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

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN HỌC PHẦN**

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

MÃ SỐ LỚP HP: **IPPA233277\_03**

Tên chủ đề :**LẬP TRÌNH PYTHON PHÂN TÍCH THĂM DÒ CHẤT LƯỢNG KHÔNG KHÍ Ở CHICAGO TỪ NĂM 1987 ĐẾN 2005.**

**Họ tên sinh viên: Đào Quang Duy [09]**

**Mã số sinh viên: 21110398**

**Lớp: 211103D [CQ]**

**Ngày nộp: 17/05/2023**

**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\_03**

Tên đề tài: **LẬP TRÌNH PYTHON PHÂN TÍCH THĂM DÒ CHẤT LƯỢNG KHÔNG KHÍ Ở CHICAGO TỪ NĂM 1987 ĐẾN 2005**

**Giảng viên giảng dạy: VÕ XUÂN THỂ**

**Họ tên sinh viên: Đào Quang Duy [09]**

**Mã số sinh viên: 21110398**

**Lớp: 211103D [CQ]**

**TÊN CÁC FILE SẢN PHẨM ĐỀ TÀI:**

**Tên sản phẩm đề tài: G309DQDuy\_DAHP.PyPro\_Chicago(1987-2005).rar**

**Tập tin mã nguồn: G309DQDuy\_Chicago(1987-2005).ipynb**

**Tập dữ liệu thực nghiệm: G309DQDuy\_Chicago(1987-2005).csv**

**Giới thiệu thư mục bài làm: G309DQDuy\_DAHP.Intro.docx**

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

# GUI: tkinter

# EDA: pandas, numpy, scipy, sklearn

# Game: Pygame

#Image & frames: cv2, ImageTk, Image

**NHẬN XÉT CỦA GIẢNG VIÊN**

**………………………………………………………..**

**………………………………………………………..**

**Ngày……./……../202…..**

**Giảng viên: Ký tên**

**TP.HCM, ngày 10 tháng 5 năm 2023**

**LỜI CẢM ƠN**

Kính thưa thầy, giáo viên dạy môn lập trình Python, thầy Võ Xuân Thể.

Em xin gửi lời cảm ơn chân thành nhất đến thầy vì những kiến thức vô giá mà thầy đã truyền đạt cho em trong quá trình học môn lập trình Python. Nhờ những kiến thức và kỹ năng được thầy truyền đạt, Em đã có thể ứng dụng thành công trong nhiều dự án của mình.

Thầy luôn dành thời gian, tâm huyết và nhiệt tình để giảng dạy và hỗ trợ em trong quá trình học tập. Những bài giảng, tài liệu và bài tập của thầy luôn rất sát với thực tiễn và giúp em hiểu bài học một cách dễ dàng và nhanh chóng.

Em rất may mắn và tự hào khi được học tập dưới sự hướng dẫn của thầy. Một lần nữa, em xin gửi lời cảm ơn sâu sắc đến thầy và hi vọng được học hỏi và phát triển thêm nhiều từ thầy trong tương lai.

Trân trọng,

Đào Quang Duy

# DANH MỤC CÁC TỪ VIẾT TẮT

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

**DANH MỤC CÁC THUẬT NGỮ**

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

**MỤC LỤC**

[**LỜI CẢM ƠN** 3](#_Toc135338831)

[DANH MỤC CÁC TỪ VIẾT TẮT 4](#_Toc135338832)

[**DANH MỤC CÁC THUẬT NGỮ** 4](#_Toc135338833)

[Chương 1: GIỚI THIỆU VỀ ĐỒ ÁN HỌC PHẦN 9](#_Toc135338834)

[1.1. Tổng quan về ĐỒ ÁN HỌC PHẦN 9](#_Toc135338835)

[1.2. Nội dung chuyên môn chính của ĐỒ ÁN HỌC PHẦN 9](#_Toc135338836)

[1.3. Công cụ và nền tảng kỹ thuật thực hiện ĐỒ ÁN HỌC PHẦN 9](#_Toc135338837)

[Bố cục của báo cáo 10](#_Toc135338838)

[Chương 2: CƠ SỞ LÝ THUYẾT CỦA HỌC PHẦN 11](#_Toc135338839)

[2.1. GIỚI THIỆU VỀ LẬP TRÌNH PYTHON 11](#_Toc135338840)

[2.1.1. Giới thiệu chung 11](#_Toc135338841)

[2.1.2. Cài đặt “bộ thảo chương” (soạn thảo chương trình IDE) và “bộ dịch” 11](#_Toc135338842)

[2.1.3. Giới thiệu các bộ tương tác lập trình 11](#_Toc135338843)

[2.1.4. Cấu hình mô trường tương tác lập trình 11](#_Toc135338844)

[2.2. TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH PYTHON 11](#_Toc135338845)

[2.2.1. Danh hiệu (Identifier) 11](#_Toc135338846)

[2.2.2. Về viết mã lệnh lập trình (Code) 11](#_Toc135338847)

[2.2.3. Về chú giải (Comment) 11](#_Toc135338848)

[2.2.4. Các phép toán 11](#_Toc135338849)

[2.2.5. Chuỗi 11](#_Toc135338850)

[2.2.6. Lệnh 11](#_Toc135338851)

[2.2.7. Định nghĩa hàm 11](#_Toc135338852)

[2.2.8. Danh sách 11](#_Toc135338853)

[2.2.9. Bộ dữ liệu: Tuple 11](#_Toc135338854)

[2.2.10. Tập hợp: Set 11](#_Toc135338855)

[2.2.11. Từ điển: Dictionary 11](#_Toc135338856)

[2.2.12. Module 11](#_Toc135338857)

[2.2.13. Package 11](#_Toc135338858)

[2.2.14. Lớp 11](#_Toc135338859)

[2.2.15. Xử lý ngoại lệ 11](#_Toc135338860)

[2.3. GIỚI THIỆU CÁC THƯ VIỆN VÀ CÔNG CỤ CĂN BẢN 11](#_Toc135338861)

[2.3.1. Nạp thư viện 11](#_Toc135338862)

[2.3.2. Pandas 11](#_Toc135338863)

[2.3.3. Tensorflow 11](#_Toc135338864)

[2.3.4. NumPy 11](#_Toc135338865)

[2.3.5. SCIPY 11](#_Toc135338866)

[2.3.6. Theano 11](#_Toc135338867)

[2.3.7. Matplotlib 11](#_Toc135338868)

[2.3.8. scikit-learn = sklearn 11](#_Toc135338869)

[2.3.9. Keras 12](#_Toc135338870)

[2.3.10. NLTK 12](#_Toc135338871)

[2.3.11. PyTorch 12](#_Toc135338872)

[2.3.12. LightGBM 12](#_Toc135338873)

[2.3.13. Eli5 12](#_Toc135338874)

[2.4. LẬP TRÌNH PYTHON XỬ LÝ GIỌNG NÓI (Trợ lý ảo: Voice Assistant) 12](#_Toc135338875)

[2.4.1. Giới thiệu chủ đề 12](#_Toc135338876)

[2.4.2. Tài liệu và nguồn thực nghiệm 12](#_Toc135338877)

[2.4.3. Giới thiệu Các thư viện Python liên quan 12](#_Toc135338878)

[2.4.4. Giới thiệu bài toán 12](#_Toc135338879)

