

## פרויקט גמר אסמבלי



### משחק כדורגל פיפ"א 18 אסמבלי

שם התלמיד: נעם כירם

מספר תעודת זהות: 322568718

כיתה : י"2

בית הספר: גימנסיה ריאלית ראשון לציון.

מורה מנחה: אלה לב

שנת לימוד: תשע"ז 2017

## תוכן עניינים

<u>נושא</u>	<u>עמוד</u>
מבוא	3
הוראות הפעלה	4
טבלת פרוצדורות	5
תרשים זרימה כללי של הפרויקט	8
תרשימי זרימה עיקריים של הפרוצדורות	9
צילומי מסך עיקריים בפרויקט	11
אלגוריתמים מיוחדים בפרויקט	12
סיכום	15
קודים	16

## **מבוא**

המשחק שבחרתי לעשות הוא כדורגל. במשחק זה השוער שנמצא על המסך בתור כוכבית צריך להדוף את הכדור מהשער 10 פעמים. כאשר הכדור נכנס לשער מופיעה הודעה: "goal! Your time is...", ויוצג משך המשחק בשניות. במידה והשחקן מצליח לבצע 10 הצלות תופיע תמונה שמראה על הניצחון.

בחרתי לעשות את עבודת בגמר על נושא זה כי אני אוהב כדורגל ומשחק במשחק פיפ"א שממנו לקחתי את הרעיון.

## הוראות הפעלה למשחק

בהתחלה יש להעתיק לתיקיית bin את שם הקובץ שדרכו ניתן להיכנס למשחק - "fifa18" ואת קבצי התמונות "open.bmp", "end.bmp", "done.bmp". לאחר הכניסה לדוס בוקס, יש להקליד את שם הקובץ. לאחר מכן מופיע מסך הפתיחה. יש ללחוץ על מקש מסוים במקלדת כדי להתחיל לשחק. כאשר השחקן והכדור מופיעים, השוער זז באמצעות מקשי החצים למעלה ולמטה.

