תיק פרויקט - עידן משיח

**הקדמה ותודות:**

תחילה הייני רוצה להודות למוריי אלה וגיא שעזרו לנו במהלך בניית הפרויקט. זהו היה פרויקט שנמשך כמעט שנה שלמה ונדרשנו להשקיע בו שעות רבות. אצלי בחירת המשחק לקחה זמן רב עד שמצאתי את המשחק אותו אני מגיש לבגרות ולולא עזרתם של גיא ואלה ספק אם הייתי מוצא אותו. בנוסף לעזרה על בחירת המשחק הם גיא ואלה עזרו לנו בכתיבת קודים נכונים ותיקון בעיות. אלה אף הביאה את בניה אשר למדו פרולוג על מנת לעזור לנו בפרויקט.

בנוסף לתודות למורים הנני רוצה להגיד תודה גם לחבריי לכיתה אשר יש להם חלק לא קטן בפרויקט שלי.

**תוכן עניינים:**

פרק ראשון עמודים : 2-3

פרק שני עמודים: 4-8

פרק שלישי עמודים: 9-10

פרק רביעי עמודים: 11-14

פרק חמישי עמודים: 15-20

פרק שישי עמודים: 21-24

פרק שביעי עמודים: 25-26

פרק שמיני עמודים: 27

פרק תשיעי עמודים: 28

**פרק 1 - הצעת עבודה**

**נושא העבודה:** משחק לוח בשילוב בינה מלאכותית

**שם מגיש העבודה:** עידן משיח

**שם המורה המנחה:** אלה לב

**שנת הגשת העבודה:** 2015, תשע"ה

**מומחה אנושי:** חקרתי לבדי את המשחק, חוקיו ואסטרטגיות אפשריות והפכתי למומחה האנושי.

**קהל היעד:** המערכת מיועדת לכל אדם המעוניין לשחק במשחק Snort אשר צורת הלוח בו היא שונה (משושה) ולכן גם כמה מהחוקים, אפשרי לשחק בכל גיל.

**מבוא:**

המשחק מתאים לשני שחקנים: השחקנים יכולים להיות אנושיים, אחד מול השני או משחק של שחקן אנושי מול המחשב ברמות קושי שונות.

אין תזוזה על הלוח אלא הצבה של הכלים.

**מטרות:** המערכת הממוחשבת תהיה בעלת בינה שתוכל לנצח מומחה אנושי. יתבצעו מהלכים אסטרטגים ומחושבים.המערכת המומחית תהיה חלק ממשחק שישמש הנאה למשתמשים בו.

**שאלות עיקריות עליהן המערכת עונה:** המערכת המומחית תאפשר לשחקן לשחק במשחק נגד שחקן אחר, נגד שחקן מחשב. תהיינה אפשרויות של יציאה , עצה.

**אילוצים ודרישות:** 🖫 המערכת הממוחשבת תציע למשתמש מגוון אפשרויות (כמו בתפריטים השונים) ולא תגיע להחלטות בעצמה, חוץ ממהלכי השחקן הממוחשב.

🖫 שפת הכתיבה: פרולוג, השתמשתי בתוכנת Prolog Amzi.

🖫 לוח זמנים :

בחירת המשחק עד למצב עבודה תקין של "אנושי מול אנושי" - ספטמבר 08' עד ינואר 09'

רמת בינה 0 גרפיקה בסיסית, כולל תפריטים - ינואר 09' עד פברואר 09'

המשך עבודה כולל שיפור גרפיקה ותכנות רמות בינה 1,2,3 - פברואר 09' עד אפריל 09'

הוספת רמת בינה 4 בשילוב MiniMax וגראפיקה סופית - אפריל 09' עד מאי 09' (הגשה סופית)

🖫 ידידותיות למשתמש: במערכת מגוון תפריטים נוחים וידידותיים בשפה פשוטה. במידת הצורך, מתקבלות הודעות שגיאה ומתבצעות בדיקות תקינות למשתמש האנושי. המשחק מעוצב בצורה גרפית יפה ונעימה לעין.

🖫 המערכת מסוגלת לתת עצה/המלצה : המהלך הכי טוב שניתן לבצע באותו מצב ספציפי.

**פרק 2 - תיאור תחום הידע**

**שיטות לשאיבת הידע**

לאחר שבחרתי את המשחק Snort, שיחקתי בו מספר רב של פעמים. המטרה הייתה להבין מספר דברים חשובים:

🖫 מטרת המשחק

🖫 מהם המהלכים האפשריים שניתן לבצע בכל תור

🖫 מהו ניצחון במשחק

🖫 מהו המהלך הכי טוב שניתן לעשות בכל שלב במשחק, ולמה

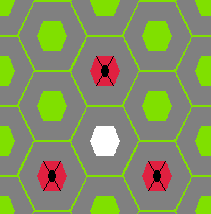
🖫 רעיונות לשדרוג המשחק הקיים - גרפיקה ושיפורים אפשריים נוספים, אך בלי לפגוע בחוקי המשחק, אופן ביצוע המהלכים וכו'.

**סיכום תחום הידע**

מהלך חוקי במשחק:

מהלך חוקי במשחק Snort הוא מהלך אפשרי שניתן לעשות בכל שלבי המשחק.

