

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÀI TẬP KẾT THÚC MÔN HỌC
MÔN: LẬP TRÌNH PYTHON

NGÀNH : KỸ THUẬT MÁY TÍNH

HỆ : ĐẠI HỌC CHÍNH QUY

THÁI NGUYỄN – 2025

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

BỘ MÔN CÔNG NGHỆ THÔNG TIN



**BÀI TẬP KẾT THÚC MÔN HỌC MÔN:
LẬP TRÌNH PYTHON**

GIÁO VIÊN HƯỚNG DẪN : TS. Nguyễn Văn Huy

HỌ TÊN SINH VIÊN : Ngụy Đình Tuấn Hà

MSSV : K225480106011

LỚP : K58KTP

LINK GITHUB: <https://github.com/Tuanha0302/BaitaplonPYTHON>

LINK VIDEO: <https://youtu.be/WNngSLokbSo>

Thái nguyên – 2025

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: Ngụy Đình Tuấn Hà **MSSV:** K225480106011

Lớp: K58KTP **Ngành:** Kỹ Thuật Máy Tính

Giáo viên hướng dẫn: TS. Nguyễn Văn Huy

Ngày giao nhiệm vụ: 19/05/2025 **Ngày hoàn thành nhiệm vụ:** 07/06/2025

Tên đề tài: *Máy tính đơn giản (Simple Calculator GUI)*

Yêu cầu: Viết chương trình máy tính có giao diện GUI cho phép người dùng nhập hai số và chọn phép toán +, -, ×, ÷ để tính toán.

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

TRƯỜNG ĐHKTCN CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN TỬ

Độc lập - Tự do - Hạnh phúc

PHIẾU GHI ĐIỂM

Sinh viên: Nguyễn Đình Tuấn Hà

Lớp: K58KTP

GVHD: TS. Nguyễn Văn Huy

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

[illegible]

Xếp loại : Điểm :

Thái Nguyên, ngày tháng năm 2025.

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

MỤC LỤC

LỜI CAM ĐOAN.....	4
DANH MỤC BẢNG VẼ VÀ ĐỒ THỊ.....	5
CHƯƠNG I: GIỚI THIỆU ĐẦU BÀI.....	6
1.1. Tên đề tài.....	6
1.2. Mô tả bài toán	6
1.3. Mục tiêu của chương trình.....	6
1.4. Thách thức và giải pháp.....	6
CHƯƠNG II: CƠ SỞ LÝ THUYẾT.....	8
2.1. Lập trình Python	8
2.2. Thư viện Tkinter	8
2.3. Giao diện người dùng	9
2.4. Bắt lỗi bằng try – except.....	10
2.5. Vẽ đồ họa bằng Canvas	11
CHƯƠNG III: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH....	12
3.1. Sơ đồ khối hệ thống	12
3.1.1. Các module chính của chương trình	12
3.1.2. Biểu đồ phân cấp chức năng	12
3.2. Sơ đồ khối các thuật toán chính.....	13
3.2.1. Thuật toán đoán chữ.....	13
3.2.2. Thuật toán chơi lại	14
3.3. Cấu trúc dữ liệu.....	14
3.4. Chương trình	15
CHƯƠNG 4. THỰC NGHIỆM VÀ KẾT LUẬN	16
4.1. Thực nghiệm	16
4.1.1. Ghi lại kết quả các tính năng	16
4.1.2. Mô tả các tính năng.....	19
4.2. Kết luận	21
4.2.1. Những điều học được trong quá trình thực hiện.....	21
4.2.2. Hướng cải tiến trong tương lai.....	21
TÀI LIỆU THAM KHẢO.....	22

LỜI CAM ĐOAN

Em xin cam đoan rằng bài tập kết thúc môn Lập trình Python do em tự thực hiện và là kết quả của riêng em dưới sự hướng dẫn của thầy Nguyễn Văn Huy. Các nội dung, số liệu và kết quả trình bày trong bài là trung thực, không sao chép từ bất kỳ nguồn nào một cách trái phép và không được trích dẫn rõ ràng.

Em cam kết mọi thông tin, số liệu và kết quả trong báo cáo là chính xác và trung thực. Nếu có bất kỳ sai phạm nào trong quá trình thực hiện, em xin hoàn toàn chịu trách nhiệm trước nhà trường và giảng viên phụ trách.

Em xin chân thành cảm ơn!

