



קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
סמל מוסד 441287



שון באזוב - Multithreaded Chat

תיק פרוייקט

שם התלמיד - שון באזוב

ת.ז - 212553192

בית ספר - מקיף עירוני ח'

שם המנחה - קובי שוצמן

מועד הגשה - יוני 2020

שם הפרוייקט - Multithreaded Chat





קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
סמל מוסד 441287



שון באזוב - Multithreaded Chat

תוכן עניינים

1	תיק פרויקט
2	תוכן עניינים
3	מבוא
4	מבנה/ארכיטקטורה של הפרויקט
5	מדריך למשתמש
15	מדריך למפתח
23	סיכום אישי / רפלקציה
25	ביבליוגרפיה

מבוא

הנושא בו עוסקת עבודתי, הוא תקשורת בין מספר רב של מחשבים, ע"י שימוש ב"תהליכון" (Multithreading) והוא נבחר מכמה סיבות. המרכזית שבהן היא כמובן הסקרנות לשמה. מתוך תחומי העניין הקיימים שלי, ביניהם גם עיסוק משחקי מחשב, שבתוכם יש מערכת צ'אט בנויה, ואף פעם לא הבנתי איך זה עובד, או איך אפשר לחבר גם משחק, וגם תקשורת בין לקוחות, וליישם אותם ביחד בתוכנה אחת. כתוצאה מכך, נבע בי הרצון לחקור, ללמוד ואף ליישם בעצמי מערכת שתהיה מסוגלת ליישם את הסקרנות הזאת, ולבנות ממשק המתקשר בין מחשבים בכמה דרכים. העובדה כי לא ידעתי, בתחילה, כיצד להוציא לפועל דבר שכזה, היא זו שהיוותה את האתגר והמריצה אותי להשגת ביצועו. הבעיה העיקרית שבה נתקלתי היא כיצד אני יכול להשתמש בשיטת ה Threading ולגרום לכמה שורות בקוד לרוץ במקביל, משום שלשיטה זו יש הרבה השלכות כאשר אני משתמש בה, וגם צריך לשים כל הזמן לב לאיזה שורות רצות בקוד במקביל. בבניית הצ'אט הבנתי תוך כדי שצ'אט הוא לא כל כך פשוט לבנייה כמו שחושבים, יש הרבה מקרי קיצון ויש הרבה סיטואציות שצריך לחשוב עליהם עוד לפני, למשל - כיצד לקבל מידע מן הלקוח, ובאותו הזמן לתת את האפשרות שליחת מידע, ושאחד לא יפריע לביצועו של השני. סיבה נוספת ליצירת הפרוייקט היא כדי להבין בצורה יותר מסובכת כיצד עובדת התקשורת בין שרת ולקוח. המטרה אם כן, הייתה ליצור ממשק גרפי המהווה פלטפורמה לתקשורת בין מחשבים שונים דרך שימוש בשרת כמקור ל"חיבור" ביניהם, ולתת להם גם את האפשרות לשחק אחד נגד השני במשחק הנקרא "Battleship". הפרוייקט מציע פיתרון בכך שהוא מראה כיצד ניתן לקשר בעיקר בין ממשק גרפי לעבודות בסיסיות בפיתרון או בכללי בשפות תכנות, הוא מצליח לקשר בין התוכנית של הצ'אט רב משתתפים וגם בונה לו ממשק גרפי מתאים, וגם מצליח לקשר לכך משחק בעל ממשק גרפי נוסף שהוא גם תוכנית בשם עצמה. אני יודע שבפרוייקט הזה, אני עשיתי את מה שאני תמיד רציתי לעשות ומה שסיקרן אותי, ועצם כך שאני הצלחתי והבנתי זאת, נותן לי את האפשרות לעשות דברים שימושיים יותר בעתיד. שאלות שהמערכת עונה עליהן -

- כיצד ניתן לקוח יכול לתקשר עם לקוח אחר?
- כיצד ניתן לחבר שתי תוכניות מורכבות לתוכנית אחת?



קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
סמל מוסד 441287



שון באזוב - Multithreaded Chat

מבנה/ארכיטקטורה של הפרויקט



קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
סמל מוסד 441287



שון באזוב - Multithreaded Chat

מדריך למשתמש

כדי לכתוב את הפרויקט שלי השתמשתי בעיקר בתוכנת PyCharm כאשר ה-Interpreter שלה בהגדרות התוכנית הוא בגרסת 3.7 (גם גרסת 3.6 יכולה לעבוד) והממשק של התוכנה נראה כך:

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help proj [C:\Users\Shonb\PycharmProjects\proj] - server.py - PyCharm
proj
  Project
  Project Files
  venv library root
  gui.py
  server.py
  External Libraries
  < Python 3.7 (proj) > C:\Users\Shonb\PycharmProjects\proj
  Scratches and Consoles

