**ממ"ן 17 – אהרן שנוולד – 200532521**

בחרתי לבנות בפרוייקט הסיום משחק דמקה. במשחק זה בחרתי לממש את חוקי דמקה כמו שאני גדלתי עליהם(American checkers)[[1]](#footnote-1), כאשר המתמודדים במשחק הם שחקן אנושי אשר משחק מול המחשב.

**א. סביבת עבודה וכלי פיתוח**

הלוגיקה שמאחורי המשחק בנויה בפרולוג (סביבת עבודה - גרסא SWI-Prolog 7.4.2) עם ממשק משתמש טקסטואלי שכתבתי בעצמי.

רמת הקושי במשחק ניתנת לבחירה ע"י המשתמש, כמו"כ גודל הלוח(אך תמיד יהיה NXN לפי חוקי המשחק).

ב. אלגוריתמים בשימוש

Alpha Beta – אימצתי את הגרסא מהספר[[2]](#footnote-2). ביצעתי שינוי בהגבלה של עומק החיפוש(הוספת המשתנה Depth ליחסים goodenough, boundedbest, alphabeta), מכיוון שפרישה של כל עץ המשחק תארך זמן רב ותפגע בחווית המשחק. ניתן להגדיר את רמות הקושי של המשחק מהקל אל הקשה, הרמה הקלה ביותר היא עומק חיפוש 2, והרמה הקשה ביותר היא עומק חיפוש 5.  
את היחס moves המצויין בספר שיניתי לפי צרכי המשחק אשר מתואר בסעיף ד'.

ג. פונקציה היוריסטית

תפקיד הפונקציה היוריסטית באלגוריתם אלפא-בטא הוא להחזיר "ציון" למהלך נתון(במקרה שלנו – עד כמה מצב מסויים(לוח מסויים) משתלם למחשב). ככל שהמהלך מקבל ציון גבוה יותר, כך הוא אטרקטיבי יותר למחשב. במהלך הכנת הפרוייקט חיפשתי פונקציה היוריסטית טובה, נתקלתי בפונקציות היוריסטיות מורכבות[[3]](#footnote-3) ובחרתי לממש את העקרון הפשוט המכונה Material Count, שמהותו היא ספירה של ה"רכוש" העדכני אל מול היריב. הפונקציה מחשבת את הציון באופן הבא:

X = מספר החיילים של המחשב

Y = מספר החיילים של השחקן האנושי

Z = מספר המלכים של המחשב

W = מספר המלכים של השחקן האנושי

נוסחא:

(X-Y)+1.4(Z-W)

לנוסחא זו נוסיף נוסחת פיצוי באופן הבא-

לכל מלך שנמצא בין השורות 3 עד 6 נפצה ב0.3 נקודות. באופן דומה כל מלך שנמצא בין העמודות 3-6 יפוצה ב0.3. כלומר ה"ציון" המקסימלי של מלך יכול להגיע ל2 נקודות סה"כ(לעומק נקודה בודדת לשחקן רגיל). מטרת נוסחת הפיצוי היא לגרום לדומיננטיות גדולה יותר של המלכים במרכז המשחק באלגוריתם אלפא-בטא(ולחווית משחק טובה יותר).

ד. חלוקה למודולים

לשם הקריאות, הסדר והתחזוק ארכיטקטורת המשחק מורכבת מקובץ מרכזי אחד שמרכז את עיקר הפעלת הלוגיקה ולמודולים שמטפלים בכל בעיה במשחק באופן נפרד(אך יודעים להתממשק זה עם זה). המודולים נמצאים בתיקים modules, ולצורך הטעינה שלהם השתמשתי ביחס ensure\_loaded המובנה בSWI שטוען את המודול במידה והוא לא נטען עדיין. להלן פירוט המודולים:

