

**תיק פרויקט -** **Ragnarök**

**שם התלמיד:** עודד מאיוסט

**ת.ז:** 213262876

**שם המנחה:** יוסי זהבי

**שם החלופה:** הגנת סייבר

**תאריך הגשה:** 22.11.2021



**תוכן עניינים - תיק פרויקט: Ragnarök**

**עמוד שער** \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 1

**תוכן עניינים - תיק פרויקט: Ragnarök** \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 2

**מבוא - Ragnarök**  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 3

הרקע לפרויקט \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 3

מהו הפרויקט \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 4

חקר שוק \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 5

סקירת ספרות \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 7

אתגרים מרכזיים\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 9

**ארכיטקטורה של הפרויקט** \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 10

יחידות הפרויקט \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 10

זרימת המידע בין היחידות השונות \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 11

הודעות הפרויקט \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 13

הלולאות הראשיות של הפרויקט\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 23

הצפנות וקידודים \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 27

**מדריך למשתמש** \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 28

תרשים מסכי מחשב \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 29

תרשים מסכי טלפון\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 30

**בסיס הנתונים** \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 32

**עיצוב תוכנה** \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 33

**רפלקציה** \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 49

**ביבליוגרפיה** \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 50

**מבוא** **-** **Ragnarök**

**הרקע לפרויקט**

תמונה שמכילה אלקטרוניקה

התיאור נוצר באופן אוטומטיבסוף שנות ה-70 בעקבות התפתחות השבב האלקטרוני נכנסו לחיינו קונסולות המשחקים - מחשב ייעודי המשמש לשחק במשחקי וידאו. קונסולות משחקים מורכבות משלט וצג – בדרך כלל הצג הינו מסך הטלוויזיה או מסך המחשב המשמש להקרנת המשחק, אך ישנן גם קונסולות ניידות שיש להן צג משל עצמן, וניתן לקחת אותן ממקום למקום בקלות.

מעבר לכך, בשנים האחרונות החלו להכניס לקונסולות המשחקים את משחקי האונליין, בנוסף למשחקי האופליין.

כיום, עולם קונסולות המשחקים נשלט על ידי מספר יצרניות בולטות – "SONY" עם סדרת "Sony Playstation", "Microsoft" עם סדרת ה-Xbox ו-"Nintendo" עם מגוון רחב של קונסולות ("Nintendo Wii", "Nintendo switch"). אך, לכל החברות הללו מכנה אחד משותף – כולן מלוות במחיר כספי גבוהה אשר לא מאפשר לכל אדם לקחת חלק במשחק בקונסולה.

הדבר הוביל אותי ליצירת הרעיון שלי, ה- Ragnarök: קונסולת משחקים אבסטרקטית אשר תפתח עולם חדש. הקונסולה תכיל במקום שלטים וקונסולה פיזית, אפליקציה לטלפון ותוכנה למחשב – הטלפון עליו האפליקציה ידמה את השלט, והמחשב עליו התוכנה יציג את הפן וויז'ואלי של המשחק (הארנה של המשחקים). כך יפתח ענף חדש בתוך עולם המשחקים:

* בניית שלט מותאם למשחק - עצם העובדה ששלט הקונסולה הינו הטלפון יאפשר ליצרניות המשחקים "לבנות" את השלט בהתאם למשחק, ולנצל יכולות מיוחדות שיש לטלפון: תאוצה של הטלפון, דינמיות של השלט, שימוש בקול...
* התניידות נוחה - לנוכח העובדה ששלט הקונסולה הינו הטלפון (אפלקציה) והקונסולה עצמה הינה אבסטרקטית (תוכנת מחשב), ניתנת האפשרות להתנייד בקלות ממקום למקום עם הקונסולה, ולשחק בה עם המשתמש שלך בכל מקום בעולם בקלות, ללא כל מאמץ פיזי של העברת קונסולה פיזית.
* מחיר עלות שימוש אפסי - השימוש בקונסולה הינו חינמי, דבר המאפשר באופן מלא לכל אדם להשתמש בה, ולהנות משירות המשחקים שהיא מציעה.

מעבר ליתרונות המוצגים לעיל, הרעיון שלי כולל את האפשרות לשחק במשחקים השונים בקונסולה אונליין עם אנשים נוספים המחזיקים גם הם בקונסולת ה- Ragnarök.

**מהו הפרויקט - Ragnarök**

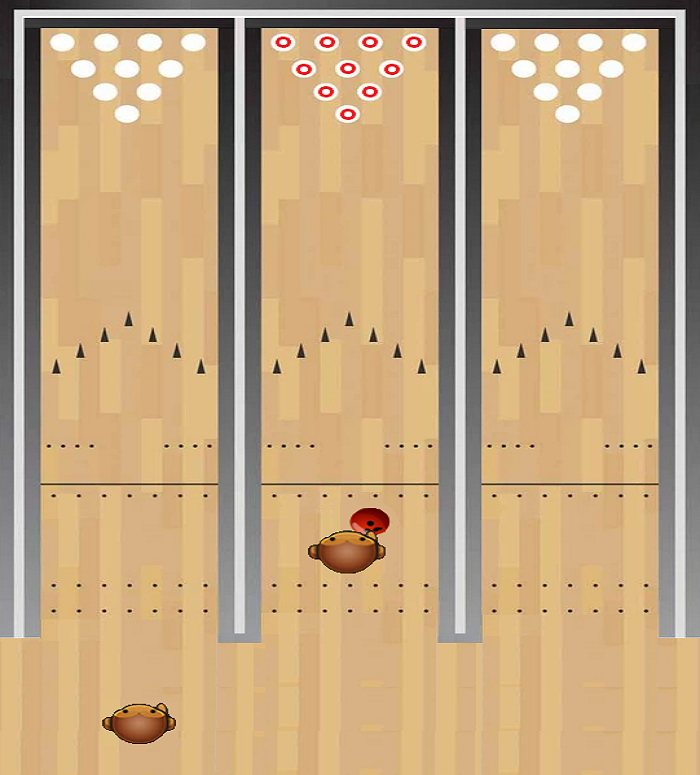
קונסולת משחקים אבסטרקטית אשר מדמה מן מערכת המקבילה ל-"ווי", ובה במקום קונסולה פיזית ושלטים פיזיים, הטלפון ידמה את שלט המשחק ומחשב נוסף יציג את הצד הוויז'ואלי של המשחק.   
את המשחקים השונים במערכת יהיה ניתן לשחק אונליין עם אנשים נוספים.

ממשק המשתמש יהיה מחולק לשני חלקים:

* הטלפון – הטלפון עצמו יכיל אפליקציה אשר תדמה את שלט הקונסולה. בעזרתה יהיה ניתן להתחבר למשתמש קיים או ליצור משתמש חדש, יהיה ניתן לבחור באיזה משחק המשתמש מעוניין לשחק, ולאחר בחירת המשחק היא תשמש כשלט המותאם לכל משחק ומשחק.
* המחשב – בתוך המחשב תהיינה תוכנה אשר תציג את המשחקים המוחלים בקונסולה ובהם ניתן לשחק. מעבר לכך, היא תציג את מהלך המשחק בין המשתמש ליריב.

תחום חקר שלא נלמד בכיתה – אנדרואיד (בניית אפליקציה ושימוש בחיישני הטלפון).

תחומים בהם יעסוק הפרויקט – תקשורת (שרת לקוח), אבטחת מידע (הצפנת מידע והגנות רשת), ניהול מסדי נתונים, ממשקים גרפיים.



**חקר שוק**

כיום הקונסולה אשר הכי משקפת את הרעיון שלי ומקבילה ל-Ragnarök, היא קונסולת Wii של חברת נינטנדו. על כן, החלטתי לבצע סקר שוק והשוואה עימה.

Wii (מבוטאת כמו המילה "We") היא קונסולת המשחקים החמישית של חברת נינטנדו, אשר יצאה לשוק בתאריך ה-19 בנובמבר 2006. קונסולת המשחקים כללה את משחק הוידאו Wii Sports, אשר כלל אוסף של חמש סימולציות ספורט, שנועדו להמחיש את יכולות חישת התנועה של השלט של Wii. חמשת ענפי הספורט שכולל המשחק הם טניס, בייסבול, באולינג, גולף ואגרוף. השחקנים משתמשים בשלט כדי לשחק ובכך מדמים את התנועות במציאות, לדוגמה סיבוב מחבט טניס, זריקת כדור באולינג, חבטה עם מחבט בייסבול... המשחק כולל גם מצבי אימון וכושר המפקחים על התקדמות השחקנים. וזהו המשחק הנמכר ביותר בפלטפורמה אחת בכל הזמנים, והרביעי בסך הכל.

הקונסולה כוללת – קונסולה פיזית של Wii, שלטים ומשחק Wii Sports.  
לעומת זאת הקונסולה שלי כוללת – קונסולת Ragnarök אבסטרקטית, שלט אבסטרקטי של הקונסולה הנמצא על הטלפון, ומכיל 3 משחקים – טניס, באולינג ואגרוף.

|  |  |  |
| --- | --- | --- |
|  | Nintendo Wii  תמונה שמכילה טקסט, משחק, בקר, שלט  התיאור נוצר באופן אוטומטי | Ragnarök |
| שלט | שלט פיזי:  ג&#39;ויסטיקים ואביזרי משחק ‏Wii - זאפ | שלט אבסטרקטי – אפליקציה לסמרטפון (מותאמת לאנדואיד):  תמונה שמכילה טקסט, אלקטרוניקה, iPod  התיאור נוצר באופן אוטומטי |
| מסך | מסך טלוויזיה:  מסך לד 40&#39;&#39; LED HDR TV F402 פירלס | מסכים | מסכי טלוויזיה | מסך מחשב:  מסך מחשב &#39;&#39;23.8 Mag LED S24HD | ציוד הקפי- מחשבים | מסכי מחשב |
| קונסולה | קונסולה פיזית – Wii:  תמונה שמכילה טקסט, משחק, בקר, שלט  התיאור נוצר באופן אוטומטי | קונסולה אבסטרקטית – תוכנת מחשב:  התקנת תוכנה במחשב נייד או נייח מור לוי |
| משחקים | * טניס * בייסבול * באולינג * גולף * אגרוף | * טניס * באולינג * אגרוף |
| מחיר | מחיר הקונסולה המלאה – 325$ | המוצר ללא עלות |

**סקירת ספרות**

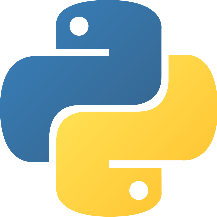
רשימת השפות אשר בשימוש בפרויקט:

* **Java –** שימוש בשפה לשם כתיבת אפליקציית Android
* **Python** **–** שימוש בשפה לשם כתיבת קוד, ושרת ↔ לקוח
* **SQLITE–** תוכנה המאפשרת עבודה עם בסיסי נתונים של SQL

רשימת המודולים שהפרויקט ישתמש בהם:

* **PyGame** – לשם שימוש בגרפיקה למשחקים ולממשק המשתמש
* **Moviepy** – לשם הפעלת סרטון המסייע לבניית המשחקים
* **Socket** – לשם התקשורת בין השרת והלקוח
* **Threading** – לשם הטיפול בלקוחות מרובים
* **OS** – מאפשרת עבודה עם פעולות מערכת הפעלה
* **Sensor (Accelerometer)** – לשם מדידת תאוצת הטלפון וכיוונה

**Android (Java)** **–** אפליקציות אנדרואיד מפותחות בדרך כלל בשימוש בשפת תכנות Java. שפת Java, שפותחה לאחר השפות C ו- C++, אימצה תכונות עוצמתיות רבות מהשפות הללו תוך שהיא משפרת כמה מחסרונותיהן. ובכל זאת, שפות תכנות הן חזקות רק בזכות הספריות שלהן. הספריות הללו קיימות כדי לסייע למפתחים לבנות אפליקציות.

**Python –** שפת Python הינה שפת תכנות אשר שמה דגש על קריאוּת הקוד, וכוללת מבנים המיועדים לאפשר ביטוי של תוכניות מורכבות בדרך קצרה וברורה. לשפה שתי גרסאות ראשיות, Python 2 ו- Python 3. אך, Python 2 פחות נפוצה היום בשימוש שלה, ולרוב משתמשים כיום ב-Python 3.  
הפרויקט שלי מורכב משלוש יחידות – שרת מרכזי, מחשב וטלפון. בפרויקט שלי אשתמש בשפתPython לבניית השרת ותוכנת המחשב.

