1. 題目，小組成員，口頭報告影片youtube網址

題目:可傳檔及錄音聊天室。

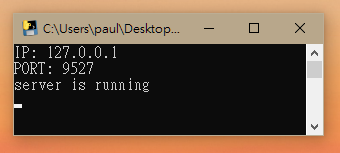
小組成員:林聖博，陳柏凱。

口頭報告youtube網址:

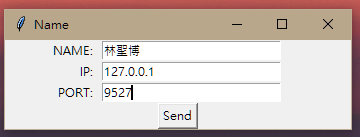
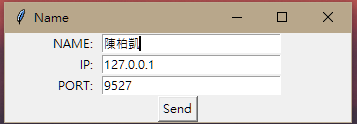
1. 簡要作品說明

一開始我們是想要做一個類似google classroom可以做分享畫面的應用，但嘗試了一下發現時間不夠以及能力不足以及醜陋的長相和林聖博沒有女朋友的情況下，導致沒辦法做出來，但我們有做出可以直播自己螢幕的功能，只是無法使另一方看到我們分享畫面的功能，於是我們就退而求其次，先不做分享畫面，做著做著發現越來越像msn，只好將錯就錯，我們還有製作錄音以及傳送檔案之功能。

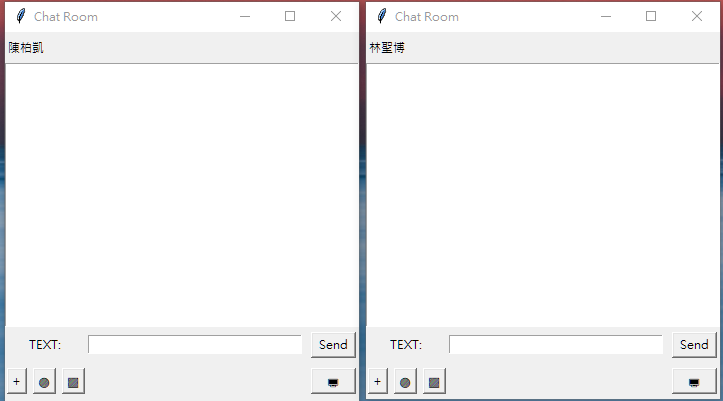
1. 作品執行過程說明(配合畫面)



