Snake Eye

|  |  |
| --- | --- |
| שם התלמיד: נביה אבו חמוד |  |
| ת.ז:212230981  בית ספר והכיתה: בית ספר למנהיגות ומדעים דרכא ירכא , כיתה י"ב  שם המוקד ומספר הכיתה: מוקד אורנים, כיתה 17  שם הרש"צ: יובל עופר |  |
| שם המנטור: אין |  |
|  |  |

**עבודת גמר**

מסלול 14.50

הנדסת תכנה -  שם המסלול

שנת הלימודים תשע"ח

|  |  |
| --- | --- |
| עמוד ראשון – שער | 1 |
| מבנה התיק - פרקים | 2 |
| תיאור הפרויקט | 3 |
| פתיחה, תיאור כללי והצעת פרויקט | 4 |
| מסמך ייזום אפיון וארכיטקטורה | 6 |
| מחלקות\מתודות | 15 |
| מסמך מפרט התוכנה Detailed Design | 16 |
| מסמך הבדיקות | 18 |
| הוראות שימוש – התקנה והפעלה | 19 |
| משוב | 23 |
| קוד הפרויקט | 24 |
| התייחסות אישית | 25 |
| רשימת מקורות (ביבליוגרפיה) | 27 |

**תיאור הפרויקט**

הפרויקט הוא בעצם שילוב טכנולוגיית הבניה מלאכותית בתוך משחק ה"נחש". שזה מאפשר לנו לבנות בוט (מכונה) הלומדת לשחק המשחק בעצמה (עם אפשרות להתקדם ולהתפתח בהמשך) בנוסף המשחק מאפשר למשתמש לשחק במשחק בעצמו . וגם מאפשר למשתמש לאמן בוטים שונים משלו וגם להשתמש בבוטים אחרים .

מטרת הפרויקט היא ללמוד נושא חדש - AI (בניה מלאכותית) שהוא הנושא הלהיט בתקופה הזאת , תוך שילוב החומר שלמדנו במהלך תכנית מגשימים שקשור לתכנות ולבניית משחקים בעזרת PYGAMES.

המשתמשים בפרויקט יוכלו לראות את תהליך הלמידה וההתפתחות של הבוט שבנינו.

הפרויקט דורש למידה מעמיקה בנושא בניה מלאכותית וגם איך לשלב את זה בקוד כלומר איך לממש את זה.

מושגים בסיסים:

* **רשת נוירונים** - רשתות נוירונים הן רשתות מחשבים מתקדמות שמחקות את החשיבה האנושית. נוירון הוא תא עצב. רשת נוירונים היא "רשת עצבית" מלאכותית, שמעתיקה את הפעילות משכב לשכבה. כל נוירון מחובר בסינפסות אל נוירונים נוספים, במה שקרוי "רשת נוירונית".
* **אלגוריתמים גנטיים** - מתאר משפחה של אלגוריתמים לחיפוש, מידול ומיטוב (אופטימיזציה), שבהם משלבים זה בזה אלמנטים של פתרונות אפשריים לבעיה, ומפעילים הליכים של ברירה מלאכותית כדי לבחור את המועמדים שיעברו לשלבים הבאים. רעיון תכנותי בסיסי זה מושפע מההצלחה של האבולוציה בפתרון בעיות אמיתיות.

**פתיחה, תיאור כללי והצעת פרויקט**

בחרנו בפרויקט זה כי המטרה העיקרית שלנו היא תהליך הלמידה שלנו - המשימה שהבוט יפתור לא תהיה חשובה או חדשנית אך דרכה אנחנו נוכל ללמוד בצורה הטובה ביותר איך לבנות בוט המשלב בין 'למידה עצמית' לתהליך של 'התפתחות גנטית', כלומר בדרך הזו אנחנו נכנס לעולם הבניה המלאכותית וככה נוכל להבין איך זה עובד ומתנהל בצורה הכי יעילה .

הפרויקט יבנה למטרת למידה אך כל מי שהנושא מעניין אותו יוכל להפעיל את הפרויקט ולראות את תהליך הלמידה וההתפתחות של הבוט במשחק, וגם יכול להשתמש בקוד שכתבנו כדי ללמוד אותו.

למשתמשים בפרויקט לא ידרש ידע קודם לשימוש אך כדי להבין את התהליך אותו הם רואים הם יצטרכו ידע כללי בנושא תכנות ובניה מלאכותית.

לכל המשתמשים בפרויקט יהיה את אותו התפקיד בתפעול הפרויקט.

***הצעת פרויקט:***

הפרויקט הוא בעצם שילוב טכנולוגיית הבניה מלאכותית בתוך משחק ה"נחש". שזה מאפשר לנו לבנות בוט (מכונה) הלומדת לשחק המשחק בעצמו (עם אפשרות להתקדם ולהתפתח בהמשך) בנוסף המשחק מאפשר למשתמש לשחק במשחק בעצמו .

הפרויקט מורכב מכמה חלקים שונים שבסוף הם קשורים זה לזה:

* משחק הנחש שבניתי אותו בעזרת שפת PYTHON .
* רשת נוירונים ואלגוריתמים גנטיים שזה בא מתחום הAI גם כתוב בעזרת PYTHON.
* שרת וצד לקוח UI, כתובים באמצעות PYTHON .
* מסד נתונים .

ממשק המשתמש יאפשר ללקוח להירשם למשחק דרך שם משתמש וסיסמה, יש אפשרות לשחק את המשחק ויש אפשרות לצפות בBOT שמשחק את המשחק בעצמו וגם המשתמש יכול לאמן BOT משלו .

