

פרויקט תכנות – רשתות מחשבים ואינטרנט 1

מגישים: יהל אורגד 325010809

מיכאל דוד 212679567

מערכת הפעלה: windows 10

שפת תכנות: python גרסה: python3.10

רקע תיאורטי:

Socket: הוא מספק ממשק תקשורת בין שכבת הרשת לשכבת האפליקציה (חבילות משכבת הרשת עוברות דרך socket לשכבת האפליקציה).

הוא מוגדר ע"י כתובת ip וע"י port.

כתובת ip: כל רכיב בשכבת הרשת (מחשב וכו') מיוצג ע"י הכתובת הזאת.

היא עוזרת לנו לנתב דברים אל מיקומם הנכון (כתובת זו יכולה להשתנות בזמן ובמיקום).

Port: הוא תהליך ספציפי שדרכו יכולות תוכנות להעביר נתונים באופן ישיר, במקום אמצעים אחרים כגון העברת קובצי נתונים.

השימוש הנפוץ ביותר בפורט הוא במסגרת הפרוטוקולים הנפוצים בשכבת התעבורה TCP ו-U: UDP. פורט מזהה לכל כתובת או פרוטוקול מסוים על ידי מספר באורך 16 ביטים היוצר 65536 כתובות אפשריות ל-U-65535 כתובות אפשריות ל-TCP. כתובת זו נקראת "מספר הפורט".

TCP: הוא פרוטוקול בתקשורת נתונים הפועל בשכבות התעבורה של מודל ה-OSI ובמודל ה-TCP/IP, ומבטיח העברה אמינה של נתונים בין שתי תחנות ברשת מחשבים באמצעות יצירת חיבור מקושר (Connection Oriented).

כאשר הוא משמש כחלק מחבילת הפרוטוקולים TCP/IP עושה הפרוטוקול שימוש בפרוטוקול ה-IP ולצורך העברת הנתונים TCP. מעביר את הנתונים שהועברו באמצעות IP, מוודא את נכונותם, ומאשר את קבלת הנתונים במלואם או מבקש שליחה מחדש של נתונים שלא הגיעו בצורה תקינה.

UDP: הוא פרוטוקול השייך לשכבת התעבורה של מודל ה-OSI ולשכבת התעבורה של מודל ה-TCP/IP המאפשר העברת נתונים לא אמינה (חבילות מידע).

UDP אינו מספק אמינות או שימור סדר כפי שקורה ב-TCP. חבילות מידע עשויות להגיע בסדר שונה מזה שבו הן נשלחו, להגיע מספר פעמים או ללכת לאיבוד ולא להגיע כלל.

חסכון המידע המבטיח את אמינות השליחה ועובדת היותו של הפרוטוקול "connectionless" (אינו מייסד קשר בדומה לפרוטוקול TCP), גורמים לפרוטוקול UDP לספק דיוור מהיר לחבילות הנשלחות דרכו. המהירות היחסית של הפרוטוקול הופכת אותו מתאים ביותר לאפליקציות שאינן דורשות אמינות מלאה של המידע (בין אם הן לא זקוקות לו או מממשות זאת בעצמן), לאפליקציות עבורן יש חשיבות גבוהה לזמן דיוור קצר (לדוגמה DNS) ולאפליקציות בהן שני הפרמטרים מתאימים (לדוגמה VoIP בה עיכוב מוריד מאיכות השיחה ולעומת זאת מידע שלא הגיע לא ישודר שוב).

הסבר לקוד:

פונקציות:

put in board(board game, col, player):

הפונקציה הזאת מקבלת את המצב של לוח המשחק , את איזה שחקן צריך לשים (מחשב או משתמש) ואת העמודה שהוא רוצה לשים בה.

היא דואגת לכך שהיא שמה אותו בעמודה הנכונה + דואגת לכך שאם יש מישהו בעמודה הזאת הוא ישים מעליו ולא עליו.

היא גם דואגת לכך שאם אי אפשר לשים באותה עמודה (העמודה מלאה) המשתמש יקבל הודעה שהוא שם בשורה שהוא לא יכול וישים מחדש .

smarter column choose(board game):

פונקציה זו היא למשתמש שמעדיף לשחק ברמה הקשה ולא בקלה .

היא מקבלת את מצב הלוח וע"י סריקה שלו עושה מהלך חכם יותר מאשר סתם לעשות random. לדוג אם בעמודה 1,2,3 שורה 0 יש אסימונים של המשתמש (1) אזי אם יש בעמודה 0/4 מקום אז הוא ישים שם וינצח את המשחק.

isBoardFull(board game):

פונקציה זו מקבלת את הלוח ובודקת אם עוד יש בו מקום ותחזיר בהתאם האם אפשר לשחק.