**SQLITE**– SQL הינה שפה המאפשרת שמירה, קריאה ושינוי מידע במסדי נתונים. שפה זו היא שפה טקסטואלית, כלומר ניתן להשתמש בה בתוכניות רבות, גם אם הן כתובות בשפות תכנות שונות.   
בפרויקט אני אשתמש במודול sqlite3 של python שהוא מימוש של שפת python בשפת SQL לבסיסי נתונים מסוג SQLite. בסיסי הנתונים הללו דומים מאוד לבסיסי הנתונים שלשמם נועדה שפת SQL. ההבדל העיקרי בין בסיסי הנתונים הוא ש-SQLite אינו מופעל מתוך תהליך עצמאי, אלא מתוך תהליכים אחרים, כלומר לפרויקט זה הבדל זה נותן את היתרון של הרצת פקודות SQL מתוך קובץ python.



**PyGame** – PyGame הינו מודול ב- Pythonהמאפשר עבודה עם גרפיקה.  
הפרויקט כולל בתוכו כתיבת ממשק משתמש ומשחקים בשפת Python, דבר הדורש שימוש בגרפיקה. על כן, ישנו שימוש במודול זה בפרויקט.

**Moviepy** – Moviepy הינו מודול של Python המאפשר הפעלה של סרטוני וידאו.  
הפרויקט כולל בתוכו כתיבת ממשק משתמש ומשחקים בשפת Python, וברצוני ליצור ממשק משתמש כמה שיותר איכותי, ברור ומרשים, בין היתר על ידי שימוש בסרטונים. על כן, ישנו שימוש במודול זה בפרויקט.

**Socket** – Socket הוא נקודת הקצה של חיבור תקשורתי דו-כיווני בין שני רכיבי רשת – כתובת ה- Socket היא שילוב כתובת ה-IP ומספר פורט. בעזרת ה-Socket ניתן למצוא את רכיב התקשורת איתו אנו רוצים לקיים את התקשורת, ולגשת כך לתוכנה עצמה במחשב.   
בפרויקט שלי אני אשתמש במודול של sockets על מנת להעביר מידע בין היחידות השונות של הפרויקט. השימוש בsockets יהיה בעזרת שימוש בפרוטוקול TCP.

**tcp-by-size** – tcp-by-size הוא מודול העוזר לתקשורת בפרוטוקול TCP. מודול זה מוסיף לכל הודעה בשליחה את גודלה, בכדי שבקליטת ההודעה הוא יוכל לדעת בדיוק את כמות הבתים שצריך לקלוט. משתמשים במודול זה כאשר רוצים לשלוח הודעות שונות עם גדלים שאינם קבועים.  
בפרויקט שלי אני אשתמש במודול זה בכדי להעביר מידע בין היחידות השונות של הפרויקט.

**Threading** – threads מאפשר ל-process לבצע מספר משימות באופן "מקבילי". למעבד יחיד אין את היכולת להריץ יותר מקטע קוד אחד במקביל, כלומר אין אפשרות להריץ בצורה מקבילית מלאה threads, על כן מתבצעת החלפה בין התהליכים השונים אותם הוא מבצע, ובכך תכנית לא צריכה לסיים משימה אחת לפני שהיא תתחיל עוד אחת.  
בפרויקט שלי אני אשתמש במודול Threading של Python הנותן את האפשרות להפעיל פעולה כ- thread ובכך לבצע מספר משימות ביחד. המשימות אותן אבצע בו זמנית הן ניהול התקשורת בין היחידות השונות.

**Sensor (Accelerometer)** – מחלקה המאפשרת גישה לחיישנים הקיימים בסמרטפון, ובדיקה של הנתונים הנקלטים על ידם. שימוש במחלקה זאת מאפשרת לקבל את תאוצת הטלפון, בצירים השונים – X, Y, Z. בפרויקט שלי השימוש במחלקה זאת הינו לשם קלט התאוצה והכיוון ביחידת הטלפון.

**Hashlib** – hashlib הוא מודול של python המממש ממשק לאלגוריתמים שונים של hash מאובטח.   
אשתמש במודול זה בכדי לעשות Hash (בעזרת אלגוריתם sha256) לסיסמאות הנשמרות ב-database בשרת המרכזי.

**Crypto** – ספריית python המכילה סוגים שונים של הצפנות.   
אני אשתמש בספרייה זו בשביל להצפין מידע רגיש (סיסמאות) בעזרת ההצפנה הסימטרית AES.   
אני אשתמש בספרייה זו גם בשל פעולת unpad הנותנת לי אפשרות להוציא את ריפוד האפסים המוסף למידע לפני הצפנתו.

**base64** - base64 הוא קידוד המשמש להמרת בתים שיש להם נתונים בינאריים או טקסטואליים לתווי ASCII. קידוד זה מונע מהנתונים להיפגם כאשר הם מועברים או מעובדים באמצעות מערכות טקסטואליות בלבד. אני אשתמש במודול base64 של python ושל java בזמן ההצפנה של AES.

**OS** – מודול OS בשפת Python מאפשר גישה לפעולות שונות של מערכת ההפעלה.

**אתגרים מרכזיים**

האתגר המרכזי שליווה אותי בפרויקט הינו בחירתו והגדרתו – היה לי חשוב למצוא פרויקט שיהיה מיוחד ברעיון שלו, שאני אתחבר אליו באופן רגשי, שהעשייה שלו תסקרן ותמשוך אותי, ובסופו של דבר יספק אותי. למצוא רעיון כזה זוהי משימה לא פשוטה כלל, אנשים חושבים שנים בניסיון למצוא רעיון שיענה על כל הדרישה האישית שלהם. למצוא רעיון כזה בטווח זמנים קצר כל כך זוהי משימה כמעט בלתי אפשרית, אבל לאחר מחשבה עמוקה, התייעצות עם אנשים רבים ושילוב רעיונות הגעתי ל-Ragnarök, הפרויקט שלי.

אתגר נוסף שליווה אותי בפרויקט היה למידה של חומר חדש – אחת המטרות המרכזיות של הפרויקט היא לחקור על מידע חדש, ללמוד אותו, ולשלב אותו בפרויקט שלך. אני הייתי צריך ללמוד Android (Java) במחיוחד לביצוע הפרויקט. ומעבר ללימוד Android הייתי צריך ללמוד על ספריות מיוחדות בה אשר מתעסקות בתאוצת הטלפון. אך, למרות הקושי שבלמידת נושא חדש נהניתי מאוד מהחקר ומיישומו.

מעבר לכך, פיתוח משחקים הינו אחד מהתחומים שהכי מעניינים אותי בעולם התכנות – העבודה, הרעיון והמחשבה מאחורי משחקי המחשב השונים מסקרנים ומרתקים אותי. לכן, שנה אחר שנה בחרתי בבניית משחקים בתור הפרויקט שלי – כל פרויקט נגע בתחום אחר בעולם המשחקים, וגם השנה אני רוצה להיחשף לעוד חלק מהעולם הזה שמסקרן אותי כל כך.

לנוכח העובדה שמצאתי פרויקט שעונה על כל דרישותיי, פרויקט שמספק אותי ומגרה את המחשבה שלי אני רווי במוטיבציה ורצון ליישם ולממש אותו. זה לא מובן מאליו מבחינתי למצוא רעיון שאני מאמין בו ומתחבר אליו, והדבר מעניק לי עוד יותר מוטיבציה לעבודה ולעשייה. מעבר לכך, החיבור לנושא פיתוח משחקי המחשב מוסיף גם הוא למוטיבציה לעשיית הפרויקט.



**ארכיטקטורה של הפרויקט**

**מטרת השירות –** קונסולת משחקים אבסטרקטית המאפשרת חוויה מהנה בין שחקנים/משתמשים שונים.

**צורת המענה וייצוג השדות של הפרוטוקול –** צורת המענה של הפרוטוקול הינה פרוטוקול אסינכרוני, וייצוג השדות הינו ייצוג מחרוזתי.

**פרוטוקול תקשורת: TCP –** פרוטוקול TCP הוא פרוטוקול המבטיח אמינות בהעברת המידע. הפרוטוקול בודק שהמידע שהתקבל נכון, מהימן ומלא. במידה והמידע שהועבר לוקה בחסר, המידע שנשלח מבוקש מחדש, ובכך הפרוטוקול מבטיח גם את שלמות המידע. תחילת התקשורת בפרוטוקול TCP מתבססת על three-way handshake – מקים התקשורת שולח SYN, היעד שולח חזרה למקים התקשורת SYN+ACK, ולבסוף מקים התקשורת משיב לו ACK.

**יחידות הפרויקט**

**Super server:** תפקיד השרת המרכזי הוא לנהל את התקשורת בין המשתמשים אשר קשורה למשחקי הקונסולה. השרת מקבל קלטים מהלקוחות וממסד הנתונים, ואת הפלט של פעולותיו שולח בחזרה אליהם.

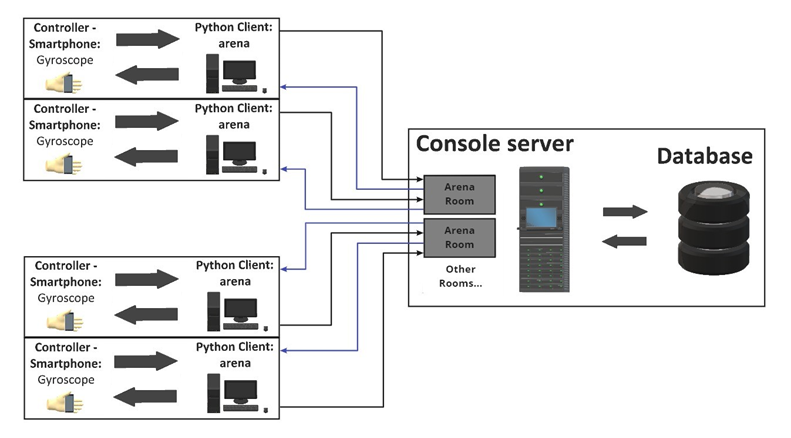
**מחשב:** למחשב ישנם מספר שימושים בפרויקט – הוא משמש כממשק למשתמש, וכחיבור בין הטלפון למחשב ולשרת. רכיב התקשורת אשר המחשב מהווה בתקשורת בינו לבין הטלפון הוא השרת. ורכיב התקשורת אשר המחשב מהווה בתקשורת בינו לבין ה- Super serverהוא לקוח.

* לקוח – מחשב: הקלט והפלט של הלקוח הוא מה-Super server.
* שרת – מחשב: הקלט והפלט של השרת הוא מהטלפון.

**טלפון – הלקוח:** תפקיד הטלפון הוא להוות את השלט של המשחק – ובכך מאפשר שליטה דרך הטלפון. המחשב מהווה את הפלט הוויזואלי אותו רואה המשתמש, והטלפון מהווה את הקלט מהמשתמש – כך שדרכו הוא שולט במשחק.  
הקלט של הטלפון הוא מהמשתמש ומהמחשב (שרת המחשב), והפלט שלו מועבר למחשב (שרת המחשב).

**מסד נתונים:** תפקידו של מסד הנתונים הוא לשמור את כל המידע של הפרויקט ולספק אותו לשרת. מסד הנתונים שומר מידע על המשתמשים, ועל היסטוריית השימוש שלהם בקונסולה.