תחומים שהפרויקט מכיל:

* תקשורת (שרת לקוח)
* ניהול מסדי נתונים – משתמשים בזה כדי לשמור את הפרטים השונים הקשורים במשתמשים וגם כדי לאחסן את הגרסאות השונות של הBOT.
* AI – בניה מלאכותית
* ממשק משתמש גרפי בשפת פייתון

## מסמך ייזום אפיון וארכיטקטורה

**מסמך ייזום:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **סדר** | **משימה** | **תלויות תשתית** | **נימוק והערות** | **רכיב רלוונטי** |
| 1 | בניית בסיס למשחק (UI) |  | בניית משחק SNAKE בסיסי. |  |
| 1 | למידה - איך לעבוד עם tensorflow |  | ללמוד איך ליישם את הקוד בעזרת הספרייה. |  |
| 1 | חלוקה ובנייה בקוד של רשת הנוירונים | Tensorflow למידה | מימוש ראשוני של רשת הניורונים. |  |
| 1 | בחירה של חישובים מתאמים לניורונים | תשתית לאלגוריתם הגנטי | בחירה של חישובים מתאמים לניורונים אשר בעזרת המשקלים המתאימים יעשו את הבחירות הרצויות |  |
| 1 | עדכון המשחק | בסיס למשחק | הוספת חישוב פרמטרים בשביל רשת הניורונים | UI |
| 1 | חיבור רשת הניורונים למשחק SNAKE שבנינו | רשת הנוירונים | בניית הבסיס לרכיב הלוגיקה | UI |
| 1 | בניית תשתית לאלגוריתם הגנטי | רשת הנוירונים | קבלת מטריצה של משקלים שיחושבו בהמשך על ידי האלגוריתם הגנטי |  |
| 1 | הרצה ובדיקות | רשת הנוירונים | הרצה ובדיקות של המשחק מחובר לרשת הניורונים | UI |
| 2 | לתכנן את האלגוריתם הגנטי | רשת נוירונים | אחראי על חישוב המשקלים של רשת נוירונים | UI |
| 2 | להכין את החישובים שנשתמש בהם | רשת נוירונים | כוללהמשקלים |  |
| 2 | לבחור בספריה מתאימה מTENSORWFLOW | Tensorflow למידה | יש הרבה ספריות ב Tensorflow, אז נבחר הספריה הכי מתאימה שתעזור לנו עם החשובים |  |
| 2 | להתחיל לכתוב קוד | רשת נוירונים |  |  |
| 2 | להתאים את הקוד עם רשת נוירונים | רשת נוירונים |  |  |
| 2 | הרצה ובדיקות |  | הרצה ובדיקות של המשחק מחובר לרשת הניורונים |  |
| 3 | תכנון DB מקומי | אלגוריתם גנטי | כל התוצאות של החישובים נשמרים במסד נתונים מקומי כדי לא לאבד את הנתונים של ה BOT . | LOGIC |
| 3 | DB WRAPPER | DB | אחראי על התקשורת עם מסד הנתונים המקומי | LOGIC,DB |
| 3 | הכנת המשחק לקראת ספרינט 4 | אלגוריתם גנטי | הכנת בסיס או איך יראה ה UI של המשחק. |  |
| 3 | בניית תפריט בסיסי |  | תפריט ראשי לפרוייקט שיוכל להפעיל את המשחק |  |
| 3 | פונקציית שמירה של הבוט | DB WRAPPER | פונקצייה אשר תוכל לשמור את הדור הנוכחי לתוך מסד הנתונים | DB |
| 3 | פונקציית טעינה של הבוט | DB WRAPPER | פונקצייה אשר תוכל לטעון דור שנשמר מתוך מסד הנתונים | DB |
| 4 | UI - משחק שמציג נתונים |  | ממשק של המשתמש שמאפשר לשחק במשחק ולצפות בבוט. |  |
| 4 | אלגוריטם גנטי אינטרקטיבי |  | אפשרות לשלוט בפרמטרים בזמן הרצת המשחק |  |
| 4 | UI - תפריט |  | ממשק של המשתמש שמאפשר לו לפתוח מסכים שונים בפרוייקט . |  |
| 4 | מסכי הרשמה למשחק | SERVER | ממשק של המשתמש שמאפשר להרשם כמשתמש הישמר על השרת. | NETWORKING |
| 4 | מסכי התחברות למשחק | SERVER | ממשק של המשתמש שמאפשר להתחבר למשתמש השמור על השרת. | NETWORKING |
| 4 | UI - פעולות על מסד הנתונים | SERVER | ממשק של המשתמש הנותן לו לעדכן את מסד הנתונים | NETWORKING |
| 5 | הרשמה והזדהות - פיתוח תקשורת צד שרת | UI -מסכי התחברות והרשמה | כל חיבור לשרת ידרוש מהמשתמש להזדהות בהתאם לפרוטוקול שיוגדר, ויבדוק את הנתונים מול בסיס הנתונים בצד השרת. בנוסף אפשרות של מספר קליינטים להתחבר במקביל | שרת - תקשורת |
| 5 | הרשמה והזדהות - תקשורת צד לקוח | UI -מסכי התחברות והרשמה | בהתאם לפרוטוקול | לקוח - תקשורת |
| 5 | DB WRAPPER | DB | אחראי על התקשורת עם מסד הנתונים בשרת | LOGIC,DB |
| 5 | DB למשתמשים |  |  |  |

***מסמך אפיון:***

**איטרציה 1 - לבנות רשת נוירונים neural network**

תקציר, הסבר ותוצר:

* רשת נוירונים היא הבסיס לAI, והיא אחראית על קבלת המידע מהמשחק(כלומר מהשחקן) כדי לסייע בתהליך הלימוד.
* התוצר של איטרציה זו תהיה רשת נוירונים שתהיה מוכנה לקבל מידע ותעביר אותו הלאה לשלב הבא שהאלגוריתם הגנטי אחראי עלו.