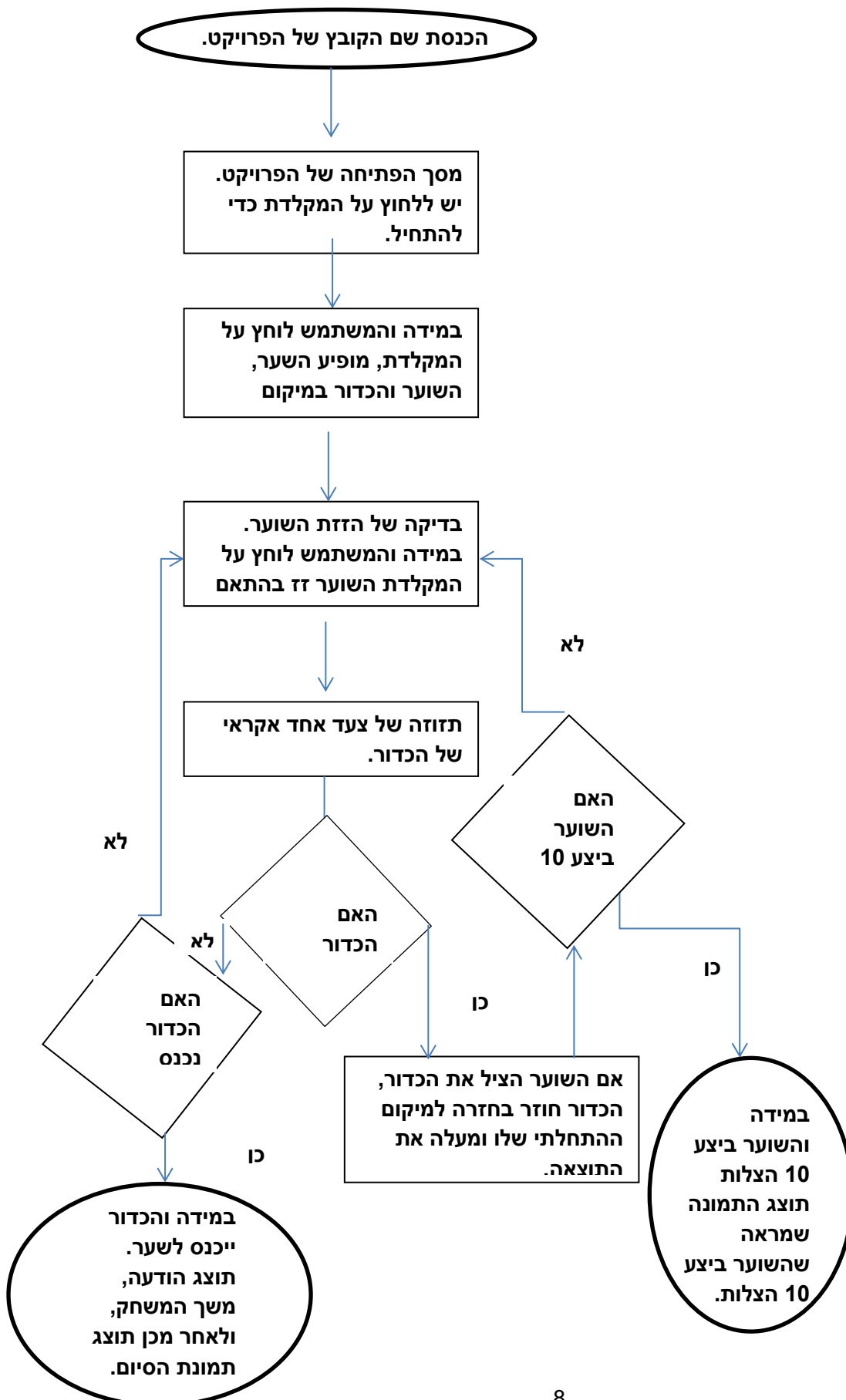
## טבלת פרוצדורות

מספר עמוד בקודים	ערכים שהפרוצדורה מחזירה	ערכים שהפרוצדורה מקבלת	תפקידה בתכנית	שם הפרוצדורה	
7	הפרוצדורה אינה מחזירה ערכים.	הפרוצדורה אינה מקבלת פרמטרים.	הפרוצדורה מדפיסה את השער בצד השמאלי של המסך.  bx - אחראי על הצבע של השער בתוך bh. cx - אחראי על מיקום הנקודה השמאלית העליונה של השער – שורה 0, עמודה מספר 72.  dx – מחזיק את הערך של הנקודה הימנית התחתונה של השער – שורה 25h טור מספר 80h.  לאחר מכן השתמשתי בפסיקה 10h.	setReka	1.
6	הפרוצדורה מחזירה את תוצאת החישוב למשתנה posGK המכיל את המיקום של השוער על המסך.	הפרוצדורה מקבלת את ערכי הטור והשורה הנוכחיים של השוער.	מטרת הפרוצדורה היא החישוב הנוכחי של השוער על המסך לפי מספר הטור ומספר השורה. החישוב מתבצע כך:  $(rowGK * 80 + colGK) * 2 = posGk.$	Calc_pos	2.
6-7	הפרוצדורה אינה מחזירה ערכים.	הפרוצדורה מקבלת את המשתנה posGk.	הפרוצדורה מדפיסה את השוער לפי הערך של posGk המצבי על המיקום של השוער במסך.	putGK	3.
8-9	הפרוצדורה אינה מחזירה ערכים.	הפרוצדורה אינה מקבלת פרמטרים.	הפרוצדורה מדפיסה מיקום התחלתי של הכדור בשורה רנדומלית שערכו נקבע לפי הפרוצדורה rand_row ובטור מספר 38 (מספר הטור נקבע באופן שרירותי).	putBall	4.
11-12	הפרוצדורה מחזירה את ערכה של השורה ב dl.	הפרוצדורה אינה מקבלת פרמטרים.	הפרוצדורה מחשבת מספר שורה אקראי לפי דגימת שעון לפי הפסיקה  mov ah,2ch int 21h.	Rand_row	5.
10-11	הפרוצדורה אינה מחזירה ערכים.	הפרוצדורה מקבלת את המשתנה count המכיל את התוצאה הנוכחית. ערכו ההתחלתי הוא אפס.	מדפיסה את מספר ההצלחות הנוכחי.	priScore	6.