**זרימת המידע בין היחידות השונות**

****

**הודעות הפרויקט**

תמונה שמכילה טקסט, שעון

התיאור נוצר באופן אוטומטיהודעות – טלפון (לקוח) 🡨 מחשב (שרת):

**SGNUP =** רשום אותי כמשתמש חדש.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** שם המשתמש של השחקן – אורך השדה בגודל משתנה ומוגבל עד 30 תווים
* **שדה רביעי:** תו מפריד " ~ " – באורך תו אחד
* **שדה חמישי:** הסיסמה של השחקן – אורך השדה בגודל משתנה ומוגבל עד 30 תווים
* **שדה שישי:** תו מפריד " ~ " – באורך תו אחד
* **שדה שביעי:** השם המלא של השחקן (שם פרטי ושם משפחה) – אורך השדה בגודל משתנה ומוגבל עד 30 תווים
* **שדה שמיני:** תו מפריד " ~ " – באורך תו אחד
* **שדה תשיעי:** הטלפון של השחקן – אורך השדה בגודל משתנה ומוגבל עד 10 תווים

**SGNIN =** חבר אותי לקונסולה עם משתמש קיים.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** שם המשתמש של השחקן – אורך השדה בגודל משתנה ומוגבל עד 30 תווים
* **שדה רביעי:** תו מפריד " ~ " – באורך תו אחד
* **שדה חמישי:** הסיסמה של השחקן – אורך השדה בגודל משתנה ומוגבל עד 30 תווים

**MOVES =** הודעה המייצגת תזוזה על מסך הבית במחשב.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** סוג התזוזה – למעלה או למטה: 1 = למעלה, 2 = למטה – באורך תו אחד

**CHOSE =** הודעה המייצגת בחירה של מקש במחשב השרת המייצג את סוג המשחק הרצוי.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים

|  |  |
| --- | --- |
| הודעות – "Bowling Game":  **PMOVE =** השחקן ביצע תזוזה – ימינה או שמאלה.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים * **שדה שני:** תו מפריד "~" – באורך תו אחד * **שדה שלישי:** כיוון התנועה של השחקן – 1 = ימינה, 2 = שמאלה – באורך תו אחד   **ACLRT =** הגודל והכיוון של תאוצת זריקת כדור הבאולינג של השחקן.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים * **שדה שני:** תו מפריד "~" – באורך תו אחד * **שדה שלישי:** גודל התאוצה – אורך השדה בגודל משתנה * **שדה רביעי:** תו מפריד "~" – באורך תו אחד * **שדה חמישי:** כיוון התאוצה – אורך השדה בגודל משתנה | הודעות – "Rock Hero":  **PRESS =** עדכון – הלקוח לחץ על מקש. כיוון ההודעה הוא מהלקוח לשרת.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים * **שדה שני:** תו מפריד " ~ " – באורך תו אחד * **שדה שלישי:** משתנה המייצג מהי סוג הלחיצה – 1 = ימינה, 2 = שמאלה,  3 = למעלה, 4 = למטה |

תמונה שמכילה טקסט, שעון

התיאור נוצר באופן אוטומטיהודעות – מחשב (שרת) 🡨 טלפון (לקוח):

**REGIS =** האם רישום המשתמש התבצע בהצלחה.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** משתנה המייצג האם ההרשמה הצליחה – 1 = ההרשמה הצליחה,   
  2 = ההרשמה נכשלה

**LOGIN =** התחבר שחקן למשתמש קיים.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** משתנה המייצג האם ההתחברות הצליחה – 1 = ההתחברות הצליחה,   
  2 = ההתחברות נכשלה

**START =** המשחק מתחיל.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** סוג המשחק – 1 = "Bowling Game", 2 = "Rock Hero"

**ENDED =** המשחק נגמר.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים

**דוגמה לביצוע הפרוטוקול – טלפון (לקוח) →← שרת:**

**לקוח 🡨 SGNIN**~oded~mayost

הלקוח ניסה להתחבר למשתמש קיים.

**שרת 🡨** 1~ **LOGIN**

המשתמש התחבר למשתמש קיים.

**לקוח 🡨 CHOSE**

הלקוח בחר את המשחק בו הוא מעוניין לשחק.

**שרת 🡨 START**~1

השרת הודיע ללקוח שהמשחק מתחיל, וסוגו הינו באולינג.

**לקוח 🡨 PMOVE**~1

הלקוח ביצעה תזוזה ימינה.

**לקוח 🡨 PMOVE**~1

הלקוח ביצעה תזוזה ימינה.

**לקוח 🡨 ACLRT**~18.3~65

הלקוח זרק את הכדור.

**שרת 🡨 ENDED**

השרת הודיע ללקוח שהמשחק נגמר.

(התקשורת בין השרת המרכזי ללקוח הסתיימה – ככל הנראה התקשורת נותקה עם המשתמש השני)

הודעות – מחשב (לקוח) 🡨 שרת:

**HELLO =** חיבור המחשב לשרת המרכזי.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים

**SGNUP =** רשום אותי כמשתמש חדש.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** שם המשתמש של השחקן – אורך השדה בגודל משתנה ומוגבל עד 30 תווים
* **שדה רביעי:** תו מפריד " ~ " – באורך תו אחד
* **שדה חמישי:** הסיסמה של השחקן – אורך השדה בגודל משתנה ומוגבל עד 30 תווים
* **שדה שישי:** תו מפריד " ~ " – באורך תו אחד
* **שדה שביעי:** השם המלא של השחקן (שם פרטי ושם משפחה) – אורך השדה בגודל משתנה ומוגבל עד 30 תווים
* **שדה שמיני:** תו מפריד " ~ " – באורך תו אחד
* **שדה תשיעי:** הטלפון של השחקן – אורך השדה בגודל משתנה ומוגבל עד 10 תווים

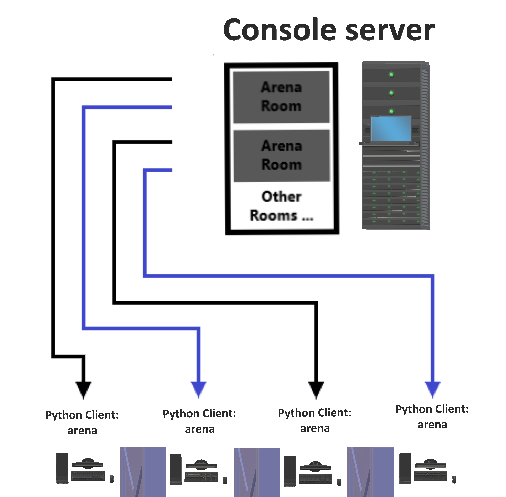
**SGNIN =** חבר אותי לקונסולה עם משתמש קיים.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** שם המשתמש של השחקן – אורך השדה בגודל משתנה ומוגבל עד 30 תווים
* **שדה רביעי:** תו מפריד " ~ " – באורך תו אחד
* **שדה חמישי:** הסיסמה של השחקן – אורך השדה בגודל משתנה ומוגבל עד 30 תווים

**TYGAM =** סוג המשחק אותו אני רוצה לשחק – טניס, באולינג, אגרוף.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** סוג המשחק – 1 = "Bowling Game", 2 = "Rock Hero"

|  |  |
| --- | --- |
| הודעות – משחק באולינג:  **ENDED =** המשחק נגמר – המשתמש הפסיד במשחק.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים   **RNDOV =** המערכה נגמרה.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים   **PINUM =** כמות הפינים שהופלו באותו הסיבוב.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים * **שדה שני:** תו מפריד "~" – באורך תו אחד * **שדה שלישי:** כמות הפינים שהופלו – אורך השדה בגודל משתנה   **PMOVE =** השחקן ביצע תזוזה – ימינה או שמאלה.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים * **שדה שני:** תו מפריד "~" – באורך תו אחד * **שדה שלישי:** כיוון התנועה של השחקן – 1 = ימינה, 2 = שמאלה – באורך תו אחד   **ACLRT =** הגודל והכיוון של תאוצת זריקת כדור הבאולינג של השחקן.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים * **שדה שני:** תו מפריד "~" – באורך תו אחד * **שדה שלישי:** גודל התאוצה – אורך השדה בגודל משתנה * **שדה רביעי:** תו מפריד "~" – באורך תו אחד * **שדה חמישי:** כיוון התאוצה – אורך השדה בגודל משתנה | הודעות – "Rock Hero":  **ENDED =** המשחק נגמר – המשתמש הפסיד במשחק.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים   **PRESS =** עדכון – הלקוח לחץ על מקש. כיוון ההודעה הוא מהלקוח לשרת.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים * **שדה שני:** תו מפריד " ~ " – באורך תו אחד * **שדה שלישי:** משתנה המייצג האם הייתה משמעות ללחיצה על המקש – 1 = True, 2 = False   **MISSA =** עדכון – הלקוח פספס חץ. כיוון ההודעה הוא מהלקוח לשרת.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים * **שדה שני:** תו מפריד " ~ " – באורך תו אחד * **שדה שלישי:** משתנה המייצג האם השחקן בחיים – 1 = השחקן בחיים, 2 = השחקן איננו בחיים   **הודעה אופציונלית (Admin):**  **ARROW =** עדכון השחקנים על יצירת חץ חדש במשחק. כיוון ההודעה הוא מהשרת ללקוח.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים * **שדה שני:** תו מפריד " ~ " – באורך תו אחד * **שדה שלישי:** מספר בין 0 ל-3 המייצג את סוג החץ – באורך תו אחד |

הודעות – שרת 🡨 מחשב (לקוח):

**COMID =** העברת ה-ID של המחשב המוגדר ב-database.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** משתנה המכיל את ה-ID של המחשב המוגדר ב-database

**REGIS =** האם רישום המשתמש התבצע בהצלחה.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** משתנה המייצג האם ההרשמה הצליחה – 1 = ההרשמה הצליחה,   
  2 = ההרשמה נכשלה

**LOGIN =** התחבר שחקן למשתמש קיים.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** משתנה המייצג האם ההתחברות הצליחה – 1 = ההתחברות הצליחה,   
  2 = ההתחברות נכשלה

**START =** התחלת משחק חדש בין שני משתמשים – התחבר משתמש חדש למשחק המבוקש.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** משתנה המייצג את סוג המשחק – 1 = "Bowling Game", 2 = "Rock Hero"   
  (אם True 4 תווים, אם False 5 תווים)
* **שדה רביעי:** תו מפריד " ~ " – באורך תו אחד
* **שדה חמישי:** משתנה המייצג איזה מהמשתמשים מתחיל קודם – 1 = המשתמש מתחיל,   
  2 = המשתמש הנגדי מתחיל

|  |  |
| --- | --- |
| הודעות – משחק באולינג:  **ENDED =** המשחק נגמר – המשתמש ניצח במשחק.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים   **RNDOV =** המערכה נגמרה.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים   **ENPIN =** כמות הפינים שהפיל היריב באותו הסיבוב.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים * **שדה שני:** תו מפריד "~" – באורך תו אחד * **שדה שלישי:** כמות הפינים שהופלו – אורך השדה בגודל משתנה   **ENMOV =** היריב ביצע תזוזה – ימינה או שמאלה.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים * **שדה שני:** תו מפריד "~" – באורך תו אחד * **שדה שלישי:** כיוון התנועה של השחקן – 1 = ימינה, 2 = שמאלה – באורך תו אחד   **ENACL =** הגודל והכיוון של תאוצת זריקת כדור הבאולינג של היריב.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים * **שדה שני:** תו מפריד "~" – באורך תו אחד * **שדה שלישי:** גודל התאוצה – אורך השדה בגודל משתנה * **שדה רביעי:** תו מפריד "~" – באורך תו אחד * **שדה חמישי:** כיוון התאוצה – אורך השדה בגודל משתנה | הודעות – "Rock Hero":  **ENDED =** המשחק נגמר – המשתמש ניצח במשחק.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים   **GSONG =** עדכון השחקנים מהו שיר המשחק. כיוון ההודעה הוא מהשרת ללקוח.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים * **שדה שני:** תו מפריד " ~ " – באורך תו אחד * **שדה שלישי:** מספר בין 0 ל-4 המייצג את שיר המשחק – באורך תו אחד   **EMISS =** עדכון היריב - השחקן השני פספס חץ. כיוון ההודעה הוא מהשרת ללקוח.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים   **ENKEY =** עדכון היריב – השחקן השני לחץ על מקש במקלדת. כיוון ההודעה הוא מהשרת ללקוח.   * **שדה ראשון:** קוד ההודעה – באורך 5 תווים * **שדה שני:** תו מפריד " ~ " – באורך תו אחד * **שדה שלישי:** משתנה המייצג האם הייתה משמעות ללחיצה על המקש – 1 = הייתה משמעות, 2 = לא הייתה משמעות |

**דוגמה לביצוע הפרוטוקול – מחשב (לקוח) →← שרת מרכזי:**

**לקוח 🡨 HELLO**

הלקוח ניסה להתחבר לשרת המרכזי.

**שרת 🡨** 5~ **COMID**

המשתמש התחבר לשרת, והוא בתגובה שלח לו את ה-ID שלו השמור ב-database.

**לקוח 🡨 SGNIN**~oded~mayost

הלקוח ניסה להתחבר למשתמש קיים.

**שרת 🡨** 1~ **LOGIN**

המשתמש התחבר למשתמש קיים.

**לקוח 🡨 TYGAM**~1

הלקוח בחר את המשחק בו הוא מעוניין לשחק.

**שרת 🡨 START**~1

השרת הודיע ללקוח שהמשחק מתחיל, וסוגו הינו באולינג.

**לקוח 🡨 PMOVE**~1

הלקוח ביצעה תזוזה ימינה.

**לקוח 🡨 PMOVE**~1

הלקוח ביצעה תזוזה ימינה.

**לקוח 🡨 ACLRT**~18.3~65

הלקוח זרק את הכדור.