Sinh viên cam đoan

Nguyễn Đình Tuấn Hà

DANH MỤC BẢNG VẼ VÀ ĐỒ THỊ

Hình 3.1: Biểu đồ phân cấp chức năng	12
Hình 3.2: Sơ đồ khối thuật toán đoán chữ.....	13
Hình 3.3: Sơ đồ khối thuật toán chơi lại	14
Hình 4.1: Kết quả chữ thường biến thành chữ hoa	16
Hình 4.2: Kết quả nhập chữ in hoa.....	17
Hình 4.3: Kết quả nhập số bị báo lỗi khi nhập không đúng.....	17
Hình 4.4: Kết quả nhập quá chữ báo lỗi nhập quá.....	17
Hình 4.5: Kết quả đoán đúng chữ	18
Hình 4.6: Kết quả đoán sai chữ.....	18
Hình 4.7: Code giao diện đồ họa GUI.....	19
Hình 4.8: Code xử lý logic của trò chơi.....	20
Hình 4.9: Code xử lý logic của trò chơi.....	20
Hình 4.10: Tạo giao diện của trò chơi.....	21
Hình 4.11: Thông báo thắng/thua.....	21

CHƯƠNG I: GIỚI THIỆU ĐẦU BÀI

1.1. Tên đề tài

- Trò chơi đoán chữ (Game Hangman GUI)

1.2. Mô tả bài toán

Bài toán yêu cầu xây dựng một trò chơi Hangman (Treo cổ) có giao diện người dùng (GUI) bằng thư viện Tkinter trong Python. Trò chơi chọn ngẫu nhiên một từ từ file words.txt, người chơi đoán từng chữ cái để tìm ra từ. Mỗi lần đoán sai sẽ được tính và hình ảnh người treo cổ sẽ dần được vẽ. Trò chơi kết thúc khi người chơi đoán đúng toàn bộ từ hoặc sai quá 6 lần. Chương trình cần xử lý ngoại lệ khi file không tồn tại, và phải cập nhật giao diện một cách linh hoạt sau mỗi thao tác.

1.3. Mục tiêu của chương trình

Các mục tiêu:

- Tạo giao diện đồ họa trực quan, dễ sử dụng cho trò chơi Hangman.
- Đọc từ ngẫu nhiên từ file words.txt và hiển thị từ ở dạng ẩn (ví dụ: _ _ _ _ _).
- Hỗ trợ người chơi nhập từng chữ cái để đoán.
- Hiển thị số lần đoán sai và vẽ hình người treo cổ từng bước.
- Hiển thị thông báo thắng/thua sau khi kết thúc trò chơi.
- Cho phép chơi lại bằng một nút bấm “Chơi lại”.

1.4. Thách thức và giải pháp

Thách thức chính là làm cho giao diện cập nhật linh hoạt theo trạng thái trò chơi, vẽ hình người treo cổ theo từng bước, và xử lý các trường hợp ngoại lệ như: người chơi nhập trùng chữ, nhập nhiều ký tự cùng lúc, hoặc file words.txt bị thiếu.

Giải pháp tiếp cận:

- Sử dụng cấu trúc class để quản lý toàn bộ trạng thái trò chơi: từ bí mật, các chữ đã đoán, số lần sai, trạng thái thắng/thua,...
- Dùng try-except để xử lý lỗi khi đọc file.
- Cập nhật giao diện bằng cách sử dụng Label, Canvas, và messagebox.
- Tạo chức năng "Chơi lại" giúp người chơi khởi động trò chơi mới dễ dàng.

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

2.1. Lập trình Python

Python là một ngôn ngữ lập trình bậc cao, có cú pháp đơn giản, dễ học và rất phù hợp với người mới bắt đầu. Python hỗ trợ lập trình hướng đối tượng, xử lý ngoại lệ, và đặc biệt có sẵn nhiều thư viện mạnh mẽ để xây dựng các ứng dụng thực tế, bao gồm cả ứng dụng có giao diện người dùng (GUI).

Các kiến thức Python được vận dụng trong chương trình bao gồm:

- **Biến và kiểu dữ liệu:** Sử dụng các kiểu dữ liệu như chuỗi (str), danh sách (list), tập hợp (set) và số nguyên (int) để lưu trữ từ khóa, gợi ý, các chữ cái đã đoán và số lần đoán sai.
- **Hàm:** Chương trình được tổ chức thành nhiều hàm nhỏ nhằm thực hiện các chức năng riêng biệt như: khởi động lại từ mới, xử lý đoán chữ, cập nhật giao diện, vẽ hình nhân, kiểm tra trạng thái thắng/thua...
- **Cấu trúc điều kiện (if, elif, else):** Được sử dụng để xác định kết quả đoán chữ, xử lý khi người chơi đoán đúng, đoán sai, hoặc nhập dữ liệu không hợp lệ.
- **Xử lý ngoại lệ (try – except):** Dùng để xử lý lỗi khi đọc dữ liệu từ các file văn bản chứa danh sách từ khóa và gợi ý. Điều này giúp chương trình không bị gián đoạn nếu xảy ra lỗi và vẫn thông báo rõ ràng cho người dùng.

Lựa chọn ngẫu nhiên: Sử dụng thư viện random để chọn ngẫu nhiên một từ khóa và gợi ý cho mỗi ván chơi mới, tạo sự đa dạng và tăng tính hấp dẫn cho trò chơi.

