**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\_02CLC**

Tên chủ đề EDA: **EDA PHÂN TÍCH THĂM DÒ VỀ CHỈ SỐ CƠ BẢN CỦA POKEMON5**

Chủ đề CV: HỖ TRỢ CV NHẬN DIỆN NGÔN NGỮ KÍ HIỆU

Chủ đề Game: LẬP TRÌNH PYTHON CAR RACING

**Họ tên sinh viên: Nguyễn Thành Đạt [13]**

**Mã số sinh viên: 22110129**

**Lớp: 22110CL4A [CQ]**

**Ngày nộp: ……./……./2023**

**Ký tên:**

**TP.HCM, ngày .... Tháng .... 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\_05CLC**

Tên đề tài: **EDA phân tích thăm dò về việc làm và mức lương trong ngành phát triển phần mềm năm 2024**

Chủ đề CV: LẬP TRÌNH PYTHON HỖ TRỢ CV NHẬN DIỆN TRÁI CÂY

Chủ đề Game: LẬP TRÌNH PYTHON GAME CAR RACING

**Giảng viên giảng dạy: VÕ XUÂN THỂ**

**Họ tên sinh viên: Nguyễn Thành Đạt [13]**

**Mã số sinh viên: 22110129**

**Lớp: 221101CL4A [CQ]**

**TÊN CÁC FILE SẢN PHẨM ĐỀ TÀI:**

Thư mục bài làm: **G213NTD**

Tập tin báo cáo bài làm: **2\_G213NguyenThanhDat.docx**

Tập tin mã nguồn: **G213NguyenThanhDat.py**

Thư mục bài làm phần EDA: **G213NguyenThanhDat\_EDA**

Tập dữ liệu thực nghiệm EDA: POKEMON\_COMBINED\_csv (nên sử dụng Keysearch)

Thư mục bài làm phần CV: **G213NguyenThanhDat\_CV**

File Video thực nghiệm CV: sg2.mp4 (nên sử dụng Keysearch)

Thư mục bài làm phần Game: **G213NguyenThanhDat\_Game**

Giới thiệu thư mục bài làm: **1\_G213NguyenThanhDat\_Intro.docx**

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

# GUI: tkinter

# EDA: pandas, numpy, scipy, sklearn

**NHẬN XÉT CỦA GIẢNG VIÊN**

**………………………………………………………..**

**………………………………………………………..**

**Ngày……./……../202…..**

**Giảng viên: Ký tên**

**TP.HCM, ngày …. tháng …. năm 2023**

**LỜI CẢM ƠN**

H

# 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](#_Toc183276088)** [3](#_Toc183276088)

[DANH MỤC CÁC TỪ VIẾT TẮT 4](#_Toc183276089)

**[DANH MỤC CÁC THUẬT NGỮ](#_Toc183276090)** [4](#_Toc183276090)

[Chương 1: GIỚI THIỆU VỀ ĐỒ ÁN HỌC PHẦN 9](#_Toc183276091)

[1.1. Tổng quan về Đồ án học phần 9](#_Toc183276092)

[1.2. Nội dung chuyên môn chính của Đồ án học phần 9](#_Toc183276093)

[1.3. Công cụ và nền tảng kỹ thuật thực hiện Đồ án học phần 10](#_Toc183276094)

[1.4. Bố cục của báo cáo 10](#_Toc183276095)

[Chương 2: CƠ SỞ LÝ THUYẾT CỦA HỌC PHẦN 11](#_Toc183276096)

[2.1. GIỚI THIỆU VỀ LẬP TRÌNH PYTHON 11](#_Toc183276097)

[2.1.1. Giới thiệu chung 11](#_Toc183276098)

[2.1.2. Cài đặt “bộ thảo chương” (soạn thảo chương trình IDE) và “bộ dịch” 11](#_Toc183276099)

[2.1.3. Giới thiệu các bộ tương tác lập trình 11](#_Toc183276100)

[2.1.4. Cấu hình mô trường tương tác lập trình 11](#_Toc183276101)

[2.2. TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH PYTHON 11](#_Toc183276102)

[2.2.1. Danh hiệu (Identifier) 11](#_Toc183276103)

[2.2.2. Về viết mã lệnh lập trình (Code) 11](#_Toc183276104)

[2.2.3. Về chú giải (Comment) 11](#_Toc183276105)

[2.2.4. Các phép toán 11](#_Toc183276106)

[2.2.5. Chuỗi 11](#_Toc183276107)

[2.2.6. Lệnh 11](#_Toc183276108)

[2.2.7. Định nghĩa hàm 11](#_Toc183276109)

[2.2.8. Danh sách 11](#_Toc183276110)

[2.2.9. Bộ dữ liệu: Tuple 11](#_Toc183276111)

[2.2.10. Tập hợp: Set 11](#_Toc183276112)

[2.2.11. Từ điển: Dictionary 11](#_Toc183276113)

[2.2.12. Module 11](#_Toc183276114)

[2.2.13. Package 11](#_Toc183276115)

[2.2.14. Lớp 11](#_Toc183276116)

[2.2.15. Xử lý ngoại lệ 11](#_Toc183276117)

[2.3. GIỚI THIỆU CÁC THƯ VIỆN VÀ CÔNG CỤ CĂN BẢN 11](#_Toc183276118)

[2.3.1. Nạp thư viện 11](#_Toc183276119)

[2.3.2. speech\_recognition 11](#_Toc183276120)

[2.3.3. gtts 11](#_Toc183276121)

[2.3.4. playsound 11](#_Toc183276122)

[2.3.5. tkinter 11](#_Toc183276123)

[2.3.6. os 11](#_Toc183276124)

[2.3.7. Pandas 11](#_Toc183276125)

[2.3.8. Tensorflow 11](#_Toc183276126)

[2.3.9. NumPy 12](#_Toc183276127)

[2.3.10. SCIPY 12](#_Toc183276128)

[2.3.11. Theano 12](#_Toc183276129)

[2.3.12. Matplotlib 12](#_Toc183276130)

[2.3.13. scikit-learn = sklearn 12](#_Toc183276131)

[2.3.14. Keras 12](#_Toc183276132)

[2.3.15. NLTK 12](#_Toc183276133)

[2.3.16. PyTorch 12](#_Toc183276134)

[2.3.17. LightGBM 12](#_Toc183276135)

[2.3.18. Eli5 12](#_Toc183276136)

[2.4. KỸ THUẬT LẬP TRÌNH ĐỆ QUY: THÁP HÀ NỘI (HANOI TOWER) 12](#_Toc183276137)

[2.5. LẬP TRÌNH PYTHON XỬ LÝ GIỌNG NÓI (Trợ lý ảo: Voice Assistant) 12](#_Toc183276138)

[2.6. LẬP TRÌNH GIAO DIỆN ĐỒ HỌA (GUI) 12](#_Toc183276139)

[2.7. LẬP TRÌNH PHÂN TÍCH KHÁM PHÁ (EDA) [Thăm dò] 12](#_Toc183276140)

[2.8. LẬP TRÌNH ĐỒ THỊ & BIỂU ĐỒ TRỰC QUANG HÓA DỮ LIỆU (VISUALIZAION: PLOT) 12](#_Toc183276141)

[2.9. LẬP TRÌNH HỖ TRỢ KỸ THUẬT THỊ GIÁC MÁY TÍNH (CV) 12](#_Toc183276142)

[2.10. LẬP TRÌNH GAMEs CĂN BẢN 12](#_Toc183276143)

[Chương 3: PHÂN TÍCH VÀ XÁC ĐINH CÁC CƠ SỞ KỸ THUẬT 13](#_Toc183276144)

[3.1. GIỚI THIỆU CHUNG VỀ CÁC CHỦ ĐỀ CỦA ĐỒ ÁN 13](#_Toc183276145)

[3.2. KỸ THUẬT LẬP TRÌNH ĐỆ QUY: THÁP HÀ NỘI (HANOI TOWER) 13](#_Toc183276146)

[3.2.1. Mô tả bài toán 13](#_Toc183276147)

[3.2.2. Các yêu cầu bài làm 13](#_Toc183276148)

[3.2.3. Full Codes (mã lệnh lập trình) 13](#_Toc183276149)

[3.2.4. Kết quả thực nghiệm 13](#_Toc183276150)

[3.2.5. Các bài tập tương tự 30](#_Toc183276151)

[3.3. LẬP TRÌNH PYTHON XỬ LÝ GIỌNG NÓI (Trợ lý ảo: Voice Assistant) 32](#_Toc183276152)

[3.3.1. Giới thiệu Các thư viện Python liên quan 32](#_Toc183276153)

[3.3.2. Giới thiệu bài toán 32](#_Toc183276154)

[3.3.3. Yêu cầu của bài làm 33](#_Toc183276155)

[3.3.4. Full code bài làm 34](#_Toc183276156)

[3.3.5. Các bài tương tự 35](#_Toc183276157)

[3.4. LẬP TRÌNH GIAO DIỆN ĐỒ HỌA (GUI) 36](#_Toc183276158)

[3.4.1. Giới thiệu chủ đề 36](#_Toc183276159)

[3.5. LẬP TRÌNH TRỰC QUANG HÓA DỮ LIỆU (PLOT) 62](#_Toc183276160)

[Chương 4: LẬP TRÌNH PYTHON PHÂN TÍCH ........................................ 63](#_Toc183276161)

[4.1. Giới thiệu chủ đề EDA ……. 63](#_Toc183276162)

[4.2. Giới thiệu bài toán (Yêu cầu đặt ra) 63](#_Toc183276163)

[4.3. Thông tin về file dữ liệu thực nghiệm EDA 63](#_Toc183276164)

[4.4. Các nền tảng kỹ thuật liên quan 63](#_Toc183276165)

[4.4.1. Hệ thống thư viện sử dụng 63](#_Toc183276166)

[4.4.2. Tập dữ liệu thực nghiệm 64](#_Toc183276167)

[4.4.3. Mô tả giải thuật 64](#_Toc183276168)

[4.5. Thiết kế giao diện 67](#_Toc183276169)

[4.6. Mã lệnh lập trình (python) 67](#_Toc183276170)

[Chương 5: LẬP TRÌNH HỖ TRỢ KỸ THUẬT THỊ GIÁC MÁY TÍNH (CV) NHẬN DIỆN ......................TỪ CLIP….. 71](#_Toc183276171)

[5.1. Giới thiệu chủ đề lập trình 71](#_Toc183276172)

[5.2. Cắt frames ảnh từ Video Clips 71](#_Toc183276173)

[5.2.1. Giới thiệu về cắt frame ảnh: frame 71](#_Toc183276174)

[5.2.2. Full codes bài làm 71](#_Toc183276175)

[5.2.3. Yêu cầu của bài thực hành 72](#_Toc183276176)

[5.3. Xử lý ảnh 72](#_Toc183276177)

[5.3.1. Giới thiệu về xử lý ảnh: images 72](#_Toc183276178)

[5.3.2. Full codes bài làm 72](#_Toc183276179)

[5.3.3. Yêu cầu của bài thực hành 73](#_Toc183276180)

[5.4. Yêu cầu bài thực nghiệm: Xử lý ảnh trên các frames đã cắt từ Video Clips 73](#_Toc183276181)

[5.4.1. Giới thiệu các yêu cầu của bài thực nghiệm 73](#_Toc183276182)

[5.4.2. Giới thiệu các Video clip …….thực nghiệm 73](#_Toc183276183)

[5.4.3. Full codes bài làm 74](#_Toc183276184)

[5.4.4. Giới thiệu sản phẩm kết quả thực nghiệm bài làm 78](#_Toc183276185)

[Chương 6: LẬP TRÌNH PYTHON GAME ..................... 83](#_Toc183276186)

[6.1. Giới thiệu chủ đề Game ……. 83](#_Toc183276187)

[6.2. Phân tích chủ đề Game ……. 83](#_Toc183276188)

[6.3. Các nền tảng kỹ thuật Lập trình Game 83](#_Toc183276189)

[6.3.1. Các thư viện liên quan 83](#_Toc183276190)

[6.3.2. Giới thiệu Các mẫu media…….thực nghiệm (mẫu xe và đường đua) 83](#_Toc183276191)

[6.4. Mô tả giải thuật Game…. 83](#_Toc183276192)

[6.5. Giao diện Game …. 83](#_Toc183276193)

[6.6. Mã lệnh lập trình Game…… 83](#_Toc183276194)

[Chương 7: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỒ ÁN HỌC PHẦN 84](#_Toc183276195)

[7.1. Những kết quả đạt được 84](#_Toc183276196)

[7.2. Hạn chế & hướng khắc phục các hạn chế 84](#_Toc183276197)

[7.3. Hướng mở rộng ĐỒ ÁN HỌC PHẦN 84](#_Toc183276198)

[DANH MỤC TÀI LIỆU THAM KHẢO 85](#_Toc183276199)

1. **GIỚI THIỆU VỀ ĐỒ ÁN HỌC PHẦN**

## Tổng quan về Đồ án học phần

Đồ án Học phần 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) với các chủ đề:

+ Phân tích thăm dò (EDA: Epolore Data Analysing) ……… tại ……..từ ……………đến………….đối với tập dữ liệu thực nghiệm theo link: …………………..; có trực quang hóa dữ liệu dùng các kỹ thuật biểu đồ và đồ thị (Plot)

+ CV nhận diện……………..từ clip ………………

+ Games Car Racing

## 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: . thông tin nhập có 2 phương án (lời nói dạng “chọn” OR nhập ký tự)

. tin xuất “kết quả thăm dò” ở cả 2 dạng: text và lời nói

+ Phân tích dữ liệu khám phá || thăm dò [EDA] ……. tại ….. từ ….. đến….

+ Trực quang hóa dữ liệu thăm dò …………bằng biểu đồ | đồ thi.

+ Giao diện đồ họa dạng [WinForm or Webform or Mobileform]

+ CV: Thị giác máy tính cho phép nhận diện…… từ clip……

+ Game căn bản Car Racing (“Đua xe”).

Nội dung chuyên môn của Đồ án này là nền tảng cho các lĩnh vực:

1. Lập trình AI (TTNT), ML (HM), Deep Learning (HS)
2. Data Science: Lập trình Khoa học dữ liệu, đặc biệt EDA (…..): Phân tích dữ liệu thăm dò
3. CV: Lập trình Thị giác máy tính
4. Games: Lập trình Games
5. Data Visualization = Plot: Lập trình trực quang hóa dữ liệu, dùng biểu đồ & đồ thi
6. GUI (….): Lập trình giao diện đồ họa

## 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: **Spyder 4 || 5 (Anaconda3)**

+ Các thư viện chính:

# B1: NẠP THƯ VIỆN

# Speech

import speech\_recognition as sr

from gtts import gTTS

import playsound

# tkinter

import tkinter as tk

from tkinter import messagebox

# 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

## 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 Phân tích ........................................: DataSetName

Chương 5: Lập trình CV nhận diện ........................................

Chương 6: Lập trình Game ........................................

Chương 7: 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.

1. **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

### speech\_recognition

### gtts

### playsound

### tkinter

### os

### Pandas

### Tensorflow

### NumPy

### SCIPY

### Theano

### Matplotlib

### scikit-learn = sklearn

### Keras

### NLTK

### PyTorch

### LightGBM

### Eli5

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

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

## LẬP TRÌNH GIAO DIỆN ĐỒ HỌA (GUI)

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

## LẬP TRÌNH ĐỒ THỊ & BIỂU ĐỒ TRỰC QUANG HÓA DỮ LIỆU (VISUALIZAION: PLOT)

## LẬP TRÌNH HỖ TRỢ KỸ THUẬT THỊ GIÁC MÁY TÍNH (CV)

## LẬP TRÌNH GAMEs CĂN BẢN

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

## GIỚI THIỆU CHUNG VỀ CÁC CHỦ ĐỀ CỦA ĐỒ ÁN

Đồ án này gồm 3 chủ đề chính:

+ Lập trình EDA thăm dò….. tại ……từ ….. đến…………[Chương 4:]; có trực quang hóa dữ liệu dùng Plot vẽ đồ thị và biểu đồ [3.4].

+ Lập trình CV hỗ trợ nhận diện …………….từ clip……………[Chương 5:]

+ Lập trình Game căn bản: Car Racing [Chương 6:]

Được thực hiện trên hệ thống giao diện GUI [3.3] với cơ chế nhập thông tin vào (input) và xuất kết quả (output) theo hình thức Text và Voice nhờ xử lý “Trợ lý ảo” [3.2].

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

### Mô tả bài toán

+ ….

A pixel art of a pyramid

Description automatically generated

### Các yêu cầu bài làm

Lập trình đệ quy bài toán Tháp Hà Nội: HaNoi Tower

# YÊU CẦU BÀI TẬP HaNoiTower

#1. Cá nhân hóa thông tin bài làm đối với các Biến, Hàm, Hằng, Bí danh,...

thông tin cá nhân hóa = sv tự chọn, VD; stt, tên, họ, viết tắt họ tên,...

