תיק פרויקט

שם בית ספר: תיכון על שם דוד קלעי.

שם העבודה: משחק סטרטגו.

שם התלמיד: טל דיבון.

ת"ז של התלמיד:214159873.

שם המנחה: איתי בן סעדון.

שם החלופה: הגנת סייבר (תכנון תוכנות מערכות).

תאריך הגשה:



תוכן עניינים

מבוא

שם פרויקט: משחק סטרטגו.

הקדמה: מאז שהייתי קטן אהבתי לשחק משחקי קופסא. אם זה מונופול, דמקה או סולמות ונחשים.

הייתי משחק במשך שעות עם חברים ומשפחה ואחד המשחקים שאהבתי במיוחד היה משחק הסטרטגו. המשחק מיועד לשני שחקנים ואורכו בין עשר לעשרים דקות. אני חושב שמה שמשך אותי למשחק היה האסטרטגיה הרבה שהיה צריך להפגין בשביל לנצח והעבודה שהמשחק לא דורש מזל בכלל.

ניסיתי כמה וכמה פעמים למצוא אפליקציה או אתר אינטרנטי שבו אוכל לשחק את המשחק מול שחקנים מרחבי העולם. אבל אני אישית לא התחברתי למה שמצאתי באינטרנט ולפי דעתי הגרפיקה של המשחק לא הייתה מספקת.

התחלתי לתכנת בגיל צעיר יחסית אבל מעולם לא היה לי מורה או תוכנית למידה מסודרת לכן עד התיכון, התכנות היה בסך הכל אחד התחביבים שלי. לאחר העלייה לתיכון נרשמתי למגמת הסייבר. שם לראשונה גיליתי שאני באמת אוהב לתכנת ושאני רוצה להמשיך לעסוק בזה בצבא ואז בחיים.

במסגרת המגמה התלמידים צריכים להגיש עבודה גמר בהיקף 5 יחידות לימוד. בתחילת השנה התחלתי להעלות רעיונות למה יהיה פרויקט הסיום שלי. בהתחלה, לא חשבתי על המשחק וחשבתי לבנות משהו אחר. לאחר שחבר סיפר שהוא רוצה לבנות משחק שש בש, חשבתי על הרעיון שגם אני אבנה משחק. ישר קפץ לי לראש משחק הסטרטגו והחלטתי שזה יהיה הפרויקט שלי.

הגדרת הלקוח: קהל היעד של המשחק הוא אנשים שאוהבים אסטרטגיה ומשחקי קופסה ושיש להם כרבע שעה פנויה. המשחק הינו מתאים לכל הגילאים. כל מה שצריך בשביל לשחק במשחק הוא מכשיר אלקטרוני כמו מחשב ואייפד וחיבור לאינטרנט.

אתגר מרכזי: האתגר המרכזי בשבילי היה לבצע חיבור בין השרת ללקוח. עשיתי כמה משימות במהלך השנתיים האחרונות בפייתון שכוללת שרת ולקוח. החלטתי את הצד לקוח לכתוב בגאווה סקריפט ולא בפייתון מכיוון שתכננתי בשפה זו בעבר ואהבתי את הפשטות שבו אפשר ליצור אתרים. אבל אף פעם לא למדתי לבצע תקשורת בגאווה סקריפט אז האתגר המרכזי הראשון שלי היה ללמוד איך לבצע חיבור בין שרת בפייתון ללקוח בגאווה סקריפט.

אתגר נוסף מרכזי בשבילי היה לייצר שרת מרובה משתתפים שידע להבדיל בין השחקנים והמשחקים השונים. בשביל להתגבר על הקושי יצרתי מחלקה שמטרה לנהל משחק (שעליה אני ארחיב בהמשך). במחלקה של השרת יצרתי מילון שהמפתח שלו הוא מזהה ייחודי והערך שלו הוא מופע של המחלקה "משחק". ככה שבכל פעם שיש בקשה לשרת לא צריך לעבור על כל המשחקים ולבדוק מה המשחק המתאים אלא ישירות לגשת עליו. ככה, השרת מוצא את המשחק במהירות רבה יותר וכשיש אלפי או עשרות אלפי משחקים בו זמנית החסכון הזה הוא הכרחי.

הגדרת יעדים:

1) הכנת משחק סטרטגו בין שני משתתפים.

2) הכנת צאט שבו שני השחקנים יוכלו לדבר.

3) הכנת שרת רב משתתפים שידע להריץ מספר משחקנים בו זמנים.

4) דף ליצירת משתמש ודף רישום.

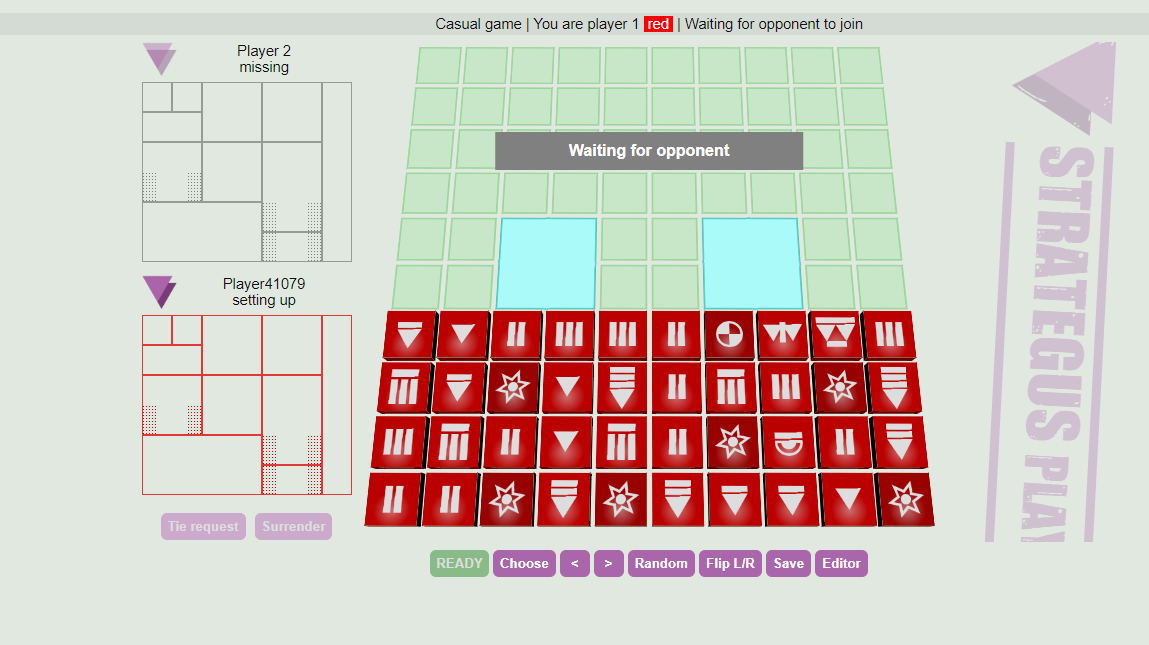
5) שמירה מוצפנת של משתמשים.

6) קשר מוצפן ומאבטח עם השרת אשר מאחסן את המידע.