server.py
9
10 class ChatServer:
11     clients_list = []
12
13     last_received_message = ""
14
15     def __init__(self):
16         self.server_socket = None
17         self.create_listening_server()
18
19     def create_listening_server(self):
20         self.server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
21         local_ip = "127.0.0.1"
22         local_port = 18319
23         # this will allow you to immediately restart a TCP server
24         self.server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
25         # this makes the server listen to requests coming from other computers on the network
26         self.server_socket.bind((local_ip, local_port))
27         print("listening for incoming messages..")
28         self.server_socket.listen(5)
29         self.receive_messages_in_a_new_thread()
30
31     def receive_messages(self, so):
32         while True:
33             incoming_buffer = so.recv(1024)
34             if not incoming_buffer:
35                 break
36             self.last_received_message = incoming_buffer.decode('utf-8')
37             self.broadcast_to_all_clients(so) # send to all clients
38             so.close()
39
40     def broadcast_to_all_clients(self, sender_socket):
41         ChatServer.create_listening_server()
```



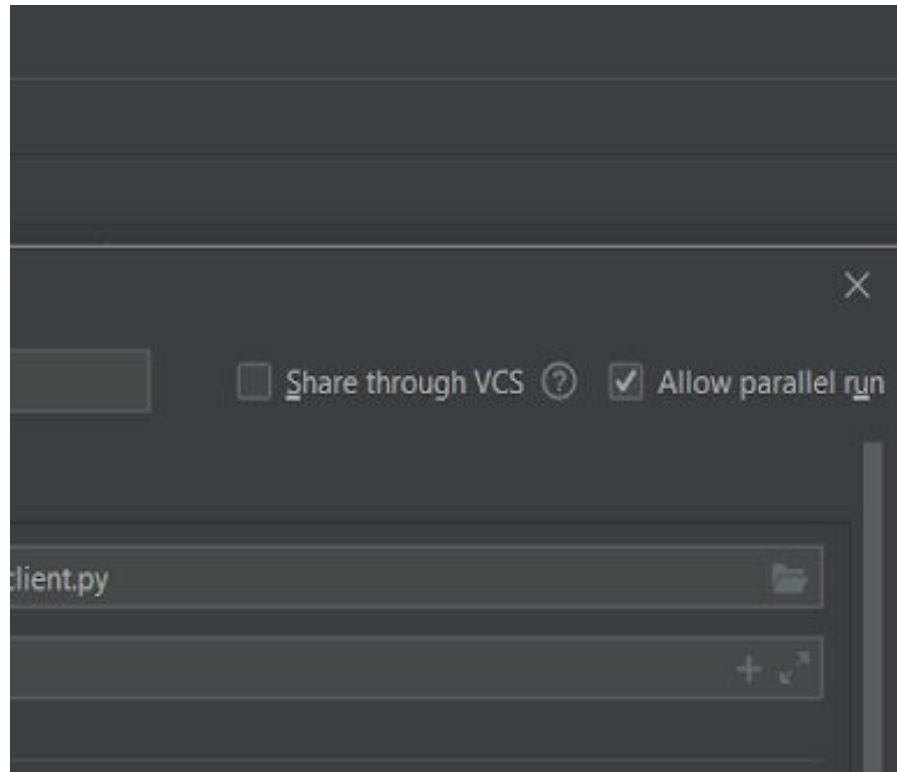
קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
סמל מוסד 441287



