מטרת התוכנית – ניהול משחק דמקה

פונקציה 1: printStop – מדפיסה את הודעת ה-stop

קלט: במחסנית: 0/1/2 כאשר 0 מסמן תיקו, 1 מסמן השחקן הלבן ניצח ו-2 השחקן השחור ניצח, תא הבא: השחקן הנוכחי.
(משתמש גם ב-msgWP משתנה גלובאלי לצורך הדפסת WP)

פלט: מדפיס את הודעת ה-stop על המסך.

מס"ד	Label	תפקיד
1	printTie	במקרה של תיקו- מדפיס הודעת תיקו מתאימה

פונקציה 2: damka – ניהול המשחק המרכזי ששומר על כל חוקי המשחק.

קלט: הפונקציה מקבלת קלט מהמשתמש לפי הפקודות שהוא מכניס.

פלט: הפונקציה תדפיס את המשחק על המסך.

מס"ד	Label	תפקיד
1	gameOff	צריך לשנות לאות קטנה בתחילת המילה
2	humTurn	מנהלת את המשחק עבור תור של שחקן אנושי
3	wfc	לולאה הממתינה לפקודה מהמשתמש בעת מהלך אנושי
4	noTime	לייבל זה מטפל בסיטואציה בו נגמר הזמן במהלך תור אנושי
5	victory	לייבל זה מטפל במצב שבו יש ניצחון לאחר מהלך אנושי או של בינה מלאכותית כאחד.
6	nextT	מאפס את כל המשתנים לקראת התור הבא, מחליט אם זה תור אנושי או של מחשב.
7	comTurn	לייבל זה מטפל במהלך של מחשב.
8	stopCom	לייבל זה מטפל במצב שמחשב קיבל פקודת stop במהלך החישובים שלו ועזב את הלוח "מבולגן", הלייבל ישחזר את הלוח לקדמותו.

פונקציה 3: opMove - הפונקציה בודקת האם הפקוד move שהמשתמש הקליד תקינה.

קלט: הפוקנציה משתמש בלייבל Input המכיל את הקלט מהמשתמש.

פלט: אם הפקודה הייתה תקינה – 1 במחסנית אחרת -0.

מס"ד	Label	תפקיד
1	notMove	במידה והפקודה לא הייתה תקינה ייתבצע שחזור לערכי האוגרים ויוחזר 0 במחסנית

פונקציה 4: iniTime – הפונקציה מאתחלת את המשתנים הגלובליים של השעון כדי לבדוק את הזמן.

קלט: אין, פלט: מעדכנת את המשתנים הגלובליים בהתאם.

פונקציה 5: inp - הפונקציה מטפלת בכל הקלט מהמשתמש

מקרא רגיסטרים:

r0 – הערך האסקי של התו המבוקש.

מס"ד	Label	תפקיד
1	inp	בודקת דגלים (האם מותר להשתמש בקלט כעת) והאם חרגנו מגודל ה input המותר
2	Get	אחראית לקבל תו מהמשתמש ולהדפיס אותו למסך (echo) – במידה והתו דורש טיפול מיוחד תשלח ללייבל המתאים
3	inpEnt	במידה והמשתמש מקליד קלט חדש לאחר שהקליד enter בקלט הישן תווית זו תאתחל את המקום בזיכרון ל Input.
4	endEn	תווית זו אחראית לבדוק האם הפקודה שהוכנסה היא חוקית (לאחר ש get זיהתה שהוקש enter)
5	backspace	תווית זו אחראית לבצע מחיקות כאשר תו backspace נקלט
6	inputPointer	משמשת כpointer למקום בזיכרון שאליו אנו כותבים את ה Input.

פונקציה 6: OPstop – פונקציה המטפלת במקרה שבו הוקלדה הפקודה stop בצורה תקינה.

קלט: אין קלט, שימוש בלייבל input המכיל את הקלט שהוקלד על ידי המשתמש/שחקן.

פלט: במחסנית 1 – אם התבצעה פקודת ה-stop אחרת 0.

מקרא רגיסטרים:

r0 – מחזיק מבציע לקלט.

r1 – WP של השחקן הלבן.

r2 - WP של השחקן השחור.