#2. Cho NSD chọn số lương "Đĩa" => Comment # cho biết số "đĩa" tối đa chạy được

n = int(input("Quý vị nhap so Dia: ")) kiểu int

#3. In ra thời gian chạy ? import time

#4. kiểm tra số lượng đĩa nhập trong #2 hợp lệ: n >0 && số nguyên [nên lặp = có thoát]

#5. IN RA SỐ THỨ TỰ LƯỢT CHUYỂN "ĐĨA"

### Full Codes (mã lệnh lập trình)

def HNT\_13NguyenThanhDat(n , source, destination, auxiliary):

    if n==1:

        print ("Move disk 1 from source",source,"to destination",destination)

        return

    HNT\_13NguyenThanhDat(n-1, source, auxiliary, destination)

    print ("Move disk+",n,"+from source",source,"to destination",destination)

    HNT\_13NguyenThanhDat(n-1, auxiliary, destination, source)

# Driver code

def main():

    n = int(input("Hay chon so vong ma ban muon: "))

    HNT\_13NguyenThanhDat(n,'A','B','C')

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

    main()

### Kết quả thực nghiệm

**+ Giá trị thực nghiệm: số đĩa**

. n = 9

. n max = 65555.. ; kiểu dữ liệu int

KẾT QUẢ:

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 4 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 5 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 4 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 6 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 4 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 5 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 4 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 7 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 4 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 5 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 4 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 6 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 4 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 5 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 4 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 8 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 4 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 5 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 4 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 6 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 4 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 5 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 4 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 7 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 4 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 5 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 4 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 6 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 4 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 5 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 4 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 9 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 4 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 5 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 4 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 6 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 4 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 5 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 4 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 7 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 4 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 5 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 4 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 6 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 4 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 5 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 4 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 8 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 4 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 5 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 4 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 6 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 4 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 5 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 4 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 7 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 4 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 5 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 4 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 6 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 4 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 3 +from source B to destination C

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 5 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 3 +from source C to destination A

Move disk 1 from source B to destination C

Move disk+ 2 +from source B to destination A

Move disk 1 from source C to destination A

Move disk+ 4 +from source C to destination B

Move disk 1 from source A to destination B

Move disk+ 2 +from source A to destination C

Move disk 1 from source B to destination C

Move disk+ 3 +from source A to destination B

Move disk 1 from source C to destination A

Move disk+ 2 +from source C to destination B

Move disk 1 from source A to destination B

TỔNG SỐ LẦN CHUYỂN: **498**

**+ Cấu hình máy tính 1:**

Device name Dell Vostro 15 3510

Processor 11th Gen Intel(R) Core(TM) i5-1135G7 2.4 GHz

Installed RAM 16.0 GB (15.7 GB usable)

Device ID F5E8BA96-B880-4084-84D5-0DFD1D78D9BD

Product ID 00330-80000-00000-AA555

System type x64-based PC

Edition Windows 11 Pro

Version 24H2

Installed on ‎10/‎2/‎2024

OS build 26100.1882

Experience Windows Feature Experience Pack 1000.26100.23.0

THỜI GIAN THỰC HIỆN: n = 9 là : ……………

**+ Cấu hình máy tính 2:**

**….**

### Các bài tập tương tự

Cá nhân hóa thông tin trong bài làm

Bài 1.1. **In danh sách 100 số nguyên tố đầu tiên** (chỉ chia hết cho 1 và chính nó: 2, ,3, 5, 7, 11, 13, 17,…)

#Sàng số nguyên tố

import math

SNT\_13NguyenThanhDat = [True] \* 100

def SNT\_13NTD():

    SNT\_13NguyenThanhDat[0] = SNT\_13NguyenThanhDat[1] = False

    for i in range(2, int(math.sqrt(100))):

        if SNT\_13NguyenThanhDat[i]:

            for j in range(i\*i, 100, i):

                SNT\_13NguyenThanhDat[j] = False

def main():

    SNT\_13NTD()

    for i in range(100):

        if SNT\_13NguyenThanhDat[i] == True:

            print(i)

main()

Bài 1.2. **Viết hàm đảo ngược số:** 123 => 321

#Đảo ngược số

def G13NTD\_DaoNguocSo(n):

    m = 0

    while n != 0:

        m = m\*10 + (n%10)

        n //= 10

    return m

def main():

    m = int(input("Nhap so n: "))

    print(G13NTD\_DaoNguocSo(m))

main()

Bài 1.3.1 **Viết hàm giải PT Bậc 1**

a\_13NTD = float(input("a = "))

b\_13NTD = float(input("b = "))

print("{0}x + {1} = 0".format(a\_13NTD, b\_13NTD))

if a\_13NTD == 0 and b\_13NTD == 0:

    print("Phuong trinh vo so nghiem")

elif a\_13NTD == 0 and b\_13NTD != 0:

    print("Phuong trinh vo nghiem")

else:

    x\_13NTD = -b\_13NTD/a\_13NTD

    print("Nghiem cua phuong trinh la " + str(x\_13NTD))

Bài 1.3.2 **Bậc 2,…**

import math

a\_13NTD = float(input("a = "))

b\_13NTD = float(input("b = "))

c\_13NTD = float(input("c = "))

print("{0}x^2 + {1}x + {2} = 0".format(a\_13NTD, b\_13NTD, c\_13NTD))

if a\_13NTD == 0:

    print("Day khong phai phuong trinh bac hai")

else:

    delta = math.pow(b\_13NTD, 2) - 2\*a\_13NTD\*c\_13NTD

    if delta < 0:

        print("Phuong trinh vo nghiem")

    elif delta == 0:

        x = -b\_13NTD/(2\*a\_13NTD)

        print("Phuong trinh co nghiem kep x = ", +str(x))

    else:

        x1 = (-b\_13NTD-math.sqrt(delta))/(2\*a\_13NTD)

        x2 = (-b\_13NTD+math.sqrt(delta))/(2\*a\_13NTD)

        print("Phuong trinh co 2 nghiem x1 = {0} va x2 = {1}".format(x1, x2))

Bài 1.4. **Viết hàm đọc số:**  123 => Một Hai Ba

G213NTD\_digit\_to\_word = {

    '0': 'Không', '1': 'Một', '2': 'Hai', '3': 'Ba', '4': 'Bốn',

    '5': 'Năm', '6': 'Sáu', '7': 'Bảy', '8': 'Tám', '9': 'Chín'

}

def G213NTD\_Read\_Number(number):

    num\_str = str(number)

    words = [G213NTD\_digit\_to\_word[digit] for digit in num\_str]

    return ' '.join(words)

G213NTD\_number\_input = input("Nhập số: ")

if G213NTD\_number\_input.isdigit():

    print(G213NTD\_Read\_Number(G213NTD\_number\_input))

else:

    print("Vui lòng nhập một số hợp lệ.")

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

### Giới thiệu Các thư viện Python liên quan

speech\_recognition: nhận diện giọng nói

gTTS: google

playsound:

Ngoài ra, thêm:

os : thư viện hỗ trợ lưu file, mở file, đường dẫn,..

time: thư viện lấy thời gian = đồng hồ

google\_trans\_new: thư viện hỗ trợ mô phỏng Google Translate.

### Giới thiệu bài toán

+ 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

Part<các chuyên môn ứng dụng> ..: Các kỹ thuật AI, . . .

Part2: Voice….: chính là part2 của chủ đề này

**+ Bài tập theo HP: Lập trình App xử lý "lời nới" (Speech): với 2 giai đoạn (part)**

+Part 1: Nghe tiếng Việt => Text

+Part2: Nhập vào Text và nói ra bằng tiếng Việt : Text => Nói tiếng Việt

+ Vận dụng: thực hiện Game “My Talking Tom” = Part1 + Part2

**[1] Nghe tiếng Việt => Text**

B1: Cài đặt thư viện

conda install …. Các thư viện sau:

------------------gốc hướng dẫn / dùng python trong visual (pip3)

pip3 install SpeechRecognition

*pip3 install SpeechRecognition pydub # Long audio source*

pip3 install pyaudio

B2 : Nạp thư viện vào App

import speech\_recognition as sr

B3 : Chọn nguồn âm thanh nhập :

+ Từ file âm thanh

Tham khảo English : https://github.com/x4nth055/pythoncode-tutorials/find/master

filename = "16-122828-0002.wav"

+ Từ Microphone

# initialize the recognizer

r = sr.Recognizer()

B4 : Xử lý nhận diện: từ MIC OR Files

….

### Yêu cầu của bài làm

Voice Assistant (Lập trình xử lý giọng nói [Âm thanh: Audio] = Trợ lý “ảo”)

# YÊU CẦU BÀI TẬP = B2Ex2\_VoiceAssistant

#. Cá nhân hóa thông tin bài làm đối với các Biến, Hàm, Hằng, Bí danh,...

# thông tin cá nhân hóa = sv tự chọn, VD; stt, tên, họ, viết tắt họ tên,...

#. Thư viện:

# google\_tran\_new & googletrans[chuyển ngữ, vd vi <=> en]

# os [xóa, rename files]

# time[lấy dmyhms->đặt tên file: tránh bị trùng tên]

#[Phần 1: Nói -> Text]

#Cho NSD chọn các thông tin input (vd: số n, thời gian,...) & hình thức output (vd: Text, âm thanh,...)

# 1. Cho NSD chọn cách nhập âm thanh vào: từ MIC hay từ File .mp3, .wav

# 2. Cho NSD chọn thời gian chờ (kiểm tra NSD nhập từ 0->...)

# 3. Cho NSD chọn thời gian nghe (kiểm tra NSD nhập từ 1->...)

# 4. Cho NSD chọn ngôn ngữ dùng (Tự tìm 2 ký tự chọn ngôn ngữ, VD vi, en,...)

# 5. Phát âm thanh song song với text khi = yêu cầu nhập lựa chọn..

#[Phần 2: Text->Nói]

# 6. Cho NSD nhập tên file lưu âm thanh:

# -> cần kiểm tra: trong thư mục hiện tại đã có file đó chưa: Nếu chưa -> cho lưu

# -> Nếu đã có: cho NSD chọn phương án xử lý (dùng thư viện os): xóa file cũ OR remane

# 6.1. Lưu chồng lên file đã có

# 6.2. Tự động đổi tên file cũ ->..kèm yymmddhhmmss (dùng thư viện time): nhớ thông báo

# 6.3. Tự động đổi tên file mới ->..kèm yymmddhhmmss (dùng thư viện time): nhớ thông báo

# 6.4. Chon NSD đổi tên các file (để sẵn tên file đã có); nhớ kiểm tra phải khác nhau.

# 7. Chuyển ngữ: nói vào vi => đọc ra en (dùng thư viện googletran...)

# 7.1. Chọn ngôn ngữ vào; VD vi

# 7.2. Chọn ngôn ngữ ra: VD en

# ví dụ: tiếng Ả rập, tiếng Anh, Tiếng Hàn, tiếng Hoa, tiếng Nhật.

# [mượn translate.google.com : input speech]

# chú ý thêm : giọng Nam or Nữ / Miền nào ? [Chưa yêu cầu]

# 8. Viết chương trình “thư ký” soạn thảo văn bản theo lời đọc của người dùng:   
# dạng tương tự học sinh viết chính tả.

# 9. Viết lại chương trình: nhận lệnh và xuất ra mã lệnh một đoạn chương trình mẫu

# bằng các ngôn ngữ lập trình khác nhau (cho người dùng chọn); VD: C#.NET, Python, Java, ..

# 10. Viết lại chương trình: mô phỏng google dịch

# YÊU CẦU TỐI THIỂU: THỰC HIỆN BÀI MẪU = CÁ NHÂN HÓA THÔNG TIN

# tương tự = My Talking Tom: vd nói vi = nói lại vi

# Làm thử "CHUYỂN NGỮ" cho bài My Talking Tom: vd nói vi = nói lại en

# TÙY KHẢ NĂNG = SV LÀM TIẾP CÁC Y/C TRÊN

### Full code bài làm

# B1: NẠP THƯ VIỆN

import speech\_recognition as sr

from gtts import gTTS

import playsound

[1] Nghe tiếng Việt => Text

#PHẦN 1: IN RA TEXT THEO LỜI NÓI = Nghe tiếng Việt => Text

# B2: CHỌN PHƯƠNG ÁN NHẬP ÂM THANH TỪ MICROPHONE

r = sr.Recognizer()

# B3: XỬ LÝ NHẬN DIỆN MIC

with sr.Microphone() as Source:

#hiệu chỉnh mic để chuẩn bị nói

print("1. Chọn thoi gian cho là 1 s")

print("2. Chọn thoi gian cho là 2 s")

print("Quý vị chọn 1 hay 2")

ptml = int(input("Quý vị chọn 1 hay 2: "))

print("Hieu chinh nhieu trươc khi noi!")

r.adjust\_for\_ambient\_noise(Source, duration=ptml)

print("Hieu chinh nhieu trươc khi noi!")

r.adjust\_for\_ambient\_noise(Source, duration=1)

#nhận lời nói của người dùng từ MIc mặc định lưu dữ liệu âm thanh vào audio\_data

print("Nói tiếng Việt đi, sau 5s sẽ in ra văn bản!")

audio\_data\_ptml = r.record(Source, duration = 5)

#In ra văn bản text

print("KẾT QUẢ NHẬN DIỆN ..................")

#chuyển lời nói thành văn bản

try:

G213\_text = r.recognize\_google(audio\_data\_ptml,language="vi")

except:

G213\_text = "Quý vị nói gì nghe không rõ...!"

#in kết quả ra

print("Quý vị đã nói là : ",format(text))

[2] Text => đọc tiếng Việt

#PHẦN 2: XUẤT RA LỜI NÓI THEO VĂN BẢN ĐÃ NHẬP = Trả lời bằng tiếng Việt : Text => Nói tiếng Việt

# ĐỌC CHỮ VIỆT => NÓI RA TIẾNG VIỆT

def G213NTD(t):

ptm = gTTS(text=t, lang = 'vi')

soundG213 = '06VXT3.mp3'

vx.save(soundG213 )

playsound.playsound(soundG213 )

PTML(text)

#PTML("số thứ tự 18, họ tên: Phan Thị Mỹ Linh là sinh viên siêu cấp đỉnh cao của TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH")

### Các bài tương tự

Cá nhân hóa thông tin trong bài làm

**Bài 1:** Viết chương trình “thư ký” soạn thảo văn bản theo lời đọc của người dùng: dạng tương tự học sinh viết chính tả.

**Bài 2:** Viết lại chương trình mẫu với nhiều dạng khác:

+ Cho phép chọn Ngôn ngữ các quốc gia khác nhau: thay vì chỉ tiếng Việt, ví dụ: tiếng Ả rập, tiếng Anh, Tiếng Hàn, tiếng Hoa, tiếng Nhật.*[mượn translate.google.com : input speech]*

+ Tiếng Việt: chọn giọng nam or nữ || giọng miền Nam or Miền Bắc or…  
 *[nạp thư viện khác của anh/chị Việt Nam: khác thư viện gtts]*

**Bài 3:** Viết lại chương trình: nhận lệnh và xuất ra mã lệnh một đoạn chương trình mẫu bằng các ngôn ngữ lập trình khác nhau (cho người dùng chọn); VD: C#.NET, Python, Java, ..

**Bài 4:** Viết lại chương trình: mô phỏng google dịch

## LẬP TRÌNH GIAO DIỆN ĐỒ HỌA (GUI)

# Giới thiệu chủ đề

+ GUI : Graphical User Interface = giao diện đồ họa dạng WinForm

+ Thư viện chính: Tkinter

[https://docs.python.org/3/library/tkinter.html#module-tkinter](https://docs.python.org/3/library/tkinter.html" \l "module-tkinter)

+ Thư viện khác:

* Kivy: Kivy là một framework mã nguồn mở bằng Python được dùng để phát triển ứng dụng đa nền tảng (cross-platform), đặc biệt là trên thiết bị di động như Android và iOS. Nó hỗ trợ tạo ra giao diện người dùng (GUI) với các thành phần đồ họa được tối ưu cho cảm ứng.
* Kivy cung cấp một số công cụ quan trọng như:
* Widget: Thành phần cơ bản của giao diện người dùng, bao gồm Button, Label, TextInput, Slider,... Mọi thứ bạn hiển thị trong ứng dụng Kivy đều là các widget.
* Kv Language: Một ngôn ngữ khai báo tùy chọn cho phép bạn định nghĩa giao diện người dùng dễ dàng hơn so với viết bằng mã Python.
* Canvas: Kivy sử dụng một hệ thống canvas để vẽ giao diện đồ họa. Bạn có thể thêm các thành phần đồ họa như hình chữ nhật, hình tròn hoặc hình ảnh vào canvas của mỗi widget.
* Layouts: Các lớp bố trí giúp sắp xếp widget trên màn hình như BoxLayout, GridLayout, FloatLayout,...
* Event Dispatcher: Đây là công cụ quản lý các sự kiện trong ứng dụng, ví dụ như sự kiện nhấn nút, thay đổi giá trị,...
* Gesture Support: Kivy hỗ trợ các cử chỉ (gesture) như vuốt, kéo thả, chạm đa điểm (multi-touch), phù hợp cho ứng dụng cảm ứng.
* Ưu điểm
* Đa nền tảng: Bạn có thể phát triển một ứng dụng và chạy trên các nền tảng khác nhau như Android, iOS, Windows, Linux và macOS mà không cần thay đổi mã nguồn.
* Cảm ứng mạnh mẽ: Kivy được thiết kế đặc biệt cho các ứng dụng cảm ứng với sự hỗ trợ tốt cho cảm ứng đa điểm.
* Mã nguồn mở: Là mã nguồn mở, nên bạn có thể sử dụng và đóng góp cho cộng đồng miễn phí.
* Tích hợp tốt với Python: Kivy hỗ trợ tận dụng các thư viện Python khác, rất phù hợp cho các nhà phát triển đã quen thuộc với Python.
* Tùy biến cao: Với canvas và Kv language, bạn có thể tạo ra các giao diện tùy biến mà không bị ràng buộc bởi các widget có sẵn.
* Nhược điểm
* Hiệu suất thấp hơn so với native apps: Dù chạy trên nhiều nền tảng, Kivy không đạt được hiệu suất tốt như ứng dụng viết bằng ngôn ngữ native (ví dụ: Swift cho iOS hay Java cho Android).
* Thiếu các thành phần giao diện nâng cao: So với các framework khác như Flutter, Kivy thiếu các widget giao diện phức tạp, bạn cần tự phát triển hoặc tích hợp thêm.
* Hỗ trợ cộng đồng hạn chế: So với một số framework khác, tài liệu và sự hỗ trợ từ cộng đồng Kivy có thể ít hơn.
* Ứng dụng:
* Ứng dụng di động: Kivy rất phù hợp để phát triển ứng dụng di động, đặc biệt là các ứng dụng cảm ứng với giao diện động.
* Game đơn giản: Bạn có thể tạo các game nhỏ với đồ họa tương đối.
* Ứng dụng cảm ứng cho các thiết bị đặc thù: Kivy có thể được dùng để phát triển ứng dụng cho các thiết bị cảm ứng như kiosk, bảng điều khiển,...

Một số ứng dụng tiêu biểu của Kivy có thể kể đến là hệ thống điều khiển thông qua màn hình cảm ứng, các ứng dụng giáo dục, hay các ứng dụng phục vụ cho các sự kiện (event apps).