Board to string(board game): +print board(board game):

פונקציות שמקבלות את הלוח .

Board to string לוקחת את הלוח והופכת אותו לסוג string כדי שנוכל לשלוח אותו כמו שצריך בין הסרבר לקליינט .

print board פשוט מציירת את הלוח כדי שהמשתמש /הסרבר יראה את המצב .

do someone won(board game):

הפונקציה מקבלת את הלוח ועוברת עליו כדי לבדוק האם יש 4 מאותו סוג באותה שורה / עמודה / אלכסון (/ אלכסון הפוך) .

הפונקציה פועלת אחרי כל תור ובודקת אם מישהו ניצח ומחזירה פלט בהתאם.

Get run num(connection):

הפונקציה שדואגת לבדיקה של כמה סיבובים/ כמה ניצחונות צריך כדי לסיים את המשחק הכללי.

היא מתחילה מלשלוח למשתמש הודעה של כמה סיבובים הוא רוצה ומודיע לו שאם הוא עושה יותר מידי טעויות הוא יושעה לזמן כלשהו (על כל 5 טעויות יושעה לדקה ואם ימשיך אז כל פעם הזמן יוכפל).

ניתן לראות בקוד איך בעזרת count2 אנו מחליטים לכמה זמן הוא יושעה (בעזרת time.sleep) ושי count1 עוזר לנו להחליט מתי צריך להוסיף עוד זמן של השעיה.

Threaded client(client):

בעצם זה ה"main" שלנו. כל התקשורת בין הסרבר לקליינט קוראת בעזרת זה .

הוא מתחיל מלשאל את המשתמש האם הוא רוצה לצאת או לשחק ואז לפי התגובה שהוא מקבל הוא בוחר האם לסיים או להמשיך. אם בוחר להמשיך: הוא שואל האם ללכת על מצב קשה או מצב קל . ע"פ הבחירה שלו בסוף הסרבר ידע אם להשתמש ב-smarter column choose או לא.

לאחר בחירה זו מפעילים את Get run num שדרכו הסרבר ידע כמה משחקים לעשות .

נקבל ממנו את מספר המשחקים ואז הסרבר יתחיל בלולאה שתיגמר כשמספר הניצחונות יגיע למספר שקיבלנו מהמשתמש.

ואז יתחיל המשחק : הסרבר יוצר לוח מלא באפסים ושואל את המשתמש באיזה עמודה לשים. הוא שם אותה בעזרת put in board ואז הסרבר שם את האסימון שלו לפי בחירת הרמה של המשתמש) שם עם רנדום ב-put in board אם בחרו ברמה הקלה או שם עם smarter column choose אם בחרו ברמה הקשה) .

בכל תור מריצים את do someone won ובודקים אם מישהו ניצח או לא . אם מישהו ניצח יודיעו על כמות הניצחונות שלו ועל פרטים נוספים כמו מספר הסיבובים ויתחילו משחק חדש עם לוח חדש וכו'...

אחרת ממשיכים לתור הבא...

(כמובן יש גם את המקרה של תיקו)

כשיגמרו כל המשחקים המשתמש יקבל הודעת סיכום של המשחקים והסרבר ינתק אותו מהתקשורת.

המדדים המוצגים בסוף כל סיבוב ומשחק הם:

מספר הניצחונות של כל אחד מהשחקנים, מספר המהלכים בסיבוב

ומספר הבחירות השגויות (עמודה מלאה, עמודה לא בטווח) בסיבוב .

בסוף המשחק יש את אותם מדדים רק שהספירה היא לכל המשחק.

אצל המשתמש יש פשוט לולאה שמקבל הודעות מהסרבר ומחזירה על פי מה שהמשתמש רוצה.

הסבר על פתיחת Socket:

יצרנו חיבור סוקט ע"י הפקודה `socket.socket()` והגדרנו את `ip` ו`port` (אמרנו ברקע תיאורטי שע"י הוא מוגדר).

לאחר מכן עשינו `bind` וחיבינו ל `connection`. הגדרנו שיוכלו להתחבר עד 5 אנשים לרשת:

```
try:
    ServerSocket.bind((host, port))
except socket.error as e:
    print(str(e))

print('Waiting for a Connection..')
ServerSocket.listen(5)
```

הפקודה `bind` מקשרת בין הסרבר `host` בפתיחת הסרבר.