מס"ד	Label	תפקיד
1	notStop	במידה והקלט לא היה נכון מהמשתמש – יוחזר הערך 0 במחסנית ותודפס הודעה על שגיאה
2	tieWP	לאחר חישוב ה-WP, במידה ויש תיקו, נדפיס הודעה בהתאם
3	blackWin	מטפל במקרה בו השחקן השחור ניצח ומדפיס הודעה בהתאם
4	whiteWin	מטפל במקרה בו השחקן הלבן ניצח ומדפיס הודעה בהתאם

5	wpBlack	מעדכנת את הלייבל שיישלח להדפסה להיות השחור ומשנה אותו לערך אסקי לפניי (שיהיה מוכן להדפסה) ומדפיסה הודעת stop מתאימה
6	wpWhite	מעדכנת את הלייבל שיישלח להדפסה להיות הלבן ומשנה אותו לערך אסקי לפניי (שיהיה מוכן להדפסה) ומדפיסה הודעת stop מתאימה

<u>פונקציה 7:</u> decideTurn – הפונקציה אומרת האם הטור הבא הוא של המחשב או של אדם.

קלט: משתנים גלובלים

פלט: "1" אם התור הבא הוא של שחקן אנושית "-1" אם התור הבא הוא של שחקן מחשב

מס"ד	Label	תפקיד
1	identifyB	בודקת האם השחקן השחור הוא אדם, אם לא הוא מחשב. הערך הנכון יוחזר במחסנית.

<u>פונקציה 8:</u> - printP- מדפיסה את השחקן הנוכחי.

קלט: בלייבל הגלובלי CP יושב השחקן הנוכחי.

פלט: מדפיס את השחקן הנוכחי.

<u>פונקציה 9:</u> validMove - הפונקציה בודקת האם המהלך שנקלט הוא חוקי על הלוח עצמו.

קלט: אין קלט, שימוש במשתנים גלובליים.

פלט: במחסנית – 1 אם המהלך חוקי, אחרת 0.

<u>מקרא רגיסטרים:</u>

r0 – מכיל את כתובת המקור. (מאיפה לזוז)

r1 – מצביע למערך המהלכים (כמות הכתובות שנדרש להוסיף או לחסר כדי לבצע מהלך).

r2 – מצביע למהלך הבא שאנו בודקים.

r3 – מצביע אריתמטי לחישובים כדי לדעת את המיקום בלוח.

מס"ד	Label	תפקיד
1	checkValid	בודקת האם כתובת המקור מכילה את השחקן של השחקן הנוכחי, במידה ולא המהלך לא חוקי.
2	location	לייבל זה בודק את מיקום העמודה של השחקן על הלוח לצרכי בדיקת כיווני מהלכים
3	Left	לייבל זה אחראי לבדוק האם המהלך הוא מהלך שמאלה ואם כן לבדוק את תקינותו
4	EL	לייבל זה אחראי לבדוק האם המהלך הוא מהלך אכילה לשמאל ואם כן לבדוק את תקינותו
5	Right	לייבל זה אחראי לבדוק האם המהלך הוא מהלך ימינה ואם כן לבדוק את תקינותו
6	RE	לייבל זה אחראי לבדוק האם המהלך הוא מהלך אכילה לימין ואם כן לבדוק את תקינותו

7	goodCor	לייבל זה מטפל בערכי החזרה לקורדינטות תקינות
8	badCor	לייבל זה מטפל בערכי החזרה לקורדינטות לא תקינות

<u>פונקציה 10:</u> checkINP - הפונקציה בודקת את הקלט ומחולקת למקרים:

caseS0 - במצב זה ה-stage הוא 0 כלומר המשחק עדיין לא פעיל, הדבר היחידי שניתן לקלוט הוא start.

caseS1 – במצב זה ה-stage הוא 1 כלומר המשחק פעיל והשחקן הנוכחי הוא מחשב. פקודות מותרות בזמן ריצה במצב זה: stop, time

caseS2 - במצב זה ה-stage הוא 2 כלומר המשחק פעיל והשחקן הנוכחי הוא אדם. פקודות מותרות בזמן ריצה במצב זה: stop, time, move

קלט: אין.

פלט: אין ערך החזרה, מעדכן משתנים גלובליים ובודק את הקלט.

