

Elektrotehnicki fakultet

Katedra za elektroniku

Projekat iz Racunarske elektronike:

Pocket Tanks

Autori:

Stefan Jović 167/2014

Nemanja Višnjic 410/2014

15.8.2017.

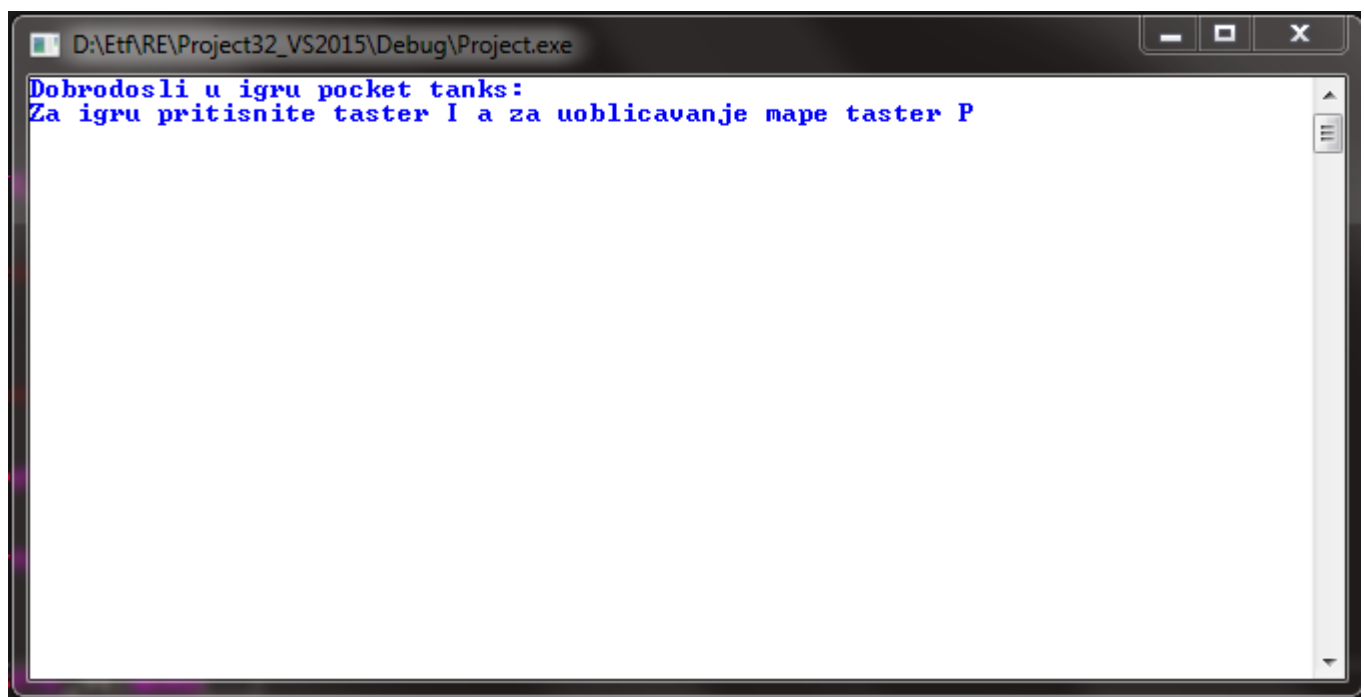
Uvod

Cilj projekta bila je realizacija igrice pocket tanks.

Igra je podeljena na dva dela, izradu mape na kojoj će se odvijati borba tenkova, i sama igra. U toku borbe, igrač kontroliše kretanje tenka i ugao pod kojim se ispaljuje granata u cilju uništavanja protivničkog tenka. Svaki tenk ima 10 života. Tenk koji prvi ostane bez života je izgubio. U igrici postoji model gravitacije koja utiče i na tenkove i na delove mape.

Projekat je realizovan u Assembler programskom jeziku za Intel x86 arhitekturu procesora.