2.2. Thư viện Tkinter

Tkinter là thư viện tiêu chuẩn được tích hợp sẵn trong Python, dùng để xây dựng giao diện người dùng (GUI). Trong chương trình **Trò chơi đoán chữ**,

tkinter được sử dụng để tạo ra các thành phần tương tác giữa người chơi và trò chơi, mang lại trải nghiệm trực quan, sinh động hơn so với giao diện dòng lệnh.

Một số widget chính của tkinter được sử dụng:

- **Entry:** Là nơi người chơi nhập vào chữ cái muốn đoán. Chỉ một ký tự hợp lệ được chấp nhận tại mỗi lượt đoán.
- **Label:** Dùng để hiển thị các thông tin như: từ đang đoán (dưới dạng gạch chân và chữ cái đã đoán đúng), gợi ý cho từ, và số lần đoán sai. Các nhãn này được định dạng rõ ràng để người dùng dễ quan sát.
- **Button:** có 2 nút chính là “Đoán” xử lý chữ cái người chơi nhập vào và “Chơi lại” khởi động lại trò chơi với từ mới và gợi ý mới.
- **Frame:** Dùng để nhóm các thành phần liên quan lại với nhau như ô nhập chữ và nút đoán, giúp bố cục gọn gàng, dễ nhìn.
- **Messagebox:** Dùng để hiển thị các thông báo như: cảnh báo nhập sai, thông báo chiến thắng, thua cuộc hoặc lỗi đọc dữ liệu từ file.
- **Bind:** Cho phép người chơi nhấn phím Enter để đoán thay vì chỉ nhấp chuột vào nút, tăng tính tiện lợi và linh hoạt trong thao tác.
- **Tkinter:** Cho phép bố trí các thành phần theo layout grid(), giúp dễ dàng tổ chức giao diện theo hàng – cột.
- **Canvas:** Đây là một phần đặc biệt, nơi vẽ hình giá treo cổ và người treo dần dần theo số lần đoán sai. Mỗi phần cơ thể (đầu, thân, tay, chân) được vẽ lần lượt tương ứng với các bước sai của người chơi. Điều này tạo hiệu ứng thị giác hấp dẫn và góp phần nâng cao sự căng thẳng trong mỗi lượt chơi.

2.3. Giao diện người dùng

Trò chơi sử dụng **Tkinter** để xây dựng giao diện. Người chơi chỉ cần nhập một chữ cái và nhấn "Đoán", giao diện sẽ cập nhật từ đang đoán, số lần sai và vẽ dần người treo cổ nếu đoán sai.

Các thành phần chính gồm:

- Nhấn hiển thị từ đang đoán (dạng chữ và gạch dưới).
- Nhấn gợi ý giúp người chơi định hướng suy nghĩ.
- Ô nhập chữ cái và nút “Đoán”.
- Vùng vẽ hình người treo cổ.
- Nút “Chơi lại” để bắt đầu ván mới.

2.4. Bắt lỗi bằng try – except

Trong trò chơi Treo Cổ, dữ liệu đầu vào rất quan trọng vì người chơi phải nhập từng ký tự để đoán từ khóa. Để đảm bảo chương trình không bị dừng khi người chơi nhập sai (ví dụ nhập nhiều ký tự hoặc ký tự không phải chữ cái), chương trình sử dụng khối **try – except** để bắt và xử lý lỗi một cách linh hoạt.

Cụ thể, chương trình có thể gặp các tình huống lỗi như sau:

- Người chơi nhập số hoặc ký tự đặc biệt (ví dụ: 1, @, ab) thay vì một chữ cái.
- File từ hoặc gợi ý (words.txt, cauhoi.txt) không tồn tại, rỗng, hoặc có số dòng không khớp nhau.

Để xử lý những lỗi này, chương trình sử dụng cơ chế try – except kết hợp với **hộp thoại cảnh báo (messagebox)** để:

- **Hiển thị thông báo lỗi rõ ràng**, giúp người chơi biết mình đã thao tác sai.
- **Hướng dẫn nhập lại đúng cách**, đảm bảo trải nghiệm không bị gián đoạn.
- **Ngăn chương trình dừng đột ngột**, giúp game luôn ổn định và thân thiện với người dùng.

Việc xử lý lỗi thông minh như vậy góp phần làm cho trò chơi Treo Cổ trở nên chuyên nghiệp và dễ sử dụng hơn.

2.5. Vẽ đồ họa bằng Canvas

Trong trò chơi Đoán chữ, để giúp người chơi dễ hình dung số lần đoán sai, em đã sử dụng một công cụ của Tkinter có tên là **Canvas**. Đây là khu vực cho phép em **vẽ các hình ảnh đơn giản**, và em tận dụng điều đó để dựng nên từng phần của "người bị treo cổ" mỗi khi người chơi đoán sai.

Tối đa số lần sai trong bài của là 6 lần nếu quá 6 lần thì sẽ hiển thị thông báo thua cuộc và hiển thị chữ cần đoán, còn nếu nhỏ hơn 6 lần thì hệ thống sẽ báo thắng và người chơi có thể chơi hoặc chơi lại.

