Games with Python* (4th Edition). No Starch Press.

TutorialsPoint. (2023). *Python GUI Programming with Tkinter*. Truy cập từ: https://www.tutorialspoint.com/python/python_gui_programming.htm

