

פרויקט תכנות- רשתות ומחשבים ואינטרנט 1

מבוא-

אתם הולכים לממש גרסה online של השעשועון "המרדף" (למי שלא מכיר את המשחק מומלץ להסתכל באתר YouTube בערוץ של כאן).

הפרויקט נועד לשפר את כישורי התכנות שלכם ולחשוף אותכם לתכנות מסוג client-server ומסוג socket programming. כמו כן באמצעות הפרויקט תכירו יותר לעומק את שכבות התעבורה ואת שכבת האפליקציה (שכבה 4 ו-5).

נזכיר שבארכיטקטורת client-server יש שני צדדים- **הלקוח** שמקבל שירות **והשרת** שמספק שירות. במקרה שלנו-

- צד הלקוח יהיה משתמש במחשב שהוא יקבל שאלות ויענה עליהם.
- צד השרת יהיה המחשב שאחראי לניהול המשחק (שירותיו כוללים סיפוק שאלות, מעקב אחרי התקדמות המשחק, דווח למשתמש לגבי נכונות ולגבי התקדמותו וכו') בנוסף צד השרת יהיה הצ'ייסר שבמקרה שלנו יהיה אוטומטי ויענה נכון על שאלות בהסתברות גבוהה.

כללי המשחק-

להלן לוח של המשחק-

לוח התקדמות	שלבים
צ'ייסר	שלב הצ'ייסר
	1
	2
	3
	4
	5
	6
בנק	7

שלב התחלה-

בתחילת המשחק השחקן מקבל הודעת פתיחה שבה שואלים את השחקן אם הוא רוצה לשחק ויש לו שתי אופציות-

1. השחקן עונה כן ואז מתבצע חיבור עם השרת. המשחק ממשיך לשלב א'.
2. השחקן עונה לא ואז לא מתבצע החיבור ונסיים את ריצת תוכנית המשתמש.

שלב א'-

המשתמש מקבל שלוש שאלות ולפיהן יקבע כמה כסף יש לו בחשבון, רק אם המשתמש צדק בלפחות שאלה אחת הוא ממשיך לשלב הבא אחרת הוא מתחיל מהתחלה-

1. אם צדק ב 0 שאלות : 0 ₪ והוא עובר לשלב התחלה.
2. אם צדק ב 1 שאלות : 5000 ₪ והוא ממשיך לשלב הבא.
3. אם צדק ב 2 שאלות : 10000 ₪ והוא ממשיך לשלב הבא.
4. אם צדק ב 3 שאלות : 15000 ₪ והוא ממשיך לשלב הבא.

שלב ב'-

למשתמש יש 3 אופציות-

1. להתחיל משלב 3 עם הסכום הנוכחי.
2. להתקרב צעד אחד לצ'ייסר ולהתחיל משלב 2 עבור פי 2 מסכום הנוכחי.
3. להתרחק צעד אחד מהצ'ייסר ולהתחיל משלב 4 עבור $1/2$ מסכום הנוכחי.

לדוגמא אם המשתמש צדק בשלב א' בשתי שאלות אז בחשבון כרגע יש 10000 ₪ ויהיה לו 3 אופציות כמו בשרטוט הבא-

	צ'ייסר
1	
2	20000
3	10000
4	5000
5	
6	
7	בנק

שלב ג'-

כעת מתחיל המרדף-

שואלים את המשתמש שאלה ומציגים לו 4 תשובות אפשריות. למשתמש יש 2 אופציות-

1. לבחור תשובה כלשהי.
 2. המשתמש יוכל לבקש "גלגל הצלה" ולהוריד 2 אופציות שגויות מתוך 4 האופציות לתשובה(למשתמש יש רק גלגל הצלה אחד).
- אם המשתמש בוחר את התשובה הנכונה הוא מתקדם צעד אחד לקראת הבנק, אם הוא טועה הוא נשאר באותו שלב.

הצ'ייסר גם צריך לענות על השאלה אבל במקרה שלנו הצ'ייסר הוא מחשב ולכן נתכנת אותו כך שתהיה לו הסתברות של 0.75 לענות נכון.

אם הצ'ייסר בוחר את התשובה הנכונה הוא גם מתקדם שלב אחד לקראת הבנק, אם הוא טועה הוא נשאר באותו שלב.

לאחר שהמשתמש והצ'ייסר ענו נשלח למשתמש עדכון של הלוח הנוכחי שכולל 3 דברים –

1. כמה כסף יש למשתמש.
2. באיזה שלב נמצא המשתמש.
3. באיזה שלב נמצא הצ'ייסר.
4. האם נותר גלגל הצלה.

נחזור לשלב ג' אם המשחק עוד לא הסתיים או שנעבור לשלב סיום אם המשחק הסתיים (הכוונה בהסתיים מוסברת בשלב סיום).

לדוגמא –

נניח שבתור הנוכחי זה מצב הלוח –

1	
2	צ'ייסר
3	
4	שחקן
5	
6	
7	בנק

צ'ייסר ושחקן צודקים –

1	
2	
3	צ'ייסר
4	
5	שחקן
6	
7	בנק

צ'ייסר צודק ושחקן טועה-

1	
2	
3	צ'ייסר
4	שחקן
5	
6	
7	בנק

צ'ייסר טועה ושחקן צודק-

1	
2	צ'ייסר
3	
4	
5	שחקן
6	
7	בנק

צ'ייסר טועה ושחקן טועה(אין שינוי)-

1	
2	צ'ייסר
3	
4	שחקן
5	
6	
7	בנק

שלב סיום-

המשחק יסתיים באחת משתי מהאופציות הבאות-

1. המשתמש מגיע לבנק(שלב 7) לפני שהצ'ייסר מספיק להגיע לשלב שלו. ניצחון של המשתמש.
2. הצ'ייסר מגיע לשלב של המשתמש לפני שהמשתמש מגיע לבנק. ניצחון של הצ'ייסר.