תכולות טכנולוגיות:

* בניית בסיס למשחק:

-בניית UI בסיסי עם משחק בו המשתמש יכול לשחק.

* תכנון הנוירונים :

-יש לנו בהתחלה עשר ניורונים האחראים על הקלט שהוא (אם יכול להמשיך קדימה,שמאלה,ימינה, ואת כיוון האוכל).

-ויש גם עוד שלושה נוירונים האחראים על הפלט.

* חישובים לניורונים:

- בחירה של חישובים ופרמטרים מתאימים אשר בהם רשת הניורונים תוכל להשתמש.

* עדכון המשחק:

-יישום של החישובים בקוד.

* חיבור רשת הניורונים למשחק:

-הוספת קריאות מתאימות לשימוש של המשחק ברשת הניורונים.

* בניית תשתית לאלגוריתם הגנטי:

-הוספת קריאה לפונקציה אשר תוכל לקבל את תוצאות המשחק ולערוך את המשקלים ברשת הניורונים.

* נוסיף את הספריה המתאימה לקוד חיצוני שנכתוב, וזה יעזור  לנו לבנות את רשת הנוירונים .

קשיים ומקרי קצה שעשויים להתקל:

* הנוירונים לא יתאימו מבחינת הלוגיקה.
* יש מצב שנתקל בבעיה של קישור הקוד של רשת הנוירונים עם קוד המשחק.

# **איטרציה 2 - בניית תשתית לאלגוריתם גנטי**

תקציר, הסבר ותוצר:

* לבנות אלגוריתם גנטי שבעזרתו אנחנו יכולים ללמד את רשת הנוירונים בצורה יעילה יותר ומהירה יותר .

תכולות טכנולוגיות:

* תכנון את האלגוריתם הגנטי  : יעזור בחישוב המשקלים של רשת נוירונים.
* הכנת החישובים שנשתמש בהם שזה יעזור לנו לקבל תוצאות הכי גבוהות.
* התאמת הקוד של רשת נוירונים עם קוד אלגוריתם גנטי.

# **איטרציה 3 - לבנות DB לשמירת גרסאות ה BOT השונות.**

תקציר, הסבר ותוצר:

* נבנה DB מקומי אשר יוכל לשמור ולטעון גרסאות שונות של הבוט/ דורות של הבוט בשביל המשתמש.

תכולות טכנולוגיות:

* אנו נשתמש בפעולות קיימות בספרייה NUMPY ו PICKLE שיעזור לנו בשמירת ושליפת המידע שנוצר כתוצאה של הרצת הקוד.
* נשמור את המידע בקבצים כנראה בבינארית.
* הספרינט יאפשר לנו להמשיך ללמד גרסאות של הבוט לאורך הרצות שונות.
* נוסיף גם בסיס לUI שיוכל להריץ את המשחק, אותו נפתח בספרינט הבא.

# **איטרציה 4 - UI ולמידה אינטרקטיבית.**

תקציר, הסבר ותוצר:

* נבנה UI אשר דרכו יהיה  ניתן להציג את המשחק וללמד את  הבוט בצורה אינטרקטיבית/ דינמית.

תכולות טכנולוגיות:

* UI - משחק שמציג נתונים, ממשק של המשתמש שמאפשר לשחק במשחק ולצפות בבוט.
* אלגוריטם גנטי אינטרקטיבי- אפשרות לשלוט בפרמטרים המשנים את תהליך הלמידה בזמן הרצת המשחק.
* UI - תפריט בשביל המשתמש המאפשר לו לפתוח מסכים שונים בפרוייקט.
* מסכי הרשמה למשחק - ממשק של המשתמש שמאפשר לו להרשם כמשתמש הנשמר על השרת.
* מסכי התחברות למשחק - ממשק של המשתמש שמאפשר להתחבר למשתמש השמור על השרת.
* UI - פעולות על מסד הנתונים - ממשק של המשתמש הנותן לו לעדכן את מסד הנתונים.

# **איטרציה 5 - מסך התחברות , צד שרת , צד לקוח , ומסד נתונים.**

תקציר, הסבר ותוצר:

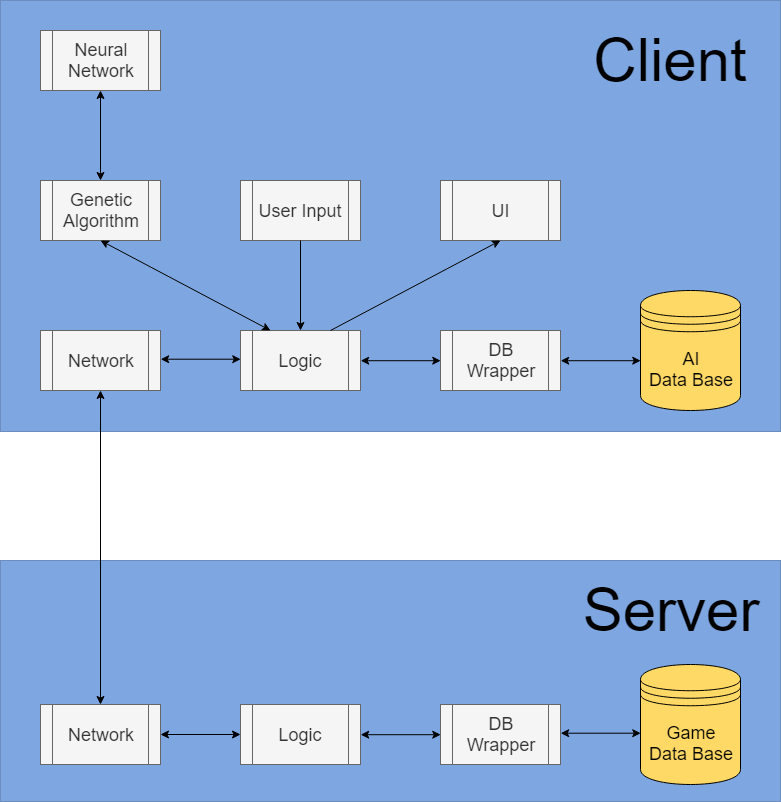
* נבנה DB שישמור את הפרטים של המשתמשים .
* צד שרת , שיקשר בין המשחק לבין מסד הנתונים
* צד לקוח , שהוא בעצם עוטף את המשחק ואחראי על העברת מידע לשרת.
* מסך התחברות שיאפשר לשחקנים להרשם או להכנס למשחק .

תכולות טכנולוגיות:

* יש מסד נתונים ששומרים בו : שם משתמש וסיסמה, מייל, וגם הניקוד הכי גבוה שצבר, וכמות הפעמים שהוא נכנס למשחק.
* נוצר שרת שהוא מקבל הודעות מהלקוח ומחלק את ההודעות האלה ומעביר את המידע למסד נתונים.
* נוצר צד שרת שהוא בעצם מעביר המידע הדרוש לשרת .
* יש עכש מסך התחברות שהוא בעצם מאפשר למשתמשים להכנס ולהרשם למשחק .

***מסמך ארכיטקטורה:***

***מבט על***



* Client - תוכנת לקוח
  + UI רכיב ממשק משתמש - מכיל את המסכים השונים של התפריט והמודים השונים של המשחק. מתקשר מול רכיב הלוגיקה אשר אחראי על להעביר אליו את הקלט של המשתמש והAI לשליטה בנחש.
  + Networking רכיב תקשורת - אחראי על ביצוע התקשורת מול השרת, כדי לאחסן תוצאות בהתאם למשתמש שמחובר. מקבל פקודה לאחסון התוצאות כאשר המשתמש מחובר בסוף כל משחק מתאים מרכיב הלוגיקה.
  + Logic רכיב לוגיקה - אחראי על חיבור, הפעלה וקריאה לרכיבים השונים: מעביר קלט מהאלגוריתם הגנטי ומהמשתמש ומעביר אותו ולממשק המשתמש, מעביר את התוצאות לתוך רכיב התקשרות ולרכיב בסיס הנתונים.
  + DB Wrapper רכיב בסיס נתונים - אחראי לשלוף, להוסיף ולעדכן נתונים מול בסיס הנתונים (ניתן להניח שהטבלאות כבר קיימות בו ונוצרו מראש). רכיב הלוגיקה מתקשר איתו על מנת לעבוד באופן מסודר, בטוח ויעיל מול מסד הנתונים של התוכנה.
  + AI Data Base מסד הנתונים - יכיל בתוכו דורות ישנים של הבוט ויתקשר עם רכיב בסיס הנתונים שיכניס ויוציא ממנו מידע.
  + User Input קלט מהמשתמש - רכיב האחראי לקבל בצורה מסודרת קלט מהמשתמש בזמן משחק או בתפריט.
  + Genetic Algorithm רכיב האלגוריתם הגנטי - בזמן משחק עם הבוט רכיב זה אחראי על לקבל את מצב המשחק ולהעביר אותו לרשת הניורונים ולהחזיר את התזוזה שנבחרה, בין המשחקים רכיב זה אחראי על לקבל את התוצאות של המשחק ולעדכן את רשת הניורונים בהתאם לתוצאות.
  + Neural Network רשת הניורונים - לפני משחק תקבל מהאלגוריתם הגנטי את הדור הנוכחי, בזמן משחק תקבל את מצב של הנחש ותשלח בחזרה (בהתאם לדור הנוכחי) תגובה מתאימה.
* Server - תוכנת שרת
  + Network רכיב תקשורת - אחראי להקשיב לבקשות חדשות מרכיב התקשורת של הלקוח ולהעביר את המידע מהם לרכיב הלוגיקה ולקבל מידע מרכיב הלוגיקה ולשלוח אותו בחזרה ללקוח.
  + Logic רכיב לוגיקה - אחראי על עיבוד הנתונים והבקשות המתקבלים בתקשורת, אימות או עדכון שלהם מול רכיב בסיס הנתונים, ומתן פלט מתאים (הכנסה למסד הנתונים או שליחת תוצאות).
  + DB Wrapper רכיב בסיס נתונים - אחראי לשלוף, להוסיף ולעדכן נתונים מול מסד הנתונים (ניתן להניח שהטבלאות כבר קיימות בו ונוצרו מראש). רכיב הלוגיקה מתקשר איתו על מנת לעבוד באופן מסודר, בטוח ויעיל מול מסד הנתונים של התוכנה.
  + Game Data Base מסד הנתונים - יכיל בתוכו משתמשים ותוצאות שלהם ויתקשר עם רכיב בסיס הנתונים שיכניס ויוציא ממנו מידע.

