

**מגמת הנדסת תוכנה**

**פרויקט גמר בהגנת סייבר ומערכות הפעלה**

**שם מוסד** – ישיבת בני עקיבא אורות יהודה

**שם תלמיד** – משה בקריצקי

**תעודת זהות** – 337609374

**שם המנחה** – שמעון סוויסה

**תאריך הגשה** – מאי 2025

**שם העבודה** – **SimultaneousTTT**

[גיט האב: https://github.com/MosheStudent/TicTacToe.git](https://github.com/MosheStudent/TicTacToe.git)

תוכן עניינים:

[**מבוא**](#_sbnw6oo9om75) **2**

[**תיאור תחום הידע**](#_legjlxl1y7j0) **4**

[יכולות מצד השרת-](#_qywnz4xi4v9k) 4

[יכולות מצד הלקוח](#_xfyitlvhm53j) 4

[**ארכיטקטורה של הפרויקט**](#_854xybdxtw2s) **6**

[תיאור הארכיטקטורה של המערכת המוצעת:](#_9vwqf4d83pol) 7

[תיאור הטכנולוגיה הרלוונטית](#_s5flb6jv7skn) 7

[תיאור זרימת המידע](#_olcrwo5cjx6f) 7

[תיאור אלגוריתמים מרכזיים בפרויקט](#_bk2aespx4la2) 7

[איך AES עובד? (ADVANCED ENCRYPTION STANDARD)](#_2dcnsz9hgid7) 7

[סביבת העבודה](#_iiudl1a12gwm) 8

[תיאור פרוטוקול התקשורת](#_4c604fi18sf8) 8

[עיקרי הפרוטוקול](#_yctam5sya55p) 8

[**סקירת מסכים (GUI)**](#_l0eihx9seif) **9**

[מסך כניסה:](#_2icczpiy7dp9) 9

[מסך לוח משחק:](#_h2uar7pbu4ef) 10

[מסך סוף משחק:](#_euobmxaf1mny) 10

[**שמירה למחשב**](#_1fnwg4rn10nn) **11**

[**סקירת חולשות ואיומים**](#_ra24haukhfjx) **11**

[תהליך התחברות:](#_rd09uck5m6zb) 11

[מתקפת DOS/DDOS:](#_hn0p4rda36r2) 11

[ניתוח תנועה (TRAFFIC ANALYSIS):](#_caor7cycuhe1) 11

[**מימוש הפרויקט**](#_412n3ahhmdpk) **12**

[סקירת המחלקות והמודולים המיובאים:](#_2q8fb9n35sn4) 12

[socket:](#_9xxsr5m6mydh) 12

[tkinter:](#_1c2q3rntujy6) 12

[threading:](#_rlubdv7xuts7) 12

[os:](#_ry6qgshudb8p) 12

[**קטעי קוד שאהבתי**](#_nz0xqnns2n7j) **12**

[**רפלקציה:**](#_t1nn310r0gd) **13**

[**מדריך לשימוש:**](#_7ih0i4ydqoad) **13**

[**https://github.com/MosheStudent/TicTacToe.git**](#_2cys5rr83o65) **13**

[**תודות**](#_28dawccjlo92) **13**

# מבוא

השימוש במשחקי מחשב אוננליין (על הרשת) בימינו גדל בכל רגע. בעשור האחרון, הדבר הפך לעסק בשווי מיליארדי דולרים.

אמנם, עם השימוש הנרחב בטכנולוגיה - במיוחד בקרב בני נוער, הסכנה של פלישת תוכניות לא רצויות למשחק, ורמאות גדלות כל יום. נדרש פיתוח של דרכי תקשורת בטוחים, המונעים פלישה.

במערכת שאותה פיתחתי השתמשתי בהצפנה, כדי לאפשר משחק מהנה ובטוח לכולם. השתמשתי בהצפנת AES - הצפנה סימטרית חזקה - ובכך שמרתי על יציבות המשחק מפני פולשים לא רצויים (במידה מסוימת).

נתקלתי בהרבה קשיים, בבאגים ובלוגיקה של הקוד - אך בסוף המערכת מומשה.