Việc làm này không chỉ giúp **tăng tính trực quan**, mà còn khiến trò chơi trở nên **sinh động và hấp dẫn hơn**. Em đã dùng các hàm như `create_line()` để vẽ các đoạn thẳng (ví dụ như tay, chân) và `create_oval()` để vẽ hình tròn (đầu người). Hình ảnh hiện lên từng bước một.

Qua việc sử dụng Canvas, trò chơi không chỉ đơn thuần là nhập chữ và hiện chữ, mà còn mang lại cảm giác như một trò chơi thật sự có yếu tố đồ họa. Đây cũng là một cách em học thêm về lập trình giao diện và xử lý sự kiện trong Python.

CHƯƠNG III: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

3.1. Sơ đồ khối hệ thống

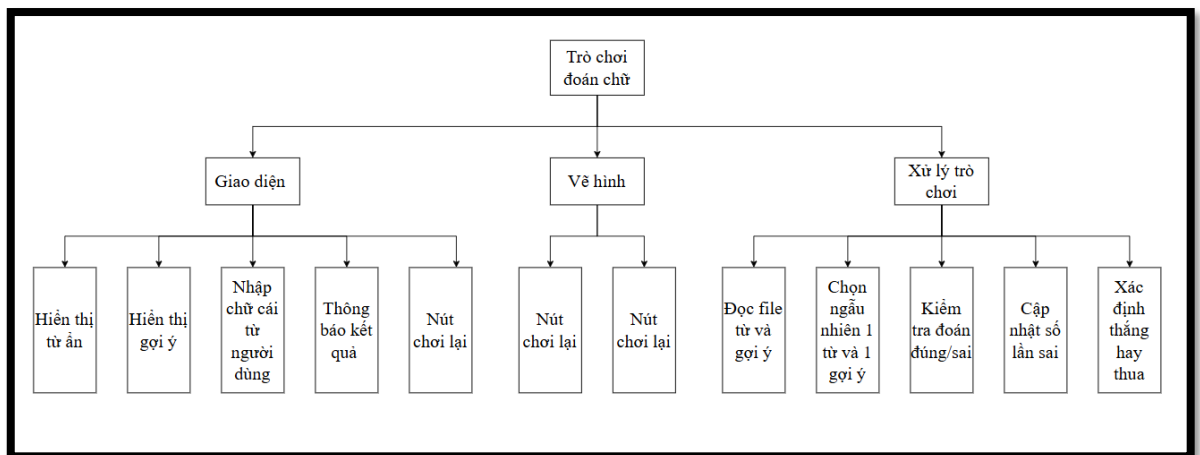
3.1.1. Các module chính của chương trình

Chương trình được tổ chức thành 2 phần chính, mỗi phần đảm nhận một nhiệm vụ riêng:

- **Xử lý logic trò chơi:** Gồm các chức năng chọn từ ngẫu nhiên, kiểm tra đoán đúng/sai, cập nhật trạng thái trò chơi (thắng/thua), lưu lại các chữ cái đã đoán, và đếm số lần sai.
- **Giao diện người dùng (GUI):** Hiển thị từ cần đoán dưới dạng dấu gạch dưới, nhận chữ cái người chơi nhập vào, thông báo kết quả, và vẽ từng phần cơ thể người treo cổ trên Canvas.

3.1.2. Biểu đồ phân cấp chức năng

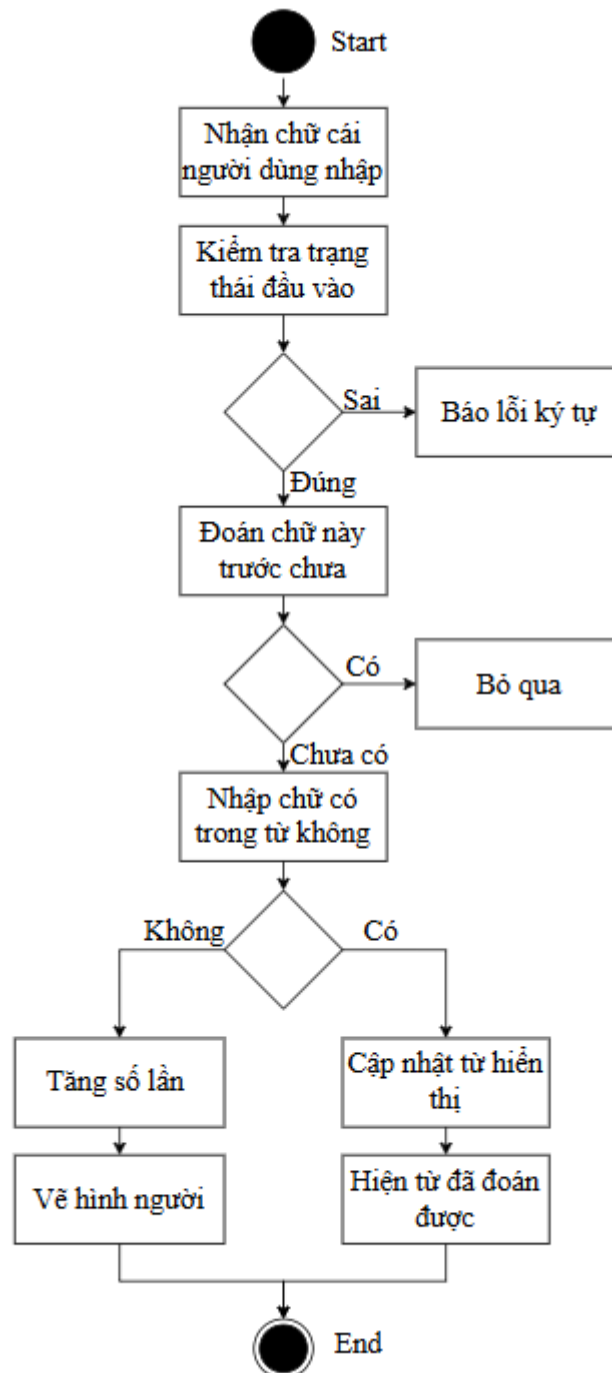
Dựa trên phân tích yêu cầu của bài toán, em có biểu đồ phân cấp chức năng sau:



