**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN HỌC PHẦN**

TÊN HỌC PHẦN: **Lập trình Python (PYPRO)**

MÃ SỐ LỚP HP: **IPPA233277\_03**

Tên chủ đề : **LẬP TRÌNH PYTHON PHÂN TÍCH DỮ LIỆU THĂM DÒ (EDA) VỀ “NỮ ĐẠI KIỆN TƯỚNG CỜ VUA (8/2020)**”

Họ tên sinh viên:LÊ THÀNH VINH [84]

Mã số sinh viên: **21110940**

Lớp:211103 [CQ]

Ngày nộp: **10/05/2023**

Ký tên:

**TP.HCM, ngày 10 Tháng 05 năm 2023**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI ĐỒ ÁN HỌC PHẦN**

TÊN HỌC PHẦN: **Lập trình Python (PYPRO)**

MÃ SỐ LỚP HP: **IPPA233277\_03**

Tên đề tài: **LẬP TRÌNH PYTHON PHÂN TÍCH DỮ LIỆU THĂM DÒ (EDA) VỀ “NỮ ĐẠI KIỆN TƯỚNG CỜ VUA (8/2020)**”

**Giảng viên giảng dạy: VÕ XUÂN THỂ**

**Họ tên sinh viên:** LÊ THÀNH VINH [84]

**Mã số sinh viên: 21110940**

**Lớp:** 211103 [CQ]

**TÊN CÁC FILE SẢN PHẨM ĐỀ TÀI:**

**Tên sản phẩm đề tài: G3\_84LeThanhVinh\_ChessWomen.rar**

**Tập tin mã nguồn: G3\_84LeThanhVinh\_ChessWomen\_Main.py, G3\_84LeThanhVinh\_ChessWomen\_EDA.py, G3\_84LeThanhVinh\_ChessWomen\_Frame.py, G3\_84LeThanhVinh\_ChessWomen\_Game.py**

**Tập dữ liệu thực nghiệm: G3\_84LeThanhVinh\_ChessWomen.csv**

**Giới thiệu thư mục bài làm: G3\_84LeThanhVinh\_DAHP.PyPro\_ChessWomen.Intro.docx**

**Tập các thư viện sử dụng:** # Speech: speech\_recognition, gtts, playsound, re

# GUI: tkinter, customtkinter, os

# EDA: pandas, numpy, scipy, sklearn, PIL, seaborn, matplotlib

# Frame: cv2, time

# Game: cv2, pygame, sys, random

**NHẬN XÉT CỦA GIẢNG VIÊN**

**………………………………………………………..**

**………………………………………………………..**

**Ngày …/…/2023 Giảng viên: Ký tên**

**TP.HCM, ngày 10 Tháng 05 năm 2023**

**LỜI CẢM ƠN**

Em xin được gửi lời cảm ơn chân thành đến Thầy Võ Xuân Thể về sự hỗ trợ và định hướng trong quá trình học môn Lập trình Python Phân tích (EDA). Sự dạy dỗ và sự tận tâm của Thầy đã giúp em có một trải nghiệm học tập vô cùng đáng nhớ và hữu ích.

Trong suốt quá trình học, Thầy đã truyền đạt kiến thức với sự rõ ràng và tỉ mỉ, tạo điều kiện để em hiểu và ứng dụng các khái niệm Lập trình Python vào việc phân tích dữ liệu. Nhờ vào phong cách giảng dạy tận tâm của Thầy, em đã có cơ hội nắm vững những kỹ năng và công cụ cần thiết để thực hiện các phân tích EDA một cách hiệu quả.

Thầy cũng đã tạo một môi trường học tập tích cực và khuyến khích sự tham gia chủ động của emi trong quá trình bài tập và dự án. Nhờ vào sự hướng dẫn chi tiết và phản hồi đầy kiến thức của Thầy, em đã có thể áp dụng kiến thức vào thực tế và khám phá thêm nhiều khía cạnh thú vị của phân tích dữ liệu.

Cuối cùng, em muốn bày tỏ lòng biết ơn sâu sắc với Thầy về sự tạo điều kiện thuận lợi cho sự phát triển cá nhân của em trong môn học này. Những kiến thức và kỹ năng mà em đã học được sẽ không chỉ hữu ích trong lĩnh vực Lập trình Python Phân tích (EDA), mà còn cung cấp cho em nền tảng vững chắc để khám phá những lĩnh vực khác trong tương lai.

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| CSDL hoặc DB | Cơ sở dữ liệu: DataBase |
| NSD = Users = Account | Người Sử Dụng = là quyền làm việc trên WebApp được đảm bảo bởi tối thiểu là 2 yếu tố: tên đăng nhập (UserName) và mật khẩu (Password) |
|  |  |
| AI | Artificial intelligence: Trí tuệ nhân tạo |
| EDA | Exploratory Data Analysis: Phân tích Khám phá Dữ liệu (Còn gọi là: phân tích dữ liệu thăm dò) |
| GUI | Graphical User Interface: Giao diện đồ hoạ |
| PYPRO | Lập trình Python |

**DANH MỤC CÁC THUẬT NGỮ**

|  |  |
| --- | --- |
| Giao diện người dùng:  User-Interface | Là hệ thống các màn hình giao tiếp cho phép người sử dụng tương tác với các thành phần phần mềm trong HTTT, thường là 1 trong các dong: Win-form, Web-form, Mobile-Form. |
| Tài khoản (Account) | Là một quyền làm việc trên HTTT được cấp phát cho một cá nhân thông qua tên tài khoản (username) và mật khẩu (password). |
| Trợ lý ảo: Voice Assistant | Là các hệ thống có khả năng “nghe” và “nói” với con người, nhờ đó hỗ trợ con người trong một số chức năng như một “trợ lý”. |

**MỤC LỤC**

[LỜI CẢM ƠN 3](#_Toc9295)

[DANH MỤC CÁC TỪ VIẾT TẮT 4](#_Toc23274)

[DANH MỤC CÁC THUẬT NGỮ 5](#_Toc31797)

[Chương 1: GIỚI THIỆU VỀ ĐỒ ÁN HỌC PHẦN 9](#_Toc23924)

[1.1. Tổng quan về ĐỒ ÁN HỌC PHẦN 9](#_Toc27907)

[1.2. Nội dung chuyên môn chính của ĐỒ ÁN HỌC PHẦN 9](#_Toc23648)

[1.3. Công cụ và nền tảng kỹ thuật thực hiện ĐỒ ÁN HỌC PHẦN 9](#_Toc19552)

[1.4. Bố cục của báo cáo 10](#_Toc14730)

[Chương 2: CƠ SỞ LÝ THUYẾT CỦA HỌC PHẦN 12](#_Toc5179)

[2.1. GIỚI THIỆU VỀ LẬP TRÌNH PYTHON 12](#_Toc20617)

[2.1.1. Giới thiệu chung 12](#_Toc23982)

[2.1.2. Cài đặt “bộ thảo chương” (soạn thảo chương trình IDE) và “bộ dịch” 12](#_Toc14975)

[2.1.3. Giới thiệu các bộ tương tác lập trình 12](#_Toc30717)

[2.1.4. Cấu hình mô trường tương tác lập trình 12](#_Toc24999)

[2.2. TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH PYTHON 12](#_Toc9324)

[2.2.1. Danh hiệu (Identifier) 12](#_Toc27199)

[2.2.2. Về viết mã lệnh lập trình (Code) 12](#_Toc1551)

[2.2.3. Về chú giải (Comment) 12](#_Toc25525)

[2.2.4. Các phép toán 12](#_Toc3256)

[2.2.5. Chuỗi 12](#_Toc3133)

[2.2.6. Lệnh 12](#_Toc7343)

[2.2.7. Định nghĩa hàm 12](#_Toc585)

[2.2.8. Danh sách 12](#_Toc219)

[2.2.9. Bộ dữ liệu: Tuple 12](#_Toc23152)

[2.2.10. Tập hợp: Set 12](#_Toc25476)

[2.2.11. Từ điển: Dictionary 12](#_Toc1471)

[2.2.12. Module 12](#_Toc10713)

[2.2.13. Package 12](#_Toc5019)

[2.2.14. Lớp 12](#_Toc12501)

[2.2.15. Xử lý ngoại lệ 12](#_Toc29588)

[2.3. GIỚI THIỆU CÁC THƯ VIỆN VÀ CÔNG CỤ CĂN BẢN 12](#_Toc7750)

[2.3.1. Nạp thư viện 12](#_Toc16082)

[2.3.2. Pandas 12](#_Toc20217)

[2.3.3. Tensorflow 12](#_Toc3600)

[2.3.4. NumPy 12](#_Toc29530)

[2.3.5. SCIPY 12](#_Toc18490)

[2.3.6. Theano 12](#_Toc13838)

[2.3.7. Matplotlib 12](#_Toc30504)

[2.3.8. scikit-learn = sklearn 13](#_Toc25672)

[2.3.9. Keras 13](#_Toc23312)

[2.3.10. NLTK 13](#_Toc3566)

[2.3.11. PyTorch 13](#_Toc31394)

[2.3.12. LightGBM 13](#_Toc972)

[2.3.13. Eli5 13](#_Toc7405)

[2.4. LẬP TRÌNH PYTHON XỬ LÝ GIỌNG NÓI (Trợ lý ảo: Voice Assistant) 13](#_Toc3167)

[2.4.1. Giới thiệu chủ đề 13](#_Toc17274)

[2.4.2. Tài liệu và nguồn thực nghiệm 13](#_Toc11267)

[2.4.3. Giới thiệu Các thư viện Python liên quan 13](#_Toc11848)

[2.4.4. Giới thiệu bài toán 13](#_Toc906)

[2.4.5. Giới thiệu code tham khảo 13](#_Toc133)

[2.5. KỸ THUẬT LẬP TRÌNH ĐỆ QUY: THÁP HÀ NỘI (HANOI TOWER) 13](#_Toc8320)

