משחק בתווים דינאמי - חללית

בדף עבודה זה נתנסה בעבודה במצב טקסט של חללית החומקת ממטאורים.

ניצור חללית שתנוע על המסך ימינה ושמאלה, מטאור שיפול מחלקו העליון של המסך. אם הוא פוגע בחללית יש פסילה.

# מצב טקסט

מסך ה – dos box מחולק לשורות ועמודות, ישנם שני מצבים של תצוגת הטקסט על המסך:

* מצב של 80 עמודים ו 25 שורות (המצב ברגיל בו אנו עובדים)
* ומצב של 40 עמודות ו – 25 שורות.

הפסיקות הקובעות את מצב תצוגת המסך והמתנה למקש

|  |  |
| --- | --- |
| כניסה למוד טקסט 25\*80 25 שורות ו - 80 עמודות תווים | mov al, 03h  mov ah, 0 int 10h |
| כניסה למוד טקסט 25\*40 25 שורות ו - 40 עמודות תווים | mov al, 13h mov ah, 0 int 10h |
| המתנה למקש  התו נשמר לרגיסטר al, והתכנית עוצרת עד להקשה על מקש. | mov ah, 0h  int 16h |

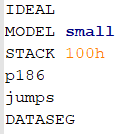
כתבו תכנית קצרה שתעזור לנו לעבוד בצורה רציפה ונוחה.

* מעבר למוד של 40\*25
* המתנה למקש
* יציאה למוד טקסט רגיל.

**הריצו ובדקו את מצב המסך.**

תקבלו מסך גרפי שחור ובהקשה על מקש נצא למוד טקסט רגיל.

**p186 ו - jumps**

הוסיפו בתחילת הקובץ את הפקודה p186 כדי שנוכל לעשות push ו – pop לכל הרגיסטרים יחד. (pusha ו - popa)

והוסיפו את הפקודה jumps כדי שנוכל לעשות קפיצות "גדולות" בתכנית שלנו.

# מיקום תו על המסך

ראשית ממקמים את הסמן על גבי המסך ואחר כך מציירים את התו הרצוי:

לשם כך נשתמש בשלושה משתנים אשר יציינו את:

* מיקום התו בשורה – צירה ה – x
* מיקום התו בעמודה ציר ה – y
* צבע התו

נמקם את המטאור שלנו באחת השורות העליוניות של המסך ובאמצע.

y\_meteor db 2

x\_meteor db 19

colour\_meteor db 11 ; Light Cyan

צרו ואתחלו את המשתנים במקטע הנתונים.

כתבו שתי פרוצדורות למיקום סמן וציור שחקן.

|  |  |
| --- | --- |
| מיקום סמל אל פני המסך:  proc Cursor\_location\_meteor  ;Place the cursor on the screen  mov bh, 0  mov dh, [y\_row] ; in row  mov dl, [x\_column] ; row  mov ah, 2  int 10h  ret  endp Cursor\_location\_meteor | ציור תו על המסך:  proc DrawMeteor  ; draw player in cursor position  mov ah, 9  mov al, 15 ;AL = character to display  mov bh, 0h ;BH=Page  mov bl, [colour] ; BL = Foreground  mov cx, 1 ; number of times to write character  int 10h ; Bois -> show the character  ret  endp DrawMeteor |

נזמן את הפרוצדורות לאחר שרטוט לוח המשחק (המחרוזת שלנו), ונריץ את התכנית.

call Cursor\_location\_meteor

call DrawMeteor

**הריצו ובדקו שאתם רואים על המסך את השחקן, הקישו על מקש וסיימו את ההרצה.**

**שמרו את הקובץ, בשם חדש והמשיכו בקובץ החדש.**

# הוספת מקש

נוסיף לתכנית שלנו :

תווית המתנה למקש

בדיקה האם הוקש מקש במקלדת

אם הקישו על מקש p נתחיל במחשק קפיצה לתווית Move

אם הקישו על מקש Esc נצא מהמשחק קפיצה לתווית TheEnd

קפיצה לתווית – המתנה למקש

מבנה התכנית לאחר הדפסת השחקן למסך

call Cursor\_location\_meteor

call DrawMeteor

; Read character and save it to AL

WaitForCharacter:

mov ah, 0h

int 16h

; Was Esc pressed?

cmp al, 27

je TheEnd

; Was p pressed?

cmp al, 'p'

je Move

jmp WaitForCharacter

Move:

התווית **Move**

תהייה התווית של הלולאה הראשית במשחק שלנו.

