**תיעוד עבור ex3.s11**

פעילות התוכנית – התוכנית תקבל לוח דמקה במצב חוקי של מהלך משחק, ואת השחקן הנוכחי, ותחזיר את המהלך   
המיטבי עבור אותו שחקן. כאשר ההגדרה של מהלך מיטבי היא מהלך אשר עבורו כל מהלך שיעשה היריב, יהיה הכי רחוק מהפסד אשר נוכל. כלומר מהלך אשר עושה מינימליזציה לנזקים.

אופן קבלת פרמטרים - התוכנית מקבלת קלט in common area:

בתווית Board: צריך להיכנס הלוח דמקה עם מיקומי החיילים כאשר 0 מייצג תא ריק, 1 מייצג שחקן לבן ו 2 מסמן שחקן שחור.

בתווית Player: שמור מספר השחקן הנוכחי.

בתווית Steps: נמצא פרמטר עומק הבדיקה עבור מהלך. אשר מסמל כמה מהלכים קדימה נרצה לבדוק בכדי להחליט מה המהלך האולטימטיבי.

התוכנית מוציאה את הפלט שלה גם כן in common area:

בתווית SrcPos: ישמר האינדקס של המשבצת בה נמצא השחקן שאותו צריך להזיז בשביל לבצע את המהלך האולטימטיבי.

בתווית DstPos: ישמר האינדקס של המשבצת אלייה נרצה להזיז את אותו השחקן

בתווית WinParam: ישמר פרמטר ניצחון שהוא תוצאה עבור חישוב העדיפות של כל מהלך, כאשר מהלכים עם WinParam יותר גבוה עדיפים על Winparam יותר נמוך.

Defines נחוצים לפעולת הפונקציות של התוכנית:

|  |  |
| --- | --- |
|  | Black = 2 |
|  | White = 1 |
|  | SkipC = 16 |
|  | SkipF = 22 |
|  | Empty = 0 |
|  | EdgeR = 16 |
|  | EdgeL = 0 |
|  | Win = 20 |
|  | Loss = 177760 ;which is actually -20 |
|  | Tie = 0 |
|  | Lowest = 177757; which is actually -21 |
|  | Highest = 21 |

שגרות -   
**checkWin**: בודקת עבור השחקן אם ניצח עבור מצב לוח נתון  
**שיטת העברת פרמטרים:** מקבלת את הפרמטרים דרך המחסנית. מכניסים את הפרמטרים לפי הסדר הנ"ל:  
הלוח במצב הנתון מוכנס לפי כתובת, אחריו נכנס השחקן הנוכחי לפי ערך.

הפלט מוחזר בתחתית המחסנית לפני כל הפרמטרים אשר הועברו. מחזיר 1 עבור נצחון לאותו שחקן ו -1 לכל מצב אחר

רגיסטר הקישור של השגרה – PC

**לייבלים אשר תפוסים על ידי הפונקציה:**

countP2, movesP1, movesP2, findDiff, playWin, PLYnoWIN, PlayCnt, OppCnt

gwpaPrep

gwpBlack

gwpWhite

prepMov

getMaxWP

getWP\_aux

gwpaDoB

gwpaDoW

**movPiece**: מזיזה חייל כלשהו בלוח ממקום אחד למקום אחר (לשים לב, הפונקציה לא מבצעת אכילה)

**שיטת העברת פרמטרים:** מקבלת את לוח המשחק הנתון דרך המחסנית לפי כתובת. ומקבלת את הפרמטרים הבאים inline לפי הסדר הנ"ל: האינדקס בלוח אשר ממנו מזיזים את החייל והאינדקס בלוח אשר אליו מזיזים את החייל, שניהם לפי ערך.  
\*האינדקסים הם כל אחד מהמספרים בין 0 ל 63 בדצימלי, לא מוגבל רק לערכים זוגיים\*

הפלט הוא השינוי בלוח ומתעדכן ישר בלוח המקורי.

רגיסטר הקישור של השגרה – r5

**לייבלים אשר תפוסים על ידי הפונקציה:** אין.

**calcWP**: מחשבת את פרמטר הניצחון עבור מצב מסויים בלוח אשר אינו נצחון. הנוסחה עבור ההירויסטיקה בה אנו משתמשים היא ההפרש בין מספר השחקנים של השחקן ומספר השחקנים של יריבו.