* Main – מרכז את הלוגיקה הראשית במשחק ומהווה את המפעיל המקשר בין המודולים.
* basic\_game\_relations – מרכז את יחסי היסוד של המשחק(לדוג, שחקנים, יחס בין שחקן למלך וכו').
* Board – מרכז את כל הפעולות על לוח המשחק(לדוגמא – בדיקת נכונות של מיקום מסויים בלוח, המרה מעמודות לאינדקס, יצירת לוח משחק משתנה וכו')
* game\_moves – מרכז את כל תנועת השחקנים על המגרש(לדוגמא – בדיקת חוקיות של תנועה מבוקשת על לוח המשחק, בדיקת חוקיות של ביצוע מהלך אכילה על המגרש ובהתאם לשחקנים וכו').
* Tui – text user interface – מרכז את כל הפעולות של אינטראקציה מול המשתמש.
* useful\_predicates – פרדיקטים גנריים שנעזרתי בהם במימוש הלוגיקה(לדוגמא – החלפת איבר ברשימה, פיצול רשימה ארוכה לתתי רשימות באורכים קבועים, ספירת ספציפיים ברשימה).
* alpha\_beta – מרכז את כלל הפעולות הקשורות להפעלת האלגוריתם אלפא ביתא, את האלגוריתם לקחתי מספר הלימוד(מקור[[4]](#footnote-4)) והוספתי את הפונקציה היוריסטית ואת הבונוסים.

**ה. יחסים מרכזיים בשימוש**

בסעיף זה אתייחס ליחסים מרכזיים בשימוש.

|  |  |
| --- | --- |
| **שם היחס** | **תיאור** |
| board | * לוח המשחק – שם היחס הוא board, יחס זה מכיל 64 אלמנטים(לוח המשחק הוא 8x8). מרחב המצבים של כל אלמנט בboard מיוצג ע"י אטום:   + **w –** שחקן לבן.   + **b –** שחקן שחור.   + **kw -** מלך לבן.   + **kb -** מלך שחור.   + **x –** משבצת שאי אפשר לנוע עליה.   + **e –** משבצת ריקה. |
| get\_legit\_move | יחס המקבל לוח משחק ושחקן נתון, ומחזיר מהלכי משחק חוקיים. היחס בודק קודם בהתאם לחוקי המשחק האם יש מהלך זמין של אכילה ולאחר מכן תנועה רגילה. |
| get\_legit\_regular\_movement | יחס המקבל לוח משחק ושחקן נתון ומחזיר מהלך **תנועה** בלבד. |
|  |  |
| IsLegalRegularMovement | יחס הבודק האם מהלך תנועה הוא חוקי – יחס המקבל מצב נתון, ומצב תוצאה של מהלך מסויים(תנועה) ובודק אם מהלך זה חוקי לפי חוקי המשחק. |
| isLegalEatMovement | יחס הבודק האם מהלך אכילה הוא חוקי – יחס המקבל מצב נתון, ומצב תוצאה של מהלך מסויים(תנועה) ובודק אם מהלך זה חוקי לפי חוקי המשחק. |
|  | יחס שבודק האם מהלך משחק מסוים הוא חוקי – מקבל מצב נתון, ומצב לאחר המהלך ובודק האם מהלך זה חוקי לפי תנאי המשחק(מפורטים בסעיף "חוקי המשחק"). נבדיל בין מהלך אכילה(בו אוכלים שחקן יריב אחד או יותר) לבין מהלך תנועה רגיל. מכיוון שלפי חוקי המשחק במידה וישנה אפשרות לאכול שחקן יריב חובה לבצע את המהלך, נבדוק האם קיימת אפשרות לאכול שחקן יריב ולממש את אפשרות זו, במידה ולא קיימת אפשרות נבחן את מהלכי התנועה האפשריים. |
| **moves** | יחס שפורש את עץ המשחק – מקבל מצב משחק נתון ומחזיר את כל האפשרויות למהלכים חוקיים במשחק. |
|  |  |
|  |  |
|  |  |

* -
* -
* יחס שבודק האם מהלך משחק מסוים הוא חוקי – מקבל מצב נתון, ומצב לאחר המהלך ובודק האם מהלך זה חוקי לפי תנאי המשחק(מפורטים בסעיף "חוקי המשחק"). נבדיל בין מהלך אכילה(בו אוכלים שחקן יריב אחד או יותר) לבין מהלך תנועה רגיל. מכיוון שלפי חוקי המשחק במידה וישנה אפשרות לאכול שחקן יריב חובה לבצע את המהלך, נבדוק האם קיימת אפשרות לאכול שחקן יריב ולממש את אפשרות זו, במידה ולא קיימת אפשרות נבחן את מהלכי התנועה האפשריים.