[2.6. LẬP TRÌNH PHÂN TÍCH KHÁM PHÁ (EDA) [Thăm dò] 13](#_Toc5180)

[Chương 3: PHÂN TÍCH VÀ XÁC ĐINH CÁC CƠ SỞ KỸ THUẬT 14](#_Toc13487)

[3.1. GIỚI THIỆU CHUNG VỀ CHỦ ĐỀ 14](#_Toc15444)

[3.1.1. Mô tả yêu câu của chủ đề 14](#_Toc5788)

[3.1.2. Phân tích yều cầu của chủ đề 14](#_Toc28555)

[3.2. CÁC NỀN TẢNG KỸ THUẬT 14](#_Toc15126)

[3.2.1. Hệ thống thư viện sử dụng 14](#_Toc27739)

[3.2.2. Tập dữ liệu thực nghiệm 15](#_Toc6157)

[3.2.3. Mô tả giải thuật 15](#_Toc9444)

[3.3. LẬP TRÌNH XỬ GIỌNG NÓI (TRỢ LÝ ẢO: VOICE ASSISTANT) 16](#_Toc9033)

[3.3.1. Giới thiệu chung về chủ đề 16](#_Toc31570)

[3.3.1.1. Mô tả yêu cầu của chủ đề 16](#_Toc29627)

[3.3.1.2. Phân tích yều cầu của chủ đề 16](#_Toc23137)

[3.3.2. Các nền tảng kỹ thuật 17](#_Toc640)

[3.3.2.1. Nền tảng kỹ thuật cơ bản 17](#_Toc21212)

[3.3.2.2. Hệ thống thư việc sử dụng 17](#_Toc12495)

[3.3.3. Thiết kế giao diện 17](#_Toc28244)

[3.3.4. Mã lệnh lập trình (python) 17](#_Toc8797)

[Chương 4: LẬP TRÌNH PYTHON LẬP TRÌNH PYTHON PHÂN TÍCH DỮ LIỆU THĂM DÒ (EDA) VỀ “NỮ ĐẠI KIỆN TƯỚNG CỜ VUA (8/2020)” 19](#_Toc10343)

[4.1. Thiết kế giao diện 19](#_Toc14348)

[4.1.1 Form Main: 19](#_Toc19267)

[4.1.2 Form EDA: 21](#_Toc5337)

[4.1.3 Form Cắt Frame Video 22](#_Toc4435)

[4.1.4 Form Game 23](#_Toc10082)

[4.2. Mã lệnh lập trình (python) 23](#_Toc15665)

[4.2.1 Giao diện chính: G3\_84LeThanhVinh\_ChessWomen\_Main.py 23](#_Toc17535)

[4.2.2 EDA: G3\_84LeThanhVinh\_ChessWomen\_EDA.py, 27](#_Toc8790)

[4.2.3 Frame: G3\_84LeThanhVinh\_ChessWomen\_Frame.py 37](#_Toc960)

[4.2.4 Game: G3\_84LeThanhVinh\_ChessWomen\_Game.py 43](#_Toc11295)

[Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỒ ÁN HỌC PHẦN 50](#_Toc21720)

[5.1. Kết luận 50](#_Toc7711)

[5.1.1. Những kết quả đạt được 50](#_Toc15659)

[5.1.2 Hạn chế 50](#_Toc24531)

[5.2. Hướng phát triển 50](#_Toc26406)

[5.2.1. Hướng khắc phục các hạn chế 50](#_Toc30521)

[5.2.2. Hướng mở rộng ĐỒ ÁN HỌC PHẦN 51](#_Toc14605)

[DANH MỤC TÀI LIỆU THAM KHẢO 51](#_Toc12504)

# GIỚI THIỆU VỀ ĐỒ ÁN HỌC PHẦN

## Tổng quan về ĐỒ ÁN HỌC PHẦN

Đồ án Học phần Lập trình Python: LẬP TRÌNH PYTHON PHÂN TÍCH DỮ LIỆU THĂM DÒ (EDA) VỀ “NỮ ĐẠI KIỆN TƯỚNG CỜ VUA (8/2020)” vận dụng kiến thức và kỹ năng về Lập trình Python để xây dựng giao diện đồ hoạ (GUI) cho phép: tương tác với người dùng bằng giọng nói (Trợ lý ảo: Voice Assistant) và phân tích thăm dò (EDA: Epolore Data Analysing) đối với tập dữ liệu thực nghiệm về G3\_84LeThanhVinh\_ChessWomen.csv

## Nội dung chuyên môn chính của ĐỒ ÁN HỌC PHẦN

Bao gồm 6 nội dung lập trình chính dùng ngôn ngữ lập trình Python:

+ Trợ lý ảo

+ Phân tích dữ liệu khám phá và thăm dò

+ Giao diện đồ họa dạng WinForm

+ Vẽ đồ thị

+ Cắt Frame video

+ Game

## Công cụ và nền tảng kỹ thuật thực hiện ĐỒ ÁN HỌC PHẦN

+ Ngôn ngữ lập trình **Python**

+ Công cụ lập trình và biên dịch: **VS Code**

+ Các thư viện chính:

    # Speech

import speech\_recognition as vinh84

from gtts import gTTS

import playsound

import re

    # tkinter

import tkinter as tkVinh

from tkinter import messagebox

from tkinter import ttk, filedialog

from tkinter import \*

from tkinter.font import Font

    # thư viện OS (lập thư mục, files)

import os  # THƯ VIỆN OS MS. WINDOWS = Lập thư mục & lưu file

   # Nạp các thư viện cần thiết EDA

import numpy as np #Numeric Python: Thư viện về Đại số tuyến tính tính

import pandas as pd #Python Analytic on Data System: For data processing (Thư viện xử lý dữ liệu)

from scipy import stats # thư viện cung cấp các công cụ thống kê [statistics] sub-lib của science python [các công cụ khoa học]

from sklearn import preprocessing # Thư viện tiền xử lý DL (XL ngoại lệ: Isolated)

from sklearn.feature\_selection import SelectKBest, chi2 # Nạp hàm Thư viện phân tích dữ liệu thăm dò

from PIL import ImageTk, Image

import matplotlib.pyplot as plt

from mpl\_toolkits import mplot3d

import seaborn as sns

## Bố cục của báo cáo

Báo cáo gồm những nội dung như sau:

Chương 1: Giới thiệu đồ án của Học phần

Chương 2: Các cơ sở lý thuyết của Học phần phục việc thực hiện đề tài

Chương 3: Phân tích và xác đinh các cơ sở kỹ thuật thực hiện đề tài

Chương 4: Lập trình xây dựng sản phầm đề lài: LẬP TRÌNH PYTHON PHÂN TÍCH DỮ LIỆU THĂM DÒ (EDA) VỀ “NỮ ĐẠI KIỆN TƯỚNG CỜ VUA (8/2020)”

Chương 5 Tổng kết các kết quản đạt được và còn hạn chế của đồ án, đồng thời đề xuất hướng khắc phục hạn chế và phát triển Đồ án.

# CƠ SỞ LÝ THUYẾT CỦA HỌC PHẦN

## GIỚI THIỆU VỀ LẬP TRÌNH PYTHON

### Giới thiệu chung

### Cài đặt “bộ thảo chương” (soạn thảo chương trình IDE[[1]](#footnote-0)) và “bộ dịch”

### Giới thiệu các bộ tương tác lập trình

### Cấu hình mô trường tương tác lập trình

## TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH PYTHON

### Danh hiệu (Identifier)

### Về viết mã lệnh lập trình (Code)

### Về chú giải (Comment)

### Các phép toán

### Chuỗi

### Lệnh

### Định nghĩa hàm

### Danh sách

### Bộ dữ liệu: Tuple

### Tập hợp: Set

### Từ điển: Dictionary

### Module

### Package

### Lớp

### Xử lý ngoại lệ

## GIỚI THIỆU CÁC THƯ VIỆN VÀ CÔNG CỤ CĂN BẢN

### Nạp thư viện

### Pandas

### Tensorflow

### NumPy

### SCIPY

### Theano

### Matplotlib

### scikit-learn = sklearn

### Keras

### NLTK

### PyTorch

### LightGBM

### Eli5

## LẬP TRÌNH PYTHON XỬ LÝ GIỌNG NÓI (Trợ lý ảo: Voice Assistant)

### Giới thiệu chủ đề

### Tài liệu và nguồn thực nghiệm

### Giới thiệu Các thư viện Python liên quan

### Giới thiệu bài toán

### Giới thiệu code tham khảo

**[1] Nghe tiếng Việt => Text**

**[2] Text => đọc tiếng Việt**

## KỸ THUẬT LẬP TRÌNH ĐỆ QUY: THÁP HÀ NỘI (HANOI TOWER)

## LẬP TRÌNH PHÂN TÍCH KHÁM PHÁ (EDA) [Thăm dò]

1. **PHÂN TÍCH VÀ XÁC ĐINH CÁC CƠ SỞ KỸ THUẬT**

## GIỚI THIỆU CHUNG VỀ CHỦ ĐỀ

### Mô tả yêu câu của chủ đề

Vận dụng kỹ thuật lập trình EDA kết hợp với GUI và Voice Assistant cho phép dùng tập dữ liệu thực nghiệm về “Nữ đại kiện tướng cờ vua” để dự đoán và tư vấn các nhà phát triển phần mềm các trang cờ vua online để có các chiến lược tối ưa hóa hệ thống của họ

Dữ liệu thực nghiệm gồm thông tin của các nữ kiện tướng cờ vua đến tháng 8/2020 của FIDE:

G3\_84LeThanhVinh\_ChessWomen.csv : dạng file Excel .csv

. Số records: 8553

. Số thuộc tính input: 9 để đoán thuộc tính thứ 10: target variable:

= Inactive\_flag

### Phân tích yều cầu của chủ đề

+ Chủ đề lập trình python này là cơ sở của các ứng dụng về DataWarehouse, Bigdata, Data Analysis, Data Sciences, . . .