**שיטת העברת פרמטרים:** מקבלת דרך המחסנית את הערכים הבאים לפי הסדר הנ"ל:  
מקבלת את הלוח של המצב הנתון לפי כתובת, והשחקן הנוכחי אחריו לפי ערך.

השגרה מחזירה את WP דרך הרגיסטר r4.

רגיסטר הקישור של השגרה – PC

**לייבלים אשר תפוסים על ידי הפונקציה:** אין

**makeMov**: שגרה רקורסיבית, אשר מקבלת חייל להזיז ואת המשבצת אליה נרצה להזיז את החייל, הפונקציה מזיזה את החייל, אם היא עוברת מעל חייל של השחקן הנגדי היא גם אוכלת את החייל (מעלימה אותו). הפונקציה גם מחסירה אחד Steps לפני שהיא קוראת לפונקציה הרקורסיבית הבאה, getWP\_aux אשר קוראת לmakeMov בהתאם. בעת קיפול הרקורסיה הפונקציה תקבל ערך WP מקסימלי או מינימלי בהתאם לעומק הרקורסיה והשחקן עבורו בודקים.  
הפונקציה תעדכן את הערך WP המיטבי אשר נמצא עבור השחקן שקרא לפונקציה הזאת, וכך באופן רקורסיבי תעדכן את השחקן המקורי בWP המיטבי שאפשר לקבל על פי קריטריוני minimax.  
לאחר מכן הפונקציה תחזיר את השינויים שעשתה, תחזיר את השחקן למיקומו המקורי ותחזיר ללוח את השחקן שנאכל.

**שיטת העברת פרמטרים**: האינדקס של המשבצת של השחקן שנרצה להזיז תועבר ברגיסטר r4 לפי ערך, והאינדקס של המשבצת אליה נעביר את השחקן מועברת ברגיסטר r5 לפי ערך. שאר הערכים מועברים במחסנית בסדר הבא:  
הלוח במצב הנוכחי שלו מועבר במחסנית לפי כתובת, אחריו מועבר השחקן המקורי אשר עבורו מחפשים מהלך אופטימלי לפי כתובת. אחריו מועבר steps נותרים לפני עצירת הרקורסיה, לפי ערך.   
אחריו מועבר הערך המקסימלי/המינימאלי של WP אשר נמצא עד כה עבור השחקן עבורו makeMov בודקת כרגע לפי ערך.  
ובסוף המחסנית מועבר השחקן אשר עבורו אנו בודקים את המהלך הנוכחי במטרה לקבל WP אולטימטיבי.

הפלט יצא בצורת עדכון של הערך המקסימלי/מינימאלי של WP אשר נמצא עבור הצעדים שנעשו עד כה.  
אם המהלך לא יוציא WP עדיף על הערך שנמצא עבור צעדים אחרים, הערך לא יתעדכן.

רגיסטר הקישור של השגרה – PC

לסיכום, המחסנית בעת הכניסה נראית בצורה הבאה

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Current\_player (the one currently trying a move) | Local\_apex | steps | #Original Player | #Board |

ב r4 נמצאת משבצת המקור ו ב r5 נמצאת משבצת היעד

**לייבלים אשר תפוסים על ידי הפונקציה:**

positive, noLeap, findMin, reverse, noLeapReverse, RvrsSrc, RvrsDst

**cntPlay**: סופרת כמה שחקנים יש על הלוח של השחקן.

**שיטת העברת פרמטרים:** מקבלת דרך המחסנית את הערכים הבאים לפי הסדר הנ"ל: הלוח המצב המסויים נשלח לפי כתובת,  
אחריו נשלח השחקן הנוכחי עבורו נרצה לספור על פי ערך. הפלא נשלח במחסנית לפני כל אחד מן הפרמטרים אשר הוכנסו לשגרה.

רגיסטר הקישור של השגרה – PC

**לייבלים אשר תפוסים על ידי הפונקציה:**

loop1, exit2, BoardS

**canMove**: בודקת אם נשארו עוד מהלכים לשחקנים.

**שיטת העברת פרמטרים:** מקבלת דרך המחסנית את הלוח במצב הנתון לפי כתובת. ומחזירה בתחתית המחסנית 1 אם שני השחקנים לא יכולים לזוז ואחרת מחזירה -1.