14-15	הפרוצדורה אינה מחזירה ערכים.	הפרוצדורה אינה מקבלת פרמטרים.	<p>הפעולה קולטת מהמקלדת לפי הפסיקה :</p> <pre>mov ah, 1 int 16h.</pre> <p>לאחר מכן הפעולה מזיזה את השוער בהתאם ללחיצות המשתמש.</p>	move	7.
13-14	הפרוצדורה אינה מחזירה ערכים.	הפרוצדורה אינה מקבלת פרמטרים.	<p>הפעולה אחראית על תזוזת הכדור במסך.</p> <p>הפעולה מגרילה כיוון באמצעות הפרוצדורה randStep שאליו הכדור ינוע (ימינה, שמאלה, או למעלה) ולאחר מכן הכדור יזוז לפי הכיוון שהוגרל.</p>	moveBall	8.
12	הפרוצדורה מחזירה את הערך של הצעד האקראי שנוצר מהחישוב ב dl מספר בין 1-3.	הפרוצדורה אינה מקבלת פרמטרים.	<p>הפרוצדורה מחשבת מספר צעד אקראי לפי דגימת שוען באמצעות הפסיקה</p> <pre>mov ah,2ch int 21h.</pre>	randStep	9.
16	הפרוצדורה מחזירה ערך במשתנה saveOK : אם יש הצלה של השוער ערכו של המשתנה יהיה 1 אחרת יהיה 0.	הפרוצדורה מקבלת את מספר הטור והשורה של השוער ושל הכדור.	הפרוצדורה בודקת האם השוער והכדור נמצאים באותה שורה ובאותו טור.	checkSave	10.
9-10	הפרוצדורה אינה מחזירה ערכים.	הפרוצדורה מקבלת את תוצאת החישוב של משך המשחק.	הפרוצדורה מדפיסה את תוצאת החישוב של משך המשחק. הפרוצדורה מזמנת כל ספרה לפרוצדורה printCharacter.	printNumber	11.
10	הפרוצדורה אינה מחזירה ערכים.	הפרוצדורה מקבלת כל פעם ספרה אחת של תוצאת החישוב ומדפיסה אותה.	הפרוצדורה מדפיסה כל פעם ספרה באמצעות הפסיקה	printCharacter	12.
11	הפרוצדורה אינה מחזירה ערכים.	הפרוצדורה אינה מקבלת פרמטרים.	<pre>mov dl, al int 21h</pre> <p>הפרוצדורה מדפיסה את ההודעה על המסך שהכדור נכנס לשער.</p>	priGoal	13.
12-13	הפרוצדורה מחזירה את תוצאת החישוב למשתנה pos המכיל את המיקום של ההכדור על המסך.	הפרוצדורה מקבלת את ערכי הטור והשורה של הנוכחיים של הכדור.	<p>מטרת הפרוצדורה היא החישוב הנוכחי של הכדור על המסך לפי מספר הטור ומספר השורה.</p> <p>החישוב מתבצע כך :</p> $(row * 80 + col) * 2 = pos$	calc_posStep	14.
5-6	הפרוצדורה אינה מחזירה ערכים.	הפרוצדורה אינה מקבלת פרמטרים.	מבצעת עיכוב של כשנייה במשחק.	delay	15.