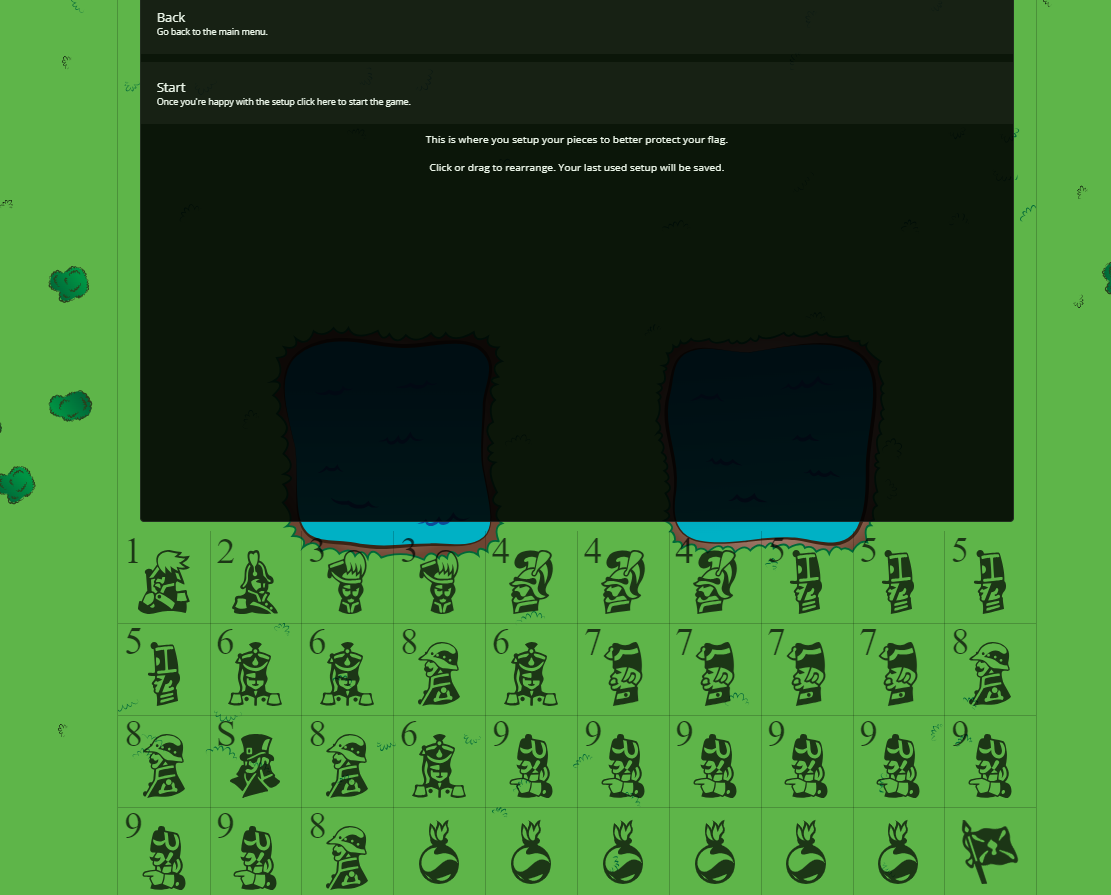
סיקור מצב השוק כיום:

ישנם מספר אתרים שבהם אפשר לשחק סטרטגו באינטרנט אך בעיני הם לא נוחים ולא אטרקטיביים. לדוגמא:

<https://strategus.appspot.com/play>



<https://www.stratego.io/>



אלה דוגמאות למשחקים הקיימים בשוק. כשמחפשים סטרטגו אונליין שני האתרים הם בין האתרים הראשונים שעולים. בשביל למצוא שחקן אחר שישחק מולי רוב הפעמים צריך לחקות כמה דקות, בנוסף אני לא אוהב את הגרפיקה שהמשחקים מציעים.

אפיון הפרויקט

תיאור מפורט של המערכת- המשתמש מתחיל בדף הבית בו הוא יכול להירשם לאתר או ליצור משתמש חדש. בשני המקרים נשלחת בקשה לשרת שבודק את תקינות הבקשה. במקרה של יצירת משתמש. קודם כל נבדק שהשם משתמש והסיסמא עומד בדרישות שיפורטו בהמשך. במידה והמשתמש עומד בכל הדרישות נוצר משתמש חדש. השרת מחזיר ללקוח שהמשתמש נוצר שמציג הודאה למשתמש.

בנוסף, הלקוח עושה על הסיסמה פונקציית האש (hash) ככה שבמסד הנתונים נשמר רצף אותיות ומספרים ולא הסיסמא עצמה. זאת על מנת שבמקרה שפורצים למסד הנתונים הפורץ לא ימצא את הסיסמאות.

במקרה שהמשתמש כבר רשום וברצונו להיכנס למשתמש שלו. הוא שולח בקשת הרשמה. השרת פונה למסד הנתונים ובודק האם קיים משתמש והאם הסיסמא (המוצפנת ולא הסיסמה עצמה) היא אותו הדבר. במידה שכן הוא שולח שהמשתמש התחבר.

בנוסף יכול המשתמש לצפות בחוקי המשחק הכתובים בשפה האנגלית בשביל להיזכר או ללמוד.

לאחר ההרשמה המשתמש יכול ללחוץ על כפתור החל משחק. המחשב מחכה עד ששחקן שני מחכה, ולאחר מכן יוצר משחק חדש.

לאחר המשחק לשני השחקנים 5 דקות לארגן את הכלים שלהם בצורה שלפי דעתם תהיה הטובה ביותר. לאחר חמש הדקות כל שחקן משחק בתורו. המנצח הוא השחקן שמצא את דגל היריב.

לרשות השחקנים צאט שבו יוכלו לדבר אחד עם השני במהלך המשחק. בצאט מוצג מי מהשחקנים כתב מה.

לאחר סיום המשחק לשחקן המנצח יוספו נקודות ולשחקן המפסיד ירדו נקודות מהדירוג שלו.

פירוט היכולות:

1) הרשמה לאתר.

2) יצירת משתמש חדש.

3) צפייה בדף ההוראות.

4) משחק חדש דרך האינטרנט.

5) דיבור עם השחקן היריב דרך צאט.

פירוט הבדיקות (קופסא שחורה):

א. יצירת משתמש חדש - בדיקת הדרישות הבאות:

1) שם המשתמש והסיסמא הם בעלי 3 ספרות לפחות ומתחת ל16 ספרות.

2) שם השתמש אינו תפוס.

3) שם המשתמש והסיסמא לא מכילים רווח.

בנוסף בעת יצירת משתמש חדש על המשתמש להקליד את הסיסמא פעמיים. זאת על מנת לוודא שהמשתמש לא טעה בטעות בכתיבת הסיסמא.

ב. התחברת למשתמש קיים:

ראשית נבדקים אותם התנאים (תנאים 1-3) של יצירת משתמש חדש. נוסף על כן נבדק ששם המשתמש והסיסמא שהוקלדו מופיעים במסד הנתונים.

\* בשני הבדיקות יוצג למשתמש הודעה מתאימה שתואמת את הבעיה או כאשר הבדיקה הסתיימה בהצלחה.

לאחר שהמשתמש נכנס יוצג לפניו כפתור של התחל משחק שבלחיצה עליו יתבצע הבדיקה הבאה.

ג. בדיקת האם התחיל המשחק.

לאחר לחיצת הכפתור כל שניה תשלח לשרת בקשה של האם המשחק התחיל.

השרת מחכה עד ששחקן שני גם הוא שולח את הבקשה. עד שנמצא השחקן השני השרת יענה שלא נמצא שחקן שני. לאחר ששחקן שני נכנס השרת יגריל מזהה ייחודי למשחק ויחזיר אותו לשני השחקנים. השרת גם יוצר משחק חדש.

לאחר החזרת המזהה הייחודי תתבצע הבדיקה הבאה.

ד. בדיקת עדכון המשחק.

כל שנייה נשלח בקשה המורכבת מתתי הבקשות הבאות:

1) קבלת הלוח כמערך דו ממדי- בדיקה האם השחקן השני שיחק.

2) קבלת המשפטים האחרונים שהיו בצאט.

3) סטטוס המשחק. האם המשחק עדיין ממשיך או שנגמר.

4) הזמן של השחקנים-לשחקנים זמן מוגדר לסדר את הלוח שלהם (חמש דקות).

ה. בדיקת האם תורי.

כששחקן רוצה לבצע מהלך הוא שלוח בקשה של האם תורי לשחק במקרה שחוזר נכון השחקן שולח את המהלך ובמקרה שחוזר שקר המהלך לא מבוצע.

ו. בדיקת של האם הצעד שבוצע הוא מהלך חוקי.

למשחק חוקים שקובעים איך כל כלי יכול לזוז. אז לפני שמזיזים את הכלי יש לבדוק האם המהלך שנעשה הוא נכון. לפי החוקים הבאים:

1. פצצה ודגל לא יכולים ללכת.

2. שחקנים לא יכולים ללכת יותר ממשבצת אחת. הכלי יכול לזוז במאונך, במאוזן או באלכסון.

3. יוצא מכלל זה הוא כלי 2 כלי זה יכול ללכת כמה משבצות שהשחקן רוצה במאוזן או במאונך בתנאי שהוא לא מדלג על כלי אחר או מעל ים.

4. שחקנים יכולים ללכת רק על היבשה (מסומן במשבצת ירוקה) ולא על הים (מסומן במשבצת כחולה).

תכנון וניהול הלוז.

ראשית בתחילת השנה התחלתי לשבת ולחשוב על רעיון לפרויקט. לאחר שהחלטתי שמשחק הסטרטגו יהיה הפרויקט שלי התחלתי לעבוד על איך המשחק יראה.

בהתחלה יצרתי בכלליות את לוח המשחק ואת התמונות שאני ארצה להשתמש בהם בעתיד. הכנתי את הסידור ההתחלתי של התמונות. קבעתי את מיקום הים והיבשה בהתאם ללוח המשחק. לאחר מכן עבדתי על כך שאצליח להחליף בין התמונות בעזרת העכבר. מה שבעתיד יסמל את הליכת הכלים.