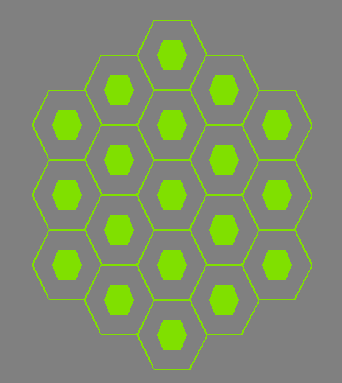
כל שחקן אשר מציבים בלוח יוצר משבצות אסורות אשר נמצאות בכיוונים 8 , 1 , 3 (אם 5 היא המשבצת בה הצבנו) אשר השחקן השני אינו יכול להציב בהם.



זוהי דוגמא בה ניתן לראות שחקן לבן ואת המשבצות שהוא משפיע עליהם וגורם להם להפוך לבלתי חוקיות עבור השחור.

**מצבים שונים של הלוח:**

**לוח התחלתי לדוגמה (משושה של שלוש):**

****

כל הלוחות הם בצורת משושה אשר אין בו כלים כלל.

ישנם 3 גדלים שונים ללוח המשחק:

לוח משושה בגודל 3

לוח משושה בגודל 4

לוח משושה בגודל 5

**ב. מטרת המשחק**

ניצחון במשחק מתקיים כאשר לא נותרים כלים אשר השחקן היריב יכול להציב על הלוח, כלומר המטרה היא להשתלט על כמה שיותר משבצות על הלוח.

**ג. מהלכים אפשריים**

המהלך האפשרי (ללא הכפתורים) היחיד המותר במשחק הינו הצבה של שחקן שלך על הלוח.

מהלכים אפשריים במשחק:

ניתן לבצע מהלכים לצורך הגעה לניצחון במשחק.

בנוסף, יש עוד שלל מהלכים אפשריים שניתן לבצע במהלך המשחק, כמו לחיצה על כפתורים שונים.

❖ שמירה –

save

ניתן ללחוץ על כפתור השמירה בכל שלב במהלך המשחק , אשר יגרום לשמירת המצב הנוכחי.

❖ עצה -

advice

כפתור העצה יאפשר למשתמש לקבל המלצה "חכמה" - המהלך הכי טוב שניתן לעשות על אותו הלוח כדי להגיע לניצחון.

בעת לחיצה על כפתור העצה, יוצג על הלוח המהלך המומלץ.

לאחר מכן יהיה ניתן לבחור האם לקבל את עצת המחשב או לבצע מהלך על פי בחירתו של השחקן.

❖ הוראות המשחק –

ora

ניתן ללחוץ על כפתור הוראות המשחק בתפריט הראשי.

לחיצה על כפתור זה יציג בפני המשתמש את הוראות המשחק. בסיום קריאת ההוראות יהיה ניתן ללחוץ על כפתור הenter וכך הוראות המשחק ייעלמו מהמסך.

❖ יציאה –

exit

בעזרת כפתור היציאה ניתן לצאת מהמשחק.

❖ חזור –

back

על לחיצת הכפתור חזור המצב יוחזר למצב הקודם שלו לפני ביצוע המהלך האחרון.

❖ משחק חדש –

NewGame

על לחיצת הכפתור משחק חדש המצב יוחזר למצב ההתחלתי שנקבע על ידי המשתמש.

❖ טען משחק –

reload

על לחיצת הכפתור טען משחק המצב שנשמר על ידי כפתור השמירה יטען למצב הנוכחי.

**פרק 3 - מבנה המערכת המומחית**

מערכת מומחית מורכבת מארבעה חלקים עיקריים :

1. בסיס הידע - KNOWLEDGE BASE

2. מערכת היסק - INFERENCE ENGINE

3. מערכת הסבר – EXPLANATORY INTERFACE

4. ממשק למשתמש – USER INTERFACE

**היררכית ההשתייכות של תוכניות מחשב מתחום הבינה המלאכותית:**

תכניות בינה מלאכותית

מערכות מבוססות ידע

מערכות מומחות

משלבות התנהגות אינטילגנטית בעזרת שיטות היוריסטיות

בסיס הידע ניפרד משאר המערכת

מוסיפות ידע מומחי כדי לפתור בעיות מן העולם האמיתי

### הסבר למושגים בסיסים במערכות מומחות :

**בסיס ידע –** בסיס הידע מתייחס לעולם התוכן המומחי של המערכת.

הוא מתאר את המרכיבים של תחום הידע, ואת הקשרים הקיימים בין המרכיבים השונים.   
בדרך כלל מתואר בסיס הידע באמצעות נתוני יסוד – עובדות, וכללים המתארים את הקשרים ביניהם ,קרי, חוקים. (במקרה של פרולוג)

העובדות מתארות את תחום הידע והחוקים את היחסים בין פרטים או מצבים בתחום הידע.

**מערכת היסק** – מערכת ההיסק מורכבת ממנגנון המסוגל להסיק מסקנות בהסתמך על בסיס הידע ומנגנון ההיסק קובע כיצד לנתח את הידע המתקבל.

המערכת מתייחסת אל בסיס הידע כאל אוסף של טענות הכוללות עובדות וחוקים בתחום הידע.  
המנגנון מקשר בין בסיס הידע כדי להגיע למסקנה.