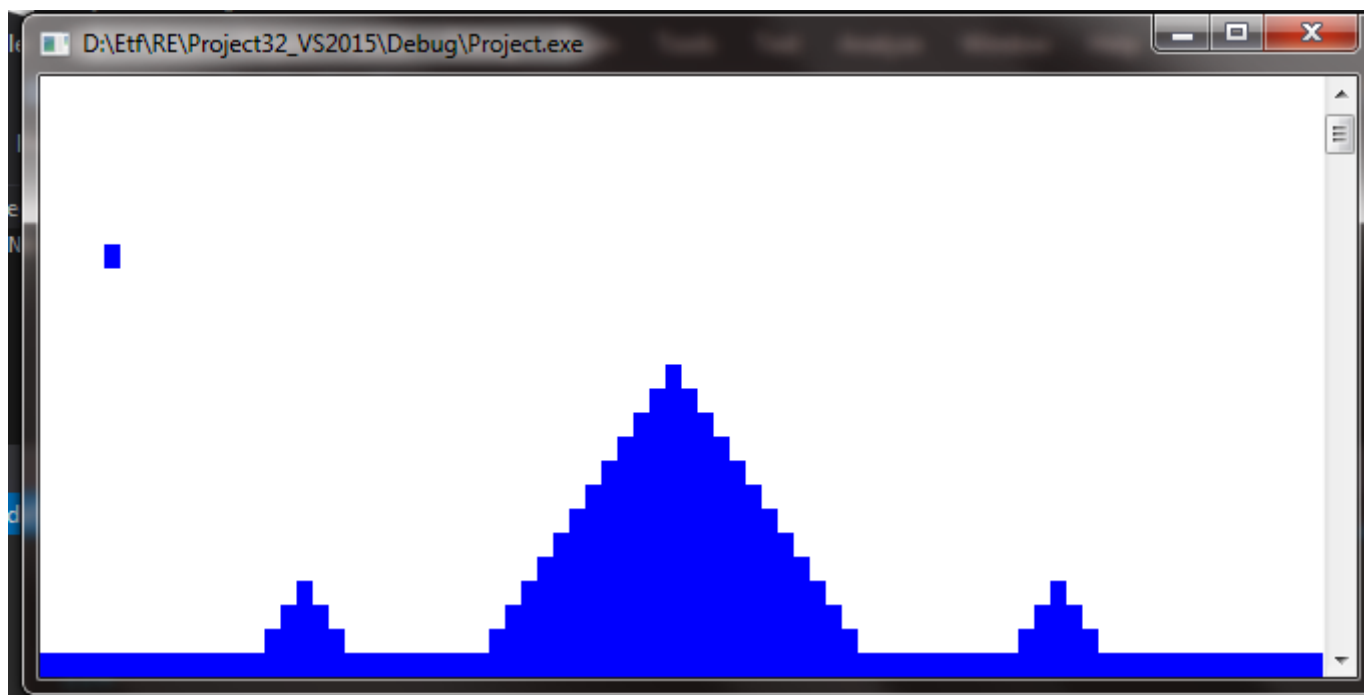
Realizacija



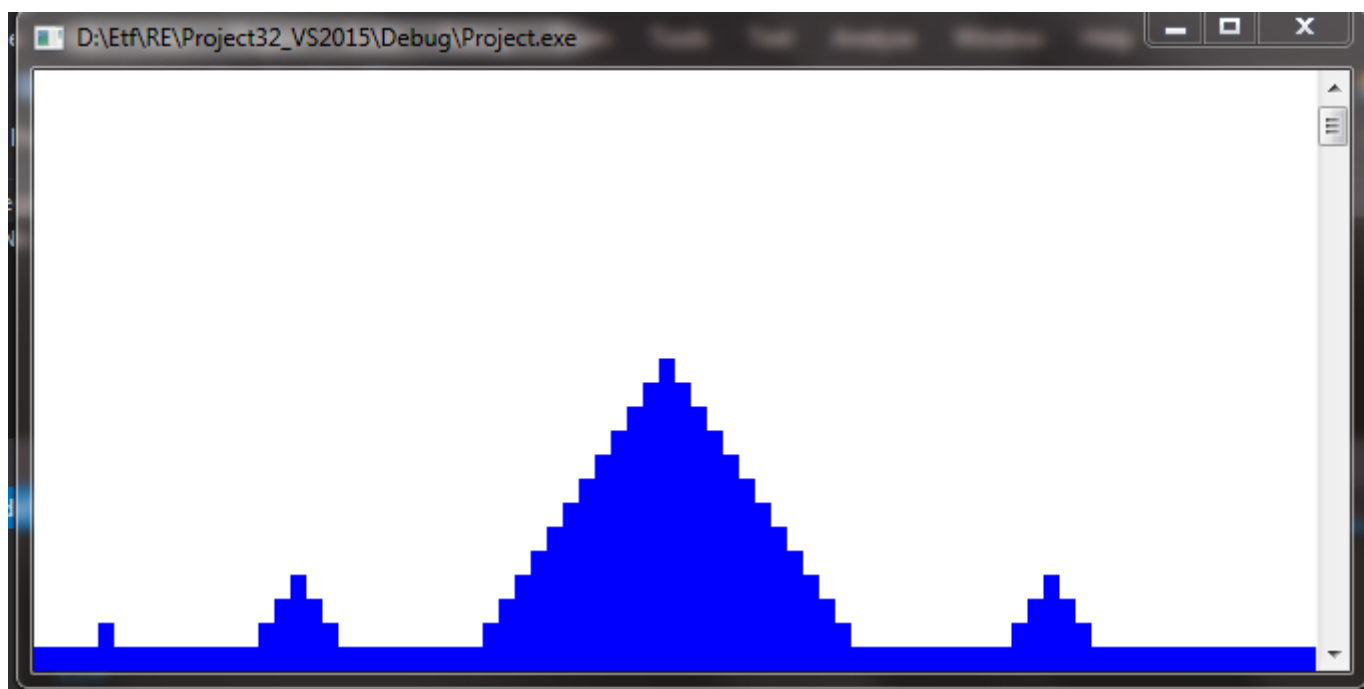
Pravljenje mape

U ovom segmentu igrači imaju mogućnost da oblikuju bojište na kome će se boriti njihovi tenkovi. Iz početnog ekrana pritiskom na taster P ulazi se u editor mape. Editor mape otvara mapu koja već postoji u "output.txt" fajlu, i omogućiti njeno prepravljanje. Kroz editor se kreće kursorom koji se pomera pomoću strelica. Kursor se nikada ne može naći u prva dva reda konzolnog ekrana jer su oni rezervisani za ispis života i ugla topa, takođe iako se može naći u trećem i četvrtom redu, u njima se blokovi ne mogu postavljati jer su ta dva reda rezervisana za pojavljivanje tenka (ako neko popuni celu mapu). Blok se postavlja pritiskom na taster "space" dok se briše priztiskom na taster "D". U editoru postoji model gravitacije i pri postavljanju novog bloka i pri brisanju starih.

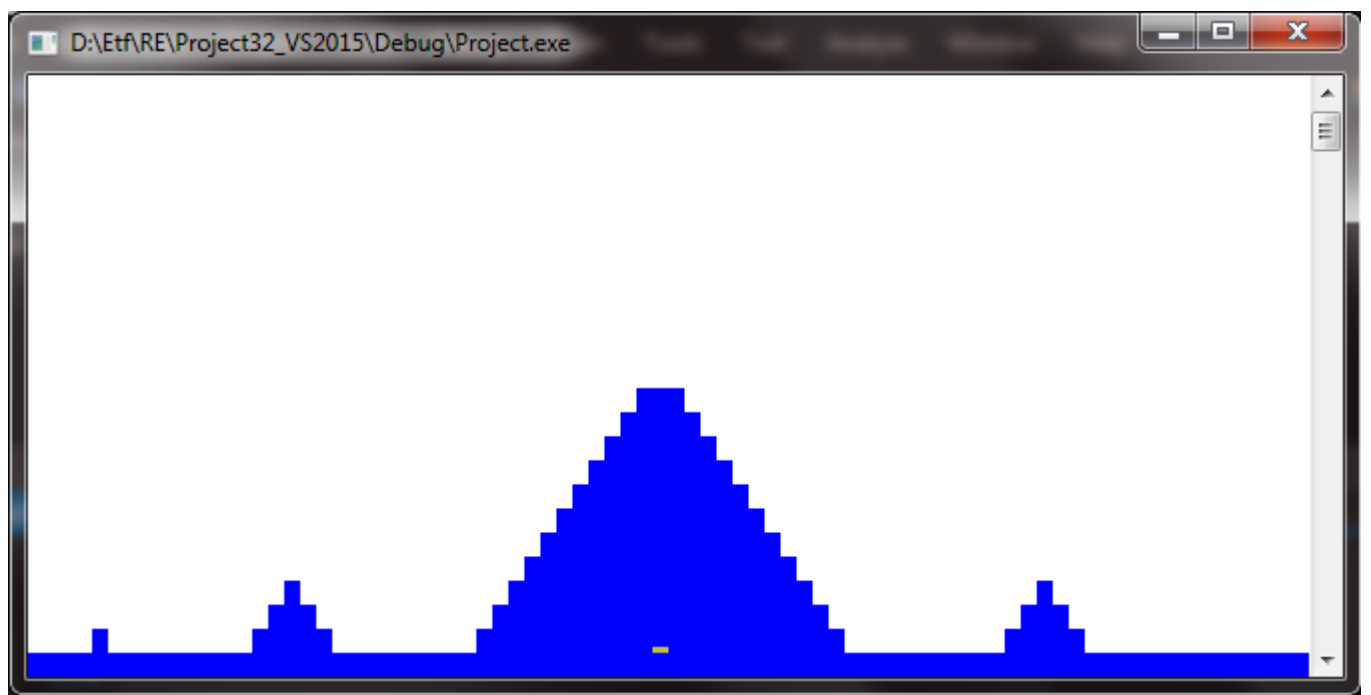
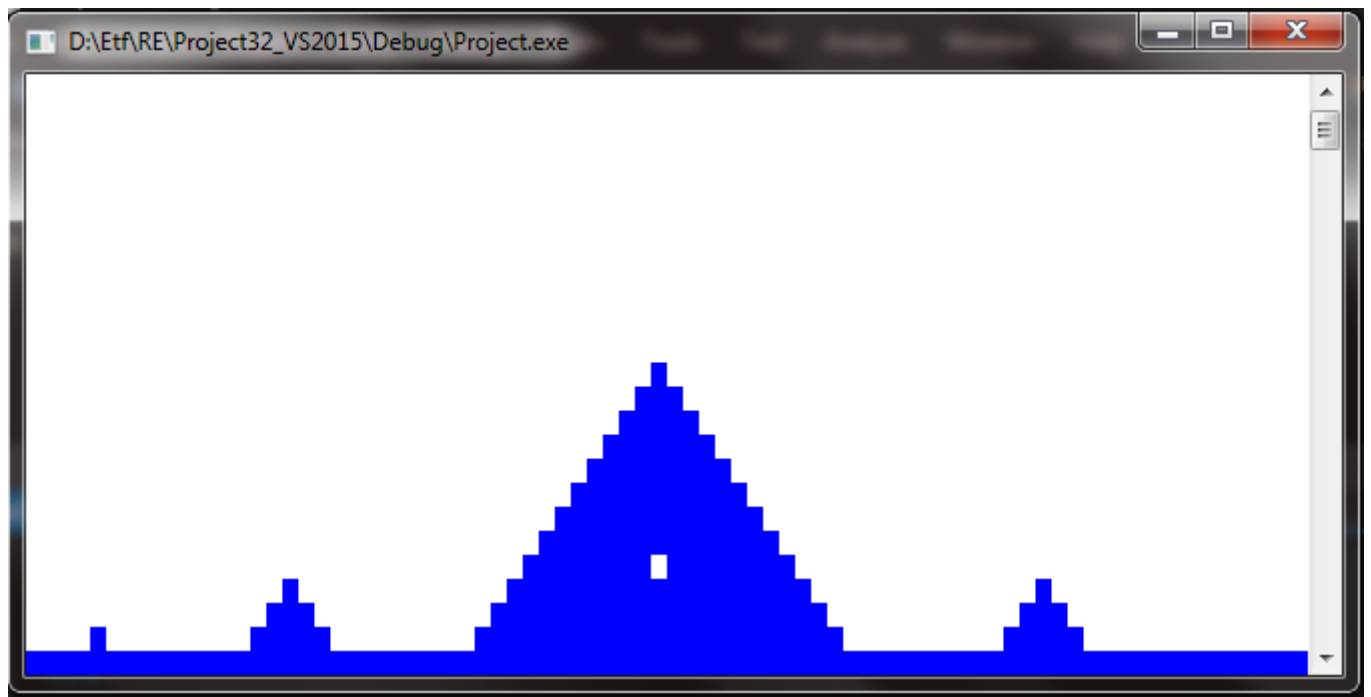
Vidimo da se blok postavljen udaljen od "zemlje"