Ví dụ:

from kivy.app import App

from kivy.uix.gridlayout import GridLayout

from kivy.uix.button import Button

from kivy.uix.textinput import TextInput

from kivy.uix.boxlayout import BoxLayout

# Lớp CalculatorGrid kế thừa từ BoxLayout, là bố cục chính của máy tính

class G213CalculatorGrid(BoxLayout):

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

        # Gọi phương thức khởi tạo của lớp cha (BoxLayout)

        super(G213CalculatorGrid, self).\_\_init\_\_(\*\*kwargs)

        # Thiết lập hướng bố cục là 'vertical' (theo chiều dọc)

        self.orientation = 'vertical'

        # Tạo màn hình hiển thị kết quả, chỉ cho phép đọc, không cho phép người dùng nhập trực tiếp

        self.display = TextInput(font\_size=48, readonly=True, halign="right", multiline=False, size\_hint\_y=0.2)  # Chỉnh chiều cao nhỏ lại

        # Thêm màn hình hiển thị vào bố cục chính

        self.add\_widget(self.display)

        # Tạo một lưới (GridLayout) với 4 cột để chứa các nút của máy tính

        grid\_layout = GridLayout(cols=4)

        # Danh sách các nút số và phép toán cần thiết cho máy tính

        buttons19 = [

            '7', '8', '9', '/',  # Hàng thứ nhất với các nút số và nút chia

            '4', '5', '6', '\*',  # Hàng thứ hai với các nút số và nút nhân

            '1', '2', '3', '-',  # Hàng thứ ba với các nút số và nút trừ

            'C', '0', '=', '+'   # Hàng thứ tư với nút xóa, số 0, dấu bằng, và cộng

        ]

        # Vòng lặp qua danh sách các nút và thêm chúng vào bố cục lưới

        for button in buttons19:

            # Nếu là nút '=' thì thiết lập màu nền là vàng

            if button == '=':

                grid\_layout.add\_widget(Button(text=button, on\_press=self.on\_button\_press, background\_color=(1, 1, 0, 1)))

            else:

                # Các nút khác được thêm với hành động khi nhấn là gọi hàm `on\_button\_press`

                grid\_layout.add\_widget(Button(text=button, on\_press=self.on\_button\_press))

        # Thêm bố cục lưới chứa các nút vào bố cục chính

        self.add\_widget(grid\_layout)

    # Hàm xử lý sự kiện khi người dùng nhấn nút

    def on\_button\_press(self, instance):

        # Lấy giá trị hiện tại trên màn hình

        current = self.display.text

        # Lấy nội dung của nút vừa được nhấn

        button\_text = instance.text

        # Nếu nhấn nút '=', thực hiện tính toán biểu thức

        if button\_text == '=':

            try:

                # Dùng hàm eval() để tính toán biểu thức trên màn hình

                self.display.text = str(eval(current))

            except Exception:

                # Nếu có lỗi, hiển thị "Error" trên màn hình

                self.display.text = "Error"

        # Nếu nhấn nút 'C', xóa toàn bộ nội dung trên màn hình

        elif button\_text == 'C':

            self.display.text = ""

        # Các nút khác, thêm nội dung của nút vào màn hình

        else:

            self.display.text = current + button\_text

# Lớp chính của ứng dụng, kế thừa từ App

class CalculatorG213(App):

    # Phương thức build để khởi tạo giao diện

    def build(self):

        # Trả về bố cục của máy tính (CalculatorGrid)

        return G213CalculatorGrid()

# Kiểm tra xem có phải chương trình chính đang chạy hay không

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

    # Chạy ứng dụng

    CalculatorG213().run()

* Chức năng:

Ứng dụng này là một máy tính đơn giản với các chức năng chính sau:

* Nhập và hiển thị: Hiển thị các con số và phép toán khi nhấn các nút.
* Tính toán: Thực hiện các phép tính số học cơ bản như cộng, trừ, nhân, chia khi nhấn dấu =.
* Xóa: Xóa toàn bộ nội dung màn hình khi nhấn nút C.
* Xử lý lỗi: Hiển thị "Error" nếu biểu thức không hợp lệ.

Ứng dụng cung cấp các chức năng máy tính cơ bản qua giao diện trực quan.

* PyQT: là một bộ công cụ phát triển ứng dụng GUI (giao diện người dùng đồ họa) đa nền tảng dựa trên bộ thư viện Qt, một framework mạnh mẽ cho việc phát triển ứng dụng trên nhiều nền tảng khác nhau như Windows, macOS, và Linux. PyQt cung cấp các ràng buộc Python cho Qt, cho phép bạn sử dụng Python để phát triển các ứng dụng giao diện người dùng phức tạp.
* Các công cụ quan trọng trong PyQt:
* Qt Designer: Một công cụ kéo-thả để thiết kế giao diện người dùng. Bạn có thể tạo ra các giao diện phức tạp mà không cần viết mã trực tiếp, sau đó xuất thành tệp .ui để sử dụng trong PyQt.
* Widgets: PyQt cung cấp rất nhiều widget giao diện người dùng, từ các thành phần đơn giản như Button, Label, đến các thành phần phức tạp như TreeView, TableView, và các widget tùy chỉnh khác.
* Signal and Slot: Cơ chế Signal and Slot trong PyQt cho phép quản lý sự kiện một cách dễ dàng và linh hoạt. Signal là sự kiện xảy ra khi có một hành động nào đó (như nhấn nút), còn Slot là hàm phản hồi được kích hoạt khi Signal phát ra.
* Layouts: PyQt hỗ trợ nhiều lớp bố trí để sắp xếp các widget như QHBoxLayout, QVBoxLayout, QGridLayout,...
* QtCore và QtGui Modules: Cung cấp các thành phần quan trọng như thao tác với dữ liệu, thời gian, tệp tin, các thành phần đồ họa, và quản lý các thao tác nền tảng của hệ thống.
* Internationalization (I18n): PyQt hỗ trợ quốc tế hóa, giúp bạn dễ dàng phát triển ứng dụng hỗ trợ đa ngôn ngữ thông qua Qt Linguist.
* QML: Một ngôn ngữ giao diện khai báo tương tự như Kv trong Kivy, giúp tạo ra giao diện động và mượt mà hơn.
* Ưu điểm:
* Đa nền tảng: Phát triển một lần và chạy trên nhiều hệ điều hành khác nhau mà không cần phải chỉnh sửa nhiều.
* Thư viện giao diện người dùng phong phú: PyQt cung cấp bộ widget rất phong phú và linh hoạt, giúp phát triển các ứng dụng GUI chuyên nghiệp và phức tạp.
* Hỗ trợ tốt cho MVC/MVVM: PyQt hỗ trợ các mô hình kiến trúc như MVC (Model-View-Controller) và MVVM (Model-View-ViewModel), giúp quản lý ứng dụng dễ dàng và có cấu trúc rõ ràng.
* Qt Designer: Công cụ thiết kế giao diện trực quan, giúp tiết kiệm thời gian phát triển và dễ dàng tùy chỉnh giao diện người dùng.
* Hiệu suất cao: PyQt dựa trên Qt, một framework native với hiệu suất rất tốt và tối ưu cho các ứng dụng phức tạp.
* Hỗ trợ đồ họa 2D và 3D: PyQt có khả năng xử lý đồ họa 2D và 3D mạnh mẽ thông qua các module như QtGui và QtOpenGL.
* Nhược điểm:
* Giấy phép thương mại: PyQt có hai loại giấy phép: GPL (mã nguồn mở) và giấy phép thương mại. Nếu bạn muốn phát triển các ứng dụng đóng nguồn mà không muốn chia sẻ mã nguồn, bạn phải mua giấy phép thương mại.
* Độ phức tạp cao: Vì PyQt cung cấp nhiều tính năng phức tạp và mạnh mẽ, nên việc học và sử dụng PyQt có thể mất nhiều thời gian hơn so với các framework khác, đặc biệt đối với người mới.
* Tài liệu rải rác: Mặc dù có nhiều tài liệu về Qt, nhưng tài liệu chính thức và các tài nguyên dành cho PyQt (Python bindings) không phong phú bằng các thư viện khác như Kivy hay Tkinter.
* Đòi hỏi kiến thức về Qt: Để tận dụng hết tiềm năng của PyQt, bạn cần hiểu rõ về framework Qt, điều này có thể gây khó khăn cho người chỉ quen với Python.
* Ứng dụng:
* Phát triển ứng dụng desktop: PyQt là lựa chọn phổ biến cho các ứng dụng desktop phức tạp như IDE, phần mềm quản lý dữ liệu, công cụ chỉnh sửa ảnh, phần mềm kỹ thuật.
* Ứng dụng thương mại: Do hiệu suất cao và hỗ trợ giao diện người dùng phong phú, PyQt thường được dùng trong các phần mềm doanh nghiệp và thương mại, như hệ thống quản lý thông tin, phần mềm tài chính, phần mềm xử lý văn bản,...
* Công cụ nội bộ: Nhiều tổ chức sử dụng PyQt để phát triển các công cụ nội bộ cho nhân viên, bao gồm các công cụ xử lý dữ liệu, báo cáo, và giao diện điều khiển cho các hệ thống lớn.

Với khả năng linh hoạt và hiệu suất cao, PyQt là một trong những framework mạnh mẽ cho các dự án phát triển ứng dụng giao diện người dùng chuyên nghiệp.

* + - Ví dụ:

from PyQt5 import QtCore, QtGui, QtWidgets

from PyQt5.QtGui import QImage, QPixmap

from matplotlib.backends.backend\_qt5agg import FigureCanvasQTAgg as FigureCanvas

from matplotlib.figure import Figure

import cv2

import sys

class PlotCanvas(FigureCanvas):

    def \_\_init\_\_(self, parent=None):

        fig = Figure(figsize=(6, 4), dpi=100)

        self.axes = fig.add\_subplot(111)

        super(PlotCanvas, self).\_\_init\_\_(fig)

        self.setParent(parent)

    def plot(self, data\_on, data\_off, labels):

        """Vẽ biểu đồ cột đôi"""

        self.axes.clear()

        bar\_width = 0.35

        index = range(len(labels))

        # Vẽ các cột

        self.axes.bar(index, data\_on, bar\_width, label="Bật", color='b')

        self.axes.bar([i + bar\_width for i in index], data\_off, bar\_width, label="Tắt", color='r')

        # Gắn nhãn và tiêu đề

        self.axes.set\_xlabel('Thiết bị')

        self.axes.set\_ylabel('Số lần')

        self.axes.set\_title('Số lần Bật và Tắt của từng thiết bị')

        self.axes.set\_xticks([i + bar\_width / 2 for i in index])

        self.axes.set\_xticklabels(labels)

        self.axes.legend()

        # Vẽ lại biểu đồ

        self.draw()

class MainWindow\_CuaLinh(object):

    def \_\_init\_\_(self, MainWindow):

        MainWindow.setObjectName("MainWindow")

        MainWindow.resize(1500, 700)

        MainWindow.setWindowTitle("tìm hiểu PyQt5")

        # Biến lưu số lần bật/tắt của từng thiết bị

        self.tv\_on\_count = 0

        self.tv\_off\_count = 0

        self.lamp\_on\_count = 0

        self.lamp\_off\_count = 0

        self.aircon\_on\_count = 0

        self.aircon\_off\_count = 0

        # Biến lưu trạng thái của TV

        self.tv\_is\_on = False  # TV mặc định đang tắt

        self.current\_tv\_channel = 1  # Mặc định kênh 1

        # Biến điều khiển đèn nhấp nháy

        self.is\_mode\_blink\_on = False

        # Tạo QTimer để điều khiển nhấp nháy đèn

        self.blink\_timer = QtCore.QTimer()

        self.blink\_timer.timeout.connect(self.toggle\_lamp)

        # Widget trung tâm

        self.central\_widget = QtWidgets.QWidget(MainWindow)

        MainWindow.setCentralWidget(self.central\_widget)

        # Bố cục chính (bố cục dọc với nhãn thông tin ở trên cùng)

        self.main\_layout = QtWidgets.QVBoxLayout(self.central\_widget)

        # Thêm nhãn thông tin cá nhân ở trên cùng

        self.info\_label = QtWidgets.QLabel("Báo cáo tìm hiểu thư viện GUI PyQt5 của SV: Nguyễn Thành Đạt, MSSV: 22110129, Lớp: PyPro2, HCMUTE", self.central\_widget)

        self.info\_label.setAlignment(QtCore.Qt.AlignCenter)

        self.info\_label.setStyleSheet("font-size: 16px; font-weight: bold;")

        # Thêm nhãn vào bố cục chính

        self.main\_layout.addWidget(self.info\_label)

        # Bố cục ngang cho 3 cột (thiết bị, nút chế độ, biểu đồ)

        self.hbox\_layout = QtWidgets.QHBoxLayout()

        # Bố cục bên trái cho các thiết bị (bố cục dọc)

        self.left\_layout = QtWidgets.QVBoxLayout()

        # Thêm các thiết bị với hình ảnh vào bố cục bên trái

        self.tv\_label = self.create\_appliance\_ui("TV", "object-images/TVoff.png", self.left\_layout)

        self.lamp\_label = self.create\_appliance\_ui("Đèn", "object-images/lamp-off.png", self.left\_layout)

        self.aircon\_label = self.create\_appliance\_ui("Điều hòa", "object-images/airconditioner-off.png", self.left\_layout)

        # Bố cục giữa cho các nút chế độ và textbox (bố cục dọc)

        self.middle\_layout = QtWidgets.QVBoxLayout()

        # Thêm nhãn và textbox vào cột 2

        self.user\_label = QtWidgets.QLabel("Nhập tên người dùng:", self.central\_widget)

        self.user\_textbox = QtWidgets.QLineEdit(self.central\_widget)

        # Thêm thanh trượt cho độ tuổi

        self.age\_label = QtWidgets.QLabel("Chọn độ tuổi:", self.central\_widget)

        self.age\_slider = QtWidgets.QSlider(QtCore.Qt.Horizontal, self.central\_widget)

        self.age\_slider.setMinimum(0)  # Đặt độ tuổi tối thiểu

        self.age\_slider.setMaximum(100)  # Đặt độ tuổi tối đa

        self.age\_slider.setValue(20)  # Đặt giá trị mặc định

        # Hiển thị giá trị độ tuổi đã chọn

        self.age\_value\_label = QtWidgets.QLabel(f"Độ tuổi hiện tại: {self.age\_slider.value()}", self.central\_widget)

        self.age\_slider.valueChanged.connect(self.update\_age\_value)

        # Thêm nhãn, textbox và thanh trượt vào bố cục giữa

        self.middle\_layout.addWidget(self.user\_label)

        self.middle\_layout.addWidget(self.user\_textbox)

        self.middle\_layout.addWidget(self.age\_label)

        self.middle\_layout.addWidget(self.age\_slider)

        self.middle\_layout.addWidget(self.age\_value\_label)

        # Thêm label hiển thị hình ảnh từ camera

        self.image\_label = QtWidgets.QLabel(self.central\_widget)

        self.middle\_layout.addWidget(self.image\_label)

        # Sử dụng OpenCV để mở camera

        self.cap = cv2.VideoCapture(0)

        # Sử dụng QTimer để cập nhật hình ảnh từ camera

        self.timer = QtCore.QTimer(self.central\_widget)

        self.timer.timeout.connect(self.update\_frame)

        self.timer.start(20)  # Cập nhật mỗi 20ms

        # Thêm các nút chế độ vào bố cục giữa

        self.add\_mode\_buttons(self.middle\_layout)

        # Bố cục bên phải cho biểu đồ (bố cục dọc)

        self.right\_layout = QtWidgets.QVBoxLayout()

        # Thêm canvas biểu đồ vào bố cục bên phải

        self.plot\_canvas = PlotCanvas(self.central\_widget)

        self.right\_layout.addWidget(self.plot\_canvas)

        # Thêm các bố cục trái, giữa và phải vào bố cục ngang

        self.hbox\_layout.addLayout(self.left\_layout)  # Thêm cột thiết bị

        self.hbox\_layout.addLayout(self.middle\_layout)  # Thêm cột nút chế độ

        self.hbox\_layout.addLayout(self.right\_layout)  # Thêm cột biểu đồ

        # Thêm bố cục ngang vào bố cục chính

        self.main\_layout.addLayout(self.hbox\_layout)

    def create\_appliance\_ui(self, name, image\_path, layout):

        """Tạo giao diện cho một thiết bị với hình ảnh và các nút Bật/Tắt"""

        # Nhãn thiết bị với hình ảnh

        appliance\_label = QtWidgets.QLabel(self.central\_widget)

        appliance\_label.setAlignment(QtCore.Qt.AlignCenter)

        appliance\_label.setPixmap(QtGui.QPixmap(image\_path).scaled(100, 100, QtCore.Qt.KeepAspectRatio))

        # Các nút Bật/Tắt và Chuyển Kênh cho TV

        on\_button = QtWidgets.QPushButton(f"Bật {name}", self.central\_widget)

        off\_button = QtWidgets.QPushButton(f"Tắt {name}", self.central\_widget)

        if name == "TV":

            # Nút chuyển kênh cho TV

            channel\_button = QtWidgets.QPushButton("Chuyển kênh", self.central\_widget)

            channel\_button.clicked.connect(self.change\_tv\_channel\_mode)

        if name == "Đèn":

            # Nút nhấp nháy cho đèn

            blink\_button = QtWidgets.QPushButton("Nhấp nháy Đèn", self.central\_widget)

            blink\_button.clicked.connect(self.blink\_lamp\_mode)

        # Kết nối nút với các phương thức tương ứng

        if name == "TV":

            on\_button.clicked.connect(self.tv\_on\_mode)

            off\_button.clicked.connect(self.tv\_off\_mode)

        elif name == "Đèn":

            on\_button.clicked.connect(self.lamp\_on\_mode)

            off\_button.clicked.connect(self.lamp\_off\_mode)

        elif name == "Điều hòa":

            on\_button.clicked.connect(self.aircon\_on\_mode)

            off\_button.clicked.connect(self.aircon\_off\_mode)

        # Bố cục ngang cho các nút

        button\_layout = QtWidgets.QHBoxLayout()

        button\_layout.addWidget(on\_button)

        if name == "TV":

            button\_layout.addWidget(channel\_button)  # Thêm nút chuyển kênh giữa nút bật và tắt

        if name == "Đèn":

            button\_layout.addWidget(blink\_button)  # Thêm nút nhấp nháy cho đèn

        button\_layout.addWidget(off\_button)

        # Bố cục dọc cho thiết bị (hình ảnh + nút)

        appliance\_layout = QtWidgets.QVBoxLayout()

        appliance\_layout.addWidget(appliance\_label)

        appliance\_layout.addLayout(button\_layout)

        # Thêm bố cục thiết bị vào bố cục được truyền vào (bố cục bên trái)

        layout.addLayout(appliance\_layout)

        return appliance\_label

    def add\_mode\_buttons(self, layout):

        """Thêm các nút Chế độ bật tất cả, Tắt tất cả, Chế độ 3 vào bố cục bên phải"""

        check\_user\_age\_button = QtWidgets.QPushButton("Nhấn vào tôi trước khi nhấn nút khác", self.central\_widget)

        turn\_on\_all\_button = QtWidgets.QPushButton("Bật tất cả thiết bị", self.central\_widget)

        turn\_off\_all\_button = QtWidgets.QPushButton("Tắt tất cả thiết bị", self.central\_widget)

        # Kết nối sự kiện bật/tắt tất cả với nút

        turn\_on\_all\_button.clicked.connect(self.turn\_on\_all\_mode)

        turn\_off\_all\_button.clicked.connect(self.turn\_off\_all\_mode)