תיאור סכמטי של המערכת המומחית וההסבר

בסיס הידע KNOWLEDGE BASE

משתמש - USER

ממשק למשתמש USER INTERFACE

מעטפת (SHELL)

מערכת הסבר INTERFACE EXPLANATORY

מערכת היסק INFERENCE ENGINE

**מדוע המערכת שלי הינה מערכת מומחית?**

המערכת המומחית שבניתי, מכילה את כל המאפיינים של המערכת המומחית הנזכרים לעיל, החוקים והעובדות אותם בניתי מהווים את בסיס הידע הדרוש להרצת הפרויקט לפי הגדרותיו, מסך המשחק, הכפתורים ומסך ההסבר משמשים כממשק למשתמש, מערכות ההיסק וההסבר אשר משולבים במהלכי המחשב ובהמלצת המהלך לשחקן מסבירה מדוע התוכנית פעלה בדרך שפעלה. מערכות אלו מהוות "מטא אינטרפטר" שהוא מרכיב מפתח במערכות מומחות.  
המערכת שבניתי מכילה את כל מרכיבי המערכת הדרושים והמצויינים לעיל, ולכן היא מערכת מומחית.

**פרק 4**

**4.1 – בחירת שיטה לייצוג מידע**

מצב(\_תור\_מי,\_לוח)

אצלי המצב מורכב משני פרמטרים , \_תור\_מי אשר מציין לי התור של מי עכשיו , כלומר מי מבצע עכשיו את המהלך , ו - \_לוח שזהו רשימת כלים שבה כל כלי נמצא על הלוח.

**4.2 – תרשים לולאת המשחק**

בחירת אפשרויות משחק

הדפסת לוח

האם קיימים מהלכים אפשריים

כן לא

השחקן שתורו בוחר היכן לשים כלי

המחשב בודק תקינות המקום המבוקש

תקין לא תקין

מתבצע המהלך – הכלי מוצב במקומו הודעה מתאימה

התור מוחלף

סוף משחק?

כן לא

מי ניצח – הודעה מתאימה

סגור חלון

הלולאה בפרולוג עובדת על ידי הכשל , השפה מבצעת את כל השלבים על ובסוף כאשר היא בודקת אם סוף המשחק (בתנאי שאין סוף משחק ) היא נכשלת וחוזרת חזרה למעלה ל - 'חזור' וככה היא ממשיכה עד שהיא מצליחה ואז היא סוגרת את המשחק.

**4.3**

משחק:-

%% 1. האתחל משחק בעצם הוא זה שמנקה את כל מה שהיה בזיכרון לפני זה ויוצר מצב %% חדש , בנוסף האתחל משחק גם פותח לי חלון ובו הוא בונה את הלוח והכפתורים , %% ואז מגיעים למצב שאפשר לשחק

אתחל\_משחק,

%% הלולאת חזור היא זו שממשיכה את המשחק עד שהסוף משחק מצליח

חזור,

%% 2. המצב הנוכחי לוקח את המצב הנוכחי והופך אותו לנגיש לשאר החוקים בתור %%... משתנה \_מצב

מצב\_נוכחי(\_מצב),

%% 3. הקבל מהלך מחכה לקבל מהלך מהפונקציה הדינמית (הפונקציה הדינמית בעצם %% ממירה את הלחיצות על הלוח למהלכים) וברגע שהוא מקבל אותו הוא הוא בודק %% האם הוא חוקי , אם הוא חוקי הוא מעביר אותו הלאה ואם לא הוא דורש מהלך חדש

קבל\_מהלך(\_מצב,\_מהלך),

%% 4. הבצע מהלך לוקח את המהלך מהקבל מהלך ומבצע אותו , כלומר משנה את המצב %% למצב לאחר ביצוע המהלך

בצע\_מהלך(\_מצב, \_מהלך, \_מצב\_חדש),

%% 5. שמירת המצב הקודם נועדה לכפתור חזור והעדכן מצב נועד על מנת להפוך את %% המצב לאחר ביצוע המהלך למצב הנוכחי שעליו אנו עובדים

שמירת\_מצב\_קודם(\_מצב),

עדכן\_מצב(\_מצב\_חדש),

%% 6. ההדפס מצב והעדכן גרפי נועדו להציג למשתמש את המצב לאחר ביצוע המהלך

הדפס\_מצב(\_מצב\_חדש),

עדכן\_גרפי(\_מצב\_חדש),

%% 7. בדוק סוף משחק נועד לבדוק אם המשחק יסתיים בין אם זה על ידי כך שמישהו %% מנצח או על ידי כך שנלחץ כפתור היציאה , אם המשחק מסתיים הלולאת משחק %% עוברת לסור משחק אבל אם זה לא סוף המשחק הבדוק סוף משחק נכשל ואז %%.. הלולאת משחק חוזרת ל'חזור'

בדוק\_סוף\_משחק(\_מצב\_חדש, \_מנצח),!,

%% 8. הסגור משחק סוגר את המשחק ומציג הודעה בהתאם

סגור\_משחק(\_מנצח).

**4.4 – שלבי פיתוח המשחק**

טקסטואלי לשני שחקנים אנושיים

גרפי לשני שחקנים אנושיים

גרפי לשחקן אנושי וגרפי לשחקן ממוחשב בעל רמת בינות

הוספת מינמקס לבינות

**4.5 – רמות הבינה השונות**

בינה 0 : זוהי לא בדיוק בינה , המחשב מבצע מהלכים אקראיים

בינה 1 : זוהי בינה שבה המחשב מבצע מהלכים אקראיים אך אם הוא רואה סוף משחק הוא מבצע אותו

בינה 2 : זוהי בינה שעובדת על מספר המקומות שתופס השחקן על הלוח , ככל שהשחקן יתפוס יותר מקום כך הציון שלו יהיה יותר גבוה

בינה 3 : זוהי הבינה הכי חכמה שהיא לא מינימקס , בעצם הבינה לוקחת המהלך שיגרום לכך שיהיה כמה שפחות מהלכים ליריב (אצלי מהלך זה בעצם תפיסת משבצת).

