

Assembler project fire boy & water girl

שם: נוגה שמש

תעודת זהות: 325763373

מורה: אנטולי פיימר

כיתה: י'3

בית ספר: אורט נעמי שמר

תוכן עניינים:

מבוא.....	3	▶
דרישות מערכת.....	3	▶
נושא העבודה.....	4	▶
הפעלת המשחק.....	4	▶
חוקים+ אופן המשחק.....	5	▶
אופציות לשדרוג.....	6	▶
הסברת לוגיקת המשחק.....	7	▶
תרשימי זרימה על פעולות מרכזיות.....	8	▶
הסבר כללי על הפעולות.....	15	▶
דוגמאות הרצה.....	30	▶
סיכום.....	33	▶

מבוא

- ▶ שם העבודה: fire boy & water girl
- ▶ שם הקובץ: fireboywatergirl.exe
- ▶ קבצים נלווים: macro.asm, charac1.dat, image.dat, board
- ▶ סביבת עבודה: Turbo Assembler
- ▶ סביבת פיתוח: ++Notepad
- ▶ סביבת הרצה: DosBox
- ▶ דרישות מערכת:
- ▶ מערכת הפעלה: כל מערכת הפעלה שאפשר להריץ בה DosBox
- ▶ האמולטור DosBox
- ▶ מעבד 386

נושא העבודה + הפעלת המשחק

▶ העבודה שלי היא משחק הנקרא: בן האש ובת המים. בחרתי במשחק זה מפני שכשהייתי יותר קטנה אהבתי מאוד לשחק בו עם בני דודים שלי. נוסף על כך, הרגשתי שתכנות משחק זה יגרום לי לאתגר ורציתי לאתגר את עצמי.

▶ אחרי הרצת המשחק יופיע תפריט. בתפריט יש שלוש אופציות:

1. משחק –נפתח חלון המשחק

2. עזרה – במידה ובחרת בעזרה, יופיעו חוקי המשחק והסבר.

3. יציאה – יוצא מהמשחק.

חוקים + אופן המשחק

- ▶ במשחק יש שני שחקנים, בן האש – שחקן בצבע אדום. בת המים – שחקנית בצבע כחול. המטרה של המשחק הוא ששני השחקנים יעברו את המסלול ויגיעו כל אחד אל הדלת הסופית שלו.
- ▶ בן האש זז באמצעות החצים. חץ ימין כדי לזוז ימינה, חץ שמאלה כדי לזוז שמאלה וחץ למעלה בשביל לקפוץ. בזמן הקפיצה ניתן לזוז גם לשני הצדדים. לבן האש אסור לגעת במכשול הכחול ובמכשול הירוק.
- ▶ בת האש זזה באמצעות אותיות. D כדי לזוז ימינה, A כדי לזוז שמאלה, W כדי לקפוץ. לבת האש אסור לגעת במכשול האדום ובמכשול הירוק.

אופציות לשדרוג

במידה והיה לי עוד זמן הייתי מוסיפה למשחק: ►

1. טיימר, הצגה של הזמן שלוקח לעבור את המסלול.

2. שלבים נוספים עם מכשולים קשים יותר.

3. צלילים

הסברת לוגיקת המשחק

► לוח המשחק הוא מתמונת pcx ולכן הבעיה הראשונה שהייתי צריכה לפתור הייתה שבזמן הזזת השחקנים הן לא ימחקו את הרקע. לשם כך, כל פעם ששחקן זז התבצעו הפעולות הבאות:

1. ציור הרקע שהיה מאחורי השחקן במקום הישן.

2. שמירת הרקע מאחורי שני השחקנים במיקומם הנוכחי.

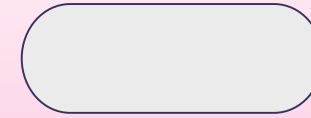
3. שמירת המקום הנוכחי.

4. ציור השחקנים.

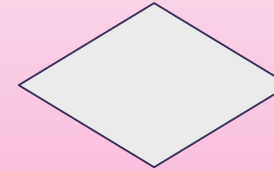
► רציתי שבזמן הקפיצה השחקנים יוכלו לזוז לצדדים ולכן לכל שחקן יש שני מצבים, מצב אחד שהוא על הקרקע ומצב שני שהוא בזמן קפיצה. במידה והשחקן בזמן קפיצה הוא עולה במספר במסוים ואחר כך בודקים אם נלחץ מקש, לאחר מכן השחקן עולה שוב, כך קורה עד גובה קבוע מקסימלי של קפיצה. לכן הבדיקה אם השחקן נמצא במצב קפיצה מתבצע בלולאה באופן קבוע.