שון באזוב - Multithreaded Chat

כדי שיהיה אפשר להפעיל כמה משתמשים באותו המחשב, במקרה ורוצים, אז יש צורך להפעיל זאת (תלוי בגרסה של PyCharm) וכדי להפעיל זאת הולכים ל-

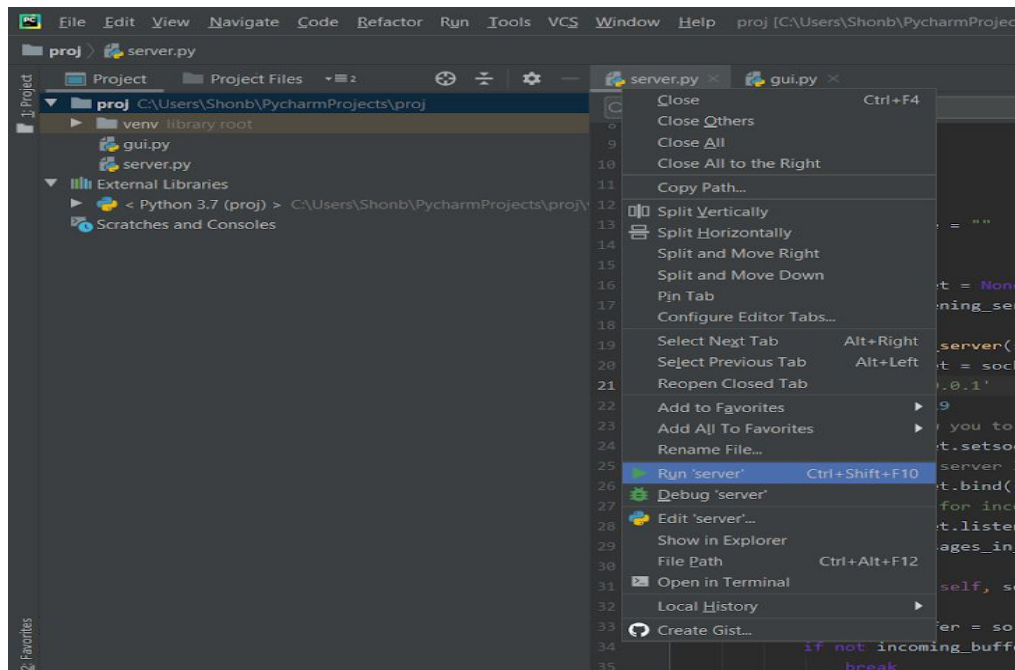
Run > Edit Configurations > **Allow parallel run**



שון באזוב - Multithreaded Chat

לצורך כך שהתוכנית תעבוד, צריך להריץ קודם את התוכנית של השרת, על מנת להפעיל אותו ולגרום לו להאזין ללקוחות שנכנסים:

server	12/06/2020 21:57	Python File	3 KB
--------	------------------	-------------	------



לאחר הרצת השרת, השרת מתחיל להאזין ומחכה ללקוחות שיתחברו אליו, והוא יספק להם את מה שהוא יודע, כלומר מה שכתוב בתוכנית וכך זה נראה:

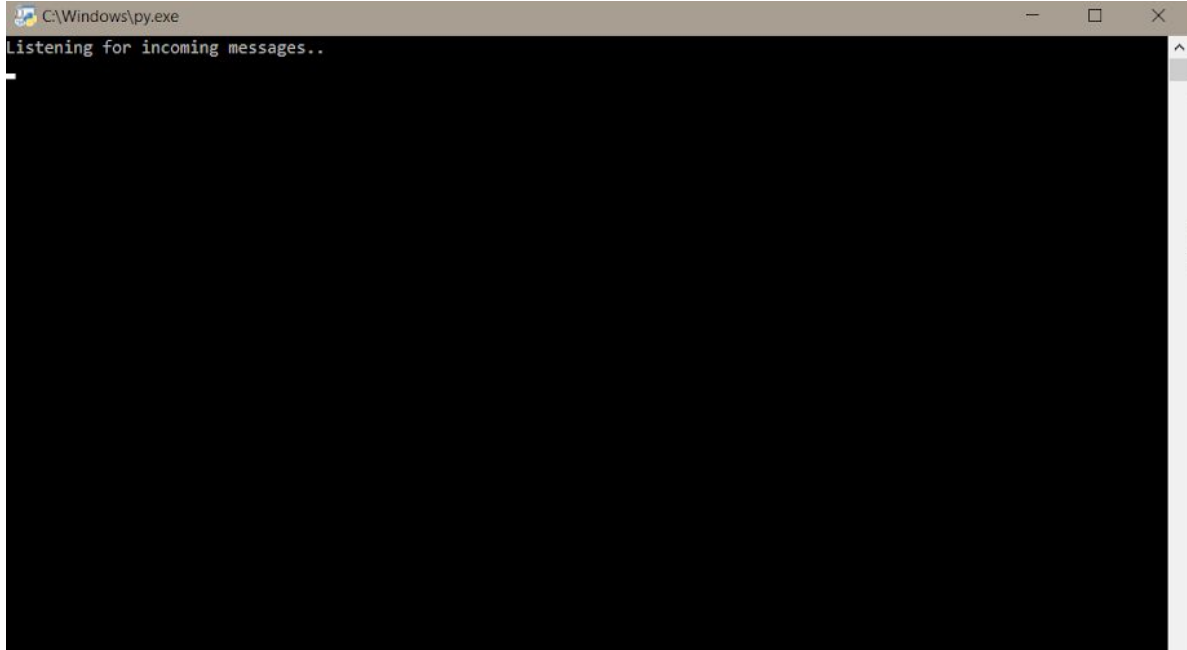
```
C:\Users\Shonb\PycharmProjects\proj\venv\Scripts\python.exe C:/Users/Shonb/PycharmProjects/proj/server.py
Listening for incoming messages..
```



קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
סמל מוסד 441287



שון באזוב - Multithreaded Chat



אחרי שהצלחנו להפעיל את השרת, אפשר להתחיל להריץ את הלקוח, בדיוק כמו שמפעילים את השרת, רק במקום להריץ את התוכנית server.py, מריצים את התוכנית שנמצאת ביחד איתה בתיקייה, את client.py (שבעבר היה נקרא gui.py כפי שרואים בתמונות למעלה, אך השתנה למטרת נוחות). ניתן גם להריץ כמה לקוחות במקביל משום שנפתחים פורטים שונים בכל לקוח בתוך המחשב, לכן אפשר תאורטית עד 65535 לקוחות במקביל (מספר הפורטים במחשב לפרוטוקול TCP), באותו המחשב. לאחר ההרצה של התוכנית, יפתח ממשק גרפי הממחיש את נראות הצ'אט:

venv	10/06/2020 11:00	File folder	