בינה מינימקס : זוהי בעצם הבינה הכי חכמה , הבינה לוקחת את בינה 3 ומחפשת איתה לעומק את המהלך הכי טוב .

**4.6 – התוספת שלי בפרוייקט**

הדבר הראשון שהוספתי בעצמי היה האלגוריתם ללוח משושה שיצרתי בעצמי שזוהי סדרה של כמה חוקים היוצרים משבצות חוקיות ולא חוקיות ככה שיווצר לוח משושה עם הכיוונים שרציתי.

בנוסף יצרתי את מערך החוקים של איך מציבים את הכלי (איזה משבצות לא חוקיות הוא יוצר) ואיך הוא נראה על הלוח.

בנוסף לחוקים אני יצרתי את התמונות לבד של הכלים והלוח.

**פרק חמישי – תיאור מנגנון ההיסק.**

הסבר כללי:

מערכת ההיסק היא מנגנון המסוגל להסיק מסקנות ע"פ הידע הקיים בבסיס הידע. תפקידה, בדומה למומחה אנושי לנתח את הידע הקיים ולהגיע בעזרתו למסקנות הנכונות.

על מערכת ההיסק להשתמש בחוקים ובעובדות הקשורות ליעדה על מנת להגיע למסקנה. מערכת ההיסק בדומה למערכת ההסבר אינה תלויה בבסיס הידע ולכן שייכת למעטפת הכללית של המערכת המומחית. ישנם שינויים קלים אשר יכולים להתאים מערכות היסק שונות בצורה טובה יותר למומחיות מסוימת, אך באופן כללי, המערכת אמורה לפעול על כל בסיס ידע.

מהי מערכת ההיסק ומטא אינטרפטר?

מערכת ההיסק היא למעשה מטא אינטרפטר,דהיינו תכנית העוברת ועוקבת אחר תוכנית אחרת, מנגנון מפרש. המיוחד במנגנון זה הוא עצם כתיבת המפרש באותה שפה בה כתובה התכנית עצמה. המערכת פועלת ע"פ שיטת **Backward Chaining**, כאשר בשיטה זו ההיסק מתחיל מן המטרה עצמה ומסתיים בעובדות הקיימות בבסיס הידע. מנגנון ההיסק משתמש בחוקים על מנת למצוא עובדות המוכיחות את הצלחת המטרה.

המטרה:

מטרת מנגנון ההיסק היא להוכיח יחסים בין חוקים ועובדות על ידי חיקוי הפעולות של התכנית בפרולוג, ולבסוף להציגן למשתמש.

מרכיבי מערכת ההיסק:

המערכת מבוססת על החוק **הוכחה/2** המשתמש בחוק **פסוק/2.**

הסבר לחוק פסוק/2:

המתאר **פסוק( \_ראש\_החוק, \_גוף\_החוק )** מקבל ראש חוק ומחזיר את גופו.

לדוגמא אם קיים החוק:

**ניסיון(\_ניסוי,\_ניסוי2):-**

**\_ניסוי<5,**

**\_ניסוי2>40.**

עבור הפעלת החוק  **פסוק(ניסיון, \_ביצוע\_ניסוי)** נקבל

\_ביצוע\_ניסוי = ( \_ניסוי<5, \_ניסוי2>40).

הסבר החוק **הוכחה/2**

**הוכחה מטרה, הסבר( \_מטרה, \_גוף, \_הסבר\_גוף ) ):-**

**פסוק( \_מטרה, \_גוף ),**

**המשך\_הוכחה( \_גוף, \_הסבר\_גוף ).**

החוק **הוכחה/2** יקבל את המטרה לה אנו רוצים לקבל הסבר, יפרק אותה **לראש וגוף** בעזרת המתאר **פסוק/2**, לאחר מכן ימשיך את החוק, את ההוכחה על הגוף ויחזיר הסבר בהתאם, הסבר זה יכנס לתוך רשימת ההסבר שתתורגם לשפה טבעית דרך מנגנון ההסבר.

טיפול בעובדה:

**המשך\_ההוכחה( true, [] ):-!**

במקרה שהוחזר True סימן שההוכחה הגיעה לעובדה.

טיפול במטרה חסומה:

**המשך\_הוכחה(\_מטרה\_בודדת,[]):-**

**מטרה\_חסומה(\_מטרה\_בודדת),!,**

**\_מטרה בודדת.**

חוק זה מטפל במקרה שישנו חלק בתוכנית **שאין אנו רוצים להציג את ההסבר שלו**.

החוק יחזיק רשימה ריקה. שרשרת ההיסק תסתיים אבל המטרה תתבצע.

דוגמא:

מטרה\_חסומה(!).

מטרה\_חסומה(אקראי(\_)).

טיפול בתנאי לוגי,בביטויים חשבוניים ובמתארי רשימות:

**הוכחה( \_מטרה, הסבר(\_מטרה, \_מסוג, \_ההסבר ) ):-**

פסוק\_מסוג(\_מטרה, \_מסוג, \_הסבר),!,

\_מטרה.