Hình 3.1: Biểu đồ phân cấp chức năng

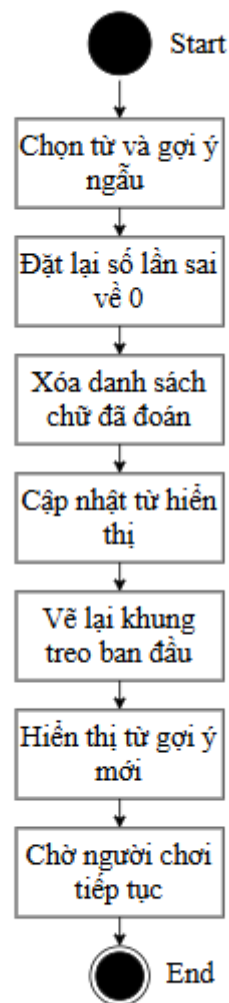
3.2. Sơ đồ khối các thuật toán chính

3.2.1. Thuật toán đoán chữ



Hình 3.2: Sơ đồ khối thuật toán đoán chữ

3.2.2. Thuật toán chơi lại



Hình 3.3: Sơ đồ khối thuật toán chơi lại

3.3. Cấu trúc dữ liệu

Trong chương trình Đoán chữ, em sử dụng các kiểu dữ liệu cơ bản để lưu trữ và xử lý thông tin trong suốt quá trình chơi:

- **Kiểu str:** dùng để lưu từ cần đoán (`tu_dap_an`) và gợi ý tương ứng (`goi_y`). Đây là 2 thông tin quan trọng, được chọn ngẫu nhiên từ 2 file `words.txt` và `cauhoi.txt`.
- **Kiểu list:** được sử dụng tạm thời để đọc toàn bộ nội dung file theo từng dòng, giúp chuyển thành danh sách các từ và gợi ý để chọn ngẫu nhiên.

- **Kiểu set:** dùng để lưu **các chữ cái người chơi đã đoán**. Việc sử dụng tập hợp giúp kiểm tra trùng lặp nhanh và tránh đoán lại những chữ đã nhập trước đó.
- **Kiểu int:** được sử dụng để đếm **số lần đoán sai** (`so_lan_doan_sai`). Dựa vào giá trị này, chương trình sẽ vẽ từng phần của người treo cổ và kiểm tra điều kiện thua.

Tuy không sử dụng các cấu trúc dữ liệu phức tạp như dictionary hoặc class lồng nhau, nhưng cách tổ chức dữ liệu đơn giản, dễ hiểu vẫn đủ để xử lý toàn bộ logic của trò chơi.

3.4. Chương trình

Chương trình được tổ chức thành hai lớp chính, mỗi lớp đảm nhiệm một nhóm chức năng riêng biệt:

- **Lớp thứ nhất là TroChoiTreoCo**, phụ trách toàn bộ phần xử lý logic. Lớp này thực hiện các chức năng như đọc từ và gợi ý từ file, chọn ngẫu nhiên một từ mới, kiểm tra chữ cái đoán đúng hay sai, theo dõi số lần sai, kiểm tra điều kiện thắng hoặc thua, và cung cấp dữ liệu cần thiết cho giao diện để hiển thị.
- **Lớp thứ hai là GiaoDienTreoCo**, phụ trách giao diện người dùng. Trong lớp này, các thành phần giao diện như nhãn hiển thị từ, ô nhập chữ cái, nút đoán, nút chơi lại và khung vẽ người treo cổ được tạo và sắp xếp bằng Tkinter. Khi người dùng nhập chữ cái và nhấn đoán, lớp này sẽ gọi hàm xử lý trong lớp logic, sau đó cập nhật lại giao diện theo kết quả.