client	13/06/2020 16:22	Python File	6 KB



קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
סגל מוסד 441287



שון באזוב - Multithreaded Chat

Socket Chat

Chat Box:

Enter your name :

Enter message:

Join

Play

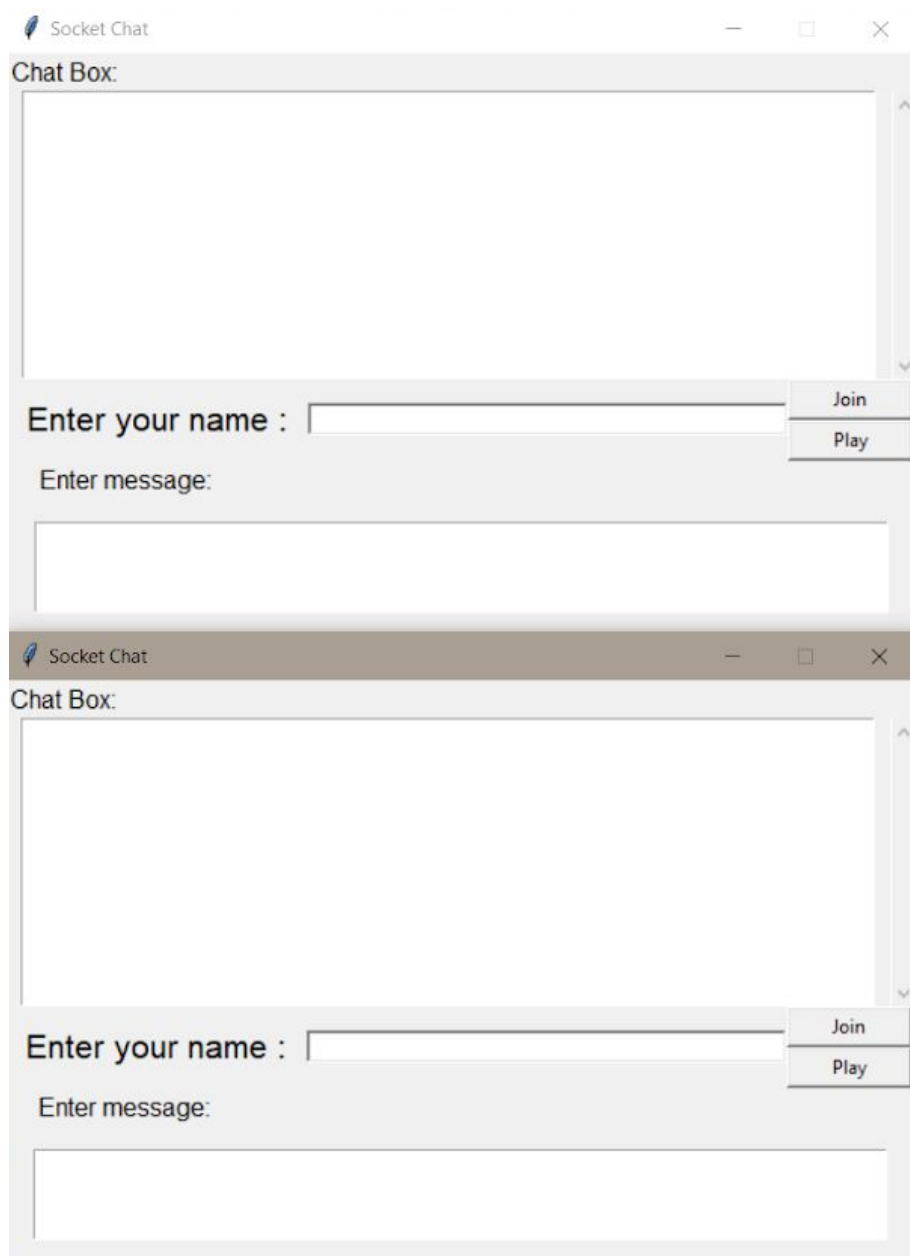


קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
סמל מוסד 441287



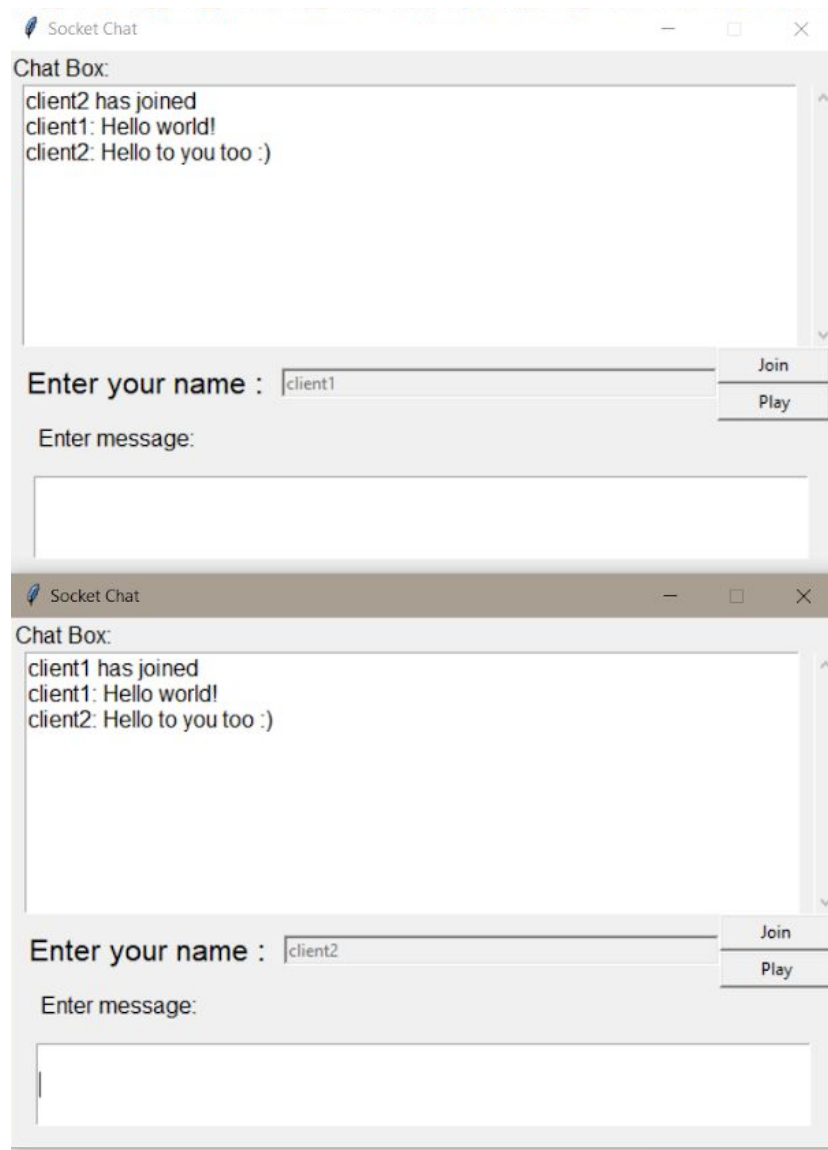
שון באזוב - Multithreaded Chat

פתיחת כמה לקוחות במקביל:

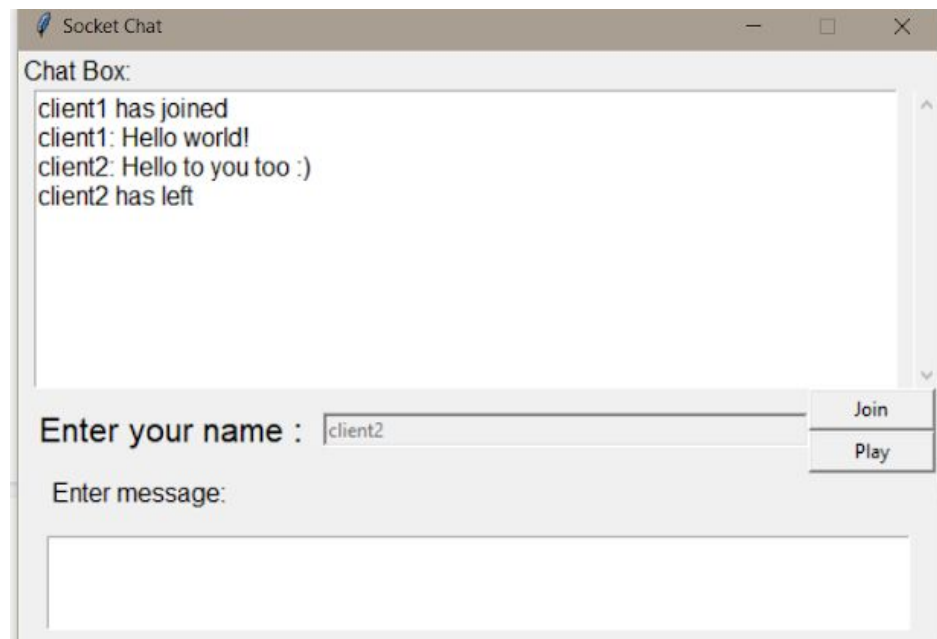
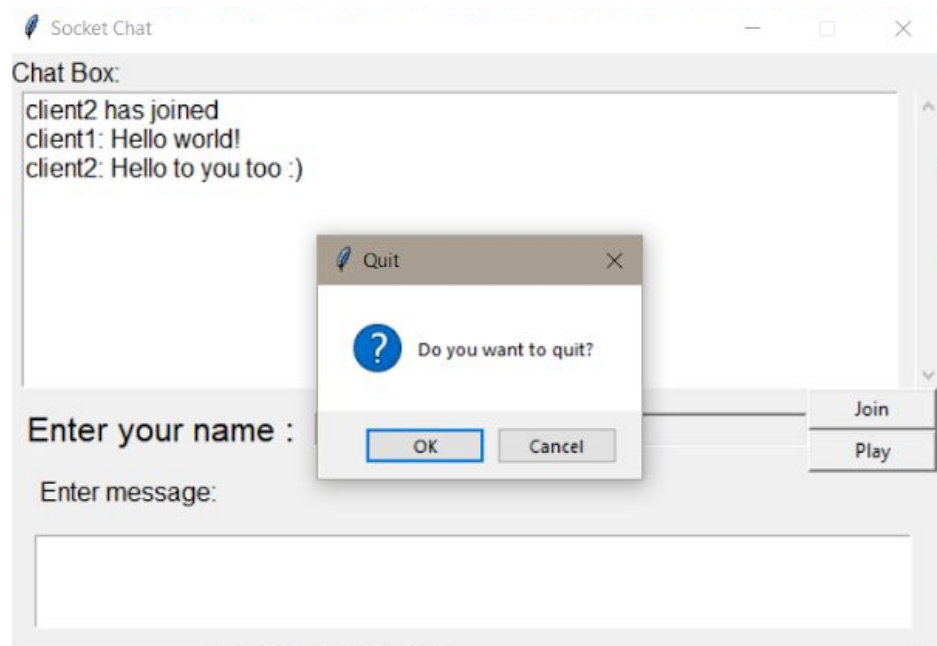


שון באזוב - Multithreaded Chat

לאחר שהפעלנו את הלקוחות והצ'אט נפתח, כל לקוח מתבקש לכתוב מהו שמו, ולאחר שכותב את שמו הוא יכול להצטרף לשיחת צ'אט ע"י לחיצה עם העכבר על כפתור ה"Join". לאחר שמצטרף לצ'אט, מופיע שמו כדי שהלקוחות האחרים יוכלו לראות מי הלקוח הנוסף שנכנס. כל לקוח ששולח הודעה הודעתו מוצגת עם שם ומהי הודעתו, וכדי לשלוח את ההודעה המשתמש צריך לכתוב את מה שהוא רוצה בתיבת טקסט שנמצאת מתחת למשפט "Enter message" ולאחר לחיצה על כפתור ה ENTER במקלדת ההודעה תישלח לשרת וכך גם לשאר המשתמשים שבצ'אט. ניתן לצאת מהצ'אט על-ידי לחיצה על כפתור האיקס והלקוח נשאל האם הוא באמת רוצה לצאת:

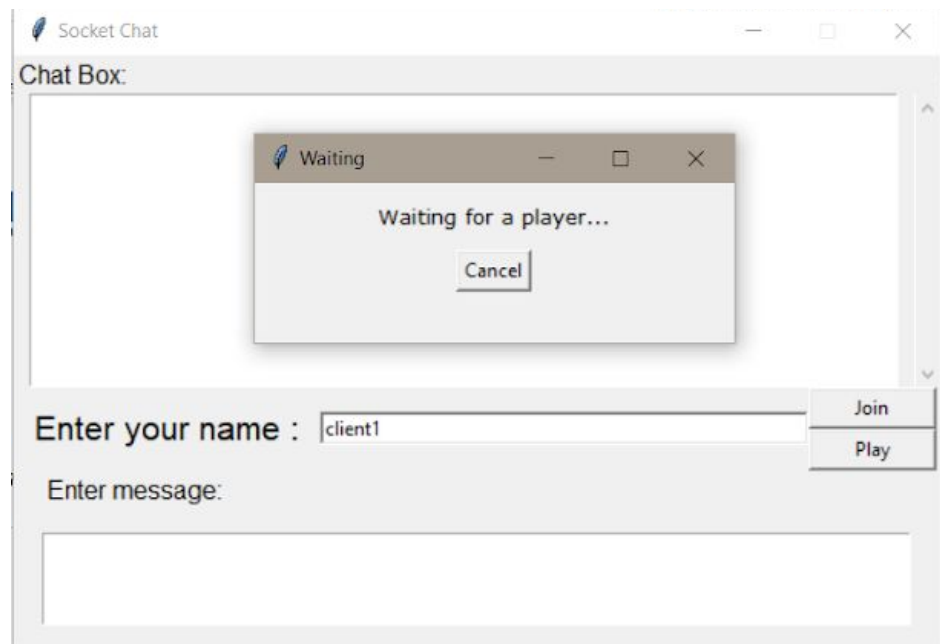


לאחר לחיצה על `ok` המשתמש יוצא מהצ'אט והחיבור בין השרת ללקוח מתנתק. במקרה ויש 2 משתמשים או יותר אז לאחר שמשתמש כלשהו יוצא הצ'אט מראה מי המשתמש שיצא לכל המשתמשים האחרים.



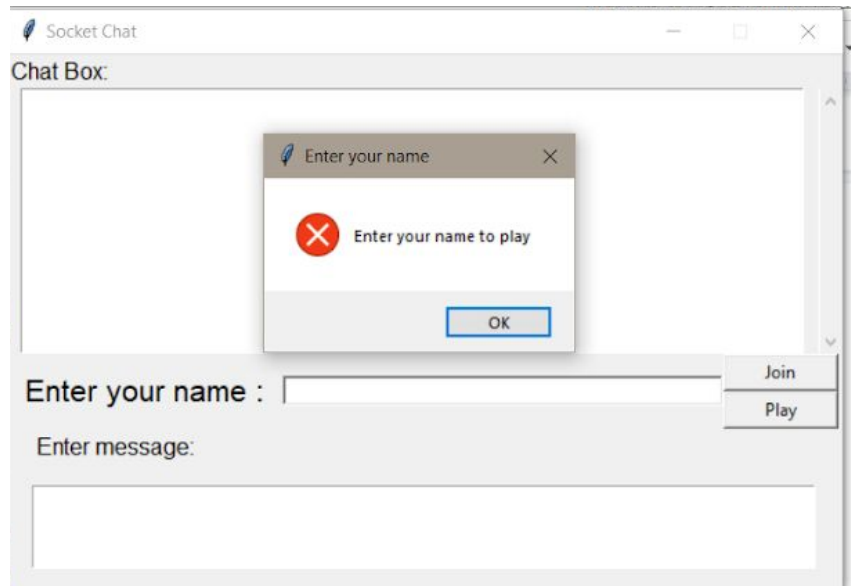
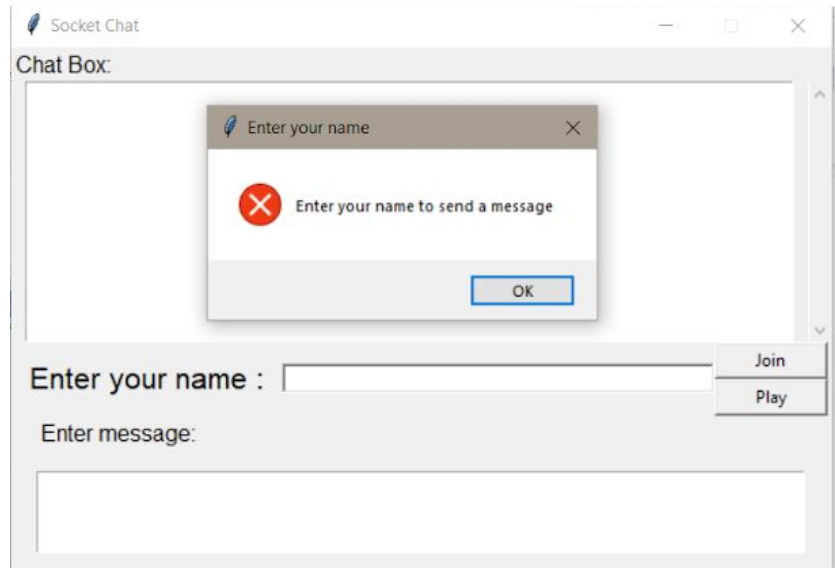
שון באזוב - Multithreaded Chat