לאחר עשיית הגרפיקה הבסיסית האתגר הבא שלי היה לבצע תקשורת בינתיים לא מוצפנת בין הלקוח שכתוב בגאווה סקריפט לשרת שכתוב בפייתון. אחר כך התחלתי לייצר את המחלקות של הלוח ושל הקלף ואת השרת שיעבד את בקשות הלקוח ויחזיר תשובות.

לאחר מכן התחלתי לבצע משחק בסיסי בין שני לקוחות. השרת עדיין לא תמך במספר משתתפים. ראיתי שאני מצליח לעשות מהלך ושהוא יופיע במשחק של השחקן השני.

החלקים הבאים כללו בדיקה של המהלכים האם הם חוקיים או לא ומה קורה כשכלי אחד תוקף כלי שני. מתן זמן התחלתי לשחקנים לייצר את הלוח לפי רצונם ובדיקת ניצחון והפסד.

לאחר כל השלבים האלה היה לי משחק בסיסי עובד. החלק הבא היה לוודא שהשרת יתמוך במספר משתמשים בו זמנית. הוספת צאט בין כל שני שחקנים שמשחקים בדיוק.

אחרי זה יצרתי את פונקציית ההתחברות וההרשמה, יצרתי את מסד הנתונים. הצפנת הסיסמא, בדיקת תקינות שם המשתמש והסיסמא ואת הדפים שיוצגו ללקוחות שייכנסו לאתר.

במצב הזה היה לי כבר אב טיפוס ראשוני והתחלתי לעבוד על התיק פרוייקט.

לאחר מכן הוספתי הצפנות, שיפרתי את הגרפיקה ועשיתי עוד תיקונים קטנים.

סיכונים:

החלטתי לכתוב את הצד שרת בפייתון מכיוון שזוהי שפת התכנות המרכזית שלמדנו בכיתה בשנתיים האחרונות. אך החלטתי את הצד לקוח לכתוב בגאווה סקריפט, ,css ו- html מכיוון שהם מספקים יכולות גרפיות בצורה פשוטה ואלגנטית.

אחד הסיכונים היה לכתוב את הקוד בשני שפות שונות. לכל שפה חוקים משלה ויתרונות משלה. בנוסף לא למדתי לפני הפרויקט איך מתבצעת התקשורת בגאווה סקירפט. זה היה הימור גם מכיוון שמעולם לא עשיתי את זה קודם.

תיאור תחום הידע

יכולות הלקוח:

שם היכולות: החלפת מפתחות

מהות: החלפת מפתחות התחלתית עם השרת – פעולה אוטומטית

אוסף יכולות נדרשות:

* קליטת מפתח ציבורי של השרת
* הכנת מפתח ציבורי
* הצפנה עם המפתח של הרשת
* שליחת מפתח ציבורי לשרת
* החזרת מפתח סימטרי אישי להמשך תקשורת

• אובייקטים נחוצים: הצפנה / פענוח, תקשורת

שם היכולות: חיבור לשרת

מהות: חיבור לשרת- מתחילה אחרי יצירת משתמש או התחברות.

אוסף יכולות נדרשות:

* התחברות לשרת
* החלפת מפתחות
* קליטת כינוי ממשתמש
* הצפנה
* שליחה לשרת
* הרצת ממשק התחברות

אובייקטים נחוצים: ממשק משתמש, הצפנה /פענוח, תקשורת

שם היכולות: הרשמה למערכת.

מהות: רישום משתמש חדש במערכת – קליטת פרטים אישיים נדרשים

אוסף יכולות נדרשות:

* ממשק משתמש – מסך הרשמה
* קליטת נתונים
* הצפנה
* בדיקת הקלט
* שליחה לשרת
* רשימה במאגרי המידע
* קבלת תשובה מהשרת
* פענוח
* הצגת התשובה למשתמש

אובייקטים נחוצים: ממשק משתמש, הצפנה /פענוח, תקשורת, בסיס נתונים

שם היכולות: התחברות למערכת

מהות: חיבור משתמש קיים למערכת – קליטת פרטים אישיים נדרשים

אוסף יכולות נדרשות:

* ממשק משתמש – מסך התחברות
* קליטת נתונים
* הצפנה
* בדיקת הקלט
* שליחה לשרת המרכזי
* בדיקה מול מאגרי המידע
* קבלת תשובה מהשרת
* פענוח
* הצגת התשובה למשתמש
* טעינת כפתור חדש

אובייקטים נחוצים: ממשק משתמש, הצפנה /פענוח, תקשורת, בסיס נתונים

שם היכולות: הצגת ההוראות

מהות: הצגת ההוראות למשתמש בצורה נוחה.

אוסף יכולות נדרשות:

* ממשק משתמש-דף הוראות
* טעינת תמונות וכפתורים.
* עיבוד הלחיצה וחזרה לדפים אחרים.

אובייקטים נחוצים: ממשק משתמש.

שם היכולות: התחברות למשחק חדש.

מהות: בקשה להתחיל משחק- לחקות עד ששחקן אחר יצטרף.

אוסף יכולות נדרשות:

* ממשק משתמש – לחיצה על כפתור
* הצפנה
* שליחה בקשה לשרת המרכזי
* המתנה לשחקן חדש
* קבלת תשובה מהשרת
* פענוח
* הצגת התשובה למשתמש
* טעינת משחק

אובייקטים נחוצים: ממשק משתמש, הצפנה /פענוח, תקשורת

שם היכולות: הזזת כלי.

מהות: הזזת הכלי- השחקן גורר את הכלי ממשבצת אחת לשנייה.

אוסף יכולות נדרשות:

* ממשק משתמש – הצגת הלוח וגרירת התמונה
* הצפנה
* שליחת הודעה לשרת המרכזי
* פענוח המעבר (האם חוקי או לא)
* קבלת תשובה מהשרת
* פענוח
* שינוי הלוח
* טעינת משחק

אובייקטים נחוצים: ממשק משתמש, הצפנה /פענוח, תקשורת

שם היכולות: דיבור דרך הצאט.

מהות: בקשה להתחיל משחק- לחקות עד ששחקן אחר יצטרף.

אוסף יכולות נדרשות:

* ממשק משתמש – קליטת הטקסט והצגתו למשתמשים
* הצפנה
* שליחת הטקסט לשרת המרכזי
* קבלת תשובה מהשרת
* פענוח
* הצגת התשובה למשתמש
* טעינת הטקסט החדש לשני המשתמשים.

אובייקטים נחוצים: ממשק משתמש, הצפנה /פענוח, תקשורת

יכולות צד שרת:

שם היכולות: החלפת מפתחות

מהות: החלפת מפתחות עם לקוח כאשר הוא מתחבר

אוסף יכולות נדרשות:

* הכנת מפתח ציבורי ופרטי
* שליחת מפתח ציבורי ללקוח
* פענוח קלט עם מפתח פרטי
* החזרת מפתח סימטרי של לקוח להמשך תקשורת

אובייקטים נחוצים: הצפנה / פענוח, תקשורת.

שם היכולות: חיבור לקוח חדש

מהות: חיבור לקוח חדש למערכת. הלקוח שולח את שם המשתמש שלו.

אוסף יכולות נדרשות:

* האזנה
* התחברות המשתמש
* החלפת מפתחות
* קליטת שם משתמש מהלקוח
* פענוח
* המתנה ללקוח נוסף על מנת להתחיל משחק ביניהם.

אובייקטים נחוצים: הצפנה / פענוח, תקשורת

שם היכולות: אימות משתמש

מהות: אימות פרטי התחברות של משתמש עם מאגר נתונים – קבלת פלט מלקוח נדרש

אוסף יכולות נדרשות:

* קבלת בקשה מהמשתמש
* פענוח
* בדיקת תקינות הבקשה
* בדיקה עם מאגר הנתונים
* ההצפנה
* החזרת תשובה למשתמש

אובייקטים נחוצים: הצפנה / פענוח, תקשורת, בסיס נתונים

שם היכולות: רישום משתמש

מהות: רישום פרטי התחברות של משתמש במאגר נתונים – קבלת פלט מלקוח נדרש

אוסף יכולות נדרשות:

* קבלת בקשה מהמשתמש
* פענוח
* בדיקת תקינות הבקשה
* רשימה במאגר הנתונים
* הצפנה
* החזרת תשובה למשתמש

• אובייקטים נחוצים: הצפנה / פענוח, תקשורת, בסיס נתונים

שם היכולות: יצירת משחק חדש

מהות: התחלת משחק עם שני לקוחות.