תרשימי זרימה על פעולות מרכזיות

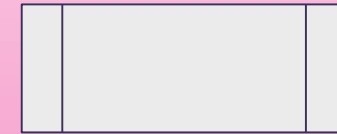
Start/End



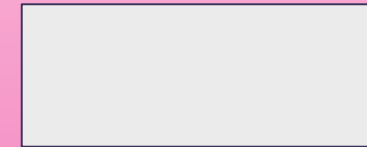
Condition



Function



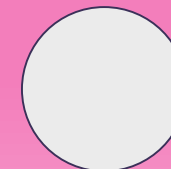
Command/Operation



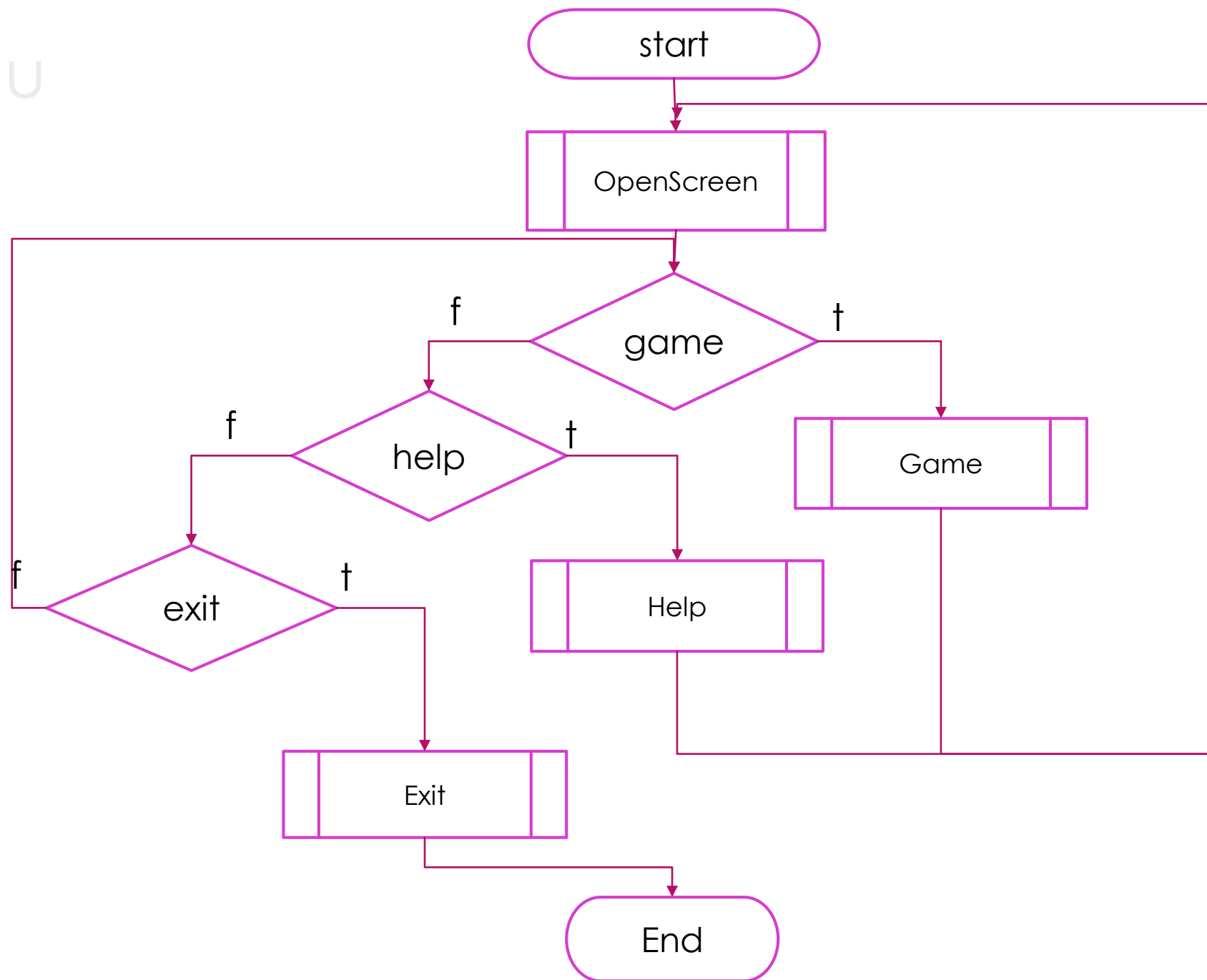
Input/Output



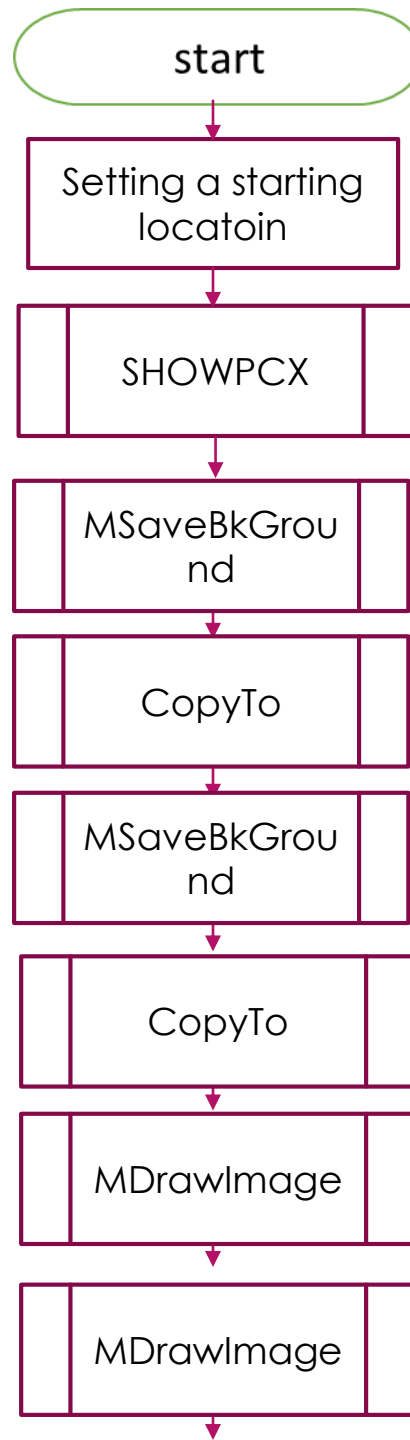
Reference to a different place in the flowchart