עיצוב הנתונים ויישויות מידע

מידע שנשמר בDB:

-משתמש: יש לכל משתמש שם משתמש וסיסמה הסיסמה היא ייחודית, וכל פעולה שמבצע אותה המשתמש בתוך המשחק נשמרת בשדה נתונים תחת השם שלו, כלומר שדה נתונים למשתמש הזה.

-שמירת גרסת הבוטים כדי לאפשר לכל שחקן לבחור בגרסה שהוא רוצה.

-  לכל שחקן יש מספר נקודות יחיודי שהוא צובר כלומר זה לא תלוי באחרים.

-יש DB שמתקשר עם המשחק, רוב הפעיליות שכל שחקן מבציע תוך כדי השימוש במשחק ישלחו לשרת ואז השרת יאחסן את זה בצורה הדרושה.

מידע שעובר בין מודולים:

* Client
  + קלט מן במשתמש יעבור לרכיב הלוגיקה.
  + קריאה לבוט תשלח מרכיב הלוגיקה לאלגוריתם הגנטי ולרשת הניורונים ותחזור לרכיב הלוגיקה.
  + כל השינויים במצב המשחק והתפריט ישלחו מרכיב הלוגיקה לUI.
  + פעולות לטעינת ושמירת גרסאות ישלחו מקלט המשתמש ללוגיקה ולרכיב בסיס הנתונים.
  + פעולות הקשורות למשתמש ישלחו מקלט המשתמש ללוגיקה ורכיב התקשורת (שבמידה ויקבל תשובה יחזיר אותה לרכיב הלוגיקה).
* Server

ממודל התקשורת יתקבלו בקשות שיעובדו במודל הלוגיקה וישלחו למסד הנתונים בעזרת מודל בסיס הנתונים. המידע שיחזור ממסד הנתונים יעובד להודעות במודל הלוגיקה וישלח בחזרה בעזרת רכיב תקשורת.

טכנולוגיות עיקריות

הנושא שהפרויקט שלנו מתבסס עלו הוא ה AI, אנחנו עדיין לא התנסינו בו וזאת הפעם הראשונה שלנו לתכנת בצורה הזו ולהשתמש ב העקרונות של ה AI.

אנחנו נשתמש בPYTHON כשפת תכנות שלנו לפרויקט הזה בסביבת העבודה PYCHARM, בגלל שבפיטון היא השפה האידיאלית לתכנת AI וגם זה מאפשר לנו להשתמש בספריות שיכולות לעזור לנו כמו tensorflow.

וגם העיבוד והקימבול של פיטון הוא יותר מהר משאר השפות ! וזה מהווה יתרון משמעותי,כי תכנות בינה מלאכותית צריך הרבה עיבוד!.

התאמה לאפיון

*עבור כל דרישה מפרק הייזום (מספיק לזהות אותה לפי מספר מזהה + תיאור קצר במספר מילים), ציינו מהם הרכיבים אשר מספקים מענה עבורה. במידה וישנם מספר רכיבים הנובעים מדרישה אחת, יש לציין את סדר הפעולה ביניהם.*

לדוגמה:

|  |  |
| --- | --- |
| **פיצ'ר** | **רכיבים רלוונטים** |
| 1. שחקן יחיד | קלט למשחק עובר דרך מודל USER INPUT ללוגיקה, שם מתעדכן המשחק והשינוי נשלח לUI. |
| 2.    בוט לומד לשחק | קלט למשחק עובר דרך מודל רשת הניורונים והאלגוריתם הגנטי ללוגיקה, שם מתעדכן המשחק והשינוי נשלח לUI. |
| 3.    לשחק נגד הבוט | שילוב של שני הפיצרים הראשונים הפועלים במקביל לתוך שני לוחות משחק שונים, רכיב הלוגיקה מחשב את הניקוד ומציג אותו בנוסף לUI. |
| 4 + 5.    הרשמה למשחק וגישה לנתוני המשתמש | דרך קלט המשתמש נשלחות ללוגיקה בקשות למסד הנתונים בשרת, הלוגיקה בונה את הבקשה ושולחת אותה בעזרת רכיב התקשורת לשרת. בשרת רכיב התקשורת מקבל את ההודעה, מעבד אותה ומעביר אות לרכיב בסיס הנתונים שניגש למסד הנתונים. מידע חוזר בחזרה דרך אותם המודלים אל הUI. |
| 6 + 7.    גישה למסד הנתונים המקומי | דרך קלט המשתמש נשלחות ללוגיקה בקשות לטעינה ושמירה של מידע, הלוגיקה מעבירה את הבקשה לרכיב מסד הנתונים ומחזירה את התגובה לUI. |

## מחלקות\מתודות

* מסד נתונים – בניתי מסד נתונים באמצעות SQLITE
* מחלקת DB - כתבתי מחלקה שמקשרת בין המשחק(שרת) למסד נתונים.
* שרת - מסדר את ההודעות שיוצאות מהלקוח ומטפל בהם כשתצריך.
* DetailsPage.py – דף השער של הלקוח .

\*\*עבודה משותפת(100%)\*\*:

* GeneticAlgorithm.py – אלגוריתם גנטי.
* NeuralNetwork.py – רשת נוירונים.

## מסמך מפרט התוכנה  Detailed Design

הפרויקט הוא בעצם שילוב טכנולוגיית הבניה מלאכותית בתוך משחק ה"נחש". שזה מאפשר לנו לבנות בוט (מכונה) הלומדת לשחק המשחק בעצמה (עם אפשרות להתקדם ולהתפתח בהמשך) בנוסף המשחק מאפשר למשתמש לשחק במשחק בעצמו .וגם מאפשר למשתמש לאמן בוטים שונים משלו וגם להשתמש בבוטים אחרים .

