עבודת גמר בתכנון ותכנות מערכות  
 תוך התמחות בהגנת סייבר.  
סמל מקצוע: 883589



**משחק ניחוש באמצעות ציור רב משתתפים**

נושא העבודה: Multiplayer Drawing And Guessing Game

שם התלמיד: רון אייזן

תעודת זהות: 213412422

בית ספר: ראשון לציון, מקיף ה' אמירים.

שם המנחה: מוטי מתתיהו

מועד הגשה: תשפ"א 2021

**תוכן עניינים**

[מסמך ייזום 4](#_Toc72809924)

[מבוא ורקע כללי לנושא העבודה 5](#_Toc72809925)

[נושא הפרויקט 5](#_Toc72809926)

[תקציר הפרויקט 5](#_Toc72809927)

[הגדרות 5](#_Toc72809928)

[מטרת המערכת ויעדיה 5](#_Toc72809929)

[ארכיטקטורת המערכת 6](#_Toc72809930)

[הגדרת רכיבי המערכת 6](#_Toc72809931)

[תיחום המערכת 7](#_Toc72809932)

[סיקור מצב השוק כיום: 7](#_Toc72809933)

[אילוצים ודרישות: 7](#_Toc72809934)

[פירוט סביבת העבודה והכלים הנדרשים לפיתוח: 8](#_Toc72809935)

[תיאור המוצר המוגמר: 8](#_Toc72809936)

[ניהול פרויקט עתידי: 9](#_Toc72809937)

[מסמך אפיון 10](#_Toc72809938)

[פונקציונליות המערכת 11](#_Toc72809939)

[הערכת המוצר לעומת התכנון 14](#_Toc72809940)

[פיתוחים לעתיד 14](#_Toc72809941)

[אילוצים ודרישות המערכת 14](#_Toc72809942)

[סביבת עבודה 14](#_Toc72809943)

[מסמך עיצוב 15](#_Toc72809944)

[תיאור בסיס מערכת הנתונים במערכת 16](#_Toc72809945)

[שדות בסיס הנתונים 16](#_Toc72809946)

[הסבר השדות 16](#_Toc72809947)

[מודולים 17](#_Toc72809948)

[Server 17](#_Toc72809949)

[Surface 18](#_Toc72809950)

[Screen 19](#_Toc72809951)

[19](#_Toc72809952)

[אלגוריתם המערכת 21](#_Toc72809953)

[הגדרת התקשורת (RFC) 23](#_Toc72809954)

[תיאור ממשק המשתמש 24](#_Toc72809955)

[קטעי קוד מיוחדים 28](#_Toc72809956)

[יומן רפלקציה 30](#_Toc72809957)

[בחירת GUI 31](#_Toc72809958)

[וובליוגרפיה 32](#_Toc72809959)

[נספחים 34](#_Toc72809960)

[קישור לMVP שיצרתי: 35](#_Toc72809961)

[קישור לקבצי הפרויקט בגיט-האב: 35](#_Toc72809962)

­­­­

מסמך ייזום

מבוא ורקע כללי לנושא העבודה

בחרתי לעשות את הפרויקט הזה מפני שלדעתי, הוא שילוב מעניין של החומר שנלמד במהלך השנים- רשתות ומערכות הפעלה. הפרויקט משלב מספר קונספטים שונים שנלמדו במהלך  
השנה האחרונה והשנים שקדמו לה. בנוסף, בסוף הפרויקט יוצא מוצר שמיש וייחודי שגם  
מעניין מבחינת הנושא שלו. את הרעיון לפרויקט קיבלתי מהאפליקציה skribbl.io שהיא אפליקציה אינטרנטית חינמית בעלת רעיון דומה לרעיון עליו מתבסס הפרויקט שלי. כמו כן, בעת חיפוש רעיון לפרויקט תחילה בחרתי ברעיון "עוזרת אישית קולית", אך לאחר מכן התברר כי רעיון זה נתפס על ידי תלמיד אחר בכיתה, אז החלטתי להתמקד בתחום המשחקים כך שהמשחק יכיל את כל דרישות החובה שרצויות בפרויקט. לבסוף חשבתי על רעיון הדומה לאפליקציית skribbl.io, ונעזרתי בה לרעיונות עיצוביים ותוכניים עבור הפרויקט שלי.

נושא הפרויקט

משחק ניחוש באמצעות ציור, רב משתתפים.

תקציר הפרויקט

הפרויקט הוא משחק רב משתתפים, בו כל משתמש נרשם באמצעות יצירת שחקן חדש או שמתחבר לשחקן קיים ובאחת מהדרכים נכנס למאגר המשתמשים, משתמש אחד נבחר מבין שאר השחקנים ומוצגת בפניו באופן פרטי (ולא בפני שאר המשתתפים) מילה ממאגר מידע מוכן של פריטים, חפצים, בעלי חיים, מקצועות ועוד... ועליו להביע את המילה על ידי ציורה על הלוח בעוד שעל שאר המשתתפים לנחש את המילה על ידי כתיבתה בתיבת טקסט מוכנה על המסך.

הגדרות

משתמשי המערכת

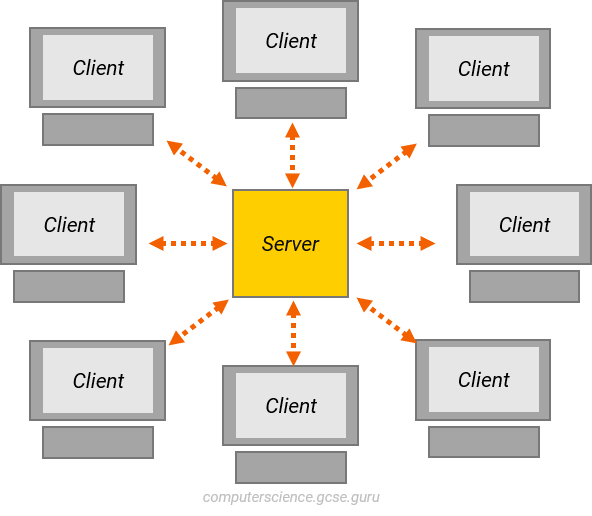
המערכת מיודעת לכל אדם עם מחשב המעוניין לשחק במשחק ניחוש מהנה.

מטרת המערכת ויעדיה

מטרת המערכת היא לתת למשתמש חווית משחק נוחה ומהנה, שליטה ובקרה נוחה. במשחק זה רוכז רעיון מהנה וייחודי, ונעטף בממשק גרפי נוח, ברור וקל לתפעול. בנוסף היא גם מאפשרת משחק רב משתתפים, למשחק בחברותא. המשחק עצמו הוא משחק תחרותי מהנה ומאתגר בין השחקנים המשתתפים בו.

ארכיטקטורת המערכת

המערכת נבנתה בארכיטקטורת שרת-לקוח. המערכת תומכת במספר רב של לקוחות, תוך תמיכה בריבוי תהליכים במקביל- הן מספר תהליכים שונים על אותו לקוח והן מספר תהליכים שונים בין לקוחות שונים. המערכת מטפלת בכל לקוח בעזרת תהליך (THREAD) שונה.