לאחר מכן קיבלנו את בקשת הקליינט להתחבר בלולאה אינסופית

```
while True:
    Client, address = ServerSocket.accept()
    print('Connected to: ' + address[0] + ':' + str(address[1]))
    start_new_thread(threaded_client, (Client, ))
    ThreadCount += 1
    print('Thread Number: ' + str(ThreadCount))
ServerSocket.close()
```

כדי שיהיה מקסימום 5 משתמשים.

(בעצם כאשר מגיע משתמש חדש יוצרים לו `thread` ומריצים לו את המשחק במקביל לשאר המשתמשים).

ניתן לראות את התחברות הקליינט לסרבר בקוד הבא:

```
try:
    ClientSocket.connect((host, port))
except socket.error as e:
    print(str(e))
```

ניתן לראות את `connect` שבצעם יוצר את החיבור.

בנוסף ניתן לראות אילך השתמש קולט הודעות מהסרבר בקוד הבא:

```
while True:
    msg = ClientSocket.recv(1024)
    print(msg.decode())
```

כמובן שאנו רוצים חיבור אמין ומשתמשים בפרוטוקול TCP.

תיעוד מהלך המשחק:

אם הסרבר פועל ואין יותר מידי משתמשים באותו הזמן אז כאשר client רץ הוא יקבל הודעה :

```
Server Says: choose 1 to quit ,choose 2 to play against AI
```

והמשתמש צריך לבחור אם לצאת או להמשיך . אם הוא לוחץ 1 הוא מתנתק

אחרת הוא יקבל את השאלה הבאה:

```
Server Says: choose 1 to quit ,choose 2 to play against AI
```

2

```
CHOOSE 1 to play in EASY mode and CHOOSE 2 to play in HARD mode-->
```

שלפיה המשתמש בוחר את רמת המשחק .

נדגים כאן קודם מה קורה כאשר הוא רוצה רמה קלה, ולאחר מכן מה קורה ברמה קשה.

רמה קלה :

המשתמש מקבל את ההודעה הבאה

1

```
Please choose the number of rounds required to win the game
,if you will choose inappropriate value 5 times in a row
you will get suspended for one minute
and for more 5 times for 2 minutes and so on...
```

שבעצם הוא בוחר כמה משחקים הוא רוצה ומקבל הזהרה לא לעשות דברים לא חוקיים .

נראה שהוא באמת מושעה על דברים לא חוקיים:

```
and for more 5 times for 2 minutes and so on...
-1
You chose a bad value for rounds number to win. TRY AGAIN NOW:
-1
You chose a bad value for rounds number to win. TRY AGAIN NOW:
-1
You chose a bad value for rounds number to win. TRY AGAIN NOW:
-1
You chose a bad value for rounds number to win. TRY AGAIN NOW:
-1
You chose a bad value for rounds number to win. TRY AGAIN NOW:
-1
You chose a bad value for rounds number to win FOR 5 TIMES YOU WILL GET SUSPENDED.THEN PRESS ENTER AND THEN YOU WILL BE ABLE TO TRY AGAIN
```

זמן ההשעיה יוכפל אם ימשיך לטעות עוד 5 פעמים וכו'...

כאשר הוא ישים אחרי ההשעיה מספר טוב הוא יקבל את ההודעה הבאה:

```
You chose a bad value for rounds number to win. TRY AGAIN NOW:
```

2

```
You chose to play 2 rounds to win properly, press ENTER to continue
```

וכאשר הוא ילחץ ENTER הוא יקבל את ההודעה הבאה:

```
NOW ROUND NUMBER 1 IS STARTING
CHOOSE a COLUMN between 0-6 by press the number of the column
and then press ENTER to continue and see the AI choose-->:
```

שמבשרת לו לבחור עמודה לשים בה את האסימון שלו : כאשר הוא יבחר וילחץ ENTER הוא יקבל את הלוח :

1

0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

וכאשר הוא ילחץ שוב ENTER הוא יקבל את בחירת הסרבר .

כאן נראה בחירה רנדומלית של הסרבר כי אנו ברמה הקלה :

2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
CHOOSE a COLUMN between 0-6 by press the number of the column and then press ENTER to continue and see the AI choose-->:							

ניתן לראות את בחירת הסרבר ולאחר מכן את ההודעה של להמשיך לבחור עמודה .