[2.4.5. Giới thiệu code tham khảo 12](#_Toc135338880)

[2.5. KỸ THUẬT LẬP TRÌNH ĐỆ QUY: THÁP HÀ NỘI (HANOI TOWER) 12](#_Toc135338881)

[2.6. LẬP TRÌNH PHÂN TÍCH KHÁM PHÁ (EDA) [Thăm dò] 12](#_Toc135338882)

[Chương 3: PHÂN TÍCH VÀ XÁC ĐINH CÁC CƠ SỞ KỸ THUẬT 13](#_Toc135338883)

[3.1. GIỚI THIỆU CHUNG VỀ CHỦ ĐỀ 13](#_Toc135338884)

[3.1.1. Mô tả yêu câu của chủ đề 13](#_Toc135338885)

[3.1.2. Phân tích yều cầu của chủ đề 13](#_Toc135338886)

[3.2. CÁC NỀN TẢNG KỸ THUẬT 13](#_Toc135338887)

[3.2.1. Hệ thống thư viện sử dụng 13](#_Toc135338888)

[3.2.2. Tập dữ liệu thực nghiệm 13](#_Toc135338889)

[3.2.3. Mô tả giải thuật 14](#_Toc135338890)

[3.3. LẬP TRÌNH XỬ GIỌNG NÓI (TRỢ LÝ ẢO: VOICE ASSISTANT) 14](#_Toc135338891)

[3.3.1. Giới thiệu chung về chủ đề 14](#_Toc135338892)

[3.3.1.1. Mô tả yêu cầu của chủ đề 14](#_Toc135338893)

[3.3.1.2. Phân tích yều cầu của chủ đề 15](#_Toc135338894)

[3.3.2. Các nền tảng kỹ thuật 15](#_Toc135338895)

[3.3.2.1. Nền tảng kỹ thuật cơ bản 15](#_Toc135338896)

[3.3.2.2. Hệ thống thư việc sử dụng 15](#_Toc135338897)

[3.3.3. Thiết kế giao diện 15](#_Toc135338898)

[3.3.4. Mã lệnh lập trình (python) 15](#_Toc135338899)

[Chương 4: LẬP TRÌNH PYTHON PHÂN TÍCH THĂM DÒ CHẤT LƯỢNG KHÔNG KHÍ Ở CHICAGO TỪ NĂM 1987 ĐẾN 2005 17](#_Toc135338900)

[4.1. Giới thiệu chung. 17](#_Toc135338901)

[4.2. Giao diện chính. 18](#_Toc135338902)

[4.2.1. Form. 18](#_Toc135338903)

[4.2.1. Source code. 18](#_Toc135338904)

[4.3. EDA. 20](#_Toc135338905)

[4.3.1. Form. 20](#_Toc135338906)

[4.3.2. Source code. 20](#_Toc135338907)

[4.4. Plot. 26](#_Toc135338908)

[4.4.1. Form. 26](#_Toc135338909)

[4.4.2. Source code. 26](#_Toc135338910)

[4.5. Image processing and Frames. 29](#_Toc135338911)

[4.5.1. Form. 29](#_Toc135338912)

[4.5.2. Source code. 29](#_Toc135338913)

[4.6. Game. 33](#_Toc135338914)

[4.6.1. Form. 33](#_Toc135338915)

[4.6.2. Source code. 33](#_Toc135338916)

[Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỒ ÁN HỌC PHẦN 41](#_Toc135338917)

[5.1. Kết luận 41](#_Toc135338918)

[5.1.1. Những kết quả đạt được 41](#_Toc135338919)

[5.1.2. Hạn chế 41](#_Toc135338920)

[5.2. Hướng phát triển 41](#_Toc135338921)

[5.2.1. Hướng khắc phục các hạn chế 41](#_Toc135338922)

[5.2.2. Hướng mở rộng ĐỒ ÁN HỌC PHẦN 41](#_Toc135338923)

[DANH MỤC TÀI LIỆU THAM KHẢO 42](#_Toc135338924)

# GIỚI THIỆU VỀ ĐỒ ÁN HỌC PHẦN

## Tổng quan về ĐỒ ÁN HỌC PHẦN

Đồ án Học phần Lập trình Python phân tích thăm dò chất lượng không khí ở Chicago từ năm 1987 – 2005 vận dụng kiến thức và kỹ năng về Lập trình Python để xây dựng giao diện đồ hoạ (GUI) cho phép: tương tác với người dùng bằng giọng nói (Trợ lý ảo: Voice Assistant) và phân tích thăm dò (EDA: Expolore Data Analysing) đối với tập dữ liệu thực nghiệm về chỉ số chất lượng không khí không khí lịch sử (AQI).

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

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

+ Trợ lý ảo.

+ Phân tích dữ liệu khám phá || thăm dò.

+ Giao diện đồ họa dạng WinForm.

+ Image\_Proccesing & Frames from Video.

+ Vẽ đồ thị \_ Biểu đồ (Plot).

+ Game.

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

+ Ngôn ngữ lập trình **Python**

+ Công cụ lập trình và biên dịch: **jupyter (visual studio code)**

+ Các thư viện chính:

# B1: NẠP THƯ VIỆN

import numpy as DQDuy\_np

import pandas as DQDuy\_pd

from scipy import stats as DQDuy\_stats

from sklearn import preprocessing as DQDuy\_pre

from sklearn.feature\_selection import SelectKBest, chi2

from tkinter import \*

import tkinter as tk09DQDuy

import os as os09DQDuy

import speech\_recognition as sr09DQDuy

from gtts import gTTS as gt09DQDuy

import playsound as ps09DQDuy

import tkinter.messagebox as DQDuyTB

from tkinter import filedialog as fd

import pygame, sys, random

from pygame.locals import \*

## Bố cục của báo cáo

Báo cáo gồm những nội dung như sau:

Chương 1: Giới thiệu đồ án của Học phần

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

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

Chương 4: Lập trình xây dựng sản phầm đề lài: PHÂN TÍCH THĂM DÒ CHẤT LƯỢNG KHÔNG KHÍ Ở CHICAGO TỪ NĂN 1987 ĐẾN NĂM 2005.

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

# CƠ SỞ LÝ THUYẾT CỦA HỌC PHẦN

## GIỚI THIỆU VỀ LẬP TRÌNH PYTHON

### Giới thiệu chung

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

### Giới thiệu các bộ tương tác lập trình

### Cấu hình mô trường tương tác lập trình

## TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH PYTHON

### Danh hiệu (Identifier)

### Về viết mã lệnh lập trình (Code)

### Về chú giải (Comment)

### Các phép toán

### Chuỗi

### Lệnh

### Định nghĩa hàm

### Danh sách

### Bộ dữ liệu: Tuple

### Tập hợp: Set

### Từ điển: Dictionary

### Module

### Package

### Lớp

### Xử lý ngoại lệ

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

### Nạp thư viện

### Pandas

### Tensorflow

### NumPy

### SCIPY

### Theano

### Matplotlib

### scikit-learn = sklearn

### Keras

### NLTK

### PyTorch

### LightGBM

### Eli5

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

### Giới thiệu chủ đề

### Tài liệu và nguồn thực nghiệm

### Giới thiệu Các thư viện Python liên quan

### Giới thiệu bài toán

### Giới thiệu code tham khảo

**[1] Nghe tiếng Việt => Text**

**[2] Text => đọc tiếng Việt**

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

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

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

## GIỚI THIỆU CHUNG VỀ CHỦ ĐỀ

### Mô tả yêu câu của chủ đề

Vận dụng kỹ thuật lập trình EDA kết hợp với GUI và Voice Assistant cho phép dùng tập dữ liệu thực nghiệm về các thành phần trong không khí ở Chicago để thống kê đưa ra các kết luận về không khí để đưa ra các vấn đề để giải quyết nó.