2	הפרוצדורה אינה מחזירה ערכים.	הפרוצדורה מקבלת את שם הקובץ ברגיסטר dx.	הפרוצדורה מדפיסה את התמונה המבוקשת בהתאם לשלב במשחק.	printBMP	.16
	הפרוצדורה אינה מחזירה ערכים.	הפרוצדורה מקבלת את סכום השניות בתחילת המשחק.	הפרוצדורה דוגמת את הזמן בסוף המשחק באמצעות הפסיקה mov ah,2ch int 21h. מחסרת מזמן השניות המקורי ומדפיסה על המסך.  הנוסחה לחישוב הזמן הכולל: total time = (finMin*60+fin sec) - (startMin*60+startSec).	calcTime	.17
2	מחזיר את המצביע על הקובץ (handle).	שם הקובץ.	פותח את קובץ bmp לקריאה.	OpenBMP	.18
3	מכניס את מבנה הצבעים לזיכרון.	המצביע על הקובץ (handle).	קורא את כותרת bmp.	ReadHeader	.19
3-4	מכניס את מבנה הצבעים לזיכרון.	המצביע על הקובץ (handle).	קורא את מבנה הצבעים מהקובץ.	ReadPalette	.20
4	הפרוצדורה אינה מחזירה ערכים.	מצביע על הזיכרון של מבנה הצבעים.	מעתיק את מבנה הצבעים מהזיכרון הכללי לזיכרון הגרפי.	CopyPal	.21
5	הפרוצדורה אינה מחזירה ערכים.	הפרוצדורה מקבלת את התמונה לזיכרון הגרפי מהסוף להתחלה.	מעתיק את התמונה לזיכרון הגרפי מהסוף להתחלה.	CopyBitmap	.22