+ Đây là xử lý nền tảng cho các nội dung chuyên môn (nêu trên) chủ yếu là giai đoạn tiền xử lý dữ liệu đầu vào (Input Preporcessing), như: xử lý NULL, lọc dữ liệu, tích hợp dữ liệu (Integration),…

**Một số trường hợp tiền xử lý dữ liệu đầu vào (input) của các hệ thống EDA**

[1] Xử lý NULL là Fide id, Name, Federation, Gender, Year\_of\_birth, Title.

[2] Xử lý dữ liệu quá khác biệt = “ngoại lệ” (isolated) là 3230 dòng có dữ liệu cá biệt

## CÁC NỀN TẢNG KỸ THUẬT

### Hệ thống thư viện sử dụng

**Bước 1: Nhập các thư viện cần thiết**

# For linear algebra (Thư viện về Đại số tuyến tính tính)

import numpy as np

# For data processing  (Thư viện xử lý dữ liệu)

import pandas as pd

### Tập dữ liệu thực nghiệm

**Bước 2: Tải tập dữ liệu**

#Load the data set (Nạp tập dữ liệu)

df = pd.read\_csv('G3\_84LeThanhVinh\_ChessWomen.csv')

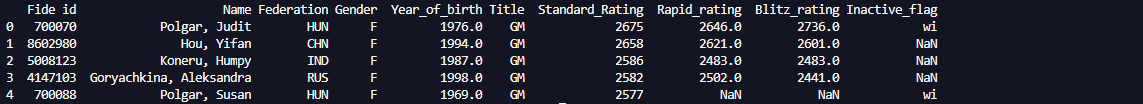
#Display the shape of the data set (Vẽ biểu đồ tập dữ liệu)

print('Độ lớn của bảng [frame] dữ liệu chess:',df.shape)

#Display data (Hiển thị dữ liệu dạng mảng 5 dòng đầu)

print(df[0:5])

Chạy G3\_84LeThanhVinh\_ChessWomen.py sẽ hiển thị kết quả:



### Mô tả giải thuật

**Bước 3: Xử lý dữ liệu**

# Checking for null values (Kiểm tra giá trị null = đếm số dòng có dữ liệu ứng từng thuộc tính)

print(df.count().sort\_values())

**Kết quả:**

Rapid\_rating 3608

Inactive\_flag 5852

Year\_of\_birth 8261

Fide id 8553

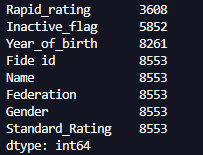
Name 8553

Federation 8553

Gender 8553

Standard\_Rating 8553

dtype: int64



## LẬP TRÌNH XỬ GIỌNG NÓI (TRỢ LÝ ẢO: VOICE ASSISTANT)

## Giới thiệu chung về chủ đề

### Mô tả yêu cầu của chủ đề

+ Lập trình App xử lý "lời nới" (Speech): Ứng dụng “Trợ lý ảo” (Voice Assistant)

+ Gồm 2 phần (Part) cơ bản sau

1. Nghe tiếng Việt => Text: Các số từ 1 đến 9 tương ứng với từng yêu cầu

2. Trả lời bằng tiếng Việt : Text => Hiển thị lên label của GUI

### Phân tích yều cầu của chủ đề

+ Chủ đề lập trình python này là cơ sở của các ứng dụng về AI, ML, Deep ML.

+ Các hệ thống này thường có 3 phần :

Part1: Speech chính là part1 của chủ đề này

Part2: Các kỹ thuật AI

Part2: Voice chính là part2 của chủ đề này

## Các nền tảng kỹ thuật

### Nền tảng kỹ thuật cơ bản

### Hệ thống thư việc sử dụng

B1: Cài đặt thư viện

Cài đặt trong môi trường ảo .venv

pip install SpeechRecognition

*pip install SpeechRecognition pydub # Long audio source*

pip install pyaudio

pip install regex

B2 : Nạp thư viện vào App

import speech\_recognition as vinh84

from gtts import gTTS

import playsound

import re

## Thiết kế giao diện

+ Xử lý trong cửa sổ Terminal của VS code

## Mã lệnh lập trình (python)

**FULL CODE**

import speech\_recognition as vinh84

from gtts import gTTS

import playsound

import re

  def voice(self):

      r\_vinh84 = vinh84.Recognizer()

      with vinh84.Microphone() as source:

          text\_audio = r\_vinh84.record(source, duration=5)

          try:

              text\_audio = r\_vinh84.recognize\_google(text\_audio, language="vi-VN")

              # Tìm kiếm các số trong đoạn văn bản đã nhận dạng

              numbers = re.findall(r'\d+', text\_audio)

              # Nếu tìm thấy ít nhất một số, hiển thị số đầu tiên lên nhãn

              if len(numbers) > 0:

                  self.label\_nhan\_dien.config(text=numbers[0])

              else:

                  self.label\_nhan\_dien.config(text="Không nhận dạng được số")

                  self.label\_nhan\_dien.update()

          except:

              self.label\_nhan\_dien.config(text="Không nhận dạng được")

              self.label\_nhan\_dien.update()

  def submit\_voice(self):

      if self.label\_nhan\_dien.cget("text") == "Không nhận dạng được":

          messagebox.showinfo("Thông báo", "Không nhận dạng được")

      elif self.label\_nhan\_dien.cget("text") == "1":

          self.open\_file()

      elif self.label\_nhan\_dien.cget("text") == "2":

          self.columns\_null()

      elif self.label\_nhan\_dien.cget("text") == "3":

          self.rows\_null()

      elif self.label\_nhan\_dien.cget("text") == "4":

          self.zcore()

      elif self.label\_nhan\_dien.cget("text") == "5":

          self.chuan\_hoa()

      elif self.label\_nhan\_dien.cget("text") == "6":

          self.histogram()

      elif self.label\_nhan\_dien.cget("text") == "7":

          self.phan\_tan()

      elif self.label\_nhan\_dien.cget("text") == "8":

          self.pie()

      elif self.label\_nhan\_dien.cget("text") == "9":

          self.scatter\_3d()

      else:

          messagebox.showinfo("Thông báo", "Không nhận dạng được")

1. **LẬP TRÌNH PYTHON LẬP TRÌNH PYTHON PHÂN TÍCH DỮ LIỆU THĂM DÒ (EDA) VỀ “NỮ ĐẠI KIỆN TƯỚNG CỜ VUA (8/2020)”**

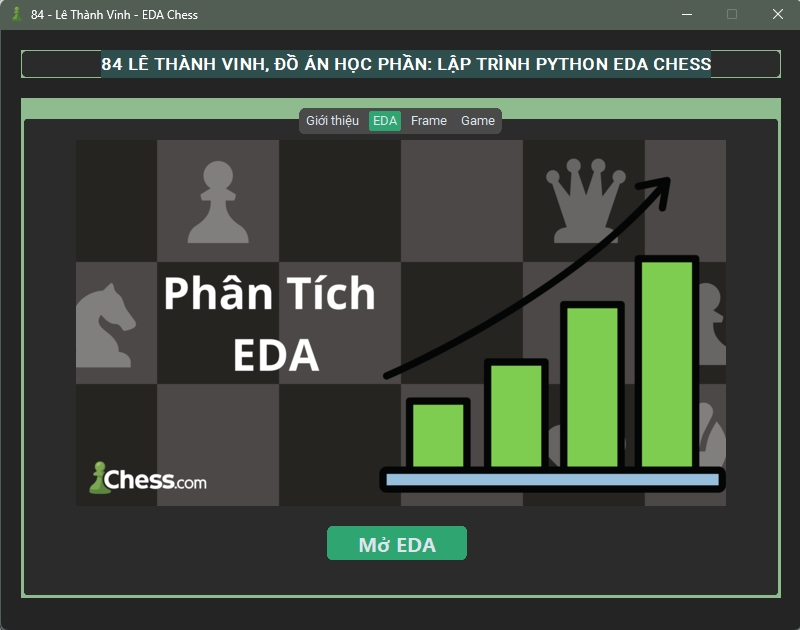
## Thiết kế giao diện

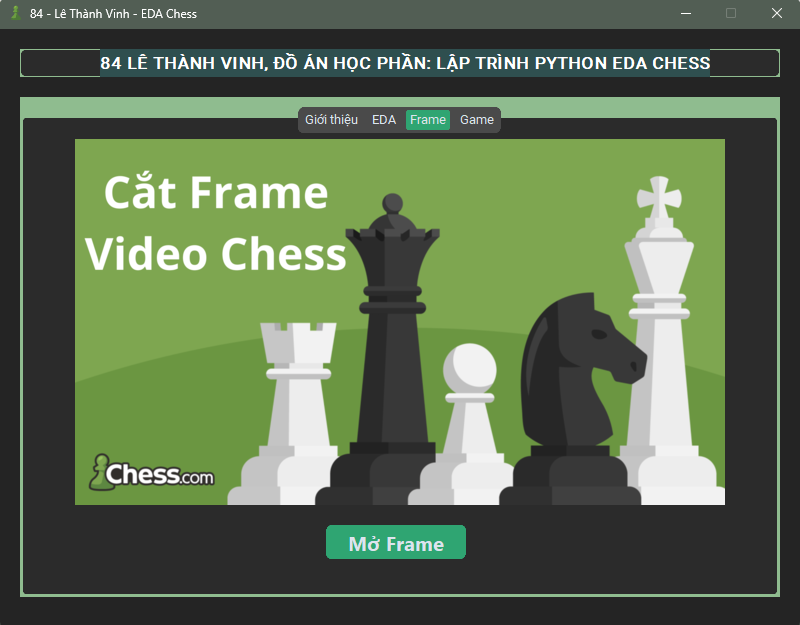
Đồ án được chia làm các form sau:

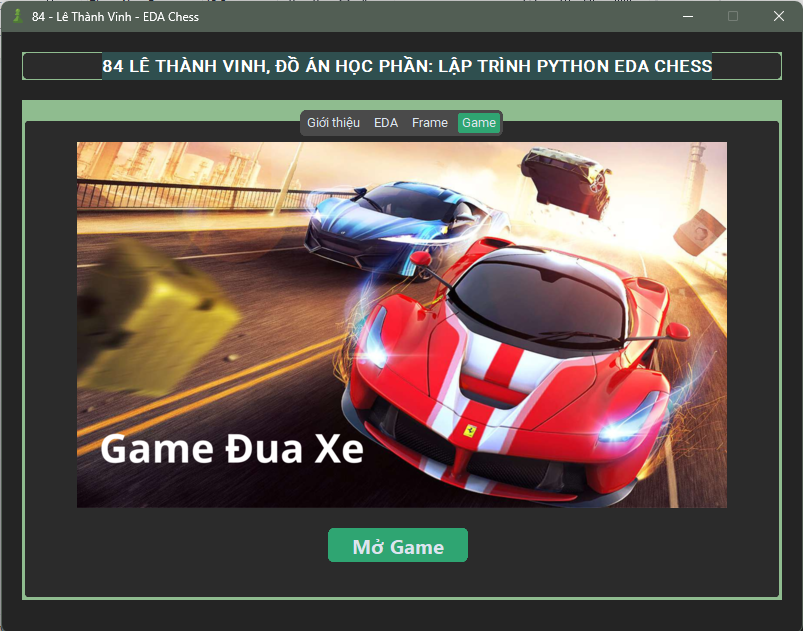
### 4.1.1 Form Main:

- Dùng thư viện customtkinter để tạo giao diện dễ nhìn





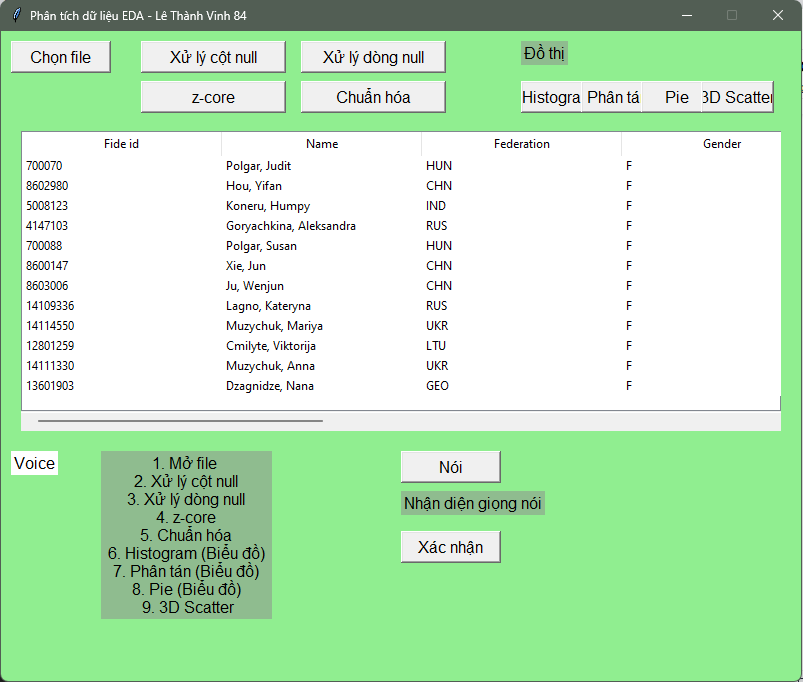




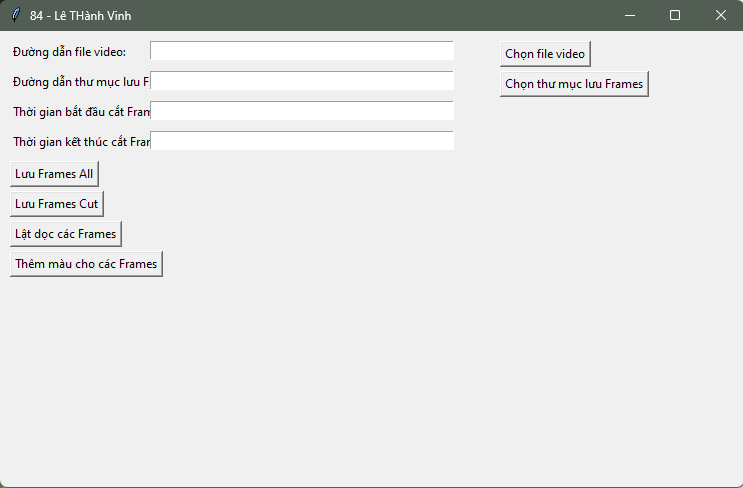
### 4.1.2 Form EDA:

- Voice Assistant: Tích hợp trên EDA

- Đồ thị: Cũng tích hợp trên EDA (Histogram, Phân tán, Pie, 3D-scatter)



### 4.1.3 Form Cắt Frame Video



### 4.1.4 Form Game



## Mã lệnh lập trình (python)

### 4.2.1 Giao diện chính: G3\_84LeThanhVinh\_ChessWomen\_Main.py

'''

NGười viết: Lê THành VInh

MSSV: 2110940

Ngày viết: 19/04/2023

Mô tả: Báo cáo cuối kì: EDA Chess - Giao diện chính

'''

import tkinter as tkVinh

import customtkinter as ctk

from tkinter import \*

from tkinter.font import Font

from CTkMessagebox import CTkMessagebox

import cv2

import PIL.Image, PIL.ImageTk

from PIL import ImageTk, Image

import tkinter.filedialog as filedialog

from PIL import Image

import os

import matplotlib.pyplot as plt

# import pygame,sys

# from pygame.locals import \*

import random

# import BaiTuan.Tuan14\_5 as game

import G3\_84LeThanhVinh\_ChessWomen\_Frame as Frame

import G3\_84LeThanhVinh\_ChessWomen\_Game as game

import G3\_84LeThanhVinh\_ChessWomen\_EDA as eda

ctk.set\_appearance\_mode("system")

ctk.set\_default\_color\_theme("green")

class App(ctk.CTk):

    width = 650

    height = 366

    def \_\_init\_\_(self, \*args, \*\*kwargs):

        super().\_\_init\_\_(\*args, \*\*kwargs)

        self.iconbitmap("Images/Icon.ico")

        self.title("84 - Lê Thành Vinh - EDA Chess")

        self.geometry("800x600")

        self.resizable(False, False)

        # Tạo Frame cho label giới thiệu

        self.frame = ctk.CTkFrame(self, width=800,height= 200, bg\_color="#8FBC8F", border\_width=1, border\_color="#8FBC8F")

        self.frame.grid(row=0, column=3, padx=(20, 0), pady=(20, 0), sticky="nsew")

        # Tạo label giới thiệu

        self.label = ctk.CTkLabel(self.frame, text="84 LÊ THÀNH VINH, ĐỒ ÁN HỌC PHẦN: LẬP TRÌNH PYTHON EDA CHESS", font=ctk.CTkFont(size=17, weight="bold"), text\_color="white", bg\_color="#2F4F4F")

        # self.label.grid(row=0, column=3, padx=(20, 0), pady=(20, 0), sticky="w")

        self.label.grid(row=0, column=3,padx=(80, 0), sticky="nsew")

        # tạo tabview

        self.tabview = ctk.CTkTabview(self, width=760, height=500, bg\_color="#8FBC8F", border\_width=3, border\_color="#8FBC8F")

        self.tabview.grid(row=3, column=3, padx=(20, 0), pady=(20, 0), sticky="nsew")

        self.tabview.add("Giới thiệu")

        self.tabview.tab("Giới thiệu").grid\_columnconfigure(0, weight=1)  # configure grid of individual tabs

        self.tabview.add("EDA")

        self.tabview.tab("EDA").grid\_columnconfigure(0, weight=1)

        self.tabview.add("Frame")

        self.tabview.tab("Frame").grid\_columnconfigure(0, weight=1)

        self.tabview.add("Game")

        self.tabview.tab("Game").grid\_columnconfigure(0, weight=1)

        # thêm hình nền cho tabview Giới thiệu

        current\_path = os.path.dirname(os.path.realpath(\_\_file\_\_))

        self.bg\_image = ctk.CTkImage(Image.open(current\_path + "/Images/gioi\_thieu.png"),

                                                  size=(self.width, self.height))

        self.bg\_image\_label = ctk.CTkLabel(self.tabview.tab("Giới thiệu"), image=self.bg\_image, text=None)

        self.bg\_image\_label.grid(row=0, column=0, sticky="nsew")

        # thêm hình nền cho tabview EDA

        self.bg\_image = ctk.CTkImage(Image.open(current\_path + "/Images/eda.png"),

                                                    size=(self.width, self.height))

        self.bg\_image\_label = ctk.CTkLabel(self.tabview.tab("EDA"), image=self.bg\_image, text=None)

        self.bg\_image\_label.grid(row=0, column=0, sticky="nsew")

        # thêm hình nền cho tabview Frame

        self.bg\_image = ctk.CTkImage(Image.open(current\_path + "/Images/Frame.png"),

                                                    size=(self.width, self.height))

        self.bg\_image\_label = ctk.CTkLabel(self.tabview.tab("Frame"), image=self.bg\_image, text=None)

        self.bg\_image\_label.grid(row=0, column=0, sticky="nsew")

        # thêm hình nền cho tabview Game

        self.bg\_image = ctk.CTkImage(Image.open(current\_path + "/Images/Game.png"),

                                                    size=(self.width, self.height))

        self.bg\_image\_label = ctk.CTkLabel(self.tabview.tab("Game"), image=self.bg\_image, text=None)

        self.bg\_image\_label.grid(row=0, column=0, sticky="nsew")