Dữ liệu thực nghiệm gồm nồng độ vật chất hạt mịn, nhiệt độ mà nồng độ các khí liên quan trong quá khứ của chất lượng không khí:

G309DQDuy\_Chicago(1987-2005).csv: dạng file Excel .csv

. Số records: 6039

. Số thuộc tính input: 9 để đoán thuộc tính thứ 6 target variable:

= “pm25tmean2”[cao/thấp]

### 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 xóa bỏ hàng có nhiều thuộc tính null hoặc thay thế nó bằng các giá trị chỉ định phù hợp.

[2] Xử lý dữ liệu quá khác biệt = “ngoại lệ” (isolated: quá cao hay quá thấp) là ta sẽ lấy giá trị trung bình hoặc giá trị có tần số xuất hiện nhiều nhất của cột để thay vào.

## CÁC NỀN TẢNG KỸ THUẬT

### Hệ thống thư viện sử dụng

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

import numpy as DQDuy\_np

import pandas as DQDuy\_pd

### Tập dữ liệu thực nghiệm

**Bước 2: Tải tập dữ liệu**

DQDuy\_fro = DQDuy\_pd.read\_csv('./ G309DQDuy\_Chicago(1987-2005).csv')

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

print(df[0:5])

Chạy python G3B5EX3DA\_09DQDuy.py sẽ hiển thị kết quả:

Độ lớn của bảng [frame] dữ liệu không khí: (6940, 9)

stt city tmpd dptp date pm25tmean2 pm10tmean2 o3tmean2 \

0 0 chic 31.5 31.500 1/1/1987 NaN 34.000000 4.250000

1 1 chic 33.0 29.875 1/2/1987 NaN NaN 3.304348

2 2 chic 33.0 27.375 1/3/1987 NaN 34.166667 3.333333

3 3 chic 29.0 28.625 1/4/1987 NaN 47.000000 4.375000

4 4 chic 32.0 28.875 1/5/1987 NaN NaN 4.750000

no2tmean2

0 19.988095

1 23.190994

2 23.815476

3 30.434524

4 30.333333

### Mô tả giải thuật

**Bước 3: Xử lý dữ liệu**

print(DQDuy\_fro.count().sort\_values())

**Kết quả:**

pm25tmean2 2493

pm10tmean2 6698

dptp 6938

tmpd 6939

stt 6940

city 6940

date 6940

o3tmean2 6940

no2tmean2 6940

dtype: int64

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

## Giới thiệu chung về chủ đề

### Mô tả yêu cầu của chủ đề

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

+ Gồm 2 phần (Part) cơ bản sau

1. Nghe tiếng Việt => Text: Nhận vào âm thanh là 1 số và in ra số đó

2. Trả lời bằng tiếng Việt : Text => Nói tiếng Việt: Sẽ phát ra âm thanh câu nói

### Phân tích yều cầu của chủ đề

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

+ Các hệ thống này thường có 3 phần :

Part1: Speech to text

Part2: Các kỹ thuật AI

Part2: Voice recognition

## Các nền tảng kỹ thuật

### Nền tảng kỹ thuật cơ bản

### Hệ thống thư việc sử dụng

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

**import** speech\_recognition **as** sr

**from** gtts **import** gTTS

**import** os

**import** time

**import** playsound

## Thiết kế giao diện

+ Xử lý trong cửa sổ Console Output của jupyter

## Mã lệnh lập trình (python)

**FULL CODE**

import speech\_recognition as Duy

from gtts import gTTS

import playsound

Duy\_bien = Duy.Recognizer()

lg = ''

#Hàm chỉ cho nhập vào số

def intInput(str):

    try:

        return int(input(str))

    except:

        print('Please enter correct!!!')

        return intInput(str)

#hàm hiện ra các lựa chọn

def Menu(str ,list):

    print(str)

    #Hiện ra các lựa chọn

    for i in list:

        print(list.index(i)+1 , ':' + i)

    #Nhập vào sự lựa chọn nếu sai nhập lại

    while True:

        sl = intInput('Your selection(int): ')

        if sl > 0 and sl <= len(list):

            return sl

        print('Please enter correct!!!')

#Hàm chọn Yes/no

def yesNo():

    while True:

        a = Menu('Press key 1 to record: ', ['Yes', 'No'])

        if a == 1:

            return

#Hàm chọn ngôn ngữ

def selete\_language\_cuaDuy():

    a = Menu("which language do you want?", ['Vietnamese', 'English'])

    match a:

        case 1:

            return 'vi'

        case 2:

            return 'en'

with Duy.Microphone() as duySource:

    print("think before Duy say!")

    Duy\_bien.adjust\_for\_ambient\_noise(duySource, duration=1)

    lg = selete\_language\_cuaDuy()

    yesNo()

    print("Please say!")

    duy\_audio = Duy\_bien.record(duySource, duration = 5)

    print("Result ..................")

    try:

        text = Duy\_bien.recognize\_google(duy\_audio,language=lg)

    except:

        text = "I don't hear well...!"

    print("You said: ",format(text))

def DaoQuangDuy(t):

    Ten = gTTS(text=t, lang = lg)

    print('What do you want to name the file?')

    filename = f'{input()}.mp3'

    Ten.save(filename)

    playsound.playsound(filename)

    return

DaoQuangDuy("số thứ tự 09, họ tên: Đào Quang Duy là sinh viên viên của TRƯỜNG ĐẠI HỌC spkt")

1. **LẬP TRÌNH PYTHON PHÂN TÍCH THĂM DÒ CHẤT LƯỢNG KHÔNG KHÍ Ở CHICAGO TỪ NĂM 1987 ĐẾN 2005**

## Giới thiệu chung.

Đồ án gồm source code và các form sau đây:

* EDA: Lập trình python phân tích chất lượng không khí ở chicago từ năm 1987 đến 2005(Xử lý null, Rời rạc, Chuẩn hóa số liệu, Sử dụng zcore để loại cỏ các giá trị cô lập, sử dụng chi2, f\_classif để chọn lọc các thuộc tính)
* Plot: Vẽ đồ thị để trực quan hóa số liệu(scatter, scatter3d, pie, hist, bar, barh)
* Image processing and Frames: Xử lý ảnh và video.(xoay, resize và hiện ảnh xám, cắt frame)
* Game: Lập trình game trong python.(Cho người dùng thay đổi các thuộc tính game như FPS, Cuộn nền, Bg, Điều khiển, ….)