## תרשים זרימה כללי של הפרויקט



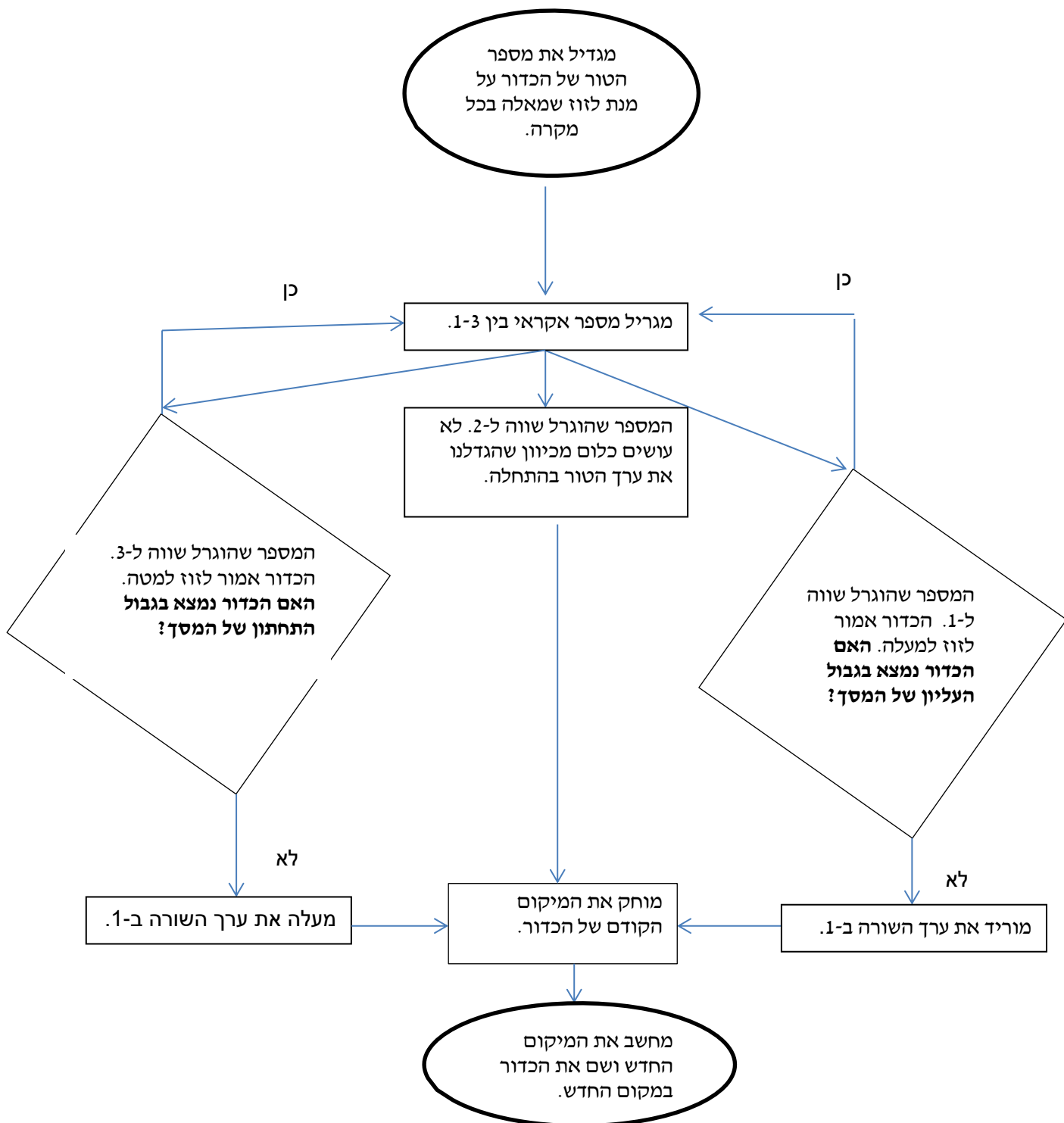


## תרשימי זרימה עיקריים של הפרוצדורות

**moveBall – אחראית כל פעם על צעד אקראי אחד של הכדור.