המשחק

# השהייה והמקלדת

## bg9

## "תזוזה" של השחקן

עד עתה הזזנו את השחקן על בלוח המשחק בהקשה של מקש.

הלולאה שיצרנו, הזיזה את השחקן בכל הקשה על מקש.

לולאה:

מחיקת השחקן

שינוי הערכים של הסמן

מיקום הסמן

ציור הסמן במיקום החדש

המתנה להקשה של מקש במקלדת.

עתה נרצה שהשחקן שלנו יזוז באופן עצמאי על מסך.

לכן נבטל את הפקודה של המתנה למקש.

**לולאה:**

מחיקת השחקן

שינוי הערכים של הסמן

הבעיה בתכנית זו, שהמחשב עובד מאוד מהר ולכן לא נצליח לראות את השחקן שלנו זז על לוח המשחק.

מיקום הסמן

ציור השחקן במיקום החדש

נוסיף פעולה של השהייה ונזמן אותה בכל איטרציה (סיבוב) של הלולאה.

* proc delay
* pusha
* mov cx, 03h ;High Word

פסיקה זו ממתינה חמישית שניה.

משהה את תזוזת השחקן ב – 200 מאיות השנייה.

* mov dx, 4240h ;Low Word
* mov al, 0
* mov ah, 86h ;Wait
* int 15h
* popa
* ret
* endp delay
* **INT 15h** / **AH = 86h** - BIOS wait function.
* **CX:DX** = interval in microseconds
* Note: 1 million in hex is '000f 4240',
* so CX = 0fh : DX = 4240h = 1 second
* **CX = 3h : DX = 4240H = 1/5 second**

## תזוזה של השחקן

עתה נרצה שהשחקן שלנו יזוז באופן עצמאי על מסך.

לכן נבטל את הפקודה של המתנה למקש.

**לולאה:**

מחיקת השחקן

שינוי הערכים של הסמן

מיקום הסמן

ציור הסמן במיקום החדש

השהייה (המתנה של 200 מאיות השנייה)

## המקלדת - Scan Codes

כל לחיצה או שחרור של מקש במקלדת מייצרים קוד מיוחד – scan code.

המספר שנוצר נשמר בזיכרון מיוחד של המעבד שנקרא זיכרון I/O (input/output)

* המשתמש לוחץ או משחרר מקש
* בזיכרון ה I/O נכתב ה **scan code** המתאים
* ה־PIC מקבל אינטרפט מהמקלדת.
* ה־PIC שולח למעבד אינטרפט (int 9), שאומר למעבד שיש מידע בזיכרון ה I/O של המקלדת
* כתגובה לפסיקה, המעבד מריץ ISR שמטפל בפסיקה 9. ה־ISR "מעתיק" ה־code scan אל מקום מוגדר בזיכרון שנקרא

**Type Ahead Buffer**

PIC - Peripheral Interface Controller – בקר ממשק "היקפי"

## Type Ahead Buffer

מקום בזיכרון בו המעבד שומר את המידע מהמקלדת.

עבור כל תו המעבד שומר את ה Scan Code ואת התרגום שלו ל ASCII

כלומר לכל תו נשמרים 2 בתים

גודל הבאפר הוא 32 בתים. כלומר יש מקום ל 16 תווים.

לכן עלינו לדאוג לנקות את הבאפר

## מהמשתמש למעבד

## קריאת תו מהמקלדת

עד עכשיו עבדנו עם int 21h   
כאשר **ah = 0**, התוכנית הייתה עוצרת ומחכה שנקליד תו, לא תמיד זה נוח.