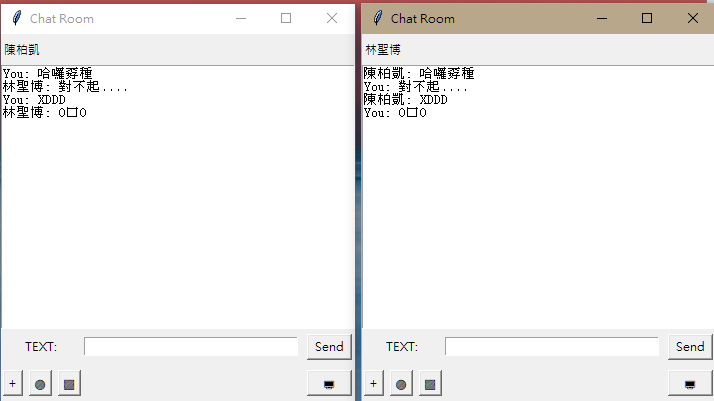
Step 1. 先在server輸入IP以及PORT。



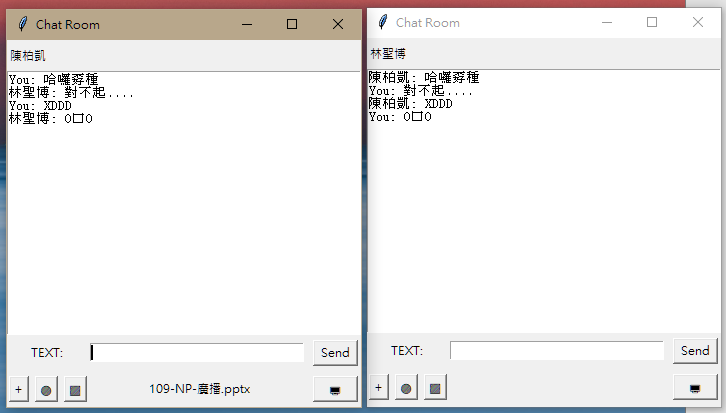
Step 2. 點開client執行檔 輸入名字、IP 及PORT。



Step 3. Step 2按出send之後會跑出聊天視窗。

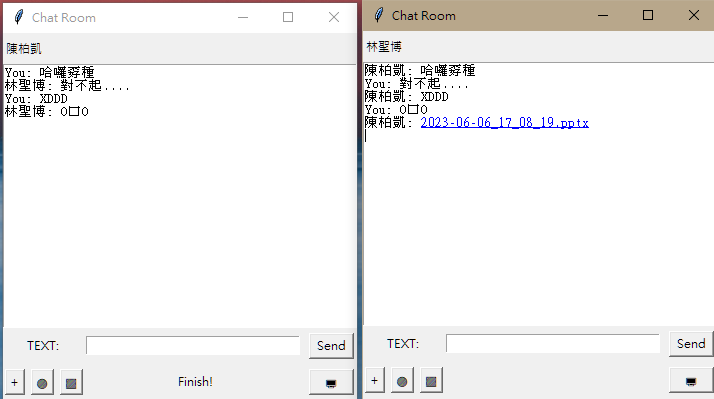


Step 4. 傳送文字。



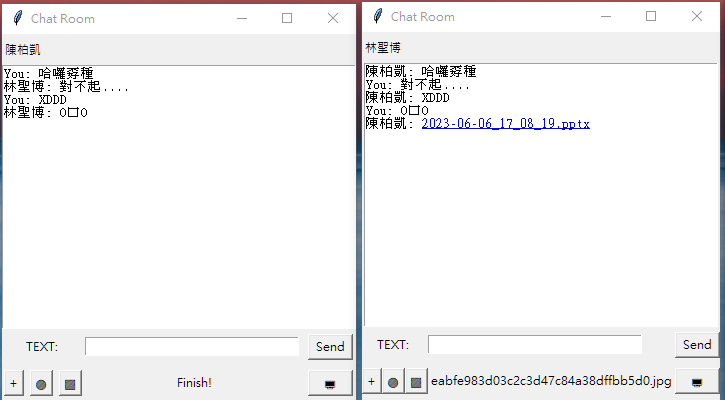


Step5. 按 ’ + ’ 號是傳送檔案(如果是圖片檔的話會直接顯示圖片出來)，選好檔案之後按下SEND就會送出了。



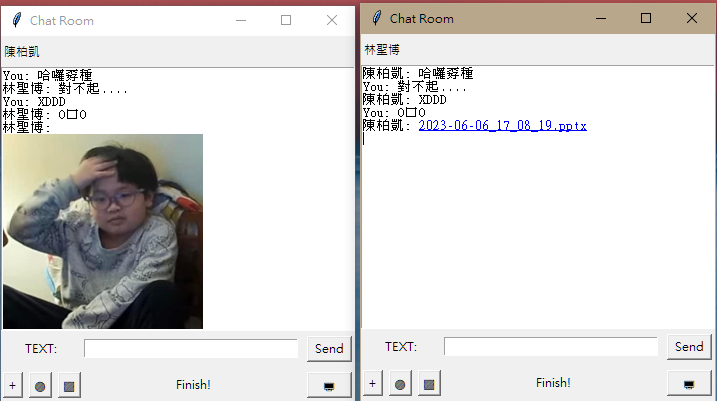


Step6. Step5的送出畫面。



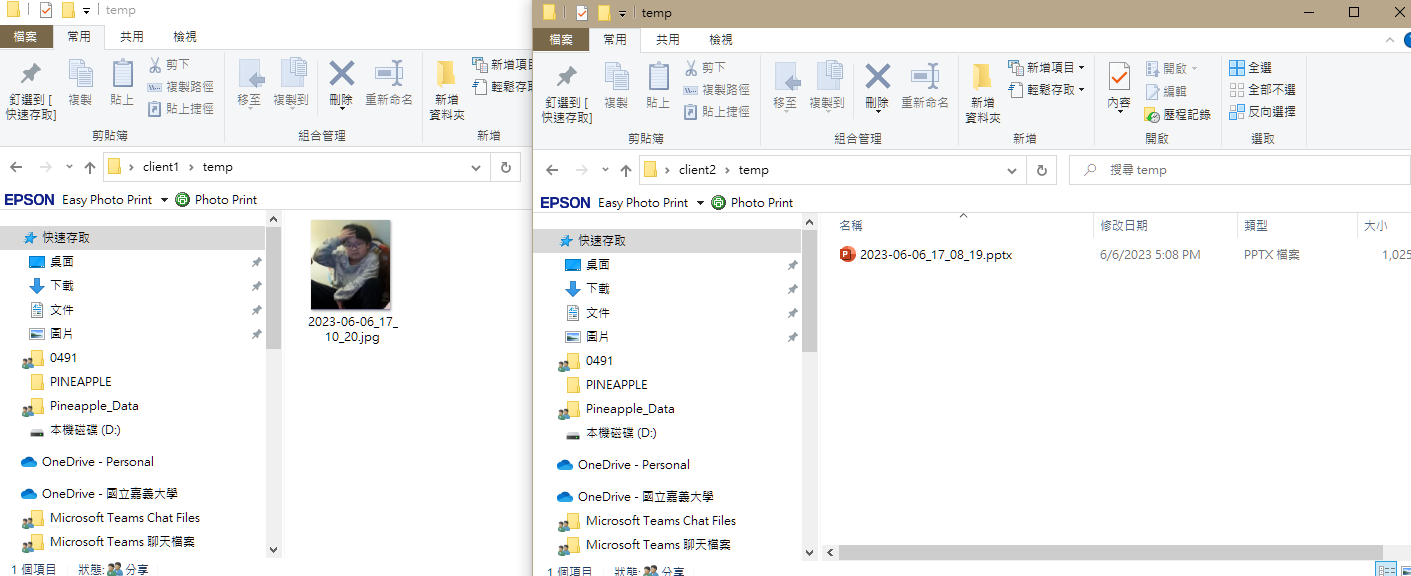


Step7.此為傳送圖片檔。

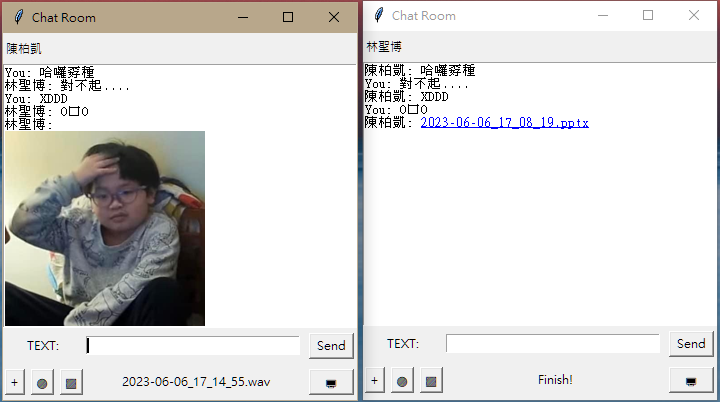