TheEnd:

; Return text mode 80x25 (x,y)

mov al, 03h

mov ah, 0

int 10h

**הריצו ובדקו.**

שימו לב 🎔, רק בהקשה על Esc או p נצא מהתכנית.

שמרו את הקובץ, בשם חדש והמשיכו בקובץ החדש.

# הזזת המטאור

כדי להזיז את המטאור "מוחקים אותו":

* מחיקת מטאור (ציור המטאור במיקום שהיה בצבע שחור)
* מגדילים את הערך של המטאור על ציר y ב – 1.
* ממקמים את הסמן במיקום הבא
* משנים את המשתנה של color לצבע
* מציירים את המטאור בצבע במיקום החדש.
* קופצים חזרה לתווית Move

לאחר התווית Move ניצר את הפקודות המוחקות ומזיזות את המטאור כלפי מטה.

Move:

; Del meteor

mov [color\_meteor], 0

call Cursor\_location\_meteor

call DrawMeteor

; Move meteor down and draw it in new location

inc [y\_meteor]

call Cursor\_location\_meteor

mov [color\_meteor], 11

call DrawMeteor

jmp Move

הריצו את התכנית:  
לתכנית שלנו שתי בעיות: המטאור נופל מהר מידי אין אפשרות לצאת בצורה מסודרת.

## השהיה

ניצור פעולה המשהה את המשחק:

ההפרש בין רגיסטר cx ורגיסטר dx קובע את זמן ההשהיה.

proc delay

pusha

**פסיקת השהייה:**

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

CX:DX = interval in microseconds

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

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

CX = 03h : DX = 4240h = 200000 microsec. = 1/5 second

פסיקה זו ממתינה חמשית שניה.

כלומר, משהה את תזוזת ב 200,000 מיליוניות שניה

mov cx, 03h ;High Word

mov dx, 4240h ;Low Word

mov ah, 86h ;Wait

int 15h

popa

ret

endp delay

## נוסיף לתכנית פסיקה הממתינה להקשת מקש במקלדת.

נכתוב את הפסיקה בלולאה הראשית של התכנית לאחר שרטוט המטאור למסך.

הפסיקה ממתינה למקש אם לא הוקש מקש, נקפוץ לתווית **noKey**

; check if there is a character to read

mov ah, 1h

int 16h

jz noKey

אם נקלטה הקשה במקלדת נבדוק מהו התו שהוקש?

; read a character and save it to al

mov ah, 0h

int 16h

אם הוקש מקש Esc (ascii 27)

נקפוץ לתווית TheEnd

נחזור למוד תצוגה רגיל ונסיים את התכנית

cmp al, 27 ; if esc

je theEnd

אם לא הוקש מקש

נזמן את הפעלה של ההשהיה

ונחזור לתווית **Move** .  
(לתחילת הלולאה הראשית של המשחק)

**noKey:**

call delay

jmp Move

**TheEnd:**

הריצו את התכנית ובדקו שהמטאור נופל לא מהר מידי, ובהקשה על Esc יוצאים מהתכנית.

שמרו את הקובץ בשם חדש, והמשיכו בקובץ החדש.

# מיקום ותזוזת המטאור

ראשית נרצה להוסיף תנאי שברגע שהמטאור מגיע לתחתית המסך הוא יחזור לחלקו העליון אך במיקום חדש, מיקום אקראי.

לצורך כך ניצור שני משתנים במקטע הנתונים:

bx\_saver dw ?

rad db ?