<u>מקרא רגיסטרים:</u>

r0 – מחזיק את המצביע לקלט שהוקלד.

מס"ד	Label	תפקיד
1	endCase1	במידה והקלט לא תקין מסיים את ה-case
2	humanP1	מעדכן את המשתנה Player1 להיות אדם בהתאם למה שהוכנס בקלט
3	ComputP1	מעדכן את המשתנה Player1 להיות מחשב בהתאם למה שהוכנס בקלט
4	errorINP	במידה והערכי המספרים שהוכנסו לא היו תקינים – מודפסת שגיאה
5	humanP2	מעדכן את המשתנה Player2 להיות אדם בהתאם למה שהוכנס בקלט
6	ComputP2	מעדכן את המשתנה Player2 להיות מחשב בהתאם למה שהוכנס בקלט
7	validINP	במידה וכל הקלט היה תקין – יש אתחול למשתנים המתאימים והדפסת הודעת start game
8	toError	קופץ לשגיאה

<u>פונקציה 11:</u> skipSpace – הפוקנציה מדלגת על רווחים בקלט.

קלט: מצביע לתו הנוכחי בקלט.

פלט: מצביע לתו הנוכחי הבא שאינו רווח.

<u>פונקציה 12:</u>

checkNum- הפונקציה בודקת את תקינות ערכי המספרים שנקלטו. (למשל נדרש שהמספרים לא יהיו שליליים או שונים מתווים שאינם מספרים)

קלט: במחסנית יוקצו 2 מקומות ריקים ומצביע לתו הנוכחי.

פלט: במחסנית: מספר התווים שהיו בקלט, 1- אם המספרים שנקלט תקינים אחרת 0.

מקרא רגיסטרים:

מס"ד	Label	תפקיד
1	loopN	מעבר על התווים בלולאה – עד לקבלת התו האחרון שמייצג מספר ומעדכן בהתאם במחסנית שהמספר תקין
2	validN	במידה והמספר היה תקין, מעדכן את המצביע לתו הבא ומשחזר רגיסטרים
3	invalidN	במידה והמספר לא תקין – מעדכן את המצביע לתו הבא, מעדכן במחסנית את ה-counter וערך ההחזרה להיות 0

פונקציה 13:

chkNum7- הפונקציה בודקת את תקינות ערכי המספרים שנקלטו (למשל נדרש שהמספרים לא יהיו שליליים או שונים מתווים שאינם מספרים) ובנוסף בודקת שהמספרים שונים מ-7.

קלט: במחסנית יוקצו 2 מקומות ריקים ומצביע לתו הנוכחי.

פלט: במחסנית: מספר התווים שהיו בקלט, 1- אם המספרים שנקלט תקינים אחרת 0.

פונקציה 14: opTime – הפונקציה מחשבת את הזמן שנשאר ומעדכנת את המשתנים הגלובליים בהתאם לזמן הנוכחי ומכינה אותם לקראת הודעת הדפסה מתאימה לקלט time.

קלט: אין. פלט: עדכון משתנים גלובליים ופלט במחסנית – 1 אם התבצעה פקודת TIME אחרת 0.

מקרא רגיסטרים:

r0 – מחזיק את המצביע לקלט הנוכחי.

מס"ד	Label	תפקיד
1	notTime	במידה ופקודת time שהוקלדה לא הייתה תקינה – מעודכן במחסנית 0 כלומר שלא התבצעה פקודת time

פונקציה 15: printTime - הפונקציה מדפיסה הודעה מתאימה לפקודה time.

פונקציה 16: aiMove – הפונקציה מנהלת את כלל החישובים עבור המחשב לצורך ביצוע מהלך אופטימאלי.

מס"ד	Label	תפקיד
1	aiMove	מאתחלת פרמטרים עבור מהלכי הרקורסיה ומגבה את הלוח המקורי לפני שהמחשב עושה "בלאגן" על הלוח המקורי.
2	aiLoop	לולאה שאחראית להגדיל את עומק הרקורסיה ב-2, בודקת האם נגמר הזמן או הגיעה פקודת stop
3	escape	מחזירה את המחסנית לקדמותה לפני כניסתה לרקורסיה

פונקציה 17: moveTime – הפונקציה תדפיס את כמות הזמן שלקח מהלך.