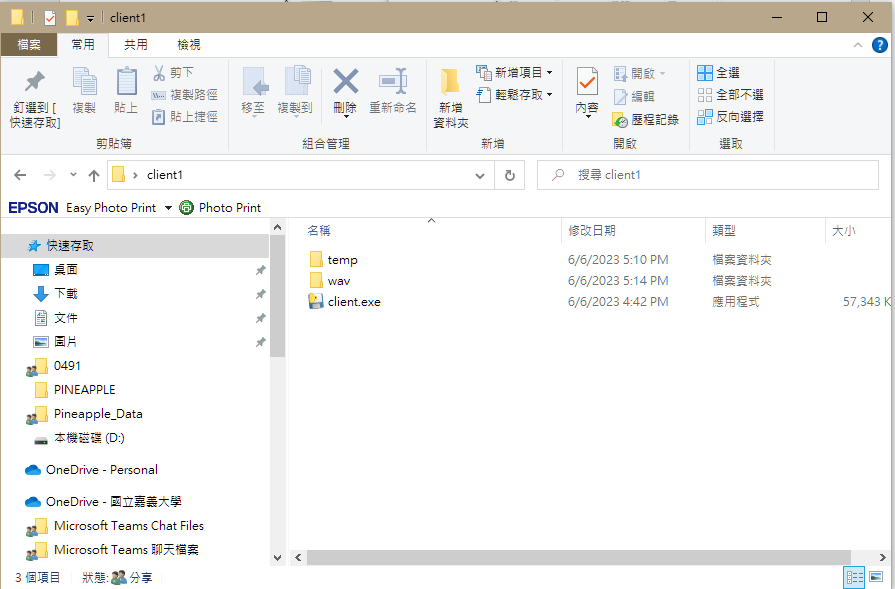
Step8. Step7的送出畫面。



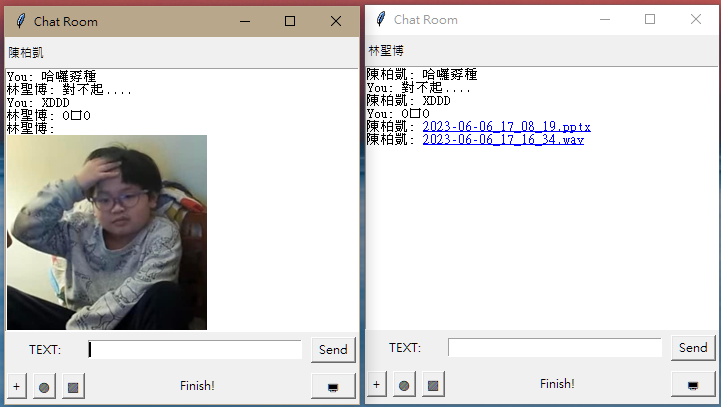
Step9. 剛剛所傳送的檔案會被存在一個名叫temp的資料夾內





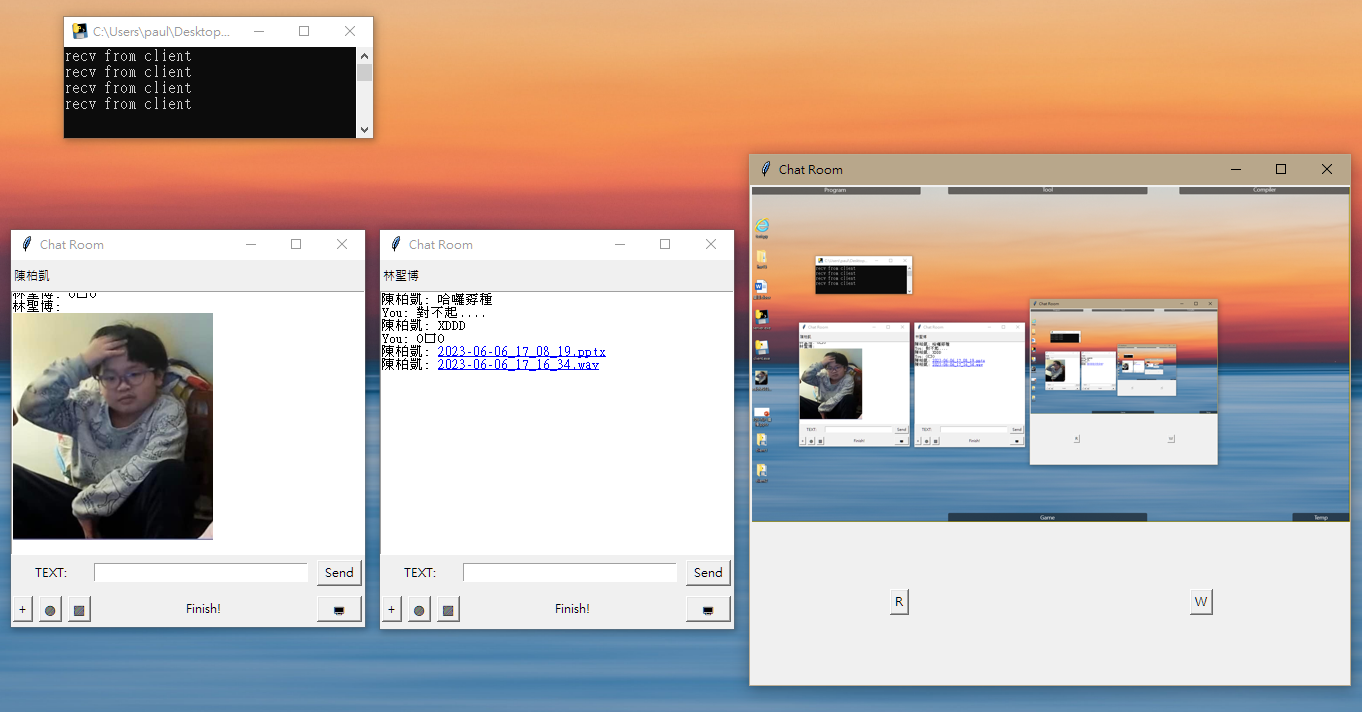


Step10. 按下圈圈是開始錄音，開始錄音按下方形是停止錄音，下面就會出現WAV檔(錄好的音會被存在傳送方wav資料夾)。





Step11. Step10的送出畫面(此傳送之錄音檔也會被存在被傳送方的temp資料夾裡)。





Step12. 按下send下方的螢幕圖示，將會直播自己的畫面(此功能本是要直播自己的畫面給對方看到，但只做好了前者，後者因為在傳送封包過去時會有bug，沒有時間了QAQ，深感遺憾QQ)

(按鈕R為直播自己畫面，按鈕W為分享自己的畫面給對方)

作品說明結束。

1. 本作品架構

一對多。(server對多個client)

1. 本作品的功能