**לקוח 🡨 PINUM**~8

כמות הפינים אשר הלקוח הפיל לאחר זריקת הכדור.

**שרת 🡨 ENDED**

השרת הודיע ללקוח שהמשחק נגמר, הלקוח ניצח במשחק.

(המשחק נגמר – ככל הנראה התקשורת נותקה עם המשתמש השני, ועל כן הודיע השרת למשתמש על סיום המשחק וניצחונו התכני בו)

הודעות – טלפון (לקוח) 🡨 שרת:

**COMID =** העברת ה-ID של המחשב אשר קיבל הלקוח מהמחשב**.**

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** משתנה המכיל את ה-ID של המחשב

**DIFHL =** העברת המפתח הציבורי של אפליקציית הטלפון, על מנת לבצע בשרת המרכזי את האלגוריתם   
של Diffie Hellman.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** משתנה (A) המייצג את ה-public key של אפליקציית הטלפון
* **שדה רביעי:** תו מפריד " ~ " – באורך תו אחד
* **שדה חמישי:** משתנה (p) המייצג primary number
* **שדה שישי:** תו מפריד " ~ " – באורך תו אחד
* **שדה שביעי:** משתנה (g) המייצג primary number המקיים p>g

הודעות – שרת 🡨 טלפון (לקוח):

**COMIP =** העברת ה-IP של המחשב המוגדר ב-database.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** משתנה המכיל את ה-IP של המחשב המוגדר ב-database

**DIFHL =** העברת המפתח הציבורי של השרת המרכזי, על מנת לבצע באפליקציית הטלפון את האלגוריתם   
של Diffie Hellman בכדי לייצר את המפתח המשותף שלהם.

* **שדה ראשון:** קוד ההודעה – באורך 5 תווים
* **שדה שני:** תו מפריד " ~ " – באורך תו אחד
* **שדה שלישי:** משתנה (B) המייצג את ה-public key של השרת המרכזי

**דוגמה לביצוע הפרוטוקול – טלפון (לקוח) →← שרת מרכזי:**

**לקוח 🡨 COMID**~5

הלקוח ביקש מהשרת המרכזי להביא לו את ה-IP של המחשב שה-ID הינו 5.

**שרת 🡨** 192.168.1.153~**COMIP**

השרת החזיר ללקוח את ה-IP המבוקש.

**לקוח 🡨 DIFHL**~"A"~"p"~"g"

הלקוח שלח לשרת המרכזי את המפתח הציבורי שיצר.

**שרת 🡨 DIFHL**~"B"

השרת המרכזי החזיר ללקוח את המפתח הציבורי שלו.

(התקשורת בין השרת המרכזי ללקוח הסתיימה – התקשורת נותקה)

**הלולאות הראשיות של הפרויקט**

**השרת הראשי – Super Server:**

משתנים קבועים:

* port = 55555 – ה-port שדרכו מתחברים.
* ip = "0.0.0.0" – ה-IP אליו מתחברים בכדי להתחבר לשרת המרכזי.
* threads\_mode\_lock = threading.Lock() – משתנה המאפשר נעילת גישה למשתנה הגלובלי, threads\_mode, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת.
* users\_lock = threading.Lock() – משתנה המאפשר נעילת גישה למשתנה הגלובלי, users, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת.
* clients\_lock = threading.Lock() – משתנה המאפשר נעילת גישה למשתנה הגלובלי, clients, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת.
* connect\_bowling\_lock = threading.Lock() - משתנה המאפשר נעילת גישה למשתנה הגלובלי, connect\_bowling, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת.
* connect\_game\_lock = threading.Lock() - משתנה המאפשר נעילת גישה למשתנה הגלובלי, connect\_game, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת.

משתנים גלובליים:

* threads = [] – רשימת ה-Threads הרצים בשרת המרכזי
* threads\_mode = {} – לכל לקוח המתחבר לשרת נפתח thread, והוא יוצר thread עזר. המילון מצביע על מצבו של thread העזר, האם הוא עדיין רץ או שהרצתו הופסקה.  
  המפתח של מילון זה מכיל את ה-client id אליו משוייך ה-thread, וערכו הינו משתנה בוליאני המייצג האם הוא עדיין רץ או שהרצתו הופסקה.
* clients = {} – השרת המרכזי מנהל את התקשורת בין הלקוחות השונים. המילון מכיל רשימה של הודעות המיועדות לשליחה ללקוח.  
  המפתח של מילון זה מכיל את ה-client id אליו משוייך ה-thread, וערכו הינו רשימה של הודעות המיועדות לשליחה ללקוח.
* users = {} - השרת המרכזי אחראי בין היתר על חיבור בין שני משתמשים וניהול התקשורת בניהם, לכדי משחק אחד נגד השני.  
  המפתח של מילון זה מכיל את ה-client id אליו משוייך ה-thread, וערכו הינו ה-id של היריב נגדו הוא משחק.
* connect\_bowling = None – משתנה גלובלי המסייע לחיבור בין שני משתמשים למשחק "באולינג".
* connect\_game = None – משתנה גלובלי המסייע לחיבור בין שני משתמשים למשחק   
  "Rock Hero".

הלולאה הראשית:

{

מחכה להתחברות של לקוח חדש.

פותח thread לניהול התקשורת עם הלקוח – בתוך ה- threadהלקוח פותח משתמש בקונסולה (נרשם) או מתחבר למשתמש קיים, ומנהל את המשחק בינו לבין משתמש אחר, וכל זאת על פי פרוטוקול התקשורת.

}

**תוכנת המחשב:**

משתנים קבועים:

* ip = "0.0.0.0" – ה-IP אליו מתחברים בכדי להתחבר לתוכנת המחשב.
* port\_android\_client = 50000 - ה-port דרכו מתחברת אפליקציית הטלפון לתוכנת המחשב (השרת).
* ip\_super\_server = "192.168.1.153" - ה-IP אליו מתחברת תוכנת המחשב (לקוח) לשרת המרכזי.
* port\_super\_server = 55555 - ה-port דרכו מתחברת תוכנת המחשב (לקוח) לשרת המרכזי.
* android\_message\_lock = threading.Lock() – משתנה המאפשר נעילת גישה למשתנה הגלובלי, android\_message, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת.
* android\_thread\_mode\_lock = threading.Lock() - משתנה המאפשר נעילת גישה למשתנה הגלובלי, android\_thread\_mode, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת.
* super\_server\_message\_lock = threading.Lock() - משתנה המאפשר נעילת גישה למשתנה הגלובלי, super\_server\_message, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת.
* super\_server\_thread\_mode\_lock = threading.Lock() - משתנה המאפשר נעילת גישה למשתנה הגלובלי, super\_server\_thread\_, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת.
* game\_message\_lock = threading.Lock() - משתנה המאפשר נעילת גישה למשתנה הגלובלי, game\_message, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת.

משתנים גלובליים:

* threads = [] - רשימת ה-Threads הרצים בתוכנת המחשב
* threads\_mode = True – בתוכנת המחשב רצים כמה threads במקביל. מטרת המשתנה הינה שברגע סגירת ה-thread המנהל את המשחק, כל שאר ה-threads הרצים גם באותה העת יסגרו גם הם.
* android\_message = [] – תוכנת המחשב (שרת) מקיים תקשורת בינו לבין אפליקציית הטלפון (לקוח). הרשימה מכילה הודעות המיועדות לשליחה ללקוח.
* android\_thread\_mode = True – ברגע הרצת ה-thread המנהל את התקשורת עם אפליקציית הטלפון, מורץ מתוכו thread עזר. מטרת המשתנה הינה שברגע סגירת ה-thread המנהל את התקשורת עם אפליקציית הטלפון, גם thread העזר יסגר באותה העת.
* super\_server\_message = [] - תוכנת המחשב (לקוח) מקיים תקשורת בינו לבין השרת המרכזי. הרשימה מכילה הודעות המיועדות לשליחה לשרת המרכזי.
* super\_server\_thread\_mode = True – ברגע הרצת ה-thread המנהל את התקשורת עם השרת המרכזי, מורץ מתוכו thread עזר. מטרת המשתנה הינה שברגע סגירת ה-thread המנהל את התקשורת עם השרת המרכזי, גם thread העזר יסגר באותה העת.
* game\_message = [] – קיים thread אשר תפקידו לנהל את מהלך המשחק עצמו. על thread זה להיות מעורב וחשוף להודעות העוברות דרך תוכנת המחשב. רשימה זאת מכילה את ההודעות הרלוונטיות אשר התקבלו מאפלקציית הטלפון והשרת המרכזי.
* type\_game = None – משתנה המייצג סוג המשחק הנבחר על ידי המשתמש.
* game\_mode = True – משתנה המייצג האם המשחק מתנהל זה עתה, או שהוא נגמר.

לולאה ראשית:

}

מפעיל פעולה האחראית על מסך הבית – הפעולה מדפיסה את מסך הבית, ואחראית על קביעת סוג המשחק הרצוי.

מפעיל פעולה האחראית על מסך הממתנה – הפעולה ממתינה להתחברות שחקן נוסף למשחק המבוקש, ומחזירה מיהו השחקן בעל התור הראשון (במידה והמשחק הינו באולינג).

בדיקה מהו סוג המשחק:

במידה והמשחק הנבחר הינו באולינג מופעלת פעולה האחראית על ניהול והפעלת משחק הבאולינג.

במידה והמשחק הנבחר הינו "Rock Hero" מופעלת פעולה האחראית על הפעלת המשחק וניהולו.

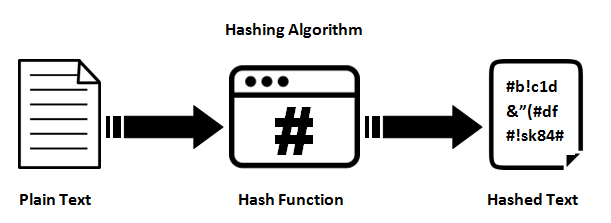
}

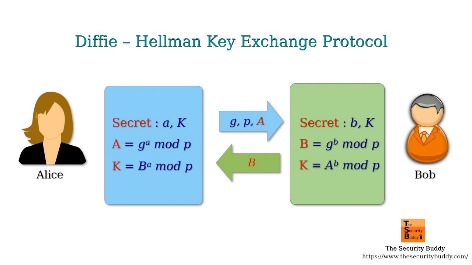
**אפליקציית הטלפון:**

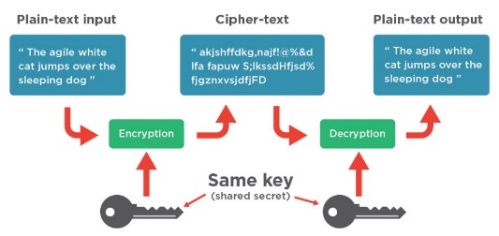
* MainActivity – Activity זה אחראי על הפעלת סרטון הפתיחה של האפליקציה.
* StartupActivity – Activity זה אחראי על הכנסת ה-computer ID המחבר בין האפליקציה לתוכנת המחשב, ומאפשר בחירה של הרשמה למשחק או לחילופין חיבור השחקן למשתמש קיים.
* SignInActivity – Activity זה אחראי על חיבור השחקן למשתמש קיים.
* SignUpActivity – Activity זה אחראי על רישום של משתמש חדש לקונסולה.
* HomepageActivity – Activity זה מדמה שלט אשר בעזרתו ניתן לבחור את המשחק המבוקש על ידי המשתמש.
* BowlingGame – Activity זה מהווה כשלט של משחק הבאולינג.
* RockHero – Activity זה מהווה כשלט של המשחק "Rock Hero".

**הצפנות וקידודים**

בפרויקט אשתמש בהצפנות ופרוטוקולי אבטחה בכדי להגן על המידע של המשתמשים, ובייחוד על הסיסמאות שלהם.

**SHA-256:** SHA256 היא פונקציית גיבוב קריפטוגרפית ממשפחת SHA-2. פונקציית גיבוב קריפטוגרפית היא אלגוריתם הממפה קלט כלשהו (באורך שרירותי) לפלט בעל אורך מסוים (במקרה זה 256 בתים).  
הפונקציה היא חד-כיוונית, כלומר בהינתן הקלט ניתן בקלות להגיע לפלט, אך בהינתן הפלט כמעט בלתי אפשרי להגיע לקלט.   
בנוסף, לכל קלט יש פלט יחיד ובשינויים קטנים של הקלט, הפלט משתנה כמעט לחלוטין.   
על כן, אשתמש בפונקציה זו לשמירה של מעין חתימה של סיסמאות המשתמשים במקום לשמור את הסיסמאות עצמן.