חוק זה מטופל במתארי רשימות, ביטויים חשבוניים וביטויים לוגיים אשר מוגדרים עם הסבר מראש ואשר אין צורך לכלול את הסברם ברשימת ההסבר. סימן ה- ! מונע מן החוק למצוא את המטרה בדרך אחרת.

המתאר **פסוק\_מסוג/3** יכיל את כל הביטויים הלוגיים החשבוניים ומתארי הרשימה והסברם.

דוגמאות:

**טיפול בביטוי חשבוני.**

**פסוק\_מסוג(\_תוצאה הוא \_חישוב, חשבוני, ['מחישוב מקבלים ש',\_תוצאה1,' הוא ',\_חישוב1]):-**

**\_תוצאה הוא \_חישוב,**

זו התוספת שלי להוכחה.

**תרגם\_מספר\_לערך\_תווי(\_תוצאה,\_תוצאה1),**

**תרגם\_ביטוי\_לערך\_תווי(\_חישוב,\_חישוב1).**

ביתר פירוט:

**תרגם\_ביטוי\_לערך\_תווי(\_חישוב,\_חישוב1):-**

מספר(\_חישוב),

תרגם\_מספר\_לערך\_תווי(\_חישוב,\_חישוב1).

**תרגם\_ביטוי\_לערך\_תווי(\_חישוב,\_חישוב1):-**

פנקטור(\_חישוב,\_פעולה,2),

תרגם\_פעולה\_לערך\_תווי(\_פעולה,\_פעולה1),

ארגומנט(1,\_חישוב,\_חלק\_ראשון),

תרגם\_ביטוי\_לערך\_תווי(\_חלק\_ראשון,\_חלק\_ראשון1),

ארגומנט(2,\_חישוב,\_חלק\_שני),

תרגם\_ביטוי\_לערך\_תווי(\_חלק\_שני,\_חלק\_שני1),

שרשר\_מחרוזות(['( ',\_חלק\_ראשון1, \_פעולה1 ,\_חלק\_שני1,' )'],\_חישוב1),!.

תרגם\_פעולה\_לערך\_תווי(+,' + ').

תרגם\_פעולה\_לערך\_תווי(-,' – ').

תרגם\_פעולה\_לערך\_תווי(\*,' \* ').

תרגם\_פעולה\_לערך\_תווי(/,' / ').

תרגם\_פעולה\_לערך\_תווי(\_,' # ').

**טיפול בביטויים לוגיים.**

פסוק\_מסוג(\_א = \_ב,בהשוואה,[\_א,'שווה ל',\_ב]).

פסוק\_מסוג(\_א \= \_ב,בהשוואה,[\_א,'שונה מ',\_ב]).

פסוק\_מסוג(\_א =< \_ב,בהשוואה,[\_א,'קטן או שווה ל',\_ב]).

פסוק\_מסוג(\_א < \_ב,בהשוואה,[\_א,'קטן מ',\_ב]).

פסוק\_מסוג(\_א > \_ב,בהשוואה,[\_א,'גדול מ',\_ב]).

פסוק\_מסוג(\_א >= \_ב,בהשוואה,[\_א,'גדול או שווה ל',\_ב]).

פסוק\_מסוג(\_א =:= \_ב,בהשוואה,[\_א,'ו',\_ב,'בעלי אותו ערך']).

**טיפול במתארי רשימה**

פסוק\_מסוג(חבר(\_איבר,\_רשימה),מתארי\_רשימה, ['האיבר',\_איבר,'חבר ברשימה' \_רשימה]).

פסוק\_מסוג(שרשר(\_א,\_ב,\_רשימה),מתארי\_רשימה, ['משרשור עולה ש',\_א,'ו',\_ב,'בונים את',\_רשימה])

פסוק\_מסוג(ראשון\_ברשימה(\_א,\_רשימה),מתארי\_רשימה,['האיבר',\_א,'נמצא במקום הראשון ברשימה' \_רשימה]).

פסוק\_מסוג(מספר\_איברים(\_מ,\_רשימה),מתארי\_רשימה, [\_מ,'הוא מספר האיברים ברשימה',\_רשימה]).

פסוק\_מסוג(מצא\_כל(\_מ,\_תנאי,\_רשימה),מתארי\_רשימה,['מהפעלת מצא\_כל מתקבלת הרשימה',\_רשימה]).

**טיפול בטענת שלילה:**

פסוק\_מסוג(לא (\_מטרה),שלילה,['אי אפשר להסיק ש',\_מטרה]).

טענת השלילה תטופל ביחד עם החוק האחראי לטיפול במתאר פסוק\_מסוג/3 כאשר ההסבר יהיה "לא ניתן להסיק ש" ולאחר מכן הסבר הוכחת המטרה.

**חוק המטפל במתארי מערכת:**

פסוק\_מסוג(\_מטרה,מערכת,[\_מטרה,'היא פועלת מערכת']):-

\_מטרה\=(\_,\_),

לא (פסוק(\_מטרה,\_)).

כאשר פסוק אינו מצליח לפרק חוק לראש וגוף אזי החוק הינו פעלת מערכת. לדוגמא "קרא", "כתוב", "פתח\_חלון" וכו'

(גם !).