我們在上課所學到的聊天室之外又新增了GUI (錄音、傳送檔案，顯示圖片、顯示對方名稱)，並且把其改成多緒的處理，SERVER端採用轉送的方法，就是當SERVER接收到CLIENT傳過來的資料時，會將所接收到的資料傳給其他所有CLIENT(可以一次有多個CLIENT同時開啟並聊天)。本作品困難的點在於檔案傳送以及封包處理，後來就是利用傳送不同類型的封包方法來分割檔案。我們有挑戰分享畫面的功能，但後來因時間不足以及能力不足下只能做出直播自己畫面的功能，深感遺憾。花最多精力的地方就是傳送檔案，為了做出這個功能我們上網查了許多資料以及多次DEBUG才終於大功告成!! 非常有成就感!!!

1. 本作品使用到課程所教的技術 以及其他自學技術

有用到課程所教的TCP、THREAD以及GUI。

自學的有pyautogui 、opencv以及pillow。

1. 人員分工與時間配合

Code方面為林聖博同學所打，陳柏凱同學最後調整GUI介面。

Word方面為陳柏凱同學所打，YouTube為林聖博同學報告。

1. 心得

林聖博：

這次的作業主要的code架構是我負責的，基本上是從上課教的聊天室去做延伸，從還沒教thread時就已經開始做了，所以那個時候是用select，但後來學會thread後覺得用這個方法的話我寫起來會比較方便，於是又大改了一次，也多弄出GUI還有其他應用。