במשחק הצ'אט יש כפתור עליו כתוב "Play" ולאחר לחיצה על הכפתור המשתמש יאלץ לחכות עד שמשתמש אחר גם ילחץ על כפתור Play והמערכת תזהה שיש 2 משתמשים שרוצים לשחק ביחד אחד מול השני. במקרה והמשתמש לא רוצה להמשיך לחכות עד שמשתמש אחר יכנס לשחק איתו, או שאף אחד לא לוחץ על כפתור Play, המשתמש יכול ללחוץ על כפתור Cancel ולהפסיק את החיפוש לאחר משתמש אחר לשחק נגדו.



שון באזוב - Multithreaded Chat

במקרה והמשתמש לא הזין את שם, והוא לוחץ על Join או על Play הוא מקבל הודעה מתאימה שהוא לא יכול להמשיך כל עוד הוא לא הזין את שמו.



לאחר שמתחברים 2 שחקנים והסרבר מזהה ששניהם לחצו על כפתור הPLAY הוא יעביר אותם לחלון המשחק, והמשחק מתחיל אחד נגד השני.



קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
סגל מוסד 441287



שון באזוב - Multithreaded Chat

מדריך למפתח

התיקיה עליה אני עובד נקראת proj, זוהי תיקייה שיצרתי בpycharm ושמרתי אותה. ה-PATH של התיקיה - C:\Users\Shonb\PycharmProjects

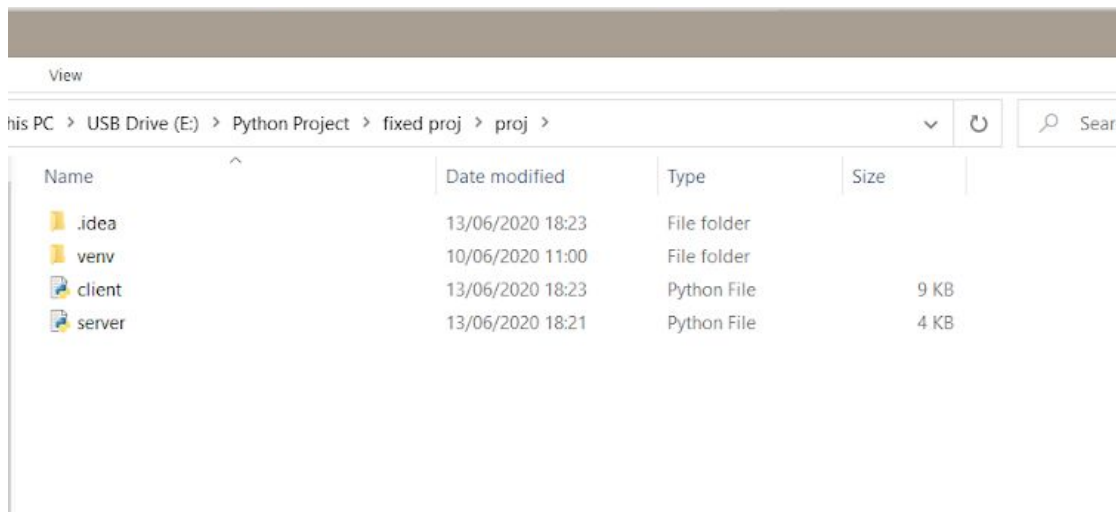
או

E:\Python Project\fixed proj\proj

התיקיה מכילה 2 קבצים עיקריים -

קובץ הפייתון שמכיל את התוכנית של השרת - server.py

קובץ הפייתון שמכיל את התוכנית של הלקוח עם הממשק הגרפי - client.py



בפרויקט שלי השתמשתי בהרבה מחלקות, כדי להקל על הכתיבה, וכי מומלץ להשתמש במחלקה שמכילה כבר דברים שימושיים שאני צריך בתוכנית שלי, במקום לכתוב אותם בעצמי.



קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
סמל מוסד 441287



שון באזוב - Multithreaded Chat

בקובץ השרת server.py השתמשתי במחלקות הבאות -