בנוסף לא תמיד נוח שהתו שהקשנו יופיע על המסך

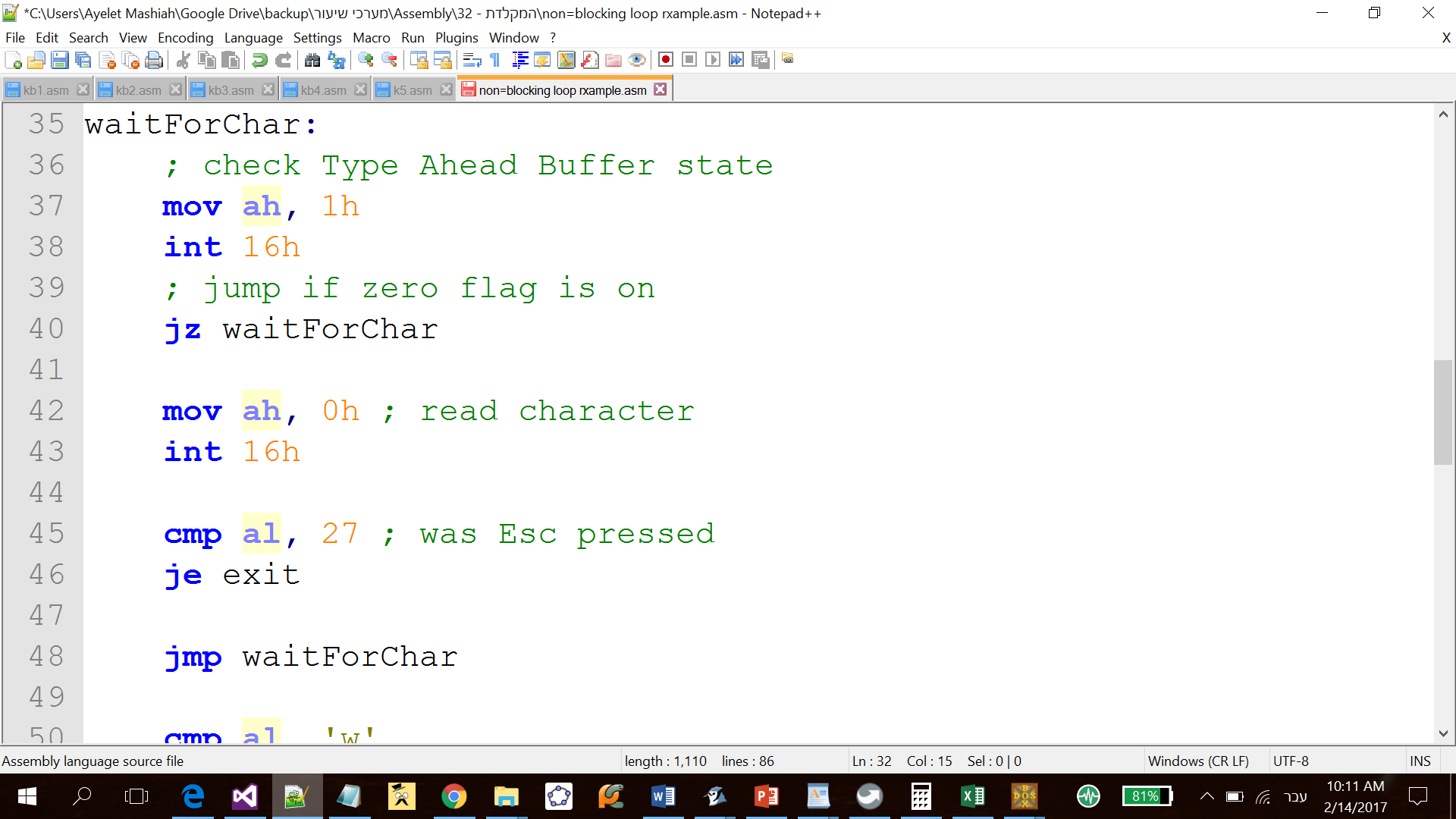
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **חוסמת?** | **תו מודפס** | **תיאור** | **סוג** | **פסיקה** |
| כן | לא | הפסיקה מחזירה בתוךal את קוד ה־ ASCII של התו שנמצא בראש הבאפר ובתוךah את ה־code scan שלו.  בנוסף, הפסיקה "מנקה" את התו מהבאפר | BIOS | mov ah, 0h  int 16h |
| לא | לא | הפסיקה מדליקה (מאפסת) את ה zero flag– **אם יש תו מוכן לקריאה (zf = 0) ,** **ומכבה אם אין תו מוכן (zf = 1).**  אם יש תו מוכן, al ו-ah יקבלו את ערכי ה־ASCII וה־code scan של התו. הפסיקה אינה "מנקה" את התו מהבאפר | BIOS | mov ah, 1h  int 16h |
| כן | לא | מנקה את הבאפר של המקלדת, ואז לוקחת הערך ששמנו ב־al כפרמטר, ומריצה h21 int עם הקוד הזה. לדוגמה אם al = 7h לאחר ניקוי באפר המקלדת, הפסיקה תעבור לקוד 7h, שהוא קוד של קליטת תו מהמקלדת ללא הדפסת התו על המסך. | Dos | mov ah,0Ch  Int 21h |