在傳送檔案的時候遇到比較多困難，但解決後其他功能也就比較好處理了，雖然沒有達成一開始想做出分享畫面的目標，但也多新增了錄音這個原先沒想到要做的功能，就長得越來越像陽春版的MSN了。

陳柏凱:

這次作業程式方面我都是在旁觀摩學習，畢竟我能力不足，只能多學習別人長處，在一些我能幫上忙的地方盡一份心力。在這次的專題中，我學到了不少東西，林聖博同學總是具有耐心及細心地跟我講解，我被他的教學深深感動，讓我燃起了學習程式的鬥志，最後我也要求GUI介面給我設計，讓我有一點參與感，很謝謝他給我這一次的機會>w<。在打word時因為之前有了聖博同學的講解code環節，使我能夠真正了解到哪段程式有哪些功能之類的，使我比較好說明作品功能，使期末專題能夠圓滿結束!

1. 程式碼

Server code:

**import** socket

**import** threading

**import** time

cli\_sd\_TCP **=** **[]**

# cli\_sd\_UDP = []

# def TCP(s):

"""

def UDP(s):

while True:

data = b''

while True:

temp, addr = s.recvfrom(1024)

if temp == b'|END|':

break

if addr not in cli\_sd\_UDP:

cli\_sd\_UDP.append(addr)

print('Connected by', addr)

data += temp

for other in cli\_sd\_UDP:

if other is not addr:

other.sendall(data)

other.sendall(b'|END|')

"""

**def** receive\_message\_TCP**(**client**):**

**while** **True:**

data **=** b''

**try:**

**while** **True:**

temp **=** client**.**recv**(**1024**)**

**if** temp **==** b'|END|'**:**

**break**

**if** temp **==** b'|EXIT|'**:**

**print(**"client exit"**)**

cli\_sd\_TCP**.**remove**(**client**)**

client**.**close**()**

**break**

data **+=** temp

**if** data**:**

**print(**'recv from client'**)**

**try:**

**for** other **in** cli\_sd\_TCP**:**

**if** other **is** **not** client**:**

other**.**sendall**(**data**)**

other**.**sendall**(**b'|END|'**)**

**except:**

**pass**

**except:**

**pass**

HOST **=** **str(input(**'IP: '**))**

PORT\_TCP **=** **int(input(**'PORT: '**))**

# PORT\_UDP = 1268

s\_TCP **=** socket**.**socket**(**socket**.**AF\_INET**,** socket**.**SOCK\_STREAM**)**

# s\_UDP = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

s\_TCP**.**bind**((**HOST**,** PORT\_TCP**))**

# s\_UDP.bind((HOST, PORT\_UDP))

s\_TCP**.**listen**(**1**)**

**print(**"server is running"**)**

**while** **True:**

conn**,** addr **=** s\_TCP**.**accept**()**

**print(**'Connected by'**,** addr**)**

cli\_sd\_TCP**.**append**(**conn**)**

threading**.**Thread**(**target**=**receive\_message\_TCP**,** args**=(**conn**,)).**start**()**

# threading.Thread(target=TCP, args=(s\_TCP,)).start()

# threading.Thread(target=UDP, args=(s\_UDP,)).start()

Client code:

**import** time

**import** tkinter **as** tk

**import** numpy **as** np

**from** tkHyperlinkManager **import** HyperlinkManager

**from** tkinter **import** messagebox

**from** tkinter **import** filedialog

**from** PIL **import** Image**,** ImageTk

**from** functools **import** partial

**import** webbrowser

**import** pyautogui

**import** threading

**import** datetime

**import** imghdr

**import** socket

**import** pyaudio

**import** wave

**import** cv2

**import** os

s **=** socket**.**socket**(**socket**.**AF\_INET**,** socket**.**SOCK\_STREAM**)**

# s\_UDP = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

name **=** **str()**

file\_exist **=** **False**

photo **=** **[]**

is\_running **=** **True**

is\_watching **=** **True**

streaming **=** b''

is\_recording **=** **True**

**def** internet**(**HOST**,** PORT**):**

s**.**connect**((**HOST**,** PORT**))**

s**.**setblocking**(False)**

**def** on\_closing**():**

**if** tk**.**messagebox**.**askokcancel**(**"Quit"**,** "Do you want to quit?"**):**

s**.**sendall**(**b'|EXIT|'**)**

s**.**close**()**

root**.**destroy**()**

**def** receive**():**

**global** streaming

**while** **True:**

data **=** b''

length **=** 0

**try:**

**while** **True:**

temp **=** s**.**recv**(**1024**)**

# length += len(temp)

# print(len(temp), length)

**if** temp **==** b'|END|'**:**

**break**

data **+=** temp

**except:**

**pass**

**if** data**:**

**if** data**.**startswith**(**b'|STREAMING|'**):**

file\_data **=** data**[**11**:]**

frame **=** cv2**.**cvtColor**(**np**.**array**(**file\_data**),** cv2**.**COLOR\_RGB2BGR**)**