**תרשים מספר 1: ארכיטקטורת המערכת – שרת לקוח**

מקור התמונה: <https://www.computerscience.gcse.guru/theory/client-server-networks>

הגדרת רכיבי המערכת

1. רשת מקומית- (Network Area Local- LAN) רשת מקומית. רשת מחשבים המתפרסת על אזור קטן ומוגבל, לרוב בתוך בניין.
2. שרת- (server) ממנו מופעלות הפעולות השונות של הלקוחות, מודל אשר מדומה לצינור המחבר על ידי IP ו Port לשני מכשירים שונים ומאפשר להעביר [חבילות נתונים](https://he.wikipedia.org/wiki/%D7%97%D7%91%D7%99%D7%9C%D7%AA_%D7%9E%D7%99%D7%93%D7%A2) (data packets) מאחד לשני .
3. לקוח- (client) מחשב של משתמש המחובר למערכת.
4. :Client-Server-model [ארכיטקטורת תוכנה](https://he.wikipedia.org/wiki/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) ל[חישוב מבוזר](https://he.wikipedia.org/wiki/%D7%97%D7%99%D7%A9%D7%95%D7%91_%D7%9E%D7%91%D7%95%D7%96%D7%A8), אשר מגדירה את היחס בין [תוכנות](https://he.wikipedia.org/wiki/%D7%AA%D7%95%D7%9B%D7%A0%D7%94) משתפות פעולה. המודל מחלק את המשימות או עומס העבודה בין ספק השירות או המשאבים - ה[שרת](https://he.wikipedia.org/wiki/%D7%A9%D7%A8%D7%AA_(%D7%9E%D7%97%D7%A9%D7%91%D7%99%D7%9D)), לבין מבקש השירות – הלקוח.

תיחום המערכת

הפרויקט עוסק ומשלב בתוכו תחומים רבים וביניהם:

1. תקשורת- נעשית תקשורת בין מספר מחשבים (4) שרת ו-3 לקוחות (כמספר המשתתפים).  
2. בסיס נתונים- נעשה שימוש בבסיסי נתונים למאגר שמות השחקנים במשחק.  
3. אבטחת מידע- הצפנת מידע רגיש של השחקן במשחק, לדוגמה סיסמת השחקן.  
4.חלק ויזואלי (GUI)- יצירת ממשק גרפי בעל מקשים נוח וידידותי למשתמש.

המודולים ותתי הנושאים בהם אני משתמש בפרויקט:

1. בתחום התקשורת נעשה שימוש במודולים socket וthreads כשאר שניהם מאפשרים יחד תקשורת בין שרת למספר לקוחות.
2. בתחום בסיס הנתונים נעשה שימוש במסד הנתונים SQlite.
3. יצירת הממשק הגרפי תעשה על ידי שימוש במודול tkinter.

סיקור מצב השוק כיום:

כיום ישנם מספר משחקים בעלי רעיון דומה. את ההשראה לפרויקט שאבתי מאפליקציה בשם SKRIBBL.  
ישנן וריאציות שונות למשחק (חלקן ברשת וחלקן מחוצה לה) ובמרבית האפליקציות ניתן לשחק עם אנשים שונים מכל העולם.

אילוצים ודרישות:

1. על המערכת לעבוד על מחשב שמערכת ההפעלה שלו היא WINDOWS.
2. לוח זמנים צפוף, מתכוון לבצע שלושה מועדי פסיכומטרי בעת תקופת ביצוע הפרויקט.
3. קהל היעד הוא עבור ילדים בגילאים 8-18.

פירוט סביבת העבודה והכלים הנדרשים לפיתוח:

שפת התכנות: Python 3.7.3

סביבת העבודה: WINDOWS. ו- PYCHARM

כלים הנדרשים לפיתוח:

1. בתחום התקשורת נעשה שימוש במודולים socket וthreads כשאר שניהם מאפשרים יחד תקשורת בין שרת למספר לקוחות.
2. בתחום בסיס הנתונים נעשה שימוש במסד הנתונים SQlite.
3. אימות דו שלבי, הצפנת המדע תעשה באמצעות הצפנת RSA.
4. יצירת הממשק הגרפי תעשה על ידי שימוש במודול tkinter.

הכלים הנדרשים לבדיקה וסיוע בפיתוח:

1. מספר מחשבים על מנת לבדוק את תחום התקשורת.
2. תיעוד והסבר בקוד מה מקבלת כל פונקציה או כל אובייקט ומתי הלקוח שולח מידע לשת ומתי השרת אל הלקוח וכו'...
3. תכנות מונחה עצמים (OOP).

תיאור המוצר המוגמר:

שם המוצר: Draw & Guess.  
מה המוצר המוגמר אמור לבצע: על המוצר המוגמר להיות משחק רב משתתפים של 3 שחקנים, שבעבור כל משתתף מוצג לוח הציור של אותו השחקן שבפניו מוצגת המילה.  
מטרות מרכזיות:

1. חשיפה למודולים חדשים בשפת פייתון.
2. התנסות ראשונית ביצירת פרויקט בהיקף של חמש יח"ל.
3. העמקה והרחבת הידע בתחום הסייבר והצפנת המידע.
4. התנסות בתכנות עם בסיסי נתונים.
5. יצירת ממשק גרפי (GUI).

המערכת עונה על **כל** המטרות המרכזיות שציינתי וניתן לראות זאת בהרחבה בחלק תיחום הפרויקט.

ניהול פרויקט עתידי:

**תרשים מספר 2: טבלת ניהול פרויקט עתידי**

|  |  |
| --- | --- |
| יעד | תאריך יעד |
| קישור בין שרת ולקוח ויצירת בסיס נתונים מלא עם מילים, ובסיס נתונים ריק של שחקנים. | 10.11.20 |
| קישור בין בסיסי הנתונים לשרת.  יצירת פעולה שמקבלת את המילה מהשרת | 30.11.20 |
| פעולה ששולחת את הניחוש לשרת | 15.12.20 |
| הוספת ממשק גרפי פעולה שבאמצעותה מצייר הלקוח את המילה שקיבל. | 15.1.21 |
| שיפורים/ תיקונים אחרונים | 1.4.21 |

מסמך אפיון

פונקציונליות המערכת

פעולות רקע:

1 .חיבור כל המשתמשים לשרת.

2 .שמירת נתוני המשתמשים.

3 .פתיחת מסך הפתיחה.

פעולות השרת:

1. שידור מסכים בזמן אמת.
2. שידור הציור לכלל המשתמשים.
3. שולח לשחקנים אם המשחק הסתיים.
4. חישוב ושליחת הניקוד לכל שחקן ולצייר.
5. הגרלת המילה ממאגר הנתונים ושליחתה.

פעולת הלקוח:

1. התחברות לשרת.
2. שליחה לשרת האם הצליח לנחש את המילה או לא.
3. הרשמה ראשונית למשחק.
4. כניסה למשחק עם משתמש קיים (log in).
5. בדיקה האם הניחוש נכון.
6. טיימר.
7. האם הסיבוב הסתיים או לא.

