פרוייקט בפרולוג 20596 - דמקה



ינאי אלטר 211377775

**תיאור המשחק**

המשחק הוא למעשה דמקה בגודל NXN, כאשר טוענים את הפרוייקט ניתן לבחור את גודל הלוח. המשחק הוא נגד המחשב. המחשב עובד לפי פונקציה היוריסטית והאלגוריתם מינימקס עם גיזום אלפא ביטא. יש לציין כי הוספתי שדרוג למינימקס ככה שיהיה מוגבל בעומק (כי אחרת נתקע בעץ מאוד מאוד גדול וזמן ביצוע גדול מידי).

**צילומי מסך מהמשחק והסברים**



הסבר: כאשר אנו מפעילים את התכנית (אפשר מהCLI בעזרת הפקודה (swipl -s <project\_name>, נוכל לאתחל את המשחק בעזרת InitialState(N), כאשר N זה גודל השורות (והעמודות) בלוח.

כעת תורנו לשחק, נוכל לבצע תזוזה על ידי הפרדיקט realMoveWhite(Row,Column,NewRow,NewColumn). למשל נבצע את הפקודה:

realMoveWhite(5,1,4,2). ונקבל:



כעת נוכל לבקש מהמחשב להגיב על ידי הפרדיקט makeBlackMove. להלן הדוגמה:



המחשב (המסומן בb) ישאף לאכול כמה שיותר וגם לעשות מלך. הוא יכול לאכול כמה ברצף אם זה יחשב מהלך טוב עבורו. כאשר אחד מהשחקנים מגיע למשבצת הראשונה של היריב, הוא הופך אוטומטית למלך ואז יש לו אפשרות לחזור אחורה.

אם אנחנו (המשתמש ולא המחשב) רוצים לאכול שחקן של המחשב, אפשר לעשות זאת על ידי הפרדיקט eatBlack(StartRow,StartColumn,ListOfEatPositions,EndRow,EndColumn), כלומר, נותנים את המיקום ההתחלתי, את הרשימה של כל המיקומים של השחקנים שאנחנו רוצים לאכול, ואת המיקום הסופי, התוכנה תבדוק שהאכילה חוקית ולאחר מכן האכילה תתבצע.

לדוגמה:



דוגמה בה המחשב עושה מלך (מלך של המחשב מסומן בB גדולה, מלך של המשתמש מסומן בW גדולה).



מצב ניצחון – כאשר לא נותרו חיילים למחשב, המשתמש ניצח, וכנ"ל הפוך. במצב כזה המערכת תדע להתריע מי ניצח.

**ארכיטקטורה כללית**

מצבי המשחק הם זוגות סדורים של מטריצת ייצוג של כל הלוח והתור (black or white). המשחק מבוסס על אלגוריתם המינימקס ועיקר קושיו היה לממש את הפעולות כפי שמוגדרות בממ"ן 16:

moves(Pos,PosList) – בהינתן מצב משחק Pos הפרדיקט יחזיר את רשימת מצבי המשחק אליהם ניתן להגיע.

Max\_to\_move – מחזירה אמת אם תור המחשב לזוז

Min\_to\_move – מחזירה אמת אם תור המשתמש לזוז

Staticval – בהינתן מצב משחק מחזירה את ערך הפונקציה היוריסטית של מצב המשחק

בנוסף, שיניתי את האלגוריתם שמופיע בספר לalphabeta כך שיעבוד עם עומק מוגבל. עשיתי זאת על ידי הוספת פרמטר Depth שבכל פעם שאנחנו מפתחים מצב חדש, אנחנו מורידים אותו באחד. ניתן לראות זאת בקוד. השינוי עצמו לא גדול אך לקח לי זמן לחשוב עליו, הייתי צריך לקרוא את האלגוריתם בספר ולהבין אותו טוב בשביל לעשות את זה כמו שצריך.

הפעולה moves מתחלקת ל2 לפי התור של המשתמש. ללא הגבלת כלליות וסימטריה ניתן לדבר למשל על המחשב. השיטה moves מאגדת את כל האכילות שהמחשב יכול לעשות במצב משחק נתון, ואת כל התזוזות ללא אכילות שהוא יכול לעשות, ויוצרת משתי הרשימות האלה רשימה אחת. למעשה היא משתמשת בפרדיקטים allBlackMoves(Pos,Res),allBlackEats(Pos,Result), ואת שתי התוצאות המוחזרות היא מאגדת לתוך תוצאה אחת.

הפרדיקט allBlackMoves מבוסס על הפרדיקט moveBlack, שבהינתן מיקום של אחד מהחיילים, זה נותן רשימה של כל המקומות שאפשר ללכת אליהם. התפקיד של allBlackMove זה פשוט לעשות findall ולעבור על כל המיקומות של החיילים בלוח ולהפעיל עליהם את moveBlack.

באופן דומה עובדת השיטה allBlackEats.