**

טענת כניסה - הפרוצדורה אינה מקבלת פרמטרים.

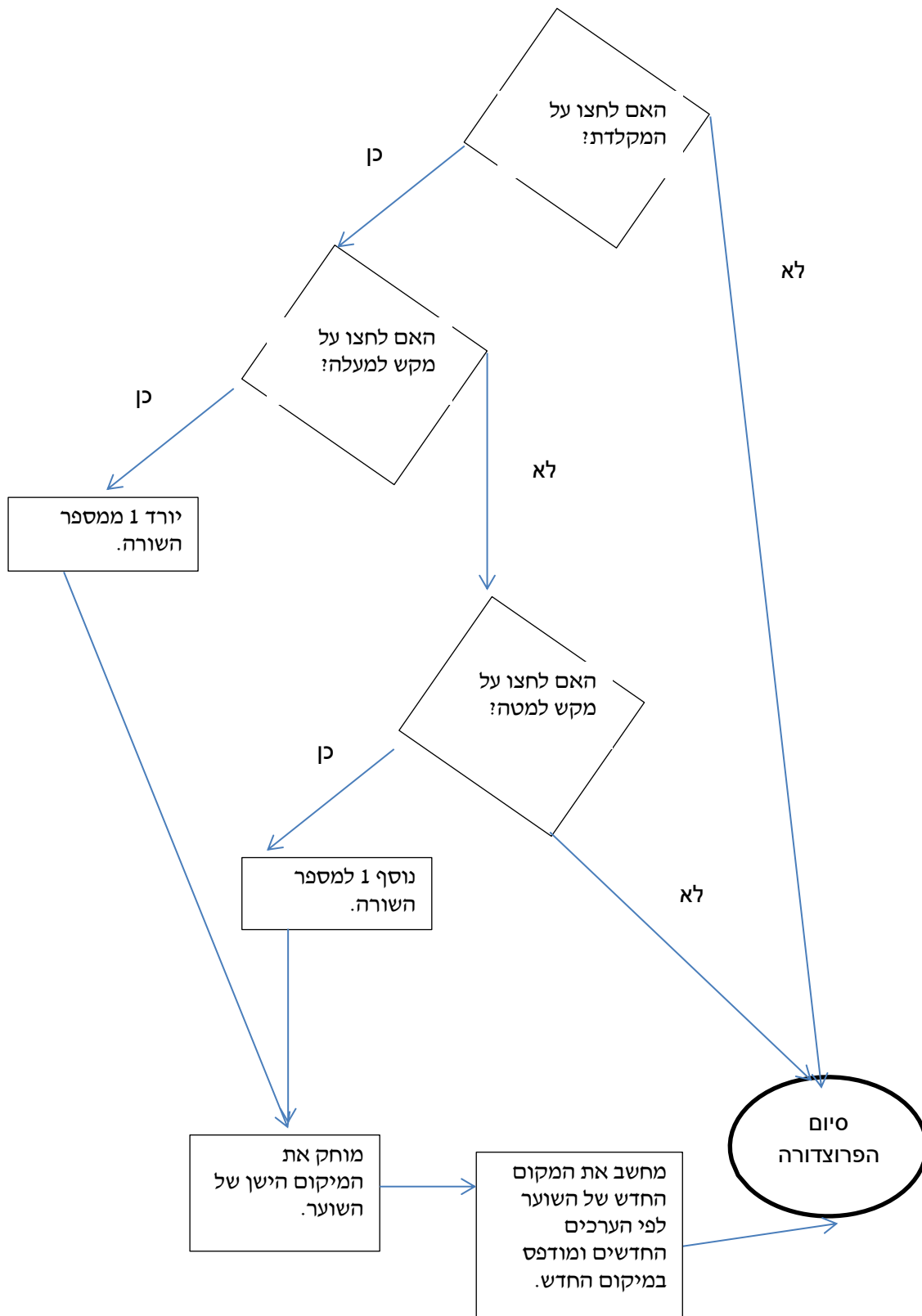
טענת יציאה - הפרוצדורה אינה מחזירה ערכים.