* המשתנה rnd ישמור את המספר האקראי שיצרנו בפעולה
* המשתנה bx\_saver יסייע לנו לעשות מניפולציה על המספר שנמשוך מהשעון כדי שאם נגריל שני מספרים בטווח זמן קצר הם יהיו שונים

**נכתוב פעולה ליצירת מספר אקראי:**

proc random

;generates a random number and keeps it in rnd

pusha

mov bx, [bx\_saver]

;put segment number in register es

mov ax, 40h

mov es, ax

;move random number to ax

mov ax, [es:6Ch]

xor ax,[bx]

add [bx\_saver],ax

and al, 27h ;check if number < 39, which is 27 in hex

mov [rnd],al

popa

ret

endp random

נוסיף ללולאה של המשחק הראשי שלנו

את התנאי של האם מיקום המטאור על ציר y שווה ל – 25 (הגענו לתחתית המסך)

נשנה את ערכו ל 2 (תחילת המסך)

ונשנה את המיקום של מטאור על ציר ה – x למספר חדש שאותו נגריל בפעולה – random

שימו לב 🎔, את השורות הכתובת באפור יש לכם כבר בתכנית

Move:

mov [color\_meteor], 0

call Cursor\_location\_meteor

call DrawMeteor

התנאי הבודק האם הערך של מיקום המטאור על ציר y שווה ל – 25.

אם כן, נקפוץ לתווית - changePlace

inc [y\_meteor]

cmp [y\_meteor], 25

je changePlace

call Cursor\_location\_meteor

mov [color\_meteor], 11

call DrawMeteor

; check if there is a character to read

mov ah, 1h

int 16h

jz noKey

; read a character and save it to al

mov ah, 0h

int 16h

cmp al, 27 ; if esc

je theEnd

noKey:

call delay

jmp Move

**תווית changePlace**

* מזמנת את הפעולה של יצירת מספר אקראי
* מעבירה את המספר למשתנה x\_meteor כדי שבשרטוט הבא של המטאור על המסך הוא יוצג במיקום חדש על ציר x.
* משנה את הערך של מיקום הסמן על ציר y מחזירה את מיקום המטאור לתחילת המסך.
* קופצת לתווית Move (חוזרת לתחילת הלולאה הראשית של המשחק)

changePlace:

call random

mov al, [rnd]

mov [x\_meteor], al

mov [y\_meteor],2

jmp Move

TheEnd:

; Return text mode 80x25 (x,y)

mov al,03h

mov ah,0

int 10h

הריצו את התכנית, בדקו שבכל פעם בה המטאור מגיע לתחתית המסך הוא חוזר לחלקו העליון במיקום חדש.

שמרו את הקובץ בשם חדש והמשיכו לעבוד בקובץ החדש.

# ציור החללית

צרו שתי פעולות חדשות למיקום הסמן של החללית וציור החללית.

לצורך כך צרו שלושה משתנים חדשים במקטע הנתונים.

x\_ship db 19

y\_ship db 13

color\_ship db 14

במקטע הקוד כתבו את שתי הפעולות:  
בחרו בתו ascii לחללית. (טבלה בסוף הקובץ)

Cursor\_location\_ship

DrawShip

מקמו את ציור החללית בלולאה הראשית לאחר הבדיקה עם המטאור הגיע לשורה 25 ושרטוט המטאור במיקומו החדש. (את קטע הקוד באפור כר יש לכם כבר בתכנית).

Move:

mov [color\_meteor], 0

call Cursor\_location\_meteor

call DrawMeteor

inc [y\_meteor]

cmp [y\_meteor], 25

je changePlace

call Cursor\_location\_meteor

mov [color\_meteor], 11

call DrawMeteor

call Cursor\_location\_ship

call DrawShip

הריצו את התכנית ובדקו שהחללית שלכם מופעיה על המסך.

## תזוזת החללית

החללית יכולה לזוז ימינה ושמאלה אבל מיקומה על ציר y נשאר קבוע.

ניצור פעולה המזיזה את החללית ימינה.