אנחנו משתמשים בPYTHON3 מכיוון שזאת שפה מאוד חזקה בתחום הזה כי העיבוד שלה הוא יותר מהר משפות אחרות.

התחלנו את העבודה בסביבת הPYCHARM, העבודה בסביבה הזאת הייתה קצת קשה מבחינת טעינת הקבצים (פתיחת קבצי הקוד לתוך הסביבה) , אחר כך החלפנו לסביבת ה visual code שזה בעצם דומה מאוד ל visual studio אבל מתאים לpython.

המודולים העיקריים של הפרויקט:

-הלקוח של המשחק.

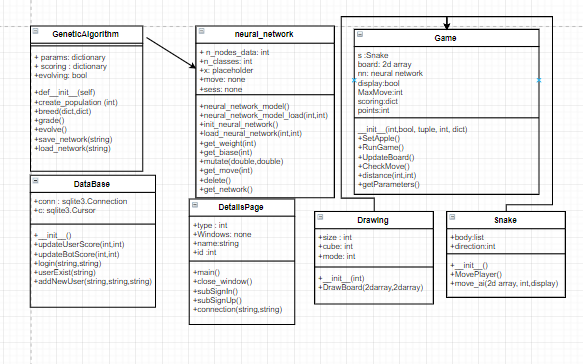
-התפריט שמפעיל את הפיצרים השונים.

-אלגוריתם שרץ ברקע בזמן שימוש בבוט המחשב את מהלכיו (רשת נוירונים).

-אלגוריתם שמקבל את תוצאות המשחק של הבוט ומנסה לשפר אותם (אלגוריתם גנטי).

-פונקציה של טעינה ושמירה של דורות מתקדמים של הבוט.

תקשורת צד לקוח שרת היא לוקאלית כלומר הכל קורה בטווח האינטרנטית של המחשב עצמו כלומר המחש הוא השרת וגם הלקוח.



פרוטוקול תקשורת:

מטרת פרוטוקול תקשורת הוא לסדר את תוכן ההודעות שעוברות בין לקוח לשרת וההפך וזה עוזר לקבל את ההודעה ולבציע את הפעולה המתאימה.

סוג ההודעה שאנחנו שולחים הוא טקסט אבל אנחנו הופכים אותה לבינארית כדי לאפשר העברת ההודעה .

מבנה ההודעה - code, username len ,username, pass len , pass

#example - 0104dodo04pass7785

Code: 01 ,username length:04 ,username:dodo ,pass length:04 , pass:7785

מסד נתונים:

ממשתי את מסד הנתונים באמצעות פייטון וגם sqlite3 .

פקודות SQL שהשתמשתי בהם , דוגמאות:

1. ('SELECT SingleScore FROM info WHERE id =' + str(id))
2. ('UPDATE info SET SingleScore = ' + str(newScore) + ' WHERE id =' + str(id))
3. ('SELECT \* FROM info WHERE name ="' + name +'" and password ="' + pas +'"')
4. ('INSERT INTO info VALUES(NULL,"' + name + '","'+ pas + '","' + email + '",0,0,0)')

פעולות הן פשוטות מאוד.

## מסמך הבדיקות

## בפרויקט שלנו הבוטים שנוצרו הם חלק עיקרי בבדיקות כי באמצעותם אנחנו בודקים את איכות הקוד שכתבנו והתפתחותם , כלומר בודקים את רשת הנוירונים וגם האלגוריתם הגנטי.

## בדיקת תקשורת לקוח שרת: בדיקת השרת והלקוח אם עובדים היטב וגם אם יכולים לתקשר עם כתובית מקומית localhost.

## בדיקת קלט מהמשתמש: כאשר שם משתמש מכניס USERNAME או PASSWORD, עשינו בדיקה מול מסד נתונים.

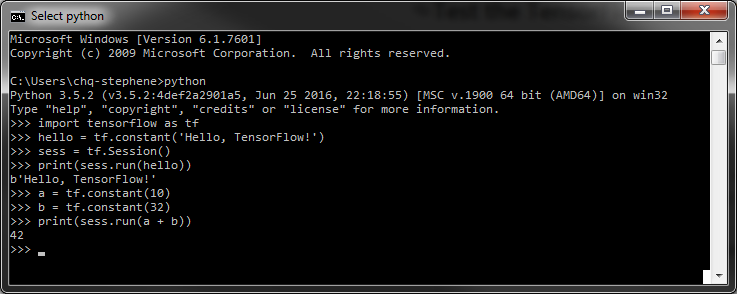
## בדיקת שם משתמש: הבדיקה הייתה שנכניס שם משתמש ונבדוק אם המסד נתונים יכול לזהות אם הוא קיים או לא.

## בדיקת סיסמה: בדיקת המסד נתונים אם הוא יכול לקבל את הסיסמה ויתאים אותה לשםמשתמש.

## הוראות שימוש – התקנה והפעלה

***1. מדריך התקנה:***

דרך התקנת **tensorflow**:

1. Install the Python development environment on your system
2. To install TensorFlow, start a terminal. Make sure that you run the cmd as an administrator
3. Just enter this command: C:\> pip3 install --upgrade tensorflow
4. Testing your TensorFlow:
5. 

התקנת **PYGAMES** :

1. Make sure you install python3.6 with the "Add python 3.6 to PATH" option selected. This means that python, and pip will work for you from the command line.
2. Just enter this command: C:\> py -m pip install -U pygame –user

**2. התאמת סביבת העבודה להרצה – קונפיגורציה**:

יש לוודא שכל הקבצים נמצאים בשתי התיקיות Serverו Client.