spušta sve dok je ne dotakne:



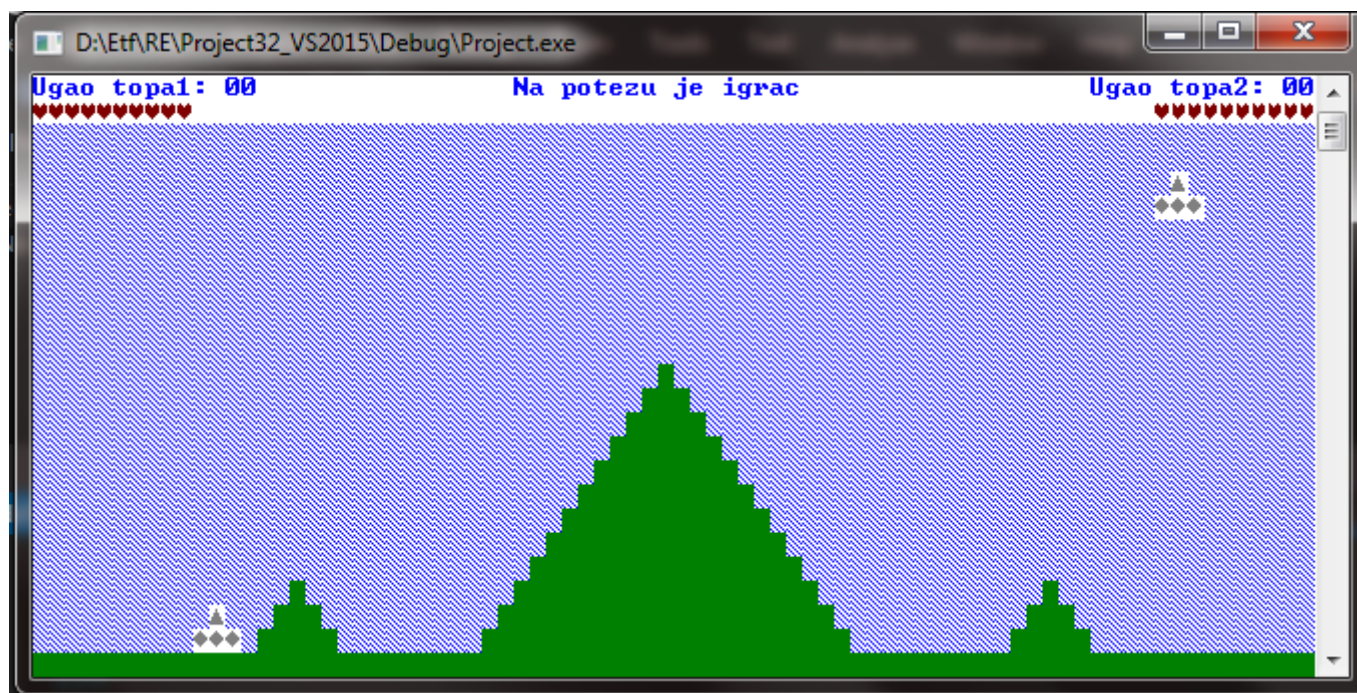
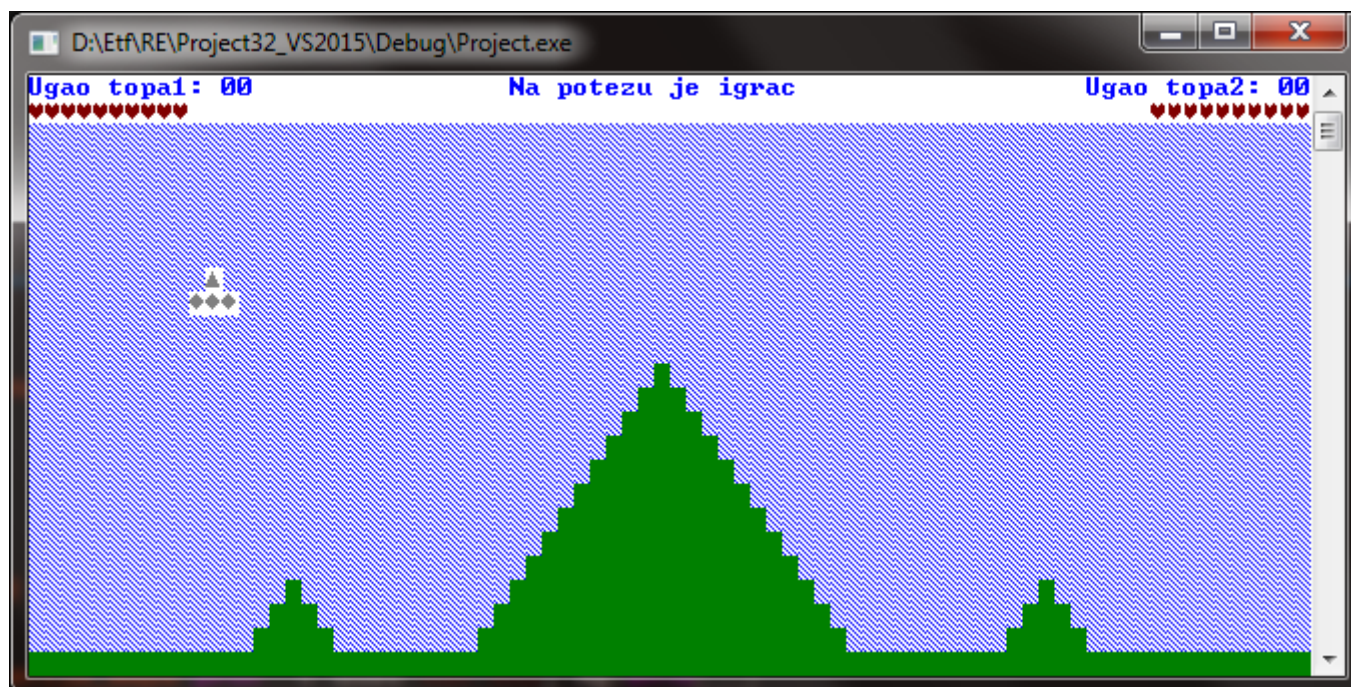
Pri brisanju takođe možemo primetiti da se isto dešava:

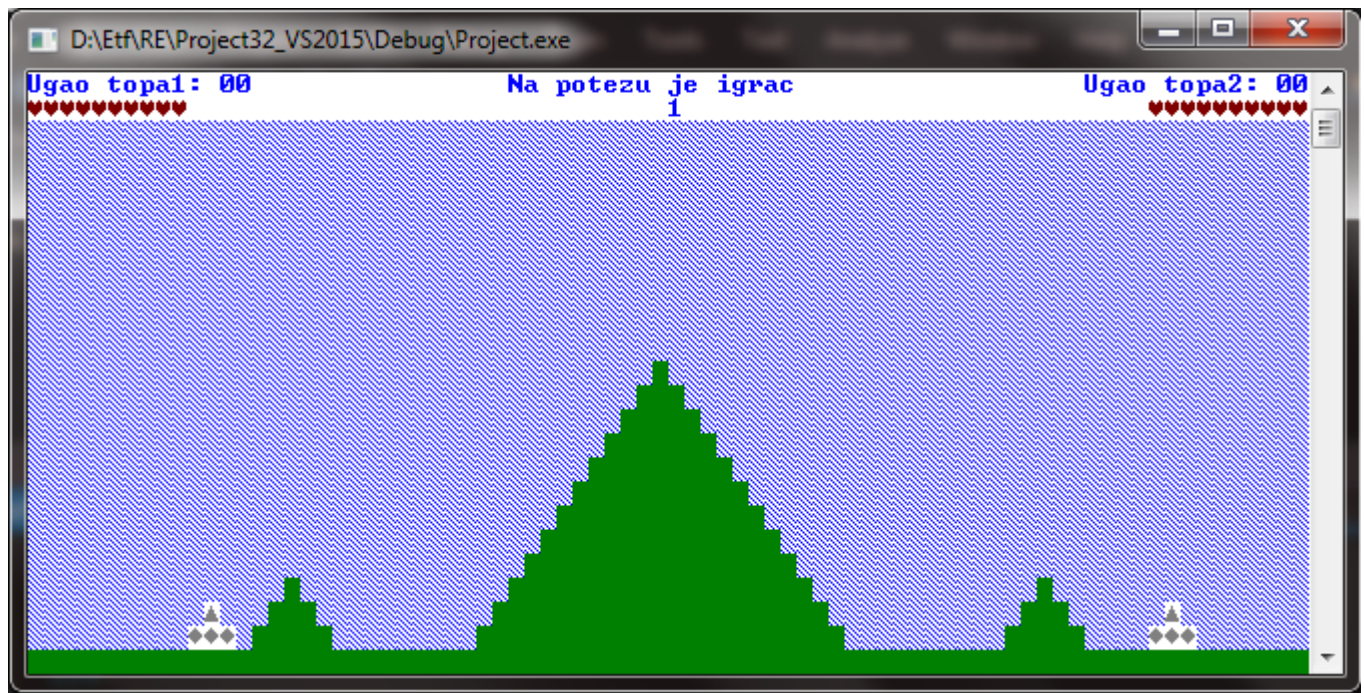


Na kraju, pritiskom na taster "ESC" mapa se snima u "output.txt" i vraća se na glavni meni.

Igra

Igra počinje tako što se tenkovi naizmenično spuštaju sa vrha ekrana:





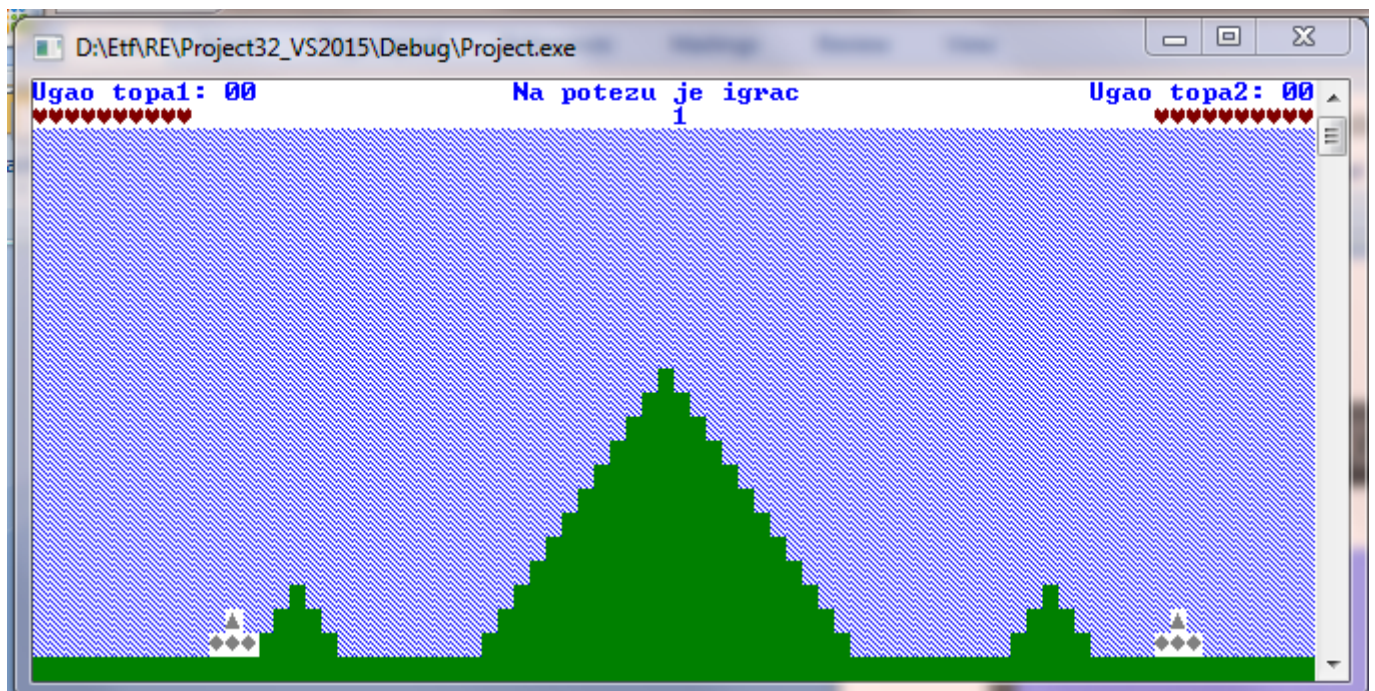
Nakon toga, na potezu je igrač1.