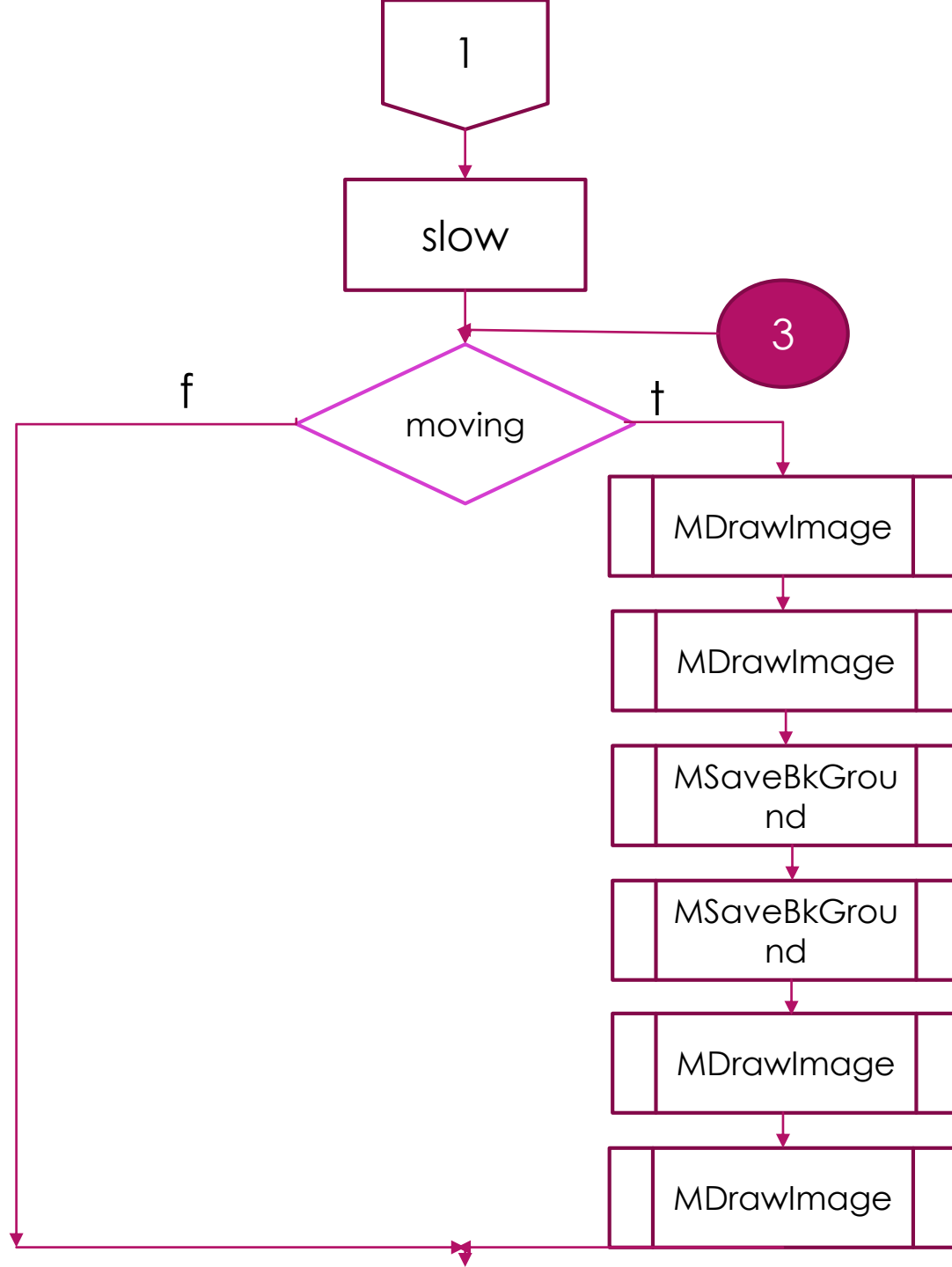
Menu



game



game



2

game

BoysObstacle
s

GirlsObstacles

MoveVertical

MoveVertical

Gets a key

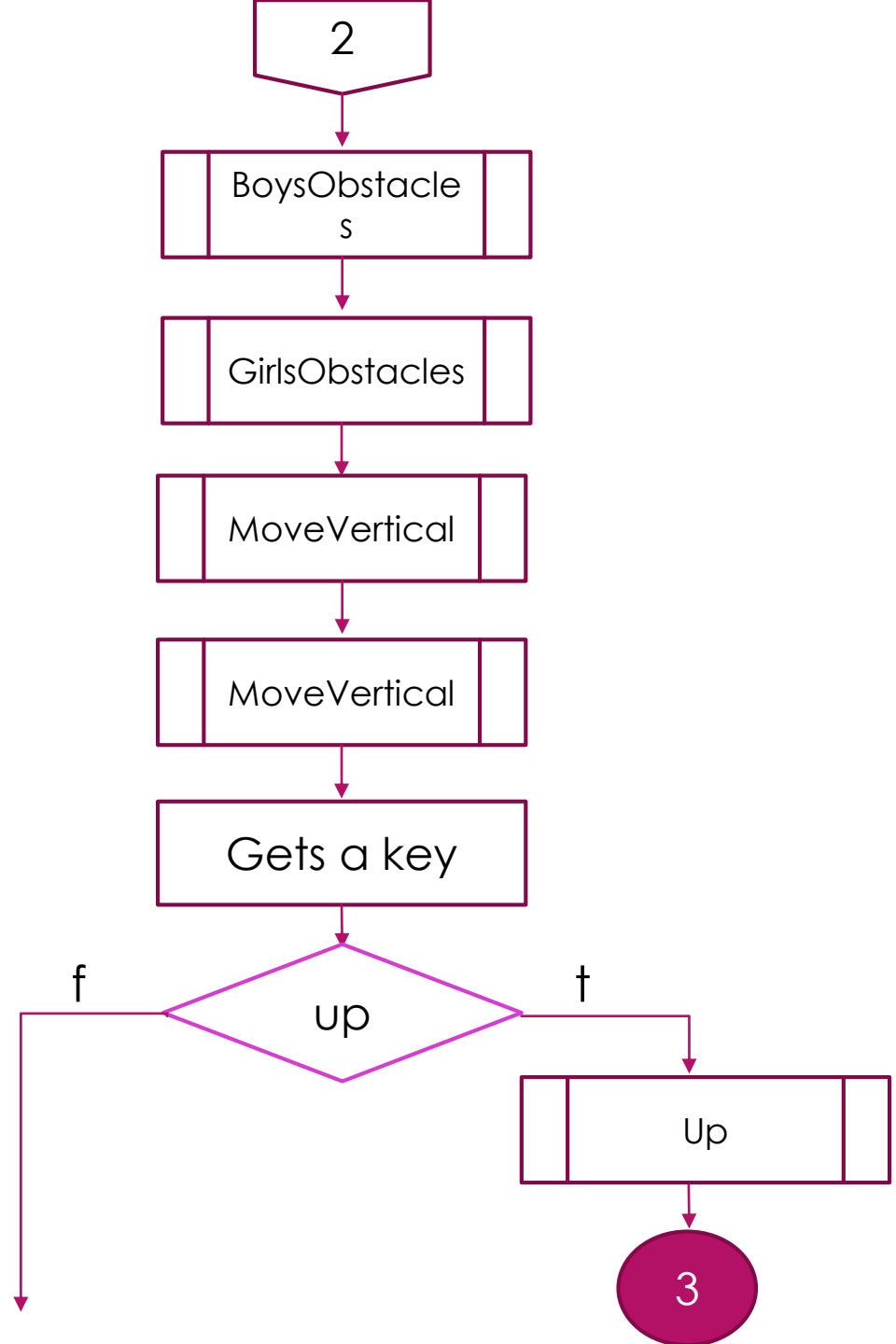
up

f

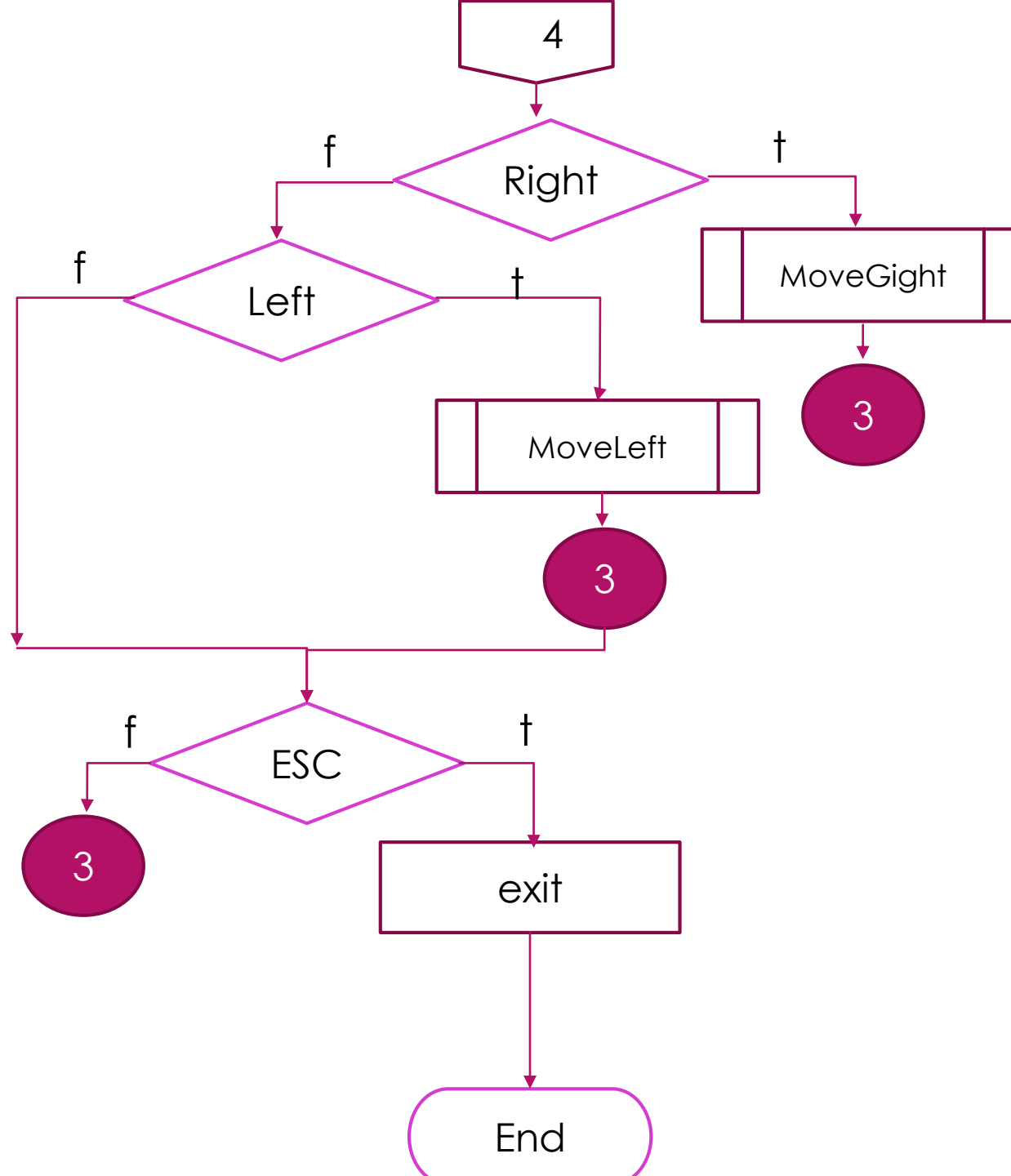
t

Up

3



game



הסבר כללי על הפעולות

```
;------▶
; SHOWPCX - מקבל מיקום התחלה ומיקום סופי, שם קובץ ושולח לפונקציה שמייצרת▶
;------▶
; Input:▶
;   StartX1, StartY1 - pcx location▶
;   fName - the pcx address▶
; Output:▶
;   None▶
; Registers▶
;   AX, dx▶
;------▶
```

```
;------▶
; Up player is in jump - אם השחקן לחץ על למעלה לשנות סטטוס▶
;------▶
; Input:▶
;   status - the player mode in the bord▶
;▶
; Output:▶
;   None▶
; Registers▶
;   None▶
;------▶
```

```
;------▶
; MDrawImage ; מקבל מיקום ומה לצייר, שולח לפונקציה▶
;------▶
; Input:▶