        # thêm button vào tabview Giới thiệu

        self.buttonGioiThieu = ctk.CTkButton(self.tabview.tab("Giới thiệu"), text="84 - Lê Thành Vinh", command=self.mo\_GioiThieu, font=ctk.CTkFont(size=20, weight="bold", family="Segoe UI"))

        self.buttonGioiThieu.grid(row=1, column=0, padx=(300, 0), pady=(20, 0), sticky="w")

        # thêm button vào tabview Frame

        self.buttonFrame = ctk.CTkButton(self.tabview.tab("Frame"), text="Mở Frame", command=self.mo\_Frame, font=ctk.CTkFont(size=20, weight="bold", family="Segoe UI"))

        self.buttonFrame.grid(row=1, column=0, padx=(300, 0), pady=(20, 0), sticky="w")

        # thêm button vào tabview EDA

        self.buttonEDA = ctk.CTkButton(self.tabview.tab("EDA"), text="Mở EDA", command=self.mo\_EDA, font=ctk.CTkFont(size=20, weight="bold", family="Segoe UI"))

        self.buttonEDA.grid(row=1, column=0, padx=(300, 0), pady=(20, 0), sticky="w")

        # thêm button vào tabview Game

        self.buttonGame = ctk.CTkButton(self.tabview.tab("Game"), text="Mở Game", command=self.mo\_Game, font=ctk.CTkFont(size=20, weight="bold", family="Segoe UI"))

        self.buttonGame.grid(row=1, column=0, padx=(300, 0), pady=(20, 0), sticky="w")

    def mo\_GioiThieu(self):

        # Nếu nhấn vào thì sẽ xuất hiện messagebox

        CTkMessagebox(message="Được tạo bởi: 84 - Lê Thành Vinh (21110940) \n Đồ án học phần: Lập trình Python EDA Chess",

                  icon="check", option\_1="Ok", title="Giới thiệu", font=ctk.CTkFont(size=16, family="Segoe UI"))

    def mo\_Game(self):

        # Nếu button 1 được nhấn thì mở game

        game.main()

    def mo\_EDA(self):

        # Nếu button 2 được nhấn thì mở EDA

        # eda.mainEDA()

        eda.mainEDA()

    def mo\_Frame(self):

        # Nếu button 3 được nhấn thì mở Frame

        Frame.GUI()

if \_\_name\_\_ == "\_\_main\_\_":

    app = App()

app.mainloop()

### 4.2.2 EDA: G3\_84LeThanhVinh\_ChessWomen\_EDA.py

'''

NGười viết: Lê THành VInh

MSSV: 2110940

Ngày viết: 19/04/2023

Mô tả: Báo cáo cuối kì: EDA Chess - phần EDA

'''

# B1: NẠP THƯ VIỆN

    # Speech

import speech\_recognition as vinh84

from gtts import gTTS

import playsound

import re

    # tkinter

import tkinter as tkVinh

from tkinter import messagebox

from tkinter import ttk, filedialog

from tkinter import \*

from tkinter.font import Font

from PIL import ImageTk, Image

import matplotlib.pyplot as plt

from mpl\_toolkits import mplot3d

import seaborn as sns

    # thư viện OS (lập thư mục, files)

import os  # THƯ VIỆN OS MS. WINDOWS = Lập thư mục & lưu file

   # Nạp các thư viện cần thiết EDA

import numpy as np #Numeric Python: Thư viện về Đại số tuyến tính tính

import pandas as pd #Python Analytic on Data System: For data processing (Thư viện xử lý dữ liệu)

from scipy import stats # thư viện cung cấp các công cụ thống kê [statistics] sub-lib của science python [các công cụ khoa học]

from sklearn import preprocessing # Thư viện tiền xử lý DL (XL ngoại lệ: Isolated)

from sklearn.feature\_selection import SelectKBest, chi2 # Nạp hàm Thư viện phân tích dữ liệu thăm dò

# # B2: KHAI BÁO TÊN THƯ MỤC & FILE LƯU CÁC THÔNG TIN BÀI LÀM

# LeVinh84\_FILE = "LeVinh84.mp3"  # lưu tên  file Input

# LeVinh84\_DIR = 'Le\_Vinh\_84'     # Thư mục lưu các file [trên]

# os.makedirs(LeVinh84\_DIR, exist\_ok=True)

class EDA:

    def \_\_init\_\_(self, form):

        self.form = form

        self.form.title("Phân tích dữ liệu EDA - Lê Thành Vinh 84")

        self.form.geometry("800x650")

        self.form.resizable(False, False)

        self.form.configure(bg='#90EE90')

        # # Tạo canvas với chiều rộng và chiều cao giống với cửa sổ gốc

        # canvas = tkVinh.Canvas(form, width=800, height=450)

        # canvas.pack()

        # # Load ảnh nền và tạo widget ảnh

        # img = Image.open("Images/EDAform.png")

        # bg\_image = ImageTk.PhotoImage(img)

        # canvas.create\_image(0, 0, image=bg\_image, anchor="nw")

        # canvas.place(relwidth=1, relheight=1)

        self.frame\_treeview = tkVinh.Frame(form)

        self.frame\_treeview.place(x=20, y=100, width=760, height=300, bordermode='outside')

        # Tạo thanh cuộn ngang bên trên cho treeview

        self.scroll = ttk.Scrollbar(self.frame\_treeview, orient='horizontal')

        self.scroll.place(x=0, y=280, width=760, height=20)

        # Tạo treeview

        self.table = ttk.Treeview(self.frame\_treeview, xscrollcommand=self.scroll.set, height=10)

        self.table.place(x=0, y=0, width=760, height=280)

        # Thiết lập kết nối giữa thanh cuộn và treeview

        self.scroll.config(command=self.table.xview)

        # KHởi tạo sẵn treeview

        self.df = pd.read\_csv('G3\_84LeThanhVinh\_ChessWomen.csv')

        self.table.delete(\*self.table.get\_children())

        self.table["column"] = list(self.df.columns)

        self.table["show"] = "headings"

        for column in self.table["column"]:

            self.table.heading(column, text=column)

        # thêm 20 dòng đầu tiên

        df\_rows = self.df.to\_numpy().tolist()

        for row in df\_rows[:20]:

            self.table.insert("", "end", values=row)

        # Button chọn file

        self.btn\_file = tkVinh.Button(form, text='Chọn file',width=10,font= Font(family="Segoe UI", size=12, weight="bold"), command=self.open\_file)

        self.btn\_file.place(x=10, y=10)

        # Button xử lý cột null

        self.btn\_cot\_null = tkVinh.Button(form, text='Xử lý cột null',width=15,font= Font(family="Segoe UI", size=12, weight="bold"), command=self.columns\_null)

        self.btn\_cot\_null.place(x=140, y=10)

        # Button xử lý dòng null (NaN)

        self.btn\_dong\_null = tkVinh.Button(form, text='Xử lý dòng null',width=15,font= Font(family="Segoe UI", size=12, weight="bold"), command=self.rows\_null)

        self.btn\_dong\_null.place(x=300, y=10)

        # Button tạo ma trận z-core

        self.btn\_zcore = tkVinh.Button(form, text='z-core',width=15,font= Font(family="Segoe UI", size=12, weight="bold"), command=self.zcore)

        self.btn\_zcore.place(x=140, y=50)

        # Button chuẩn hóa (rời rạc hóa)

        self.btn\_chuan\_hoa = tkVinh.Button(form, text='Chuẩn hóa',width=15,font= Font(family="Segoe UI", size=12, weight="bold"), command=self.chuan\_hoa)

        self.btn\_chuan\_hoa.place(x=300, y=50)

        # Label đồ thị

        self.label\_do\_thi = tkVinh.Label(form, text='Đồ thị',font= Font(family="Segoe UI", size=12, weight="bold"), bg='#8FBC8F')

        self.label\_do\_thi.place(x=520, y=10)

        # Button vẽ đồ thị (rời rạc vẽ đồ thị)

        self.btn\_histogram = tkVinh.Button(form, text='Histogram',width=7,font= Font(family="Segoe UI", size=8, weight="bold"), command=self.histogram)

        self.btn\_histogram.place(x=520, y=50)

        # Button vẽ đồ thị Phân tán

        self.btn\_chuan\_hoa = tkVinh.Button(form, text='Phân tán',width=7,font= Font(family="Segoe UI", size=8, weight="bold"), command=self.phan\_tan)

        self.btn\_chuan\_hoa.place(x=580, y=50)

        # Button vẽ đồ thị quạt (Pie)

        self.btn\_quat = tkVinh.Button(form, text='Pie',width=7,font= Font(family="Segoe UI", size=8, weight="bold"), command=self.pie)

        self.btn\_quat.place(x=640, y=50)

        # Button vẽ đồ thị 3D scatter

        self.btn\_scatter = tkVinh.Button(form, text='3D Scatter',width=7,font= Font(family="Segoe UI", size=8, weight="bold"), command=self.scatter\_3d)

        self.btn\_scatter.place(x=700, y=50)

        # Label Voice

        self.label\_voice = tkVinh.Label(form, text='Voice',font= Font(family="Segoe UI", size=12, weight="bold"), bg='white')

        self.label\_voice.place(x=10, y=420)

        # Label nội dung Voice

        self.label\_dung\_voice = tkVinh.Label(form, text='1. Mở file \n 2. Xử lý cột null \n 3. Xử lý dòng null \n 4. z-core \n 5. Chuẩn hóa \n 6. Histogram (Biểu đồ) \n 7. Phân tán (Biểu đồ) \n 8. Pie (Biểu đồ) \n 9. 3D Scatter',font= Font(family="Segoe UI", size=12, weight="bold"), bg='#8FBC8F')

        self.label\_dung\_voice.place(x=100, y=420)

        # Button Voice

        self.btn\_voice = tkVinh.Button(form, text='Nói',width=10,font= Font(family="Segoe UI", size=12, weight="bold"), command=self.voice)

        self.btn\_voice.place(x=400, y=420)