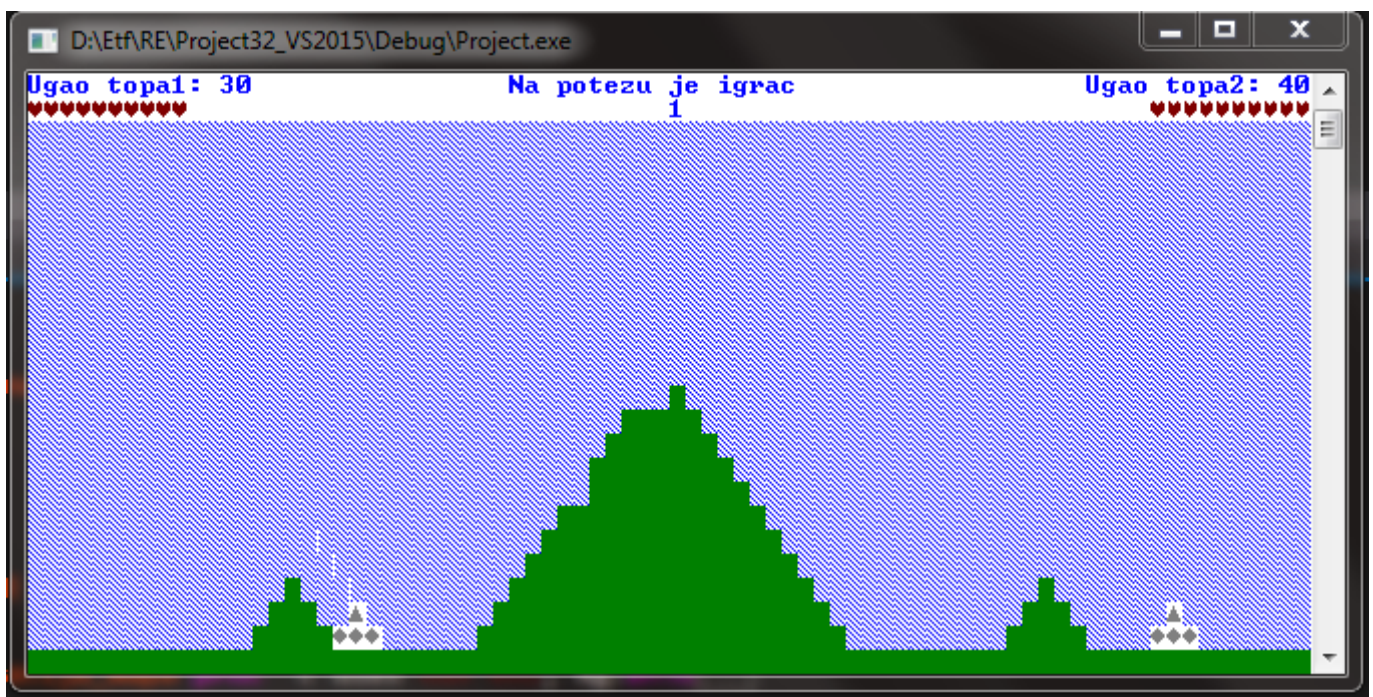
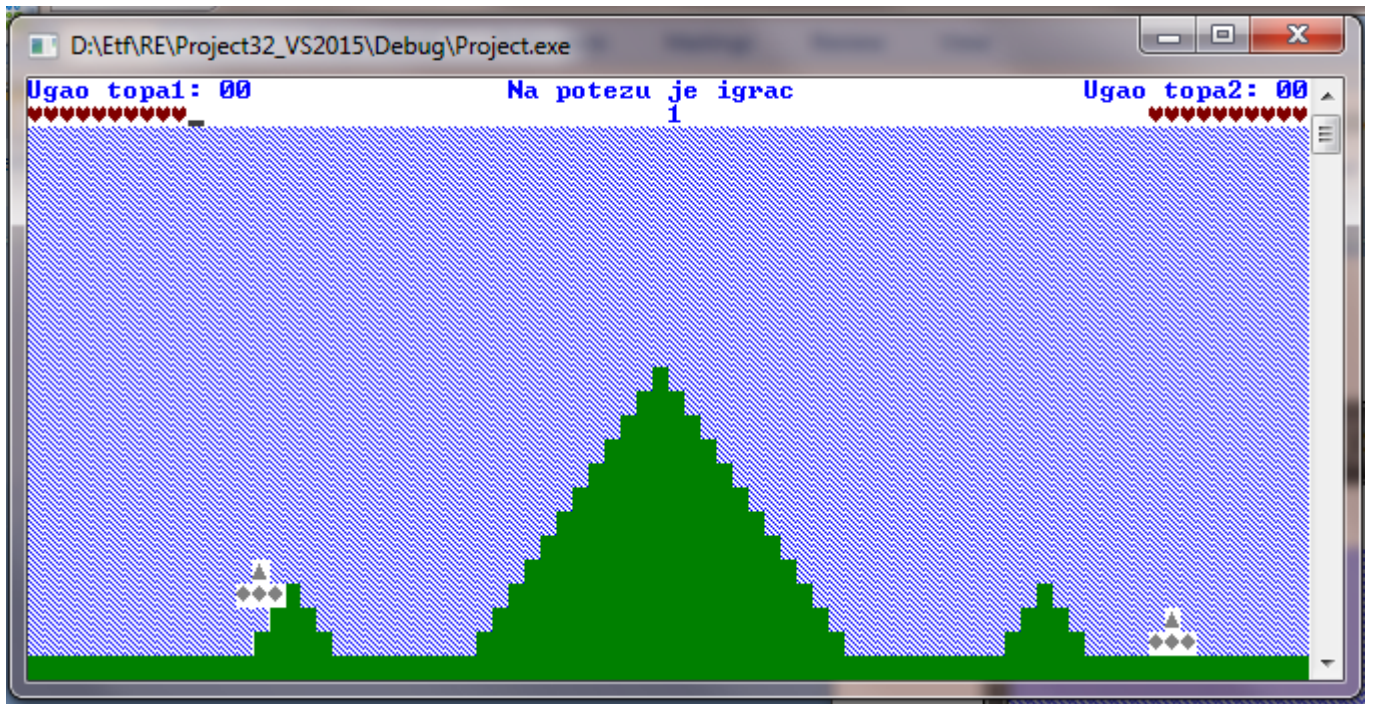
Postoje tri radnje koje igrač može da preduzme:

1. Kretanje
2. Pucanje
3. Podešavanje snage

Kretanje

Svaki igrač može napraviti po 3 pokreta po potezu. Tenk se može kretati levo ili desno uz određene specijalne slučajeve. Naime, tek može penjati na prepreke koje nisu više od jednog bloka, takođe, po istom pravilu, može i silaziti.