אני יכול לראות את עתיד הפרויקט הופך למשהו יותר משמעותי - משחק יותר מסובך, כמו שחמט.

# תיאור תחום הידע

## יכולות מצד השרת-

1. יצירת מפתחות   
   **מהות היכולת**:  
   יצירת מפתח ציבורי ופרטי לשרת.   
   בעזרת המפתח הציבורי שישלח למשתמשים ההודעות שישלחו לשרת יהיו מוצפנות. בעזרת המפתח הפרטי השרת יוכל לפענח את ההודעות מהמשתמש, ולפעול בהתאם לכללי המשחק.

1. **אוסף היכולות הנדרשות**:  
   א) יצירת מפתחות לשרת
2. ההזנה לרשת  
   **מהות היכולת**:  
   ההזנה לרשת באמצעות THREAD על מנת לאפשר למשתמשים חדשים להתחבר לשרת.

**אוסף היכולות הנדרשות:**  
א) פעולת Thread

**מהות היכולת**:  
לחכות להודעה מצד הלקוח ולתת מענה ללקוח בהתאם לכללי המשחק.

**אוסף היכולות הנדרשות:**  
א) קליטת נתונים  
ב) פיענוח ההודעה  
ג) בדיקת מיקום על הלוח  
ד) ניתנת מענה ללקוח בהתאם לכללי המשחק  
ה) הצפנת התשובה ללקוח  
ו) שליחת התשובה (סימון מיקום על שני הלוחות)

## יכולות מצד הלקוח

1. הכנסת כתובת IP   
   **מהות היכולת**:  
   הכנסת כותבת הIP של השרת על מנת שהלקוח יוכל להתחבר אל השרת.
2. **אוסף היכולות הנדרשות**:  
   א) גישה לכתובת הIP של השרת
3. יצירת מפתחות

**מהות היכולת**:

יצירת מפתח ציבורי ופרטי ללקוח.   
בעזרת המפתח הציבורי שישלח לשרת, השרת יוכל להצפין את הפעולות אשר מיועדות ללקוח הספציפי. ובעזרת המפתח הפרטי הלקוח יוכל לפענח איפה בלוח הניח את הX או O.

**אוסף היכולות הנדרשות**:  
א) יצירת מפתחות ללקוח

1. קבלת הודעות מצד השרת

**מהות היכולת**:  
הקשבה לרשת על מנת לאתר הודעות שהתקבלו מהשרת וביצוע פעולות בהתאם לכללי המשחק.

**אוסף היכולות הנדרשות**:  
א) ממשק משתמש – מסך המשחק / מסך הכניסה  
ג) פיענוח ההודעה שהתקבלה מהשרת  
ד) בדיקת מיקום על הלוח  
ה) הצגת מיקום למשתמש

1. שליחת הודעות ללקוח אחר

**מהות היכולת**:  
שליחת הודעות לשרת על מנת לשלוח הודעות לאחד מהלקוחות האחרים שמחוברים למערכת.

**אוסף היכולות הנדרשות**:  
א) ממשק משתמש – מסך המשחק  
ב) גישה למחשב אחר על הLAN  
ג) קליטת נתונים מהלקוח (מיקום)  
ד) הצפנת ההודעה בעזרת המפתח הציבורי של השרת  
ה) שליחת ההודעה לשרת  
ו) הכנסת המיקום שנשלח למסך המשחק.

1. התנתקות מהמערכת  
   **מהות היכולת**:  
   התנתקות מהמערכת בסיום השימוש

**אוסף היכולות הנדרשות**:  
א) ממשק משתמש – כפתור שנועד להתנתקות (OK)  
ב) שליחת הודעה לשרת על סיום התקשורת   
ג) ניתוק המשתמש

# ארכיטקטורה של הפרויקט

## תיאור הארכיטקטורה של המערכת המוצעת:

הפרויקט מורכב מלקוח ושרת אשר מקיימים בניהם תקשורת ברחבי הרשת באמצעות THREADING, הפרויקט משתמש בחומרה - מקלת (להאזין כתובת IP של השרת) ועכבר (ללחוץ על מיקום בלוח).A picture containing text, diagram, screenshot, line

Description automatically generated

## 

## תיאור הטכנולוגיה הרלוונטית

במהלך הפרויקט השתמשתי בשפת התכנות python ובאמצעות יבוא ספריית socket קיימתי תקשורת שרת לקוח ברחבי הרשת.