אוסף יכולות נדרשות:

* החזרת מידע ללקוח
* פענוח
* בדיקת תקינות הבקשה
* שמירת המשחק והפרדתו ממשחקים אחרים
* הצפנה
* שימוש בthreads

אובייקטים נחוצים: הצפנה / פענוח, תקשורתthreads,.

שם היכולות: צאט בין משתתפים

מהות: יצירת צאט כך שהשחקנים שמשחקים יחדיו יוכלו לדבר.

אוסף יכולות נדרשות:

* החזרת מידע ללקוח
* פענוח
* שמירת הצאט והפרדתו מצאטים אחרים
* הצפנה
* החזרת תשובה למשתמש

אובייקטים נחוצים: הצפנה / פענוח, תקשורת.

שם היכולות: החזרת הלוח

מהות: החזרת הלוח למשתמש כמערך דו ממדי

אוסף יכולות נדרשות:

* החזרת מידע ללקוח
* פענוח
* בדיקת הבקשה
* החזרת לוח ספציפי
* הצפנה

אובייקטים נחוצים: הצפנה / פענוח, תקשורת.

שם היכולות: שינוי זמן והחזרתו.

מהות: החזרת הזמן ללקוח עד להתחלת המשחק וסיום זמן ההתארגנות.

אוסף יכולות נדרשות:

* שינוי הזמן
* החזרת הזמן בצורה נוחה
* החזרת הזמן ללקוח
* פענוח
* בדיקת הבקשה
* הצפנה
* שימוש בThreads.

אובייקטים נחוצים: הצפנה / פענוח, תקשורת, threads.

חלק :3 מבנה – ארכיטקטורה

תיאור הטכנולוגיה הרלוונטית:

שפת תכנות – Python. כל הצד שרת שלי כתוב בפייתון מלבד הפקודות שקשורות למסד הנתונים שכתובים ב- SQLite. פייתון היא שפה פשוטה למדי אשר מאפשר כתיבה נוחה יותר של שפת הקוד. בחרתי בשפה זו מכיוון שזוהי שפת התכנות העקרית שלמדנו במגמה ובגלל הפשטות ומרחב הכלים שהיא מציעה.

שפת תכנות javascript- JavaScript היא אחת משפות הפיתוח הכי פופולריות בתעשייה. כיום JavaScript משמשת בין היתר לפיתוח אתרים ואפליקציות full stack . JavaScript היא שפת תכנות דינמית המאפשרת להוסיף פונקציונליות אינטראקטיבית לדפי אינטרנט. ניתן להשתמש בה כדי ליצור אנימציות, משחקים, טפסים ועוד.

Html- היא שפת תכנות שבעזרתה ניתן לבנות שלד של עמודי אינטרנט ולהכניס את כל הקוד. באמצעות השפה נבנים הרבה מאוד אתרי אינטרנט ברחבי העולם.

- Css היא שפת תכנות המשמשת לעיצוב חזותי של דפי אינטרנט. היא מאפשרת לשלוט בצבעים, בגופנים, בפריסה ועוד, ובכך מעניקה לאתר מראה ייחודי ומותאם אישית.

בעזרת JavaScript html ו-CSS כתבתי את הצד לקוח ואת הדפים המוצגים למשתמשים. Htmlמהווה את השלד של האתר, בעזרת css אפשר לעצב את האתר. JavaScript מוסיפה את "המוח" שמאחורי הקוד.

מסד נתונים – SQLite. מסד הנתונים אשר משמש אותי בכדי לשמור את נתוני

המערכת הוא מסד נתונים מסוג SQLite, אשר מחובר בצד השרת. הוא יחסית פשוט לשימוש.

חבילת http.server בפייתון מספקת שרת HTTP בסיסי וקל לשימוש המיועד לפיתוח ובדיקות מקומיות. היא כוללת מחלקות ומודולים לניהול בקשות HTTP ויצירת תגובות בצורה מהירה וללא צורך בתצורה מורכבת.

חבילתsocketserver בפייתון מספקת מסגרת עבודה לבניית שרתים מבוססי סוקט, עם תמיכה בפרוטוקולים שונים כמו TCP ו-UDP. היא מאפשרת ליצור שרתים יעילים ומותאמים אישית על ידי שימוש במחלקות בסיסיות.

תיאור זרימת המידע במערכת:

יכולות הלקוח:

1. קליטת מידע מהשרת – על הלקוח לקלוט מהשרת את המפתח הציבורי שלו.

יצירת מפתח סימטרי והצפנה שלו עם המפתח שהתקבל מהשרת.

שליחת המפתח המוצפן לשרת.

החזרת מפתח אישי לקוד הלקוח להמשך התקשורת והצפנת הודעות.

2. הרצת ממשק המשתמש לרישום המשתמש.

קבלת קלט של שם משתמש, סיסמה ואימות סיסמה מהמשתמש.

בדיקה שהשם משתמש והסיסמה עומדים בדרישות.

בדיקה שהסיסמה והאימות סיסמה שווים.

הצפנה עם המפתח הסימטרי ושליחה לשרת.

קבלת תשובה מהשרת ופענוח ההודעה.

במידה והמשתמש נקלט העבר לדף הכניסת משתמש ובמידה שיש בעיה להציג הודעה מתאימה.

3. הרצת ממשק המשתמש לכניסת המשתמש.

קבלת קלט של שם משתמש וסיסמה מהמשתמש.

בדיקה שהשם משתמש והסיסמה עומדים בדרישות.

הצפנה עם המפתח הסימטרי ושליחה לשרת.

קבלת תשובה מהשרת ופענוח ההודעה.

הצגת כפתור החל משחק.

4. המתן לתחילת המשחק. מותנה בלחיצת כפתור החל משחק.

יצירת בקשה של האם התחיל המשחק. הבקשה תשלח כל שנייה.

הצפנה עם המפתח הסימטרי ושליחה לשרת.

קבלת תשובה מהשרת ופענוח ההודעה.

במידה שהתחיל משחק תיצור את הלוח ואת הצאט.

במידה ולא שלח בקשה בשנית.

5. עדכן את הלוח, את הזמן ואת הצאט כל שניה.

יצירת בקשה של עדכון משחק. הבקשה תשלח כל שנייה.

הצפנה עם המפתח הסימטרי ושליחה לשרת.

קבלת תשובה מהשרת ופענוח ההודעה.

עדכן את כל הרכיבים ובדוק האם התקבל ניצחון או הפסד.

במידה ולא המשך לשלוח את הבקשה.

6. לאחר קבלת ההודעה של עדכון המשחק. במקרה שנגמר המשחק.

הצג הודעה מתאימה למשתמש.

עדכן את ניקוד המשתמש.

הסר את הלוח ואת כל המרכיבים.

הצג את כפתור החל משחק. בלחיצה עליו חזור לארבע.

יכולות השרת:

1. האזנה לפורט ספציפי. במקרה של המשחק שלי בחרתי בפורט 8000

2. יצירת צמד מפתחות א-סימטריים

שליחה ללקוח של מפתח ציבורי

קליטת הודעה מהלקוח ופענוח עם מפתח פרטי

שמירת המפתח הסימטרי.

3. קליטת בקשה מהמשתמש ופענוח

הפרדת המשתנים השונים שקיבלנו מהמשתמש ובדיקת תקינותם.

4. במקרה שהמשתנה שמחזיק את הפעולה הוא הירשמות לאחר הפענוח.

הפרדת השם משתמש והסיסמה משאר הפרמטרים.

פתיחת מסד הנתונים.

בדיקה ששם המשתמש אינו קיים. במקרה שלא קיים הוסף את שם המשתמש והסיסמה ועדכן את הניקוד ההתחלתי.

סגירת מסד הנתונים.

הכן הודעה מתאימה בהתאם להאם המשתמש נרשם או לא.

הצפנת ההודעה בעזרת המפתח הסימטרי.

החזרת ההודעה ללקוח.

5. במקרה שהמשתנה שמחזיק את הפעולה הוא כניסת משתמש לאחר הפענוח.

הפרדת השם משתמש והסיסמה משאר הפרמטרים.

פתיחת מסד הנתונים.

בדיקה ששם המשתמש קיים והסיסמה המוצפנת שקיבלנו שווה לסיסמה השמורה.

סגירת מסד הנתונים.

הכן הודעה מתאימה בהתאם להאם המשתמש נמצא או לא.

הצפנת ההודעה בעזרת המפתח הסימטרי.

החזרת ההודעה ללקוח.