לאחר שהמשחק מסתיים השרת ישלח הודעת סיכום של המשחק(ניצחון של הצ'ייסר או השחקן) ויציע למשתמש להתחיל משחק חדש ואם כן נחזור לשלב א' אחרת השרת ינתק את המשתמש ונסיים את ריצת תוכנית המשתמש.

לדוגמא-

אם זה מצב הלוח-

1	
2	
3	
4	צ'ייסר
5	שחקן
6	
7	בנק

אז אם הצ'ייסר יצדק בשאלה הנוכחית והשחקן יטעה אז הצ'ייסר מנצח.
(הערה- אם המשתמש צודק והצ'ייסר צודק במקרה הזה אז זה לא נחשב
שהשחקן אלא פשוט שניהם מתקדמים צעד לבנק)

אם זה מצב הלוח-

1	
2	
3	
4	צ'ייסר
5	
6	שחקן
7	בנק

אז אם השחקן צודק והצ'ייסר צודק או טועה(לא משנה) אז השחקן מנצח.

הנחיות מימוש-

- אתם יכולים לממש את הקוד **בכל מערכת הפעלה שתמצאו ובכל שפה שנוחה לכם**. אחנו ממליצים לתכנת ב python כמו שהראינו בכיתה ולדעתנו זה יחסוך זמן(למי שלא מכיר python רצוי שתלמדו קצת דרך האינטרנט לפני שתתחילו לכתוב קוד, זו שפה שיותר פשוט להבין מהר במיוחד כאשר יש היכרות עם שפה כמו C).
- אנחנו מצפים שתתכנתו עם מחלקות/struct בהתאם ללוגיקה של המשחק.
- **אסור** להשתמש בספריות שאינן סטנדרטיות לתכנות socket . לדוגמה ב python עליכם להשתמש במודול socket.
- אם אתם מתלבטים ואינכם בטוחים בספריות שלכם אז תתייעצו עם אחד המתרגלים.
- אתם יכולים לעשות הקלות במימוש אבל אתם **מחויבים** לציין בפירוש בקובץ PDF את ההקלות שביצעתם כמובן שזה גורר הורדת נקודות אבל

אם לא תציינו אז זה ייתפס כטעות וזה עלול לגרור הורדת נקודות גדולה יותר.

דגשים-

- אתם אחראים על להכין מבחר שאלות בגודל 15 לפחות הכולל 4 תשובות (אחת נכונה ושלוש לא נכונות).
- הצ'ייסר הוא אוטומטי ואינו משתמש בנושיו. הסתברות ההצלחה של הצ'ייסר היא 0.75.
- **אתם צריכים לאפשר התחברות של עד 3 שחקנים במקביל לשרת.**
כל שחקן שמתחבר הוא משחק בפני עצמו ואין קשר בינו לשחקנים אחרים. אם שחקן רביעי מנסה להתחבר שלחו לו הודעת דחייה בה מצוין שיש כבר שלושה שחקנים.
- **איננו דורשים ממכם שתכתבו את הפלט ל console בצורה ספציפית והקוד אינו יועבר בדיקה אוטומטית. כלומר אין לכם צורך להיות זהירים ולשים לב לכל אנטר, רווח וכו' . אנו מעוניינים שתשקיעו את מיטב הזמן שלכם ב-**
 - **כתיבת קוד איכותי ומתועד.**
 - **תיעוד נלווה ב PDF.**
 - **הבנת החומר התיאורטי מאחורי הקוד.**
- הבדיקה תהיה פרונטלית מול המתרגלים בו תריצו את הקוד שלכם ולכן כן חשוב שהפלט יהיה תקין לפי שלבי המשחק שהצגנו. חשוב שתבינו את החומר התיאורטי מאחורי הפרויקט כי נבחן אותכם עליו. כמו כן תהיה בדיקה נפרדת של הקוד שלכם בה נסתכל לראות שהקוד כתוב היטב ומחולק למחלקות struct לפי לוגיקת המשחק.

הנחיות תיעוד-

תתעדו את הקוד היטב.

תוסיפו דף תיעוד בפורמט PDF שמכיל הסברים בסדר הבא-

1. שם מלא ות.ז.
2. מערכת הפעלה, שפת תכנות + גרסה.
3. רקע תיאורטי.
4. הסבר לכל קובץ קוד שעשיתם- מחלקות ופונקציות.
5. הסבר על פתיחת socket- באיזה פרוטוקול פתחתם, מה הן הפקודות והסבר על כל פקודה וכו'.
6. צילומי מסך של הרצת התוכנית מהתחלה ועד סוף עבור שתי האופציות של התקדמות המשחק(שחקן מנצח + צ'ייסר מנצח).

הנחיות הגשה-

- שימו את כל הקבצים בקובץ ZIP עם שמות המשתמשים ותעודות זהות והגישו דרך המודל. רק אחד מהזוג צריך להגיש במודל.
- בכל קוד כתבו בהתחלה את שמות המגישים + ת.ז.
- יש להגיש עד לתאריך: 20.12.2020 (שימו לב שזה תאריך רחוק אז נצלו את זמנכם בתבונה).
- ההגשות בזוגות.
- הגשה באיחור לא תאושר אלא באישור של אחד המתרגלים.

קישור מומלץ-

<https://realpython.com/python-sockets/>

בהצלחה!