**טיפול בגוף חוק מורכב:**

המשך\_הוכחה((\_מטרה\_מבודדת,\_שאר),\_הסבר\_שאר):-

מטרה\_חסומה(\_מטרה\_מבודדת),!,

\_מטרה\_מבודדת,

המשך\_הוכחה(\_שאר,\_הסבר\_שאר).

המשך\_הוכחה((\_מטרה\_מבודדת,\_שאר),[\_הסבר\_מטרה|\_הסבר\_שאר]):-

הוכחה(\_מטרה\_מבודדת,\_הסבר\_מטרה),

המשך\_הוכחה(\_שאר,\_הסבר\_שאר).

המשך\_הוכחה(\_מטרה\_מבודדת,[\_הסבר\_מטרה]):-

הוכחה(\_מטרה\_מבודדת,\_הסבר\_מטרה).

טיפול בגוף חוק מורכב מקבל **רשימה עם סוגריים עגולים** כאשר **הפסיק ברשימה עם הסוגריים העגולים משמש כהפרד (ברשימה רגילה)**, כלומר מוציא מטרה אחת מבין כל המטרות, לאחר מכן בודק אם נמצאת בעבודה **מטרה\_חסומה/1** במקרה שכן לא מוחזר הסבר ושרשרת ההיסק תמשיך על שאר המטרות, במקרה שהמטרה אינה נמצאת באסורים, תמשך ההוכחה עליה דרך **המשך\_הוכחה/2** וישורשר ההסבר שלה בהתאם.

**פרק שישי – תיאור מנגנון ההסבר.**

**כתוב\_יפה(\_מטרה):-**

מילול(\_מטרה,\_מילולי),!,

כתוב\_רשימה(\_מילולי).  
  
חוק זה אחראי לתרגם את המטרה בצורה טבעית, כאשר התרגום יהיה באמצעות מילול/2.

דוגמא **למילול/2:**  
**מילול(אחזר\_תור(\_,\_תור),['כעת תורו של השחקן ה',\_צבע]):-**

תרגם\_מספר\_לצבע(\_תור,\_צבע).

לאחר הפעלת חוק זה יוחזר תורו של איזה שחקן (צבע השחקן הנוכחי).

**%הצג\_הוכחה( הסבר( \_מטרה, \_גוף\_או\_סוג, \_הסבר )).**

**הצג\_הוכחה(הסבר(\_עובדה,true,[]),\_תשובה):-**

כתוב\_יפה(\_עובדה),

כתוב\_בחלון(' נתון כעובדה'),

שורה\_בחלון,

כתוב\_בחלון('חזור/סוף'),שורה\_בחלון,

קרא\_בחלון(\_תשובה),

!.

חוק זה מטפל במקרה של עובדה, תחילה תיבדק העובדה אם קיים מילול עבורה ולאחר מכן יודפס אוטומטית בחלון "נתון כעובדה".

**הצג\_הוכחה(הסבר(\_פסוק\_מסוג,\_סוג,\_הסבר),\_תשובה):-**

פסוק\_מסוג(\_פסוק\_מסוג,\_סוג,\_הסבר),

כתוב\_רשימה\_של\_הוכחה (\_הסבר),

שורה\_בחלון,

כתוב\_בחלון('חזור/סוף'),שורה\_בחלון,

קרא\_בחלון(\_תשובה),!.

במידה וזהו פסוק הסבר מסוג מסוים יודפס ההסבר לאותו סוג.

**הצג\_הוכחה(הסבר(\_מטרה,\_גוף,\_הסבר),\_תשובה\_מוחזרת):-**  
 חזור,

כתוב\_יפה(\_מטרה),

כתוב\_בחלון(' כי:'),שורה\_בחלון,

כתוב\_תת\_מטרות(\_הסבר,1),

כתוב\_בחלון('מספר/חזור/סוף'),שורה\_בחלון,

קרא\_בחלון(\_תשובה),

המשך\_הצג\_הוכחה(\_תשובה,\_הסבר,\_תשובה\_מוחזרת),!.

במקרה של הסבר מורכב יודפס ראש החוק ואחריו הגוף בצורה מסודרת, החוק כתוב\_תת\_מטרות אחראי למיספור וסידור החוקים הנמצאים בגוף.

כתוב\_תת\_מטרות([],\_):-שורה\_בחלון.

כתוב\_תת\_מטרות([הסבר(\_תת\_מטרה,\_,\_)|\_שאר],\_מספר):-  
 כתוב\_בחלון(\_מספר),

כתוב\_בחלון(') '),

כתוב\_יפה(\_תת\_מטרה),

שורה\_בחלון,

\_מספר\_פלוס הוא \_מספר + 1,

כתוב\_תת\_מטרות(\_שאר,\_מספר\_פלוס).

חוק זה אחראי לכתיבת כל החוקים של גוף הוכחה בצורה מסודרת וממוספרת.

כתוב\_רשימה\_של\_הוכחה([]).

כתוב\_רשימה\_של\_הוכחה([סמן\_מיוחד(שורה)|\_שאר]):-

שורה\_בחלון,

כתוב\_רשימה\_של\_הוכחה(\_שאר).

כתוב\_רשימה\_של\_הוכחה([סמן\_מיוחד(רווח(\_רווח))|\_שאר]):-

רווח\_בחלון(\_רווח),

כתוב\_רשימה\_של\_הוכחה(\_שאר).