```
"""  
Server for multithreaded chat application. for any new client that the server gets, he's asked  
for his name, and shows everyone who joined.  
(This is a prototype)  
"""  
  
from socket import *  
import threading
```

המחלקה הראשונה היא מחלקת socketn שהיא מחלקה בסיסית כדי לעבוד עם תקשורת בין שרת ולקוח בנושא הרשתות.

המחלקה השניה היא מחלקת threadingn שהיא שימושית כדי לאפשר להריץ מספר תהליכונים במקביל במרחב כתובות אחד. כל תהליך חדש מתחיל את ביצועו באמצעות "תהליכון ראשי" אשר עשוי במהלך ליצור תהליכונים נוספים ולהשתמש בהם וכך מאפשר לספק לי רצף של פעולות כאשר התהליך מבצע כמה משימות במקביל.

בקובץ השרת client.py השתמשתי במחלקות הבאות -

שון באזוב - Multithreaded Chat

```
user
1 from tkinter import Tk, Frame, Scrollbar, Label, END, Entry, Text, VERTICAL, Button
2 import socket
3 import threading
4 from tkinter import messagebox
5
```

המחלקה הראשונה היא מחלקת tkinter, וזוהי המחלקה שמאפשרת לי להכניס לפרויקט שלי את הממשק הגרפי, שעליו פועלת כל התוכנית.

המחלקה הראשונה היא מחלקת socket שהיא מחלקה בסיסית כדי לעבוד עם תקשורת בין שרת ולקוח בנושא הרשתות.

המחלקה השניה היא מחלקת threading שהיא שימושית כדי לאפשר להריץ מספר תהליכונים במקביל במרחב כתובות אחד. כל תהליך חדש מתחיל את ביצועו באמצעות "תהליכון ראשי" אשר עשוי במהלך ליצור תהליכונים נוספים ולהשתמש בהם וכך מאפשר לספק לי רצף של פעולות כאשר התהליך מבצע כמה משימות במקביל.

המחלקה הרביעית היא אותה מחלקה כמו הראשונה, רק שמתוך מחלקת tkinter אני ייבאתי את השימוש במessagebox שיאפשר לי להכניס לממשק הגרפי דברים שימושיים לצורך נראות הצ'אט.

תוכן הקוד בקובץ client.py -