דרישות פונקציונליות :

**תרשים מספר 3: דרישות פונקציונליות**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | דרישה | תהליך | קלט | פלט | טיפול בשגיאות |
| 1 | התחברות לserver | המשתמש תחבר לserver של המערכת על ידי socket. | Portו IP של ה server | התחברת למערכת | הודעת שגיאה : נסה להתחבר למערכת מחדש. |
| 2 | הרשמה ראשונית למערכת | המשתמש יכניס את פרטיו יבצע בדיקת סיסמא ולאחר מכן פרטיו ישמרו בשדה הנתונים . | פרטי המשתמש (שם משתמש, סיסמא, אמייל) | הלקוח נרשם למערכת בהצלחה. | הודעה שגיאה :  - אמייל או שם משתמש שגוי, סיסמה קצרה מדי. |
| 3 | התחברות למשתמש קיים | המשתמש יכניס את שם המשתמש וסיסמתו אשר יבדקו בשדה הנתונים ולאחר מכן יכנס למערכת . | שם המשתמש והסיסמא של המשתמש. | התחברת למערכת בהצלחה. | במקרה שהמשתמש יכניס פרטים שגויים הוא יתבקש לחזור על הפעולה |
| 4 | טיימר של 80 שניות | יצירת THREAD חדש שבו ירוץ הטיימר במשך 80 שניות, מכיוון שהטיימר בתהליכון משלו גם יהיה מעבר בין מסכים או שליחות בין השרת ללקוח הטיימר ימשיך לעבוד. | זמן | טיימר על מסך המשתמש. | שימוש בTRY וEXCEPT על מנת למנוע קריסות במהלך ההרצה. |
| 5 | שליחת מילה ממאגר הנתונים ללקוח | השרת מגריל מילה ממאגר הנתונים, ושולח אותה ללקוח באמצעות socket | - | המילה המוגרלת | - |
| 6 | בדיקות הניחוש בלקוח | הלקוח בודק האם הלקוח ניחש נכונה. | הניחוש | אמת או שקר. | - |
| 7 | שידור מסכים בזמן אמת. | יצירת תהליכון שמקבל קאורדינטות מהשרת ומצייר בהן על מסך המשתמש. | קאורדינטות מופרדות ב';' מכילות X וY. | ציור על המסך. | שימוש בTRY וEXCEPT על מנת למנוע קריסות במהלך ההרצה. |
| 8 | האם יש צורך לסיים את המשחק. | אני בודק בשרת רם מספר המשתמשים שניחשו כבר או סיימו את ניחושיהם ולא הצליחו לנחש את המילה שווה למספר השחקנים פחות אחד (הצייר) ובמידה וכן השרת שולח ללקוחות 'END' והלקוח מבין שהוא צריך לעצור את הסיבוב ולהתחיל את הסיבוב הבא, או לחליפין במידה שנגמר הזמן. | - | שליחת END. | - |
| 9 | הגרלת המילה ממאגר הנתונים ושליחתה. | בשרת יש רשימת מילים מכל מיני סוגים, באמצעות המודול RANDOM השרת מגריל מילה מן המאגר ושלח אותה לשחקן בצירוף עם תפקידו בסיבוב הנוכחי, נחשן (GUESS) או צייר (DRAW). | - | המילה בסיפוח התפקיד של השחקן | - |
| 10 | 1. חישוב ושליחת הניקוד לכל שחקן ולצייר. | בעת ניחוש המילה הלקוח שולח לשרת כי השחקן ניחש נכונה והשחקן בתגובה מקבל את הניקוד שקיבל, השחקן נרשם ברשימת האנשים שניחשו את המילה, הניקוד המוענק לו מחושב עפ"י מיקומו ברשימה, אם ניחש ראשון יקבל ניקוד גבוה מהאדם שניחש אחרון...  כמו כן גם הצייר מקבל 30 נקודות בעבור כל ניחוש של שחקן. | הודעה מהלקוח כי השחקן ניחש נכונה. | שליחת הודעה המכילה את הניקוד הניתן לו בצירוף עם המילה SCORE. | - |

הערכת המוצר לעומת התכנון

באופן כולל הצלחתי לממש את החזון ההתחלתי שהיה לי, מלבד כמה שינויים ואילוצים שונים למיניהם שהכריחו אותי לשנות את התכנון את ההתחלתי.

פיתוחים לעתיד

ישנם מספר אפיקים בהם הייתי רוצה לעבוד על מנת לפתח את המשחק:

* ממשק גרפי: הייתי רוצה להפוך את הממשק הגרפי ליותר נוח וידידותי בעבור המשתמש, בכמה מובנים, שהציור יהיה יותר חלק וכך גם הצגתו בשאר המחשבים של שאר השחקנים. הוספת כפתורים שונים שיוכלו לסייע במקרים יוצאי דופן (משתמש מתנתק מהמשחק, השרת קורס וכדומה..)
* מאגר נתונים: להצפין את המידע במאגר הנתונים.
* מבחינת הקוד: הייתי רוצה למטב את הקוד אף יותר, להפוך אותו ליעיל יותר, קצר יותר ולארגן אותו טוב יותר למחלקות.

אילוצים ודרישות המערכת

בסיס הנתונים ימומש בשפת – SQL וישמרו בו נתוני המשתמש, ונתוני הקבצים ששותפו .

על המחשבים המחוברים למשחק להיות בעלי חיבור לאינטרנט וצריך שיהיה ביניהם חיבור.

סביבת עבודה

שפת פרויקט – . Python 3.7

מערכת הפעלה – WINDOWS.

ממשק גרפי – Tkinter ליצירת ה-GUI .

בסיס הנתונים – sqlite3 בשביל יצירת מאגר הנתונים. ו- DB Browser בשביל צפייה במאגר.

PyCharm סביבת עבודה - לכתיבת הקוד, Debugging , והרצה.

מסמך עיצוב

תיאור בסיס מערכת הנתונים במערכת

-Users.dbמאגר נתונים שמכיל את פרטי המשתמשים במשחק. השרת מתחבר אל מאגר הנתונים, מוסיף אליו משתתפים, בודק אם משתתפים קיימים, מקבל הרשאה להכנסת משתתפים רשומים כלומר יש לו גישה מלאה בכל הקשור למאגר הנתונים.

שדות בסיס הנתונים

**תרשים מספר 4: שדות בסיס הנתונים**

|  |  |
| --- | --- |
| Type | Field Name |
| TEXT | Username |
| TEXT | Password |
| TEXT | Email |

הסבר השדות

Username – שם המשתמש, מסוג טקסט, מפתח ראשי, חייב להיות ייחודי (כלומר לכל משתמש יש שם ואין כפילויות), לא יכול להיות ריק.

Password – סיסמת המשתמש, מסוג טקסט, לא חייבת להיות ייחודית (כלומר יכול לקרות מצב בו למספר משתמשים ישנה אותה הסיסמה), לא יכולה להיות ריקה.

Email – הדואר האלקטרוני של המשתמש, מסוג טקסט, חייב להיות ייחודי (כלומר לכל משתמש יש דוא"ל ואין כפילויות), לא יכול להיות ריק.