בתיקה **Server** :

* DB.py
* Server.py
* . SnakeDB.db

בתיקה **Client**:

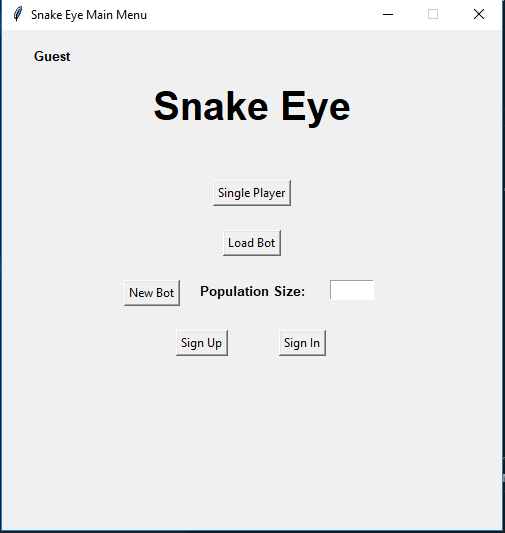
* detailsPage.py
* GeneticAlgorithm.py
* NeuralNetwork.py
* UIBot.py
* UIGame.py
* UIMenu.py
* Saves
* \_\_pycache\_\_

מסד נתונים : הוא מחובר באמצעות sqlite3 . שם משתמש פעיל :Man סיסמה:123

תקשורת צד לקוח שרת : התקשורת היא LOCAL HOST .

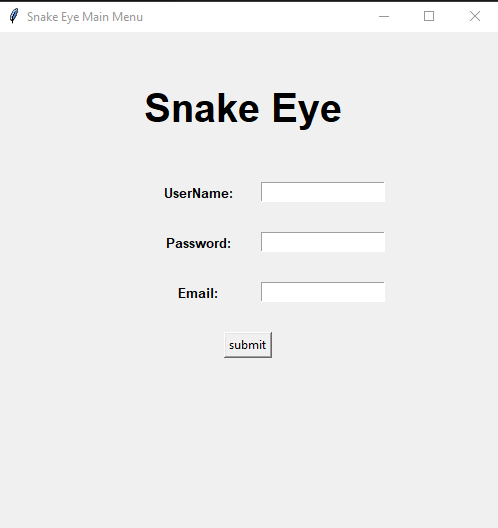
**3. מדריך האפליקציה למשתמש User Guide:**

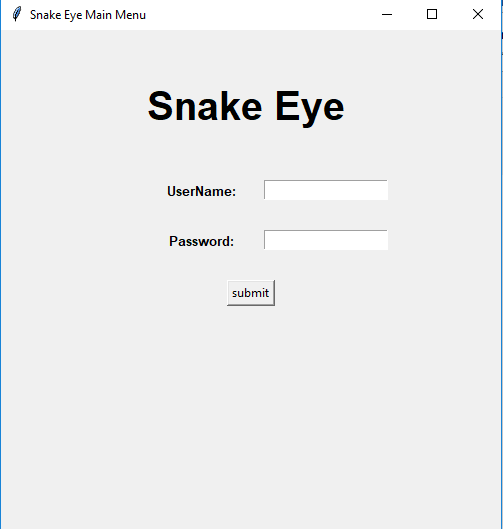
1. צריך להפעיל את השרת – לך לתיקיה בשם Server תפעיל את קובץ server.py באמצעות הCMD.
2. לך לתיקיה בשם client , תפעיל את התיקיה UIGame.py באמצעות visual studio.

אחרי שני השלבים את/ה אמור/ה לראות :

**Single Player** : מאשפרת למשתמש לשחק במשחק כרגיל.

**Load bot :** מאפשר טעינת גרסאות שונות של בוטים שכבר אומנו לשחק את המשחק בעצמם וזה גם מאפשר לך צפייה בו כשהוא משחק .

**NewBot**: מאפשר למשתמש לאמן את הבוט החדש שלו.

**Sign in & Sign up** : מאפשר הרשמה וכניסה למערכת.

## משוב

בתחילת הפרויקט לא היה לנו שום ידע קודם על הנושא של בניה מלאכותית, ולכן התחלנו מללמוד את הנושא הזה כדי שיהיה לנו רעיון איך להתחיל לכתוב קוד שבעזרתו נממש את מטרת הפרויקט שלנו.

קראנו המון מאמרים שמצאנו באינטרנטית וגם צפינו בכמה סרטונים שונים שמסבירים את הנושא .

אחרי שלמדנו את החומר הדרוש, התחלנו מלבנות משחק הסנייק בעזרת PYGAME ואחר כך התחלנו לכותב רשת נוירונים שמקבלת קלטיים שונים ומעבדת אותם.

שלב אחרי זה כתבנו את האלגוריתם הגנטי שהוא משפר את העיבוד של רשת הנוירונים .

כשהכול היה עובד כמו שצריך בנינו את הגוי וגם את מסד הנתונים וגם שרת , וקישרנו בניהם .

מטרת הפרויקט היא ללמוד את הנושא החדש - AI וכן בסוף הגעתי למצב שבו אני מבין איך מכונה עבודת ואת המונחים של התחום וגם הכי עיקרי , איך לממש את הטכנולוגיה הזאת כדי לפטור בעיות שונות.

העבודה על הפרויקט הזה לימדה אותי איך לנהל את זהמן שלי בצורה מאוד מסודרת, וגם הוסיפה לי את יכולת הלמידה העצמית שבעזרתה אני למדתי את החמור החדש הזה בכוחות עצמי.