## תיאור זרימת המידע

המידע עובר בצורה מוצפנת מהשרת אל הלקוח. הלקוח בתורו מעביר את המידע לנמען של ההודעה בצורה מוצפנת לשרת. העברת המידע מתבצעת באמצעות לחיצת יד משולשת (TCP).

## תיאור אלגוריתמים מרכזיים בפרויקט

הבעיה המרכזית בפרויקט היא איך להעביר מידע בצורה בטוחה ברחבי הרשת, על מנת לפתור בעיה זו השתמשתי בפרויקט בהצפנה מסוג AES.

בעזרת הצפנת המידע באמצעות AES נוכל לשלוח הודעות (מיקומים של שחקניים) ברחבי הרשת בצורה בטוחה. הינו יכולים להשתמש בעוד מגוון הצפנות, אך מכיוון שהAES הוא מצפין מהיר ובטיחותי מספיק לשימוש שלנו - השתמשתי בו.

## איך AES עובד? (ADVANCED ENCRYPTION STANDARD)

זו הצפנה סימטרית, כלומר – אותו מפתח (סיסמה סודית) משמש גם להצפנה וגם לפענוח. אם שני הצדדים יודעים את המפתח, הם יכולים לתקשר באופן מוצפן ובטוח.

כדי להצפין טקסט, הקוד מבצע כמה שלבים:

ראשית, הוא מקבל טקסט רגיל – לדוגמה "שלום עולם" – והופך אותו לרצף של מספרים שהמחשב יודע לעבוד איתם.

אחר כך, בגלל ששיטת AES דורשת שהטקסט יהיה באורך שמתחלק בדיוק ל-16 בתים (יחידות מידע), הקוד מוסיף "ריפוד" – תווים מיוחדים בסוף הטקסט – כדי שהוא יתאים לאורך הנדרש.

לאחר מכן, הקוד יוצר מספר אקראי שנקרא IV – זה כמו "מלח" שמוסיפים להצפנה, כדי שגם אם תצפין בדיוק את אותו טקסט פעמיים, תקבל תוצאה שונה בכל פעם. ה-IV הזה חשוב לאבטחה, והוא חלק מהנתונים שחייבים לשמור כדי שאפשר יהיה לפענח אחר כך.

עכשיו מתחיל שלב ההצפנה: הקוד משתמש באלגוריתם AES ובמצב CBC. הרעיון של מצב CBC הוא שכל חלק בטקסט (שנקרא "בלוק") מושפע גם מהתוכן של החלק הקודם. כך יוצא שגם אם יש חזרות בטקסט – הן לא נראות לעין בהצפנה. זה הופך את ההצפנה להרבה יותר חזקה.

בסיום, הקוד מחזיר את התוצאה הסופית – שהיא בעצם ה-IV (המספר האקראי) מחובר יחד עם הטקסט המוצפן. זה מה ששומרים או שולחים הלאה.

בפענוח, התהליך הפוך: הקוד לוקח את ה-IV מתוך תחילת המידע, מפעיל את אותה הצפנת AES כדי לפענח את הטקסט, מסיר את הריפוד, ומחזיר את הטקסט המקורי.

בשורה התחתונה: מדובר בשיטה בטוחה ומהירה להצפין טקסט – כל עוד שומרים את המפתח הסודי בסוד. זו הצפנה שמתאימה למקרים שבהם שני הצדדים כבר חלקו את הסוד ביניהם מראש.

## סביבת העבודה

על מנת לפתח את השרת שלי השתמשתי בסביבת הפיתוח VISUAL STUDIO CODE, ובשביל להריץ את הבדיקות הנחוצות לתקינות הצ'אט השתמשתי ב (BASH (LINUX TERMINAL.