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

streaming **=** ImageTk**.**PhotoImage**(**image**=**Image**.**fromarray**(**image**))**

**elif** data**.**startswith**(**b'FILE:'**):**

file\_extension**,** file\_data **=** data**.**split**(**b'|EXTENSION|'**)**

file\_size**,** file\_data **=** file\_data**.**split**(**b'|SIZE|'**)**

**print(**file\_size**.**decode**())**

file\_data**,** file\_N **=** file\_data**.**split**(**b'|SPLIT|'**)**

file\_extension **=** file\_extension**[**5**:]**

datetime\_str **=** datetime**.**datetime**.**now**().**strftime**(**"%Y-%m-%d\_%H\_%M\_%S"**)**

file\_path **=** os**.**getcwd**()** **+** '\\temp\\'

**if** **not** os**.**path**.**exists**(**file\_path**):**

os**.**mkdir**(**file\_path**)**

file\_name **=** datetime\_str **+** file\_extension**.**decode**()**

file\_path **+=** file\_name

**with** **open(**file\_path**,** 'wb+'**)** **as** file**:**

file**.**write**(**file\_data**)**

file\_N **=** file\_N**.**split**(**b'|END|'**)**

**if** imghdr**.**what**(**file\_path**)** **is** **not** **None:**

**global** photo

image **=** Image**.open(**file\_path**)**

width**,** height **=** image**.**size

target\_width **=** 200

target\_height **=** **int(**target\_width **/** **(**width **/** height**))**

resized\_image **=** image**.**resize**((**target\_width**,** target\_height**))**

photo**.**append**(**ImageTk**.**PhotoImage**(**resized\_image**))**

display**.**insert**(**tk**.**END**,** '{}: '**.format(**file\_N**[**0**].**decode**())** **+** '\n'**)**

display**.**image\_create**(**tk**.**END**,** image**=**photo**[-**1**])**

display**.**insert**(**tk**.**END**,** '\n'**)**

**else:**

display**.**insert**(**tk**.**END**,** '{}: '**.format(**file\_N**[**0**].**decode**()))**

hyperlink **=** HyperlinkManager**(**display**)**

display**.**insert**(**tk**.**END**,**

file\_name**,**

hyperlink**.**add**(**partial**(**webbrowser**.open,** f"file://{file\_path}"**)))**

display**.**insert**(**tk**.**END**,** '\n'**)**

**else:**

N**,** message **=** data**.**split**(**b'|NAME|'**)**

message **=** message**.**split**(**b'|END|'**)**

display**.**insert**(**tk**.**END**,** '{}: '**.format(**N**.**decode**())** **+** message**[**0**].**decode**()** **+** '\n'**)**

**def** send**():**

**global** file\_exist

message **=** entry\_message**.**get**()**

**if** file\_exist**:**

s**.**sendall**(**name**.**encode**())**

s**.**sendall**(**b'|END|'**)**

label\_filepath**.**config**(**text**=**"Finish!"**)**

file\_exist **=** **False**

**else:**

s**.**sendall**(**name**.**encode**()** **+** b'|NAME|' **+** message**.**encode**())**

s**.**sendall**(**b'|END|'**)**

display**.**insert**(**tk**.**END**,** 'You: ' **+** message **+** '\n'**)**

entry\_message**.**delete**(**0**,** tk**.**END**)**

**def** send\_file**():**

**global** file\_exist

file\_path **=** filedialog**.**askopenfilename**()**

file\_name **=** os**.**path**.**basename**(**file\_path**)**

file\_extension **=** os**.**path**.**splitext**(**file\_name**)[**1**]**

file **=** **open(**file\_path**,** "rb"**)**

data **=** file**.**read**()**

header **=** b'FILE:'

s**.**sendall**(**header**)**

s**.**sendall**(**file\_extension**.**encode**()** **+** b'|EXTENSION|'**)**

s**.**sendall**(str(len(**data**)).**encode**())**

s**.**sendall**(**b'|SIZE|'**)**

s**.**sendall**(**data**)**

s**.**sendall**(**b'|SPLIT|'**)**

file\_exist **=** **True**

label\_filepath**.**config**(**text**=**file\_name**)**

**def** send\_audio**(**file\_path**,** file\_name**):**

**global** file\_exist

file **=** **open(**file\_path **+** file\_name**,** "rb"**)**

data **=** file**.**read**()**

header **=** b'FILE:'