רגיסטר הקישור של השגרה – PC

**לייבלים אשר תפוסים על ידי הפונקציה:**

cmExit2

**minOrMax**: בתוכנית שלנו אנו משתמשים באלגוריתם minimax, האלגוריתם מסתמך על חישוב היוריסטיקה מקסימלית עבור השחקן הנתון, וחישוב WP מינימלי עבור היריב. השגרה הנתונה בודקת מתי נרצה לקחת את הWP המקסימלי.

**שיטת העברת פרמטרים:** מקבלת את את הפרמטרים במחסנית בסדר הבא: השחקן עבורו אנו "מנבאים את המהלך הבא" לפי ערך, והשחקן שעבורו מלכתחילה אנו רוצים למצוא את המהלך הטוב ביותר לפי כתובת.  
מחזיר את הפלט דרך r4, 1 אם מחפשים ערך מקסימלי ו -1 אם מחפשים ערך מינימאלי.

רגיסטר הקישור של השגרה – PC

**לייבלים אשר תפוסים על ידי הפונקציה:**

momNext

**cntMoves**: סופר כמה מהלכים אפשריים קיימים לשחקן הנתון בלוח הנתון. (שימו לב, הפונקציה לא מתייחסת לאכילה אחורה)

**שיטת העברת פרמטרים**: מקבלת דרך המחסנית את הפרמטרים דרך המחסנית בסדר הבא:  
לוח המשחק במצב חוקי אשר מתקבל במהלך המשחק לפי כתובת, ואת המספר של השחקן אשר לו אנו רוצים לבדוק את מספר המהלכים בתור ערך.  
הפלט מועבר בתחתית המחסנית.

רגיסטר הקישור של השגרה – PC

**לייבלים אשר תפוסים על ידי הפונקציה:**

loopB, leapLB, borderRB, tileRB, leapRB, nextB, loopW, leapLW, borderRW, tileRW, leapRW, nextW, BoardB, BoardE, BordLRo

**isOnLeft**: בודק עבור חייל מסויים בלוח אם הוא נמצא בצד הכי שמאלי של הלוח.

**שיטת העברת פרמטרים:** מקבלת דרך המחסנית את האינדקס של המשבצת אותה בודקים לפי ערך. מחזירה 1 אם המשבצת נמצאת בצד השמאלי של הלוח ו – 0 אם הוא לא. הפלט מוחזר בתחתית המחסנית.

רגיסטר הקישור של השגרה – PC

**לייבלים אשר תפוסים על ידי הפונקציה:**

iolNext

**isOnRght**: בודק עבור חייל מסויים בלוח אם הוא נמצא בצד הכי ימני של הלוח.

**שיטת העברת פרמטרים:** מקבלת דרך המחסנית את האינדקס של המשבצת אותה בודקים לפי ערך. מחזירה 1 אם המשבצת נמצאת בצד הימני של הלוח ו – 0 אם הוא לא. הפלט מוחזר בתחתית המחסנית.

רגיסטר הקישור של השגרה – PC

**לייבלים אשר תפוסים על ידי הפונקציה:**

iorNext

:**isOnTop** בודק עבור חייל מסויים אם הוא נמצא בחלק העליון של הלוח.

**שיטת העברת פרמטרים:** מקבלת דרך המחסנית את האינדקס של המשבצת אותה בודקים לפי ערך. מחזירה 1 אם המשבצת נמצאת בחלק העליון של הלוח ו – 0 אם הוא לא. הפלט מוחזר בתחתית המחסנית לפני כל הפרמטרים.

רגיסטר הקישור של השגרה – PC

**לייבלים אשר תפוסים על ידי הפונקציה:**

iotNext

**isOnBottom**: בודקת עבור חייל מסויים בלוח אם הוא נמצא בחלק התחתון של הלוח.

**שיטת העברת פרמטרים**: מקבלת דרך המחסנית את האינדקס של המשבצת אותה בודקים לפי ערך. מחזירה 1 אם המשבצת נמצאת בחלק התחתון של הלוח. ו – 0 אם הוא לא. הפלט מוחזר בתחתית המחסנית לפני כל הפרמטרים.

רגיסטר הקישור של השגרה – PC

**לייבלים אשר תפוסים על ידי הפונקציה:**

iobNext

לבדוק העברות של common area

לבדוק שימוש בלייבלים

הערות על הדיפיינים  
שימוש ב BOARD שורה 652