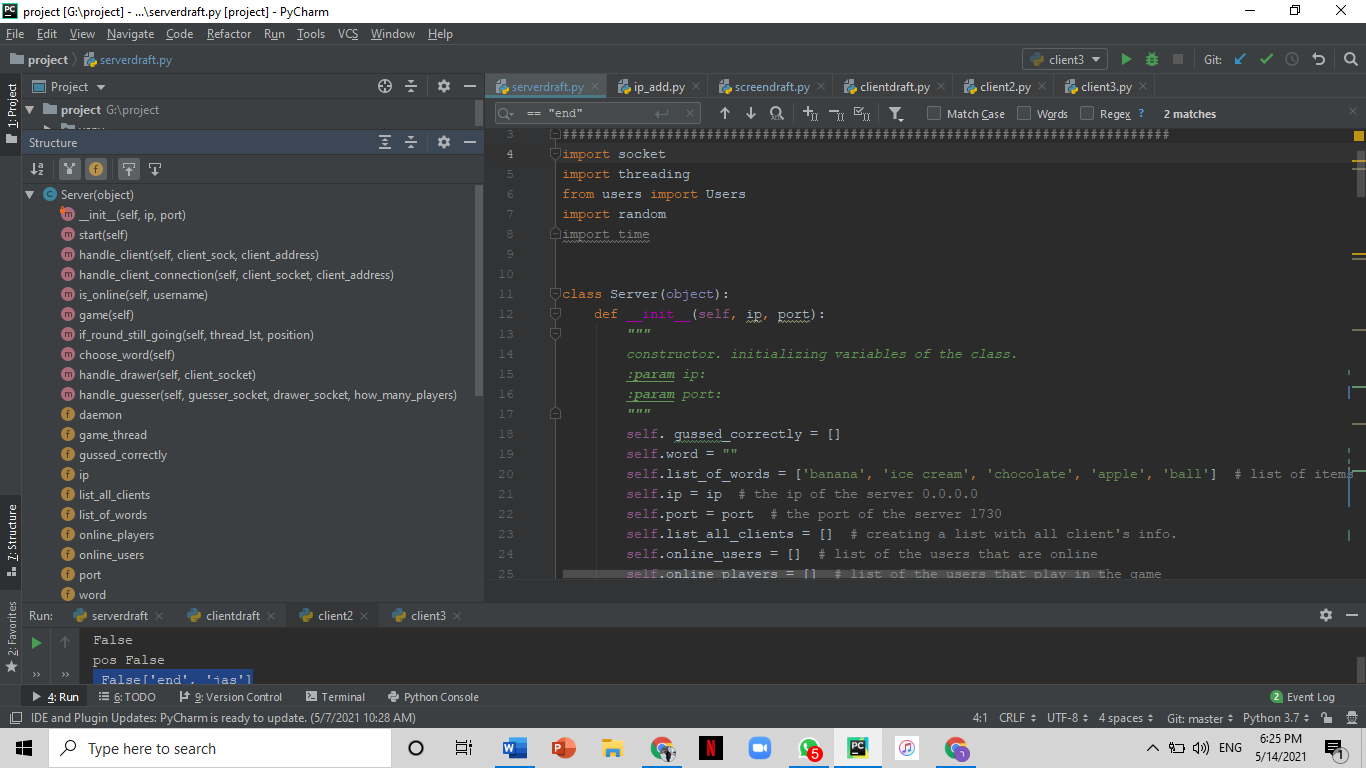
**תרשים מספר 5: דוגמה לבסיס נתונים**

|  |  |  |
| --- | --- | --- |
| Email | Password | username |
| [ronaizen@gmail.com](mailto:ronaizen@gmail.com) | 1234 | ron |
| [jasmin@gamil.com](mailto:jasmin@gamil.com) | 2704 | jas |
| [Moti@outlook.com](mailto:Moti@outlook.com) | Hello123 | Moti |
| [had@walla.com](mailto:had@walla.com) | 1234 | had |
| [hadas@yahoo.com](mailto:hadas@yahoo.com) | H2905 | Hadas |

מודולים

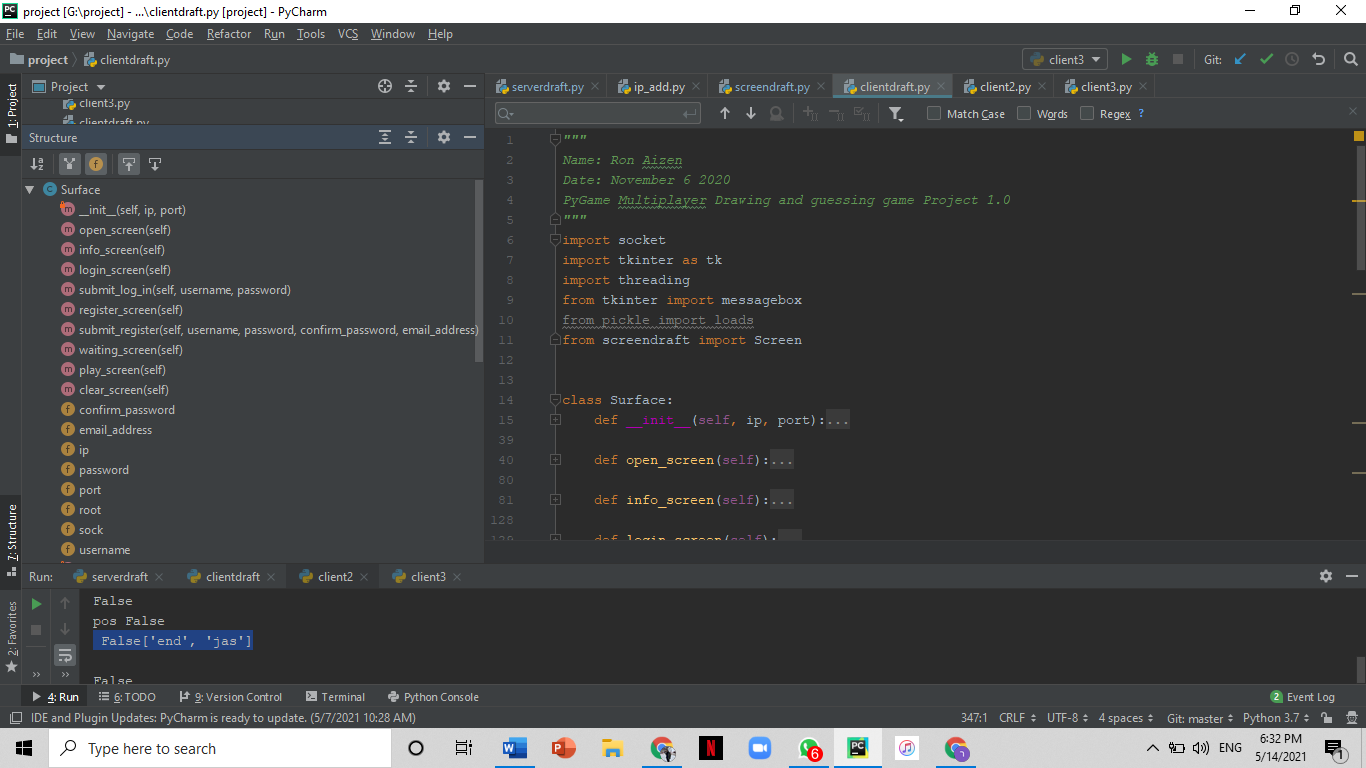
* serverdraft.py- הקובץ הראשי והעיקרי במחשב. כולל את השרת, את פעולותיו, את התקשורת שלו עם הלקוחות וכו'..
* clientdraft.py- הקובץ המכיל בתוכו את הממשק הגרפי ואת הלקוח. הקובץ מתחבר לשרת באמצעות חיבור socket.
* screendraft.py- הקובץ המכיל בתוכו את הקוד שיוצר את המסך עליו מצייר המשתמש.
* users.py- הקובץ שפונה אל מאגר הנתונים, מכניס משתמשים, שולף משתמשים וכו'...
* users.db- הקובץ שמכיל את מאגר המשתמשים.