כדי לזוז ימינה נגדיל את הערך של ציר x ב – 1.

* נמחק את החללית (ציור החללית במיקום שלה בשחור)
* נגדיל את המיקום על ציר x
* נקבע צבע
* נמקם את הסמן במיקום החדש
* נצייר את החללית במיקום החדש

proc RightKeyPress

pusha

mov [color\_ship], 0 ;black color

call Cursor\_location\_ship

call DrawShip ; draw black character

**add [x\_ship], 1**

mov [color\_ship], 13

call Cursor\_location\_ship ;move to new location

call DrawShip ; Draw color character

popa

ret

endp RightKeyPress

צרו פעולה המזיזה את החללית שמאלה.

כדי לזוז שמאלה נקטין את הערך של ציר x ב – 1.

בתכנית הראשית נוסיף את הבדיקה של

האם הקשנו על מקש חץ ימני ⇨

או האם הקשנו על מקש חץ שמאלי ⇦

נוסיף את הבדיקה של האם הקשנו על מקשי החיצים לאחר הבדיקה של האם הקשנו על מקש Esc

cmp al, 27 ; if esc

je theEnd

האם הקשנו על מקש חץ ימני

אם כן קפיצה לתווית - **RightKey**

האם הקשנו על מקש חץ שמאלי

אם כן קפיצה לתווית - **LeftKey**

**תווית – NoKey**

זימון פעולת השהייה

קפיצה לתווית Move

;Was right key pressed?

cmp ah,4dh ; Right arrow -->>

je RightKey

;Was left key pressed?

cmp ah,4bh ; Left arrow <<--

je LeftKey

noKey:

call delay

jmp Move

**תווית RightKey**

זימון פעולה של הזזת חללית ימינה

קפיצה לתווית Move

**תווית LeftKey**

זימון פעולה של הזזת חללית שמאלה

קפיצה לתווית Move

**תווית – changePlace**

מיקום מטאור בשורה העליונה של המסך במיקום חדש.

RightKey:

call RightKeyPress

jmp Move

LeftKey:

call LeftKeyPress

jmp Move

changePlace:

call random

mov al, [rnd]

mov [x\_meteor], al

mov [y\_meteor],2

jmp Move

את הפקודות בצבע אפור יש לכם כבר בתכנית

## פסילה - האם המטאור נוגע בחללית

נוסיף למטאור בדיקה האם נוגע בחללית.

נמחק את המטאור

נקדם את המיקום של המטאור על ציר y

נמקם את הסמן של המטאור במיקום זה

**נבדוק מהו התו הנמצא במיקום הסמן**

אם שווה לתו של החללית – פסילה

נקפוץ לתווית - TheEnd

נקבע צבע לחללית

נצייר את החללית במיקום החדש.

פסיקה הבודקת מהו ה"תו" במיקום של הסמן ושמה את הערך ברגיסטר al.

; check the character on coordinates

mov bh, 0h ; Page=1

mov ah, 08h ; Read character function

int 10h ;return the character value to AL

בתכנית הראשית נוסיף את הספיקה של קריאת התו במיקום הסמן בפסיקות של הזזת המטאור.

Move:

mov [color\_meteor], 0

call Cursor\_location\_meteor

call DrawMeteor

inc [y\_meteor]