move – אחראית כל פעם על תזוזת השוער לפי המקשים.

טענת כניסה - הפרוצדורה אינה מקבלת פרמטרים.

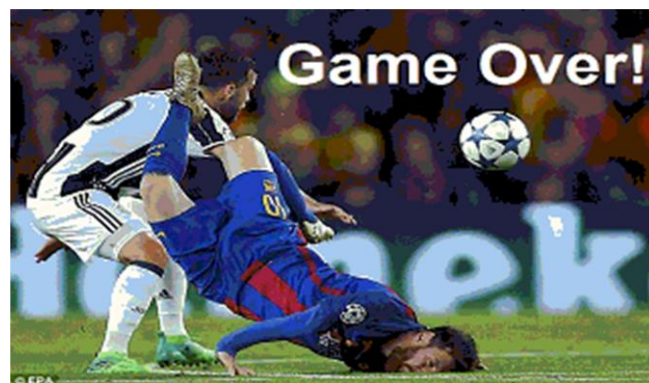
טענת יציאה- הפרוצדורה אינה מחזירה ערכים.



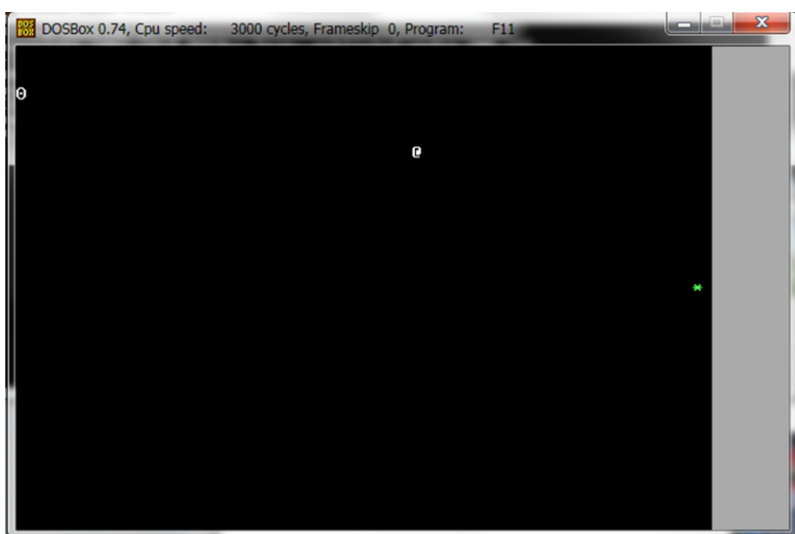
## צילומי מסך עיקריים בפרויקט



תמונת הפתיחה של המשחק



תמונת הסיום של המשחק  
במידה והשחקן נפסל



תמונה של השחקן, הכדור, והשער. בפינה  
השמאלית העליונה כמות ההצלות של השוער



תמונת הסיום של המשחק  
במידה והשחקן מנצח.

## אלגוריתמים מיוחדים בפרויקט

בחלק זה הייתי רוצה להסביר על קטע קוד מהתכנית : התנועה הרנדומלית של הכדור (הפרוצדורה שאחרית על קטע זה היא הפרוצדורה moveBall).

proc moveBall

push dx

push cx

push bx

push ax

inc [col]

cmp [col],66d

jae right

rnd:

call randStep

cmp dl,1d

je up

cmp dl,2d

je right

cmp dl,3d

je down

right:

jmp contin ;goes just right

down:

cmp [row],24d

je rnd

inc [row] ;go down

jmp contin

up:

cmp [row],0

je rnd

dec [row]

כאשר הכדור מגיע לשלושת הצעדים האחרונים שלו הוא זז רק ישר כדי להקל על המשחק.

מוחזר :

1 = למעלה וימינה.  
2 = רק ימינה.  
3 = למטה וימינה.

הסבר על הפרוצדורה בעמוד 13.

בודק האם הכדור נמצא בגבול

מגריל שוב אם כן

בודק האם הכדור נמצא בגבול העליון של המסך.

מגריל שוב אם כן

המשך בעמוד הבא.

contin:

```
push di  
mov di,[pos]  
call clear_star  
pop di
```

מוחק את  
המיקום  
של  
הכדור.

```
mov ah, 0fh  
mov al, '@'  
call calc_posStep
```

שם את  
הכדור  
במיקום  
החדש.

```
push di  
mov di,[pos]  
mov [es:di], ax  
pop di
```

```
pop ax  
pop bx  
pop cx  
pop dx
```

```
ret
```

```
endp moveBall
```

## הסבר על הפרוצדורה call randStep

proc randStep

push ax

push bx

push cx

retu:

mov ah,2ch

int 21h

and dl,0011b ;change the num between 0-3 ; the min power of the 2 that over 3

cmp dl,0 ; when the rand num = 0 rand again

je retu

pop cx

pop bx

pop ax

ret

endp randStep

הפרוצדורה מגרילה מספר אקראי באמצעות דגימת שעות. לאחר מכן הפרוצדורה עושה פעולת and על dl המודד את ערכן של 55 מילי שניות בזמן הדגימה. פעולה חשבונית זו משנה את ערכו של dl בהתאם לערכו הקודם, לאחר מכן הפעולה תחזיר את ערכו של dl.

## **סיכום**

בתהליך בניית העבודה למדתי שעם עבודה קשה, התמדה, וחלוקה נכונה של בניית העבודה אפשר לעשות הכל. בנוסף למדתי איך לדבג קוד על מנת למצוא טעויות. כלי זה מאד חשוב להמשך בעולם התכנות.

ברצוני להודות להצלחת העבודה לאלה לב, שנתנה לי רעיונות חדשים כל פעם, שלימדה אותי איך לבנות פרויקט לפי שלבים ואיך ללמוד לבד, לאבא שלי גיא על ההערות המועילות, ולדויד, תומך ההוראה, על התמיכה ועל העידוד.

קוד