6. במקרה שהמשתנה שמחזיק את הפעולה הוא לחקות למשחק לאחר הפענוח.

במידה וקיים משתמש אחר שכבר מחכה החל משחק בין שני השחקנים.

במידה ולא קיים משתמש אחר שמור את המזהה של השחקן עד ששחקן אחר נכנס.

במידה והתחיל המשחק צור משתנה יחודי שיהווה את מזהה המשחק.

הצפן אותו והחזר אותו לשני השחקנים.

במידה ולא החזר שעדיין לא נמצא שחקן נוסף (הצפן לפני השליחה).

7. במקרה שהמשתנה שמחזיק את הפעולה הוא לשחק טור למשחק לאחר הפענוח.

מצא את המשחק הספציפי על פי המזהה היחודי שהשחקן שלח.

בדוק האם תורו של השחקן לשחק

בדוק האם הצעד עומד בחוקים של המשחק.

במקרה שכן שחק את הצעד ושמור אותו.

8. במקרה שהמשתנה שמחזיק את הפעולה הוא לשלוח טקסט למשחק לאחר הפענוח.

מצא את המשחק הספציפי על פי המזהה היחודי שהשחקן שלח.

הוסף לטקסט של המשחק את מהשחקן שלח וכתוב וזהה את הטקסט על פי המזהה של השחקן.

9. במקרה שהמשתנה שמחזיק את הפעולה הוא קבלת עדכון על המשחק לאחר הפענוח.

מצא את המשחק הספציפי על פי המזהה היחודי שהשחקן שלח.

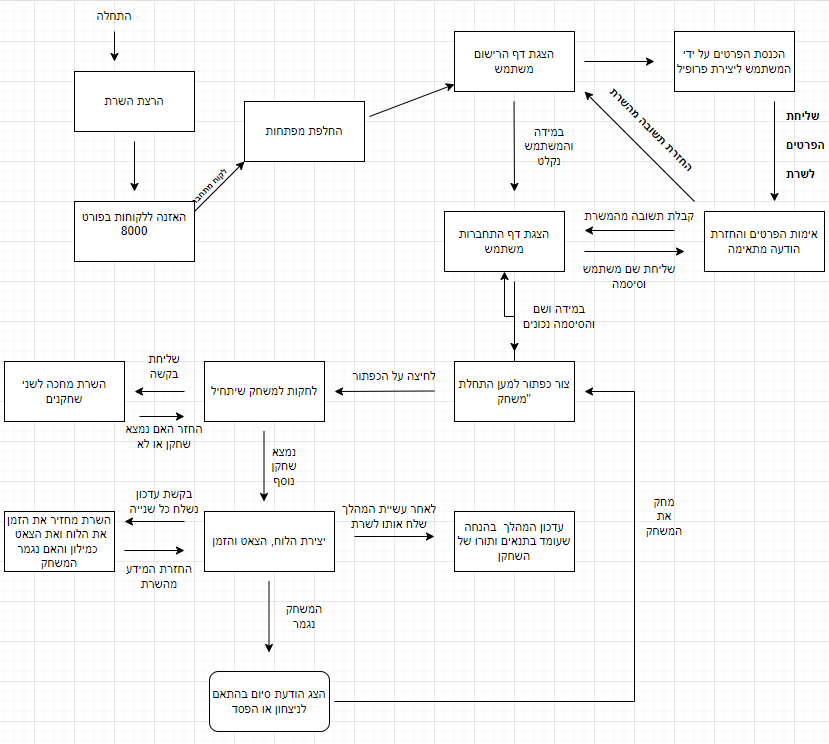
הכן את הלוח כמערך דו ממדי שבו כל מקום הוא סוג הכלי.

הכן את הטקסט כרצף של שורות.

הכן את הזמן של השחקנים.

שמור את כולם כמילון.

הצפן אותו והחזר למשתמש.

הצגת זרימת המידע ביו הרכיבים המרכזיים במערכת באמצעות שרטוט:

תיאור סביבת הפיתוח:

1. Python: הצד שרת ייכתב בשפת התכנות Python, יחד עם הספריות והמודולים

הדרושים לפרויקט שלי.

2. סביבת העבודה שלי לכתיבת השרת היא :Pycharm

Pycharm היא סביבת פיתוח משולבת לפיתוח תוכנות בעיקר בשפת פייתון, אשר פותחה על ידי תאגיד התוכנה הצ'כי JetBrains. הסביבה מספקת שירותים כגון ניתוח קוד וניפוי שגיאות בקוד גרפי. בחרתי בסביבה זו כי היא סביבת עבודה נוחה, מציעה כלים שעוזרים לעבודה וכי למדו לתכנת python בסביבת עבודה זו במסגרת הלימודים שלנו.

3. את הצד לקוח אני כתוב ב html, javascript, css.

4. סביבת העבודה לצד הלקוח הוא notepad++ .Notepad++‎ הוא עורך טקסט ועורך קוד מקור ל-Windows. התוכנה מציעה מגוון רחב של אפשרויות מעבר לפנקס רשימות המובנה בחלונות, כגון שימוש בלשוניות. התוכנה נוחה מאוד לכתיבת צד הלקוח שלי.

5. את הרצת השרת אני מריץ דרך דפדפן .chrome

6. את מסד הנתונים אני מריץ דרך SQLite. SQLite היא מערכת מסד נתונים קלת משקל מבוססת קבצים שבה אעשה שימוש לאחסון מידע משתמשים והניקוד שלהם.

תיאור התקשורת:

לאחר הרצת השרת, השרת מקשיב לפורט מסוים. במקרה שלנו השרת מקשיב לפורט 8000. התקשורת מתבצעת דרך פרוטוקול http. הלקוחות השונים שולחים בקשות אל השרת, השרת מעבד את הבקשות ובחלקן מחזיר תשובה.

לאחר הרצת לקוח, הלקוחות השונים מבקשים מהמשרת את המפתח הציבורי ומקבלים אותו. הם מצפינים בעזרתו את המפתח הסימטרי ומחזירים אותו לשרת מוצפן. השרת

לדוגמא כשהשחקן רוצה לבצע מהלך או לשלוח טקסט השרת לא יחזיר תשובה.

כשמדובר על קבלת מזהה יחודי או קבלת עדכון על המשחק השרת כן יחזיר תשובה.

הבקשות כולן הן בקשות מסוג .getכל לקוח שולח בבקשתו ארבעה פרמטרים שונים.

Action- הפעולה שאנחנו רוצים לבצע. אם מדובר בהרשמה, כניסת משתמש או פעולות הקשורות למשחק.

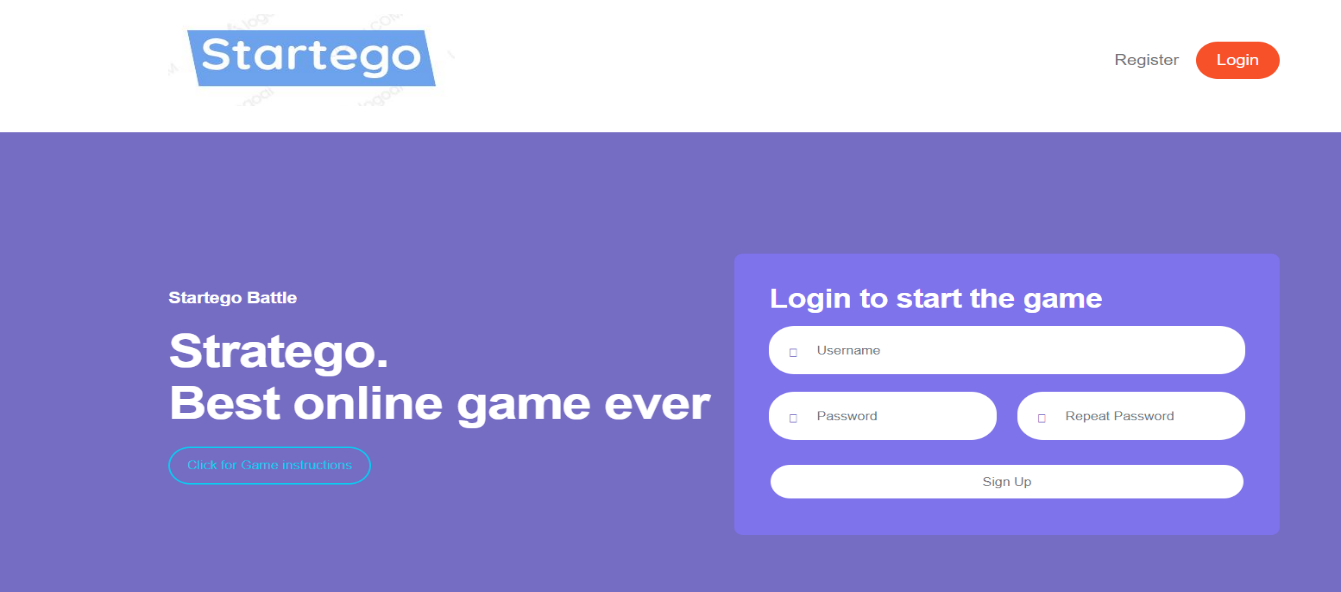
-Value שליחת הפרמטרים השונים הדרושים לכל בקשה. לדוגמה אם הפעולה היא הרשמות אז value יכיל את השם משתמש + הסיסמה. או אם הפעולה היא לשחק טור אז ה value יכיל את המשבצת שממנה הכלי זז למשבצת שלה הכלי יגיע. בחלק מהמקרים ה value ישאר רק כי אין מה לשלוח. כמו בבקשת עדכון המשחק.

gameId- מזהה ייחודי שיבדיל בין המשחקים השונים. כך נדע לדעת כל שחקן לאיזה משחק הוא שייך. עד להתחלת המשחק gameId ישאר ריק.