## Giao diện chính.

### Form.

A screenshot of a computer

Description automatically generated with medium confidence

### Source code.

import numpy as DQDuy\_np

import pandas as DQDuy\_pd

from scipy import stats as DQDuy\_stats

from sklearn import preprocessing as DQDuy\_pre

from sklearn.feature\_selection import SelectKBest, chi2, f\_classif

from tkinter import \*

import tkinter as tk09DQDuy

from tkinter import ttk as ttk09DQDuy

import os as os09DQDuy

import speech\_recognition as sr09DQDuy

from gtts import gTTS as gt09DQDuy

import playsound as ps09DQDuy

import tkinter.messagebox as DQDuyTB

from tkinter import filedialog as fd

import pygame, sys, random

from pygame.locals import \*

import cv2 as cv209

from PIL import Image as Img09

import matplotlib.pyplot as plt09

from PIL import ImageTk as ImgTk09

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

from mpl\_toolkits.mplot3d import Axes3D

mainform = tk09DQDuy.Tk()

mainform.title('09 Đào Quang Duy \_ Lớp 03 \_ HUCMUTE: ĐỒ ÁN HP: LẬP TRÌNH PYTHON')

mainform.geometry('650x400')

mainform.resizable(False, False)

btnEDA09 = Button(mainform, text="EDA", width= 25,command=EDA)

btnImage09 = Button(mainform, text="Image processing & Frames",width= 25 ,command = IMAGE09)

btnGame09 = Button(mainform, text="Game",width= 25, command=GameForm09)

lblMain = Label(mainform, text = "Đồ án cuối kì môn lập trình Python", background= 'yellow', width=42, height= 5, font= ("Arial", 20), anchor= CENTER)

lblMain.place(x = 0, y = 0)

btnEDA09.place(x = 230, y = 180)

btnGame09.place(x = 230, y = 210)

btnImage09.place(x = 230, y = 240)

mainform.mainloop()

## EDA.

### Form.

A screenshot of a computer

Description automatically generated

### Source code.

def EDA():

    Duyapp = tk09DQDuy.Tk()

    Duyapp.geometry('640x510')

    Duyapp.resizable(False, False)

    Duyapp.configure(background='orange')

    Duyapp.title('09 Đào Quang Duy \_ Lớp 03 \_ HUCMUTE: ĐỒ ÁN HP: LẬP TRÌNH PYTHON')

    Duyapp.DQDuy09Record = sr09DQDuy.Recognizer()

    Duyapp.duy09Lg = 'vi'

    Duyapp.start = 0

    Duyapp.finish = 100

    global slDta09

    slDta09 = DQDuy\_pd.DataFrame()

    def DQDuythoat():

        DQDuymes = DQDuyTB.askquestion('Thoát', 'Bạn có muốn thoát không')

        if DQDuymes == 'yes':

            Duyapp.destroy()

    def Record(t):

        with sr09DQDuy.Microphone() as DQDDuySource:

            LbHienthi09.configure(text = t)

            Duyapp.DQDuy09Record.adjust\_for\_ambient\_noise(DQDDuySource, duration=1)

            DQDuymes = DQDuyTB.askquestion('Ghi âm', 'Bạn có muốn ghi âm ngay không')

            if DQDuymes != 'yes':

                LbHienthi09.configure(text = "Xin chao")

                return

            LbHienthi09.configure(text= "Please say!Bạn có 5 dây để nói")

            duy\_audio = Duyapp.DQDuy09Record.record(DQDDuySource, duration = 5)

            try:

                Duy09text = Duyapp.DQDuy09Record.recognize\_google(duy\_audio,language=Duyapp.duy09Lg)

            except:

                Duy09text = "I don't hear well...!"

            LbHienthi09.configure(text = format(Duy09text))

            return format(Duy09text)

    def DQDuyghiam():

        a = ''

        a = Record('1: Mở file\n2: Dropna\n3: Zcore')

        if a == '':

            if a == '1' or a.lower() == 'một':

                InPutFile09()

                LbHienthi09.configure(text= 'Ok')

            elif a == '2' or a.lower() == 'một':

                combofilna09.set("Drop")

                fillNa()

            elif a == '3' or a.lower() == 'ba':

                zcore()

    def Delete():

        try:

            file.drop(columns= [Slcolumn],axis=1, inplace= True)

            MainTable(file)

        except Exception:

            LbHienthi09.configure(text= 'không tồn tại cột này')

    def zcore():

        global zcoreMainTable

        zcoreMainTable = z(file)

        MainTable(zcoreMainTable)

    def fillNa():

        try:

            if combofilna09.get() == "Drop":

                file.dropna(axis=0, inplace=True)

            for i in file:

                if combofilna09.get() == "Mode":

                    file[i].fillna(file[i].mode(), inplace= True)

                elif combofilna09.get() == "Median":

                    file[i].fillna(file[i].median(), inplace= True)

            MainTable(file)

        except:

            LbHienthi09.configure(text= 'Không thay thế giá trị được')

    def InPutFile09():

        global file

        file = DQDuy\_pd.read\_csv(fd.askopenfilename())

        MainTable(file)

    def Scale09():

        global scaledMainTable

        scaledMainTable = scale(file)

        MainTable(scaledMainTable)

    def Buoc8():

        NameA = slDta09.columns[0]

        X = scaledMainTable.loc[:,scaledMainTable.columns!= NameA]

        y = scaledMainTable[[NameA]]

        selector = SelectKBest(chi2 if int(varskit.get()) == 1 else f\_classif, k=int(spinbox09.get()))

        selector.fit(X, y)

        X\_new = selector.transform(X)

        print(X\_new)

        LbHienthi09.configure(text = str(X.columns[selector.get\_support(indices=True)]))

    def TuychonDQDuy():

        mess = DQDuyTB.askquestion("Tùy chọn", 'Yes: Bắt đầu nhập file\n No: Hủy')

        if mess == 'yes':

            InPutFile09()

    def Choosechart():

        plot09()

    def slZcore(self):

        DQDuy0909 = int(comboZcore09.get())

        MainTable(file[(zcoreMainTable < DQDuy0909).all(axis=1)])

    def ShowPopupMenu(event):

        global Slcolumn

        column\_id = tree09.identify\_column(event.x)

        Slcolumn = tree09.heading(column\_id)['text']

        popMenu = tk09DQDuy.Menu(tree09, tearoff = tk09DQDuy.FALSE)

        popMenu.add\_command(label = "Delete", command = Delete)

        popMenu.add\_command(label = "Select", command = AddToSelectedTable)

        popMenu.tk\_popup(event.x\_root, event.y\_root)#phải thiết lập x\_root, y\_root để showpopup

    def AddToSelectedTable():

        slDta09[Slcolumn] = file[Slcolumn]

        tree09Duy.delete(\*tree09Duy.get\_children())

        tree09Duy.config(column = tuple(slDta09.columns))

        for column in slDta09.columns:

            tree09Duy.heading(column, text=column)

        # Thêm dữ liệu vào tree

        for row in slDta09.to\_numpy():

            tree09Duy.insert("", "end", values=tuple(row))

        frame109.update\_idletasks()

        canvas09.config(scrollregion=canvas09.bbox('all'))

    def Bochon09():

        slDta09 = DQDuy\_pd.DataFrame()

        tree09Duy.config(column = tuple(slDta09.columns))

        tree09Duy.delete(\*tree09Duy.get\_children())

    def MainTable(DQDuy\_fro):

        tree09.delete(\*tree09.get\_children())

        tree09.config(column = tuple(DQDuy\_fro.columns))

        for column in DQDuy\_fro.columns:

            tree09.heading(column, text=column)