;   StartX, StartY - Image location▶
;   Imgname - Name of data image▶
;   IsRestore - Restore background or draw image▶
; Output:▶
;   The image ;▶
; Registers▶
;   AX, SI▶
;------▶
```

```
;------▶
; MSaveBkGround ; שומר את צבע הרקע מאחור▶
;------▶
; Input:▶
;   StartX, StartY - Image location▶
;   temp_array - the array for save a back ground▶
; Output:▶
;   None▶
; Registers▶
;   AX, SI▶
;------▶
```


;-----▶
 בודק האם השחקן רוצה לקפוץ ואם כן האם הוא יכול לקפוץ למעלה לפי הגבולות, אם כן מעלה ב- 1 אם לא משנה סטטוס
 לירידה ומוריד אותו עד שהוא מגיע לגבול
 ; MoveHorizontal- chack if the player is in jump. move the player accordingly▶
 ;-----▶
 ; Input:▶
 ; PlayerX, PlayerY - the loaction of the player▶
 ; status - the player mode in the bord▶
 ; HighJump - height of the jump▶
 ; Output:▶
 ; None▶
 ; Registers▶
 ; None▶
 ;-----▶
 ;-----▶
 ; MoveIfPossibleLeft - chack according the color if the player can move.▶
 ; if he can- move the player ; אם השחקן יכול לזוז שמאלה, מזיז את השחקן▶
 ;-----▶
 ; Input:▶
 ; newX, newY playerX - the new loaction▶
 ; playerX - the x of the player▶
 ; Output:▶
 ; None▶
 ; Registers▶
 ; ax, bx, cx, dx▶
 ;-----▶

;-----▶
 ; MoveRight calculates the new loaction ; שולח לפונקציה▶
 לבדיקה האם אפשר לזוז ימינה▶
 ;-----▶
 ; Input:▶
 ; PlayerX, PlayerY - the loaction of the player▶
 ;▶
 ; Output:▶
 ; None▶
 ; Registers▶
 ; None▶
 ;-----▶
 ;-----▶
 ; MoveLeft calculates the new location▶
 שולח לפונקציה▶
 לבדיקה האם אפשר לזוז ימינה▶
 ;-----▶
 ; Input:▶
 ; PlayerX, PlayerY - the loaction of the player▶
 ;▶
 ; Output:▶
 ; None▶
 ; Registers▶
 ; None▶
 ;-----▶