Server



**תרשים מספר 6: המחלקה Server**

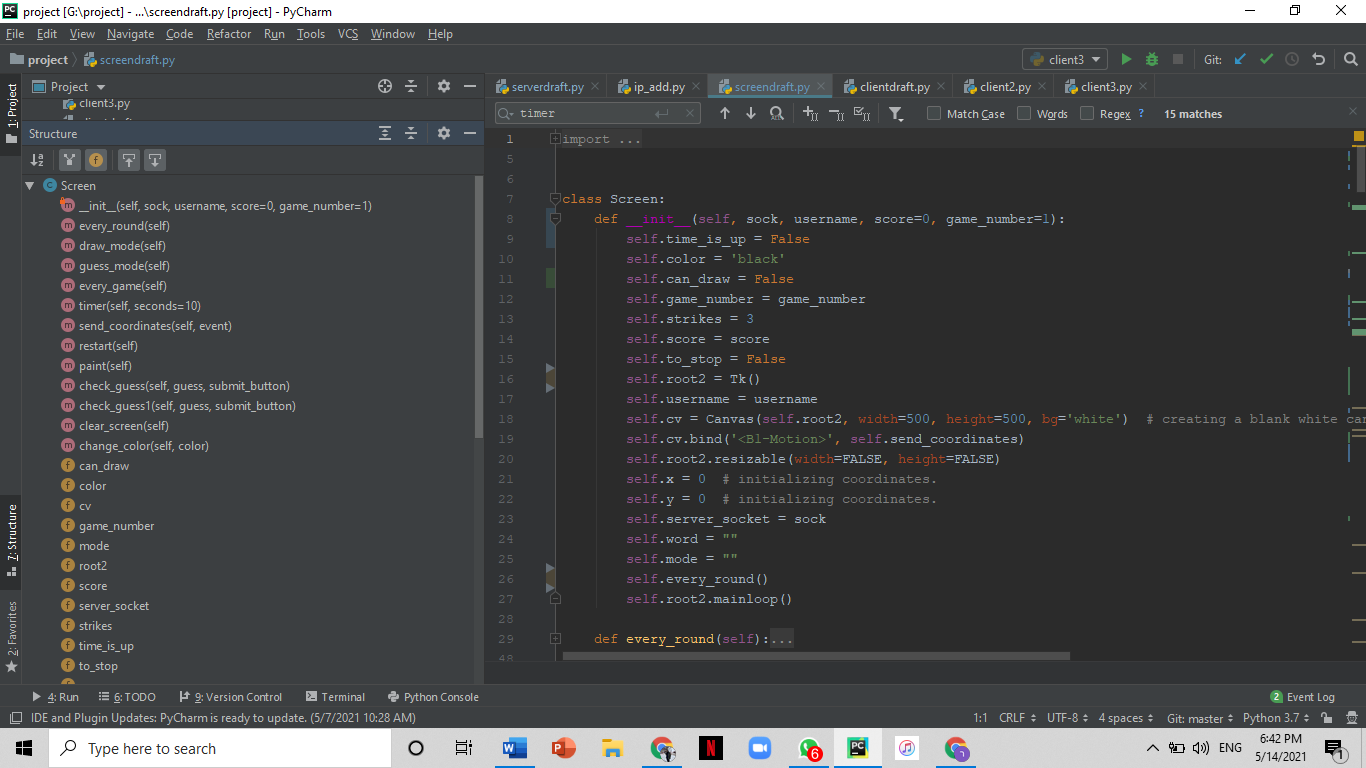
המחלקה server מייצגת, כמובן, את השרת והפעולות השונות שלו. מחלקה זו מכילה את  התכונות השונות של השרת, ואת האופן בו הוא מטפל בלקוחות המתחברים או מתנתקים  ממנו.

Surface

**תרשים מספר 7: המחלקה Surface**

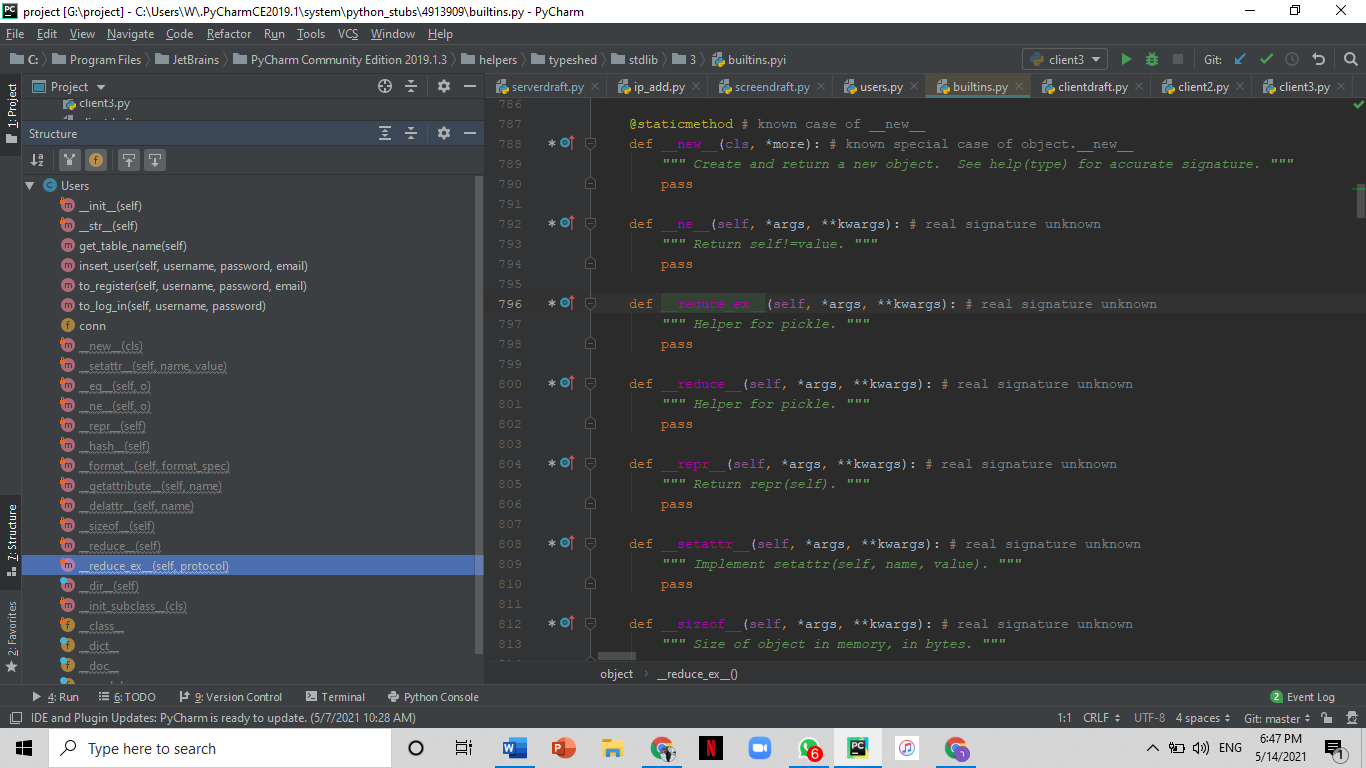
המחלקה Surface מייצגת את הלקוח, הממשק הגרפי והפעולות השונות בהם. מחלקה זו מכילה את התכונות השונות של הלקוח, ואת האופן בו הוא מטופל בעת התחברותו לשרת או בעת התנתקותו.

Screen



**תרשים מספר 8: המחלקה Screen**

המחלקה Screen מייצגת את המשך מחלקת Surface, בעוד שמחלקת Surface אחראית על ההתחברות הראשונית של השחקן לשרת ועל כל השלבים עד לשלב התחלת המשחק (מסך ראשי, התחברות/ הירשמות המשתמש ומסך ההמתנה עד להתחלת המשחק עצמו) מחלקת Screen אחראית על הממשק הגרפי ומסך המשחק עצמו (הקנבס עליו מצייר המשתמש) והפעולות השונות בהם. מחלקה זו מכילה את תכונות שונות של משתמש המשחק, ביניהן המסך עליו מציירים בעת המשחק, המילה שצריך לצייר באותו הסיבוב, הניקוד של אותו משתמש והאם הוא 'צייר' או 'נחשן' .