```
class GUI:
    """
    This class initializes the GUI for the client, and creates the socket between the client and the server.
    """

    def __init__(self, master):
        """
        The constructor of the GUI class
        :param master: the root of the GUI
        """
```



קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
סגל מוסד 441287



שון באזוב - Multithreaded Chat

```
def initialize_socket(self):
```

```
    """
```

```
    This function creates the socket and connects the client to the server.
```

```
    :return: None
```

```
    """
```

```
def initialize_gui(self):
```

```
    """
```

```
    This function starts initializing the Graphical User Interface of the program-  
    creating the window the program is based on.
```

```
    :return: None
```

```
    """
```

```
def listen_for_incoming_messages_in_a_thread(self):
```

```
    """
```

```
    This function opens a thread that always listens to incoming messages  
    and works all the time.
```

```
    :return: None
```

```
    """
```

```
def receive_message_from_server(self, so):
```

```
    """
```

```
    This function is always waiting for messages to receive in an infinite loop  
    and once it receives a message it functions accordingly.
```

```
    :param so: the socket between the client and the server
```

```
    :return: None
```

```
    """
```

```
def display_name_section(self):
```

```
    """
```

```
    This function displays on the window in a frame the label and the Entry to write  
    the name of the client, and displays the JOIN and PLAY buttons to use.
```

```
    :return: None
```

```
    """
```

```
def start_game(self):
```

```
    """
```

```
    This functions starts the game between the 2 clients who entered.
```

```
    :return: None
```

```
    """
```



קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
מגל מוסד 441287



שון באזוב - Multithreaded Chat

```
def popupmsg(self):
```

```
    """
```

```
    This function pops up a window that shows that the server is waiting for  
    another player to press PLAY and play against the client.  
    the client can wait, or it can cancel the search.
```

```
    :return: None
```

```
    """
```

```
def display_chat_box(self):
```

```
    """
```

```
    This function displays the chat box - the chat box is where all the messages  
    are written such as - who joined, what messages are displayed, who left and who  
    is searching for a player to play with.
```

```
    :return: None
```

```
    """
```

```
def display_chat_entry_box(self):
```

```
    """
```

```
    This function displays the chat entry box. The chat entry box is where  
    the user can write what he wants and send his message to the other users  
    connected to the server.
```

```
    :return: None
```

```
    """
```

```
def on_join(self):
```

```
    """
```

```
    This function checks if the user has entered a name and acts accordingly, if the  
    user entered a name the function sends it to the server to display it on the  
    chat box.
```

```
    :return: None
```

```
    """
```

```
def on_enter_key_pressed(self, event):
```

```
    """
```

```
    This function checks if the user entered a name and sends the message  
    the user has entered after pressing the ENTER key.
```

```
    :param event: ENTER key
```

```
    :return: None
```

```
    """
```



קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
סגל מוסד 441287



שון באזוב - Multithreaded Chat

```
def clear_text(self):  
    """  
    This function allows the option of deleting the message the user is  
    writing during his writing.  
    :return: None  
    """
```

```
def send_chat(self):  
    """  
    This function is responsible for sending the user data to the  
    server.  
    :return: 'break' (str)  
    """
```

```
def on_close_window(self):  
    """  
    This function is responsible for the user quitting the chat and making sure he  
    wants it. After the user quits the chat the function alerts the server by sending  
    a message saying the user has left and later it gets displayed on the chat box.  
    :return: None  
    """
```

תוכן הקוד בקובץ server.py -

```
class ChatServer:  
    """  
    This class activates the server and creates a listening socket so clients can connect.  
    """
```

```
def __init__(self):  
    """  
    The constructor of the server class.  
    """
```

```
def create_listening_server(self):  
    """  
    This function creates the listening socket of the server, making the server
```

```
wait for incoming connections.  
:return:  
""  
  
def receive_messages(self, so):  
    """  
    This function is always waiting for messages and receiving them, and after  
    receiving the messages, calls a function that broadcasts the messages to all  
    the clients.  
    :param so: The socket between the server and the current client.  
    :return: None  
    """  
  
def broadcast_to_all_clients(self, senders_socket):  
    """  
    This function broadcasts the message it was given to all the clients  
    connected to the server.  
    :param senders_socket: The socket of the client who sent the message.  
    :return: None  
    """  
  
def receive_messages_in_a_new_thread(self):  
    """  
    This function allows the server to always accept new connections from new  
    clients and makes sure the server is always receiving messages, by calling a  
    thread.  
    :return: None  
    """  
  
def add_to_clients_list(self, client):  
    """  
    This function adds the current client to the clients list, that includes all the clients  
    connected to the server.  
    :param client: The new client who connected to the server.  
    :return: None  
    """
```

הסבר על כל משתנה בקוד הקובץ client.py -

```
client_socket = None #The socket between the client and the server.
```



קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
מגל מוסד 441287



שון באזוב - Multithreaded Chat

```
last_received_message = None #Contains the last received message.
```

```
remote_ip = '127.0.0.1' #IP the client is connecting to
```

```
remote_port = 10319 #Port used
```

```
buffer = so.recv(1024) #contains the data sent from the server.
```

```
frame = Frame() #Creates a frame on the GUI for specific tasks
```

```
scrollbar = Scrollbar(frame, command=self.chat_transcript_area.yview, orient=VERTICAL) #creates a scrollbar on the chat to scroll through messages
```

```
senders_name = self.name_widget.get().strip() + ": " #name of the client who sent the message
```

```
data = self.enter_text_widget.get(1.0, 'end').strip() #the data that was sent to the client from another client
```

```
message = (senders_name + data).encode('utf-8') #name of the client who sent the message and the message itself
```

```
self.root = master #the root of the GUI
```

```
self.chat_transcript_area = None #displays the contents of the chat messages that are exchanged.
```

```
self.name_widget = None #Contains the name of the user
```

```
self.enter_text_widget = None #multiline texting area.
```

```
self.join_button = None #This parameter is responsible for the JOIN button and will include all it's data
```

```
self.play_button = None #This parameter is responsible for the PLAY button and will include all it's data
```