**Diffie Hellman**: פרוטוקול תקשורת המאפשר ביסוס בטוח של מפתח משותף בין שני צדדים בעזרת מפתח ציבורי ומפתח פרטי של כל צד. הפרוטוקול משתמש בחישובים מתמטיים על מנת לקחת את המפתח הציבורי, שני מספרים ידועים (ציבוריים) ומפתח פרטי כדי שלשני הצדדים יהיה מפתח משותף.  
בפרויקט אשתמש בפרוטוקול זה על מנת לבסס מפתח משותף בין לקוח הטלפון לשרת המרכזי כדי שיוכלו להצפין ולהעביר מידע רגיש (סיסמאות) מבלי שגורם שלישי יוכל לגלות את המידע.

**AES:** AES הוא אלגוריתם הצפנה סימטרי, כלומר בהינתן אותו מפתח הצפנה, ניתן להגיע בקלות מטקסט לצופן וגם מהצופן לטקסט.

האלגוריתם משתמש בבלוקים בכדי שיוכל להצפין מידע בגדלים שונים. אני אשתמש באלגוריתם AES בשביל להצפין את העברת הסיסמאות מהלקוח אל השרת המרכזי.

**מדריך למשתמש**

ממשק המשתמש מורכב משני חלקים:

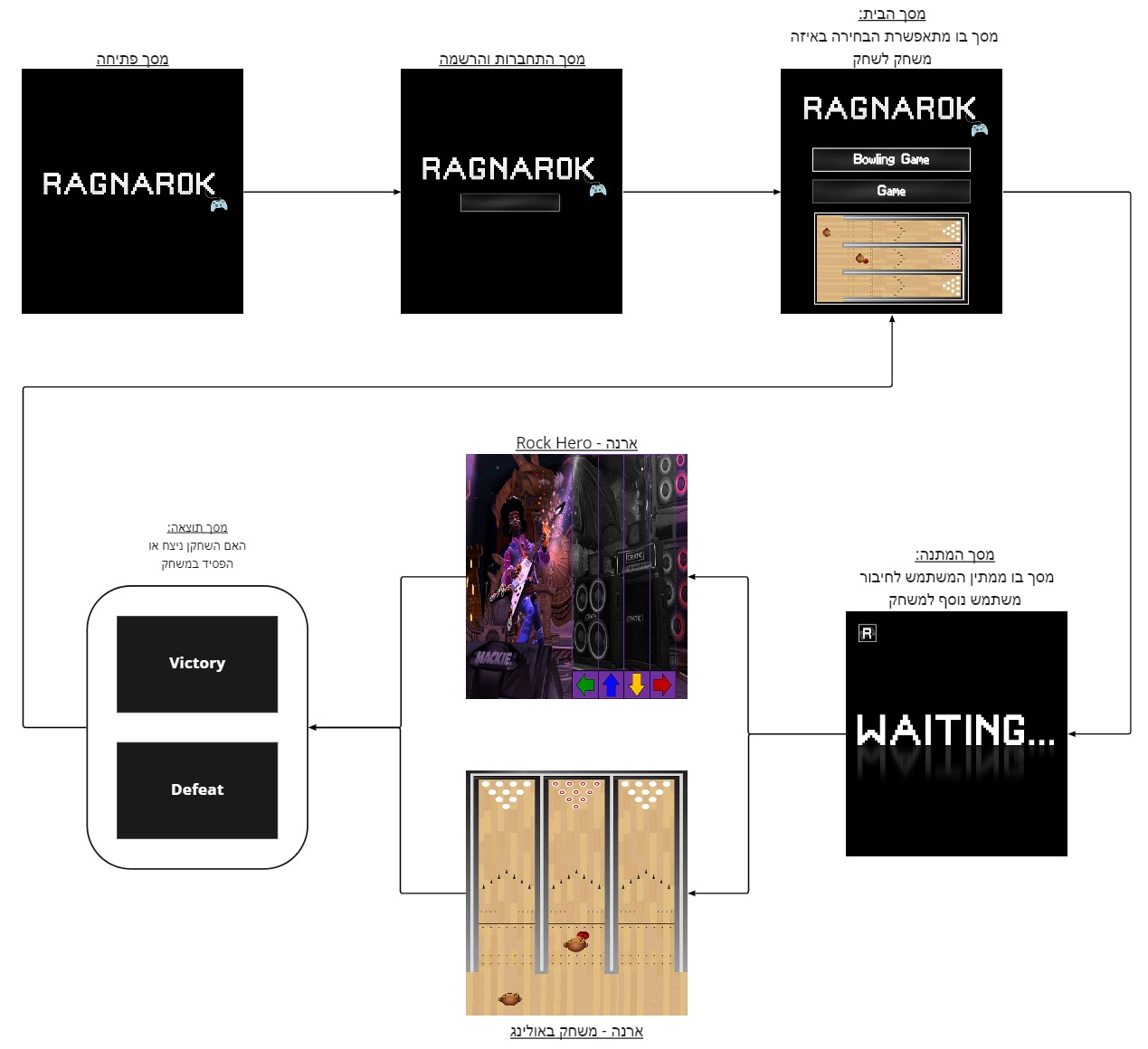
* תוכנת מחשב - תוכנה אשר תהיה מותקנת על המחשב, ותציג את הוויז'ואל של הקונסולה.  
  התוכנה תהיה כתובה בשפת Python ותשתמש במודול שנקרא PyGame אשר דרכו ניתן להשתמש בגרפיקה ברמה גבוהה. השימוש בגרפיקה מאפשר בניית משחקים, ובניית ממשק משתמש ברמה גבוהה אשר תאפשר חווית משתמש נעימה ונוחה.
* אפליקציה לטלפון – אפליקציה אשר תהיה מותקנת על הטלפון, ותציג את הוויז'ואל של שלט הקונסולה, אשר דרכו יתקבל הקלט מהמשתמש.   
  האפליקציה תהיה כתובה בפלטפורמת Android (המתבססת על כתיבה בשפת Java) אשר מאפשרת שימוש בגרפיקה, ובכך בניית ממשק משתמש ברמה גבוהה.

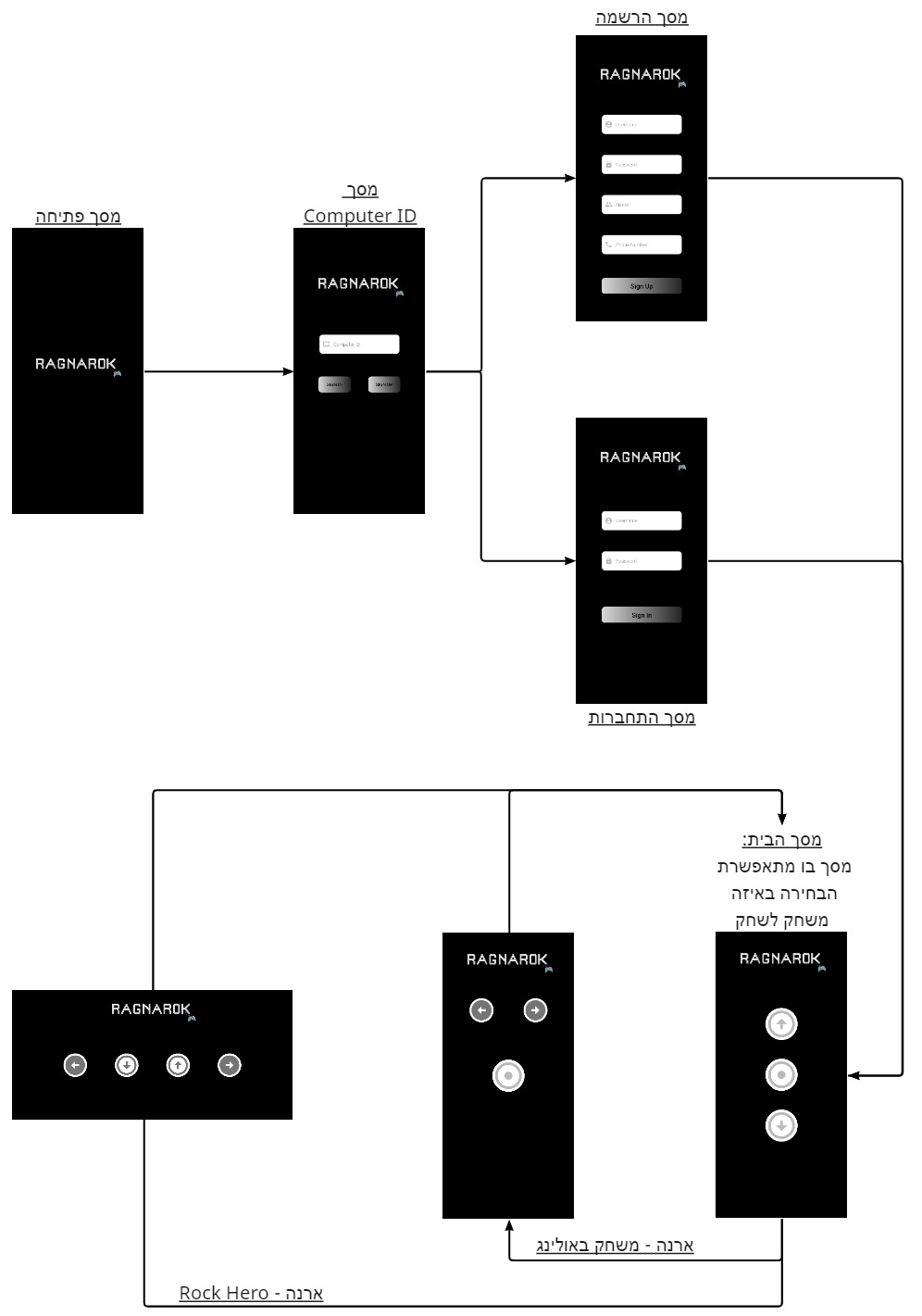
כלומר חלק אחד בממשק המשתמש יהיה כתוב בפלטפורמת Android, והחלק השני בו יהיה כתוב בשפת Python.

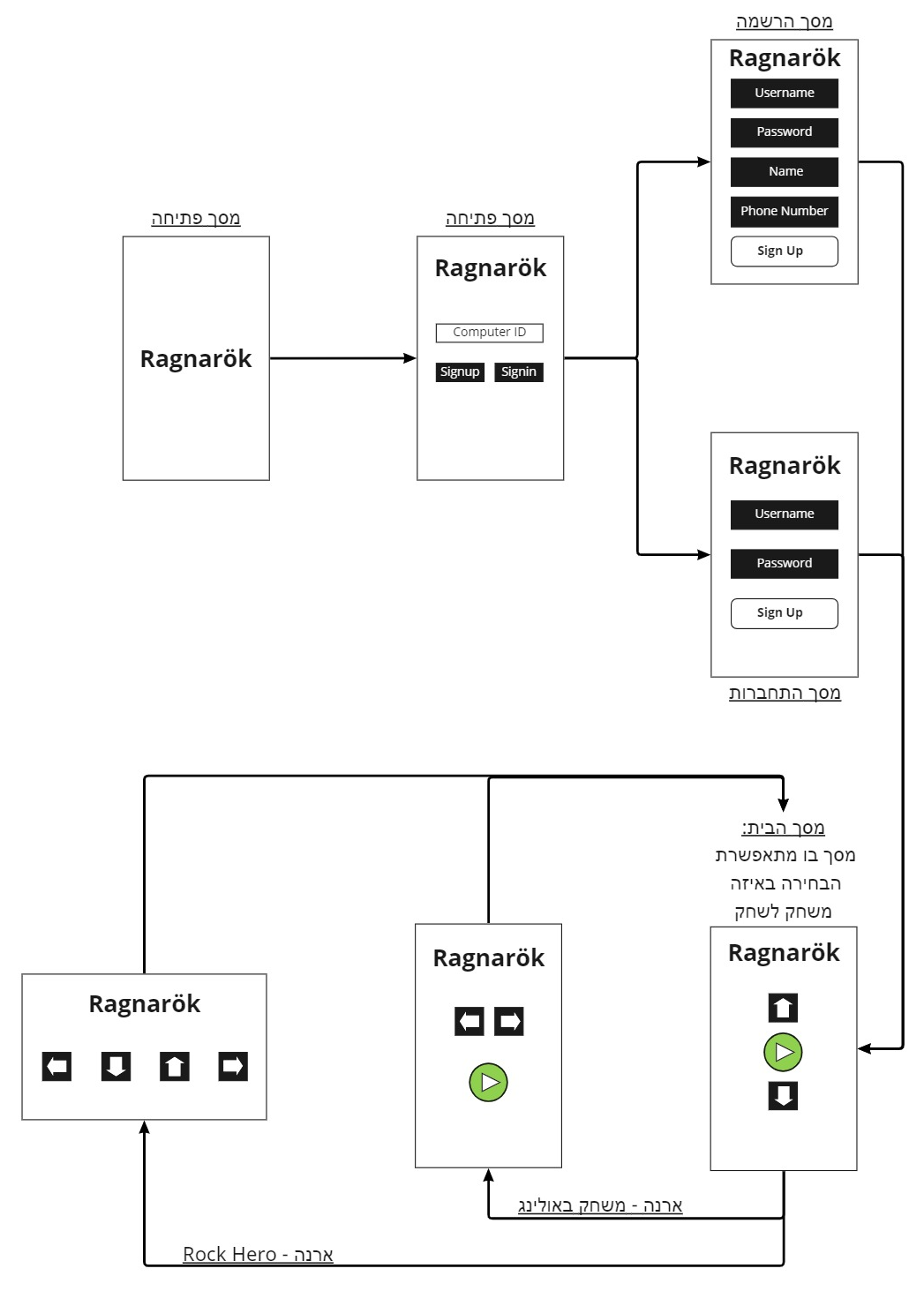
הקונסולה מאפשרת לשחק בשני משחקים שונים – "באולינג", "Rock Hero". לכל אחד מהמשחקים יש הוראות שונות:

* משחק באולינג:
* **מטרת המשחק –** מטרת המשחק היא לצבור מקסימום נקודות בסיומם של עשרת השלבים על ידי הפלת מרב הפינים בכל סבב. על מסלול הבאולינג מונחים עשרה פינים, והם מסודרים בסופו - הפלתם נעשית על ידי גלגול כדור כבד לעברם.
* **כללי המשחק –** משחק באולינג מורכב מ-10 סבבי משחק. בכל סבב כל אחד משני המשתתפים משליך ומגלגל את הכדור פעמיים על מסלול הבאולינג.
* **הוראות המשחק** – בכדי להשליך את הכדור על המשתמש ללחוץ על הכפתור במרכז המסך, להניע את הטלפון, ולבסוף לשחרר את הלחיצה.
* הוראות – "Rock Hero":
* **מטרת המשחק –** במהלך המשחק ירדו ממעלה המסך חצים בצורה רנדומלית כאשר ברקע יתנגן שיר. מטרת המשחק הינה לנסות ולהישאר כמה שיותר זמן במשחק מבלי להיפסל, וזאת על ידי לחיצה על מקשי החיצים בשלט הטלפון כאשר הם מגיעים לתחתית המסך למקום המסומן. כך יצבור השחקן כמה שיותר נקודות בנסיון לנצח במשחק.
* **כללי המשחק –** בתחילת המשחק לכל שחקן יהיו 3 לבבות. במהלך המשחק ירדו ממעלה המסך חצים בצורה רנדומלית כאשר ברקע יתנגן שיר רנדומלי, על השחקן ללחוץ על מקשי החיצים בשלט הטלפון כאשר הם מגיעים לתחתית המסך למקום המסומן. במידה ויתפספס חץ יורד לב אחד. במידה ומקש ילחץ ללא כל מטרה יורד חצי לב. כאשר לאחד מהשחקנים יגמרו החיים (כל שלושת הלבבות ירדו) המשחק יגמר, ואותו השחקן יפסיד.
* **הוראות המשחק** – על המסך ישנם 4 לחצנים המייצגים את החיצים בתחתית מסך המשחק: חץ ימינה, חץ שמאלה, חץ למעלה, חץ למטה.

**תרשים מסכי המחשב:**



**תרשים מסכי הטלפון:**



**בסיס הנתונים**

המשתמשים בקונסולה מחויבים לחבר בין אפליקציית הטלפון לבין תוכנת המחשב – לכן קיים צורך שלכל IP של המחשב עליו רצה תוכנת המחשב יהיה מזהה אשר בעזרתו יהיה ניתן לחבר בין השניים.

הצורך במבנה נתונים הוא של השרת המרכזי, ולו הגישה אליו – במבנה הנתונים ישמר מזהה המחשב   
וה-IP המשוייך אליו.

שדות הטבלה - הסבר:

* **ComputerID** – מספר מזהה של תכונת המחשב, ייחודי לכל מחשב.
* **IP** – IP של מחשב אשר התחבר לשרת המרכזי.

|  |  |
| --- | --- |
| IP | ComputerID |
|  |  |
|  |  |

המשתמשים בקונסולה מחויבים להתחבר למשתמשים שלהם או לחילופין ליצור משתמש חדש - לכן יש צורך שלכל משתמש באפליקציה יהיה חשבון פרטי משלו, בו ישמרו הנתונים האישיים שלו, ופרטים מהיסטוריית השימוש שלו בקונסולה.

הצורך במבנה נתונים הוא של השרת המרכזי, ולו הגישה אליו – במבנה הנתונים ישמרו פרטי החשבונות השונים של המשתמשים בקונסולה, הפרטים יכללו: פרט זיהוי של המשתמש (UserID), IP של מחשב המשתמש, שם המשתמש, הסיסמה, שם פרטי ומשפחה של המשתמש, מספר הטלפון של המשתמש, כמות המשחקים ששיחק, מספר הניצחונות שלו ומספר ההפסדים שלו.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AmountLosses | AmountWins | AmountGames | Phone | Name | Password | Username | IP | UserID |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

שדות הטבלה - הסבר:

* **UserID** – מספר מזהה של משתמש, ייחודי לכל משתמש
* **IP** – ה-IP של המחשב האחרון איתו המשתמש התחבר לשרת
* **Username** – שם משתמש, ייחודי לכל משתמש
* **Password** – סיסמה
* **Name** – שם מלא של המשתמש (שם פרטי + שם משפחה)
* **Phone** – טלפון שיהיה מקושר לחשבון, ייחודי לכל משתמש
* **AmountGames** – מספר המשחקים אשר בהם השתתף המשתמש
* **AmountWins** – מספר המשחקים אשר בהם ניצח המשתמש
* **AmountLosses** – מספר המשחקים אשר בהם הפסיד המשתמש

**עיצוב תוכנה**

**עיצוב תוכנה – Super Server:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| קובץ | חתימת פעולה | קלט | פלט | הסבר |
| Super Server.py  קובץ ה- Python אותו מריצים להרמת השרת המרכזי | def main(…) | אין | אין | הפעולה הראשית מגדירה משתנים לפתיחת התקשורת עם השרת. היא מכילה לולאה ראשית, אשר בעת חיבור לקוח היא מרימה thread אשר מנהל את התקשורת מול הלקוח. |
| def client\_ service(…) | client\_socket, client\_address | אין | פעולה המנהלת את התקשורת בין השרת המרכזי לתוכנת המחשב ולאפלקציית הטלפון, ופועלת בהתאם להודעות אשר התקבלו.  הפעולה מרימה מתוכה thread המסייע בשליחת ההודעות. |
| def handle\_ client\_ messages(…) | client\_socket, user\_id | אין | פעולה המסייעת ל- client\_service בשליחת ההודעות ללקוחות השונים. |
| def connect\_ player(..) | user\_id, type\_game | אין | פעולה המחברת בין שני משתמשים המעוניינים לשחק באותו המשחק. |
| def diffie\_ hellman(…) | A, p, g | B, k | פעולה זו מבצעת את האלגוריתם של Diffie hellman בעזרת המפתח הציבורי של הלקוח, מספר ראשוני גדול ומספר גדול הקטן מהמספר הראשוני (אשר אותם השרת יקבל מלקוח ה-Android). לאחר מכן הפעולה תחזיר את המפתח הפרטי המשותף של השרת ושל לקוח ה-Android. |
| def aes\_ decryption(…) | key, encrypted\_data | decrypted\_data | פעולה זו מקבלת את המפתח הפרטי המשותף ומידע המוצפן עימו, מפענחת את המידע, ומחזירה את המידע המפוענח. |
| DataBase.py - SQL operations file  קובץ המכיל את כל פעולות התקשורת הנדרשות בין ה- Super Server לבין ה-database | def \_\_init\_\_ (…) | database | אין | פעולה בונה של המחלקה האחראית לניהול התקשורת עם מסד הנתונים. |
| def add\_ computer(…) | IP | Computer\_id | פעולה האחראית על רישום של מחשב חדש למסד הנתונים. במידה והמחשב קיים כבר במסד הנתונים הפעולה תחזיר את ה- Computer\_id אשר מצוי בו ומשוייך למחשב. |
| def add\_ user(…) | IP, username, password, name, phone\_number | User\_id | פעולה האחראית על רישום משתמש חדש למסד הנתונים. |
| def user\_ login(…) | username, password, ip | User\_id | פעולה האחראית על בדיקה האם המשתמש רשום במסד הנתונים. במידה וכן, מעדכנת את המידע הרלוונטי, ומחזירה את ה-ID של המשתמש. |
| def get\_ by\_value(…) | table, column, value | מידע במסד הנתונים התואם את התנאים אשר הוגדרו על ידי קלט הפעולה. | פעולה האחראית על מתן מידע המצוי במסד הנתונים, על בסיס שם הטבלה, העמודה וערך מסויים. |
| def update(…) | table, column\_set, value\_set, column, value | אין | פעולה האחראית על עדכון מידע במסד הנתונים, וזאת על בסיס שם הטבלה, שם עמודה לזיהוי, ערך המצוי בעמודת הזיהוי, העמודה בה רוצים לשנות את המידע, המידע המעודכן. |
| tcp\_by\_size:  קובץ המכיל פעולות לניהול תקשורת (שליחת הודעות וקבלתן) עם הודעות בגדלים שונים אשר אינם קבועים. | def receive\_by \_size(…) | socket | data\_received | פעולה האחראית על קבלת מידע אשר גודלו איננו קבוע, המתקבל מ-socket מסויים . |
| Def send\_by \_size(…) | socket, binary\_data | אין | פעולה האחראית על שליחת מידע אשר גודלו איננו קבוע ל-socket מסויים. |

**משתנים קבועים – Super Server:**

|  |  |
| --- | --- |
| שם המשתנה הגלובלי | הסבר |
| ip = "0.0.0.0" | ה-IP אליו מתחברים בכדי להתחבר לשרת המרכזי. |
| port = 55555 | ה-port שדרכו מתחברים. |
| threads\_mode\_lock = threading.Lock() | משתנה המאפשר נעילת גישה למשתנה הגלובלי, threads\_mode, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת. |
| users\_lock = threading.Lock() | משתנה המאפשר נעילת גישה למשתנה הגלובלי, users, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת. |
| clients\_lock = threading.Lock() | משתנה המאפשר נעילת גישה למשתנה הגלובלי, clients, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת. |
| connect\_bowling\_lock = threading.Lock() | משתנה המאפשר נעילת גישה למשתנה הגלובלי, connect\_bowling, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת. |
| game\_mode\_lock = threading.Lock() | משתנה המאפשר נעילת גישה למשתנה הגלובלי, connect\_game, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת. |

**משתנים גלובליים – Super Server:**

|  |  |
| --- | --- |
| שם המשתנה הגלובלי | הסבר |
| threads = [] | רשימת ה-Threads הרצים בשרת המרכזי |
| threads\_mode = {} | לכל לקוח המתחבר לשרת נפתח thread, והוא יוצר thread עזר. המילון מצביע על מצבו של thread העזר, האם הוא עדיין רץ או שהרצתו הופסקה. המפתח של מילון זה מכיל את ה-client id אליו משוייך ה-thread, וערכו הינו משתנה בוליאני המייצג האם הוא עדיין רץ או שהרצתו הופסקה. |
| clients = {} | השרת המרכזי מנהל את התקשורת בין הלקוחות השונים. המילון מכיל רשימה של הודעות המיועדות לשליחה ללקוח. המפתח של מילון זה מכיל את ה-client id אליו משוייך ה-thread, וערכו הינו רשימה של הודעות המיועדות לשליחה ללקוח. |
| users = {} | השרת המרכזי אחראי בין היתר על חיבור בין שני משתמשים וניהול התקשורת בניהם, לכדי משחק אחד נגד השני. המפתח של מילון זה מכיל את ה-client id אליו משוייך ה-thread, וערכו הינו ה-id של היריב נגדו הוא משחק. |
| connect\_bowling = None | משתנה גלובלי המסייע לחיבור בין שני משתמשים למשחק "באולינג". |
| connect\_game = None | משתנה גלובלי המסייע לחיבור בין שני משתמשים למשחק  "Rock Hero". |