cmp [y\_meteor], 25

je changePlace

call Cursor\_location\_meteor

; check the character on coordinates

mov bh, 0h ; Page=1

mov ah, 08h ; Read character function

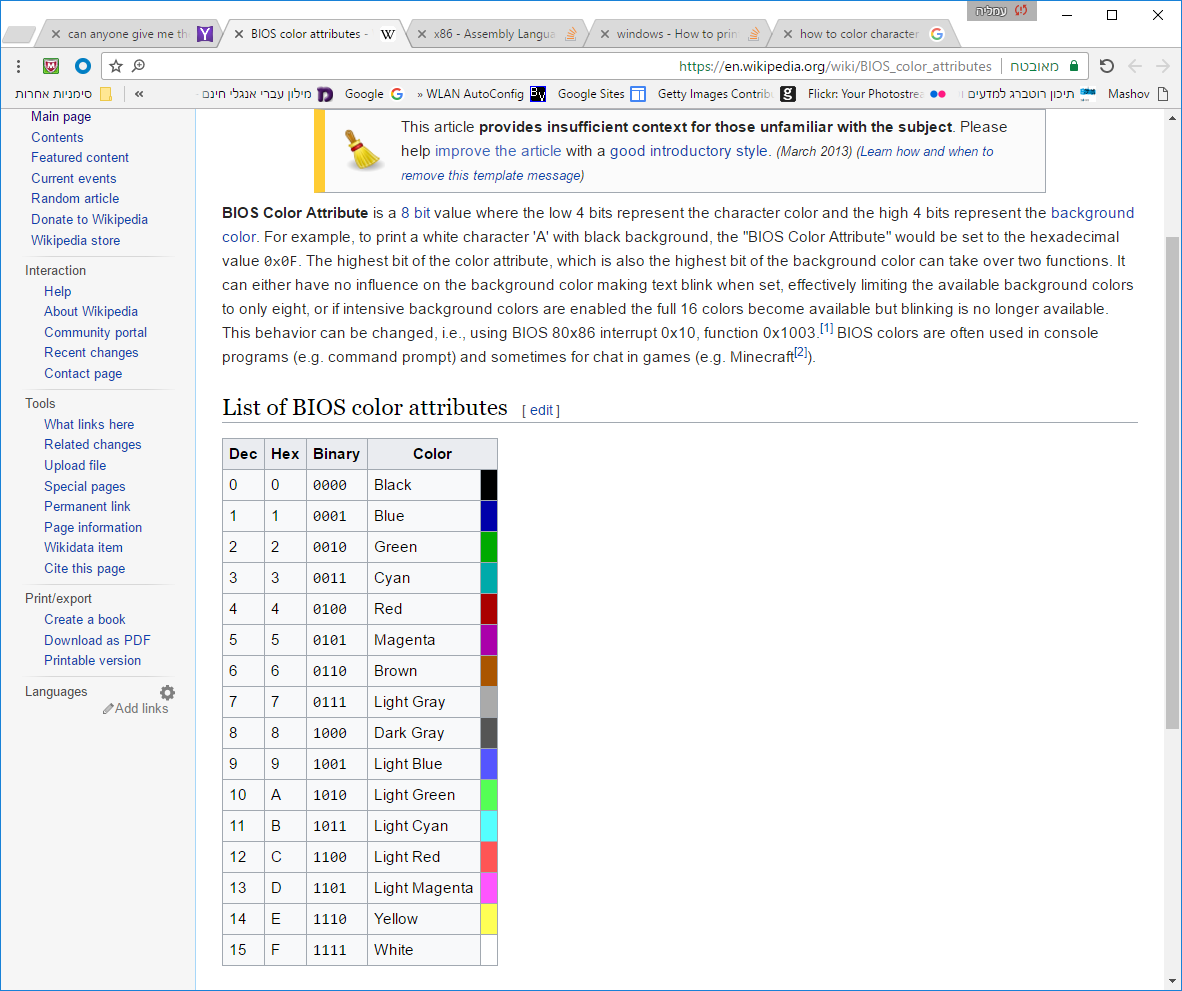
int 10h ;return the character value to AL

cmp al, 127

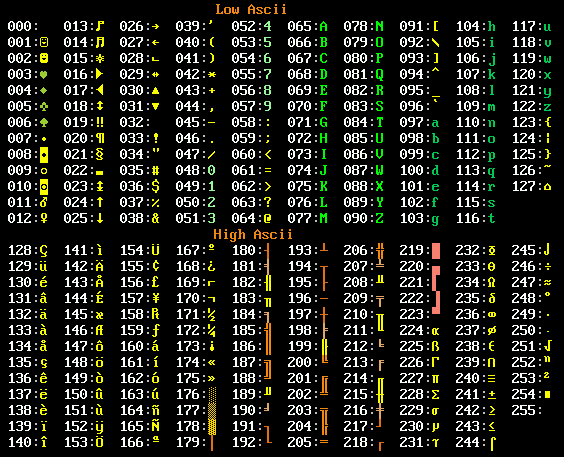
je TheEnd

mov [color\_meteor], 11

call DrawMeteor

טבלת צבעים

טבלת תווים (Ascii)