Mỗi chức năng trong chương trình được chia thành các hàm riêng, như hàm khởi động lại trò chơi, hàm xử lý đoán chữ, hàm vẽ người, hàm kiểm tra trạng thái,... giúp chương trình rõ ràng, dễ theo dõi và dễ mở rộng về sau.

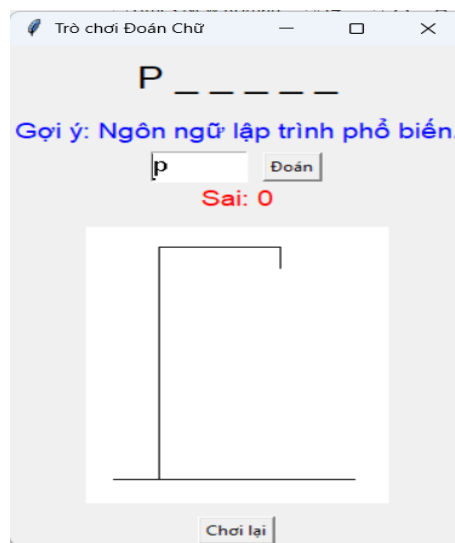
CHƯƠNG 4. THỰC NGHIỆM VÀ KẾT LUẬN

4.1. Thực nghiệm

4.1.1. Ghi lại kết quả các tính năng

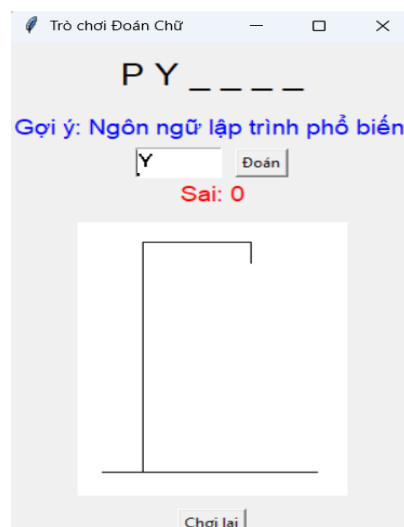
Sau khi hoàn thiện chương trình, tiến hành chạy thử với các bộ dữ liệu đầu vào khác nhau để kiểm tra hoạt động của từng chức năng, bao gồm: nhập chữ thường, chữ in hoa, nhập số, nhập quá chữ, đoán sai và đoán đúng

Nhập chữ thường:



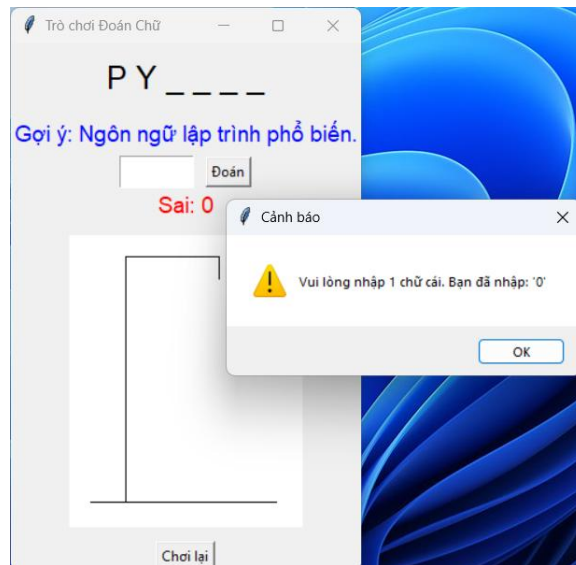
Hình 4.1: Kết quả chữ thường biến thành chữ hoa

Nhập chữ in hoa:



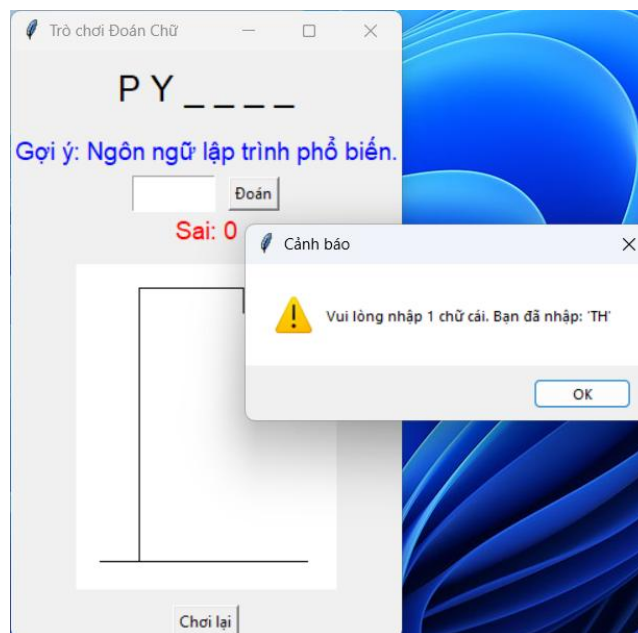
Hình 4.2: Kết quả nhập chữ in hoa

Nhập số:



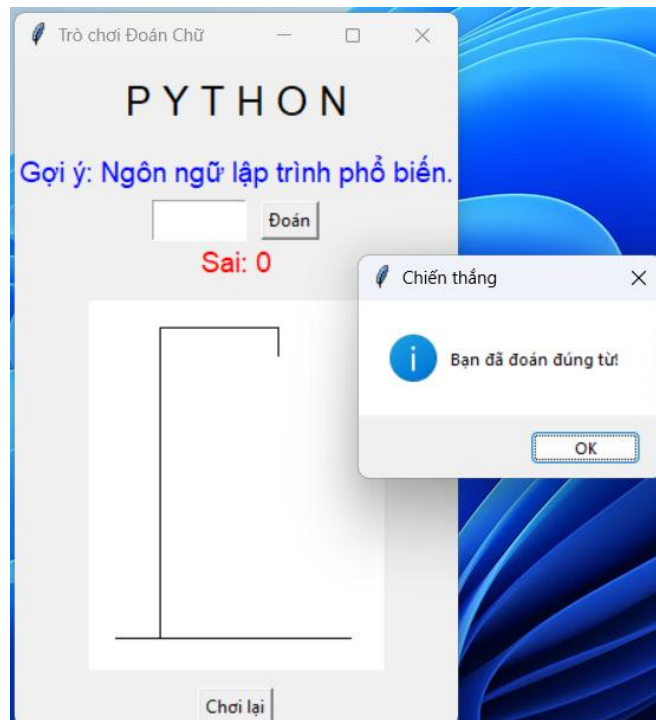
Hình 4.3: Kết quả nhập số bị báo lỗi khi nhập không đúng

Nhập quá chữ:



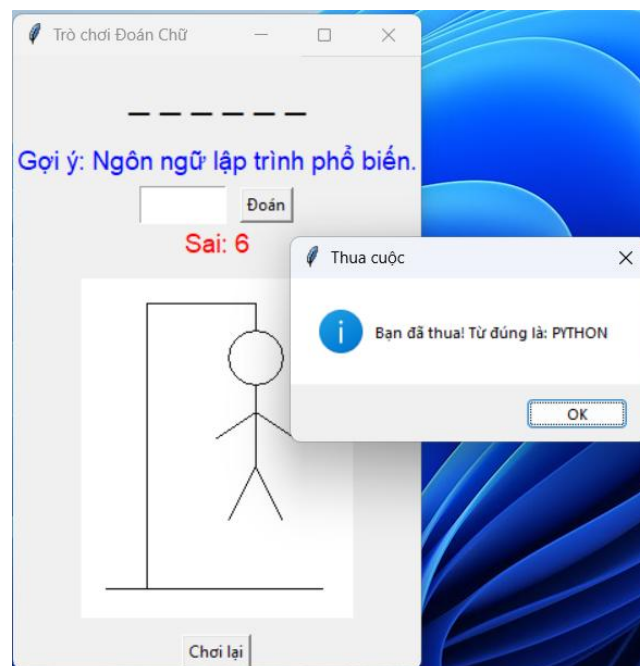
Hình 4.4: Kết quả nhập quá chữ báo lỗi nhập quá

Đoán đúng chữ:



Hình 4.5: Kết quả đoán đúng chữ

Đoán sai chữ:



Hình 4.6: Kết quả đoán sai chữ

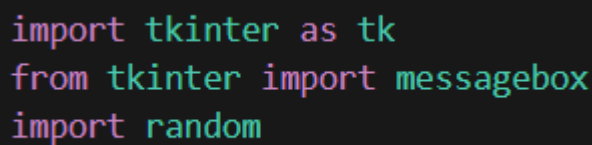
Đánh giá:

Chương trình chạy ổn định, phản hồi nhanh, xử lý đầy đủ các trường hợp thông dụng và lỗi thường gặp. Các tính năng cơ bản đều hoạt động chính xác, giao diện đơn giản, dễ sử dụng. Nhưng vẫn còn sơ sài về đồ họa và các gợi ý trong chương trình.

4.1.2. Mô tả các tính năng

a) Giao diện đồ họa GUI

- Giao diện sử dụng thư viện tkinter, với thiết kế đơn giản, gọn gàng.
- Sử dụng module messagebox của thư viện Tkinter để hiển thị thông báo
- Sử dụng thư viện random để chọn từ ngẫu nhiên



```
import tkinter as tk
from tkinter import messagebox
import random
```