## איך נבדוק אם דגל ה 0 דולק

**jz label**

קפוץ אם דגל ה 0 דולק (zf = 1) ⇦ יש תו מוכן לקריאה

## דוגמה:



* בדיקת מצב הבאפר בלולאה
* קריאת תו מהבאפר
* אם נלחץ Esc קפוץ ליציאה
* קפוץ לתחילת הלולאה

## בדיקה האם הקישו על מקש במקלדת

; check if thre is a charcter to read

mov [chr], 0

mov ah, 1h

int 16h

**jz** noKey

; waits for character

call readChr

…..

noKey:

נרצה להוסיף למשחק שלנו הוראה הבודקת אם הקישו על מקש במקלדת?

אבל אם לא הקישו על מקש, שהשחקן שלנו ימשיך לזוז על פני לוח המשחק.

לצורך הבדיקה האם הקשנו על מקש נשתמש בפסיקה שאינה מקלידה את התו ואינה ממתינה למקש כדי להמשיך לפקודות הבאות.

אם הקישו על מקש דגל האפס = 1

jz lable

קפוץ אם דגל ה – 0 דולק (zf=1)

mov cx, 10

**mainGameLoop:**

; draw blank - ascii 0 at cursor position

mov [color], 0

call drawCharacter

; increase x\_coord by 1

inc [x\_coord]

add [y\_coord], 2

mov [color], 0Eh

; set cursore location

; draw smiley - ascii 2 at cursor position

call setCursorePosition

call drawCharacter

; check if thre is a charcter to read

בדיקה האם הוקש מקש,

אם כן בודקים האם ‘q’

אחרת קופצים לתווית שלא הוקש מקש

השהייה

והפקודה לסיום איטרציה של הלולאה

mov [chr], 0

mov ah, 1h

int 16h

jz noKey

; waits for character

call readChr

; check if user asks to quit

cmp [chr], 'q'

je end\_game

noKey:

call delay

loop mainGameLoop

end\_game:

; text mode

mov ax, 2h

int 10h

## הדמות קופצת בהקשה על מקש רווח

נוסף פקודה שבהקשה על מקש הרווח הדמות שלנו תקפוץ למעלה

כדי לקפוץ למעלה ⇦ צריך **להפחית** ממיקום הדמות על ציר ה –y

לפני הכניסה ללולאה נאפס את המשתנה השומר את התו שהוקלד במקלדת.

mov [chr], 0

לאחר המחיקה של השחקן

נוסיף בדיקה האם הקשנו על מקש הרווח.

אם כן נקפוץ לתווית – jmpChr

נוסף למיקום השחקן על ציר ה – y 2 (שימשיך לזוז כלפי מטה)

ונקפוץ לתווית שמדלגת על הפקודות המחסירות מהמיקום של ציר y -2  
(לצורך "קפיצת השחקן").

נמקם את הסמן

נצייר את השחקן.

mainGameLoop:

; draw blank - ascii 0 at cursor position

mov [color], 0

call drawCharacter

; increase x\_coord by 1

inc [x\_coord]

**cmp [chr], ' '**

אם הקשנו על מקש **רווח**

במקום לרדת שני שורות עם השחקן, נקפוץ למעלה שתי שורות.