macro

```
;-----▶
מחבר שני אובייקטים▶
; AddTo - add one object to other object▶
;-----▶
; Input:▶
; destination - the object to change▶
; toAdd - the object to add▶
; Output:▶
; None▶
; Registers▶
; ax▶
;-----▶
;-----▶
; SubTo - sub one object to other object▶
מחסר שני אובייקטים▶
;-----▶
; Input:▶
; destination - the object to change▶
; toSub - the object to sub▶
; Output:▶
; None▶
; Registers▶
; ax▶
;-----▶
```

```
;-----▶
; MoveIfPossibleRight - check according the color if the player can move.▶
; אם השחקן יכול לזוז ימינה מזיז את השחקן▶
;-----▶
; Input:▶
; newX, newY playerX - the new loaction▶
; playerX - the x of the player▶
; Output:▶
; None▶
; Registers▶
; ax, bx, cx, dx▶
;-----▶
;-----▶
; CopyTo - copy one object to other object▶
מעתיק ערך מאובייקט אחד לאחר.▶
;-----▶
; Input:▶
; destination - the object to change▶
; toCopy - the object to copy▶
; Output:▶
; None▶
; Registers▶
; ax▶
;-----▶
```

procedure

ObOneGirl ; המכשול הראשון של הבת.

```
;------▶
; Input: ▶
StartPictX2, StartPictY2 ; ▶
; Output: ▶
; ▶
; Registers ▶
none ; ▶
;------▶
;------▶
ObOneBoy ; המכשול הראשון של הבן
;------▶
; Input: ▶
StartPictX, StartPictY ; ▶
; Output: ▶
; ▶
; Registers ▶
none ; ▶
;------▶
```

המכשולים של הבת, בודק איפה השחקנית נמצאת ולפי זה שולח לבדיקה המתאימה.

```
;------▶
; GirlsObstacles - all the obstacles player 2 ▶
;------▶
; Input: ▶
StartPictX2, StartPictY2 ; ▶
; Output: ▶
; ▶
; Registers ▶
none ; ▶
;------▶
```

המכשולים של הבן, בודק איפה השחקן נמצא ולפי זה שולח לבדיקה המתאימה.

```
;------▶
; BoysObstacles - all the obstacles player 1 ▶
;------▶
; Input: ▶
StartPictX, StartPictY ; ▶
; Output: ▶
; ▶
; Registers ▶
none ; ▶
;------▶
```

procedure

ObFourBoy ; המכשול הרביעי של הבן בודק אם הוא הגיע למתג ואם כן
משגר את הבת אל המתג שלה

; Input:
StartPictX, StartPictY, StartPictX2, StartPictY2;
; Output:
;
; Registers
none ;

ObFourGirl ; המכשול הרביעי של הבת, בודק אם היא הגיע למתג שלה
ואם כן משגר את שניהם להמשך

; Input:
StartPictX, StartPictY, StartPictX2, StartPictY2;
; Output:
;
; Registers
none ;

ObThreeGirl ; המכשול השלישי של הבת. בודק אם היא הגיעה לכפתור
ואם כן משגר אותה אל היהלום הכחול

; Input:
StartPictX, StartPictY, StartPictX2, StartPictY2;
; Output:
;
; Registers
none ;

ObThreeBoy ; המכשול השלישי של הבן. בודק אם הוא הגיע לכפתור ואם
כן משגר אותו אל היהלום הכחול

; Input:
StartPictX, StartPictY, StartPictX2, StartPictY2;
; Output:
;
; Registers
none ;

procedure

;-----▶ ; קורא את הקובץ pcx ; ReadPCXFile - read pcx ;-----▶	;-----▶ ; PDrawImage - draw the image הדמות. הגודל של ;-----▶
; Input:▶ file ;▶ ; Output:▶ file read ;▶ ; Registers▶ all ;▶ ;-----▶	; Input:▶ h_img, w_img, current_x, current_y, TRANSPARENT_COLOR, ;▶ is_restore▶ ; Output:▶ ;▶ ; Registers▶ ax, bx, cx, dx, si ;▶ ;-----▶
;-----▶ ;-----▶ ; מציג את הקובץ pcx ; ShowPCXFile - show PCX file ;-----▶ ; Input:▶ File name ;▶ ; Output:▶ The file ;▶ ; Registers▶ AX, BX, CX, DX, DS ;▶ ;-----▶	;-----▶ ;-----▶ ; שומר את הרקע מאחורי הדמות ; PSaveBkGround - save הרקע. ;-----▶ ; Input:▶ h_img, w_img, current_x, current_y;▶ ; Output:▶ ;▶ ; Registers▶ ax, bx, cx, dx, si ;▶ ;-----▶