כתוב\_רשימה\_של\_הוכחה([\_מילה|\_שאר]):-\_מילה \= סמן\_מיוחד(\_),

לא מספר(\_מילה),

כתוב\_בחלון(\_מילה),

כתוב\_רשימה\_של\_הוכחה(\_שאר).

כתוב\_רשימה\_של\_הוכחה([\_מילה|\_שאר]):-\_מילה \= סמן\_מיוחד(\_),

תרגם\_מספר\_לערך\_תווי(\_מילה,\_מילה\_תווי),

כתוב\_בחלון(\_מילה\_תווי),

כתוב\_רשימה\_של\_הוכחה(\_שאר).

חוק הכותב רשימה של מילול בחלון על ידי פירוק כל מילה והדפסתה.

המשך\_הצג\_הוכחה(ח,\_,ח).

המשך\_הצג\_הוכחה(ס,\_,ס).

המשך\_הצג\_הוכחה(\_מספר,\_הסבר,\_תשובה):-

מיקום\_איבר(\_מספר,\_תת\_הסבר,\_הסבר),

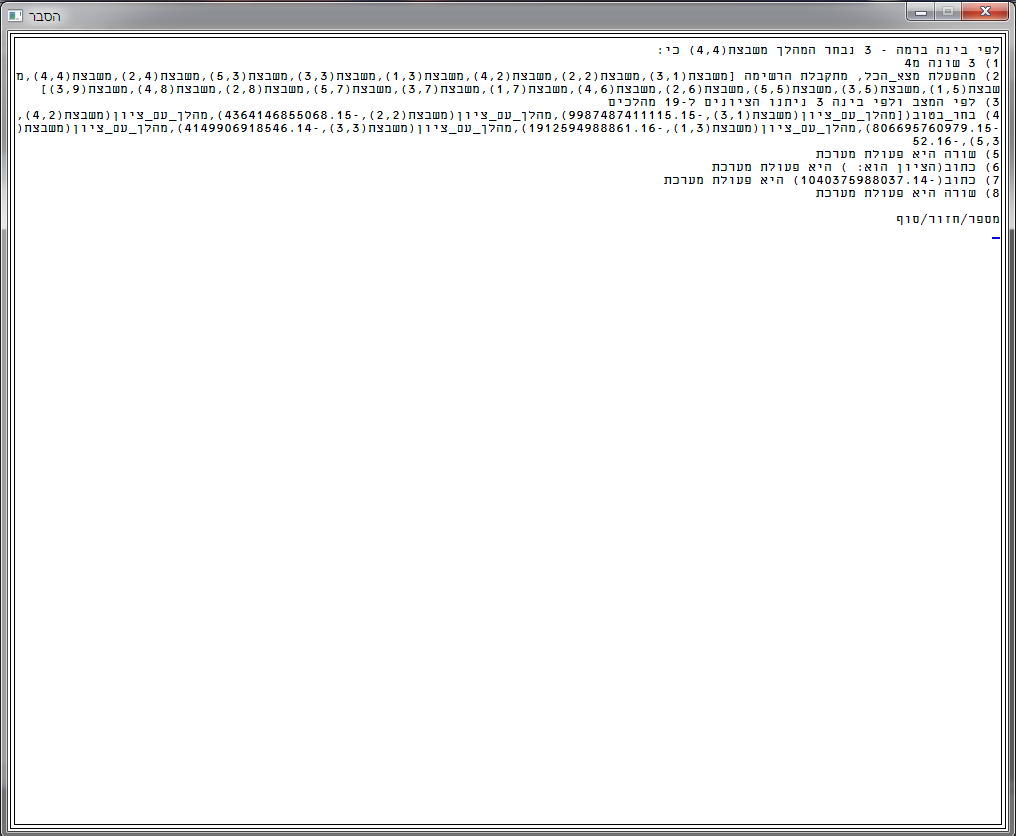
הצג\_הוכחה(\_תת\_הסבר,\_תשובה),

\_תשובה = 'ס'.  
חוק האחראי להמשך הדפסת התפריט בהתאם לבחירתו של המשתמש.

**דוגמא למנגנון ההסבר בפרוייקט שלי:**

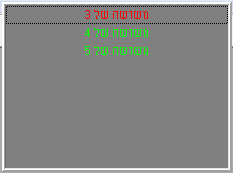
זוהי דוגמא לחלון ההוכחה/הסבר במשחק שלי:

בדוגמא זו רואים איך המשחק הגיע למתן ציון למהלך.



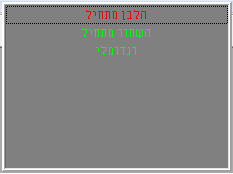
**פרק 7 – תיאור ממשק פלט/קלט**

מסך ראשון:



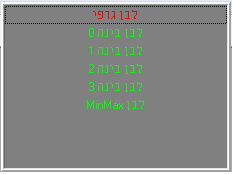
במסך זה יהיה על המשתמש לבחור את גודל הלוח בו הם רוצים לשחק.

מסך שני:



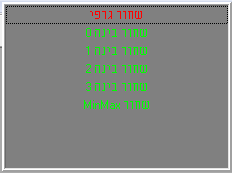
במסך זה יהיה על המשתמש לבחור מי מתחיל , לבן , שחור או רנדומלי.

מסך שלישי:



במסך זה יהיה על המשתמש לבחור האם הכלי הלבן יהיה שחקן אנושי או מחשב בעל רמות בינה שונות.