        # Kết nối chế độ 3 với phương thức kiểm tra độ tuổi

        check\_user\_age\_button.clicked.connect(self.check\_age\_mode)

        # Thêm các nút vào bố cục giữa (xếp chồng theo chiều dọc)

        layout.addWidget(check\_user\_age\_button)

        layout.addWidget(turn\_on\_all\_button)

        layout.addWidget(turn\_off\_all\_button)

        # Thêm stretch để đẩy các nút lên trên cùng

        layout.addStretch()

    def update\_age\_value(self):

        """Cập nhật nhãn hiển thị giá trị độ tuổi khi kéo thanh trượt"""

        self.age\_value\_label.setText(f"Quý vị hiện đang ở độ tuổi: {self.age\_slider.value()}")

    def check\_age\_mode(self):

        """Chế độ kiểm tra độ tuổi, hiển thị thông báo"""

        name = self.user\_textbox.text()

        age = self.age\_slider.value()  # Lấy giá trị độ tuổi từ thanh trượt

        msg = QtWidgets.QMessageBox()

        if age < 10:

            msg.setText(f"Xin chào bé {name}!!! Bạn đang ở độ tuổi {age}. Bạn quá nhỏ để nghịch đồ điện tử. Hãy tắt ứng dụng")

        elif age > 10 and age < 16:

            msg.setText(f"Xin chào em {name}!!! Bạn đang ở độ tuổi {age}. Bạn hãy cẩn thận khi sử dụng các thiết bị điện.")

        else:

            msg.setText(f"Xin chào {name}!!! Bạn đang ở độ tuổi {age}. Let's enjoy my app!")

        msg.setWindowTitle("Cảnh báo")

        msg.exec\_()

    # Phương thức bật/tắt TV

    def tv\_on\_mode(self):

        """Bật TV và hiển thị kênh hiện tại"""

        self.tv\_on\_count += 1  # Cập nhật số lần bật

        self.tv\_is\_on = True

        self.update\_tv\_display()

        self.update\_plot()

    def tv\_off\_mode(self):

        """Tắt TV"""

        self.tv\_off\_count += 1  # Cập nhật số lần tắt

        self.tv\_is\_on = False

        self.tv\_label.setPixmap(QtGui.QPixmap("object-images/TVoff.png").scaled(100, 100, QtCore.Qt.KeepAspectRatio))

        self.update\_plot()

    def update\_tv\_display(self):

        """Cập nhật hình ảnh hiển thị của TV dựa trên kênh hiện tại và trạng thái bật/tắt"""

        if self.tv\_is\_on:

            if self.current\_tv\_channel == 1:

                self.tv\_label.setPixmap(QtGui.QPixmap("object-images/tv-thiennioi.png").scaled(100, 100, QtCore.Qt.KeepAspectRatio))

            elif self.current\_tv\_channel == 2:

                self.tv\_label.setPixmap(QtGui.QPixmap("object-images/tv-jack-n-oggy.png").scaled(100, 100, QtCore.Qt.KeepAspectRatio))

    # Phương thức chuyển kênh TV

    def change\_tv\_channel\_mode(self):

        """Chuyển kênh TV (chỉ khi TV đang bật)"""

        if self.tv\_is\_on:

            # Chuyển kênh giữa 1 và 2

            if self.current\_tv\_channel == 1:

                self.current\_tv\_channel = 2

            else:

                self.current\_tv\_channel = 1

            self.update\_tv\_display()

    # Phương thức bật/tắt Đèn

    def lamp\_on\_mode(self):

        """Cập nhật hình ảnh Đèn khi bật"""

        self.lamp\_on\_count += 1  # Cập nhật số lần bật

        self.lamp\_label.setPixmap(QtGui.QPixmap("object-images/lamp-on.png").scaled(100, 100, QtCore.Qt.KeepAspectRatio))

        if self.is\_mode\_blink\_on:

            self.blink\_timer.stop()

        self.update\_plot()

    def lamp\_off\_mode(self):

        """Cập nhật hình ảnh Đèn khi tắt"""

        self.lamp\_off\_count += 1  # Cập nhật số lần tắt

        self.lamp\_label.setPixmap(QtGui.QPixmap("object-images/lamp-off.png").scaled(100, 100, QtCore.Qt.KeepAspectRatio))

        self.update\_plot()

    # Phương thức nhấp nháy đèn

    def blink\_lamp\_mode(self):

        """Bật chế độ nhấp nháy đèn"""

        if not self.blink\_timer.isActive():

            self.blink\_timer.start(500)  # Cài đặt timer 3 giây để nhấp nháy

        else:

            # Dừng chế độ nhấp nháy

            self.blink\_timer.stop()

    def toggle\_lamp(self):

        if self.is\_mode\_blink\_on:

            self.lamp\_off\_mode()

        else:

            self.lamp\_on\_mode()

        self.is\_mode\_blink\_on = not self.is\_mode\_blink\_on

    # Phương thức bật/tắt Điều hòa

    def aircon\_on\_mode(self):

        """Cập nhật hình ảnh Điều hòa khi bật"""

        self.aircon\_on\_count += 1  # Cập nhật số lần bật

        self.aircon\_label.setPixmap(QtGui.QPixmap("object-images/airconditioner-on.png").scaled(100, 100, QtCore.Qt.KeepAspectRatio))

        self.update\_plot()

    def aircon\_off\_mode(self):

        """Cập nhật hình ảnh Điều hòa khi tắt"""

        self.aircon\_off\_count += 1  # Cập nhật số lần tắt

        self.aircon\_label.setPixmap(QtGui.QPixmap("object-images/airconditioner-off.png").scaled(100, 100, QtCore.Qt.KeepAspectRatio))

        self.update\_plot()

    # Phương thức cập nhật biểu đồ

    def update\_plot(self):

        """Cập nhật biểu đồ cột với số lần bật/tắt của các thiết bị"""

        data\_on = [self.tv\_on\_count, self.lamp\_on\_count, self.aircon\_on\_count]

        data\_off = [self.tv\_off\_count, self.lamp\_off\_count, self.aircon\_off\_count]

        labels = ['TV', 'Đèn', 'Điều hòa']

        self.plot\_canvas.plot(data\_on, data\_off, labels)

    # Phương thức bật/tắt tất cả các thiết bị

    def turn\_on\_all\_mode(self):

        self.lamp\_on\_mode()

        self.tv\_on\_mode()

        self.aircon\_on\_mode()

    def turn\_off\_all\_mode(self):

        self.lamp\_off\_mode()

        self.tv\_off\_mode()

        self.aircon\_off\_mode()

    def update\_frame(self):

        ret, frame = self.cap.read()

        if ret:

            # Chuyển đổi từ định dạng BGR của OpenCV sang RGB để hiển thị trong PyQt

            frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

            height, width, channel = frame.shape

            step = channel \* width

            q\_img = QImage(frame.data, width, height, step, QImage.Format\_RGB888)

            self.image\_label.setPixmap(QPixmap.fromImage(q\_img))

    def closeEvent(self, event):

        # Khi đóng ứng dụng, dừng camera

        self.cap.release()

# Vòng lặp chính

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

    app = QtWidgets.QApplication(sys.argv)

    MainWindow = QtWidgets.QMainWindow()

    ui = MainWindow\_G213(MainWindow)

    MainWindow.show()

    sys.exit(app.exec\_())

* Các hình ảnh được import từ resources\_rc (pyrcc5 resourcses.qrc -o resources\_rc.py)

A screenshot of a computer

Description automatically generated

* Các chức năng của chương trình:

Điều khiển TV:

* Bật/Tắt TV.
* Chuyển kênh TV giữa hai kênh khác nhau.
* Theo dõi số lần bật/tắt TV và cập nhật hình ảnh của TV theo trạng thái.

Điều khiển Đèn:

* Bật/Tắt Đèn.
* Chế độ "Nhấp nháy" cho đèn với chức năng bật/tắt tự động mỗi 500ms.
* Theo dõi số lần bật/tắt và cập nhật hình ảnh của đèn.
* Điều khiển Điều hòa:

Bật/Tắt Điều hòa.

* Theo dõi số lần bật/tắt và cập nhật hình ảnh của điều hòa.

Chế độ Bật/Tắt Tất cả:

* Nút bật tất cả các thiết bị (TV, Đèn, Điều hòa).
* Nút tắt tất cả các thiết bị.

Chế độ Kiểm tra tuổi người dùng:

* Người dùng nhập tên và chọn độ tuổi bằng thanh trượt.
* Hiển thị thông báo phù hợp với từng nhóm tuổi (dưới 10, 10-16, trên 16).

Biểu đồ cột theo dõi số lần bật/tắt thiết bị:

* Hiển thị số lần bật và tắt của các thiết bị TV, Đèn, và Điều hòa dưới dạng biểu đồ cột.

Khung hình hiển thị hình ảnh lấy trực tiếp từ camera máy tính.

* PyGUI: là một thư viện Python dành cho phát triển giao diện người dùng (GUI) đa nền tảng. Thư viện này được thiết kế để tận dụng các đặc điểm độc đáo của ngôn ngữ Python và hoạt động mượt mà với các loại dữ liệu của Python

PyGUI thường được sử dụng để phát triển các ứng dụng như:

* Ứng dụng đa nền tảng: Phát triển một lần và chạy trên nhiều hệ điều hành khác nhau.
* Ứng dụng khoa học và kỹ thuật: Phát triển các ứng dụng yêu cầu giao diện trực quan và tương tác nhanh chóng.
* Ứng dụng trò chơi: Xây dựng giao diện người dùng cho các trò chơi đa nền tảng.

Ưu điểm:

* Đa nền tảng: PyGUI hỗ trợ phát triển một lần và chạy trên nhiều nền tảng khác nhau.
* Tích hợp Python: Tận dụng đặc điểm độc đáo của Python và hoạt động mượt mà với các loại dữ liệu của Python1
* Dễ sử dụng: Thư viện được thiết kế để dễ dàng học và sử dụng.

Nhược điểm:

* Không được phát triển tích cực: Thư viện này không được phát triển tích cực như một số thư viện GUI khác, có thể thiếu các tính năng và hỗ trợ.
* PySide: là một thư viện Python cho phép lập trình viên sử dụng thư viện Qt để tạo các ứng dụng giao diện người dùng (GUI) đa nền tảng. PySide là một bộ bọc (wrapper) cho thư viện Qt, cho phép bạn sử dụng các API của Qt trong các ứng dụng Python.

PySide được sử dụng rộng rãi trong việc phát triển các ứng dụng như:

* Ứng dụng máy tính: Tạo ra các phần mềm với giao diện người dùng thân thiện và đẹp mắt.
* Ứng dụng di động: Phát triển ứng dụng chạy trên nhiều nền tảng di động.
* Ứng dụng doanh nghiệp: Xây dựng các phần mềm quản lý, phân tích dữ liệu với giao diện người dùng trực quan.
  + - Ví dụ:

import sys

from PySide6.QtWidgets import QApplication, QMainWindow, QVBoxLayout, QWidget, QPushButton, QLineEdit, QLabel

from PySide6.QtCore import QTimer

from matplotlib.backends.backend\_qt5agg import FigureCanvasQTAgg as FigureCanvas

from matplotlib import pyplot as plt

# Hàm hoán đổi giá trị của hai phần tử trong danh sách A

def swap\_G213(A, i, j):

    A[i], A[j] = A[j], A[i]

# Thuật toán sắp xếp bong bóng (Bubble Sort)

def bubblesort\_G213(A):

    swapped = True

    for i in range(len(A) - 1):

        if not swapped:

            return  # Nếu không có hoán đổi thì kết thúc sắp xếp

        swapped = False

        for j in range(len(A) - 1 - i):

            if A[j] > A[j + 1]:

                swap\_G213(A, j, j + 1)  # Hoán đổi nếu phần tử bên trái lớn hơn bên phải

                swapped = True

            yield A  # Trả về trạng thái hiện tại của danh sách

# Lớp giao diện đồ họa hiển thị trực quan quá trình sắp xếp

class BubbleSortVisualizer\_G213(QMainWindow):

    def \_\_init\_\_(self):

        super().\_\_init\_\_()

        self.setWindowTitle("Sắp xếp nổi bọt cùng tìm hiểu PySide6 cùng em Linh lớp PyPro1")

        # Tạo widget trung tâm và thiết lập bố cục

        self.central\_widget = QWidget()

        self.setCentralWidget(self.central\_widget)

        self.layout = QVBoxLayout(self.central\_widget)

        # Tạo ô nhập liệu cho người dùng nhập các số

        self.label = QLabel("Nhập các số, cách nhau bằng dấu phẩy:")

        self.layout.addWidget(self.label)

        self.input\_field = QLineEdit()

        self.layout.addWidget(self.input\_field)

        # Nút để bắt đầu quá trình sắp xếp

        self.sort\_button = QPushButton("Bắt đầu sắp xếp")

        self.layout.addWidget(self.sort\_button)

        self.sort\_button.clicked.connect(self.start\_sorting)  # Khi nhấn nút, gọi hàm start\_sorting

        # Tạo khung và canvas để vẽ biểu đồ

        self.fig, self.ax = plt.subplots()

        self.canvas = FigureCanvas(self.fig)

        self.layout.addWidget(self.canvas)

        # Khởi tạo các biến

        self.A = []  # Danh sách chứa các số sẽ sắp xếp

        self.generator = None  # Bộ tạo cho thuật toán sắp xếp

        self.timer = QTimer()  # Bộ đếm giờ để cập nhật biểu đồ

        self.timer.timeout.connect(self.update\_plot)  # Cập nhật biểu đồ sau mỗi lần đếm giờ

        self.annotations = []  # Danh sách chứa chú thích (giá trị bên trong cột)

    def start\_sorting(self):

        # Lấy chuỗi nhập liệu từ người dùng và chuyển thành danh sách số nguyên

        input\_text = self.input\_field.text()

        try:

            self.A = [int(x) for x in input\_text.split(",")]

        except ValueError:

            self.label.setText("Lỗi nhập liệu. Vui lòng nhập các số, cách nhau bằng dấu phẩy.")

            return

        # Kiểm tra xem danh sách có ít nhất 2 số

        if len(self.A) < 2:

            self.label.setText("Vui lòng nhập ít nhất hai số.")

            return

        # Thiết lập lại biểu đồ với dữ liệu mới

        self.ax.clear()

        self.ax.set\_title("Sắp xếp nổi bọt O(n\N{SUPERSCRIPT TWO})")

        self.bar\_sub = self.ax.bar(range(len(self.A)), self.A, align="edge")  # Vẽ các cột

        self.ax.set\_xlim(0, len(self.A))

        self.text = self.ax.text(0.02, 0.95, "", transform=self.ax.transAxes)  # Chú thích số lượng thao tác

        self.iteration = [0]

        # Xóa các chú thích cũ nếu có

        for annotation in self.annotations:

            annotation.remove()

        self.annotations.clear()

        # Thêm chú thích (giá trị của mỗi cột) vào giữa các cột

        for rect in self.bar\_sub:

            annotation = self.ax.text(

                rect.get\_x() + rect.get\_width() / 2, rect.get\_height() / 2,

                f'{int(rect.get\_height())}', ha='center', va='center', color='white', fontsize=10

            )

            self.annotations.append(annotation)

        # Tạo bộ tạo cho thuật toán sắp xếp bong bóng

        self.generator = bubblesort\_cungLinh(self.A)

        # Bắt đầu bộ đếm giờ để cập nhật biểu đồ

        self.timer.start(50)  # Cập nhật mỗi 50ms

    def update\_plot(self):

        try:

            # Lấy trạng thái tiếp theo của danh sách từ bộ tạo

            A = next(self.generator)

            # Cập nhật chiều cao các cột và vị trí của chú thích bên trong mỗi cột

            for rect, val, annotation in zip(self.bar\_sub, A, self.annotations):

                rect.set\_height(val)

                annotation.set\_position((rect.get\_x() + rect.get\_width() / 2, rect.get\_height() / 2))

                annotation.set\_text(f'{int(val)}')

            # Tăng số lượng thao tác và cập nhật chú thích

            self.iteration[0] += 1

            self.text.set\_text(f"Số thao tác: {self.iteration[0]}")

            # Vẽ lại canvas để hiển thị thay đổi

            self.canvas.draw()

        except StopIteration:

            # Dừng bộ đếm giờ khi hoàn thành sắp xếp

            self.timer.stop()

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

    # Khởi tạo ứng dụng và hiển thị cửa sổ

    app = QApplication(sys.argv)

    window = BubbleSortVisualizer\_G213()

    window.show()

    sys.exit(app.exec())

* Chức năng:
* Nhập dữ liệu: Người dùng nhập một dãy số cách nhau bằng dấu phẩy thông qua ô nhập liệu.
* Hiển thị biểu đồ: Ứng dụng sẽ vẽ biểu đồ dạng cột biểu diễn các số đã nhập.
* Chạy thuật toán sắp xếp nổi bọt: Khi nhấn nút "Bắt đầu sắp xếp", ứng dụng sẽ khởi chạy thuật toán sắp xếp nổi bọt. Thuật toán này sẽ thực hiện việc hoán đổi vị trí các phần tử nếu cần, dần dần sắp xếp chúng theo thứ tự tăng dần.
* Cập nhật biểu đồ theo thời gian thực: Quá trình hoán đổi trong thuật toán sẽ được hiển thị bằng cách cập nhật chiều cao các cột trên biểu đồ và chú thích số lượng thao tác được thực hiện.
* Kết thúc sắp xếp: Khi quá trình sắp xếp hoàn tất, bộ đếm giờ sẽ dừng, và biểu đồ hiển thị danh sách số đã được sắp xếp.
* PySimpleGUI: là một thư viện Python giúp tạo giao diện người dùng đơn giản và nhanh chóng, phù hợp cho cả người mới bắt đầu. Nó cung cấp cú pháp dễ hiểu và trực quan, dựa trên các thư viện GUI phổ biến như Tkinter, PyQt, WxPython, và Remi, giúp rút ngắn thời gian phát triển giao diện. Với PySimpleGUI, bạn có thể nhanh chóng xây dựng các ứng dụng đa nền tảng mà không cần học quá nhiều về các thư viện GUI phức tạp khác.
  + - Ví dụ:
* Libavg: là một thư viện mã nguồn mở được thiết kế để phát triển giao diện người dùng (UI) tương tác và trực quan trong Python.

Libavg là một framework GUI sử dụng OpenGL cho Python, cho phép bạn tạo các ứng dụng tương tác với đồ họa 2D và 3D.

Thư viện này rất thích hợp cho việc phát triển các ứng dụng đa phương tiện, trình diễn, và trò chơi.