**je jmpChr**

add [y\_coord], 2

jmp continue

**jmpChr:**

**sub [y\_coord], 2**

continue:

; set cursor location & draw smiley - ascii 2 at cursor position

mov [color], 0Eh

call setCursorePosition

call drawCharacter

**; check if there is a character to read**

**mov [chr], 0**

איפס משתנה המקבל את התו במקלדת [chr]

אם יש הקשה ⇦ zf = 1

jz = האם דגל האפס דולק

**mov ah, 1h**

**int 16h**

**jz noKey**

; waits for character

call readChr

; check if user asks to quit

cmp [chr], 'q'

je end\_game

noKey:

call delay

loop mainGameLoop

end\_game:

; text mode

mov ax, 2h

int 10h

## הפסיקות בתכנית שלנו

|  |  |
| --- | --- |
| פקודה להדפסת מחרוזת  (שם משתנה המחרוזת msg) | mov dx, offset msg  mov ah, 9h  int 21h |
| פקודה לקליטת תו מהמקלדת  לא מדפיסה את תו ולא ממשיכה בהוראות על לקבלת תו. | mov ah, 0h  int 16h  הפסיקה מחזירה בתוךal את קוד ה־ ASCII של התו שנמצא בראש הבאפר ובתוךah את ה־code scan שלו. בנוסף, הפסיקה "מנקה" את התו מהבאפר |
| פקודה לקליטת תו מהמקלדת  לא מדפיסה את תו, אם לא הוקש על מקש ממשיכה בפקודות התכנית. | mov ah, 1h  int 16h  הפסיקה מדליקה את ה zero flag– אם יש תו מוכן לקריאה (zf = 0) , ומכבה אם אין תו מוכן (zf = 1).  אם יש תו מוכן, al ו-ah יקבלו את ערכי ה־ASCII וה־code scan של התו. הפסיקה אינה "מנקה" את התו מהבאפר |
| כניסה לתצוגה גרפית  25\*40 | mov ax, 13h  int 10h |
| יציאה מתצוגה גרפית  80\*25 | mov ax, 2h  int 10h |
| מיקום הסמן על המסך | mov dh, [y\_coord] ; row  mov dl, [x\_coord] ; column  mov bh, 0 ; page number  mov ah, 2  int 10h |
| ציור תו על המסך במיקום הסמן | mov ah, 9  mov al, 2 ; al = character to display  mov bx, [color] ; bh = Background bl = Foreground  mov cx, 1 ; cx = number of times to write character  int 10h |
| השהייה  המתנה של פרק זמן קצר (חמישית השנייה) | mov cx, 03h ;High Word  mov dx, 4240h ;Low Word  mov al, 0  mov ah, 86h ;Wait  int 15h |

**INT 16h / AH = 00h** - get keystroke from keyboard (no echo).

return:

AH = BIOS scan code.

AL = ASCII character.

(if a keystroke is present, it is removed from the keyboard buffer).

**INT 16h / AH = 01h** - check for keystroke in the keyboard buffer.

return:

ZF = 1 if keystroke is not available. ⇨ לא הוקש מקש

ZF = 0 if keystroke available. ⇨ הוקש מקש

AH = BIOS scan code.

AL = ASCII character.

(if a keystroke is present, it is not removed from the keyboard buffer

Back to Top).

**INT 15h** / **AH = 86h** - BIOS wait function.

**CX:DX** = interval in microseconds

Note: 1 million in hex is '000f 4240',

so CX = 0fh : DX = 4240h = 1 second

**CX = 3h : DX = 4240H = 1/5 second**

תרגול

# המשחק

## השהייה והמקלדת

שמרו את התכנית שכתבתם בשיעור קודם בשם חדש והמשיכו לעבוד בקובץ החדש.

בתרגול זה נגרום לשחקן שלנו לנוע כלפי מטה על לוח המשחק,

בהקשה על מקש הרווח השחקן שלנו יקפוץ למעלה.