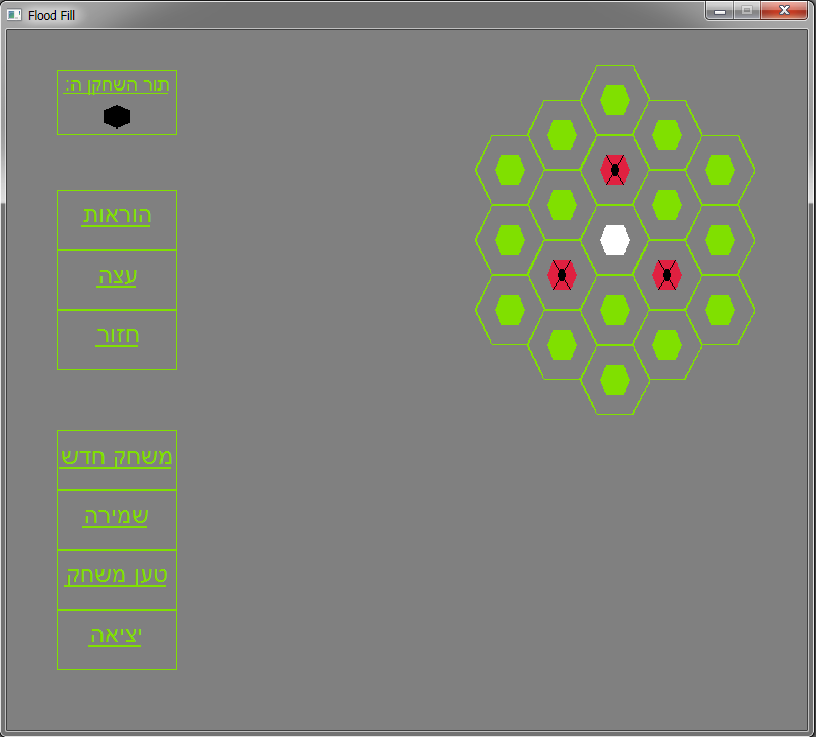
מסך רביעי:



במסך זה יהיה על המשתמש לבחור האם הכלי השחור יהיה שחקן אנושי או מחשב בעל רמות בינה שונות.

מסך המשחק:

פלט המחזיר את תור איזה שחקן



כפתור קלט המציג את הוראות המשחק

כפתור קלט הנותן עצה והסבר לה

כפתור קלט החוזר למצב הקודם

כפתור קלט היוצר משחק חדש

כפתור קלט השומר את המשחק

כפתור קלט הטוען את המשחק השמור

כפתור קלט הגרום ליציאה

מסך הפלט/קלט , לחיצה על הלוח תגרום לקליטת מיקום השחקן ולהצגתו על הלוח

**פרק 8 – סיכום – הרחבות ושיפורים אפשריים**

**8.1 – הרחבות שבוצעו במערכת**

כל מהלך הלמידה שלי התבצע לאורך כל השנה , בכל פעם אשר נדרשתי להכין חלק אחר מהפרויקט למדתי את הידע הדרוש לי להכין אותו. כמובן שרוב הידע התבסס על ידע קודם הנלמד לפני זה בכיתה י''א אך לפעמים היה הצורך ללמוד גם חומר חדש.

הידע שאני יצרתי בעצמי היה פיתוח אלגוריתם המאפשר את בניית הלוח משושה , זהו אלגוריתם שישבתי עליו זמן רב עד שלבסוף פתרתי אותו ללא שום עזרה חיצונית.

**8.2 – הצעות לשיפורים בעתיד**

לפי דעתי סביבת העבודה היא קצת בעייתית ויש לה המון באגים , אם בעתיד תהיה סביבת עבודה טובה יותר זה יקל על תהליך העבודה.

**8.3 – רפלקציה**

לפי דעתי המשחק הוא משחק מאוד חכם שיכול להביס כל אחד. היה נחמד מאוד לבצע את הפרויקט גם אם מידי פעם היה לחץ רב או המון בעיות שאנו לא ידענו כל כך איך להתעסק איתם , בסופו של דבר כאשר הצלחתי לפתור משהו תמיד זה היה מלווה בתחושה טובה.

**פרק 9 – נספחים**

**9.1 – הוראות הפעלה**

על מנת להתחיל את המשחק יש לפתוח את תוכנת ה – wideA ,לאחר מכן ללחוץ על כפתור ה – Listener וללחוץ על start . ברגע זה נפתח חלון הפקודות , כעת יש לטעון את קבצי המשחק וזאת על ידי לחיצה על כפתור ה – Listener שוב וללחוץ על Consult , כעת יפתח חלון אשר בו צריך לבחור את הקובץ אותו אתם רוצים לטעון , אצלנו זה יהיה קובץ ה – main בפרויקט. לאחר טעינת הקובץ תופיע הודעה ' הקבצים נטענו בהצלחה '. כעת צריך לרשום 'משחק.' בחלון , מה שיריץ את לולאת המשחק ויפתח לנו את התפריט הראשון. לאחר הבחירות בתפריטים לפי רצונם של המשתמשים המשחק יעלה ויהיה אפשר להתחיל לשחק בו.

**9.2 – ביבליוגרפיה - אתרי אינטרנט וספרים**

התוכנה של זיליון בה מצאתי ולמדתי את המשחק