        # Label nhận diện giọng nói

        self.label\_nhan\_dien = tkVinh.Label(form, text='Nhận diện giọng nói',font= Font(family="Segoe UI", size=12, weight="bold"), bg='#8FBC8F')

        self.label\_nhan\_dien.place(x=400, y=460)

        # Button xác nhận giọng nói

        self.btn\_nhan\_dien = tkVinh.Button(form, text='Xác nhận',width=10,font= Font(family="Segoe UI", size=12, weight="bold"), command=self.submit\_voice)

        self.btn\_nhan\_dien.place(x=400, y=500)

        self.form.mainloop()

    def open\_file(self):

        filename = filedialog.askopenfilename(initialdir="/", title="Select A File", filetypes=(("csv files", "\*.csv"), ("all files", "\*.\*")))

        if filename:

            try:

                filename = r"{}".format(filename)

                df = pd.read\_csv(filename)

            except ValueError:

                messagebox.showerror("Thông báo", "File không đúng định dạng")

        self.table.delete(\*self.table.get\_children())

        self.table["column"] = list(df.columns)

        self.table["show"] = "headings"

        for column in self.table["column"]:

            self.table.heading(column, text=column)

        # thêm 20 dòng đầu tiên

        df\_rows = df.to\_numpy().tolist()

        for row in df\_rows[:20]:

            self.table.insert("", "end", values=row)

    def columns\_null(self):

        # Xử lý cột null

        df = self.df

        df = df.drop(['Fide id','Name','Federation','Gender','Year\_of\_birth','Title','Inactive\_flag'],axis=1)

        self.table.delete(\*self.table.get\_children())

        self.table["column"] = list(df.columns)

        self.table["show"] = "headings"

        for column in self.table["column"]:

            self.table.heading(column, text=column)

        # thêm 20 dòng đầu tiên

        df\_rows = df.to\_numpy().tolist()

        for row in df\_rows[:20]:

            self.table.insert("", "end", values=row)

    def rows\_null(self):

        # Xử lý dòng null (NaN)

        df = self.df

        df = df.drop(['Fide id','Name','Federation','Gender','Year\_of\_birth','Title','Inactive\_flag'],axis=1)

        df = df.dropna(how='any')

        # Xử lý dòng null (NaN)

        self.table.delete(\*self.table.get\_children())

        self.table["column"] = list(df.columns)

        self.table["show"] = "headings"

        for column in self.table["column"]:

            self.table.heading(column, text=column)

        # thêm 20 dòng đầu tiên

        df\_rows = df.to\_numpy().tolist()

        for row in df\_rows[:20]:

            self.table.insert("", "end", values=row)

    def zcore(self):

        df = self.df

        df = df.drop(['Fide id','Name','Federation','Gender','Year\_of\_birth','Title','Inactive\_flag'],axis=1)

        df = df.dropna(how='any')

        z = np.abs(stats.zscore(df.\_get\_numeric\_data()))

        df = df[(z < 3).all(axis=1)]

        self.table.delete(\*self.table.get\_children())

        self.table["column"] = list(z.columns)

        self.table["show"] = "headings"

        for column in self.table["column"]:

            self.table.heading(column, text=column)

        # thêm 20 dòng đầu tiên

        z\_rows = z.to\_numpy().tolist()

        for row in z\_rows[:20]:

            self.table.insert("", "end", values=row)

    def chuan\_hoa(self):

        df = self.df

        df = df.drop(['Fide id','Name','Federation','Gender','Year\_of\_birth','Title','Inactive\_flag'],axis=1)

        df = df.dropna(how='any')

        scaler = preprocessing.MinMaxScaler()

        scaler.fit(df)

        df = pd.DataFrame(scaler.transform(df), index=df.index, columns=df.columns)

        self.table.delete(\*self.table.get\_children())

        self.table["column"] = list(df.columns)

        self.table["show"] = "headings"

        for column in self.table["column"]:

            self.table.heading(column, text=column)

        # thêm 20 dòng đầu tiên

        df\_rows = df.to\_numpy().tolist()

        for row in df\_rows[:20]:

            self.table.insert("", "end", values=row)

    def histogram(self):

        df = self.df

        df = df.drop(['Fide id','Name','Federation','Gender','Year\_of\_birth','Title','Inactive\_flag'],axis=1)

        df = df.dropna(how='any')

        z = np.abs(stats.zscore(df.\_get\_numeric\_data()))

        df = df[(z < 3).all(axis=1)]

        # Vẽ biểu đồ phân phối từ z-core

        fig, axs = plt.subplots(nrows=1, ncols=3, figsize=(12, 4))

        sns.distplot(df['Standard\_Rating'], ax=axs[0])

        sns.distplot(df['Rapid\_rating'], ax=axs[1])

        sns.distplot(df['Blitz\_rating'], ax=axs[2])

        plt.subplots\_adjust(wspace=0.3)

        plt.show()

    def phan\_tan(self):

        df = self.df

        df = df.drop(['Fide id','Name','Federation','Gender','Year\_of\_birth','Title','Inactive\_flag'],axis=1)

        df = df.dropna(how='any')

        z = np.abs(stats.zscore(df.\_get\_numeric\_data()))

        df = df[(z < 3).all(axis=1)]

        # Vẽ biểu đồ phân phối từ z-core với scatter

        fig, axs = plt.subplots(nrows=1, ncols=3, figsize=(12, 4))

        sns.scatterplot(x='Standard\_Rating', y='Rapid\_rating', data=df, ax=axs[0])

        sns.scatterplot(x='Standard\_Rating', y='Blitz\_rating', data=df, ax=axs[1])

        sns.scatterplot(x='Rapid\_rating', y='Blitz\_rating', data=df, ax=axs[2])

        plt.subplots\_adjust(wspace=0.3)

        plt.show()

    def pie(self):

        df = self.df

        df = df.drop(['Fide id','Name','Federation','Gender','Year\_of\_birth','Title','Inactive\_flag'],axis=1)

        df = df.dropna(how='any')

        z = np.abs(stats.zscore(df.\_get\_numeric\_data()))

        df = df[(z < 3).all(axis=1)]

        # Sắp xếp lại theo thứ tự giảm dần

        z = z.sort\_values(by=['Standard\_Rating'], ascending=False)

        # Vẽ biểu đồ phân phối từ z-core với pie

        fig, axs = plt.subplots(nrows=1, ncols=3, figsize=(12, 4))

        z['Standard\_Rating'].value\_counts().plot.pie(ax=axs[0], autopct='%.2f')

        z['Rapid\_rating'].value\_counts().plot.pie(ax=axs[1], autopct='%.2f')

        z['Blitz\_rating'].value\_counts().plot.pie(ax=axs[2], autopct='%.2f')

        plt.subplots\_adjust(wspace=0.3)

        plt.show()

    def scatter\_3d(self):

        df = self.df

        df = df.drop(['Fide id','Name','Federation','Gender','Year\_of\_birth','Title','Inactive\_flag'],axis=1)

        df = df.dropna(how='any')

        z = np.abs(stats.zscore(df.\_get\_numeric\_data()))

        df = df[(z < 3).all(axis=1)]

        # Vẽ biểu đồ phân phối từ z-core với scatter 3D

        fig = plt.figure(figsize=(10, 10))

        ax = fig.add\_subplot(111, projection='3d')

        ax.scatter(df['Standard\_Rating'], df['Rapid\_rating'], df['Blitz\_rating'], c='skyblue', s=60)

        ax.view\_init(30, 185)

        plt.show()

    # def voice(self):

    #     r\_vinh84 = vinh84.Recognizer()

    #     with vinh84.Microphone() as source:

    #         text\_audio = r\_vinh84.record(source, duration=5)

    #         try:

    #             text\_audio = r\_vinh84.recognize\_google(text\_audio, language="vi-VN")

    #             self.label\_nhan\_dien.config(text=text\_audio)

    #             self.label\_nhan\_dien.update()

    #         except:

    #             self.label\_nhan\_dien.config(text="Không nhận dạng được")

    #             self.label\_nhan\_dien.update()

    def voice(self):

        r\_vinh84 = vinh84.Recognizer()

        with vinh84.Microphone() as source:

            text\_audio = r\_vinh84.record(source, duration=5)

            try:

                text\_audio = r\_vinh84.recognize\_google(text\_audio, language="vi-VN")

                # Tìm kiếm các số trong đoạn văn bản đã nhận dạng

                numbers = re.findall(r'\d+', text\_audio)

                # Nếu tìm thấy ít nhất một số, hiển thị số đầu tiên lên nhãn

                if len(numbers) > 0:

                    self.label\_nhan\_dien.config(text=numbers[0])

                else:

                    self.label\_nhan\_dien.config(text="Không nhận dạng được số")

                    self.label\_nhan\_dien.update()

            except:

                self.label\_nhan\_dien.config(text="Không nhận dạng được")

                self.label\_nhan\_dien.update()

    def submit\_voice(self):

        if self.label\_nhan\_dien.cget("text") == "Không nhận dạng được":

            messagebox.showinfo("Thông báo", "Không nhận dạng được")

        elif self.label\_nhan\_dien.cget("text") == "1":

            self.open\_file()

        elif self.label\_nhan\_dien.cget("text") == "2":

            self.columns\_null()

        elif self.label\_nhan\_dien.cget("text") == "3":

            self.rows\_null()

        elif self.label\_nhan\_dien.cget("text") == "4":

            self.zcore()

        elif self.label\_nhan\_dien.cget("text") == "5":

            self.chuan\_hoa()

        elif self.label\_nhan\_dien.cget("text") == "6":

            self.histogram()

        elif self.label\_nhan\_dien.cget("text") == "7":

            self.phan\_tan()

        elif self.label\_nhan\_dien.cget("text") == "8":

            self.pie()

        elif self.label\_nhan\_dien.cget("text") == "9":

            self.scatter\_3d()

        else:

            messagebox.showinfo("Thông báo", "Không nhận dạng được")

def mainEDA():