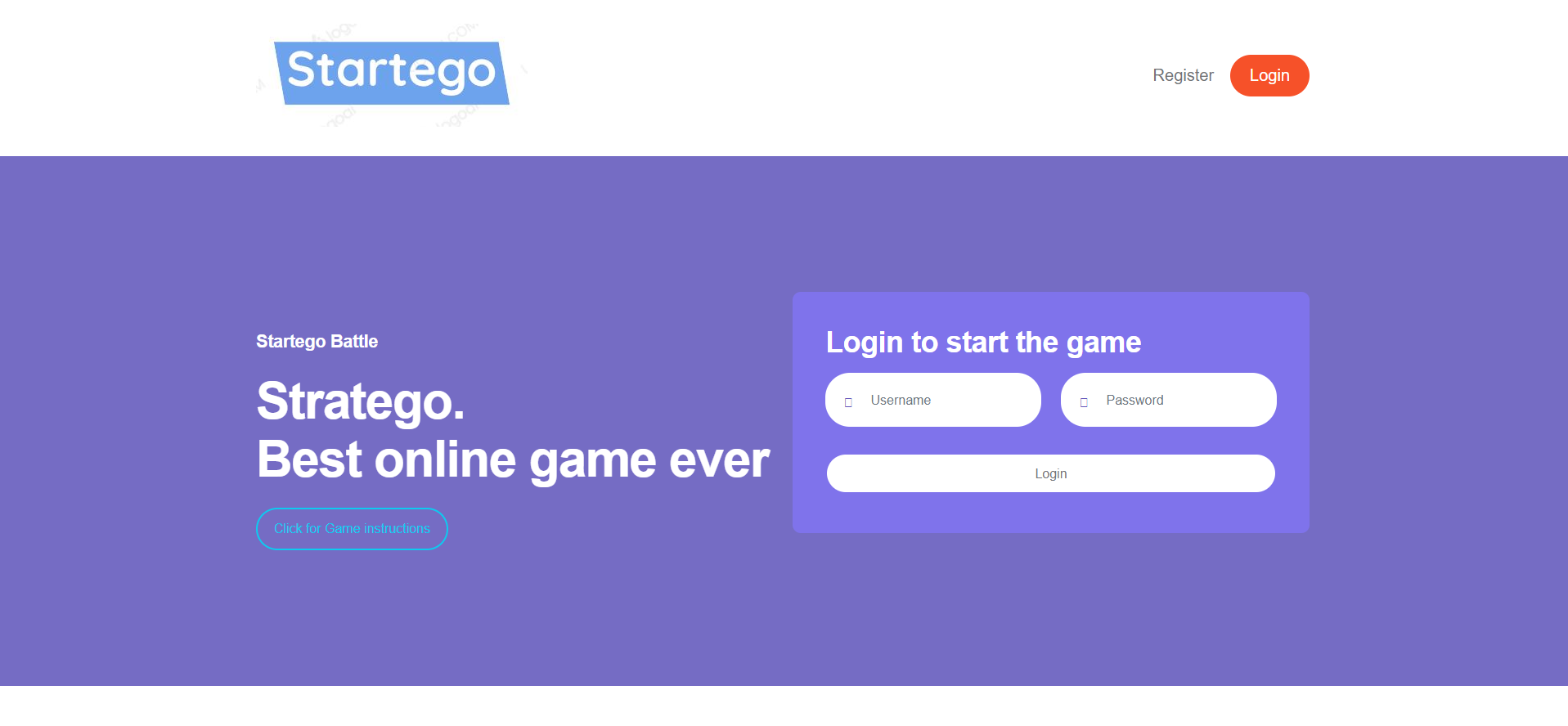
playerId- שם המשתמש של השחקן. כך נדע איזה שחקן התחבר ונדע להבדיל בין שני השחקנים של המשחק.

Response- לאחר עיבוד הבקשה השרת יחזיר תשובה בהתבסס על הפעולה ועל הפעולות שנעשו. בחלק מהמקרים response יכיל נכון או לא נכון, כמו בבקשת רישום המשתמש. הוא יכיל את מזהה המשחק היחודי במקרה שהתחיל משחק. או שבחלק מהמקרים הוא לא יכיל כלום כי אין מה להחזיר לשרת כמו בעשיית הצעד שאין מה להחזיר.

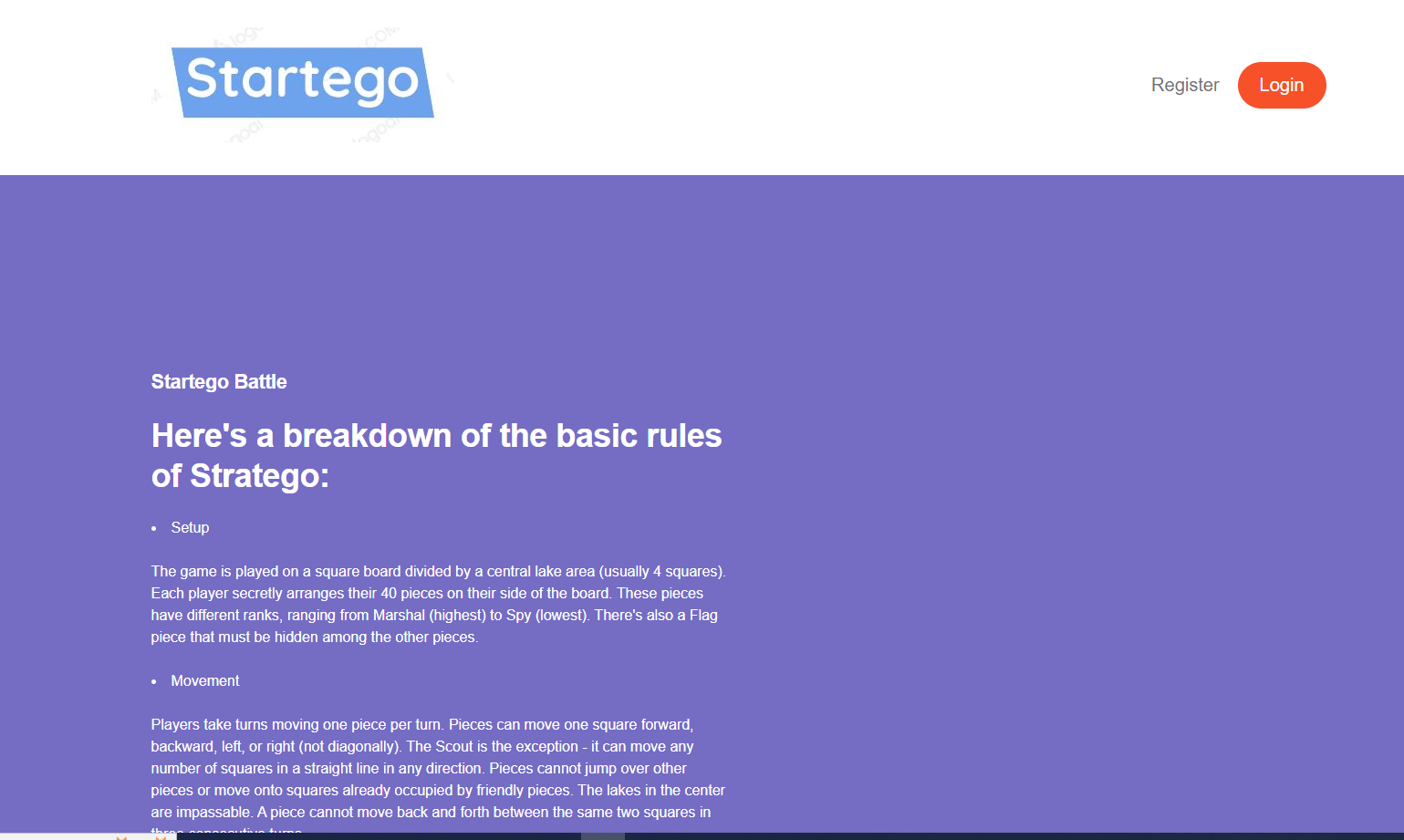
תיאור מסכי המערכת:

המסך הזה הוא מסך הכנסת המשתמש השחקן מכניס שם משתמש, סיסמה ואימות סיסמה. במידה וחוזרת הודעה חיובית הוא ממשיך למסך הבא ובמידה וחוזרת הודעה שלילית הוא מציג הודעת שגיאה ונשאר במסך.