Ưu điểm:

* Hiệu suất cao: Libavg sử dụng OpenGL để render, cho phép xử lý đồ họa hiệu quả, đặc biệt là với các hình ảnh và hiệu ứng phức tạp.
* Tương tác phong phú: Hỗ trợ các tính năng tương tác như cảm ứng, chuột, và bàn phím, giúp tạo ra các ứng dụng sống động và hấp dẫn.
* Giao diện dễ sử dụng: Cung cấp các API đơn giản để xây dựng UI, cho phép lập trình viên dễ dàng tạo và quản lý các đối tượng giao diện.
* Tính năng mạnh mẽ: Hỗ trợ các tính năng như animation, rendering layer, và event handling, giúp phát triển các ứng dụng phức tạp một cách dễ dàng.

Nhược điểm:

* Cộng đồng nhỏ: So với một số thư viện GUI phổ biến khác như Tkinter, PyQt, hay Kivy, Libavg có cộng đồng người dùng nhỏ hơn, dẫn đến ít tài liệu và hỗ trợ hơn.
* Khó học hơn: Do có nhiều tính năng mạnh mẽ, Libavg có thể gây khó khăn cho người mới bắt đầu làm quen với các khái niệm như rendering pipeline và event handling.
* Hỗ trợ hạn chế cho nền tảng: Một số tính năng có thể không tương thích với tất cả các hệ điều hành, điều này có thể gây khó khăn trong việc phát triển ứng dụng đa nền tảng.

Ứng dụng:

* Trình diễn nghệ thuật số: Libavg rất phù hợp cho các dự án nghệ thuật tương tác, cho phép lập trình viên tạo ra các trải nghiệm nghệ thuật độc đáo.
* Trò chơi và ứng dụng giải trí: Có thể được sử dụng để phát triển trò chơi 2D và các ứng dụng giải trí với đồ họa phong phú.
* Ứng dụng giáo dục: Libavg có thể được sử dụng để xây dựng các ứng dụng giáo dục tương tác, giúp người học tiếp cận thông tin một cách sinh động và thú vị hơn.

## LẬP TRÌNH TRỰC QUANG HÓA DỮ LIỆU (PLOT)

<kết cấu nội dung bài PLOT vào đây>

1. **LẬP TRÌNH PYTHON PHÂN TÍCH THĂM DÒ VỀ VIỆC LÀM VÀ MỨC LƯƠNG TRONG NGÀNH PHÁT TRIỂN PHẦN MỀM NĂM 2024**

## Giới thiệu chủ đề EDA …….

Vận dụng kỹ thuật lập trình EDA kết hợp với GUI và Voice Assistant cho phép dùng tập dữ liệu thực nghiệm về lương trung bình của từng đầu công việc của Kỹ sư phần mềm ở các thành phố ở Mỹ năm 2024 để dự đoán và tư vấn các nhà quản lý hoạch định kế hoạch tuyển dụng nhân sự ngành phần mềm vào thời gian tiếp theo.

Dữ liệu thực nghiệm gồm mức lương ở các thành phố trọng điểm, các ban trong năm ở nước Mỹ:

pokemon\_combined.csv: dạng file Excel .csv

. Số thuộc tính input: 17

## Giới thiệu bài toán (Yêu cầu đặt ra)

Xét bài toán phân tích chỉ số của từng loại Pokemon. Thông qua phân tích dữ liệu đã và đang phát triển.

## Thông tin về file dữ liệu thực nghiệm EDA

File dữ liệu gốc (Dataset): pokemon\_combined.csv

https://www.kaggle.com/datasets/crinklybrain2003/pokmon-base-stats-dataset

Tác giả (Author) : Piyush Shailesh Choudhari

+ Dạng file Excel .csv

+ Thời gian thu thập dữ liệu: từ tháng 1/2024 đến nay

+ Giải thích một số cột (thuộc tính): Attributes

* 1. Tiến trình thực hiện

+ Tiến trình phân tích dữ liệu thăm dò (EDA) thông thường được thực hiện qua 3 GĐ chính:

GIAI ĐOẠN 1: NẠP DỮ LIỆU GỐC (PRIMARY INPUT DATA LOAD): Bước 1\_Bước 2

GIAI ĐOẠN 2: TIỀN XỬ LÝ (PRE-PROCESSING): Bước 3\_Bước 7

GIAI ĐOẠN 3: PHÂN TÍCH DỮ LIỆU THĂM DÒ (EDA: Exploratory Data Analysis): Bước 8\_

* 1. 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 df = df.dropna(how='any')

[2] Xử lý dữ liệu quá khác biệt = “ngoại lệ” (isolated: ….) là……………

## Các nền tảng kỹ thuật liên quan

### Hệ thống thư viện sử dụng

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 sklearn.preprocessing import MinMaxScaler

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.preprocessing import LabelEncoder

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

from sklearn.feature\_selection import SelectKBest, f\_classif

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

# -\*- coding: utf-8 -\*-

"""

Created on Tue Oct 15 09:17:29 2024

@author: Nguyễn Thành Đạt

"""

########################################################

# GIAI ĐOẠN 1: NẠP DỮ LIỆU GỐC (PRIMARY INPUT DATA LOAD)

########################################################

#######################################

# Bước 1: Nạp các thư viện cần thiết

#######################################

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 sklearn.preprocessing import MinMaxScaler

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.preprocessing import LabelEncoder

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

from sklearn.feature\_selection import SelectKBest, f\_classif

### Tập dữ liệu thực nghiệm

**Bước 2: Tải tập dữ liệu**

# Bước 2: Tải tập dữ liệu: Load the data set (Nạp tập dữ liệu)

# ./pokemon\_combined.csv

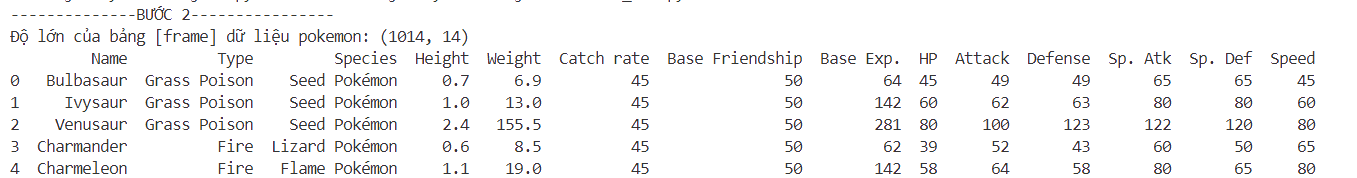
df = pd.read\_csv('./G213NTD\_EDA/pokemon\_combined.csv')

#Drop các cột Kỹ năng, Tỉ lệ phát triển và Giới tính

df = df.drop(columns=['Abilities', 'Growth Rate', 'Gender'])

# Display the shape of the data set (xem lượng dòng & cột dữ liệu của tập DL gốc)

Chạy G213NTD\_EDA.py sẽ hiển thị kết quả:



### Mô tả giải thuật

**Bước 3: Xử lý dữ liệu**

# # Bước 3: Xử lý CỘT dữ liệu NULL quá nhiều OR không có giá trị phân tích

############################################################################

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

print("--------------BƯỚC 3----------------")

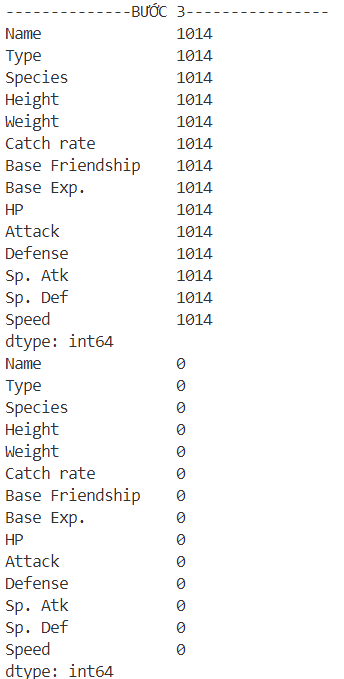
print(df.count().sort\_values()) #df.count(): đếm số lượng dòng có dữ liệu của df, .sort\_values() sx tăng dân

# kiểm tra lại sự tồn tại của giá trị null

print(df.isna().sum())

print(df.shape) # kiểm tra lại số lượng cột & dòng của df sau khi XL NULL cột

**Kết quả:**



#####################################

# # Bước 4: Xử lý DÒNG dữ liệu NULL

###################################

# Removing null values (Xóa tất cả các dòng có giá trị null trong tập FRAME dữ liệu.)

print("--------------BƯỚC 4----------------")

df = df.dropna(how='any')

print(df.shape) # kiểm tra lại số lượng cột & dòng của df sau khi XL NULL các dòng DL

# =>(1014,17) không NULL



##############################################################################

# # # Bước 5: RR THEO Mã hóa trược tiếp: Thay thế các vị trí giá trị  0 và 1 bởi CÓ (Yes) và KHÔNG (No).

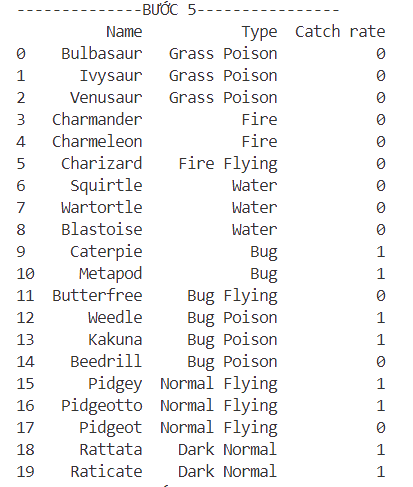
# ##############################################################################

# # #Thay thế vị trí giá trị có Type : Fire tương ứng cột|biến Fire]

print("--------------BƯỚC 5----------------")

df['Catch rate'] = df['Catch rate'].apply(lambda x: 1 if x > 100 else 0)  # Các Pokemon Type : Fire -> Có

print(df[['Name', 'Type', 'Catch rate']].head(20))



##############################################

# # # Bước 6: Xử lý loại bỏ các giá trị ngoại lệ (cá biệt): isolated

# ##################################################################

# # #kiểm tra tập dữ liệu có bất kỳ ngoại lệ nào không

print("--------------BƯỚC 6----------------")

numeric\_cols = df.select\_dtypes(include=[np.number]).columns

z = np.abs(stats.zscore(df[numeric\_cols], nan\_policy='omit')) # Dò tìm và lấy các giá trị cá biệt trong tập dữ liệu gốc thông qua điểm z (z\_score)

print('MA TRAN Z-SCORE\n')

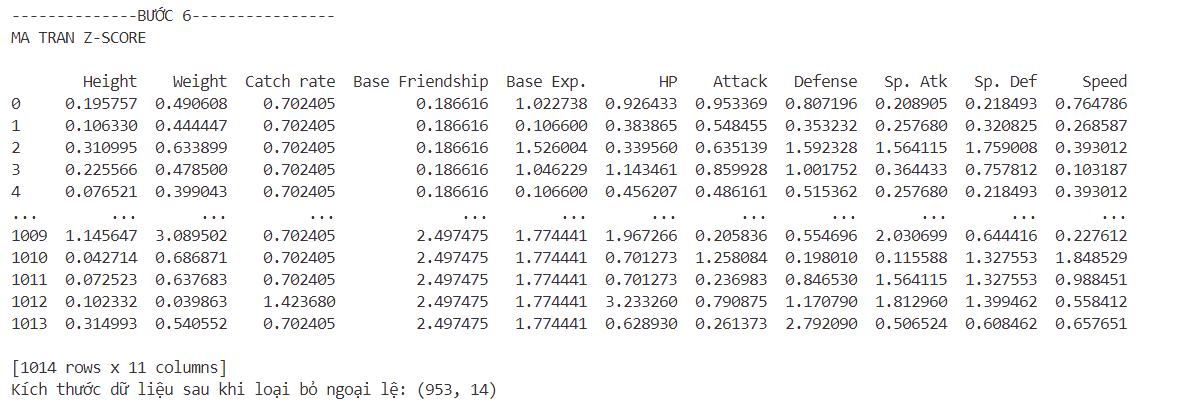
print(z) # in ra tập (ma trận) các giá trị z-score từ tập dữ liệu gốc

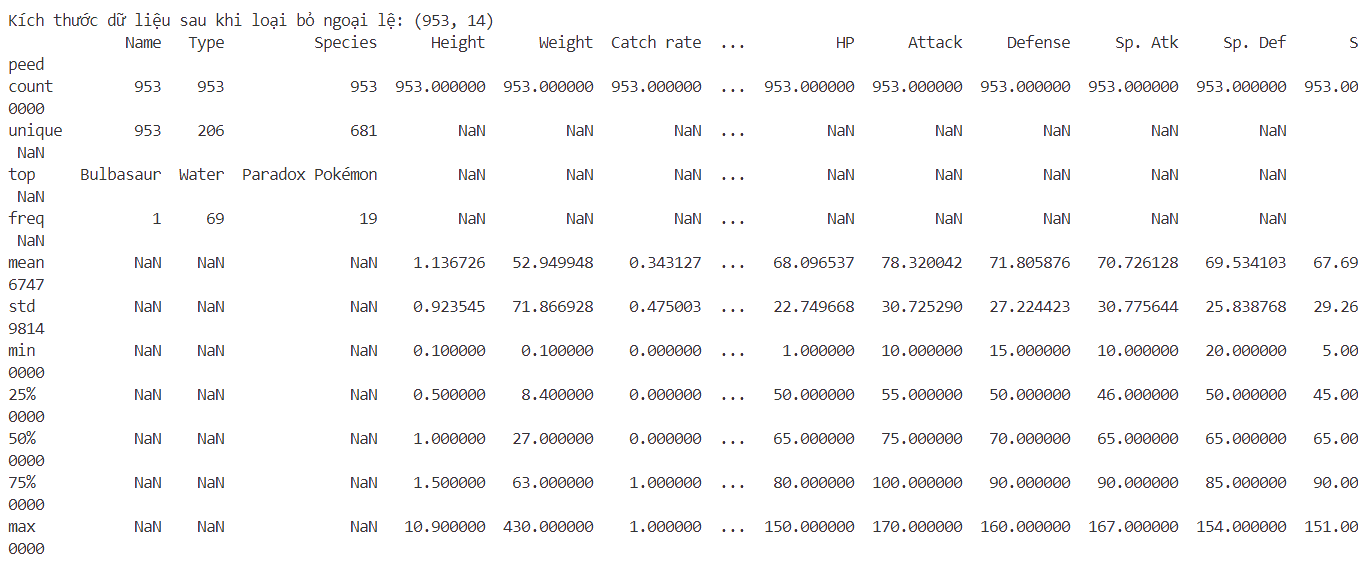
df= df[(z < 3).all(axis=1)] # kiểm tra và chỉ giữ lại trong df các giá trị số liệu tưng ứng với z-score < 3  # {loại các giá trị >= 3} vì các giá trị z-score >=3 tướng ứng với số liệu quá khác biệt so với các số liệu còn lại (“cá biệt” = “ngoại lệ” = isolated}

print('Kích thước dữ liệu sau khi loại bỏ ngoại lệ:', df.shape)

# Hiển thị thống kê tổng quan sau khi xử lý ngoại lệ

print(df.describe(include='all'))





################################################################

# # #Bước 7: RR hóa theo khoảng / đoạn = Chuẩn hóa (Rời rạc hóa) tập dữ liệu Input dùng ..MaxMin

# ####################################################################

print("--------------BƯỚC 7----------------")

# # Tách các cột không cần chuẩn hóa (ví dụ: cột 'name')

columns\_to\_exclude = ['Name', 'Type', 'Species']

columns\_to\_scale = [col for col in df.columns if col not in columns\_to\_exclude]

# Rời rạc hóa các cột số

# xác định thang đo sẽ dùng để RR hóa theo khoảng đều

scaler = MinMaxScaler()

# Áp dụng thang đo vào data frame [df] của đề tài (đã tiền xử lý đến Bước 5)

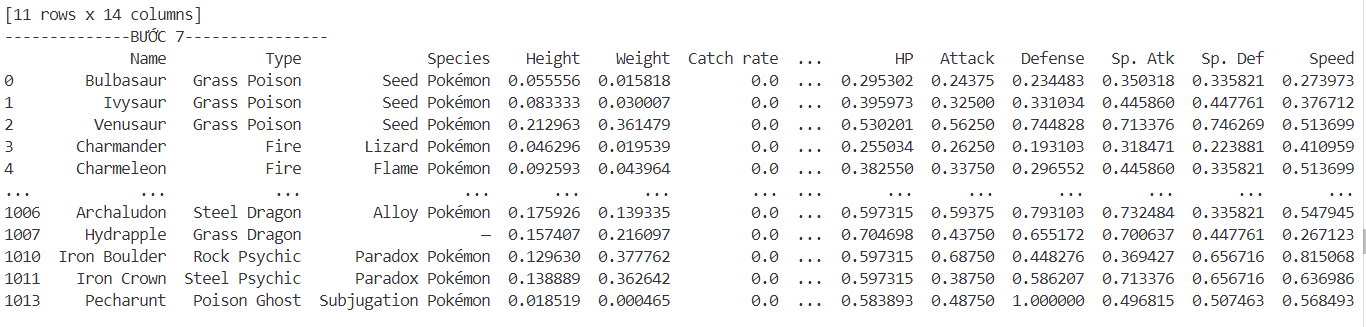
df\_scaled = pd.DataFrame(scaler.fit\_transform(df[columns\_to\_scale]), columns=columns\_to\_scale, index=df.index)

# Kết hợp lại DataFrame

df = pd.concat([df[columns\_to\_exclude], df\_scaled], axis=1)

df.iloc[4:10]

print(df)



# # #Bước 8: Xác định mô hình trích lọc các thuộc tính đặc trưng: EDA

# # xác định tỉ lệ bắt - Catch rate của Pokemon

# # dựa trên tập dữ liệu đã được rời rạc hóa ở bước 7

print("--------------BƯỚC 8----------------")

X = df\_scaled.loc[:,df\_scaled.columns!='Catch rate'] # xác định tập DL Input (X) = All trừ (chú ý !=) cột DL đoán đầu ra Catch rate

y = df\_scaled[['Catch rate']].values.ravel() # xác định tập DL ra legendary

selector = SelectKBest(f\_classif, k=5) # sd các hàm SelectKBest trong thư viện sklearn = Mô hình xác định các Thuộc tính quan trọng quyết định việc dự đoàn DL output = trích lọc Đặc trưng = Feature Extraction: k = 5 (đ/v bài này)

selector.fit(X, y) # Áp dụng mô hình trên vào SelectKBest

X\_new = selector.transform(X) # Chuyên DL Input theo mô hình

print('Ma trận input sau khi áp dụng chỉ số k Best')

print(X\_new)

