

tags: 計算機網路

socket programming 作業

資工三A 108502516 謝程偉

默契檢測器

功能

- 一個最多容納五人的server，能夠計算所有client完成輸入所花費的時間
- 當目前的所有client輸入ready時，server會宣布開始
- client輸入hit時，可以得到自己的hit時間資訊，同時server會記錄時間
- 當所有client都hit時，計算第一個和最後一個client按下hit的時間差，server廣播時間差資訊並關閉socket，結束server和client的程式

環境

- windows 10
- python 3.9.9

函式庫使用

- socket: 用於建立server和client的網路傳輸端口，在這個實作中，使用TCP方法。
- thread: 為了做到同時接收、發送訊息，並且讓server能夠為每個client建立socket，需要用到thread來平行運算處理。
- time, datetime: 用於抓取時間計算時間差

架構

server

1. 建立server socket並bind至指定IP, port。
2. 透過accept接收連接的client資訊並建立對應的client thread
3. 在thread中處理client的訊息、執行應用
4. 檢查client準備、hit狀態，當所有client都hit時關閉所有socket

5. 當server socket被關閉即可結束程式

client

可使用的指令為ready, hit

1. 通過指定的IP, port連上server
2. 建立發送訊息、接收訊息的兩種thread
3. 在thread中對特定的訊息進行處理，如自己輸入exit結束程式或接收server的訊息結束程式等。

pseudo code

server

```
import socket
import threading
import time
import datetime

bind_ip = "127.0.0.1"
bind_port = 48763
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((bind_ip, bind_port))
server.listen(5)
clientList = [] #存放所有client socket
readyList = [] #存放client的ready資訊
hitList = [] #存放client的hit資訊
end = False #結束程式判斷
allHit = False #檢查全部hit或沒hit
zeroHit = True
firstHit = datetime.datetime.now() #存放第一個hit的時間
lastHit = datetime.datetime.now() #存放第二個hit的時間

def clientThread(client, addr): #用於client的thread宣告
    global zeroHit, allHit, end, firstHit, lastHit
    send_welcome_information()
    clientIndex = len(clientList)-1 #存放自己的list引數
    while True:
        try: #當socket關閉出現錯誤可以直接關閉
            data = client.recv(48763)
            message = data.decode()
            if message == "ready":
                readyList[clientIndex] = 1
                send_ready_information()
                hitCheck = True when all ready
                if hitCheck==True:
                    send_start_information()
            elif message == "exit":
                end = True
                break
            elif message == "hit":
                hitCheck = True when all ready
                if hitCheck==False:
                    broadcast("Players are not ready!")
                elif hitCheck==True:
                    hitList[clientIndex] = 1
                    allHit = True when all hit
                    localtime = time.localtime()
                    send_hittime_information()
                    if zeroHit == True:
                        send_first_hit_information()
                        firstHit = datetime.datetime.now()
```

```
        zeroHit = False
    if allHit == True:
        send_last_hit_information()
        lastHit = datetime.datetime.now()
        timeInterval = (lastHit-firstHit).seconds
        send_all_hit_information()
        close_all_socket()
        break
    else:
        print(addr, ":", message)
except:
    break

def broadcast(msg): #對所有client廣播
    for s in clientList:
        s.send(msg.encode())

while True:
    try:
        client, addr = server.accept()
        clientList.append(client)
        readyList.append(0)
        hitList.append(0)
        print(addr, "is connected!")
        print(len(clientList), "player in this room now")
        t=threading.Thread(target=clientThread, args=(client, addr))
        t.start()
    except:
        break
```