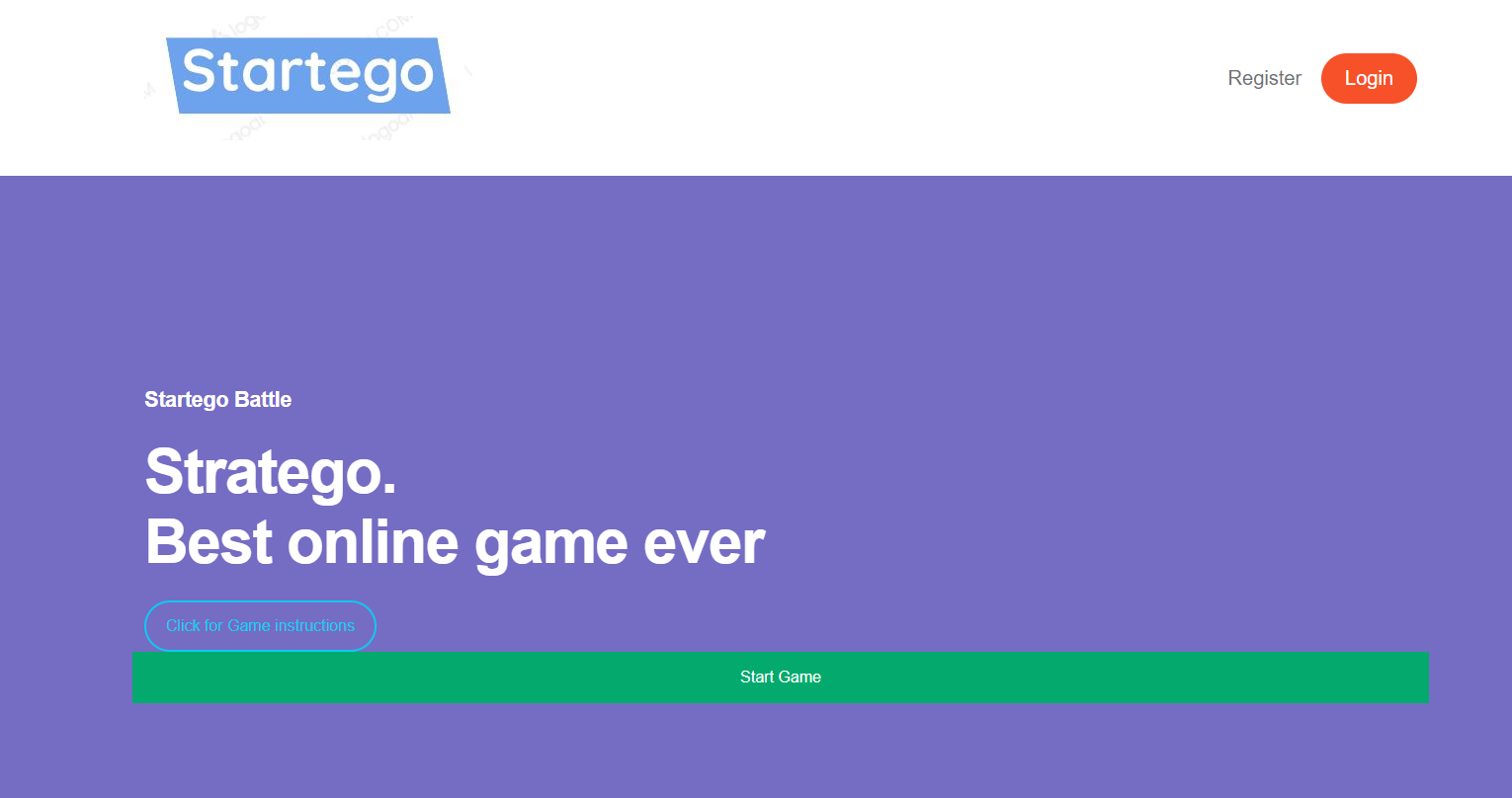
המסך הבא הוא מסך הוא מסך ההתחברות. השחקן מכניס שם משתמש וסיסמה שולח אותם לשרת. במידה וחוזר תשובה חיובית הוא הוא ממשיך לשחקן הבא ובמידה שחוזר תשובה שלילית הוא מציג הודעת שגיאה ונשאר במסך.



הוראות. השחקנים יכולים לעיין בהוראות המשחק.



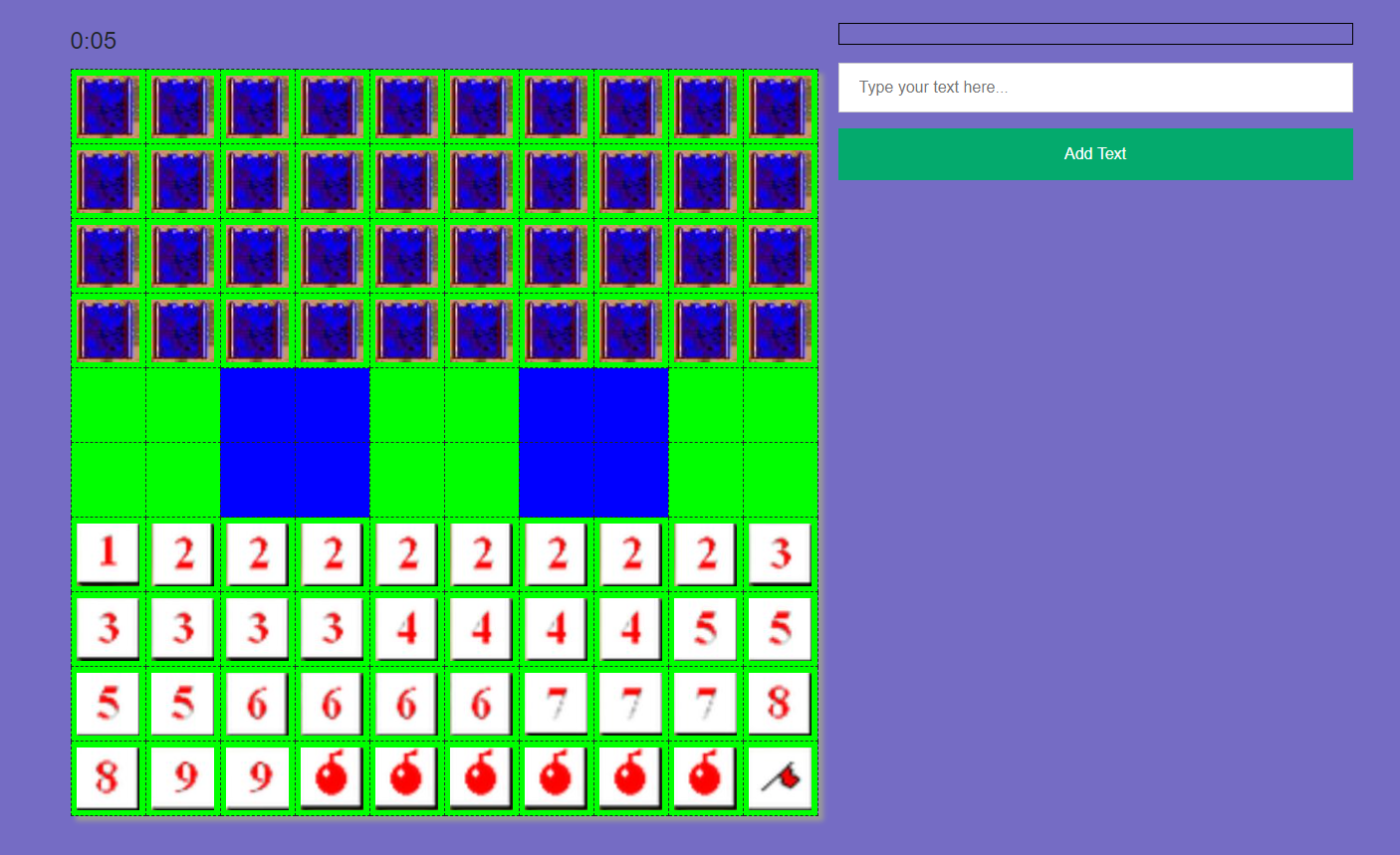
אחרי כניסת המשתמש השחקנים יצוץ כפתור שבו השחקנים יכלו ללחוץ עליו על מנת להתחיל משחק.