## האלגוריתם הכללי של התכנית:

מעבר למוד גרפי (25 \* 40)

מיקום  סמן של מטאור

ציור מטאור

**תווית – WaitForCharacter**

אם הקישו אל Esc - asci 27

קפוץ תוויית - **TheEnd**

אם הקישו על מקש p

קפוץ לתווית – **Move**

קפיצה לתווית - **WaitForCharacter**

**תווית -  Move**

מחיקת מטאור (צבע שחור, מיקום סמן, ציור שחקן)

הגדלת מיקום המטאור על ציר y

בדיקה האם מיקום על ציר y = 25

אם כן קפיצה לתווית – **changePlace**

מיקום סמן של מטאור

ציור מטאור במיקום חדש ( קביעת צבע, מיקום סמן, ציור שחקן)

בדיקה מהו התו במיקום הסמן

אם התו שווה ל – 127 (תו החללית)

קפוץ לתווית **the\_End**

ציור מטאור (קביעת צבע למטאור, וציור מטאור)

המתנה למקש

אם לא הוקש מקש קופצים לתווית   **NoKy**

; check if there is a character to read

mov ah, 1h  
int 16h  
**jz noKey**

קריאה של תו מהמסך ושמירה לרגיסטר al

; read a character and save it to al

mov ah, 0h

int 16h

אם הוקש  esc

קופצים לתווית - the\_End

אם הוקש מקש חץ  ימינה

קופצים לתווית - RightKey

**תווית -   NoKy**

זימון פעולה של השהייה

קפיצה לתווית Game

**תווית - RightKey**

זימון פעולה של  מקש ימני

קפיצה לתווית  - **Move**

**תווית - LeftKey**

זימון פעולה של  מקש שמאלי

קפיצה לתווית  - **Move**

**תווית – changePlace**

זימון פעולה ליצירת מספר אקראי

שמירת המספר האקראי למשתנה של המטאור על ציר x

שינו המשתנה של מיקום המטאור על ציר y ל -2

קפיצה לתווית Move

**תווית -  the\_End**

יציאה התצוגה גראפית וחזרה לתצוגת טקסט (25 \* 80)

סיום תכנית

# קוד התכנית

IDEAL

MODEL small

STACK 100h

p186

jumps

DATASEG

; --------------------------

x\_meteor db 19

y\_meteor db 2

color\_meteor db 11

x\_ship db 19

y\_ship db 23

color\_ship db 13

bx\_saver dw ?

rnd db ?

; --------------------------

CODESEG

proc Cursor\_location\_meteor

;set cursor location for meteor

pusha

mov bh, 0

mov dh, [y\_meteor] ;row number

mov dl, [x\_meteor] ;column number

mov ah, 2

int 10h

popa

ret

endp Cursor\_location\_meteor

proc DrawMeteor

;draw meteor

pusha

mov ah,9 ;interrupt code

mov al,15 ;character to display - ASCII code

mov bh, 0h ;page - always 0

mov bl, [color\_meteor] ;BL = foreground

mov cx,1 ;number of times to write character

int 10h ;int 10h Bios

popa

ret

endp DrawMeteor

proc delay

pusha

mov cx, 03h ;High Word

mov dx, 4240h ;Low Word

; mov al, 0

mov ah, 86h ;Wait

int 15h

popa

ret

endp delay

proc random

;generates a random number and keeps it in rnd

pusha

mov bx, [bx\_Saver]