        # Thêm dữ liệu vào tree

        for row in DQDuy\_fro.to\_numpy():

            tree09.insert("", "end", values=tuple(row))

        frame.update\_idletasks()

        canvas.config(scrollregion=canvas.bbox('all'))

    frame0 = Frame(Duyapp)

    frame0.place(x = 10, y = 200)

    scrollbar = tk09DQDuy.Scrollbar(frame0, orient='vertical')

    scrollbarx = tk09DQDuy.Scrollbar(frame0, orient='horizontal')

    scrollbar.pack(side='right', fill='y')

    scrollbarx.pack(side='bottom', fill='x')

    canvas = tk09DQDuy.Canvas(frame0,widt= 400, height= 200,background='white')

    canvas.pack(side='left')

    scrollbar.configure(command= canvas.yview)

    scrollbarx.configure(command= canvas.xview)

    canvas.configure(yscrollcommand=scrollbar.set, xscrollcommand= scrollbarx.set)

    frame = tk09DQDuy.Frame(canvas)

    canvas.create\_window((0,0), window=frame, anchor='nw')

    frame.update\_idletasks()

    canvas.config(scrollregion=canvas.bbox('all'))

##########3

    frame09 = Frame(Duyapp)

    frame09.place(x = 420, y = 200)

    scrollbar09 = tk09DQDuy.Scrollbar(frame09, orient='vertical')

    scrollbarx09 = tk09DQDuy.Scrollbar(frame09, orient='horizontal')

    scrollbar09.pack(side='right', fill='y')

    scrollbarx09.pack(side='bottom', fill='x')

    canvas09 = tk09DQDuy.Canvas(frame09,widt= 200, height= 200,background='white')

    canvas09.pack(side='left')

    scrollbar09.configure(command= canvas09.yview)

    scrollbarx09.configure(command= canvas09.xview)

    canvas09.configure(yscrollcommand=scrollbar09.set, xscrollcommand= scrollbarx09.set)

    frame109 = tk09DQDuy.Frame(canvas09)

    canvas09.create\_window((0,0), window=frame109, anchor='nw')

    frame109.update\_idletasks()

    canvas09.config(scrollregion=canvas09.bbox('all'))

    tree09Duy = ttk09DQDuy.Treeview(frame109, show="headings")

    tree09Duy.pack()

    lbttablemain = Label(Duyapp, text="Bảng xuất thông tin chính")

    lbttablesl = Label(Duyapp, text="Thông tin để vẽ chart")

    lbttablemain.place(x = 140, y = 180)

    lbttablesl.place(x = 470, y = 180)

    tree09 = ttk09DQDuy.Treeview(frame, show="headings")

    tree09.bind("<Button-3>", ShowPopupMenu)

    btnTuychon = Button(Duyapp)

    btnBoChon09 = Button(Duyapp, text='Bỏ chọn tất cả', command= Bochon09)

    LbHienthi09 = Label(Duyapp, text= 'Xin chào', background= 'yellow',width=88, height= 8,wraplength = 100)

    btnThoat09 = Button(Duyapp, text= 'thoat', command= DQDuythoat)

    btnRecord09 = Button(Duyapp, text= 'Ghi âm', command= DQDuyghiam)

    slzcore = tk09DQDuy.LabelFrame(Duyapp, text='Chon moc Zcore(<= x)')

    comboZcore09 = ttk09DQDuy.Combobox(slzcore, values=["1", "2", "3"],width=15, state='readonly')

    comboZcore09.bind("<<ComboboxSelected>>", slZcore)

    btnz = Button(slzcore, text='Xem Zcore',width=10, command=zcore)

    btnz.pack(side=BOTTOM, fill=X, ipady = 3)

    comboZcore09.pack()

    slFillNa = tk09DQDuy.LabelFrame(Duyapp, text='Chọn cách thay thế')

    btnna = Button(slFillNa, text='Filna',width=10, command=fillNa)

    combofilna09 = ttk09DQDuy.Combobox(slFillNa, values=["Mode", "Median", "Drop"],width=15, state='readonly')

    btnTuychon = Button(Duyapp, text= "Tuy chon", command=TuychonDQDuy)

    btnVeBieuDo = Button(Duyapp, text='Chart', command=Choosechart, width=10, height=2)

    slRR = tk09DQDuy.LabelFrame(Duyapp, text='Chuẩn hóa và trích lọc')

    btnChon = Button(slRR, text='Trích lọc', width=10, command= Buoc8)

    btnscale = Button(slRR, text='Rời rạc',width=10, height=3, command=Scale09)

    varskit = IntVar(Duyapp)

    rbtskit09chi = Radiobutton(slRR, variable= varskit, value=1,text = 'chi2').grid(row=0, column= 1, sticky=W)

    rbtskit09f = Radiobutton(slRR, variable= varskit, value=2, text = 'f\_classif').grid(row=1, column= 1)

    spinbox09 = Spinbox(slRR, from\_=1, to=3, width= 5)

    spinbox09.grid(row=1, column=0)

    btnChon.grid(row=0, column= 0)

    btnscale.grid(row=0, column= 2, rowspan= 2)

    btnna.pack(side=BOTTOM, fill=X, ipady =3)

    slFillNa.place(x = 10, y = 430)

    slRR.place(x = 270, y = 430)

    combofilna09.pack()

    btnBoChon09.place(x = 535, y = 430)

    btnVeBieuDo.place(x = 540, y = 460)

    btnThoat09.place(x = 580, y = 150)

    btnTuychon.place(x = 250, y = 150)

    btnRecord09.place(x = 320, y = 150)

    LbHienthi09.place(x = 10, y = 10)

    tree09.pack()

    slzcore.place(x = 130, y = 430)

    Duyapp.mainloop()

## Plot.

### Form.

A screenshot of a graph

Description automatically generated with medium confidence

### Source code.

def plot09():

    nam = [i[-4:]for i in file['date']]

    DQDuy\_np.abs(DQDuy\_stats.zscore(slDta09.\_get\_numeric\_data()))

    NameA = slDta09.columns[0]

    NameB = slDta09.columns[1]

    duydataA = slDta09.iloc[:, 0]

    duydataB = slDta09.iloc[:, 1]

    data = DQDuy\_pd.DataFrame({'age1': nam, NameA:duydataA, NameB: duydataB}).groupby('age1').mean().reset\_index()

    index = DQDuy\_np.arange(len(set(nam)))

    fig, ax = plt09.subplots()

    formChart = tk09DQDuy.Tk()

    formChart.title('Biểu đồ về trực quan về '+ NameA + " " + NameB + " ở Chicago")

    LbHienthi09 = Label(formChart, text= 'Chọn biểu đồ bạn muốn vẽ\nGreen: ' + NameA+ "    Red: " + NameB, background= 'yellow')

    LbHienthi09.pack(side=TOP, fill=X)

    def on\_select09(event):

        match event.widget.get():

            case "Nomal":

                barchartt(combo.get(),1)

            case "Bottom":

                barchartt(combo.get(),2)