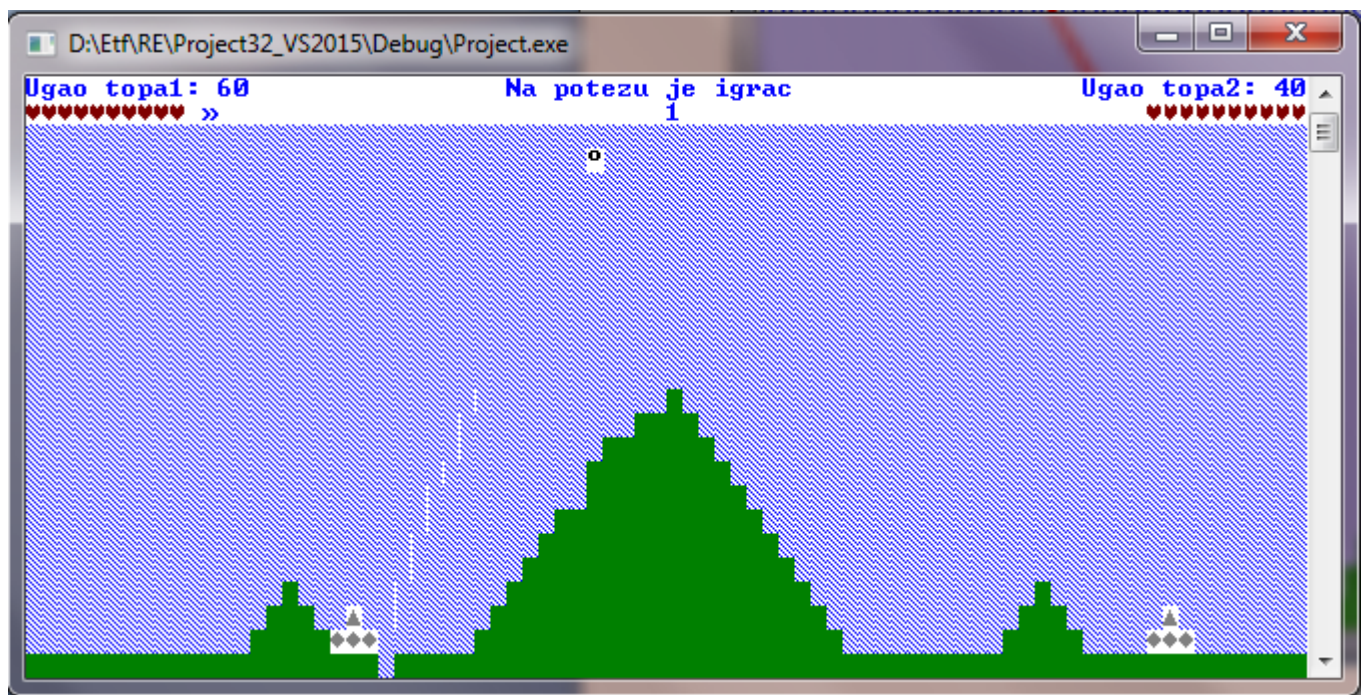


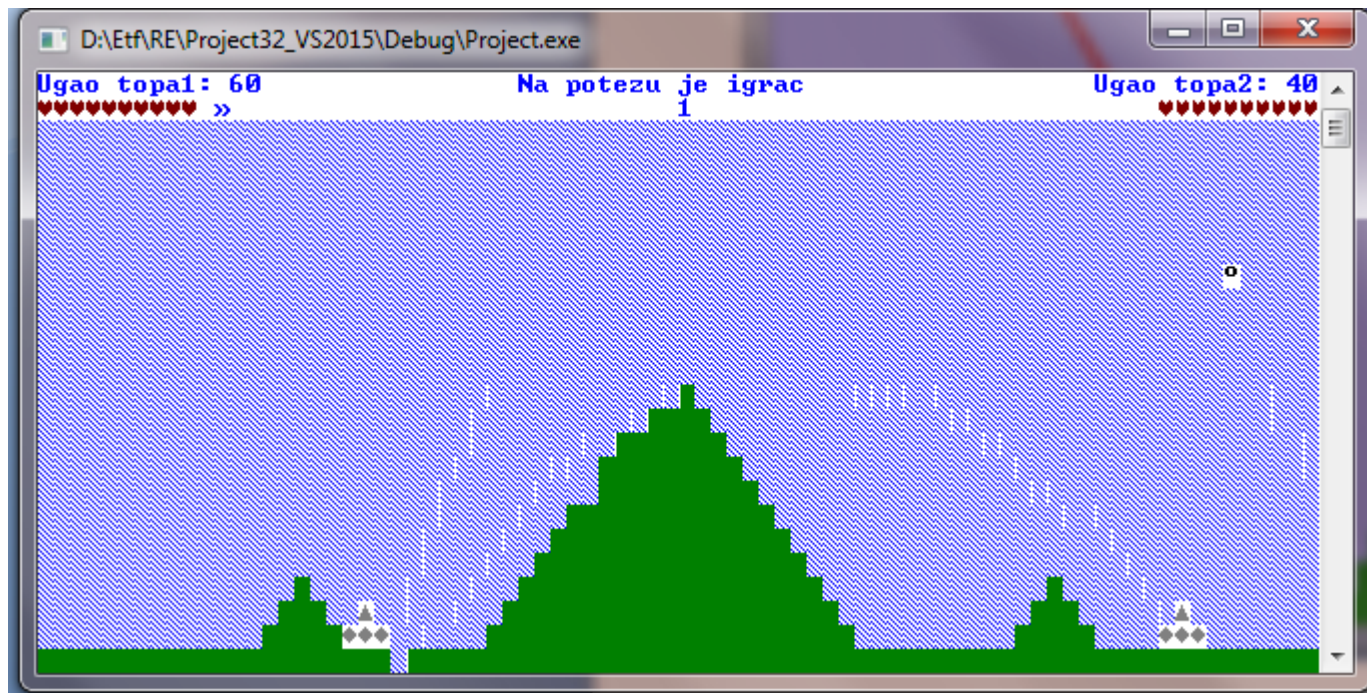
Pucanje i podešavanje snage

Strelicama na gore i dole se podešava ugao topa, koji se može kretati od -80° do 80° sa korakom od po 10° . Na taster "+" se uključuje "pojačanje" snage topa, pa granata više leti, dok se na "-" isključuje. Ova opcija je igraču vidljiva pored broja života kao na slici:

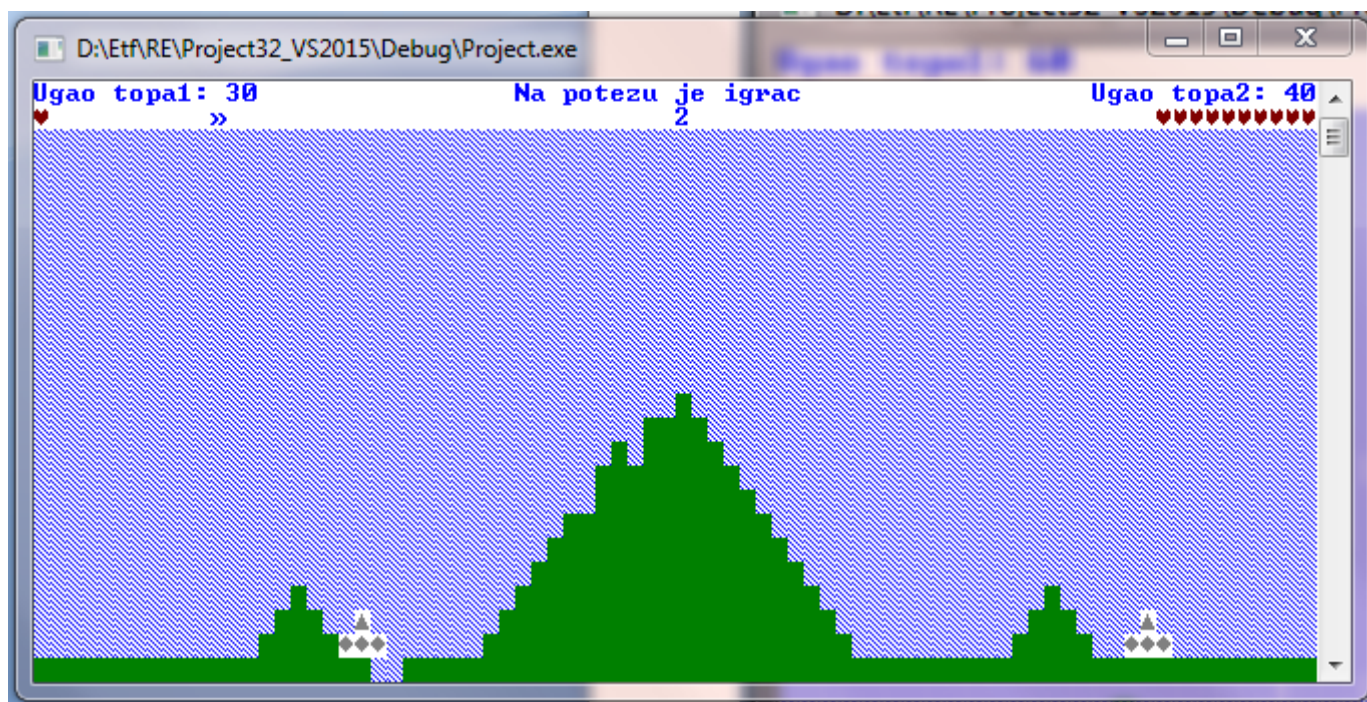


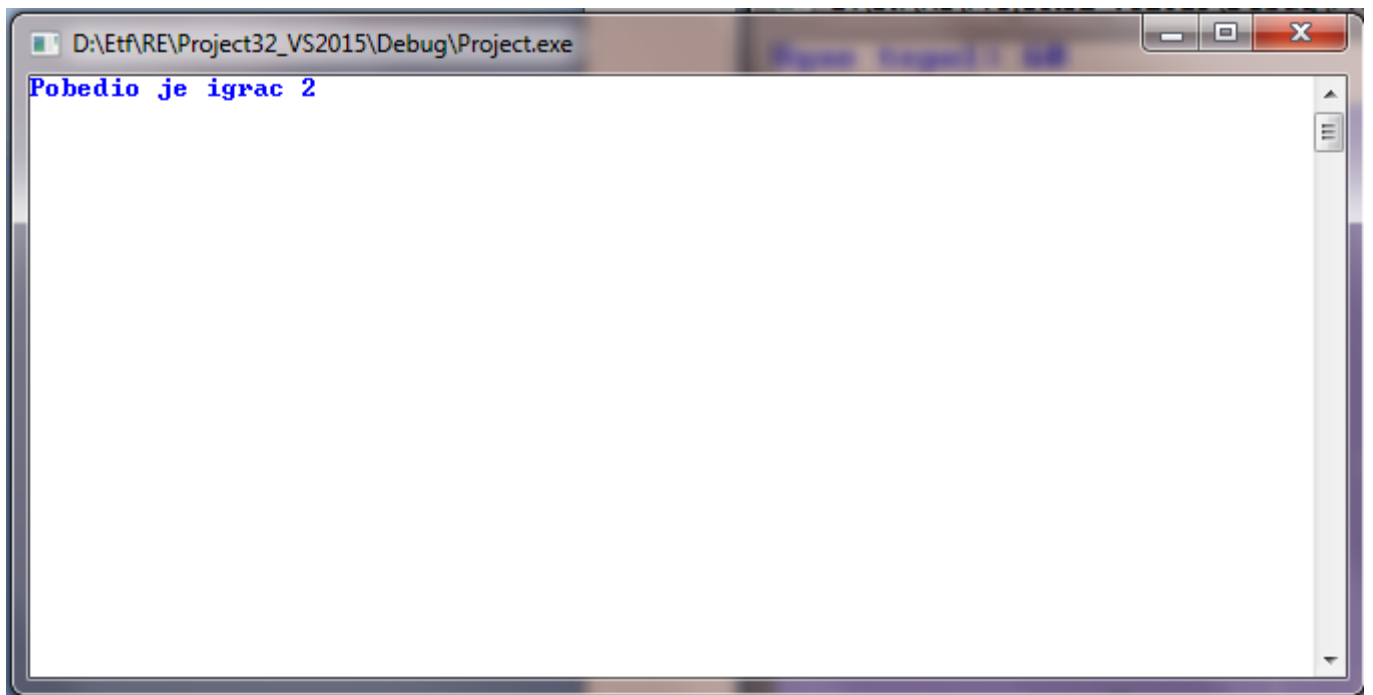
Pritiskom na taster "space", granata se ispaljuje. Ako granata pogodi deo mape, taj deo mape nestaje, kao na slici iznad, i aktiviraju se, po potrebi, modeli gravitacije, koji urušavaju mapu, tako da delovi mape ne vise, ili spuštaju tenkove ako je nestalo tlo ispod njih. Ukoliko granata pogodi bočne zidove konzole ili dno, granata nestaje. Prilikom prolaska kroz vrh ekrana granata privremeno nestaje iz vida, ali održava proračunatu putanju na osnovu koje se može vratiti u mapu.





Kada granata pogodi tenk, tenk gubi život. Kada tenk ostane bez svih života, igra se završava i ispisuje se pobjednik. Nakon nekog vremena, igra se vraća na glavni meni.





Igra se isključuje pritiskom na taster "ESC" u glavnom meniju.

Kod

```
INCLUDE Irvine32.inc
INCLUDE macros.inc

.386
.model flat,stdcall
.stack 9192
ExitProcess proto,dwExitCode:dword

.data
    donja_granica BYTE 16h
    X BYTE 0
    Y word 0
    xt1 BYTE 1
    yt1 WORD 0ah
    xt2 BYTE 1
    yt2 WORD 46h
    xgr BYTE 0
    ygr WORD 0
    x_bilo_0 BYTE 0
    ugao1 WORD 0h
    ugao2 WORD 0h
    step BYTE 0Ah
    Y1 BYTE 0
    zed WORD 256d
```

```

pom WORD 50h
niz byte 2240 DUP (0)
kasnjenje BYTE 30d
zivot1 BYTE 10
zivot2 BYTE 10
snaga1 BYTE 0
snaga2 BYTE 0
igra BYTE 0
ugao_znak1 BYTE 0h
ugao_znak2 BYTE 0h
potez BYTE 1
str1 BYTE "Ugao topa1:", 0
str2 BYTE "Ugao topa2:", 0
karakter BYTE 0
str3 BYTE "      ",0
str4 BYTE "Pobedio je igrac 1",0
str5 BYTE "Pobedio je igrac 2",0
str6 BYTE "Na potezu je igrac",0
str7 BYTE "Dobro dosli u igru pocket tanks:",0dh,0ah,0
str8 BYTE "Za igru pritisnite taster I a za uobicavanje mape taster P",0dh,0ah,
        "a za izlazak iz igre taster ESC",0dh,0ah,0
str9 BYTE "Pravila igre:",0dh,0ah,
        "Svaki tenk ima po 10 zivota a gubi onaj koji prvi ostane bez zivota.",0dh,0ah,
        "Tenk se moze kretati levo ili desno pomocu strelica.",0dh,0ah,
        "Maksimalo je dozvoljeno tri kretanja po potezu.",0dh,0ah,
        "Tenk se moze peti ili spustati po jedan blok i prelaziti rupe sirine dva.",0dh,0ah,0
str10 BYTE "Ugao topa se podesava od -80 do 80 stepeni sa korakom 10 stepeni pomocu",0dh,0ah,
        "strelica.",0dh,0ah,
        "Granata se ispaljuje na space.",0dh,0ah,
        "Ako granata pogodi u deo mape gde je prepreka prepreka nestaje.",0dh,0ah,
        "U igrici postoji gravitacija tako da se segmenti mape obrusavaju a ne ostaju da vise.",0dh,0ah,0
str11 BYTE "Takodje se i tenkovi obrusavaju ako nestane tlo pod njima.",0dh,0ah,
        "Granata nestaje ako udari bocne granice prozora kao i njegovo dno.",0dh,0ah,
        "Granata nastavlja da leti po trajektoriji ako pogodi vrh prozora",0dh,0ah,0
str12 BYTE "Na + se aktivira pojacan let granate a na - se deaktivira",0dh,0ah,
        "Pritisnite bilo sta za pocetak igre.",0dh,0ah,0
str13 BYTE "Pravljenje mape:",0dh,0ah,
        "Strelicama se krece po mapi",0dh,0ah,
        "Na space se postavljaju blokovi",0dh,0ah, 0
str14 BYTE "Blokovi se ne mogu postaviti u prva cetiri reda" ,0dh,0ah,
        "Na blokove utice gravitacija, tako da ne mogu da vise" ,0dh,0ah,
        "Blokovi se brisu pritiskom na taster D" ,0dh,0ah,
        "Pritisnite taster ESC kako biste sacuvali mapu i vratili se u glavni meni" ,0dh,0ah,
        "Pritisnite bilo koji taster da biste zapoceli pravljenje mape",0
filename  BYTE "output.txt",0
fileHandle  HANDLE ?
stringLength DWORD 2240d
bytesWritten DWORD ?