;put segment number in register es

mov ax, 40h

mov es, ax

;move random number to ax

mov ax, [es:6Ch]

xor ax,[bx]

add [bx\_saver],ax

and al, 27h ;check if number < 39, which is 27 in hex

mov [rnd],al

popa

ret

endp random

proc Cursor\_location\_ship

;set cursor location

pusha

mov bh, 0

mov dh, [y\_ship] ;row number

mov dl, [x\_ship] ;column number

mov ah, 2

int 10h

popa

ret

endp cursor\_location\_ship

proc DrawShip

;draw character

pusha

mov ah, 9 ;interrupt code

mov al, 127 ;character to display - ASCII code

mov bh, 0h ;page - always 0

mov bl, [color\_ship] ;BL = foreground

mov cx,1 ;number of times to write character

int 10h ;int 10h Bios

popa

ret

endp DrawShip

;Right key press

proc RightKeyPress

pusha

mov [color\_ship], 0 ;black color

call Cursor\_location\_ship

call DrawShip ; draw black character

add [x\_ship], 1

mov [color\_ship],13

call Cursor\_location\_ship ;move to new location

call DrawShip ; Draw color character

popa

ret

endp RightKeyPress

;Left key press

proc LeftKeyPress

pusha

mov [color\_ship], 0 ;black color

call Cursor\_location\_ship

call DrawShip ; draw black character

sub [x\_ship], 1

mov [color\_ship], 13

call Cursor\_location\_ship ;move to new location

call DrawShip ; Draw color character

popa

ret

endp LeftKeyPress

start:

mov ax, @data

mov ds, ax

; --------------------------

;enter text mode 40 x 25 (x,y)

mov al, 013h

mov ah, 0

int 10h

call Cursor\_location\_meteor

call DrawMeteor

WaitForCharacter:

; Wait for key

mov ah, 0h

int 16h

; Was Esc pressed?

cmp al, 27

je TheEnd

; Was p pressed?

cmp al, 'p'

je Move

jmp WaitForCharacter

Move:

mov [color\_meteor], 0

call Cursor\_location\_meteor

call DrawMeteor

inc [y\_meteor]

cmp [y\_meteor], 25

je changePlace

call Cursor\_location\_meteor

; check the character on coordinates

mov bh, 0h ; Page=1

mov ah, 08h ; Read character function

int 10h ;return the character value to AL

cmp al, 127

je TheEnd

mov [color\_meteor], 11

call DrawMeteor

call Cursor\_location\_ship

call DrawShip

; check if there is a character to read

mov ah, 1h

int 16h

jz noKey

; read a character and save it to al

mov ah, 0h

int 16h

cmp al, 27 ; if esc

je theEnd

;Was right key pressed?

cmp ah,4dh

je RightKey

;Was left key pressed?

cmp ah,4bh

je LeftKey

noKey:

call delay

jmp Move

RightKey:

call RightKeyPress

jmp Move

LeftKey:

call LeftKeyPress

jmp Move

changePlace:

call random

mov al, [rnd]

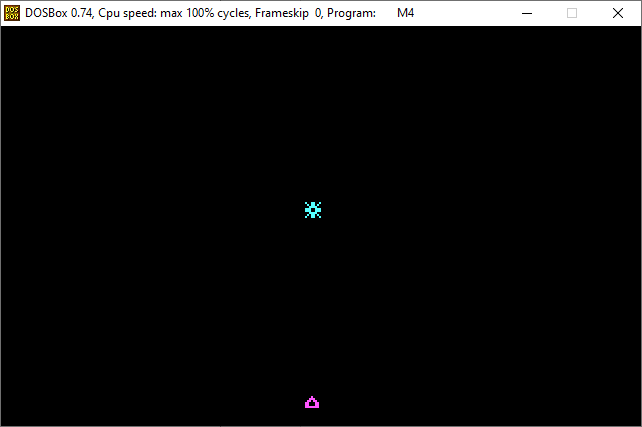
mov [x\_meteor], al

mov [y\_meteor], 2

jmp Move

TheEnd:

; Return text mode 80x25 (x,y)

 mov al, 03h

mov ah, 0

int 10h

; --------------------------

exit:

mov ax, 4c00h

int 21h

END start