## הפסיקות בתכנית שלנו

|  |  |
| --- | --- |
| פקודה להדפסת מחרוזת  (שם משתנה המחרוזת msg) | mov dx, offset msg  mov ah, 9h  int 21h |
| פקודה לקליטת תו מהמקלדת  לא מדפיסה את תו ולא ממשיכה בהוראות על לקבלת תו. | mov ah, 0h  int 16h  הפסיקה מחזירה בתוךal את קוד ה־ ASCII של התו שנמצא בראש הבאפר ובתוךah את ה־code scan שלו. בנוסף, הפסיקה "מנקה" את התו מהבאפר |
| פקודה לקליטת תו מהמקלדת  לא מדפיסה את תו, אם לא הוקש על מקש ממשיכה בפקודות התכנית. | mov ah, 1h  int 16h  הפסיקה מדליקה את ה zero flag– אם יש תו מוכן לקריאה (zf = 1) , ומכבה אם אין תו מוכן (zf = 0).  אם יש תו מוכן, al ו-ah יקבלו את ערכי ה־ASCII וה־code scan של התו. הפסיקה אינה "מנקה" את התו מהבאפר |
| כניסה לתצוגה גרפית  25\*40 | mov ax, 13h  int 10h |
| יציאה מתצוגה גרפית  80\*25 | mov ax, 2h  int 10h |
| מיקום הסמן על המסך | mov dh, [y\_coord] ; row  mov dl, [x\_coord] ; column  mov bh, 0 ; page number  mov ah, 2  int 10h |
| ציור תו על המסך במיקום הסמן | mov ah, 9  mov al, 2 ; al = character to display  mov bx, [color] ; bh = Background bl = Foreground  mov cx, 1 ; cx = number of times to write character  int 10h |
| השהייה  המתנה של פרק זמן קצר (חמישית השנייה) | mov cx, 03h ;High Word  mov dx, 4240h ;Low Word  mov al, 0  mov ah, 86h ;Wait  int 15h |

## תנועת השחקן ללא מקשים

* בדקו בתחילת התכנית שהערכים של השחקן בתחילת המשחק הם:

על ציר ה –x: 7

על ציר ה - y: 6

x\_coord db 7 ; column

y\_coord db 6 ; row

* השאירו רק לולאה אחת במשחק,

בדקו שהשחקן שלנו זז בכל איטרציה של הלולאה ימינה במיקום אחד ולמטה שני מיקומים.

add [y\_coord], 2

inc [x\_coord]

## 1. עתה נרצה שהשחקן שלנו יזוז באופן עצמאי על מסך.

לכן נבטל את הפקודה של המתנה למקש בלולאה הראשית של המשחק.

**לולאה:**

מחיקת השחקן

שינוי הערכים של הסמן

הבעיה בתכנית זו, שהמחשב עובד מאוד מהר ולכן לא נצליח לראות את השחקן שלנו זז על לוח המשחק.

מיקום הסמן

ציור הסמן במיקום החדש

נוסיף פעולה של השהייה ונזמן אותה בכל איטרציה (סיבוב) של הלולאה.

* proc delay
* pusha
* mov cx, 03h ;High Word

פסיקה זו ממתינה חמישית שניה.

משהה את תזוזת השחקן ב – 200 מאיות השנייה.

* mov dx, 4240h ;Low Word
* mov al, 0
* mov ah, 86h ;Wait
* int 15h
* popa
* ret
* endp delay

נזמן פעולה זו לפני הפקודה של סיום איטרציה (סיבוב)של הלולאה.

הריצו ובדקו את התוצאה.

## 2. קריאת תו מהמקלדת

עד עכשיו עבדנו עם int 21h   
כאשר **ah = 0**, התוכנית הייתה עוצרת ומחכה שנקליד תו, לא תמיד זה נוח.

אנו רוצים ליצור מצב בו התכנית ממתינה למקש אך אם לא הקשנו עליו היא ממשיכה בפקודות האחרות, גם אם לא הקשנו על מקש הדמות שלנו תמשיך "ליפול" מטה.