**ו. תיאור ממשק המשתמש והוראות הפעלה**

* לצורך הפעלה המשחק יש לטעון את main.pl ולהקיש:

?- start\_game.

* לאחר מכן תגיע הגדרת גודל הלוח של המשחק:

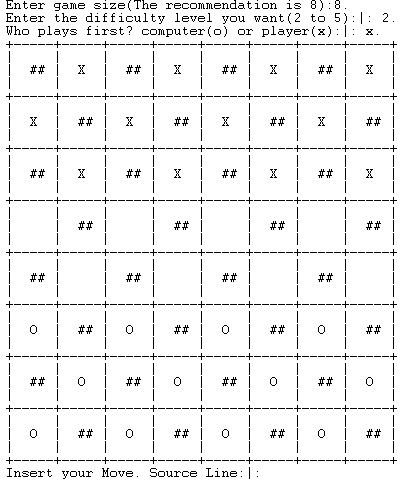
Enter game size(The recommendation is 8):

* רצוי להקיש 8, לטובת קבלת לוח 8X8. במידה ורוצים אפשר לקבל לוח בגדלים אחרים(המינימלי הוא 8).
* לאחר מכן יופיע על המסך בחירה של רמת הקושי:

Enter the difficulty level you want(2 to 5):

* רמת הקושי היא בסדר עולה, ניתן להקיש לפי דרישות המשתמש מהמשחק(ככל שהקושי עולה, כך למחשב ייקח זמן רב יותר לבצע מהלכים).
* לאחר מכן תהיה אפשרות בחירה איזה שחקן מתחיל:

Who plays first? computer(o) or player(x):

* במידה ורוצים שהמחשב יתחיל צריך להקיש o במידה ורוצים שהשחקן יתחיל יש להקיש x.לאחר תהליך ההגדרות יופיע לוח המשחק: 
* יש להקיש את מספר שורת המקור, מספר עמודת המקור, מספר שורת היעד, מספר עמודת היעד(לאחר כל הקשה לחיצה על מקש Enter). לדוגמא:

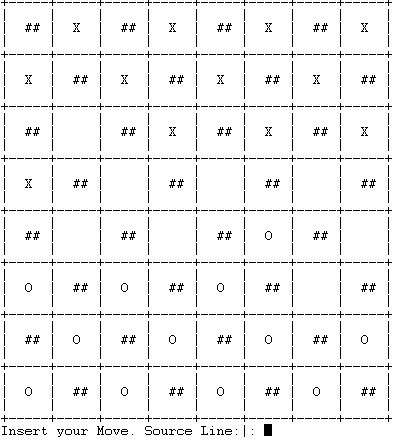
Insert your Move. Source Line:|: 3.

Source Column:|: 2.

Destination Line:|: 4.

Destination Column:|: 1.

* לאחר הקשה על תנועה **חוקית** יופיע המסך הבא לאחר ביצוע מהלך מצד המחשב:



* במידה ורוצים לצאת מהמשחק, יש ללחוץ cntrl+c ולאחר מכן להקיש את האות a.
* לצורך חזרה למשחק יש לכתוב את הפקודה start\_game.
* תשומת לב: במידה ויהיו שגיאות, בהקשה המשחק יזהה אותן וידרוש הזנה של מהלך חדש.

1. https://en.wikipedia.org/wiki/Draughts [↑](#footnote-ref-1)
2. http://media.pearsoncmg.com/intl/ema/ema\_uk\_he\_bratko\_prolog\_3/prolog/ch22/fig22\_5.txt [↑](#footnote-ref-2)
3. http://library.msri.org/books/Book29/files/schaeffer.pdf [↑](#footnote-ref-3)
4. http://media.pearsoncmg.com/intl/ema/ema\_uk\_he\_bratko\_prolog\_3/prolog/ch22/fig22\_5.txt [↑](#footnote-ref-4)