פונקציה 18: numToAsc – הפונקציה ממירה לערכי האסקי של מספר דצימלי

קלט: במחסנית (ערך לצורכי המרה, כתובת לשמירה ערכי האסקי)

פלט: בכתובת שניתנה.

פונקציה 19: ascToNum – הפונקציה ממירה ערכי האסקי למספר בערכו האוקטאלי

קלט: במחסנית (תא להחזרת ערך, כתובת של תווי האסקי)
פלט: במחסנית (בתא להחזרת הערך)
פונקציה 20: printf-

קלט: במחסנית ....msg1,msg2 , מספר המחרוזות שהוכנסו

פלט: ידפיס את כל המחרוזות שהוכנסו

מקרא רגיסטרים:

r0 – מחזיק את מספר המחרוזות שנותר להדפיס.

r1 – מצביע למחרוזת שמודפסת במחסנית.

פונקציה 21: printb – הפונקציה מדפיסה את לוח המשחק המקורי Board.

מקרא רגיסטרים:

r0 – מחזיק מצביע ללוח.

r1 – סופר להוספת שורה חדשה בלוח.

מס"ד	Label	תפקיד
1	printb	מגבה את ערכי הרגיסטרים ומדפיסה הודעות מוכנות מראש לקראת הדפסת הלוח
2	loopW_B	לייבל זה בודק האם יש לרדת שורה בהדפסה והאם אנחנו בגבולות הלוח
3	afterBr	תבדוק איזה תו יש להדפיס ותשלח אותו ללייבל הרלוונטי.
4	prW	ידפיס את האות "W" עבור שחקן לבן
5	prB	ידפיס את האות "B" עבור שחקן שחור
6	Pr_	ידפיס את התו "_" עבור משבצת ריקה
7	breakLi	לייבל זה ידפיס שורה חדשה כל 8 תווים שהודפסו
8	endPrBo	משחזר ערכי רגיסטרים וחוזר לקורא

פונקציה 22: clearInput – הפונקציה מנקה(מהקלט שהוכנס) את הלייבל של input.

פונקציה 23: printError – מדפיסה את הקלט שהוכנס ומציגה אותו כ"שגיאה". (עקב קלא לא תקין שהוכנס)

פונקציה 24: copyLabel – הפונקציה נועדה לצורך העתקת לייבל.

קלט: במחסנית: מספר התווים להעתקה, הלייבל יעד אליו נעתיק את התווים, לייבל מקור ממנו מעתיקים.
פלט: מעודכן בלייבל יעד.

פונקציה 25: startPrnt – הפונקציה מדפיסה את הודעה תחילת המשחק כאשר מתחילים משחק חדש.

פונקציה 26: clearLabel -פונקציה כללית למחיקת קלט שהוכנס לכל לייבל בתוכנית.

קלט: במחסנית את הכתובת של הלייבל שאנו רוצים למחוק לו את הנתונים. (הקלט שהוכנס לו)

פונקציה 27: copyBO – הפונקציה מעתיקה את הלוח המקורי ללוח יעד(חדש).

קלט: במחסנית – כתובת היעד של הלוח אליו נעתיק, כתובת המקור של הלוח אותו נרצה להעתיק לאחר.
פלט: מעדכן את לוח היעד.

מקרא רגיסטרים:

r0 – כתובת מקור של הלוח אותו מעתיקים.

r1 – כתובת היעד אליה נעתיק את הלוח מכתובת המקור.

טיפול בשעון:

מבוצע באמצעות partSec "חלקיקי שניה" שמוגדר באמצעות rate.

מס"ד	Label	תפקיד
1	clock	הלייבל מחסיר "חלקיק שניה" כל פעם שקוראת פסיקת שעון
2	tick	הלייבל מחסר שניה כל פעם ש"חלקיקי השניה" הגיעו ל 0
3	flagUp	הלייבל מרים את דגל timeEnded במידה והזמן נגמר, כדי שפונקציות אחרות יוכלו לבדוק זאת

<u>פונקציה 28 –</u> getCord – הפונקציה תחשב את הקורדינטות של src&dst מערכי האסקי שלהם.

קלט: משתנה גלובאלי moveLabel

פלט: מעדכנת את המשתנים src3 ו dst3