**הפסיקה:**

mov ah, 1h

int 16h

לא מדפיסה את התו, ואם לא הוקש על מקש ממשיכה בפקודות התכנית.

הפסיקה מדליקה את ה zero flag– אם יש תו מוכן לקריאה (zf = 1) ,

ומכבה אם אין תו מוכן (zf = 0).

אם יש תו מוכן, al ו-ah יקבלו את ערכי ה־ASCII וה־code scan של התו.

הפסיקה אינה "מנקה" את התו מהבאפר

**jump zero = האם דגל האפס כבוי**

**נוסיף פסיקה זו לתכנית ונמקם אותה**

**לאחר ציור הדמות במיקום החדש:**

אלגוריתם:

לולאה – mainGameLoop

"מחיקת" שחקן

הגדלת מיקום השחקן על ציר x ועל ציר ה - y

מיקום סמן

ציור שחקן במיקום החדש בצבע

בדיקה האם הוקש מקש

אם כבוי קפיצה לתווית – noKey

קריאה לפעולה הבודקת מהו המקש

בדיקה האם 'q'

אם 'q' קפיצה לתווית – end\_game

תווית **noKey**

זימון השהייה

פקודת סיום איטרציה של הלולאה

תווית - end\_game

יציאה מתצוגה גראפית.

**; check if thre is a charcter to read**

**mov [chr], 0**

**mov ah, 1h**

**int 16h**

**jz noKey**

; waits for character

call readChr

; check if user asks to quit

cmp [chr], 'q'

je end\_game

**noKey:**

call delay

loop mainGameLoop

end\_game:

הריצו ובדקו: האם השחקן "נופל" מטה והאם בהקשה על q יוצאים מהתכנית.

## 3. הדמות קופצת בהקשה על מקש רווח

נוסף פקודה שבהקשה על מקש הרווח הדמות שלנו תקפוץ למעלה

כדי לקפוץ למעלה ⇦ צריך **להפחית** ממיקום הדמות על ציר ה –y

לפני הכניסה ללולאה נאפס את המשתנה השומר את התו שהוקלד במקלדת.

mov [chr], 0

לאחר המחיקה של השחקן

נוסיף בדיקה האם הקשנו על מקש הרווח.

אלגוריתם:

לולאה – mainGameLoop

"מחיקת" שחקן

הגדלת מיקום השחקן על ציר x

בדיקה האם מקש שהקשנו [chr] = ' ' (שווה לרווח)

קפיצה לתווית - **jmpChr**

הגדלת מיקום השחקן על ציר y

קפיצה לתווית – **continue**

**תווית: jmpChr**

הפחתה של מיקום השחקן על ציר y

**תווית: continue**

מיקום סמן

ציור שחקן במיקום החדש

בדיקה האם הוקש מקש

אם כבוי קפיצה לתווית – noKey

קריאה לפעולה הבודקת מהו המקש

בדיקה האם 'q'

אם 'q' קפיצה לתווית – end\_game

תווית **noKey**

זימון השהייה

פקודת סיום איטרציה של הלולאה

תווית - end\_game

יציאה מתצוגה גראפית.

mainGameLoop:

; draw blank - ascii 0 at cursor position

mov [color], 0

call drawCharacter

increase x\_cord by 1

inc [x\_cord]

mov [color], 0Eh

**cmp [chr], ' '**

**je jmpChr**

add [y\_cord], 2

**jmp continue**

**jmpChr:**

**sub [y\_cord], 2**

**continue:**

; set cursor location

; draw smiley - ascii 2 at cursor position

call setCursorePosition

call drawCharacter

; check if there is a charcter to read

**noKey**:

call delay

loop mainGameLoop

end\_game:

; text mode

mov ax, 2h

int 10h

mov [chr], 0

mov ah, 1h

int 16h

**jz noKey**

;

waits for character

call readChr

; check if user asks to quit

cmp [chr], 'q'

הריצו ובדקו: האם השחקן "נופל" מטה והאם בהקשה על מקש רווח הדמות קופצת מעלה.

je end\_game