Hình 4.7: Code giao diện đồ họa GUI

b) Xử lý logic trò chơi

- Đọc 2 file văn bản chữ từ và gợi ý. Các từ được chuyển thành chữ in hoa để đồng bộ khi so sánh
- Xử lý lỗi nếu số dòng không khớp hoặc file bị lỗi, chương trình sẽ dừng và thông báo lỗi
- Gọi 1 hàm `khởi_dong_tu_moi` để chọn ngẫu nhiên 1 từ và 1 gợi ý và đồng thời reset số lần sai và danh sách chữ đã đoán
- Xử lý đoán chữ kiểm tra xem chữ đã đoán chưa, nếu chưa thì thêm vào. Nếu sai thì tăng số lần sai
- Xử lý các chữ đoán đúng và gạch dưới các chữ chưa được đoán
- Kiểm tra trạng thái thắng hay thua

```
class TroChoiDoanChu:
    def __init__(self):
        try:
            with open("words.txt", "r", encoding="utf-8") as f:
                self.ds_tu = [dong.strip().upper() for dong in f.readlines()]
            with open("cauhoi.txt", "r", encoding="utf-8") as f:
                self.ds_goi_y = [dong.strip() for dong in f.readlines()]

            if len(self.ds_tu) != len(self.ds_goi_y):
                raise ValueError("Số lượng từ và gợi ý không khớp!")
        except Exception as loi:
            messagebox.showerror("Lỗi", f"Lỗi khi đọc file: {loi}")
            exit()

        self.khoi_dong_tu_moi()

    def khoi_dong_tu_moi(self):
        chi_so = random.randint(0, len(self.ds_tu) - 1)
        self.tu_dap_an = self.ds_tu[chi_so]
        self.goi_y = self.ds_goi_y[chi_so]
        self.cac_chu_da_doan = set()
        self.so_lan_doan_sai = 0

    def doan_chu(self, chu):
        if chu in self.cac_chu_da_doan:
            return False
        self.cac_chu_da_doan.add(chu)
        if chu not in self.tu_dap_an:
            self.so_lan_doan_sai += 1
            return False
        return True
```

Hình 4.8: Code xử lý logic của trò chơi

```
def hien_thi_tu(self):
    return " ".join([chu if chu in self.cac_chu_da_doan else "_" for chu in self.tu_dap_an])

def da_thang(self):
    return all(chu in self.cac_chu_da_doan for chu in self.tu_dap_an)

def da_thua(self):
    return self.so_lan_doan_sai >= 6
```

Hình 4.9: Code xử lý logic của trò chơi

c) Tạo giao diện trò chơi

- Khởi tạo cửa sổ chính Tkinter và khởi tạo đối tượng trò chơi

```
class GiaoDienDoanChu:
    def __init__(self, root):
        self.root = root
        self.root.title("Trò chơi Đoán Chữ")
        self.tro_choi = TroChoiDoanChu()
```

Hình 4.10: Tạo giao diện của trò chơi

d) Thông báo thắng/thua

- Nếu người chơi đã thắng hoặc thua thì hiện thông báo và chơi lại

```
if self.tro_choi.da_thang():
    messagebox.showinfo("Chiến thắng", "Bạn đã đoán đúng từ!")
    self.khoi_dong_lai()
elif self.tro_choi.da_thua():
    messagebox.showinfo("Thua cuộc", f"Bạn đã thua! Từ đúng là: {self.tro_choi.tu_dap_an}")
    self.khoi_dong_lai()
```

Hình 4.11: Thông báo thắng/thua

4.2. Kết luận

Sau khi quá trình thiết kế và xây dựng, chương trình “ game Hangman GUI” đã hoàn thành đầy đủ các tính năng cơ bản như: Hiện thị câu hỏi gợi ý, nhập từ cần đoán, thông báo nhập sai, thông báo thắng/thua

4.2.1. Những điều học được trong quá trình thực hiện

Những điều học được trong quá trình thực hiện:

- Hiểu thêm được các kiến thức về thư viện Tkinter để thiết kế đồ họa
- Bổ sung thêm kiến thức về Canvas
- Hiểu được cách xử lý lỗi với try – except và sử dụng messagebox để thông báo lỗi
- Nắm vững thêm cách tách biệt giữa các lớp xử lý trò chơi và giao diện

4.2.2. Hướng cải tiến trong tương lai

Các hướng cải tiến trong tương lai:

- Bổ sung thêm các tính năng trợ giúp vào trò chơi
- Thiết kế giao diện thân thiện bắt mắt hơn
- Thêm tính năng có thể chơi online với bạn bè
- Có thêm cập nhật thêm nhiều gợi ý với từ cần đoán tự động

TÀI LIỆU THAM KHẢO

1. Python Software Foundation. Python 3 Documentation. Truy cập tại: <https://docs.python.org/3/>
2. Tkinter Docs - Python Interface to Tcl/Tk. Truy cập tại: <https://tkdocs.com/>
3. Python Tkinter Tutorial. Truy cập tại: <https://www.geeksforgeeks.org/python-gui-tkinter/>
4. W3Schools. https://www.w3schools.com/python/python_try_except.asp
5. ChatGPT – OpenAI (<https://chat.openai.com>)
6. Mã QR của Github:



7. Mã QR của Youtube