    EDA(tkVinh.Tk())

if \_\_name\_\_ == '\_\_main\_\_':

    mainEDA()

### 4.2.3 Frame: G3\_84LeThanhVinh\_ChessWomen\_Frame.py

'''

NGười viết: Lê THành VInh

MSSV: 2110940

Ngày viết: 19/04/2023

Mô tả: Báo cáo cuối kì: EDA Chess - phần Frame

'''

#  B1: Nạp thư viện

import cv2

import tkinter as tkVinh

import time

from PIL import Image

from tkinter import Canvas

from tkinter import \*

from tkinter import filedialog

from tkinter import messagebox as msgBox

# B2: Hàm khởi tạo Clips cắt Frames

def OpenFile():

    global filepath

    # Hộp thoại Mở thư mục

    filepath = filedialog.askopenfilename(initialdir = "all", title = "Select a File", filetypes = (("mp4 files", "\*.mp4"), ("all files", "\*.\*")))

    # Hiển thị đường dẫn file video

    txtPath.insert(0, filepath)

# B3: Hàm khởi tạo thư mục lưu Frames

def OpenFolder():

    global folderpath

    # Hộp thoại Mở thư mục

    folderpath = filedialog.askdirectory()

    # Hiển thị đường dẫn thư mục lưu Frames

    txtFolder.insert(0, folderpath)

# B4: Hàm lưu Frames vào thư mục

def SaveFrames():

    # Mở video

    cap = cv2.VideoCapture(filepath)

    # Kiểm tra video có mở được hay không

    if (cap.isOpened()==False):

        msgBox.showerror("Error", "Không thể mở file video")

    count = 0

    # Đọc video

    while (cap.isOpened()):

        ret, frame = cap.read()

        if ret == True:

            cv2.imshow('Video', frame)

            cv2.imwrite(folderpath + "/frame%d.jpg" % count, frame)

            count += 1

            # Nhấn phím ESC để thoát

            if cv2.waitKey(25) & 0xFF == 27:

                break

        else:

            break

    # Đóng video

    cap.release()

    cv2.destroyAllWindows()

# B5: Hàm chọn thời gian bắt đầu cắt Frames (tính bằng s) lấy từ textbox

def StartCut():

    # Lấy giá trị từ textbox

    txtStart = txtStartCut.get()

    # Hiển thị giá trị

    print(txtStart)

# B6: Hàm chọn thời gian kết thúc cắt Frames (tính bằng s) lấy từ textbox

def EndCut():

    # Lấy giá trị từ textbox

    txtEnd = txtEndCut.get()

    # Hiển thị giá trị

    print(txtEnd)

# B7: Hàm lưu Frames cắt được vào thư mục với thời gian bắt đầu lấy từ txtStartCut và thời gian kết thúc lấy từ txtEndCut

def SaveFramesCut():

    # Mở video

    cap = cv2.VideoCapture(filepath)

    # Kiểm tra video có mở được hay không

    if (cap.isOpened()==False):

        msgBox.showerror("Error", "Không thể mở file video")

    count = 0

    # Đọc video trong khoảng thời gian cắt (khoảng thời gian bắt đầu và kết thúc)

    while (cap.isOpened()):

        star = int(txtStartCut.get())

        end = int(txtEndCut.get())

        ret, frame = cap.read()

        if ret == True:

            cv2.imshow('Video', frame)

            if count >= star and count <= end:

                cv2.imwrite(folderpath + "/frame%d.jpg" % count, frame)

            count += 1

            # Nhấn phím ESC để thoát

            if cv2.waitKey(25) & 0xFF == 27:

                break

        else:

            break

    # Đóng video

    cap.release()

    cv2.destroyAllWindows()

# B8: Hàm xử lí ảnh lật dọc

def FlipVertical():

    # Mở video

    cap = cv2.VideoCapture(filepath)

    # Kiểm tra video có mở được hay không

    if (cap.isOpened()==False):

        msgBox.showerror("Error", "Không thể mở file video")

    count = 0

    # Đọc video

    while (cap.isOpened()):

        star = int(txtStartCut.get())

        end = int(txtEndCut.get())

        ret, frame = cap.read()

        if ret == True:

            cv2.imshow('Video', frame)

            if count >= star and count <= end:

                cv2.imwrite(folderpath + "/frame%d.jpg" % count, cv2.flip(frame, 0))

            count += 1

            # Nhấn phím ESC để thoát

            if cv2.waitKey(25) & 0xFF == 27:

                break

        else:

            break

    # Đóng video

    cap.release()

    cv2.destroyAllWindows()

# B9: Hàm xử lí ảnh thêm màu cho ảnh

def AddColor():

    # Mở video

    cap = cv2.VideoCapture(filepath)

    # Kiểm tra video có mở được hay không

    if (cap.isOpened()==False):

        msgBox.showerror("Error", "Không thể mở file video")

    count = 0

    # Đọc video

    while (cap.isOpened()):

        star = int(txtStartCut.get())

        end = int(txtEndCut.get())

        ret, frame = cap.read()

        if ret == True:

            cv2.imshow('Video', frame)

            if count >= star and count <= end:

                cv2.imwrite(folderpath + "/frame%d.jpg" % count, cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV))

            count += 1

            # Nhấn phím ESC để thoát

            if cv2.waitKey(25) & 0xFF == 27:

                break

        else:

            break

    # Đóng video

    cap.release()

    cv2.destroyAllWindows()

# Hàm khởi tạo form

def GUI():

    # Khởi tạo form

    form = tkVinh.Tk()

    # Đặt tiêu đề cho form

    form.title("84 - Lê THành Vinh")

    # Đặt kích thước cho form

    form.geometry("1280x720")

    # Khởi tạo các control trên form

    # Label

    lblPath = tkVinh.Label(form, text="Đường dẫn file video: ")

    lblPath.place(x=10, y=10)

    lblFolder = tkVinh.Label(form, text="Đường dẫn thư mục lưu Frames: ")

    lblFolder.place(x=10, y=40)

    lblStartCut = tkVinh.Label(form, text="Thời gian bắt đầu cắt Frames (Frames): ")

    lblStartCut.place(x=10, y=70)

    lblEndCut = tkVinh.Label(form, text="Thời gian kết thúc cắt Frames (Fames): ")

    lblEndCut.place(x=10, y=100)

    # Button

    btnPath = tkVinh.Button(form, text="Chọn file video", command=OpenFile)

    btnPath.place(x=500, y=10)

    btnFolder = tkVinh.Button(form, text="Chọn thư mục lưu Frames", command=OpenFolder)

    btnFolder.place(x=500, y=40)

    btnSaveFrames = tkVinh.Button(form, text="Lưu Frames All", command=SaveFrames)

    btnSaveFrames.place(x=10, y=130)

    btnSaveFramesCut = tkVinh.Button(form, text="Lưu Frames Cut", command=SaveFramesCut)

    btnSaveFramesCut.place(x=10, y=160)

    btnLatDoc = tkVinh.Button(form, text="Lật dọc các Frames", command=FlipVertical)

    btnLatDoc.place(x=10, y=190)

    btnAddColor = tkVinh.Button(form, text="Thêm màu cho các Frames", command=AddColor)

    btnAddColor.place(x=10, y=220)

    # Textbox

    global txtPath

    txtPath = tkVinh.Entry(form, width=50)

    txtPath.place(x=150, y=10)

    global txtFolder

    txtFolder = tkVinh.Entry(form, width=50)

    txtFolder.place(x=150, y=40)

    global txtStartCut

    txtStartCut = tkVinh.Entry(form, width=50)

    txtStartCut.place(x=150, y=70)

    global txtEndCut

    txtEndCut = tkVinh.Entry(form, width=50)

    txtEndCut.place(x=150, y=100)

    # Hiển thị form

    form.mainloop()

# Hàm chính

if \_\_name\_\_ == "\_\_main\_\_":

    GUI()

### 4.2.4 Game: G3\_84LeThanhVinh\_ChessWomen\_Game.py

'''

NGười viết: Lê THành VInh

MSSV: 2110940

Ngày viết: 19/04/2023

Mô tả: Báo cáo cuối kì: EDA Chess - phần Game

'''

import pygame, sys, random

from pygame.locals import \*

#PHẦN 1: ĐỊNH NGHĨA CÁC THAM SỐ

WINDOWWIDTH = 400

WINDOWHEIGHT = 600

# B2: khởi tạo thư viện pygame

pygame.init()

#B3: Lập cửa sổ game (400, 300). DISPLAYSURF là 1 biến kiểu surface (khung đen đen)

FPS = 60

fpsClockVinh = pygame.time.Clock()

#PHẦN 2: NỀN GAME

#TỐC ĐỘ CUỘN NỀN

BGSPEED = 1.5

BGIMG = pygame.image.load('GameVinh/background.png')

# LAYER (SURFACE) NỀN

DISPLAYSURF = pygame.display.set\_mode((WINDOWWIDTH, WINDOWHEIGHT))

pygame.display.set\_caption('84 - Lê Thành Vinh - 7.5 Game Xe Đua')

# LỚP HÌNH NỀN = CUỘN NỀN

class Background():

    def \_\_init\_\_(self):

        self.x = 0

        self.y = 0

        self.speed = BGSPEED

        self.img = BGIMG

        self.width = self.img.get\_width()

        self.height = self.img.get\_height()

    def draw(self):

        DISPLAYSURF.blit(self.img, (self.x, self.y))

        DISPLAYSURF.blit(self.img, (self.x, self.y - self.height))

    def update(self):

        self.y += self.speed

        if self.y >= self.height:

            self.y = 0

#PHẦN 3: XE TRONG GAME

#KÍCH THƯỚC XE

X\_MARGIN = 80

CARWIDTH = 40

CARHEIGHT = 80

CARSPEED = 10

CARIMG = pygame.image.load('GameVinh/car.png')