procedure

```
;-----▶
; Game the real game ▶
;-----▶
; Input: ▶
ah ; ▶
; Output: ▶
dx ; ▶
; Registers ▶
ah, dx ; ▶
;-----▶
;-----▶
; help for the ueser ▶
;-----▶
; Input: ▶
ah ; ▶
; Output: ▶
dx ; ▶
; Registers ▶
ah, dx ; ▶
;-----▶
```

```
;-----▶
; מציג פיקסל על המסך ; PutPixel - draw pixel ▶
;-----▶
; Input: ▶
x - Point_x, y - Point_y, Color - color ; ▶
; Output: ▶
The pixel ; ▶
; Registers ▶
AX, BH, CX, DX ; ▶
;-----▶
;-----▶
; OpenScreen the first screen in game there is the menu. ▶
;-----▶
; Input: ▶
ax ; ▶
; Output: ▶
dx; ▶
; Registers ▶
AX, dx ; ▶
;-----▶
```

```
;------▶
finishB ; בודק אם הבן הגיע למקום הסופי, אם כן שם במשתנה 1.▶
;------▶
; Input:▶
StartPictX, StartPictY;▶
; Output:▶
;▶
; Registers▶
none ;▶
;------▶
;------▶
finishG ; בודק אם הבת הגיעה למקום הסופי, אם כן שם במשתנה 1.▶
;------▶
; Input:▶
StartPictX2, StartPictY2;▶
; Output:▶
;▶
; Registers▶
none ;▶
;------▶
```

```
;------▶
; exit the game▶
;------▶
; Input:▶
ah ;▶
; Output:▶
dx ;▶
; Registers▶
ah, dx ;▶
;------▶
```

Game kod

Start:

```
mov ax, @data  
mov ds, ax
```

menu:

```
call OpenScreen
```

GameScreen:

```
;    grapic mode  
    mov ax, 013h  
    int 010h  
    mov ax, 0A000h  
    mov es, ax  
    call Game
```

HelpScreen:

```
call Help  
jmp menu
```

ExitScreen:

```
jmp exit
```


Game kod

```
ExitScreen:2  
; text mode  
mov ax,03h  
int 10h  
call ToExit  
jmp menu
```

```
finishscreen:  
mov ax,03h  
int 10h  
call ToFinish  
jmp menu
```

start_game :

```
; graphic mode  
mov ax, 13h  
int 10h  
  
mov [finish1B], 0  
mov [finish2G], 0  
  
mov [StartPictX], 20  
mov [StartPictY], 179  
mov [StartPictX20], [2  
mov [StartPictY150], [2
```

Game kod

```
mov [StartPictX0],[1d ; start loaction pcx  
mov [StartPictY0],[1d
```

```
    SHOWPCX StartPictX1emaNeliF ,1YtciPtratS ,1  
mov [w_img],11  
    mov [h_img], 16  
MSaveBkGround StartPictX, StartPictY, temp_array ; save background behind the player  
    CopyTo befor_x, StartPictX ; save the current loaction  
    CopyTo befor_y, StartPictY
```

```
    MSaveBkGround StartPictX lrig ;2 yarra_pmet ,2YtciPtratS ,2  
CopyTo befor_x 2XtciPtratS ,2  
    CopyTo befor_y 2YtciPtratS ,2
```

```
MDrawImage StartPictX, StartPictY, Img, 1  
    MDrawImage StartPictX1 ,2gml ,2YtciPtratS ,2  
    CopyTo temp2XtciPtratS ,2
```

L2:

```
    cmp [move], 0 ; if the player not move (pressure a key)  
    je no_moving
```

Game kod

```
MDrawImage befor_x, befor_y, temp_array,0
CopyTo befor_x, StartPictX
CopyTo befor_y, StartPictY

MDrawImage befor_x2,0,yarra_pmet ,2y_rofeb ,2
CopyTo befor_x2XtciPtratS ,2
CopyTo befor_y2YtciPtratS ,2

MSaveBkGround StartPictX, StartPictY, temp_array
MSaveBkGround StartPictX Irig ;2 yarra_pmet ,2YtciPtratS ,2

MDrawImage StartPictX, StartPictY, Img, 1
MDrawImage StartPictXlrig ;1 ,2gml ,2YtciPtratS ,2
```

no_moving :

```
mov [move], 0
call BoysObstacles
call GirlsObstacles
cmp finish1B, 1
jne gameNotFinish
cmp finish2G, 1
jne gameNotFinish
jmp finishscreen
```

Game kod

```
gameNotFinish:  
    mov [finish1B], 0  
    mov [finish2G], 0
```

```
moveVerticl:  
    dec currentNumber  
    cmp currentNumber, 0  
    jne readKey
```

```
MoveVertical StartPictX, StartPictY, status, HighJump ; boy  
MoveVertical StartPictXlrig ; pmuJhgiH ,2sutats ,2YtciPtratS ,2  
CopyTo currentNumber, number
```

```
readKey :  
-----;Read SCAN code from keyboard port  
    in al,060h  
    mov [key], al  
  
    cmp [key], UP_KEY  
    je Up1  
  
    cmp [key], RIGHT_KEY  
    je Right1  
  
    cmp [key], LEFT_KEY  
    je Left1
```

