```
1  .const
2  acr equ 1    ;acceleration irregardles of direction
3  floorY equ 400  ;y position of the floor (as in max value of the y character)
4  roofY equ 100   ;y position of the roof (as in min value of the y character)
5  widthbg equ 1076    ;width of the background picture
6  heightbg equ 579    ;height of the background picture
7
8  edgeObst equ -35     ;should be a bit less than 0 (reason in movObst function)
9
10 maxScore equ 9999   ;max possible score
11
12 difficulty equ 2    ;higher the value - easier the game (more explaination in
     game.inc)
13 slowPoints equ 2    ;higher the value - slower the score goes up
14 updatePace equ 35   ;higher the value - slower update procedure (score,
     movement and animation pace)
15 runAnimPace equ 100 ;higher the value - slower run animation
16
17 ;Resources struct contains the paths to pictures on the hard disk and Img
     structs to store the pictures in the ram
18 Resources struct
19     bgpath byte "pics\backg.png",0              ;background picture
20     dngrPath byte "pics\Zapper.png",0           ;zapper picture
21     deadmsgPath byte "pics\deadmsg.png",0       ;message that appears after
         user dies
22     dgtStripPath byte "pics\digitStrip.png",0   ;contains a strip of all
         decimel digits
23     highScorePath byte "pics\High Score.png",0  ;contains the text "HIGH
         SCORE:"
24     bg Img<>
25     dngr Img<>
26     deathMsg Img<>
27     dgtStrip Img<>
28     highScore Img<>
29 Resources ends
30
31
32 Score struct
33     valu dword ?        ;value of the score
34     ones dword ?        ;ones digit
35     tens dword ?        ;tens digit
36     hndrds dword ?      ;hundreds digit
37     thou dword ?        ;thousands digit
38 Score ends
39
40 ;path to cat character photos on hard disk
41 HDcat struct
42     fly byte "pics/CATFLY.png",0        ;strip of fly animation
43     run byte "pics/CATRUN.png",0        ;strip of run animation
```

```
44 HDcat ends
45
46 ;will contain the pictures in the ram of the cat character
47 RAM struct
48     fly Img<>
49     run Img<>
50 RAM ends
51
52 ;object size and placement
53 objPos struct
54     x dword ?       ;x position
55     y dword ?       ;y position
56     wdth dword ?    ;width
57     height dword ?  ;height
58 objPos ends
59
60
61
62 .data
63
64 obst1 objPos<1076,150,30,119>       ;zapper 1
65 obst2 objPos<1500,340,30,119>       ;zapper 2
66 charPos objPos<100,floorY,32,53>    ;cat character
67
68
69 slowUpdate dword 0      ;used to slow update procedure
70 slowBg dword 0          ;used to slow the pace of background
71 slowAnim dword 0        ;used to slow running animation
72 slowScore dword 0       ;used to slow score
73
74 xbg dword 0             ;used for background animation (describes how much to
     crop from background pic)
75
76 spd dword 0             ;speed of character(y axle)
77
78 shmulik HDcat<>         ;paths to character pictures in hard disk
79 shmulRM RAM<>           ;character pictures in ram
80 res Resources<>
81
82 recXFly dword 0         ;describes which animation state cat is in (if in air)
83 recXRun dword 0         ;describes which animation state cat is in (if on
     floor)
84
85
86 ;the following arrays include instructions of cropping specific pictures out
     of strips:
87 ;picture n: x coordinate of start of picture in strip is in idx(n*2), width of
     picture in idx(n*2+1)
88 ;(since the arrays contain: dwords x coordinate=idx(n*8), width=idx(n*8+1))
```

```
 89  catFlies dword 4 dup(0,60,60,55)                                    ;cat fly
       animation strip
 90  catRuns dword  8 dup(0,47,52,41,109,48,161,40,207,54)               ;cat run
       animation strip
 91  dgtArr dword 20 dup
       (0,20,26,11,44,20,70,17,88,20,112,18,132,20,156,18,177,18,199,19)   ;digit
       strip (digit = idx(digit*8+1))
 92
 93  catDead byte 0           ;boolean describes if cat is dead (0=ALIVE,1=DEAD)
 94
 95  current Score<0,0,0,0,0>    ;score of current run
 96  HISC Score<0,0,0,0,0>       ;high score
 97
 98
 99
100
```