#LỚP XE TRONG GAME

class Car84():

    def \_\_init\_\_(self):

        self.width = CARWIDTH

        self.height = CARHEIGHT

        self.x = (WINDOWWIDTH - self.width) / 2

        self.y = (WINDOWHEIGHT - self.height) / 2

        self.speed = CARSPEED

        self.surfaceVinh = pygame.Surface((self.width, self.height))

        self.surfaceVinh.fill((255, 255, 255))

    def draw(self):

        DISPLAYSURF.blit(CARIMG, (int(self.x), int(self.y)))

    def update(self, moveLeft, moveRight, moveUp, moveDown):

        if moveLeft == True:

            self.x -= self.speed

        if moveRight == True:

            self.x += self.speed

        if moveUp == True:

            self.y -= self.speed

        if moveDown == True:

            self.y += self.speed

        if self.x < X\_MARGIN:

            self.x = X\_MARGIN

        if self.x + self.width > WINDOWWIDTH - X\_MARGIN:

            self.x = WINDOWWIDTH - X\_MARGIN - self.width

        if self.y < 0:

            self.y = 0

        if self.y + self.height > WINDOWHEIGHT:

            self.y = WINDOWHEIGHT - self.height

#PHẦN 4: XE CHƯỚNG NGẠI VẬT = XE NGƯỢC CHIỀU:obstacles ##

LANEWIDTH = 60

DISTANCE = 200

OBSTACLESSPEED = 10

CHANGESPEED = 0.001

OBSTACLESIMG = pygame.image.load('GameVinh/obstacles.png')

class Obstacles():

    def \_\_init\_\_(self):

        self.width = CARWIDTH

        self.height = CARHEIGHT

        self.distance = DISTANCE

        self.speed = OBSTACLESSPEED

        self.changeSpeed = CHANGESPEED

        self.ls = []

        for i in range(5):

            y = -CARHEIGHT-i\*self.distance

            lane = random.randint(0, 3)

            self.ls.append([lane, y])

    def draw(self):

        for i in range(5):

            x = int(X\_MARGIN + self.ls[i][0]\*LANEWIDTH + (LANEWIDTH-self.width)/2)

            y = int(self.ls[i][1])

            DISPLAYSURF.blit(OBSTACLESIMG, (x, y))

    def update(self):

        for i in range(5):

            self.ls[i][1] += self.speed

        self.speed += self.changeSpeed

        if self.ls[0][1] > WINDOWHEIGHT:

            self.ls.pop(0)

            y = self.ls[3][1] - self.distance

            lane = random.randint(0, 3)

            self.ls.append([lane, y])

#PHẦN 5: TÍNH ĐIỂM

class Score():

    def \_\_init\_\_(self):

        self.score = 0

        self.font = pygame.font.SysFont('sans', 20)

    def draw(self):

        font = pygame.font.SysFont('consolas', 30)

        scoreSuface = font.render('Score: '+str(int(self.score)), True, (0, 0, 0))

        DISPLAYSURF.blit(scoreSuface, (10, 10))

    def update(self):

        self.score += 0.02

# PHẦN 6: XỬ LÝ VA CHẠM: Collision

def rectCollision(rect1, rect2):

    if rect1[0] <= rect2[0]+rect2[2] and rect2[0] <= rect1[0]+rect1[2] and rect1[1] <= rect2[1]+rect2[3] and rect2[1] <= rect1[1]+rect1[3]:

        return True

    return False

def isGameover(car84, obstacles):

    carRect = [car84.x, car84.y, car84.width, car84.height]

    for i in range(5):

        x = int(X\_MARGIN + obstacles.ls[i][0]\*LANEWIDTH + (LANEWIDTH-obstacles.width)/2)

        y = int(obstacles.ls[i][1])

        obstaclesRect = [x, y, obstacles.width, obstacles.height]

        if rectCollision(carRect, obstaclesRect) == True:

            return True

    return False

    #PHẦN 7: CÁC THỦ TỤC CHƠI GAME

def gameOver(bg, car84, obstacles, score):

    font = pygame.font.SysFont('consolas', 60)

    headingSuface = font.render('GAMEOVER', True, (255, 0, 0))

    headingSize = headingSuface.get\_size()

    font = pygame.font.SysFont('consolas', 20)

    commentSuface = font.render('Press "space" to replay', True, (0, 0, 0))

    commentSize = commentSuface.get\_size()

    while True:

            for event in pygame.event.get():

                if event.type == pygame.QUIT:

                    pygame.quit()

                    sys.exit()

                if event.type == pygame.KEYUP:

                    if event.key == K\_SPACE:

                        return

            bg.draw()

            car84.draw()

            obstacles.draw()

            score.draw()

            DISPLAYSURF.blit(headingSuface, (int((WINDOWWIDTH - headingSize[0])/2), 100))

            DISPLAYSURF.blit(commentSuface, (int((WINDOWWIDTH - commentSize[0])/2), 400))

            pygame.display.update()

            fpsClockVinh.tick(FPS)

def gameStart(bg):

        bg.\_\_init\_\_()

        font = pygame.font.SysFont('consolas', 60)

        headingSuface = font.render('RACING', True, (255, 0, 0))

        headingSize = headingSuface.get\_size()

        font = pygame.font.SysFont('consolas', 20)

        commentSuface = font.render('Press "space" to play', True, (0, 0, 0))

        commentSize = commentSuface.get\_size()

        while True:

            for event in pygame.event.get():

                if event.type == pygame.QUIT:

                    pygame.quit()

                    sys.exit()

            if event.type == pygame.KEYUP:

                if event.key == K\_SPACE:

                    return

            bg.draw()

            DISPLAYSURF.blit(headingSuface, (int((WINDOWWIDTH - headingSize[0])/2), 100))

            DISPLAYSURF.blit(commentSuface, (int((WINDOWWIDTH - commentSize[0])/2), 400))

            pygame.display.update()

            fpsClockVinh.tick(FPS)

def gamePlay(bg, car84, obstacles, score):

        car84.\_\_init\_\_()

        obstacles.\_\_init\_\_()

        bg.\_\_init\_\_()

        score.\_\_init\_\_()

        moveLeft = False

        moveRight = False

        moveUp = False

        moveDown = False

        while True:

            for event in pygame.event.get():

                if event.type == pygame.QUIT:

                    pygame.quit()

                    sys.exit()

                if event.type == KEYDOWN:

                    if event.key == K\_LEFT:

                        moveLeft = True

                    if event.key == K\_RIGHT:

                        moveRight = True

                    if event.key == K\_UP:

                        moveUp = True

                    if event.key == K\_DOWN:

                        moveDown = True

                if event.type == KEYUP:

                    if event.key == K\_LEFT:

                        moveLeft = False

                    if event.key == K\_RIGHT:

                        moveRight = False

                    if event.key == K\_UP:

                        moveUp = False

                    if event.key == K\_DOWN:

                        moveDown = False

            if isGameover(car84, obstacles):

                return

            bg.draw()

            bg.update()

            car84.draw()

            car84.update(moveLeft, moveRight, moveUp, moveDown)

            obstacles.draw()

            obstacles.update()

            score.draw()

            score.update()

            pygame.display.update()

            fpsClockVinh.tick(FPS)

def main():

    bg = Background()

    car84 = Car84()

    obstacles = Obstacles()

    score = Score()

    gameStart(bg)

    while True:

        gamePlay(bg, car84, obstacles, score)

        gameOver(bg, car84, obstacles, score)

if \_\_name\_\_ == '\_\_main\_\_':

    main()

1. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỒ ÁN HỌC PHẦN

## Kết luận

## Những kết quả đạt được

ĐỒ ÁN HỌC PHẦN đã thực hiện việc Lập trình Python LẬP TRÌNH PYTHON PHÂN TÍCH DỮ LIỆU THĂM DÒ (EDA) VỀ “TOP WOMEN CHESS PLAYER”; bao gồm:

+ Giao diện đồ họa: GUI

+ Xử lý giọng nói trợ lý ảo: Voice Assistant

+ Vẽ đồ thị

+ Phân tích khám phá trên tập dữ liệu thực nghiệm: G3\_84LeThanhVinh\_ChessWomen.csv

+ Cắt Frame video

+ Game

5.1.2 Hạn chế

Vì thời gian có hạn nên vẫn chưa trao chuốt đồng bộ toàn bộ giao diện

## Hướng phát triển

## Hướng khắc phục các hạn chế

Những đồ án sau, em sẽ lên kế hoạch và phân chia công việc hợp lý, tập trung vào các công việc quan trọng nhất trước, hoặc tìm kiếm sự hỗ trợ từ giảng viên ngay khi ấy để có thể hoàn thiện một cách tốt nhất

## Hướng mở rộng ĐỒ ÁN HỌC PHẦN

Em sẽ cải thiện đồ án của mình bằng cách thêm các tính năng mới như tăng sự tương tác với người dùng. Em sẽ tìm hiểu và áp dụng các công nghệ mới như Machine Learning và Deep Learning để tăng cường khả năng phân tích của đồ án

# DANH MỤC TÀI LIỆU THAM KHẢO

1. Võ Xuân Thể (2023), Tài liệu học tập Lập trình Python, Trường Đại học Sư phạm kỹ thuật Tp.HCM, Tài liệu lưu hành nội bộ được cung cấp theo từng buổi học
2. Võ Xuân Thể (2023), Bài tập thực hành Lập trình Python, Trường Đại học Sư phạm kỹ thuật Tp.HCM, Tài liệu lưu hành nội bộ được cung cấp theo từng buổi học
3. Dataframe: https://www.kaggle.com/datasets/vikasojha98/top-women-chess-players/code đăng bởi VIKAS OJHA. *Top Women Chess Players - Dataset of top women chess players by their FIDE rating (August 2020)* Truy cập vào tháng 5/2023

1. IDE (Integrated Development Environment) là môi trường tích hợp dùng để viết code để phát triển ứng dụng [↑](#footnote-ref-0)