## תיאור פרוטוקול התקשורת

### **עיקרי הפרוטוקול**

1. **ארכיטקטורה**:

* המשחק מבוסס על שרת (TicTacToeServer.py) ולקוחות (TicTacToeClient.py/NetworkGui.py).
* השרת מאזין לחיבורים על פורט 12345. כל זוג שחקנים מחוברים מקבלים חדר משחק (GameRoom.py) משלהם.
* כל שחקן מקבל לוח GUI משלו

1. **העברת הודעות**:

* התקשורת בין הלקוח לשרת נעשית באמצעות פרוטוקול TCP (לחיצת יד משולשת)
* כל הודעה שמועברת בין שרת ללקוח מוצפנת באמצעות צופן AES ומפוענחת בצד השני (cypher.py).
* לדוגמה, כששחקן מבצע מהלך, הלקוח שולח את המספר (1-9) לשרת. השרת בודק את המהלך, מעדכן את הלוח, ומחזיר לכל השחקנים את מצב הלוח.

1. **תהליך המשחק**:

כל שחקן בתורו מקבל הודעה "Your Move" מוצפנת.

הקליינט מפענח את ההודעה, ומציג למשתמש.

לאחר ביצוע מהלך, התשובה נשלחת לשרת, אשר בודק את תקפות המהלך, ומעדכן את הלוח.

אם יש מנצח/תיקו, השרת שולח הודעה לכל השחקנים.

1. **סיכום**:

כל ההודעות מוצפנות AES.

אין פורמט נתונים מתוחכם (JSON וכו'), אלא מחרוזות טקסט פשוטות.

כל הודעה עוברת דרך פונקציות encrypt/decrypt ב-cypher.py.

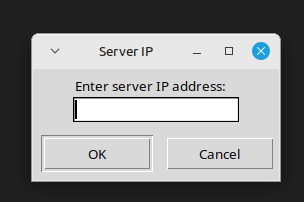
כל חדר משחק מנהל בעצמו את הסוקטים של שני השחקנים (באמצעות שרת ראשי).

# סקירת מסכים (GUI)

בפרויקט של השתמשתי בספרייה של TKINTER כדי לממש את הGUI.

## מסך כניסה:

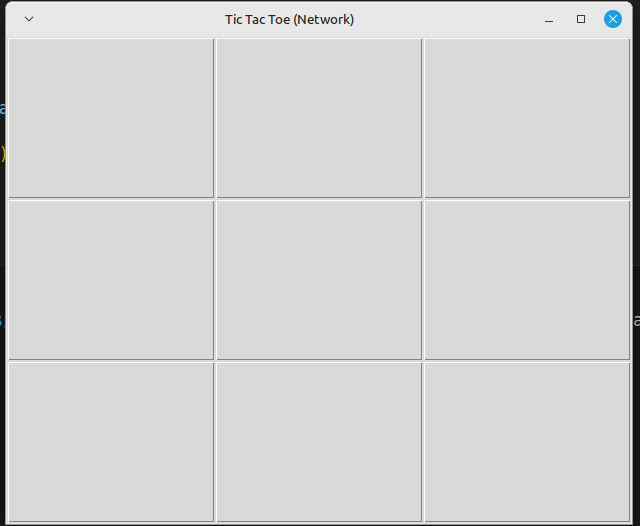
בתחילת המשחק כל שחקן נדרש להתחבר לשרת באמצעות כתובת IP:



מסך זה הוא מסך MESSAGEBOX, הוא נמצא שכבה מעל מסך רגיל (שימוש יותר פשוט) - לכן יש מסך מאחוריו (ריק).

## 

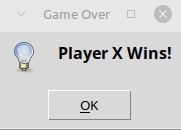
## מסך לוח משחק:



מסך זה הוא מסך ראשי בTKINTER. השתמשתי בפונקציית GRID של TKINTER כדי ליצור לוח 3X3 למשחק.

## מסך סוף משחק:

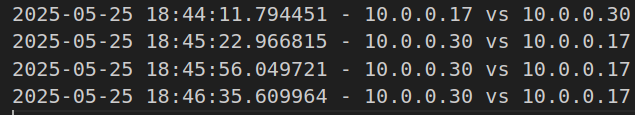
בסוף המשחק מופיע מסך MESSAGEBOX מסוג תת מסך (מסך ראשי מחובר אליו). המסך מציג את השחקן המנצח, ובעת סגירתו כל המסכים נסגרים - נגמר המשחק.