נמשיך את המשחק :

1

```

-----
| 2 | 1 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----

```

```

-----
| 2 | 1 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 2 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----

```

CHOOSE a COLUMN between 0-6 by press the number of the column and then press ENTER to continue and see the AI choose-->:

2

```

-----
| 2 | 1 | 1 | 0 | 0 | 0 | 0 |
-----
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 2 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----

```

```

-----
| 2 | 1 | 1 | 0 | 2 | 0 | 0 |
-----
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 2 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
CHOOSE a COLUMN between 0-6 by press the number of the column
and then press ENTER to continue and see the AI choose-->:

```

```

2
-----
| 2 | 1 | 1 | 0 | 2 | 0 | 0 |
-----
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
-----
| 0 | 2 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----

```

```

2
-----
| 2 | 1 | 1 | 0 | 2 | 0 | 2 |
-----
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
-----
| 0 | 2 | 1 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----

```


2

	2		1		1		0
	2		0		2		0
	2		0		2		0
	0		1		1		0
	0		1		0		0
	0		0		0		0
	0		0		0		0
	0		0		0		0
	0		0		0		0
	0		0		0		0
	0		0		0		0

	2		1		1		0
	2		0		2		0
	2		0		2		0
	0		1		1		0
	0		1		0		0
	0		0		0		0
	0		0		0		0
	0		0		0		0
	0		0		0		0
	0		0		0		0
	0		0		0		0

וכעת אחרי שהמשתמש יבחר בעמודה 2 וינצח הוא יקבל את הדברים הבאים:

Round Over, player 1 won player 1 now have 1 winnings and player 2 now have 0 winnings there were 6 moves during this round and all together there were 0 false decisions in this round SEE YOU IN THE NEXT ROUND :)	2
2 1 1 0 2 0 2	2 1 1 0 2 0 2
0 1 1 0 0 0 0	0 1 1 0 0 0 0
0 2 1 0 0 0 0	0 2 1 0 0 0 0
0 2 1 0 0 0 0	0 2 1 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0

ניתן גם לראות פה את המדדים שבחרנו להציג.

ונתחיל מחדש (בחרנו כאן 2 משחקים):

NOW ROUND NUMBER 2 IS STARTING
CHOOSE a COLUMN between 0-6 by press the number of the column
and then press ENTER to continue and see the AI choose-->:

נריץ את המשחק השני קצת מהר יותר ונראה כל סבב ולא כל תור:

0 0 0 2 1 0 0	0 0 0 2 1 0 0
0 0 0 2 1 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0

0 0 2 2 1 0 0	0 0 0 2 1 0 0
0 0 0 2 1 0 0	0 0 0 2 1 0 0
0 0 0 1 1 0 0	0 0 0 0 1 0 0
0 0 0 0 2 0 0	0 0 0 0 2 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0

2 0 2 2 1 0 0	0 0 2 2 1 0 0
0 0 2 2 1 0 0	0 0 2 2 1 0 0
0 0 0 1 1 0 0	0 0 0 1 1 0 0
0 0 0 1 2 0 0	0 0 0 1 2 0 0
0 0 0 1 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0

ובאשר ננצח שוב נקבל את ההודעות הבאות:

3

2 0 2 2 1 0 0
0 0 2 2 1 0 0
0 0 0 1 1 0 0
0 0 0 1 2 0 0
0 0 0 1 0 0 0
0 0 0 1 0 0 0

Round Over, player 1 won
 player 1 now have 2 winnings
 and player 2 now have 0 winnings
 there were 7 moves during this round
 and all together there were 0 false decisions in this round

GAME OVER
 player 1 now have 2 winnings

ובאשר כל הגמל נגמר:

SEE YOU IN THE NEXT GAME :

כעת נראה משחק אחד של האפשרות של מצב קשה :

```
Server Says: choose 1 to quit ,choose 2 to play against AI
2
CHOOSE 1 to play in EASY mode and CHOOSE 2 to play in HARD mode-->
2
Please choose the number of rounds required to win the game
,if you will choose inappropriate value 5 times in a row
you will get suspended for one minute
and for more 5 times for 2 minutes and so on...
1
```

בעצם נתחיל כך.

ונתחיל את המשחק :

```

NOW ROUND NUMBER 1 IS STARTING
CHOOSE a COLUMN between 0-6 by press the number of the column
and then press ENTER to continue and see the AI choose-->:

0
-----
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----

```

[illegible]

TAKI

[illegible]

2

	1		0		2		0		0		2		0	
	1		0		1		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	

	1		0		2		0		0		2		0	
	1		0		0		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	

2

	1		0		2		0		2		2		0	
	1		0		1		0		0		0		0	
	0		0		1		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	

	1		0		2		0		2		2		0	
	1		0		1		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	

ואז נראה שהמחשב חכם יותר ויראה את האופציה לנצח וינצח !!!

	1		0		2		2		2		2		0	
	1		0		1		0		0		0		0	
	0		0		1		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	
	0		0		0		0		0		0		0	

הסיבוב נגמר, מזל טוב!!!