```
root = Tk() #Initializes the tkinter GUI
```

```
gui = GUI(root) #An object from the GUI class
```

הסבר על כל משתנה בקוד הקובץ server.py -

```
clients_list = [] #list that includes all the clients connected to the server.
```

```
last_received_message = "" #contains the last receive message in a string format.
```

```
self.server_socket = None #The socket of the server that connects to the clients.
```

```
local_ip = '127.0.0.1' #Local IP
```

```
local_port = 10319 #Local used Port
```

```
incoming_buffer = so.recv(1024) #contains the data send from the client to the server
```



קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
מכל מוסד 441287



שון באזוב - Multithreaded Chat

סיכום אישי / רפלקציה

העבודה על הפרויקט הייתה עבורי חוויה חדשה שחיכיתי לה כבר מאז שהגעתי למגמה בכיתה י', היא הייתה מאוד שונה, כרוכה בהרבה למידה עצמית, ודרשה מוטיבציה כדי להמשיך לעבוד עליה. העבודה על הפרויקט כן הייתה מסובכת וקשה, אך בעיקר בגלל כמות הזמן והחשיבה שנדרשו כדי להביע את הרעיון לפרויקט, בתוך הפרויקט עצמו. בנוסף לכך, בעיה נוספת שעלתה לי, והיא אחת הבעיות המרכזיות, היא מה אני אעשה בפרויקט, כלומר כיצד אני רוצה שהתוצר הסופי יראה, משום שהיו לי מלא רעיונות בראש אך היה לי קשה לבחור אחד.

בזמן העבודה על הפרויקט רכשתי המון כלים, אך הכלי המרכזי והידוע ביותר הוא כלי הלמידה העצמית, או שאפשר גם להגיד, כלי הגוגל, משום שהגוגל היה עבורי מקור וכלי ללמידה העצמית. בנוסף ללמידה העצמית, למדתי גם איך להתמודד עם קשיים ואתגרים לבד, משום שאחרי הכל הפרויקט הוא שלי וכדי להבין אם יש בעיה אני אצטרך למצוא לה פתרון בעצמי. כלי נוסף שאקח מהפרויקט הזה הוא כלי הלימוד והלמידה, כלומר, בשיעורי סייבר שהתנהלו בכיתות נעזרו אחד בשני ועזרנו אחד לשני בכתיבת הפרויקטים, אם זה להביא רעיונות לפרוייקט, להבין מה הלוגיקה והאלגוריתם שצריך כדי להביע לקוד, להשתמש בקוד שחבריי לכיתה עשו שיכול לעזור לי בפרויקט וגם ההפך - אלה הם שני הכלים שאני אקח איתי לעתיד, הלמידה העצמית, והעבודת צוות. קושי שעלה לי בזמן עבודה על הפרוייקט הוא קושי די ייחודי, וזהו קושי הירידה במוטיבציה. קושי זה נבע בעיקר בגלל תחילת תקופת מגפת הקורונה, שבהתחלה הייתה תקופה מאוד מבלבלת, ולא ידענו מה הולך לקרות עם הבגרויות והציונים, וגם הפרויקט נכלל בבעיה הזו. אז לאחר חודשים של עבודה על הפרוייקט, הגעתי למצב שאני לא עובד על הפרויקט כמה שבועות עקב חוסר המוטיבציה והבלבול הרב, והייתי צריך למצוא שוב את המוטיבציה להמשיך. קושי נוסף שעלה הוא קושי של עמידה בלוחות זמנים, שוב, עקב הקורונה, אנחנו חיים במציאות שהשתנה כל יום, כך גם המציאות של המקצועות אותן למדנו במשך השנה השתנו, ועם זה גם מקצוע הסייבר, ולא ידענו איזה זמן להשקיע במה. המסקנות שלי מהפרויקט הוא שכדי להגיע לפרויקט טוב, צריך להשקיע את הזמן, המאמץ וגם לאהוב את מה שאתה עושה, אחרת התוצאה לא תהיה בדיוק כיפית.



קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
מגל מוסד 441287



שון באזוב - Multithreaded Chat

אם הייתי מתחיל את הפרוייקט היום הייתי פועל בצורה אחרת, הייתי משתמש בידע שיש לי עכשיו לעומת תחילת השנה כדי לגרום לפרוייקט להיות יעיל יותר, מאורגן יותר ויפה יותר ויזואלית בגלל הניסיון והחשיפה לנושאים חדשים בתחום. לולא מגפת הקורונה, הייתי עובד לדעתי בצורה הרבה

יותר טובה, בעיקר שלא הייתי מאבד את המוטיבציה לעבוד, וכן הייתה יותר עזרה מצד המורה וכך והייתי ממשיך לעבוד על הפרוייקט כל יום וכך גם מקדם אותו למימדים שאני לא יכולתי לחשוב שאני יכול להגיע אליהם.

בנוסף, אני כן מרוצה מהתוצאות, אך לא מסופק מהן, כן הגעתי למה שרציתי אבל אני חושב שיכולתי להגיע ליותר, בעיקר אם לא היינו נפגעים לימודית ממגפת הקורונה. אבל אני יודע שזה התחום שאני אוהב, לכן עם כל הזמן שיש לי אני יכול לעבוד על עוד פרוייקטים או אפילו לקדם את הפרוייקט הזה.



קרית החינוך השש שנתית
ע"ש חיים בר-לב
מקיף ח
סגל מוסד 441287



שון באזוב - Multithreaded Chat

ביבליוגרפיה

Github. (2007, October). Retrieved from <https://github.com/>

Newest 'multithreading' Questions. (2008, August). Retrieved from <https://stackoverflow.com/questions/tagged/multithreading>

Tkinter (GUI Programming). (2020). Retrieved from <https://pythonbasics.org/tkinter/>

ProTech. (n.d.). Python Fundamentals Tutorial: Code Organization. Retrieved from https://www.protechtraining.com/content/python_fundamentals_tutorial-code_organization

Lundh, F. (2009, August). Events and Bindings. Retrieved from [effbot.org: https://effbot.org/tkinterbook/tkinter-events-and-bindings.htm](https://effbot.org/tkinterbook/tkinter-events-and-bindings.htm)

Shutzman, J. (2010). School Search - Jacob Shutzman. Retrieved from <http://www.lamed-oti.com/school/rl/>

Simple Chat Room using Python. (2017, April 21). Retrieved from <https://www.geeksforgeeks.org/simple-chat-room-using-python/>