**עיצוב תוכנה – Computer:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| קובץ | חתימת פעולה | קלט | פלט | הסבר |
| Server.py – Main Computer  קובץ Python המחזיק את צד הקונסולה | def  main  (…) | אין | אין | פעולה המפעילה את הממשק הגרפי ההתחלתי של הקונסולה, מחברת את תוכנת המחשב לשרת המרכזי ופותחת thread לניהול התקשורת עימו, מחברת את אפליקציית הטלפון לתוכנת המחשב ופותחת גם לו thread לניהול התקשורת עימו. מעבר לכך פעולה זאת מרימה thread האחראי על ניהול מהלך המשחק. |
| def  game\_manager  (…) | screen,  clock,  font | אין | פעולה המנהלת את המשחק עצמו – הפן הגרפי שבו והלוגיקה העומדת מאחוריו. |
| def  intro  (…) | אין | אין | פעולה המפעילה את סרטון הפתיחה של המשחק. |
| def  opening\_screen  (…) | screen, computer\_id | אין | פעולה המדפיסה מסך ועליו ה-computer id של המחשב. |
| def  homepage\_screen  (…) | screen,  clock,  font | אין | פעולה האחראית על ניהול מסך הבית – הדפסתו וקביעת סוג המשחק המבוקש על ידי המשתמש. |
| def  waiting\_screen  (…) | screen,  clock | First \_player | פעולה האחראית על ניהול מסך ההמתנה – הדפסתו, והמתנה להודעה מהשרת המרכזי שהתחבר משתמש נוסף למשחק. |
| def  bowling\_game  (…) | screen,  clock,  first\_player | אין | פעולה אשר אחראית על ניהול משחק הבאולינג – הדפסת הממשק הגרפי שלו, וביצוע פעולות שונות בהתאם להודעות אשר מתקבלת תוכנת המחשב. |
| def  bowling\_player \_turn (…) | screen,  clock, scoreboard | scoreboard | פעולה המנהלת את המשחק בעת תורו של המשתמש עצמו. |
| def  bowling\_enemy \_turn  (…) | screen,  clock, scoreboard | scoreboard | פעולה המנהלת את המשחק בעת תורו של אויב המשתמש. |
| def  bowling\_winning \_screen  (…) | screen, clock | אין | פעולה האחראית על הדפסת מסך הניצחון של משחק הבאולינג (מופעלת במידה והמשתמש ניצח במשחק הבאולינג). |
| def  bowling\_losing \_game  (…) | screen, clock | אין | פעולה האחראית על הדפסת מסך ההפסד של משחק הבאולינג (מופעלת במידה והמשתמש הפסיד במשחק הבאולינג). |
| def  bowling\_draw \_screen  (…) | screen, clock | אין | פעולה האחראית על הדפסת מסך התיקו של משחק הבאולינג. |
| def  super\_server \_service  (…) | Server \_socket | אין | פעולה המנהלת את התקשורת בין תוכנת המחשב לבין השרת המרכזי, ופועלת בהתאם למידע המתקבל ממנו. |
| def  handle\_super \_server\_messages  (…) | Server \_socket | אין | פעולה האחראית על שליחת ההודעות מתוכנת המחשב לשרת המרכזי. |
| def  android\_client \_service  (…) | Client \_socket | אין | פעולה המנהלת את התקשורת בין תוכנת המחשב לבין אפליקציית הטלפון, ופועלת בהתאם למידע המתקבל ממנה. |
| def  handle\_android\_ messages  (…) | Client \_socket | אין | פעולה האחראית על שליחת ההודעות מתוכנת המחשב לאפליקציית הטלפון. |
| def  Rock\_Hero  (…) | אין | אין | פעולה אשר אחראית על ניהול המשחק "Rock Hero" – הדפסת הממשק הגרפי שלו, וביצוע פעולות שונות בהתאם להודעות אשר מתקבלת תוכנת המחשב. |
| tcp\_by\_ size.py:  קובץ המכיל פעולות לניהול תקשורת (שליחת הודעות וקבלתן) עם הודעות בגדלים שונים אשר אינם קבועים. | def  receive\_by \_size (…) | socket | Data \_received | פעולה האחראית על קבלת מידע אשר גודלו איננו קבוע, המתקבל  מ-socket מסויים. |
| Def  send\_by  \_size (…) | socket, binary \_data | אין | פעולה האחראית על שליחת מידע אשר גודלו איננו קבוע ל-socket מסויים. |
| Bowling\_ character.py  קובץ Python המחזיק את המחלקה של דמות השחקן של משחק הבאולינג | def  \_\_init\_\_  (…) | x\_pos, y\_pos, image\_type, player\_type, velocity | אין | פעולה בונה של מחלקת הדמות של שחקן הבאולינג. |
| def  update\_position (…) | type\_move | אין | פעולה המזיזה את מיקום דמות השחקן או לחילופין דמות האויב על פי הגבולות אשר נקבעו מבעוד מועד. |
| def  update\_image \_type  (…) | image\_type | אין | פעולה המגדירה מחדש שיוך תמונה לדמות מסוימת. |
| def  draw  (…) | screen | אין | פעולה המדפיסה את תמונת הדמות, על פי המיקום המשוייך לה. |
| Bowling\_ functions.py  קובץ Python המחזיק פעולות עזר של משחק הבאולינג | def  bowling \_background (…) | screen | אין | פעולה המדפיסה את רקע משחק הבאולינג. |
| def  pins  (…) | אין | pins\_list | פעולה היוצרת את הפינים של משחק הבאולינג. |
| Bowling\_ score.py  קובץ Python המחזיק מחלקה של טבלת הניקוד של משחק הבאולינג | def  \_\_init\_\_ (…) | x\_pos, y\_pos, width, height | אין | פעולה בונה של טבלת הניקוד של משחק הבאולינג. |
| def  update \_score  (…) | player\_type, score | אין | פעולה המעדכנת את ניקוד המשתמש או המשתמש הנגדי, וזאת בהתאם למספר הפינים אשר הצליחו להפיל. |
| def  update\_round \_number (…) | אין | אין | פעולה המעדכנת את מספר הסיבוב של משחק הבאולינג. |
| def  draw  (…) | screen | אין | פעולה האחראית על הדפסת טבלת הניקוד של משחק הבאולינג. |
| Bowling\_ ball.py  קובץ Python המחזיק את מחלקת כדור הבאולינג של משחק הבאולינג | def  \_\_init\_\_ (…) | x\_pos, y\_pos, image\_type, player\_type | אין | פעולה בונה של מחלקת כדור הבאולינג. |
| def  throwing \_bowling\_ball  (…) | Acceleration, throw\_angle | אין | פעולה המגדירה את מהירות התקדמות כדור הבאולינג, ואת כיוון התנועה. |
| def  update\_position  (…) | אין | אין | מעדכן את מיקום הכדור ואת תמונתו להצגה הוייז'ואלית. |
| def  ball\_position  (…) | x\_pos, y\_pos | אין | פעולה המעדכנת את מיקום הכדור. |
| def  update\_x \_pos  (…) | Update \_type, amount | אין | מעדכן את מיקום הכדור בציר x. |
| def  draw  (…) | screen | אין | פעולה האחראית על הדפסת תמונת הכדור. |
| Bowling\_ pin.py  קובץ Python המחזיק מחלקה המייצגת פין אחד של משחק הבאולינג | def  \_\_init\_\_  (…) | x\_pos, y\_pos | אין | פעולה בונה של מחלקת פין באולינג בודד. |
| def  get\_position  (…) | אין | (x\_pos, y\_pos) | פעולה המחזירה את מיקומו של פין הבאולינג. |
| Rock\_ Hero\_ Arrow.py  קובץ Python המחזיק מחלקה המייצגת חץ אחד של המשחק "Rock Hero" | def  \_\_init\_\_  (…) | type, velocity, screen, position, type\_ explosion | אין | פעולה בונה של מחלקת החץ של המשחק  "Rock Hero". |
| def  update  (…) | אין | אין | פעולה המעדכנת בציר y את מיקומו של החץ ומדפיסה אותו. |
| Rock\_ Hero\_ Life.py  קובץ Python המחזיק מחלקה המייצגת חיים של משתמש המשחק ב- "Rock Hero" | def  \_\_init\_\_  (…) | screen, xpos, ypos | אין | פעולה בונה של מחלקת החיים של משתמש במשחק "Rock Hero". |
| def  remove\_life  (…) | type | אין | פעולה המורידה מכמות החיים של השחקן או לחילופין מכצות החיים של הלב. כמות הורדת החיים תלויה במשתנה type המוכנס כקלט לפעולה. |
| def  print\_life  (…) | אין | אין | פעולה המדפיסה את החיים של השחקן או לחילופין של האויב. |
| Rock\_ Hero\_ Score.py  קובץ Python  המחזיק מחלקה המייצגת את הניקוד שצובר המשתמש במהלך משחק "Rock Hero" | def  \_\_init\_\_  (…) | screen, xpos, ypos | אין | פעולה בונה של מחלקת הניקוד של המשחק "Rock Hero". |
| def  add\_score  (…) | value | אין | פעולה אשר אחראית על הוספת ניקוד למשתמש או לחילופין למשתמש הנגדי. |
| def  print\_score  (…) | אין | אין | פעולה המדפיסה את הניקוד של המשתמש או לחילופין את של המשתמש הנגדי. |

**משתנים קבועים – Computer:**

|  |  |
| --- | --- |
| שם המשתנה הגלובלי | הסבר |
| ip = "0.0.0.0" | ה-IP אליו מתחברים בכדי להתחבר לתוכנת המחשב. . |
| port\_android\_client = 50000 | ה-port דרכו מתחברת אפליקציית הטלפון לתוכנת המחשב (השרת). |
| ip\_super\_server = "IP" | ה-IP אליו מתחברת תוכנת המחשב (לקוח) לשרת המרכזי. |
| port\_super\_server = 55555 | ה-port דרכו מתחברת תוכנת המחשב (לקוח) לשרת המרכזי. |
| android\_message\_lock = threading.Lock() | משתנה המאפשר נעילת גישה למשתנה הגלובלי, android\_message, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת. |
| android\_thread\_mode\_lock = threading.Lock() | משתנה המאפשר נעילת גישה למשתנה הגלובלי, android\_thread\_mode, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת. |
| super\_server\_message\_lock = threading.Lock() | משתנה המאפשר נעילת גישה למשתנה הגלובלי, super\_server\_message, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת. |
| super\_server\_thread\_mode\_lock = threading.Lock() | משתנה המאפשר נעילת גישה למשתנה הגלובלי, super\_server\_thread\_, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת. |
| game\_message\_lock = threading.Lock() | משתנה המאפשר נעילת גישה למשתנה הגלובלי, game\_message, בכדי להגן על המשתנה מהתנגשויות, עקב גישה ממספר threads שונים הרצים באותו העת. |

**משתנים גלובליים – Computer:**

|  |  |
| --- | --- |
| שם המשתנה הגלובלי | הסבר |
| threads = [] | רשימת ה-Threads הרצים בתוכנת המחשב |
| threads\_mode = True | בתוכנת המחשב רצים כמה threads במקביל. מטרת המשתנה הינה שברגע סגירת ה-thread המנהל את המשחק, כל שאר ה-threads הרצים גם באותה העת יסגרו גם הם. |
| android\_message = [] | תוכנת המחשב (שרת) מקיים תקשורת בינו לבין אפליקציית הטלפון (לקוח). הרשימה מכילה הודעות המיועדות לשליחה ללקוח. |
| android\_thread\_mode = True | ברגע הרצת ה-thread המנהל את התקשורת עם אפליקציית הטלפון, מורץ מתוכו thread עזר. מטרת המשתנה הינה שברגע סגירת ה-thread המנהל את התקשורת עם אפליקציית הטלפון, גם thread העזר יסגר באותה העת. |
| super\_server\_message = [] | תוכנת המחשב (לקוח) מקיים תקשורת בינו לבין השרת המרכזי. הרשימה מכילה הודעות המיועדות לשליחה לשרת המרכזי. |
| super\_server\_thread\_mode = True | ברגע הרצת ה-thread המנהל את התקשורת עם השרת המרכזי, מורץ מתוכו thread עזר. מטרת המשתנה הינה שברגע סגירת ה-thread המנהל את התקשורת עם השרת המרכזי, גם thread העזר יסגר באותה העת. |
| game\_message = [] | קיים thread אשר תפקידו לנהל את מהלך המשחק עצמו. על thread זה להיות מעורב וחשוף להודעות העוברות דרך תוכנת המחשב. רשימה זאת מכילה את ההודעות הרלוונטיות אשר התקבלו מאפלקציית הטלפון והשרת המרכזי. |
| type\_game = None | משתנה המייצג סוג המשחק הנבחר על ידי המשתמש. |
| game\_mode = True | משתנה המייצג האם המשחק מתנהל זה עתה, או שהוא נגמר. |