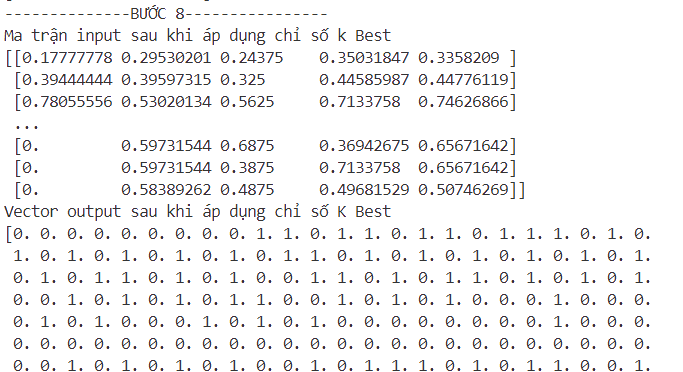
print('Vector output sau khi áp dụng chỉ số K Best')

print(y)

# #########################################

print('k cột quan trọng nhất quyết định Vector Output')

print(X.columns[selector.get\_support(indices=True)]) # in ds các tt đặc trưng



# #Bước 9: Xác định mô hình trích lọc các thuộc tính đặc trưng

# # XĐ data frame = Chiếu lấy các thuộc tính đặc trưng đã xđ trong B8

df = df[['HP', 'Attack', 'Sp. Atk', 'Sp. Def', 'Speed']] # máu, tấn công, tấn công đặc biệt, phòng thủ đặc biệt, tốc độ

# #Bước 10: EDA theo nhu cầu thực tế => input vào các mô hình AI, ML,...

# # Đơn giản nhất là lấy 1 thuộc tính đầu vào (total) để XD Mô hình

print("--------------BƯỚC 10----------------")

X = df[['HP']]

y = df[['Attack']]

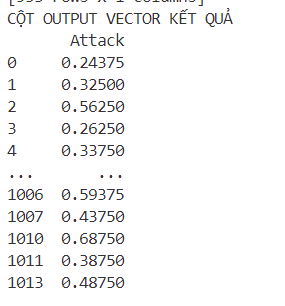
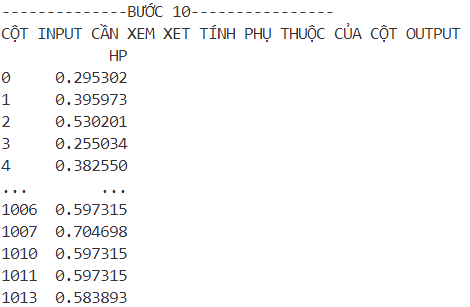
print('CỘT INPUT CẦN XEM XET TÍNH PHỤ THUỘC CỦA CỘT OUTPUT')

print(X)

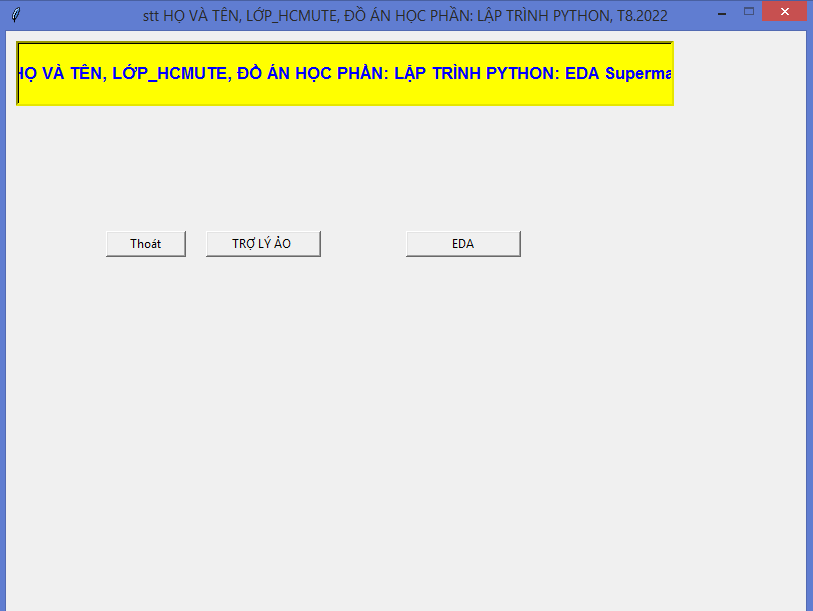
print('CỘT OUTPUT VECTOR KẾT QUẢ ')

print(y)

# # ############CÁC KQ TRÊN SẼ INPUT VÀO CÁC  MÔ HÌNH AI & ML,...



## Thiết kế giao diện



## Mã lệnh lập trình (python)

# -\*- coding: utf-8 -\*-

"""

Created on Sat Jul 16 19:59:48 2023

@author: VOXUAN

"""

# B1: NẠP THƯ VIỆN

# Speech

import speech\_recognition as sr02vtth

from gtts import gTTS

import playsound

# tkinter

import tkinter as tk02vtth

from tkinter import messagebox

# 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

vtth02\_FILE = "vtth020.mp3" # lưu tên file Input

vtth02\_DIR = 'vtth02' # Thư mục lưu các file [trên]

os.makedirs(vtth02\_DIR, exist\_ok=True) # TẠO THƯ MỤC LƯU (từ thư viện os - của OS MS. Windows))

"""

CÁC HÀM THỰC HIỆN CÁC CHỨC NĂNG: HỆ THỐNG

"""

def Thoat():

traloi = messagebox.askquestion("Xác nhận","Thiệt thoát không (Y/N)?")

if traloi == "yes": wn.destroy()

"""

CÁC HÀM THỰC HIỆN CÁC CHỨC NĂNG = SPEECH

"""

def Lenh(): # NHẬP ÂM THANH TỪ MICROPHONE

r = sr02vtth.Recognizer()

with sr02vtth.Microphone() as Source:

#hiệu chỉnh mic để chuẩn bị nói

messagebox.showinfo("Nhắc nhở", "Hieu chinh nhieu trươc khi noi!")

r.adjust\_for\_ambient\_noise(Source, duration=1)

#nhận lời nói ra lệnh từ người dùng thông qua MIc [mặc định] lưu dữ liệu âm thanh vào audio\_data

messagebox.showinfo("Cảnh báo", "Bấm OK để bắt đầu Chọn lệnh bằng tiếng Việt, trong 3s" )

audio\_data = r.record(Source, duration = 3)

try:

vlenh = r.recognize\_google(audio\_data,language="vi")

except:

vlenh = "Quý vị nói gì nghe không rõ...!"

# xuất kết quả ra

messagebox.showinfo("Quý vị đã nói là", format(vlenh))

vText = gTTS(text=vlenh, lang = 'vi')

#vFile = '06VXT.mp3'

vText.save(vtth02\_FILE)

playsound.playsound(vtth02\_FILE)

"""

CÁC HÀM THỰC HIỆN CÁC CHỨC NĂNG = EDA

"""

def EDA():

df = pd.read\_csv('./GxttHoTen\_Topic.csv')

# Display the shape of the data set (xem lượng dòng & cột dữ liệu của tập DL gốc)

messagebox.showinfo("Độ lớn của bảng [frame] dữ liệu thời tiết", df.shape)

""""""""""""""""""""""""""""""""""""""""""""

""" CÁC NỘI DUNG SAU CHƯA THỰC HIỆN

""""""""""""""""""""""""""""""""""""""""""

# Bước 3: Xử lý CỘT dữ liệu NULL quá nhiều OR không có giá trị phân tích

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

messagebox.showinfo("CÁC CỘT DỬ LIỆU SẮP XẾP THEO THỨ TỰ CÓ DỮ LIỆU TỪ ÍT => NHIỀU",df.count().sort\_values()) #df.count(): đếm số lượng dòng có dữ liệu của df, .sort\_values() sx tăng dân

df = df.drop(columns=['Sunshine','Evaporation','Cloud3pm','Cloud9am','Location','Date','RISK\_MM'],axis=1)

#df = df.drop(columns=['Sunshine','Evaporation','Cloud3pm','Cloud9am','Pressure9am',# 'Pressure3pm','WindDir3pm', 'WindDir9am', 'WindGustDir',# 'WindGustSpeed','Location','Date','RISK\_MM'],axis=1)

messagebox.showinfo("Độ lớn của bảng [frame] dữ liệu SAU KHI XỬ LÝ CỘT NULL", df.shape) # kiểm tra lại số lượng cột & dòng của df sau khi XL NULL cột

# Bước 4: Xử lý DÒNG dữ liệu NULL

# Removing null values (Xóa tất cả các dòng có giá trị null trong tập FRAME dữ liệu.)

df = df.dropna(how='any')

messagebox.showinfo("Độ lớn của bảng [frame] dữ liệu SAU KHI XỬ LÝ DÒNG NULL", df.shape) # kiểm tra lại số lượng cột & dòng của df sau khi XL NULL các dòng DL

# Bước 5: Xử lý loại bỏ các giá trị ngoại lệ (cá biệt): isolated

#kiểm tra tập dữ liệu có bất kỳ ngoại lệ nào không

z = np.abs(stats.zscore(df.\_get\_numeric\_data())) # Dò tìm và lấy các giá trị cá biệt trong tập dữ liệu gốc thông qua điểm z (z\_score)

messagebox.showinfo("MA TRAN Z-SCORE", z) # in ra tập (ma trận) các giá trị z-score từ tập dữ liệu gốc

df= df[(z < 3).all(axis=1)] # kiểm tra và chỉ giữ lại trong df các giá trị số liệu tưng ứng với z-score < 3 # {loại các giá trị >= 3} vì các giá trị z-score >=3 tướng ứng với số liệu quá khác biệt so với các số liệu còn lại (“cá biệt” = “ngoại lệ” = isolated}

messagebox.showinfo("Độ lớn của bảng [frame] dữ liệu SAU KHI XỬ LÝ NGOẠI LỆ", df.shape)# xác định số dòng & cột dữ liệu sau khu xử lý các giá trị cá biệt

# Bước 6: Thay thế các vị trí giá trị 0 và 1 bởi CÓ (Yes) và KHÔNG (No).

#Thay thế yes (CÓ) and no (KO) vào vị trí giá trị 1 (Y) và 0 (N) tương ứng cột|biến RainToday và# RainTomorrow

df['RainToday'].replace({'KHONG': 'No', 'CO': 'Yes'},inplace = True)

df['RainTomorrow'].replace({'KHONG': 'No', 'CO': 'Yes'},inplace = True)

#Bước 7: Chuẩn hóa (Rời rạc hóa) tập dữ liệu Input dùng ..MaxMin

# # CHUẨN HÓA DL

# scaler = preprocessing.MinMaxScaler() #preprocessing là Sub-Library của thư viện sklearn=> hàm .MinMaxScaler() Rời rạc hóa tập dữ liệu Input

# scaler.fit(df)

# df = pd.DataFrame(scaler.transform(df), index=df.index, columns=df.columns) # Rời rạc hóa số liệu theo thang đo scaler

# df.iloc[4:10]

# messagebox.showinfo("Độ lớn của bảng [frame] dữ liệu SAU KHI CHUẨN HÓA DL", df.shape)# xác định số dòng & cột dữ liệu sau khu xử lý các giá trị cá biệt

# # GIAI ĐOẠN 3: PHÂN TÍCH DỮ LIỆU THĂM DÒ : EDA

# # Bước 8: Nạp các thuộc tính quan trọng vào Dataset

# #The important features are put in a data frame

# df = df[['Humidity3pm','Rainfall','RainToday','RainTomorrow']]

# # Bước 9: thực hiện các tính toán trên mô hình phân tích

# #To simplify computations we will use only one feature (Humidity3pm) to build the model

# X = df

# X = df[['Humidity3pm']]

# y = df[['RainTomorrow']]

# X = df.loc[:,df.columns!='RainTomorrow']

# y = df[['RainTomorrow']]

# selector = SelectKBest(chi2, k=3)

# selector.fit(X, y)

# X\_new = selector.transform(X)

# df(['Rainfall', 'Humidity3pm', 'RainToday'], dtype='object')

# messagebox.showinfo(" KÊT QUẢ", X.columns[selector.get\_support(indices=True)])# xác định số dòng & cột dữ liệu sau khu xử lý các giá trị cá biệt

"""

"""

THỦ TỤC CHÍNH = GUI

"""

# B3: LẬP GUI (EX4)

# Tạo một cửa sổ mới

wn = tk02vtth.Tk()

#Thêm tiêu đề cho cửa sổ

wn.title("stt HỌ VÀ TÊN, LỚP\_HCMUTE, ĐỒ ÁN HỌC PHẦN: LẬP TRÌNH PYTHON, T8.2023")

#Đặt kích thước của cửa sổ

wn.geometry('800x600')

#Không cho thay đổi size

wn.resizable(tk02vtth.FALSE, tk02vtth.FALSE)

#Tiêu đề Form = tên đề tài

t = "stt HỌ VÀ TÊN, LỚP\_HCMUTE, ĐỒ ÁN HỌC PHẦN: LẬP TRÌNH PYTHON: EDA Supermarket"

lblDT = tk02vtth.Label(wn, text=t, background = "yellow", fg = "blue", relief = tk02vtth.SUNKEN, font=("Arial Bold", 13), borderwidth = 3, width = 65, height = 3)

lblDT.place(x = 10, y = 10)

# B4: CÁC NÚT LỆNH: EX1 = Speech

#Thoát

btnThoat = tk02vtth.Button(wn, text = "Thoát", width = 10, command = Thoat)

btnThoat.place(x =100, y = 200)# căn cứ vào kích thước form [wn.geometry("800x600")] => canh vị trí Button "thoát"

#Xử lý lời nói = speech = VOICE ASSISTANT

btnNoi = tk02vtth.Button(wn, text = "TRỢ LÝ ẢO ", width = 15, command = Lenh)

btnNoi.place(x = 200, y = 200)

# B5: CÁC NÚT LỆNH: EX3 = EDA = DỰ PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

btnEDA = tk02vtth.Button(wn, text = "EDA", width = 15, command = EDA)

btnEDA.place(x = 400, y = 200)

#Lặp vô tận để hiển thị cửa sổ

wn.mainloop()

1. **LẬP TRÌNH HỖ TRỢ KỸ THUẬT THỊ GIÁC MÁY TÍNH (CV) NHẬN DIỆN TRÁI CÂY TỪ CLIP GIAN HÀNG SIÊU THỊ**

## Giới thiệu chủ đề lập trình

. Hỗ trợ: Computer Vision [Thị giác máy tính]

. Gồm 2 phần chính:

+ Image\_Procceing (Xử lý ảnh): xử lý ảnh = các thao tác xử lý đơn giản, như: quay, co \_ giãn, lật, cắt, chuyển xám<->màu (? bit),…, histogram, chuyển hệ màu (RGB)-> rời rạc hóa ảnh => phân tích EDA, nhận diện biên ảnh … = PHÂN TÍCH THĂM DÒ EDA ĐỐI VỚI DỮ LIỆU MULTIMEDIA

[1] Image\_Procceing: Các xử lý đơn giản trên ảnh (images)

+Frames from Video (cắt Frames từ Video)

[2] Frames from Video: Cắt khung ảnh (image frames từ Video Clip)

HO TRO = EDA Áp dụng cho CV (Computer Vision: Thị giác máy tính)

+ Thư viện chính :  cv2

python -m pip install opencv-python

OR pip3 install opencv-python

*Successfully installed opencv-python-4.10.0.84*

## Cắt frames ảnh từ Video Clips

### Giới thiệu về cắt frame ảnh: frame

…

### Full codes bài làm

import cv2

import os

# Hàm trích xuất frame từ video

def extract\_frames(video\_path, output\_folder="frames"):

    # Mở video bằng OpenCV

    cap = cv2.VideoCapture(video\_path)

    # Kiểm tra nếu video không thể mở

    if not cap.isOpened():

        print("Không thể mở video!")

        return

    # Tạo thư mục lưu các frame nếu chưa tồn tại

    if not os.path.exists(output\_folder):

        os.makedirs(output\_folder)

    frame\_count = 0

    # Đọc video và trích xuất các frame

    while True:

        ret, frame = cap.read()

        if not ret:

            break  # Dừng khi không còn frame nào để đọc

        # Lưu frame dưới dạng ảnh

        frame\_filename = os.path.join(output\_folder, f"frame\_{frame\_count:04d}.jpg")

        cv2.imwrite(frame\_filename, frame)

        print(f"Đã lưu frame {frame\_count} với tên: {frame\_filename}")

        frame\_count += 1

    # Giải phóng tài nguyên

    cap.release()

    print(f"Đã hoàn thành trích xuất {frame\_count} frame.")