Game kod

Up :2

Up status2

jmp cont

Right:2

MoveRight StartPictX2YtciPtratS ,2

jmp cont

Left:2

MoveLeft StartPictX 2YtciPtratS ,2

jmp cont

Cont:

cmp [key], ESC_KEY

je quit

jmp L2

quit:

mov ax, 3h

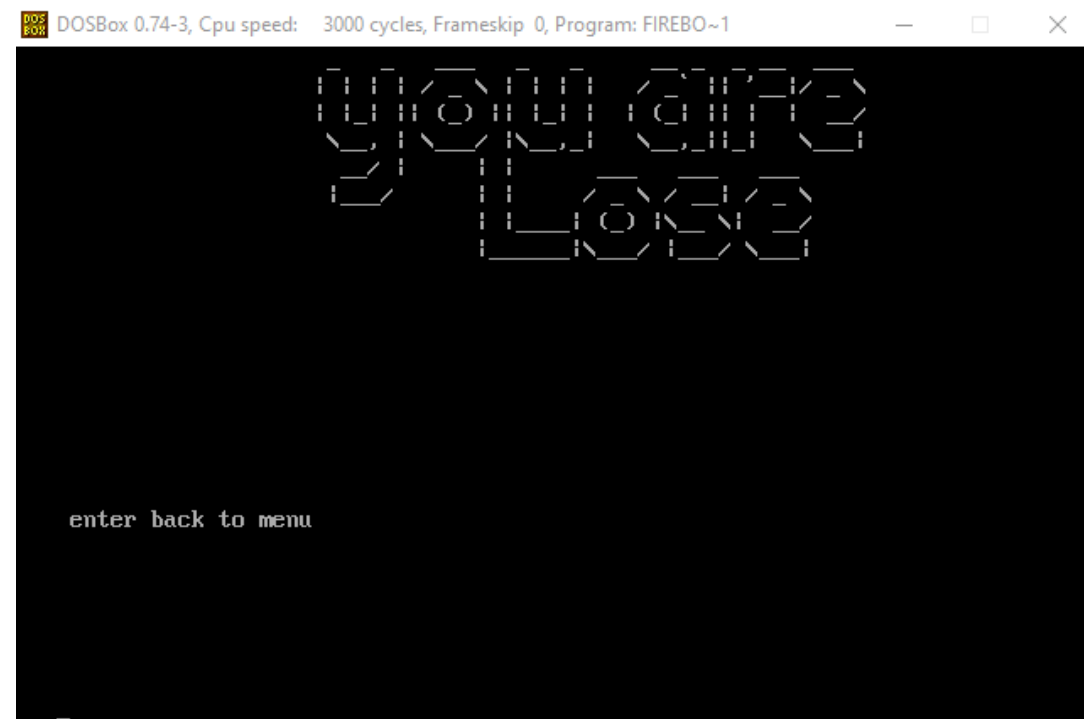
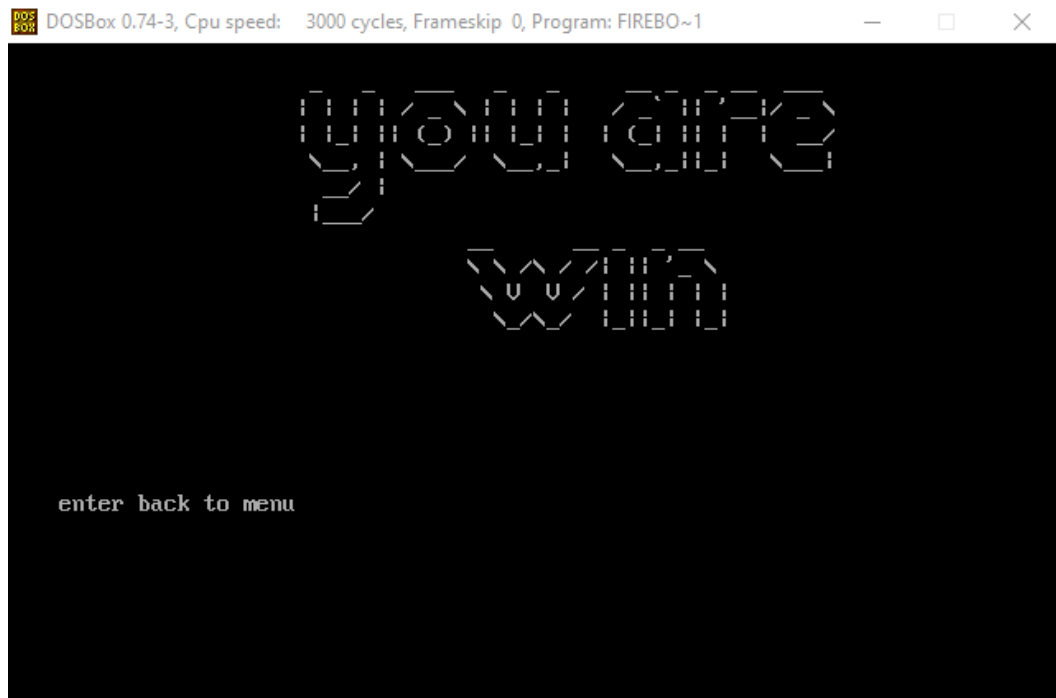
int 10h

Exit:

mov ax, 4C00h

int 21h





סיכום

- ▶ תהליך כתיבת הפרויקט בשבילי הייתה חוויה חיובית ומאתגרת. חיווה בה הייתי צריכה לעשות משהו שלא עשיתי לפני כן בצורה הזאת. למדתי איך ללמוד עצמאית ואיך לתכנן את הקוד שלי לפני הכתיבה. למדתי לשבת הרבה בלי להתייאש ושמותר וזה טוב לבקש עזרה מאחרים.
- ▶ תודה רבה לאנטולי פיימר, לאור שמש וגד דבח שעזרו לי במהלך כתיבת הפרויקט.