**עיצוב תוכנה – Phone:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| קובץ | חתימת פעולה | קלט | פלט | הסבר |
| MainActivity  קובץ Java המשמש כמסך הפתיחה | protected void onCreate (…) | Bundle savedInstanceState | - | פעולה זאת מפעילה את סרטון הפתיחה, ובסיומו מבצעת העברה ל-Activity הבא אליו הוא מפנה. |
| Startup Activity  קובץ Java המשמש כמסך הקישור בין אפליקציית הטלפון לתוכנת המחשב | protected void onCreate (…) | Bundle savedInstanceState | - | פעולה זו הינה בסיסית בכל activity, ודרושה לשם הפעלתו. |
| public void handle Message (…) | Message message | - | פעולה זו מקבלת את ההודעות מהשרת ומטפלת\מגיבה להן בהתאם לקוד ההודעה שלהם והתוכן שלהן. |
| public void send\_message (…) | String data | - | פעולה זו פותחת תקשורת עם השרת בעזרת TcpBySize, ושולחת את data. |
| public void sign\_up (…) | View v | - | פעולה אשר שולחת הודעה לשרת המרכזי, וקובעת את סוג החיבור הרצוי - הרשמה. |
| public void sign\_in (…) | View v | - | פעולה אשר שולחת הודעה לשרת המרכזי, וקובעת את סוג החיבור הרצוי – התחברות למשתמש קיים. |
| public void diffie\_hellman (…) | אין | אין | פעולה זו מגרילה מספר שמשמש כמפתח פרטי, ומחשבת בעזרת מספר ראשוני גדול, מספר גדול הקטן מהמספר הראשוני (קבועים) והמפתח הפרטי את המפתח הציבורי ושולחת אותו לשרת ביחד עם המספרים הנ"ל (קבועים). |
| public String aes\_encryption (…) | key, data | encrypted \_data | פעולה זו מקבלת את המפתח הפרטי המשותף של הלקוח עם השרת, ואת המידע להצפין, ומצפינה אותו בעזרת הצפנת AES. הפעולה מחזירה את המידע המוצפן. |
| SignIn Activity  קובץ Java המשמש כמסך התחברות למשתמש קיים | protected void onCreate (…) | Bundle savedInstanceState | - | פעולה זו הינה בסיסית בכל activity, ודרושה לשם הפעלתו.  בפעולה זאת מתבצעת שליפה של ה-IP אשר הועבר מה-Activity הקודם. |
| public void handleMessage (…) | Message message | - | פעולה זו מקבלת את ההודעות מהשרת ומטפלת\מגיבה להן בהתאם לקוד ההודעה שלהם והתוכן שלהן. |
| public void send\_message (…) | String data | - | פעולה זו פותחת תקשורת עם השרת בעזרת TcpBySize, ושולחת את data. |
| public void sign\_in (…) | View v | - | פעולה המקבלת מהמשתמש כקלט את ה-username וה-password של המשתמש אליו הוא מנסה להתחבר. הפעולה בונה הודעת התחברות, ומבצעת שליחה לתוכנת המחשב. |
| SignUp Activity  קובץ Java המשמש כמסך הרשמה של משתמש חדש | protected void onCreate (…) | Bundle savedInstanceState | - | פעולה זו הינה בסיסית בכל activity, ודרושה לשם הפעלתו.  בפעולה זאת מתבצעת שליפה של ה-IP אשר הועבר מה-Activity הקודם. |
| public void handleMessage (…) | Message message | - | פעולה זו מקבלת את ההודעות מהשרת ומטפלת\מגיבה להן בהתאם לקוד ההודעה שלהם והתוכן שלהן. |
| public void send\_message (…) | String data | - | פעולה זו פותחת תקשורת עם השרת בעזרת TcpBySize, ושולחת את data. |
| public void sign\_up (…) | View v | - | פעולה המקבלת מהמשתמש כקלט username, password, name, phone number וזאת במטרה ליצור משתמש חדש. הפעולה בונה הודעת הרשמה, ומבצעת שליחה לתוכנת המחשב. |
| Homepage Activity  קובץ Java המשמש כשלט עימו בחור המשתמש את המשחק המבוקש | protected void onCreate (…) | Bundle savedInstanceState | - | פעולה זו הינה בסיסית בכל activity, ודרושה לשם הפעלתו.  בפעולה זאת מתבצעת שליפה של ה-IP אשר הועבר מה-Activity הקודם. |
| public void handleMessage (…) | Message message | - | פעולה זו מקבלת את ההודעות מהשרת ומטפלת\מגיבה להן בהתאם לקוד ההודעה שלהם והתוכן שלהן. |
| public void send\_message (…) | String data | - | פעולה זו פותחת תקשורת עם השרת בעזרת TcpBySize, ושולחת את data. |
| public void arrow\_up (…) | View v | - | פעולה המבצעת בניית הודעת תזוזה מעלה, ושליחת ההודעה לתוכנת המחשב. |
| public void arrow\_ down (…) | View v | - | פעולה המבצעת בניית הודעת תזוזה מטה, ושליחת ההודעה לתוכנת המחשב. |
| public void choice (…) | View v | - | פעולה המבצעת בניית הודעת בחירת משחק, ושליחת ההודעה לתוכנת המחשב. |
| Bowling Game  מסך המשמש כשלט משחק הבאולינג | protected void onCreate (…) | Bundle savedInstanceState | - | פעולה זו הינה בסיסית בכל activity, ודרושה לשם הפעלתו.  בפעולה זאת מתבצעת שליפה של ה-IP אשר הועבר מה-Activity הקודם. |
| public void handleMessage (…) | Message message | - | פעולה זו מקבלת את ההודעות מהשרת ומטפלת\מגיבה להן בהתאם לקוד ההודעה שלהם והתוכן שלהן. |
| public void send\_message (…) | String data | - | פעולה זו פותחת תקשורת עם השרת בעזרת TcpBySize, ושולחת את data. |
| public boolean onTouch (…) | View v, MotionEvent event | - | פעולה הבודקת האם מתבצעת לחיצה ארוכה על כפתור, ופועלת בהתאם. |
| public void onSensorChanged (…) | SensorEvent event | - | פעולה המחשבת את תאוצת הטלפון, וכיוון התאוצה. |
| public float direction\_injury (…) | float xValue, float yValue | - | פעולה המחשבת את כיוון תאוצת הטלפון. |
| public void arrow\_right (…) | View view | - | פעולה המבצעת בניית הודעת תנועה ימינה, ושליחת ההודעה לתוכנת המחשב. |
| public void arrow\_left (…) | View view | - | פעולה המבצעת בניית הודעת תנועה שמאלה, ושליחת ההודעה לתוכנת המחשב. |
| RockHero  מסך המשמש כשלט המשחק "Rock Hero" | protected void onCreate (…) | Bundle savedInstanceState | - | פעולה זו הינה בסיסית בכל activity, ודרושה לשם הפעלתו.  בפעולה זאת מתבצעת שליפה של ה-IP אשר הועבר מה-Activity הקודם. |
| public void handleMessage (…) | Message message | - | פעולה זו מקבלת את ההודעות מהשרת ומטפלת\מגיבה להן בהתאם לקוד ההודעה שלהם והתוכן שלהן. |
| public void send\_message (…) | String data | - | פעולה זו פותחת תקשורת עם השרת בעזרת TcpBySize, ושולחת את data. |
| public void arrow\_right (…) | View view | - | פעולה המבצעת בניית הודעת תנועה ימינה, ושליחת ההודעה לתוכנת המחשב. |
| public void arrow\_left (…) | View view | - | פעולה המבצעת בניית הודעת תנועה שמאלה, ושליחת ההודעה לתוכנת המחשב. |
| public void arrow\_up (…) | View view | - | פעולה המבצעת בניית הודעת תנועה מעלה, ושליחת ההודעה לתוכנת המחשב. |
| public void arrow\_down (…) | View view | - | פעולה המבצעת בניית הודעת תנועה מטה, ושליחת ההודעה לתוכנת המחשב. |

**רפלקציה – סיכום אישי**

במשך השנתיים אשר קדמו לשנה הזאת למדנו והשכלנו רבות, נגענו בנושאים שלא הכרנו, נפתחנו לעולם חדש ומעניין, עברנו שיעורים רבים, ביצענו פרויקטים ועשינו עבודות. עם תחילת השנה השלישית בלימודי מדעי המחשב נפתחנו לסוג חדש של למידה, הגענו לשיא העשייה שלנו. היינו צריכים לרכז את כל הידע והיכולות שצברנו לכדי פרויקט גמר.   
אין ספק עשיית הפרויקט איננה הייתה קלה – החשיבה מאחורי בחירת הפרויקט, ביצוע החקר הגדול והלמידה העצמאית, עשיית פרויקט בהיקף גדול בהרבה ממה שהכרנו, כל אלו לא הקלו עלינו.   
אך, באופן אישי, הפרויקט הסב לחיי סיפוק רב, גאווה אישית, צמיחה ובנייה נפשית. התועלת שהביא לחיי ביצוע הפרויקט היה שווה את כל הקשיים שליוו אותו לאורך עשייתו.

מהפרויקט למדתי רבות – מעבר לחומר היבש אותו למדתי במיוחד בשביל הפרויקט, עשייתו גרמה לי להבין כיצד עליי לנהל פרויקט בסדר גודל כזה, היא לימדה אותי כיצד להתנהל בתוך מסגרת זמן, היא הקנתה לי נסיון מקצועי. הפרויקט גרם לי להתעקש ולא לוותר - כל פרויקט מלווה בבאגים ובעיות לוגיות, אך גם בקושי ובייאוש ההבנה שבסופו של דבר אצליח להגיע אל הפתרון הייתה במוחי, ובזכותה לא וויתרתי עד לכדי פרויקט מוגמר שעובד והנני גאה לעמוד מאחוריו ולהגיד שאני הכנתי אותו.

לו הייתי חוזר לתחילת השנה עם הידע שברשותי היום הייתי מבצע את החקר בשלבים בהתאם לחלק בו אני נמצא. במהלך הפרויקט ביצעתי את החקר בבת אחת, דבר שהצריך ממני לחזור וללמוד בחלק מהפעמים שוב את החומר כיוון שהיה פער זמן גדול מעת למידת החומר לעת ביצוע חלק זה בפרויקט.

אני חושב שאחד מהבעיות המרכזיות שליוו אותי לאורך הפרויקט היו חוסר ביטחון בעצמי, פחדתי שלא יהיה ביכולתי לעמוד בעשיית פרויקט בסדר גודל כזה, וביכולות שחשבתי שהוא מצריך. אם הייתי יותר מאמין בבעצמי בתחילת עשיית הפרויקט, אני מאמין שהעבודה שלי עליו הייתה יותר מהירה ויעילה, היא לא הייתה כוללת זמן מבוזבז (בחלק מן הפעמים חוסר הביטחון שלי מנע ממני מלהתקדם בקצב גבוהה יותר בעשיית הפרויקט). אך, עם הזמן צברתי יותר ויותר ביטחון בעצמי, הצלחתי לבנות מאפס פרויקט שאני גאה בו ושמח שניתנה לי ההזדמנות לבצע אותו.

הפרויקט שלי, "Ragnarök", הינו סיכום בעבורי של 3 שנים מופלאות – 3 שנים מאתגרות ומעניינות של מגמה מיוחדת, מגמת הסייבר בתיכון הרצוג. בזכות שלושת השנים הללו מצאתי את המקצוע בו אני חולם לעסוק, ואני בטוח שאחרי ביצוע הפרויקט הנ"ל אמשיך ללמוד, לחקור ולעסוק בתחום המחשבים בהמשך חיי הבוגרים.



**ביבליוגרפיה – מקורות המידע העיקריים של הפרויקט**

|  |  |
| --- | --- |
| שם מקור המידע | כתובת מקור המידע |
| Caleb Curry - Android App Development in Java All-in-One Tutorial Series | <https://www.youtube.com/watch?v=tZvjSl9dswg> |
| accelerometer sensor android | <https://developer.android.com/guide/topics/sensors/sensors_motion> |
| pygame | <https://www.pygame.org/news> |