s**.**sendall**(**header**)**

s**.**sendall**(**'.wav'**.**encode**()** **+** b'|EXTENSION|'**)**

s**.**sendall**(str(len(**data**)).**encode**())**

s**.**sendall**(**b'|SIZE|'**)**

s**.**sendall**(**data**)**

s**.**sendall**(**b'|SPLIT|'**)**

file\_exist **=** **True**

label\_filepath**.**config**(**text**=**file\_name**)**

**def** record\_audio**():**

**global** is\_recording

is\_recording **=** **True**

threading**.**Thread**(**target**=**recording**).**start**()**

**def** recording**():**

chunk **=** 1024

sample\_format **=** pyaudio**.**paInt16

channels **=** 1

sample\_rate **=** 44100

audio **=** pyaudio**.**PyAudio**()**

stream **=** audio**.open(format=**sample\_format**,**

channels**=**channels**,**

rate**=**sample\_rate**,**

frames\_per\_buffer**=**chunk**,**

**input=True)**

**print(**"Recording started..."**)**

frames **=** **[]**

**while** is\_recording**:**

data **=** stream**.**read**(**chunk**)**

frames**.**append**(**data**)**

**print(**"Recording finished."**)**

stream**.**stop\_stream**()**

stream**.**close**()**

audio**.**terminate**()**

datetime\_str **=** datetime**.**datetime**.**now**().**strftime**(**"%Y-%m-%d\_%H\_%M\_%S"**)**

audio\_name **=** datetime\_str **+** ".wav"

file\_path **=** os**.**getcwd**()** **+** '\\wav\\'

**if** **not** os**.**path**.**exists**(**file\_path**):**

os**.**mkdir**(**file\_path**)**

# 將錄製的音訊寫入WAV檔案

wave\_file **=** wave**.open(**file\_path **+** audio\_name**,** 'wb'**)**

wave\_file**.**setnchannels**(**channels**)**

wave\_file**.**setsampwidth**(**audio**.**get\_sample\_size**(**sample\_format**))**

wave\_file**.**setframerate**(**sample\_rate**)**

wave\_file**.**writeframes**(**b''**.**join**(**frames**))**

wave\_file**.**close**()**

send\_audio**(**file\_path**,** audio\_name**)**

**def** stop\_audio**():**

**global** is\_recording

is\_recording **=** **False**

**def** screenshare**(**share**,** img\_canvas**):**

**global** is\_running

**while** is\_running**:**

screenshot **=** pyautogui**.**screenshot**()**

width**,** height **=** screenshot**.**size

target\_width **=** 600

target\_height **=** **int(**target\_width **/** **(**width **/** height**))**

resized\_screenshot **=** screenshot**.**resize**((**target\_width**,** target\_height**))**

frame **=** cv2**.**cvtColor**(**np**.**array**(**resized\_screenshot**),** cv2**.**COLOR\_RGB2BGR**)**

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

tk\_img **=** ImageTk**.**PhotoImage**(**image**=**Image**.**fromarray**(**image**))**

data **=** resized\_screenshot**.**tobytes**()**

"""

s\_UDP.sendall(data)

s\_UDP.sendall(b'|END|')

"""

img\_canvas**.**create\_image**(**0**,** 0**,** image**=**tk\_img**,** anchor**=**tk**.**NW**)**

share**.**update**()**

share**.**destroy**()**

**def** screenshare\_close**():**

**global** is\_running

is\_running **=** **False**

**def** screenshare\_window**():**

**global** is\_running

is\_running **=** **True**

share **=** tk**.**Toplevel**(**root**)**

share**.**geometry**(**'600x500+200+100'**)**

# s\_UDP.connect(('127.0.0.1', 1268))

img\_canvas **=** tk**.**Canvas**(**share**,** height**=**400**,** width**=**600**)**

record\_button **=** tk**.**Button**(**share**,** text**=**'R'**,** command**=lambda:** screenshare**(**share**,** img\_canvas**))**

watch\_button **=** tk**.**Button**(**share**,** text**=**'W'**,** command**=lambda:** watch\_window**(**share**))**

img\_canvas**.**grid**(**row**=**0**,** column**=**0**,** columnspan**=**2**)**

record\_button**.**grid**(**row**=**1**,** column**=**0**)**

watch\_button**.**grid**(**row**=**1**,** column**=**1**)**

share**.**protocol**(**"WM\_DELETE\_WINDOW"**,** screenshare\_close**)**

**def** watch\_close**():**

**global** is\_watching

is\_watching **=** **False**

**def** watch\_window**(**share**):**

**global** streaming

**global** is\_watching

is\_watching **=** **True**

watch **=** tk**.**Toplevel**(**share**)**

watch**.**geometry**(**'600x500+200+100'**)**

img\_canvas **=** tk**.**Canvas**(**watch**,** height**=**400**,** width**=**600**)**

img\_canvas**.**grid**(**row**=**0**,** column**=**0**)**

watch**.**protocol**(**"WM\_DELETE\_WINDOW"**,** watch\_close**)**

**while** is\_watching**:**

"""

temp, addr = s\_UDP.recvfrom(1024)

data = b''

while True:

data += temp

"""

img\_canvas**.**create\_image**(**0**,** 0**,** image**=**streaming**,** anchor**=**tk**.**NW**)**