    # Tạo combobox

    combo09 = ttk09DQDuy.Combobox(formChart, values=["Nomal", "Bottom"], state='readonly')

    combo09.pack(side=RIGHT, fill=X)

    combo09.bind("<<ComboboxSelected>>", on\_select09)

    def scaterChart():

        plt09.clf()

        plt09.scatter(data['age1'], data[NameA], color = 'green')

        plt09.scatter(data['age1'], data[NameB], color = 'red')

        plt09.xlabel('Năm')

        plt09.ylabel('Ug/m3')

        ax.plot()

        canvas.draw()

    def piechart():

        plt09.clf()

        plt09.pie(data[NameA], labels= data['age1'], shadow= True, startangle=45)

        plt09.title("Trung bình "+NameA + " hằng năm")

        ax.plot()

        canvas.draw()

    def barchartt(type, style):

        plt09.clf()

        if type == "Bar":

            match style:

                case 1:

                    plt09.bar(index,data[NameA], color = 'green', width=0.3)

                    plt09.bar(index + 0.5,data[NameB], color = 'red',width= 0.3)

                    plt09.xticks(index + 0.25, data["age1"])

                    canvas.draw()

                    return

                case 2:

                    plt09.bar(index,data[NameA], color = 'green', width=0.3)

                    plt09.bar(index,data[NameB], width = 0.3, color = 'red', bottom = data[NameA])

                    plt09.xticks(index + 0.25, data["age1"])

                    canvas.draw()

                    return

        elif type == "Barh":

            match style:

                case 1:

                    plt09.barh(index,data[NameA], color = 'green', height=0.3)

                    plt09.barh(index + 0.5,data[NameB], color = 'red',height= 0.3)

                    plt09.yticks(index + 0.25, data["age1"])

                    canvas.draw()

                    return

                case 2:

                    plt09.barh(index,data[NameA], color = 'green', height=0.3)

                    plt09.barh(index,data[NameB], height = 0.3, color = 'red', left = data[NameA])

                    plt09.yticks(index + 0.25, data["age1"])

                    canvas.draw()

                    return

    def scatter3Dchart():

        ax = fig.add\_axes((0, 0, 1, 1), projection='3d')

        ax.scatter3D(data['age1'], data[NameA])

        canvas.draw()

    def histchart():

        plt09.clf()

        plt09.title("Histogram")

        plt09.xlabel('Ug/m3')

        plt09.ylabel('Tần số')

        plt09.hist(data[NameA])

        ax.plot()

        canvas.draw()

    def on\_select(event):

        match event.widget.get():

            case "Scatter":

                scaterChart()

            case "Pie":

                piechart()

            case "Hist":

                histchart()

            case "Bar":

                barchartt("Bar", 1)

            case "Barh":

                barchartt("Barh", 1)

            case "Scatter 3D":

                scatter3Dchart()

    # Tạo combobox

    combo = ttk09DQDuy.Combobox(formChart, values=["Scatter", "Pie", "Hist", "Bar", "Barh", "Scatter 3D"], state='readonly')

    combo.pack(side=LEFT, fill=X)

    combo.bind("<<ComboboxSelected>>", on\_select)

    canvas = FigureCanvasTkAgg(fig, master=formChart)

    canvas.draw()

    # Thêm canvas vào window

    canvas.get\_tk\_widget().pack(side=BOTTOM)

    formChart.mainloop()

## Image processing and Frames.

### Form.

A screenshot of a computer

Description automatically generated with medium confidence

### Source code.

def IMAGE09():

    win = tk09DQDuy.Tk()

    win.title('Image processing')

    win.geometry('1150x650')

    win.configure(background='gray', relief = tk09DQDuy.SUNKEN)

    win.duyvar = 0

    def OpendFileDialog():

        global file\_path

        file\_path = fd.askopenfilename(title="Chọn file video", filetypes=[("Video Files", "\*.mp4;\*.mov")])

    def catFrame():

        directory = fd.askdirectory()

        cap = cv209.VideoCapture(file\_path)

        count = 0

        while cap.isOpened():

            ret,frame = cap.read()

            cv209.imshow('Khung Hinh', frame)

            if cv209.waitKey(10) & 0xFF == ord('s'):#ấn  s để lưu frame

                txt = "{}/{}{}.jpg"

                cv209.imwrite(txt.format(directory,str(entryname.get()),str(count)) , frame)

                count = count + 1

            if cv209.waitKey(10) & 0xFF == ord('q'):

                break

        cap.release()

        cv209.destroyAllWindows()

    def anhXam():

        win.duyvar = 0

        global img2

        img2 = cv209.imread(file.name,cv209.IMREAD\_GRAYSCALE)

        global img\_pil

        img\_pil = Img09.fromarray(cv209.cvtColor(img2, cv209.IMREAD\_GRAYSCALE))

        global img\_tk

        img\_tk = ImgTk09.PhotoImage(img\_pil)

        lblMain09.configure(image=img\_tk)

        frame.update\_idletasks()

        canvas.config(scrollregion=canvas.bbox('all'))

    def anhMau():

        win.duyvar = 1

        global img\_pil

        img\_pil = Img09.fromarray(cv209.cvtColor(img, cv209.COLOR\_BGR2RGB))

        global img\_tk

        img\_tk = ImgTk09.PhotoImage(img\_pil)

        lblMain09.configure(image=img\_tk)

        frame.update\_idletasks()

        canvas.config(scrollregion=canvas.bbox('all'))

    def rotage():

        global img\_pil

        global img\_tk

        if win.duyvar == 1:

            (h, w, d) = img.shape

            center = (w // 2, h // 2)

            M = cv209.getRotationMatrix2D(center,int(entryAngleDuy09.get()), 1.0)

            rotated = cv209.warpAffine(img, M, (w, h))

            img\_pil = Img09.fromarray(cv209.cvtColor(rotated, cv209.COLOR\_BGR2RGB))

            img\_tk = ImgTk09.PhotoImage(img\_pil)

            lblMain09.configure(image=img\_tk)

            frame.update\_idletasks()

            canvas.config(scrollregion=canvas.bbox('all'))

        else:

            (h, w) = img2.shape

            center = (w // 2, h // 2)

            M = cv209.getRotationMatrix2D(center, int(entryAngleDuy09.get()), 1.0)

            rotated = cv209.warpAffine(img2, M, (w, h))

            img\_pil = Img09.fromarray(cv209.cvtColor(rotated, cv209.COLOR\_BGR2RGB))

            img\_tk = ImgTk09.PhotoImage(img\_pil)

            lblMain09.configure(image=img\_tk)

            frame.update\_idletasks()

            canvas.config(scrollregion=canvas.bbox('all'))

    def Rsize():

        global img\_pil

        global img\_tk

        if win.duyvar == 1:

            (h, w, d) = img.shape

            dim = (int(w\* int(entrySizeDuy09.get())), int(h \* int(entrySizeDuy09.get())))

            rd = cv209.resize(img, dim)

            img\_pil = Img09.fromarray(cv209.cvtColor(rd, cv209.COLOR\_BGR2RGB))

            img\_tk = ImgTk09.PhotoImage(img\_pil)

            lblMain09.configure(image=img\_tk)

            frame.update\_idletasks()

            canvas.config(scrollregion=canvas.bbox('all'))

        else:

            (h, w) = img2.shape

            dim = (int(w \* int(entrySizeDuy09.get())), int(h \* int(entrySizeDuy09.get())))

            rd = cv209.resize(img, dim)

            img\_pil = Img09.fromarray(cv209.cvtColor(rd, cv209.COLOR\_BGR2RGB))

            img\_tk = ImgTk09.PhotoImage(img\_pil)

            lblMain09.configure(image=img\_tk)

            frame.update\_idletasks()

    def LayFileanh():

        win.duyvar = 1

        global file

        file = fd.askopenfile(filetypes=[('Jpg Files', '\*.jpg')])

        global img

        img = cv209.imread(file.name)

        global img\_pil

        img\_pil = Img09.fromarray(cv209.cvtColor(img, cv209.COLOR\_BGR2RGB))

        global img\_tk

        img\_tk = ImgTk09.PhotoImage(img\_pil)

        lblMain09.configure(image=img\_tk)

        frame.update\_idletasks()

        canvas.config(scrollregion=canvas.bbox('all'))

    frame0 = Frame(win)

    frame0.place(x = 320, y = 10)

    scrollbar = tk09DQDuy.Scrollbar(frame0, orient='vertical')

    scrollbarx = tk09DQDuy.Scrollbar(frame0, orient='horizontal')

    scrollbar.pack(side='right', fill='y')

    scrollbarx.pack(side='bottom', fill='x')

    canvas = tk09DQDuy.Canvas(frame0,widt= 800, height= 600,background='white')

    canvas.pack(side='left')

    scrollbar.configure(command= canvas.yview)

    scrollbarx.configure(command= canvas.xview)

    canvas.configure(yscrollcommand=scrollbar.set, xscrollcommand= scrollbarx.set)

    frame = tk09DQDuy.Frame(canvas)

    canvas.create\_window((0,0), window=frame, anchor='nw')

    frame.update\_idletasks()

    canvas.config(scrollregion=canvas.bbox('all'))

    lblMain09 = Label(frame, background='white')

    lblMain09.pack()

    framFrames = LabelFrame(win, text="Frames", width=300, height=150)

    framFrames.place(x = 10, y = 10)

    openfile = tk09DQDuy.Button(framFrames, text= 'Open video', command= OpendFileDialog)

    openfile.place(x = 100, y = 100)

    name09 = tk09DQDuy.StringVar()

    name09.set('duy09')

    lblName09 = tk09DQDuy.Label(framFrames, text="Name Foder:")

    lblName09.place(x = 0, y = 0)

    entryname = tk09DQDuy.Entry(framFrames, textvariable=name09)

    entryname.place(x = 100, y = 0)

    lblLocatonSave = Label(framFrames, text="Chọn vị trí lưu thư mục: ")

    buttionFrame = tk09DQDuy.Button(framFrames, text= 'Open Folder', command=catFrame)

    buttionFrame.place(x = 160, y = 50)

    lblLocatonSave.place(x = 0, y = 50)

    framImage = LabelFrame(win, text="Imge", width=300, height=120)

    framImage.place(x = 10, y = 160)

    anhbuttion = tk09DQDuy.Button(framImage , text="open picture", command= LayFileanh)

    anhbuttion.place(x = 100, y = 70)

    anhxam = tk09DQDuy.Button(framImage, text='Anh xam', command= anhXam, width= 8)

    anhxam.place(x = 10, y = 10)

    anhmau09 = tk09DQDuy.Button(framImage, text='Anh mau', command= anhMau, width= 8)

    anhmau09.place(x = 80, y = 10)

    Rotage = tk09DQDuy.Button(framImage, text= 'Xoay', command=rotage, width= 8)

    Rotage.place(x = 150, y = 10)

    rsizes = Button(framImage, text='Resize', command=Rsize, width= 8)

    rsizes.place(x = 220, y =10)

    entryAngleDuy09 = Entry(framImage ,width=10)

    entryAngleDuy09.place(x = 150, y = 40)

    entrySizeDuy09 = Entry(framImage, width=10)

    entrySizeDuy09.place(x = 220, y = 40)

    win.mainloop()

## Game.

### Form.



### Source code.

def GameForm09():

    Game09DQDuy = tk09DQDuy.Tk()

    Game09DQDuy.resizable(False, False)

    Game09DQDuy.title('Car Game')

    Game09DQDuy.geometry('350x240')

    Game09DQDuy.keyType = {

        True : (K\_a, K\_d, K\_w, K\_s),

        False : (K\_LEFT, K\_RIGHT, K\_UP, K\_DOWN)

    }

    def Game():

        keys = Game09DQDuy.keyType[varctronl.get()]

        DUYWIDTH = int(EtWidthApp09.get())

        DUYHEIGHT = int(EtHeightApp09.get())

        ###KHỞI TẠO THƯ VIỆN ĐỂ DÙNG

        pygame.init()

        ##TỐC ĐỘ KHUNG HÌNH CỦA VIDEO

        DUY09FPS = int(EtFps09.get()) # Famres Per Second

        DUY09FPSClock = pygame.time.Clock() #Lặp theo nhịp clock (tham số DUY09FPS)

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

        #####PHẦN 2: NỀN GAME ##############

        ################################;#####

        #TỐC ĐỘ CUỘN NỀN

        DUY09BGSPEED = float(EtCuonNen09.get()) # tốc độ cuộn nền

        DUY09BGIMG = pygame.image.load(bg09) # hình nền

        # LAYER (SURFACE) NỀN

        DISPLAYSURF = pygame.display.set\_mode((DUYWIDTH, DUYHEIGHT))

        pygame.display.set\_caption('stt Ho Tên = Ex7.5: Game = Game ĐUA XE')

        # LỚP HÌNH NỀN = CUỘN NỀN

        class Background09():

            def \_\_init\_\_(self):

                self.x = 0

                self.y = 0

                self.speed = DUY09BGSPEED

                self.img = DUY09BGIMG

                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

        DUY09X\_MARGIN = 80

        DUY09CARWIDTH = 40

        CARHEIGHT = 60

        DUY09CARSPEED = 3

        CARIMG = pygame.image.load(xe09)

        #LỚP XE TRONG GAME

        class Car09():

            def \_\_init\_\_(self):

                self.width = DUY09CARWIDTH

                self.height = CARHEIGHT

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

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

                self.speed = DUY09CARSPEED

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

                self.surface.fill((255, 255, 255))

            def draw(self):

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

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

                if moveLeft == True:

                    self.x -= self.speed

                if moveRight == True:

                    self.x += self.speed

                if moveUp == True:

                    self.y -= self.speed

                if moveDown == True:

                    self.y += self.speed

        LANEWIDTH = 60

        DISTANCE = 200

        OBSTACLESSPEED = 2

        CHANGESPEED = 0.001 #toc do xe xanh

        OBSTACLESIMG = pygame.image.load(vcan09)

        class Obstacles09():

            def \_\_init\_\_(self):

                self.width = DUY09CARWIDTH

                self.height = CARHEIGHT

                self.distance = DISTANCE

                self.speed = OBSTACLESSPEED

                self.changeSpeed = CHANGESPEED

                self.ls = []

                for i in range(5):

                    y = -CARHEIGHT-i\*self.distance

                    lane = random.randint(0, 3)

                    self.ls.append([lane, y])

            def draw(self):

                for i in range(5):

                    x = int(DUY09X\_MARGIN + self.ls[i][0]\*LANEWIDTH + (LANEWIDTH-self.width)/2)

                    y = int(self.ls[i][1])

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

            def update(self):

                for i in range(5):

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

                self.speed += self.changeSpeed

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

                    self.ls.pop(0)

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

                    lane = random.randint(0, 3)

                    self.ls.append([lane, y])

        class Score09():