Users

**תרשים מספר 9: המחלקה Users**

המחלקה אחראית על הכנסת המשתמש למאגר המשתמשים, יצירת משתמש, וידוא הסיסמה המוכנסת בעת התחברותו וכל הקשור במאגר המשתמשים של המשחק.

אלגוריתם המערכת

Timeline

Description automatically generatedאלגוריתם השרת

**תרשים מספר 10: תרשים זרימה של אלגוריתם השרת**

אלגוריתם לקוח

Timeline

Description automatically generated

**תרשים מספר 11: תרשים זרימה של אלגוריתם הלקוח**

הגדרת התקשורת (RFC)

כל הפקודות הן מסוג String.

מבנה הפקודות הנשלחות בין השרת ללקוח

|  |  |
| --- | --- |
| **Parameter1** | **Command** |

**תרשים מספר 12: מבנה פקודת שרת-לקוח.**

כל אחד משדות הטבלה מופרד על ידי ; כך שהפקודה הסופית תיראה כך:

**"command; parameter1"…**

הפקודות של שרת-לקוח

|  |  |  |
| --- | --- | --- |
| פקודה | Command | פרמטרים |
| שליחת ניקוד | Score | ניקוד |
| התחלת המשחק | Play | - |
| סיום המשחק | End | - |
| הרשמה | Register | שם משתמש, סיסמה ואימייל |
| התחברות | Login | שם משתמש וסיסמה |
| הקצאת צייר | Draw | - |
| הקצאת מנחשים | Guess | - |
| שליחת קואורדינטות | X; Y | - |

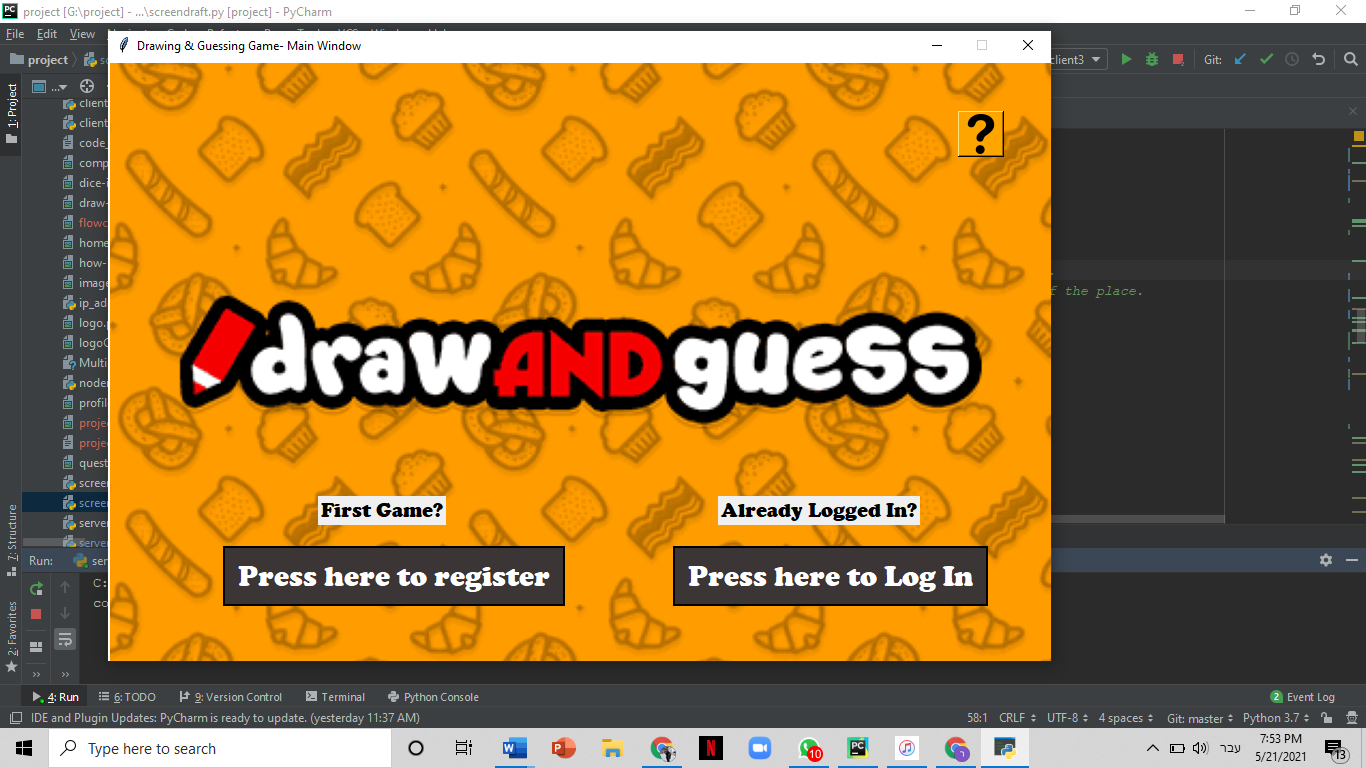
**תרשים מספר 13: רשימת פקודות שרת-לקוח.**

תיאור ממשק המשתמש

התקנה

יש להתקין את הקבצים על המחשב, ולהריץ את קובץ השרת 'Serverdraft.py' במחשב אחד ומשאר המחשבים להריץ את הלקוח 'Clientdraft.py'