# Ví dụ sử dụng hàm

video\_path = "./G213NTD\_CV/sg2.mp4"  # Đường dẫn đến video của bạn

extract\_frames(video\_path)

### Yêu cầu của bài thực hành

Sv cá nhân hóa thông tin bài làm

## Xử lý ảnh

### Giới thiệu về xử lý ảnh: images

### Full codes bài làm

import cv2

import os

import tkinter as tk

from tkinter import filedialog, messagebox

from PIL import Image, ImageTk, ImageFilter

# Hàm làm mờ ảnh

def blur\_image\_G213NTD(image):

    return image.filter(ImageFilter.GaussianBlur(radius=5))

# Hàm làm sắc nét ảnh

def sharpen\_image\_G213NTD(image):

    return image.filter(ImageFilter.UnsharpMask(radius=2, percent=150, threshold=3))

# Hàm chuyển ảnh sang đen trắng

def convert\_to\_bw\_G213NTD(image):

    return image.convert("L")

# Hàm cắt ảnh

def crop\_image\_G213NTD(image, left, top, right, bottom):

    return image.crop((left, top, right, bottom))

# Hàm resize ảnh

def resize\_image\_G213NTD(image, width, height):

    return image.resize((width, height))

# Hàm tải ảnh từ máy tính

def load\_image\_G213NTD():

    file\_path = filedialog.askopenfilename(filetypes=[("Image Files", "\*.jpg;\*.jpeg;\*.png;\*.bmp")])

    if file\_path:

        try:

            image = Image.open(file\_path)

            return image

        except Exception as e:

            messagebox.showerror("Error", f"Không thể mở ảnh: {str(e)}")

            return None

    return None

# Hàm lưu ảnh sau khi xử lý

def save\_image\_G213NTD(image):

    file\_path = filedialog.asksaveasfilename(defaultextension=".png", filetypes=[("PNG Files", "\*.png"), ("JPEG Files", "\*.jpg")])

    if file\_path:

        try:

            image.save(file\_path)

            messagebox.showinfo("Success", "Ảnh đã được lưu thành công!")

        except Exception as e:

            messagebox.showerror("Error", f"Không thể lưu ảnh: {str(e)}")

class VideoFrameExtractorApp\_G213NTD:

    def \_\_init\_\_(self, root):

        self.root = root

        self.root.title("Video Frame Extractor")

        self.root.geometry("500x600")

        # Video path và image path

        self.video\_path = None

        self.current\_image = None

        # Tạo các widget cho GUI

        self.create\_widgets()

    def create\_widgets(self):

        # Nút chọn video

        self.select\_button = tk.Button(self.root, text="Select Video", command=self.select\_video)

        self.select\_button.pack(pady=20)

        # Label hiển thị video path

        self.video\_label = tk.Label(self.root, text="No video selected", wraplength=400)

        self.video\_label.pack(pady=10)

        # Nút trích xuất frame

        self.extract\_button = tk.Button(self.root, text="Extract Frames", state=tk.DISABLED, command=self.extract\_frames)

        self.extract\_button.pack(pady=20)

        # Nút tải ảnh

        self.load\_button = tk.Button(self.root, text="Tải ảnh", command=self.load\_image\_G213NTD)

        self.load\_button.pack(pady=10)

        # Nút lưu ảnh

        self.save\_button = tk.Button(self.root, text="Lưu ảnh", state=tk.DISABLED, command=self.save\_image\_G213NTD)

        self.save\_button.pack(pady=10)

        # Nút làm mờ ảnh

        self.blur\_button = tk.Button(self.root, text="Làm mờ ảnh", state=tk.DISABLED, command=self.blur\_image\_G213NTD)

        self.blur\_button.pack(pady=5)

        # Nút làm sắc nét ảnh

        self.sharpen\_button = tk.Button(self.root, text="Làm sắc nét ảnh", state=tk.DISABLED, command=self.sharpen\_image\_G213NTD)

        self.sharpen\_button.pack(pady=5)

        # Nút chuyển sang đen trắng

        self.bw\_button = tk.Button(self.root, text="Chuyển sang đen trắng", state=tk.DISABLED, command=self.convert\_to\_bw\_G213NTD)

        self.bw\_button.pack(pady=5)

        # Nút cắt ảnh

        self.crop\_button = tk.Button(self.root, text="Cắt ảnh", state=tk.DISABLED, command=self.crop\_image\_G213NTD)

        self.crop\_button.pack(pady=5)

        # Nút resize ảnh

        self.resize\_button = tk.Button(self.root, text="Resize ảnh", state=tk.DISABLED, command=self.resize\_image\_G213NTD)

        self.resize\_button.pack(pady=5)

        # Status label

        self.status\_label = tk.Label(self.root, text="Ready to extract frames.", wraplength=400)

        self.status\_label.pack(pady=10)

        # Label để hiển thị ảnh

        self.image\_label = tk.Label(self.root)

        self.image\_label.pack(pady=20)

    def load\_image\_G213NTD(self):

        self.current\_image = load\_image\_G213NTD()

        if self.current\_image:

            self.update\_image\_display()

            # Kích hoạt các nút chỉnh sửa ảnh

            self.save\_button.config(state=tk.NORMAL)

            self.blur\_button.config(state=tk.NORMAL)

            self.sharpen\_button.config(state=tk.NORMAL)

            self.bw\_button.config(state=tk.NORMAL)

            self.crop\_button.config(state=tk.NORMAL)

            self.resize\_button.config(state=tk.NORMAL)

    def select\_video(self):

        # Mở cửa sổ chọn file video

        self.video\_path = filedialog.askopenfilename(title="Select Video", filetypes=[("MP4 files", "\*.mp4"), ("All files", "\*.\*")])

        # Hiển thị đường dẫn video

        if self.video\_path:

            self.video\_label.config(text=f"Selected Video: {self.video\_path}")

            self.extract\_button.config(state=tk.NORMAL)  # Kích hoạt nút trích xuất frame

        else:

            self.video\_label.config(text="No video selected")

    def extract\_frames(self):

        # Kiểm tra nếu không chọn video

        if not self.video\_path:

            messagebox.showerror("Error", "Please select a video first!")

            return

        # Tạo thư mục lưu các frame

        output\_folder = "frames"

        if not os.path.exists(output\_folder):

            os.makedirs(output\_folder)

        # Mở video

        cap = cv2.VideoCapture(self.video\_path)

        if not cap.isOpened():

            messagebox.showerror("Error", "Failed to open video.")

            return

        # Trích xuất frame

        frame\_count = 0

        while True:

            ret, frame = cap.read()

            if not ret:

                break

            # Lưu từng frame dưới dạng ảnh

            frame\_filename = os.path.join(output\_folder, f"frame\_{frame\_count:04d}.jpg")

            cv2.imwrite(frame\_filename, frame)

            frame\_count += 1

        cap.release()

        # Hiển thị thông báo hoàn tất

        self.status\_label.config(text=f"Extracted {frame\_count} frames successfully.")

        messagebox.showinfo("Success", f"{frame\_count} frames extracted and saved to '{output\_folder}'.")

    def save\_image\_G213NTD(self):

        if self.current\_image:

            save\_image\_G213NTD(self.current\_image)

        else:

            messagebox.showwarning("Warning", "Chưa có ảnh để lưu!")

    def blur\_image\_G213NTD(self):

        if self.current\_image:

            self.current\_image = blur\_image\_G213NTD(self.current\_image)

            self.update\_image\_display()

        else:

            messagebox.showwarning("Warning", "Chưa có ảnh để làm mờ!")

    def sharpen\_image\_G213NTD(self):

        if self.current\_image:

            self.current\_image = sharpen\_image\_G213NTD(self.current\_image)

            self.update\_image\_display()

        else:

            messagebox.showwarning("Warning", "Chưa có ảnh để làm sắc nét!")

    def convert\_to\_bw\_G213NTD(self):

        if self.current\_image:

            self.current\_image = convert\_to\_bw\_G213NTD(self.current\_image)

            self.update\_image\_display()

        else:

            messagebox.showwarning("Warning", "Chưa có ảnh để chuyển sang đen trắng!")

    def crop\_image\_G213NTD(self):

        if self.current\_image:

            # Cắt ảnh (mặc định: cắt từ 50, 50 tới 250, 250)

            left = 50

            top = 50

            right = self.current\_image.width - 50

            bottom = self.current\_image.height - 50

            self.current\_image = crop\_image\_G213NTD(self.current\_image, left, top, right, bottom)

            self.update\_image\_display()

        else:

            messagebox.showwarning("Warning", "Chưa có ảnh để cắt!")

    def resize\_image\_G213NTD(self):

        if self.current\_image:

            # Resize ảnh (mặc định: 300x300)

            width = 300

            height = 300

            self.current\_image = resize\_image\_G213NTD(self.current\_image, width, height)

            self.update\_image\_display()

        else:

            messagebox.showwarning("Warning", "Chưa có ảnh để resize!")

    def update\_image\_display(self):

        if self.current\_image:

            # Chuyển đổi ảnh từ PIL Image sang định dạng mà tkinter có thể hiển thị

            img\_tk = ImageTk.PhotoImage(self.current\_image)

            self.image\_label.config(image=img\_tk)

            self.image\_label.image = img\_tk  # Lưu tham chiếu đến ảnh để tránh bị garbage collection

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

    root = tk.Tk()

    app = VideoFrameExtractorApp\_G213NTD(root)

    root.mainloop()

### Yêu cầu của bài thực hành

Sv cá nhân hóa thông tin bài làm

## Yêu cầu bài thực nghiệm: Xử lý ảnh trên các frames đã cắt từ Video Clips

## Giới thiệu các yêu cầu của bài thực nghiệm

1. Sv cá nhân hóa thông tin bài làm
2. YÊU CẦU 1: CHO PHÉP CHỌN FILE Video (Dùng text liệt kê & chọn: tham khảo: MauCode Choice.pdf): ...mp4 OR .mov,...

..

1. YÊU CẦU 2: CHO PHÉP CHỌN PHƯƠNG ÁN XỬ LÝ TRÙNG CÁC FILE FRAMEs CẮT ĐƯỢC

+ PA1: THAY ĐỔI TÊN các FILE FRAME THEO LẦN CHẠY

+ PA2: LƯU CHÔNG LÊN FILEs CŨ

+ PA3: CHO NSD CHỌN TÊN FILE CHO CÁC FAMREs CẮT ĐƯỢC, nên gợi ý NSD, Ex: <Tên\_file><Chỉ số>

+ PA4: cHO PHÉP: CHỌN PATH LƯU CÁC FRAMEs CẮT ĐƯỢC & NEW FOLDER (nếu cần)

+ SV TỰ THÊM CÁC PA KHÁC (nếu có)

1. YÊU CẦU 3: MỞ VIDEO = CHO NSD CHỌN THỜI ĐIỂM CẮT FANRE THEO Ý MUỐN

+ x.đ độ dài

1. YÊU CẦU 4: CHO PHÉP CHỌN THỜI ĐIỂM BẮT ĐẦU & KẾT THÚC TRONG VIDEO GỐC = CẦN CẮT FRAMEs (cần lấy times duration của Video)
2. •YÊU CẦU 5: GHÉP BÀI XỬ LÝ ẢNH VÀO BÀI NÀY=XỬ LÝ ẢNH TRÊN FRAMEs CẮT ĐƯỢC (dùng Text => chọn: tham khảo: MauCode Choice.pdf)

## Giới thiệu các Video clip …….thực nghiệm

+ Seoul.mp4: 10.6 MB, dài 15s, loại mp4 = dạng clip thông thường.

## Full codes bài làm

**FULL CODE CÓ GIAO DIỆN (GUI) DÙNG THƯ VIỆN: tkinter**

import cv2

import tkinter as tk

from tkinter import filedialog, simpledialog, messagebox

import os

import cv2

import os

import tkinter as tk

from tkinter import filedialog, messagebox

from PIL import Image, ImageTk, ImageFilter

# Hàm làm mờ ảnh

def blur\_image\_G213NTD(image):

    return image.filter(ImageFilter.GaussianBlur(radius=5))

# Hàm làm sắc nét ảnh

def sharpen\_image\_G213NTD(image):

    return image.filter(ImageFilter.UnsharpMask(radius=2, percent=150, threshold=3))

# Hàm chuyển ảnh sang đen trắng

def convert\_to\_bw\_G213NTD(image):

    return image.convert("L")

# Hàm cắt ảnh

def crop\_image\_G213NTD(image, left, top, right, bottom):

    return image.crop((left, top, right, bottom))

# Hàm resize ảnh

def resize\_image\_G213NTD(image, width, height):

    return image.resize((width, height))

# Hàm tải ảnh từ máy tính

def load\_image\_G213NTD():

    file\_path = filedialog.askopenfilename(filetypes=[("Image Files", "\*.jpg;\*.jpeg;\*.png;\*.bmp")])

    if file\_path:

        try:

            image = Image.open(file\_path)

            return image

        except Exception as e:

            messagebox.showerror("Error", f"Không thể mở ảnh: {str(e)}")

            return None

    return None

# Hàm lưu ảnh sau khi xử lý

def save\_image\_G213NTD(image):

    file\_path = filedialog.asksaveasfilename(defaultextension=".png", filetypes=[("PNG Files", "\*.png"), ("JPEG Files", "\*.jpg")])

    if file\_path:

        try:

            image.save(file\_path)

            messagebox.showinfo("Success", "Ảnh đã được lưu thành công!")

        except Exception as e:

            messagebox.showerror("Error", f"Không thể lưu ảnh: {str(e)}")

class VideoFrameExtractorApp\_G213NTD:

    def \_\_init\_\_(self, root):

        self.root = root

        self.root.title("Video Frame Extractor")

        self.root.geometry("500x600")

        # Video path và image path

        self.video\_path = None

        self.current\_image = None

        # Tạo các widget cho GUI

        self.create\_widgets()

    def create\_widgets(self):

        # Nút chọn video

        self.select\_button = tk.Button(self.root, text="Select Video", command=self.select\_video)

        self.select\_button.pack(pady=20)

        # Label hiển thị video path

        self.video\_label = tk.Label(self.root, text="No video selected", wraplength=400)

        self.video\_label.pack(pady=10)

        # Nút trích xuất frame

        self.extract\_button = tk.Button(self.root, text="Extract Frames", state=tk.DISABLED, command=self.extract\_frames)

        self.extract\_button.pack(pady=20)

        # Nút tải ảnh

        self.load\_button = tk.Button(self.root, text="Tải ảnh", command=self.load\_image\_G213NTD)

        self.load\_button.pack(pady=10)

        # Nút lưu ảnh

        self.save\_button = tk.Button(self.root, text="Lưu ảnh", state=tk.DISABLED, command=self.save\_image\_G213NTD)

        self.save\_button.pack(pady=10)

        # Nút làm mờ ảnh

        self.blur\_button = tk.Button(self.root, text="Làm mờ ảnh", state=tk.DISABLED, command=self.blur\_image\_G213NTD)

        self.blur\_button.pack(pady=5)

        # Nút làm sắc nét ảnh

        self.sharpen\_button = tk.Button(self.root, text="Làm sắc nét ảnh", state=tk.DISABLED, command=self.sharpen\_image\_G213NTD)

        self.sharpen\_button.pack(pady=5)

        # Nút chuyển sang đen trắng

        self.bw\_button = tk.Button(self.root, text="Chuyển sang đen trắng", state=tk.DISABLED, command=self.convert\_to\_bw\_G213NTD)

        self.bw\_button.pack(pady=5)

        # Nút cắt ảnh

        self.crop\_button = tk.Button(self.root, text="Cắt ảnh", state=tk.DISABLED, command=self.crop\_image\_G213NTD)

        self.crop\_button.pack(pady=5)

        # Nút resize ảnh

        self.resize\_button = tk.Button(self.root, text="Resize ảnh", state=tk.DISABLED, command=self.resize\_image\_G213NTD)

        self.resize\_button.pack(pady=5)

        # Status label

        self.status\_label = tk.Label(self.root, text="Ready to extract frames.", wraplength=400)

        self.status\_label.pack(pady=10)

        # Label để hiển thị ảnh

        self.image\_label = tk.Label(self.root)

        self.image\_label.pack(pady=20)

    def load\_image\_G213NTD(self):

        self.current\_image = load\_image\_G213NTD()

        if self.current\_image:

            self.update\_image\_display()

            # Kích hoạt các nút chỉnh sửa ảnh

            self.save\_button.config(state=tk.NORMAL)

            self.blur\_button.config(state=tk.NORMAL)

            self.sharpen\_button.config(state=tk.NORMAL)

            self.bw\_button.config(state=tk.NORMAL)

            self.crop\_button.config(state=tk.NORMAL)

            self.resize\_button.config(state=tk.NORMAL)

    def select\_video(self):

        # Mở cửa sổ chọn file video

        self.video\_path = filedialog.askopenfilename(title="Select Video", filetypes=[("MP4 files", "\*.mp4"), ("All files", "\*.\*")])

        # Hiển thị đường dẫn video

        if self.video\_path:

            self.video\_label.config(text=f"Selected Video: {self.video\_path}")

            self.extract\_button.config(state=tk.NORMAL)  # Kích hoạt nút trích xuất frame

        else:

            self.video\_label.config(text="No video selected")

    def extract\_frames(self):

        # Kiểm tra nếu không chọn video

        if not self.video\_path:

            messagebox.showerror("Error", "Please select a video first!")

            return

        # Tạo thư mục lưu các frame

        output\_folder = "frames"

        if not os.path.exists(output\_folder):

            os.makedirs(output\_folder)

        # Mở video

        cap = cv2.VideoCapture(self.video\_path)

        if not cap.isOpened():

            messagebox.showerror("Error", "Failed to open video.")

            return

        # Trích xuất frame

        frame\_count = 0

        while True:

            ret, frame = cap.read()

            if not ret:

                break

            # Lưu từng frame dưới dạng ảnh

            frame\_filename = os.path.join(output\_folder, f"frame\_{frame\_count:04d}.jpg")

            cv2.imwrite(frame\_filename, frame)

            frame\_count += 1

        cap.release()

        # Hiển thị thông báo hoàn tất

        self.status\_label.config(text=f"Extracted {frame\_count} frames successfully.")

        messagebox.showinfo("Success", f"{frame\_count} frames extracted and saved to '{output\_folder}'.")

    def save\_image\_G213NTD(self):

        if self.current\_image:

            save\_image\_G213NTD(self.current\_image)

        else:

            messagebox.showwarning("Warning", "Chưa có ảnh để lưu!")

    def blur\_image\_G213NTD(self):

        if self.current\_image:

            self.current\_image = blur\_image\_G213NTD(self.current\_image)

            self.update\_image\_display()

        else:

            messagebox.showwarning("Warning", "Chưa có ảnh để làm mờ!")

    def sharpen\_image\_G213NTD(self):

        if self.current\_image:

            self.current\_image = sharpen\_image\_G213NTD(self.current\_image)

            self.update\_image\_display()

        else:

            messagebox.showwarning("Warning", "Chưa có ảnh để làm sắc nét!")

    def convert\_to\_bw\_G213NTD(self):

        if self.current\_image:

            self.current\_image = convert\_to\_bw\_G213NTD(self.current\_image)

            self.update\_image\_display()

        else:

            messagebox.showwarning("Warning", "Chưa có ảnh để chuyển sang đen trắng!")

    def crop\_image\_G213NTD(self):

        if self.current\_image:

            # Cắt ảnh (mặc định: cắt từ 50, 50 tới 250, 250)

            left = 50

            top = 50

            right = self.current\_image.width - 50

            bottom = self.current\_image.height - 50

            self.current\_image = crop\_image\_G213NTD(self.current\_image, left, top, right, bottom)

            self.update\_image\_display()

        else:

            messagebox.showwarning("Warning", "Chưa có ảnh để cắt!")

    def resize\_image\_G213NTD(self):

        if self.current\_image:

            # Resize ảnh (mặc định: 300x300)

            width = 300

            height = 300

            self.current\_image = resize\_image\_G213NTD(self.current\_image, width, height)

            self.update\_image\_display()

        else:

            messagebox.showwarning("Warning", "Chưa có ảnh để resize!")

    def update\_image\_display(self):

        if self.current\_image:

            # Chuyển đổi ảnh từ PIL Image sang định dạng mà tkinter có thể hiển thị

            img\_tk = ImageTk.PhotoImage(self.current\_image)

            self.image\_label.config(image=img\_tk)

            self.image\_label.image = img\_tk  # Lưu tham chiếu đến ảnh để tránh bị garbage collection