\* העיצוב נראה כך נכון להגשת תיק הפרויקט, עד הגשת המשחק הסופי אשפר את הנראות.

לאחר לחיצת הכפתור ועד מציאת יריב יופיע מסך טעינה.

המשחק נראה כך:



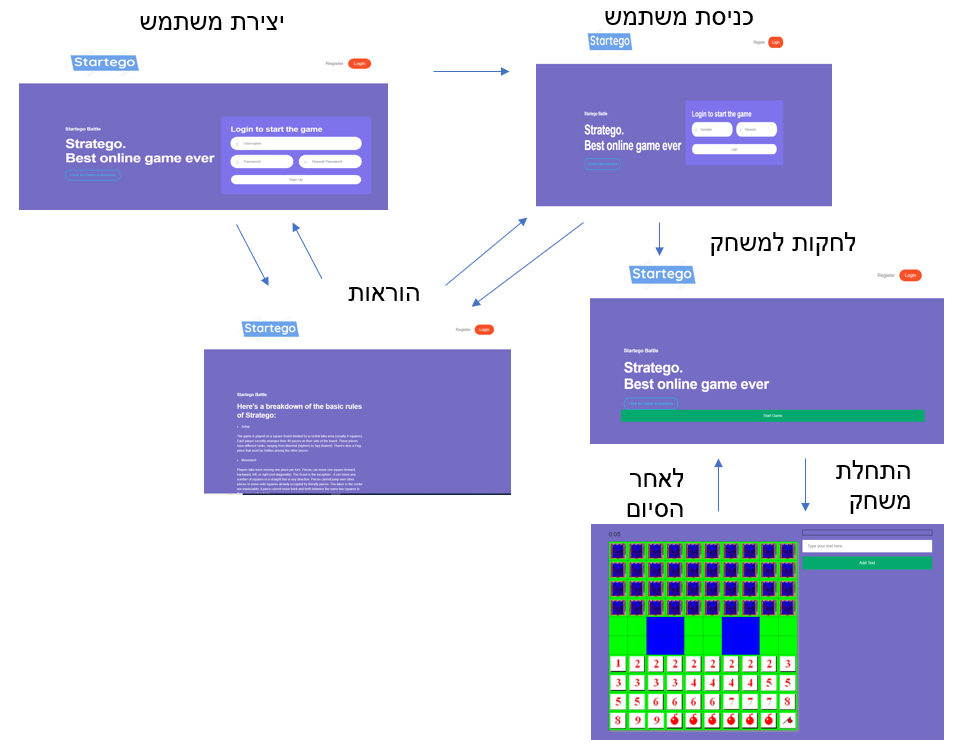
לשני היריבים יהיה חמש דקות התחלתיות לסדר את הכלים שלהם.

לאחר חמש הדקות כל אחד בתורו ישחק תור. כאשר כלי אחד תוקף כלי שני. הכלי בעל המספר הגדול יותר נשאר במשחק. הכלי שנשאר יהיה נראה לשני השחקנים למשך שלוש שניות.

לשני השחקנים צאט שבו הם יכולים לדבר אחד עם השני. הצאט נראה כך:

שם שחקן: הודעה.

\* עד הבגרות אני הוסיף את הניקוד שיראה לשני השחקנים. וזמן שונה לכל אחד מהשחקנים, כך שלאחר הסידור לשני השחקנים יהיה כמות זמן שבו הם יכולים לבצע את המהלכים שלהם.

תרשים מסכים לתיאור ההיררכיה:

מבנה הנתונים:

מסד הנתונים שלי הוא טבלה שבה שלושה עמודות.

שם משתמש- שם המשתמש של כל שחקן. השם משתמש הוא יחודי וללא כפילויות.

הסיסמה- הסיסמה של כל שחקן. יכולות להיות סיסמאות זהות בין שחקנים שונים.

על הסיסמה לפני השמירה במסד הנתונים נעשה הצפנת האש (hash) כך שאם יפרצו אל מסד הנתונים הפורץ לא ידע את הסיסמאות.

דירוג- דירוג אישי של כל שחקן. הדירוג ההתחלתי של כל שחקן הוא 100. על כל משחק השחקנים זוכים בנקודות לדירוג שלהם ובכל הפסד הם מאבדים נקודות.

הדירוג המינימלי של כל שחקן הוא 0. כלומר ברגע ששחקנים מגיעים ל0 נקודות הם לא מפסידים יותר.

סקירת החולשות והאיומים:

1. הזרקת SQL (באנגלית: SQL Injection) היא שיטה לניצול פרצת אבטחה בתוכנית מחשב בעזרת פניה אל מסד הנתונים. על ידי הזרקת sql המשתמשים יכולים לקבל מידע שמור שאין ברצונו של המתכנת להשיג אותו. כך, למשל על ידי פנייה מסוימת הפורץ יוכל לנסות להשיג סיסמאות, מספרים של כרטיסי אשראי ועוד.

הקוד שלי לביצוע השאילתה הוא:

לכניסת המשתמש:

conn = sqlite3.connect("database")  
curr = conn.cursor()  
  
# create and use the query to check if user is in the database  
check\_query = '''SELECT COUNT(\*) FROM usersInfo WHERE username = ? AND password = ?;'''  
curr.execute(check\_query, (username, password))  
  
# Fetch the result of the query  
result = curr.fetchone()[0]  
  
# Close the connection  
curr.close()  
conn.close()

ליצירת משתמש:

# Connect to the database  
conn = sqlite3.connect("database")  
curr = conn.cursor()  
  
# check if username already exists  
check\_query = '''SELECT COUNT(\*) FROM usersInfo WHERE username = ?;'''  
curr.execute(check\_query, (username,))  
  
# Fetch the result of the query  
result = curr.fetchone()[0]  
if result == 1:  
 return "False"  
  
# create and use the query to add the user to the database  
insert\_query = '''INSERT INTO usersInfo (username, password) VALUES (?, ?);'''  
curr.execute(insert\_query, (username, password))  
  
# Commit the changes  
conn.commit()  
  
# Close the connection  
curr.close()  
conn.close()

אני קודם בודק ששם המשתמש איננו תפוס ואז מכניס אותו למסד הנתונים במידה ולא.

ההגנה בפני ההתקפה היא שאת הפרמטרים אנו מסמנים בסימן שאלה (?) ולא מעבירים את הפרמטרים ישר.

על ידי שליחת הפרמטרים לא בצורה ישירה אלה דרך (?), הספרייה דואגת להכניס את הערכים בצורה בטוחה מבלי לאפשר קוד זדוני להתבצע.

לכן: הכנסת ערכים באופן נפרד: ערכי הפרמטרים (username ו־password) מועברים לפונקציה execute בנפרד מהשאילתה עצמה, כך שהספרייה דואגת לטפל בערכים בצורה שמונעת הזרקת SQL.

2. man in the middle- מתקפת "איש באמצע" (Man in the Middle) היא מתקפת אבטחה בה תוקף יושב בין שני צדדים מתקשרים ומתחזה לכל אחד מהם. כך, התוקף יכול ליירט את התקשורת ביניהם, לגנוב מידע, לשנות אותו, או אפילו למנוע את התקשורת בכלל.

על מנת למנוע מתקפה זו אני מצפין את התעבורה לאורך כל הדרך. כשרק כל לקוח מתחבר הוא שולח בקשה לשרת לבקשת המפתח הציבורי שלו. השרת מחזיר לו אותו. הלקוח מכין מפתח סימטרי מצפין אותו בעזרת המפתח הציבורי ושולח לשרת. השרת מפענח בעזרת המפתח הפרטי ושומר את המפתח הסימטרי.

ההצפנה מונעת מהאיש באמצע לפענח את התעבורה אך הוא עדיין יכול לנתק את התקשורת בין השרת ללקוח או לשבש אותה על ידי שינוי ההודעות בצורה לא הגיונית.