# שמירה למחשב

המשחק שומר אוטומטית את זמן המשחק ואת השחקניים (לפי כתובת IP), זה נעשה באמצאות פונקציית WRITE של פייטון (או APPEND).

הכל נשמר בקובץ TXT:



## 

# 

# סקירת חולשות ואיומים

### **תהליך התחברות:**

תהליך התחברות יתבצע באמצעות לחיצת יד משולשת.

### **מתקפת DOS/DDOS:**

מתקפת DOS/DDOS לא תתכן מכיוון שהשרת מסוגל לבצע תקשורת באמצעות Thread עם כמה מכשירים במקביל, לכן במידה וינסו על השרת מתקפת DOS או DDOS השרת לא יקרוס מכיוון שהוא יכול להתמודד עם מספר אינסופי של לקוחות.

### **ניתוח תנועה (TRAFFIC ANALYSIS):**

למרות שאי אפשר לנתח ישירות את המידע, אפשר להבחין בגודלם, ובקצב השליחה. במקרים יותר חשובים זה יכול לחשוף מידע!

# מימוש הפרויקט

## סקירת המחלקות והמודולים המיובאים:

### **socket:**

באמצעות מחלקה זו נבצע תקשורת בין שרת ללקוח

### **tkinter:**

באמצעות מחלקה זו נבנה את ממשק המשתמש (GUI)

### **threading:**

באמצעות מחלקה זו נוכל לקיים צ'אט מרובה משתמשים.

### **os**:

מאפשר קשר עם מערכת ההפעלה

**cryptography:**

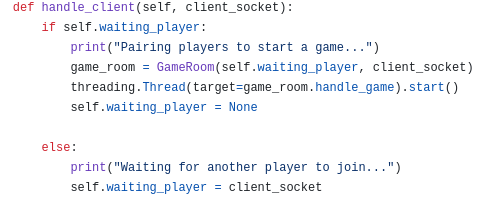
ספרייה ענקית העוזרת בכל דבר הקשור להצפנה

**DATETIME:**

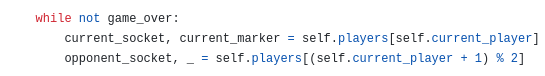
ניהול זמן - שימוש בכתיבת LOG למשחק.

# קטעי קוד שאהבתי

1. כל חדר הוא THREAD חדש. ממש אהבתי את הרעיון של יצירת מספר חדרים אינסופי שאפשר להיכנס אליהם, והשימוש בTHREAD יחד עם זאת יצא אלגנטי:



1. במקום להשתמש בCOUNT ולבזבז משתנה, הגעתי לרעיון אחר. הוספתי אחד, אך בגלל שיש שני שחקנים, אפשר לעשות מודלו למספר הזה, וזה תמיד יצא 0 או אחד, לא משנה כמה גבוה המספר:



# רפלקציה:

ממש נהנתי בבניית הפרויקט. זה לימד אותי רבות על הצפנה, על שימוש בגיט וגיט האב, על פייטון ועל סדר עבודה. היו הרבה קשיים בדרך, אך בעזרת המורה וחברים הצלחתי להתגבר.

החלפתי כמה פרויקטים מתחילת השנה - ואני שמח להגיד שאני מאוד גאה במה שיצא.

אם הייתי מתקן או משפר - הייתי מנהל באגים יותר טוב, הייתי מתקן טיפה את ה UI. משחק חוזר זאת גם רעיון שעלתה לי לראש.

סך הכל, אני ממש שבע מהלימודים במדעי המחשב והנדסת תוכנה - ואשמח בעתיד ללמוד עוד.

# 

# 

# מדריך לשימוש:

# <https://github.com/MosheStudent/TicTacToe.git>

# תודות

תודה למורה שמעון סוויסה שליווה אותי במהלך הכנת הפרויקט. שמעון תמיד שם לצידי כשאני במצוקה. הייתי יכול לומר ששמעון הוא ה"POOKIE" שלי.

אני מאחל לשמעון הצלחה רבה!