            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 += float(EtScoreApp09.get())

        def rectCollision(rect1, rect2):

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

                return True

            return False

        def isGameover(car, obstacles):

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

            for i in range(5):

                x = int(DUY09X\_MARGIN + obstacles.ls[i][0]\*LANEWIDTH + (LANEWIDTH-obstacles.width)/2)

                y = int(obstacles.ls[i][1])

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

                if rectCollision(carRect, obstaclesRect) == True:

                    return True

            return False

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

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

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

            headingSize = headingSuface.get\_size()

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

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

            commentSize = commentSuface.get\_size()

            while True:

                for event in pygame.event.get():

                    if event.type == pygame.QUIT:

                        QUITGame()

                    if event.type == pygame.KEYUP:

                        if event.key == K\_SPACE:

                            return

                bg.draw()

                car.draw()

                obstacles.draw()

                score.draw()

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

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

                pygame.display.update()

                DUY09FPSClock.tick(DUY09FPS)

        def QUITGame():

            mss09 = DQDuyTB.askquestion('Thoát', 'Bạn có muốn thoát không')

            if mss09 == 'yes':

                pygame.quit()

                Game09DQDuy.mainloop()

        def gameStart(bg):

            bg.\_\_init\_\_()

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

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

            headingSize = headingSuface.get\_size()

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

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

            commentSize = commentSuface.get\_size()

            while True:

                for event in pygame.event.get():

                    if event.type == pygame.QUIT:

                        QUITGame()

                    if event.type == pygame.KEYUP:

                        if event.key == K\_SPACE:

                            return

                bg.draw()

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

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

                pygame.display.update()

                DUY09FPSClock.tick(DUY09FPS)

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

            car.\_\_init\_\_()

            obstacles.\_\_init\_\_()

            bg.\_\_init\_\_()

            score.\_\_init\_\_()

            moveLeft = False

            moveRight = False

            moveUp = False

            moveDown = False

            while True:

                for event in pygame.event.get():

                    if event.type == pygame.QUIT:

                        pygame.quit()

                        sys.exit()

                    if event.type == KEYDOWN:

                        if event.key == keys[0]:

                            moveLeft = True

                        if event.key == keys[1]:

                            moveRight = True

                        if event.key == keys[2]:

                            moveUp = True

                        if event.key == keys[3]:

                            moveDown = True

                    if event.type == KEYUP:

                        if event.key == keys[0]:

                            moveLeft = False

                        if event.key == keys[1]:

                            moveRight = False

                        if event.key == keys[2]:

                            moveUp = False

                        if event.key == keys[3]:

                            moveDown = False

                if isGameover(car, obstacles):

                    return

                bg.draw()

                bg.update()

                car.draw()

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

                obstacles.draw()

                obstacles.update()

                score.draw()

                score.update()

                pygame.display.update()

                DUY09FPSClock.tick(DUY09FPS)

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

        #PHẦN 8: HÀM MAIN ##

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

        def main():

            bg = Background09()

            car = Car09()

            obstacles = Obstacles09()

            score = Score09()

            gameStart(bg)

            while True:

                gamePlay(bg, car, obstacles, score)

                gameOver(bg, car, obstacles, score)

        main()

    def choosebg():

        global bg09

        bg09 = fd.askopenfilename(initialdir = "./New folder/Image Game/Bg", title = "Select a File", filetypes = (("png files", "\*.png"), ("all files", "\*.\*")))

    def choosexe():

        global xe09

        xe09 = fd.askopenfilename(initialdir = "./New folder/Image Game/Car", title = "Select a File", filetypes = (("png files", "\*.png"), ("all files", "\*.\*")))

    def choosevcan():

        global vcan09

        vcan09 = fd.askopenfilename(initialdir = "./New folder/Image Game/obstales", title = "Select a File", filetypes = (("png files", "\*.png"), ("all files", "\*.\*")))

    lblFps09 = Label(Game09DQDuy, text='Fps(60-200)')

    EtFps09 = Entry(Game09DQDuy)

    lblCuonNen09 = Label(Game09DQDuy, text='Cuộn nền(1-4)')

    EtCuonNen09 = Entry(Game09DQDuy)

    lblWidthApp09 = Label(Game09DQDuy, text='Width app(>=400)')

    EtHeightApp09 = Entry(Game09DQDuy)

    lblHeightApp09 = Label(Game09DQDuy, text='Height app(>= 400)')

    btnBG09 = Button(Game09DQDuy, text= 'Background', width=10,command= choosebg)

    btnXe09 = Button(Game09DQDuy, text= 'Car', width=10, command= choosexe)

    btnVcan09 = Button(Game09DQDuy, text= 'Vat can', width=10, command= choosevcan)

    lblScoreApp09 = Label(Game09DQDuy, text='Tinh diem')

    EtScoreApp09 = Entry(Game09DQDuy)

    EtWidthApp09 = Entry(Game09DQDuy)

    btnStart09 = Button(Game09DQDuy, text= 'Start', width=10, command= Game)

    varctronl = tk09DQDuy.BooleanVar(Game09DQDuy)

    varctronl.set(False)

    CbntCtrol09 = Checkbutton(Game09DQDuy, text= 'Điều khiển bằng ASWD', variable=varctronl)

    lblLuuy = Label(Game09DQDuy, text='Bạn phải chọn đầy đủ các thông số phù hợp gợi ý và chọn hình thì Game mưới chạy bình thường được',wraplength =320)

    lblFps09.grid(row= 0, column= 0)

    EtFps09.grid(row= 0, column= 1)

    lblCuonNen09.grid(row= 1, column= 0)

    EtCuonNen09.grid(row= 1, column= 1)

    lblWidthApp09.grid(row= 2, column=0)

    EtWidthApp09.grid(row=2, column= 1)

    lblHeightApp09.grid(row=3, column= 0)

    EtHeightApp09.grid(row=3, column=1)

    btnBG09.grid(row=0, column=2)

    btnXe09.grid(row = 1, column= 2)

    btnVcan09.grid(row=2, column= 2)

    lblScoreApp09.grid(row= 4, column=0)

    EtScoreApp09.grid(row = 4, column= 1)

    btnStart09.grid(row= 5, column=1)

    CbntCtrol09.grid(row=5, column=0)

    lblLuuy.place(x = 20, y = 170)

    Game09DQDuy.mainloop()

GameForm09()

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

## Kết luận

## Những kết quả đạt được

ĐỒ ÁN HỌC PHẦN đã thực hiện việc Lập trình Python phân tích thăm dò chất lượng không khí ở chicago từ năn 1987 đến năm 2005; 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: PHÂN TÍCH THĂM DÒ CHẤT LƯỢNG KHÔNG KHÍ Ở CHICAGO TỪ NĂN 1987 ĐẾN NĂM 2005.

## Hạn chế

Vì thời gian có hạn nên chưa liên kết voice assistant để ra lệnh phân tích khám phá bằng lời nói.

## Hướng phát triển

Sẽ sử dụng voice assistant để ra lệnh phân tích khám phá bằng sử dụng các kĩ năng về MA, AI, Deep ML để thực hiện.

## Hướng khắc phục các hạn chế

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. <https://www.kaggle.com/datasets/asjad99/chicago-air-pollution> : Dữ liệu nguồn(Ngày tham khảo 15/3/2023)

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