watch**.**update**()**

watch**.**destroy**()**

**def** create**():**

**global** name

name **=** entry\_name\_input**.**get**()**

IP **=** entry\_IP\_input**.**get**()**

PORT **=** entry\_PORT\_input**.**get**()**

internet**(str(**IP**),** **int(**PORT**))**

root**.**title**(**"Chat Room"**)**

label\_name\_input**.**destroy**()**

entry\_name\_input**.**destroy**()**

label\_IP\_input**.**destroy**()**

entry\_IP\_input**.**destroy**()**

label\_PORT\_input**.**destroy**()**

entry\_PORT\_input**.**destroy**()**

button\_info**.**destroy**()**

label\_name**.**config**(**text**=**name**)**

label\_name**.**grid**(**row**=**0**,** column**=**0**,** columnspan**=**5**,** padx**=**0**,** pady**=(**5**,** 5**),** sticky**=**'W'**)**

display**.**grid**(**row**=**1**,** column**=**0**,** rowspan**=**3**,** columnspan**=**5**)**

label\_TEXT**.**grid**(**row**=**4**,** column**=**0**,** columnspan**=**3**,** padx**=**0**,** pady**=(**5**,** 5**))**

entry\_message**.**grid**(**row**=**4**,** column**=**3**,** padx**=**0**,** pady**=(**5**,** 5**),** sticky**=**'W'**)**

button\_send**.**grid**(**row**=**4**,** column**=**4**,** padx**=**0**,** pady**=(**5**,** 5**))**

button\_file**.**grid**(**row**=**5**,** column**=**0**,** padx**=**0**,** pady**=(**5**,** 5**))**

button\_audio**.**grid**(**row**=**5**,** column**=**1**,** padx**=**0**,** pady**=(**5**,** 5**))**

button\_audio\_stop**.**grid**(**row**=**5**,** column**=**2**,** padx**=**0**,** pady**=(**5**,** 5**))**

label\_filepath**.**grid**(**row**=**5**,** column**=**3**)**

button\_share**.**grid**(**row**=**5**,** column**=**4**)**

root **=** tk**.**Tk**()**

root**.**title**(**'Name'**)**

label\_name\_input **=** tk**.**Label**(**root**,** text**=**"NAME: "**)**

label\_name\_input**.**grid**(**row**=**0**,** column**=**0**,** padx**=(**45**,** 3**),** pady**=**0**,** sticky**=**'E'**)**

entry\_name\_input **=** tk**.**Entry**(**root**,** width**=**25**)**

entry\_name\_input**.**grid**(**row**=**0**,** column**=**1**,** padx**=(**0**,** 70**),** pady**=**0**)**

label\_IP\_input **=** tk**.**Label**(**root**,** text**=**"IP: "**)**

label\_IP\_input**.**grid**(**row**=**1**,** column**=**0**,** padx**=(**45**,** 3**),** pady**=**0**,** sticky**=**'E'**)**

entry\_IP\_input **=** tk**.**Entry**(**root**,** width**=**25**)**

entry\_IP\_input**.**grid**(**row**=**1**,** column**=**1**,** padx**=(**0**,** 70**),** pady**=**0**)**

label\_PORT\_input **=** tk**.**Label**(**root**,** text**=**"PORT: "**)**

label\_PORT\_input**.**grid**(**row**=**2**,** column**=**0**,** padx**=(**45**,** 3**),** pady**=**0**,** sticky**=**'E'**)**

entry\_PORT\_input **=** tk**.**Entry**(**root**,** width**=**25**)**

entry\_PORT\_input**.**grid**(**row**=**2**,** column**=**1**,** padx**=(**0**,** 70**),** pady**=**0**)**

button\_info **=** tk**.**Button**(**root**,** text**=**'Send'**,** command**=**create**)**

button\_info**.**grid**(**row**=**3**,** column**=**0**,** columnspan**=**2**)**

label\_name **=** tk**.**Label**(**root**,** text**=**""**)**

display **=** tk**.**Text**(**root**,** height**=**20**,** width**=**50**)**

label\_TEXT **=** tk**.**Label**(**root**,** text**=**"TEXT: "**)**

entry\_message **=** tk**.**Entry**(**root**,** width**=**30**)**

button\_send **=** tk**.**Button**(**root**,** text**=**'Send'**,** command**=**send**,** width**=**5**)**

button\_file **=** tk**.**Button**(**root**,** text**=**'+'**,** command**=**send\_file**)**

button\_audio **=** tk**.**Button**(**root**,** text**=**'🔴'**,** command**=**record\_audio**)**

button\_audio\_stop **=** tk**.**Button**(**root**,** text**=**'🟥'**,** command**=**stop\_audio**)**

label\_filepath **=** tk**.**Label**(**root**,** text**=**""**)**

button\_share **=** tk**.**Button**(**root**,** text**=**'🖥'**,** command**=**screenshare\_window**,** width**=**5**)**

threading**.**Thread**(**target**=**receive**).**start**()**

root**.**protocol**(**"WM\_DELETE\_WINDOW"**,** on\_closing**)**

root**.**mainloop**()**