ממשק המשתמש



1

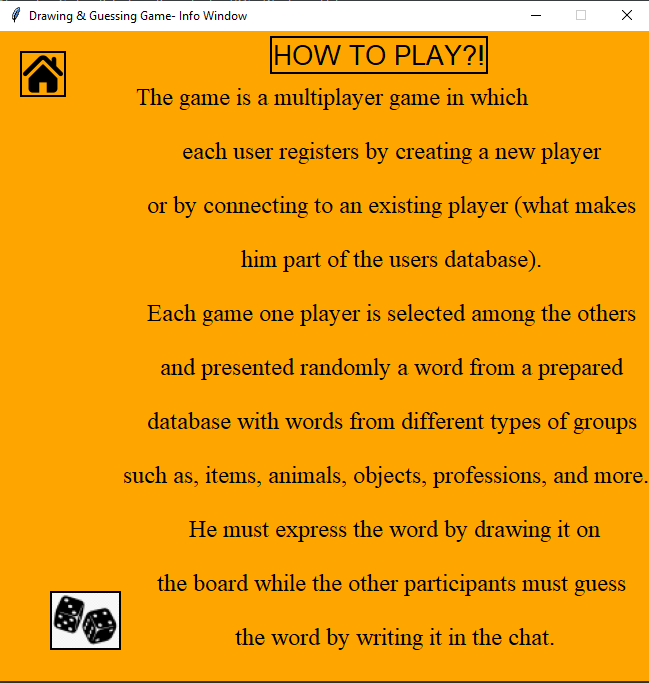
3

2

**תרשים מספר 14: תמונת מסך הפתיחה בממשק המשתמש.**

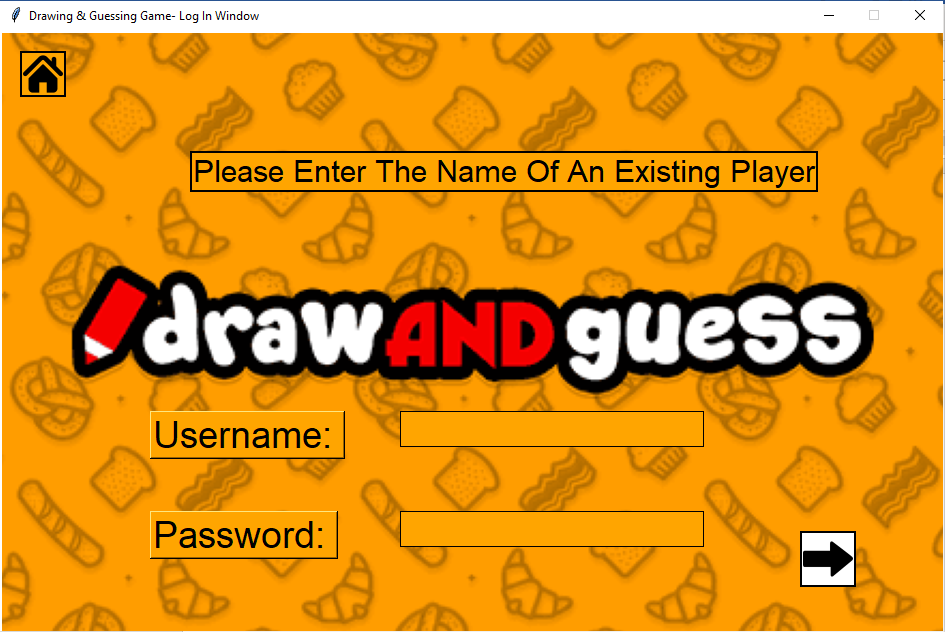
הכפתורים ופעולותיהם

1. כפתור הפותח מסך ובו מידע על המשחק והסבר כיצד פועל המשחק.
2. כפתור המעביר למסך הכניסה, לאנשים בעלי משתמש קיים.
3. כפתור המעביר למסך ההרשמה, לאנשים ללא משתמש.