הפרויקט הזה בניתי עם שותף, שזה גם מחזק לי את הצד החברתי הצורה טובה מאוד.

## קוד הפרויקט

## יש שתי תיקיות שהן server , client:

בתיקה **Server** :

* DB.py
* Server.py
* . SnakeDB.db

בתיקה **Client**:

* detailsPage.py
* GeneticAlgorithm.py
* NeuralNetwork.py
* UIBot.py
* UIGame.py
* UIMenu.py
* Saves
* \_\_pycache\_\_

## התייחסות אישית

בחלק זה יש להתייחס לשאלות הבאות באופן אישי (לא כצוות).

1. איך היתה עבודת הצוות? מה היו החלקים הטובים בעבודה בצוות? מה היו החלקים הקשים? תארו את אופן העבודה שלכם בצוות.

**תשובה**: בהתחלה עבודת הצוות הייתה קצת מסובכת כי כל אחד מאתנו למד את החומר החדש לבד, וזה נמשך כחודש , אחר כך התחלנו לממש את החלק הראשון מהפרויקט ביחד וזה היה קצת קשה כי קודם כל הינו צריכים לעבוד כל אחד מהבית שלו אבל באותו זמן וזה התנהל בעזרת SKPYE או DISCORD.

כשאחד נתקל בבעיה , אז השני היה עוזר לו לפטור אותה וזה היה יתרון משמעותי במהלך העבודה .

לכל אחד מהצצות היה לו ידע שונה בדברים שונים לכן זה עזר לנו מאוד ללמוד אחד מהשני .

1. באילו אתגרים נתקלתם במהלך העבודה? איך התמודדתם איתם? כיצד האתגרים השפיעו על פיתוח הפרויקט?

**תשובה**: האתגר הראשון והעקרי שהיה לנו הוא ללמוד את הנושא החדש , זה היה קשה מאוד , אבל בסוף התמודדנו עם זה בעזרת למידה ממקורות שונות.

היו לנו כמה בעיות במהלך כתיבת רשת הנוירונים וזה התברר כי לא ידענו איך להשתמש בTENSRFLOW לכן חקרנו ולמדנו איך להשתמש בה.

כשהתקדמנו בפרויקט, שינינו כמה פעמים את אלגורטים הגנטי כדי שנוכל לשפר את הפצועים ולראות מה הכי מתאים .

1. מהו הפער בין הפרויקט כפי שתכננתם אותו מראש לבין מה שבוצע בפועל? כיצד אתם מסבירים את הפער הזה?

**תשובה**: הפרויקט הלך לנו בדיוק כמו שתכננו .

1. האם ישנו רכיב, אלגוריתם או טכנולוגיה שעליהם למדתם ולא הכרתם קודם לכן? ספרו על חלק שכזה, שנהנתם לעבוד עליו, או שהוא מגניב במיוחד.

**תשובה**: טכנולוגית בניה מלאכותית, למידת הנושא היה קשה אבל בסוף זה היה שווה כי הגענו למצב שיש לנו בוט (מכונה) שמשחק את המחשק בעצמו והוא שורד לזמן טוב יחסית.

1. מה למדתם מהעבודה על הפרויקט?

**תשובה:** העבודה על הפרויקט הזה לימדה אותי איך לנהל את זהמן שלי בצורה מאוד מסודרת, וגם הוסיפה לי את יכולת הלמידה העצמית שבעזרתה אני למדתי את החמור החדש הזה בכוחות עצמי.הפרויקט הזה בניתי עם שותף, שזה גם חיזק לי את הצד החברתי בצורה טובה מאוד.

1. האם ישנו פתרון אפשרי אחר?

**תשובה:** יש פתרון שונה אבל דומה !! כן זה אפשרי כי לממש את המטרה שלנו שהיא בעצם שיצא לך מוצר שהוא הבוט(מכונה) שלומדת את המשחק ומשחק בו בעצמה , אתה חייב להשתמש בטכנולוגית הAI בנייה מלאכותית אבל את יכול להשתמש בה בצורה אחרת וזה בעצם מהווה פתרון אחר אפשרי.

למשל: כל שינוי ברשת נוירונים או באלגוריתמים גנטי זה מהווה פתרון אחר.

## רשימת מקורות (ביבליוגרפיה)

## 08/24/2018

<https://www.youtube.com/watch?v=cKxRvEZd3Mw&list=LLZ5pVSeXnfnGhBzo18ZkHpg&t=0s&index=4>

## 11/22/2018

pip3 install tensorflow

<https://www.analyticsvidhya.com/blog/2016/10/an-introduction-to-implementing-neural-networks-using-tensorflow/>

## 11/22/2018 - tensorflow-neural-network-tutorial

<https://stackabuse.com/tensorflow-neural-network-tutorial/>

## 11/22/2018 - Beginner Intro to Neural Networks 1: Data and Graphing

<https://www.youtube.com/watch?v=ZzWaow1Rvho&list=PLxt59R_fWVzT9bDxA76AHm3ig0Gg9S3So>

## 01/03/2019 - Let’s evolve a neural network with a genetic algorithm

<https://blog.coast.ai/lets-evolve-a-neural-network-with-a-genetic-algorithm-code-included-8809bece164>

## 01/16/2019 - How to save and load a neural network in TensorFlow

<https://lazyprogrammer.me/how-to-save-and-load-a-neural-network-in-tensorflow-deep-learning-tips/>

## 02/21/2019 - Basic Python client socket example

<https://stackoverflow.com/questions/7749341/basic-python-client-socket-example>