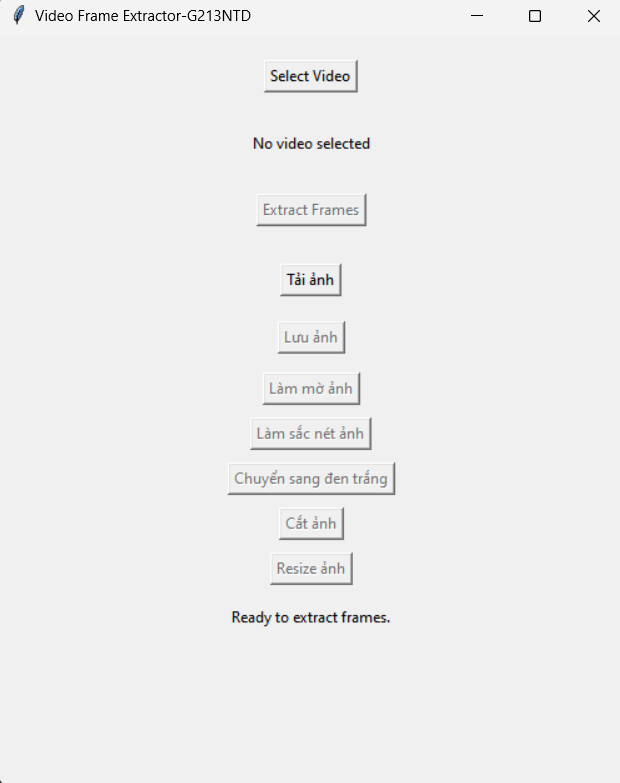
if \_\_name\_\_ == "\_\_main\_\_":

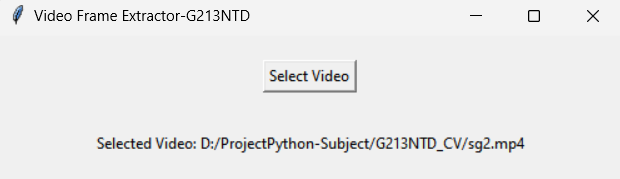
    root = tk.Tk()

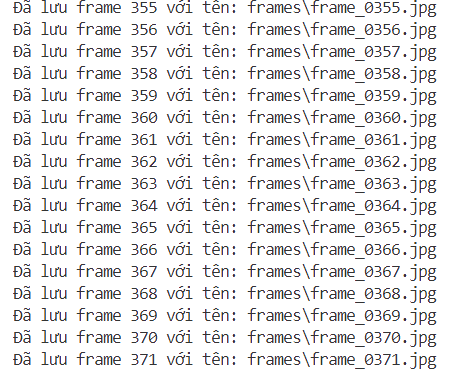
    app = VideoFrameExtractorApp\_G213NTD(root)

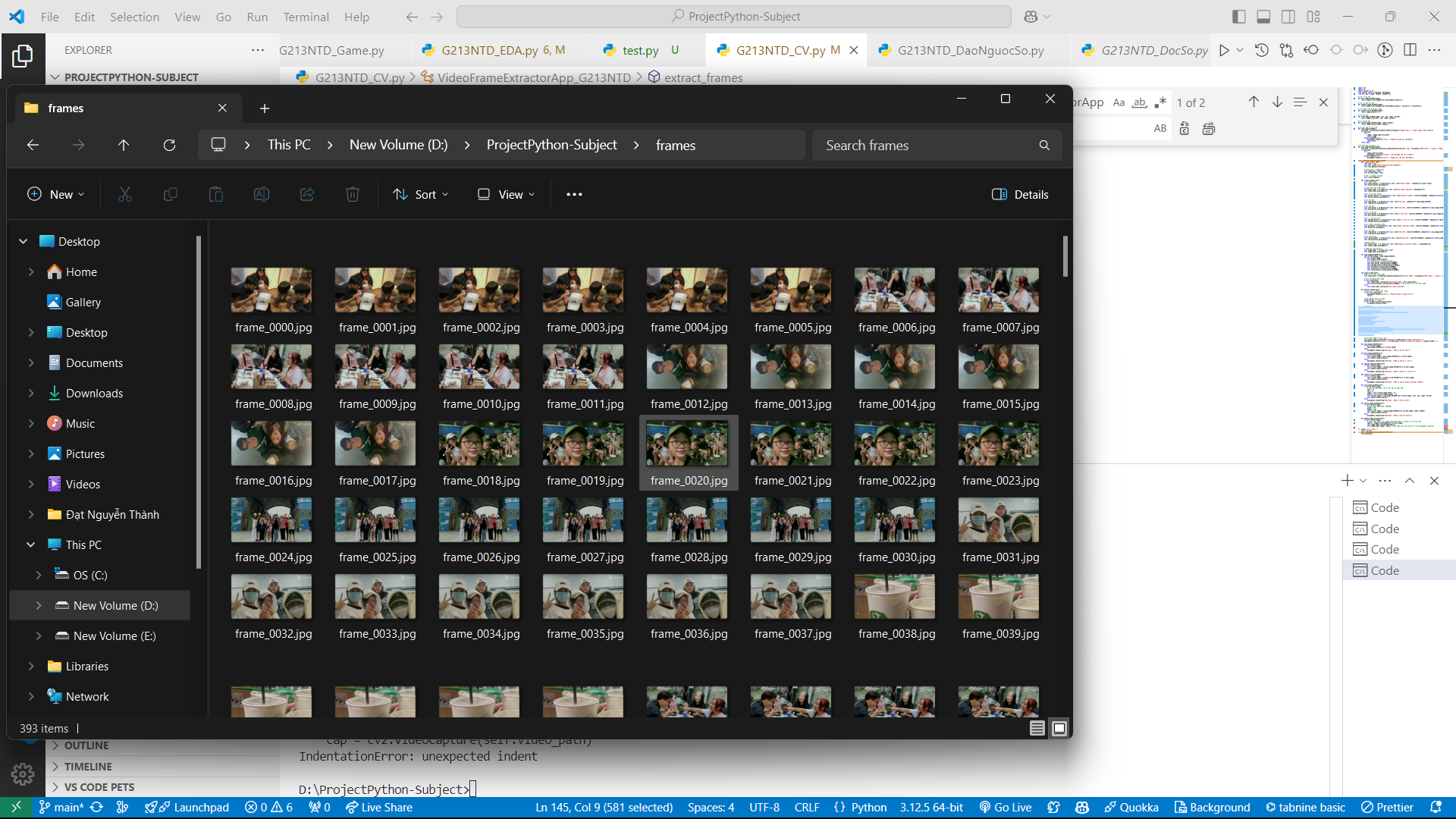
    root.mainloop()

## Giới thiệu sản phẩm kết quả thực nghiệm bài làm

D:\ProjectPythonSubject\G213NTD\_CV\sg2.mp4







Tổng kết về lập trình hỗ trợ CV nhận diện ………. từ Clip……

+ Câu lệnh chính:

# Mở video

        cap = cv2.VideoCapture(self.video\_path)

        if not cap.isOpened():

            messagebox.showerror("Error", "Failed to open video.")

            return

        # Trích xuất frame

        frame\_count = 0

        while True:

            ret, frame = cap.read()

            if not ret:

                break

            # Lưu từng frame dưới dạng ảnh

            frame\_filename = os.path.join(output\_folder, f"frame\_{frame\_count:04d}.jpg")

            cv2.imwrite(frame\_filename, frame)

            frame\_count += 1

        cap.release()

break

#while sẽ end trong 2 trường hợp: 1 khi NSD gõ q OR khi hết Video Clip hết

+ Ứng dụng:

1. **LẬP TRÌNH PYTHON GAME Car Racing**

<kết cấu nội dung bài Game vào đây>

## Giới thiệu chủ đề Game Car Racing

- Game đua xe thế hệ mới

## Phân tích chủ đề Game …….

## Các nền tảng kỹ thuật Lập trình Game

## Các thư viện liên quan

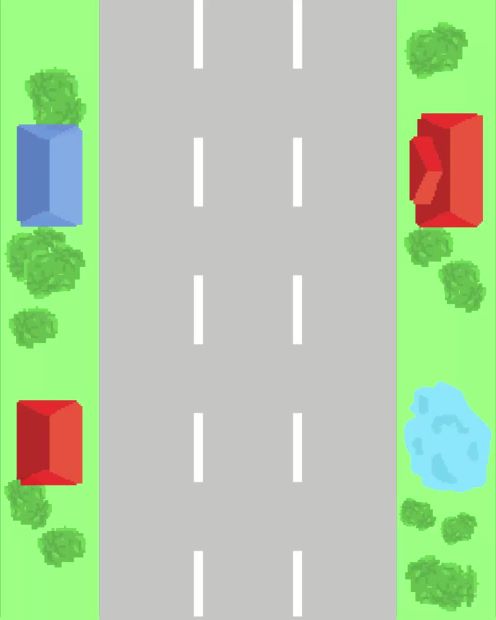
# -\*- coding: utf-8 -\*-

import pygame, sys, random

from pygame.locals import \*

## Giới thiệu Các mẫu media…….thực nghiệm (mẫu xe và đường đua)

ecarycar



## Mô tả giải thuật Game….

## Giao diện Game ….



## Mã lệnh lập trình Game……

# -\*- coding: utf-8 -\*-

import pygame, sys, random

from pygame.locals import \*

####################################

#PHẦN 1: ĐỊNH NGHĨA CÁC THAM SỐ ##

#####################################

###KÍCH THƯỚC KHUNG MÀN HÌNH GAME

WINDOWWIDTH\_G213NTD = 400

WINDOWHEIGHT\_G213NTD = 600

###KHỞI TẠO THƯ VIỆN ĐỂ DÙNG

pygame.init()

##TỐC ĐỘ KHUNG HÌNH CỦA VIDEO

FPS = 60 # Famres Per Second

fpsClock = pygame.time.Clock() #Lặp theo nhịp clock (tham số FPS)

####################################

#####PHẦN 2: NỀN GAME ##############

#####################################

#TỐC ĐỘ CUỘN NỀN

BGSPEED = 1.5 # tốc độ cuộn nền

BGIMG = pygame.image.load('./G213NTD\_GAME/img/road.jpg') # hình nền

# LAYER (SURFACE) NỀN

DISPLAYSURF = pygame.display.set\_mode((WINDOWWIDTH\_G213NTD, WINDOWHEIGHT\_G213NTD))

pygame.display.set\_caption('G213NTD\_CARRACING: Game ĐUA XE')

# 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, (int(self.x), int(self.y)))

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

    def update(self):

        self.y += self.speed

        if self.y > self.height:

            self.y -= self.height

####################################

#####PHẦN 3: XE TRONG GAME #########

"""

•   X\_MARGIN là lề hai bên trái và phải (xe không được vượt qua đó).

•   CARWIDTH và CARHEIGHT là kích thước của xe.

•   CARSPEED là tốc độ di chuyển (tiến, lùi, trái, phải) của xe.

•   CARIMG là ảnh chiếc xe.

"""

#####################################

#KÍCH THƯỚC XE

X\_MARGIN\_G213 = 80

CARWIDTH\_G213 = 40

CARHEIGHT\_G213 = 60

CARSPEED\_G213 = 3

CARIMG\_22110129 = pygame.image.load('./G213NTD\_GAME/img/ycar.png')

#LỚP XE TRONG GAME

class Car():

    def \_\_init\_\_(self):

        self.width = CARWIDTH\_G213

        self.height = CARHEIGHT\_G213

        self.x = (WINDOWWIDTH\_G213NTD-self.width)/2

        self.y = (WINDOWHEIGHT\_G213NTD-self.height)/2

        self.speed = CARSPEED\_G213

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

        self.surface.fill((255, 255, 255))

    def draw(self):

        DISPLAYSURF.blit(CARIMG\_22110129, (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\_G213:

            self.x = X\_MARGIN\_G213

        if self.x + self.width > WINDOWWIDTH\_G213NTD - X\_MARGIN\_G213:

            self.x = WINDOWWIDTH\_G213NTD - X\_MARGIN\_G213 - self.width

        if self.y < 0:

            self.y = 0

        if self.y + self.height > WINDOWHEIGHT\_G213NTD :

            self.y = WINDOWHEIGHT\_G213NTD - self.height

####################################

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

"""

•   LANEWIDTH là độ rộng của 1 làn xe (đường có 4 làn).

•   DISTANCE là khoảng cách giữa các xe theo chiều dọc.

•   OBSTACLESSPEED là tốc độ ban đầu của những chiếc xe.

•   CHANGESPEED dùng để tăng tốc độ của những chiếc xe theo thời gian.

•   OBSTACLESIMG là ảnh chiếc xe.

"""

#####################################

LANEWIDTH\_G213 = 60

DISTANCE\_G213 = 200

OBSTACLESSPEED\_G213 = 2

CHANGESPEED\_G213 = 0.001

OBSTACLESIMG\_G213 = pygame.image.load('./G213NTD\_GAME/img/ecar.png')

class Obstacles():

    def \_\_init\_\_(self):

        self.width = CARWIDTH\_G213

        self.height = CARHEIGHT\_G213

        self.distance = DISTANCE\_G213

        self.speed = OBSTACLESSPEED\_G213

        self.changeSpeed = CHANGESPEED\_G213

        self.ls = []

        for i in range(5):

            y = -CARHEIGHT\_G213-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\_G213 + self.ls[i][0]\*LANEWIDTH\_G213 + (LANEWIDTH\_G213-self.width)/2)

            y = int(self.ls[i][1])

            DISPLAYSURF.blit(OBSTACLESIMG\_G213, (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\_G213NTD:

            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

    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\_G213NTD(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\_G213(car, obstacles):

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

    for i in range(5):

        x = int(X\_MARGIN\_G213 + obstacles.ls[i][0]\*LANEWIDTH\_G213 + (LANEWIDTH\_G213-obstacles.width)/2)

        y = int(obstacles.ls[i][1])

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

        if rectCollision\_G213NTD(carRect, obstaclesRect) == True:

            return True

    return False

####################################

#PHẦN 7: CÁC THỦ TỤC CHƠI GAME ##

"""

•   gameStart() là phần chuẩn bị khi vừa mở game lên.

•   gamePlay() là phần chơi chính.

•   gameOver() là phần xuất hiện khi thua 1 màn chơi.

"""

#####################################

def gameOver\_G213NTD(bg, car, obstacles, score):

    font\_G213NTD = pygame.font.SysFont('consolas', 60)

    headingSuface\_G213NTD = font\_G213NTD.render('GAMEOVER', True, (255, 0, 0))

    headingSize\_G213NTD = headingSuface\_G213NTD.get\_size()

    font\_G213NTD = pygame.font.SysFont('consolas', 20)

    commentSuface\_G213NTD = font\_G213NTD.render('Press "space" to replay', True, (0, 0, 0))

    commentSize\_G213NTD = commentSuface\_G213NTD.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()

        car.draw()

        obstacles.draw()

        score.draw()

        DISPLAYSURF.blit(headingSuface\_G213NTD, (int((WINDOWWIDTH\_G213NTD - headingSize\_G213NTD[0])/2), 100))

        DISPLAYSURF.blit(commentSuface\_G213NTD, (int((WINDOWWIDTH\_G213NTD - commentSize\_G213NTD[0])/2), 400))

        pygame.display.update()

        fpsClock.tick(FPS)

def gameStart\_G213NTD(bg):

    bg.\_\_init\_\_()

    FONT\_G213 = pygame.font.SysFont('consolas', 60)

    headingSuface19g1 = FONT\_G213.render('RACING', True, (255, 0, 0))

    headingSizeG1Linh = headingSuface19g1.get\_size()

    FONT\_G213 = pygame.font.SysFont('consolas', 20)

    commentSufaceLINHG = FONT\_G213.render('Press "space" to play', True, (0, 0, 0))

    commentSize\_G213NTDg1 = commentSufaceLINHG.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(headingSuface19g1, (int((WINDOWWIDTH\_G213NTD - headingSizeG1Linh[0])/2), 100))

        DISPLAYSURF.blit(commentSufaceLINHG, (int((WINDOWWIDTH\_G213NTD - commentSize\_G213NTDg1[0])/2), 400))

        pygame.display.update()

        fpsClock.tick(FPS)

def gamePlay\_G213NTD(bg, car, obstacles, score):

    car.\_\_init\_\_()

    obstacles.\_\_init\_\_()

    bg.\_\_init\_\_()

    score.\_\_init\_\_()

    moveLeft\_G213 = False

    moveRight\_G213 = False

    moveUp\_G213 = False

    moveDown\_G213 = 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\_G213 = True

                if event.key == K\_RIGHT:

                    moveRight\_G213 = True

                if event.key == K\_UP:

                    moveUp\_G213 = True

                if event.key == K\_DOWN:

                    moveDown\_G213 = True

            if event.type == KEYUP:

                if event.key == K\_LEFT:

                    moveLeft\_G213 = False

                if event.key == K\_RIGHT:

                    moveRight\_G213 = False

                if event.key == K\_UP:

                    moveUp\_G213 = False

                if event.key == K\_DOWN:

                    moveDown\_G213 = False

        if isGameover\_G213(car, obstacles):

            return

        bg.draw()

        bg.update()

        car.draw()

        car.update(moveLeft\_G213, moveRight\_G213, moveUp\_G213, moveDown\_G213)

        obstacles.draw()

        obstacles.update()

        score.draw()

        score.update()

        pygame.display.update()

        fpsClock.tick(FPS)

####################################

#PHẦN 8: HÀM MAIN ##

#####################################

def main():

    bg = Background()

    car = Car()

    obstacles = Obstacles()

    score = Score()

    gameStart\_G213NTD(bg)

    while True:

        gamePlay\_G213NTD(bg, car, obstacles, score)

        gameOver\_G213NTD(bg, car, obstacles, score)

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

    main()

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

## Những kết quả đạt được

ĐỒ ÁN HỌC PHẦN đã thực hiện việc Lập trình Python …………..; bao gồm:

+ Giao diện đồ họa: GUI

+ Xử lý giọng nói trợ lý ảo: Voice Assistant

+ Phân tích khám phá trên tập dữ liệu thực nghiệm: ……………………., có Plot

+ CV ……………

+ Game………..

## Hạn chế & hướng khắc phục các hạn chế

Vì thời gian có hạn nên chưa

+ liên kết voice assistant để ra lệnh phân tích khám phá bằng lời nói.

+ ….

Đồ án sẽ tiếp tục thực hiện

+ liên kết voice assistant để ra lệnh phân tích khám phá bằng lời nói

+ …

## Hướng mở rộng ĐỒ ÁN HỌC PHẦN

Sẽ tiếp tục mở rộng sản phẩm đề tài với các ứng dụng thực tiễn.

# 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. http://… : nguồn dataFrame được sử dụng trong đề tài này
4. ….

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