****

לחצן בית, מחזיר את השחקן למסך הפתיחה

**תרשים מספר 15: תמונת מסך ההוראות בממשק המשתמש.**

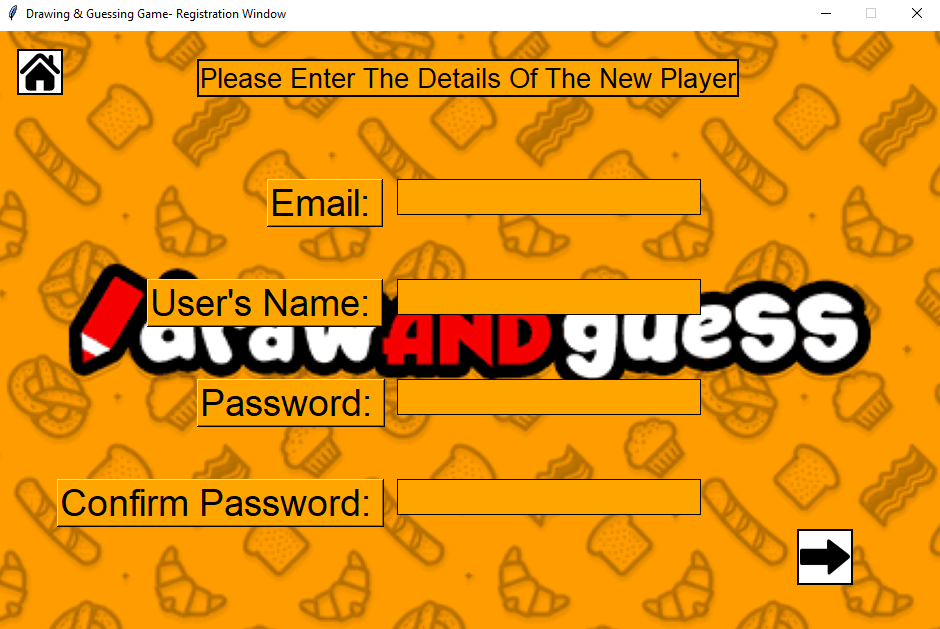


לחצן להכנסת הנתונים ואישורם

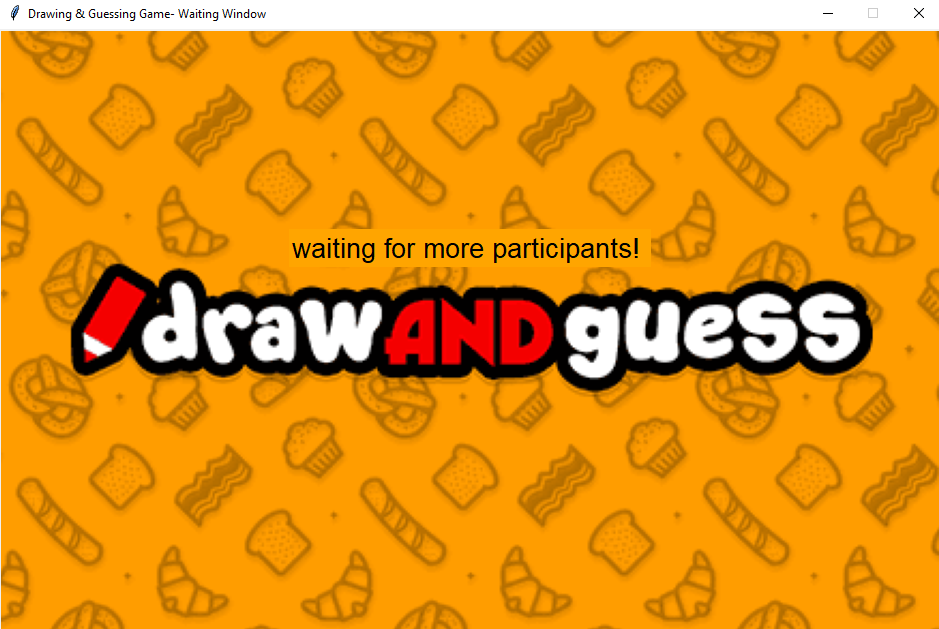
מקום להכנסת שם משתמש של שחקן קיים

**תרשים מספר 16: תמונת מסך ההתחברות בממשק המשתמש.**

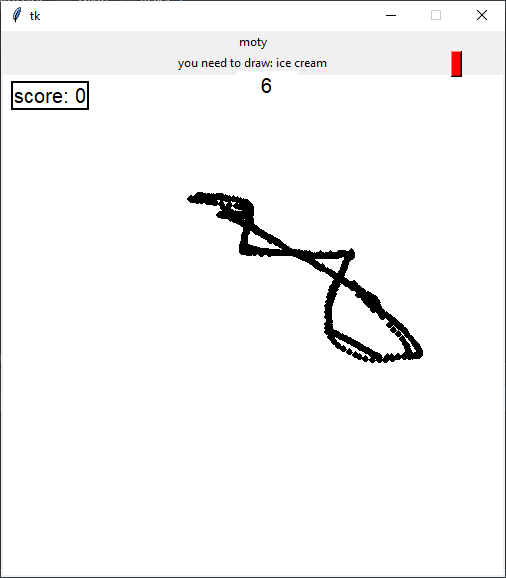
מקום להכנסת סיסמה של שחקן קיים



**תרשים מספר 17: תמונת מסך ההרשמה בממשק המשתמש**

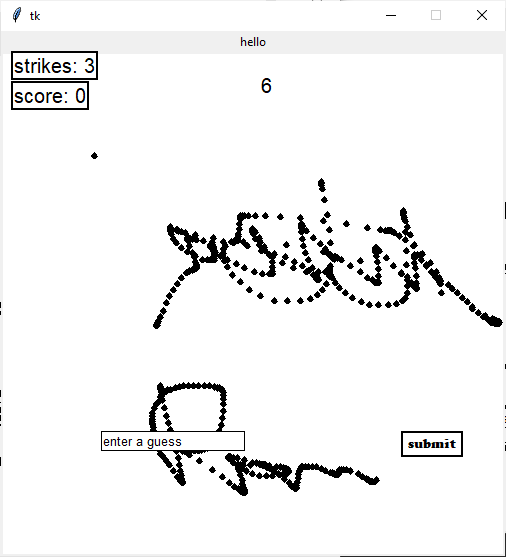


**תרשים מספר 18: תמונת מסך ההמתנה בממשק המשתמש**



**תרשים מספר 19: תמונת מסך הצייר בממשק המשתמש**

ניתן לראות את שם השחקן בראש המסך, את המילה אותה צריך לצייר מתחת (גלידה), שעון טיימר המורה על הזמן הנשאר לסיבוב (6 שניות) ואת הניקד שיש לו.

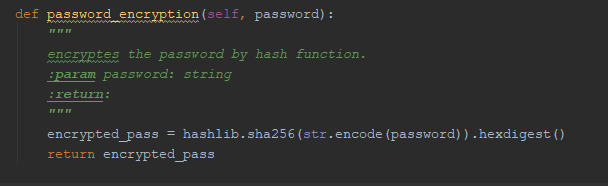


**תרשים מספר 20: תמונת מסך המנחש בממשק המשתמש**

ניתן לראות בראש המסך את שם השחקן, מתחתיו את הזמן הנותר לסיבוב, מצד שמאל יש את כמות הפסילות הנותרות (3) ומתחת יש את הניקוד. בתחתית המסך יש המקום להקליד את הניחוש ומימין את כפתור שליחת הניחוש במידה והניחוש שגוי תופיע הודעה בהתאם על המסך וכמות הפסילות הנותרות תקטן ב1, במידה ותגיע ל0 תיסגר האופציה של השחקן לנחש ותעלם התיבה להקלדת הניחוש בתחתית המסך. במידה והניחוש נכון יעודכן ניקוד השחקן בהתאם.

קטעי קוד מיוחדים

שימוש בפונקציית Hash להצפנה:



**תרשים מספר 15: קוד הצפנה באמצעות פונקציית Hash.**

הפעולה מקבלת את הסיסמה של השחקן, בין אם בעת כניסה למשתמש קיים ובין אם בעת ההרשמה כשחקן חדש ומצפינה את הסיסמה לפי פונקציית Hash נעשה שימוש במודול hashlib.  
הסיסמה נשמרת במאגר הנתונים כמוצפנת על מנת למנוע חשיפה של מידע רגיש, לא ניתן לפענח את ההצפנה אלא רק להשוות בין מידע הנכנס לא מוצפן ולהצפינו ולקוות להתאמה.

שימוש בbind על מנת להשיג את הקואורדינטות של העכבר בעת לחיצה



**תרשים מספר 16: שימוש במתודת bind.**

בעת השימוש במתודה bind עם הערך '<B1-Motion>' בכל פעם שתתבצע לחיצה על המקש הימני בעכבר תוך גרירתו במסך תתבצע שליחה של הקואורדינטות (ה-events של אותו ה-bind) לפעולה הפנימית במחלקה send\_coordinates ובאמצעותה יישלח המידע לכלל השחקנים במשחק. קטע קוד זה הוא התשתית למשחק אותו פיתחתי.

יומן רפלקציה

תיעוד מחקר

בחירת GUI

תיעוד הבעיה למחקר:

בפרויקט שביצעתי בכיתה יא' עשיתי שימוש בממשק גרפי באמצעות המודול Pygame, לכן הייתה התלבטות האם ללכת על "הישן והמוכר" או שמא לנסות וללמוד מודול אחד וחדשני יותר, כמו כן היה לי צורך לשימוש ב-tkinter בשביל הצייר ושליחת מיקום העכבר בכל רגע.

חומרי עזר

אתרים רבים באינטרנט יחד עם סרטוני הדרכה, חיפוש שאלות ספציפיות ובעיקר המצגות שהועלו למרחב הכיתתי שלנו בGOOGLE CLASROOM המסייעות בעבודה עם tkinter.

מסקנה

להשתמש במודול, ותמיד לשאוף ללמוד ולהתפתח.

רפלקציה

נהניתי מעשיית הפרויקט, לא אשקר היו עליות ומורדות. זאת הפעם הראשונה בה הכנתי פרויקט רציני ומשמעותי בהיקף כזה רחב המשלב כל כך הרבה מושגים ונושאים שלמדתי בשנים הקודמות. במהלך הפרויקט למדתי כיצד להתמודד עם בעיות לא צפויות, כיצד לחקור נושאים שונים וכיצד להכין תיק פרויקט. בנוסף, למדתי כיצד לתכנת ולתכנן מערכות באופן טוב יותר, המאפשר עבודה יעילה, מהירה, ופשוטה ברגע שבסיס המערכת גמור. יתר על כן, כמובן, גיליתי בעקבות המחקר מודולים ושיטות תכנות, שאני סמוך ובטוח שישמשו אותי גם בעתיד. לסיכום, נהניתי ולמדתי רבות מהמחקר ומהעבודה על הפרויקט, ואני מאוד שמח מהתוצר שהצלחתי להכין.

תודות

ברצוני להודות למספר אנשים שסייעו לי בתהליך ההכנה של הפרויקט. בראש ובראשונה תודה למוטי מתתיהו המורה שלי לסייבר והמנחה שלי בפרויקט שתמיד סיפק אוזן קשבת, עזר בעת הצורך ונתן רעיונות לשיפור והתקדמות בפרויקט.

כמו כן, תודה לתלמידי הכיתה שעזרו בכלל הפרויקט וביניהם תודה מיוחדת ליסמין שלקחה חלק מיוחד בעזרה בפרויקט שלי.

וובליוגרפיה

* [תהליכונים](https://he.wikipedia.org/wiki/%D7%AA%D7%94%D7%9C%D7%99%D7%9B%D7%95%D7%9F)
* [תקשורת שרת לקוח](https://he.wikipedia.org/wiki/%D7%A9%D7%A7%D7%A2_(%D7%AA%D7%A7%D7%A9%D7%95%D7%A8%D7%AA_%D7%9E%D7%97%D7%A9%D7%91%D7%99%D7%9D))
* [שרת לקוח](https://he.wikipedia.org/wiki/%D7%A9%D7%A8%D7%AA%E2%80%93%D7%9C%D7%A7%D7%95%D7%97)
* [GUI - Tkinter](https://docs.python.org/3/library/tkinter.html)
* [מסדי נתונים](https://www.sqlitetutorial.net/sqlite-python/)

נספחים

קישורים של הפרויקט

קישור לMVP שיצרתי:

<https://drive.google.com/file/d/1XJx6nvEZ9YzvY6THOetSCMkC7tsHH2eD/view?usp=sharing>

קישור לקבצי הפרויקט בגיט-האב:

<https://github.com/ron1506/DrawAndGuess>

מילון מונחים

* פונקציית Hash- <https://he.wikipedia.org/wiki/%D7%A4%D7%95%D7%A0%D7%A7%D7%A6%D7%99%D7%99%D7%AA_%D7%92%D7%99%D7%91%D7%95%D7%91>