```

.code

```
snimanje proc c uses eax edx ecx
```

```

mov edx,OFFSET filename
call CreateOutputFile

```

```
mov fileHandle,eax
```

```
mov     eax,fileHandle
    mov     edx,OFFSET niz
    mov     ecx,stringLength
    call    WriteToFile
mov     eax,fileHandle
    call    CloseFile
xor eax,eax
ret
snimanje endp
ucitavanje proc
```

```
mov     edx,OFFSET filename
    call    OpenInputFile
    mov     fileHandle,eax

mov     edx,OFFSET niz
    mov     ecx,stringLength
    call    ReadFromFile
mov     eax,fileHandle
    call    CloseFile
```

```
ucitavanje endp
```

```
koordinator proc c uses eax
```

```
mov ax, pom
mul X
add ax , Y
mov zed, ax
ret
koordinator endp
```

```
CRTANJE proc c uses eax esi, Xc:BYTE, Yc:BYTE, Yc1:WORD
```

```
xor eax, eax
mov Xc, al
mov Yc, al
mov Yc1, ax
call Clrscr
cmp igra,00
je produzi2
mov dl,1eh
mov dh,00h
call gotoxy
mov eax,lightblue + (white * 16)
call SetTextColor
mov edx, OFFSET str6
```



```

call writestring
mov dl,00h
    mov dh,00h
    call gotoxy
    mov eax,lightblue + (white * 16)
        call SetTextColor
    mov edx,OFFSET str1

    call WriteString

    mov al, ugao_znak1
    cmp al, 0h
    je pisi_poz

    mov ax, 02Dh
    jmp pisi_kon
pisi_poz: mov al , 00h

pisi_kon: call WriteChar
    mov ax,ugao1
    mov dl, 0Ah
    div dl
    add al, 030h
    call WriteChar
    mov al, 030h
    call WriteChar ; završava se pisanje "UGAO TENKA"

    mov dl,00h
    mov dh,01h
    call Gotoxy
    mov eax,red + (white * 16)
        call SetTextColor
    mov edx,OFFSET str3
    call WriteString

    xor ebx,ebx
    mov dl,00h
    mov dh,01h
    call Gotoxy
    mov al,03h
    mov bl,zivot1
opet:cmp bl,00
    je produzi
    call WriteChar
    inc dl
    call Gotoxy
    dec bl
    jmp opet

produzi: mov dl,42h
    mov dh,00h
    call gotoxy
    mov eax,lightblue + (white * 16)
        call SetTextColor
    mov edx,OFFSET str2

```

call WriteString

mov al, ugao_znak2
cmp al, 0h
je pisi_poz1

mov ax, 02Dh
jmp pisi_kon1
pisi_poz1: mov al, 00h

pisi_kon1: call WriteChar
mov ax, ugao2
mov dl, 0Ah
div dl
add al, 030h
call WriteChar
mov al, 030h
call WriteChar ; završava se pisanje "UGAO TENKA"

mov dl, 46h
mov dh, 01h
call Gotoxy
mov eax, red + (white * 16)
call SetTextColor
mov edx, OFFSET str3
call WriteString

xor ebx, ebx
mov dl, 4fh
mov dh, 01h
call Gotoxy
mov al, 03h
mov bl, zivot2
opet1: cmp bl, 00h
je produzi1
call WriteChar
dec dl
call Gotoxy
dec bl
jmp opet1

produzi1: cmp snaga1, 00
je dalje
mov dl, 0bh
mov dh, 01h
call gotoxy
mov eax, lightblue + (white * 16)
call SetTextColor
mov al, 0afh
call writechar
dalje: cmp snaga2, 00
je produzi2
mov dl, 044h
mov dh, 01h
call gotoxy
mov eax, lightblue + (white * 16)

```
    call SetTextColor
    mov al,0aeh
    call writechar
```

produzi2:.repeat

```
.repeat
    mov al,Xc
    add al,02h
    mov dh, al
    mov dl, Yc
    mov ax, pom
    mul Xc
    add ax , Yc1
    mov zed, ax
    mov esi, OFFSET niz
    mov eax, esi
    add ax, zed
    mov esi, eax
    cmp igra,00h
    je crtprav
    mov al,[esi]
    cmp al,00h
    je nista
    cmp al, 01h
    je znak0
    cmp al,02h
    je znak1
    cmp al,03h
    je znak2
    znakp: inc Yc1
           inc Yc
```

```
.until Yc > 4fh
xor eax, eax
mov Yc,al
mov Yc1, ax
inc Xc
mov al, donja_granica
```

.until Xc > al

jmp kraj

```
znak0:mov eax,green + (white * 16)
      call SetTextColor
```

```
    mov al, 0DBh
    call Gotoxy
    call WriteChar
    jmp znakp
```

```
znak1:mov eax,gray + (white * 16)
```

```
      call SetTextColor
    mov al, 04h
    call Gotoxy
    call WriteChar
```

```

    jmp znakp
znak2:mov eax,gray + (white * 16)
    call SetTextColor
    mov al, 1eh
    call Gotoxy
    call WriteChar
    jmp znakp
nista:mov eax,LightBlue + (white * 16)
    call SetTextColor
    mov al, 0b0h
    call Gotoxy
    call WriteChar
    jmp znakp

```

```

crtprav:mov al,[esi]
    cmp al, 01h
    je znak0p
    jmp znakp

```

```

znak0p:mov al, 0DBh
    call Gotoxy
    call WriteChar
    jmp znakp

```

```

xor eax, eax
mov Xc, al
mov Yc, al
call Gotoxy
kraj:
    mov dl, 0h
    mov dh, 0h
    call Gotoxy
ret

```

CRTANJE endp

```

gravitacija_mapa proc c uses eax esi, Xg:BYTE
mov al, X
mov Xg, al
cmp al,donja_granica
je kraj
poc:mov ax, pom
    mul Xg
    add ax , Y
    mov zed, ax

```

```

mov esi, OFFSET niz
    mov eax, esi
    add ax, pom
    add ax, zed
    mov esi, eax
    mov al, [esi]
    cmp al, 0h
    je pad

```


kraj: ret

pad: mov al, 01h
mov[esi], al
mov eax, esi
sub ax, pom
mov esi, eax
mov al, 00h
mov[esi], al
inc Xg
xor eax, eax
mov al, kasnjenje
call Delay
call CRTANJE
mov al, Xg
cmp al, donja_granica
je kraj
jmp poc

gravitacija_mapa endp

gravitacija_brisanje proc c uses eax esi, Xg2:BYTE, Yg2:WORD

cmp igra, 00h
je crt
mov al, xgr
mov Xg2, al
cmp al, 00h
je kraj
mov ax, ygr
mov Yg2, ax
mov dh, xgr
mov ax, ygr
mov dl, al
add dh, 02h
CALL Gotoxy

crtpov:

poc: mov ax, pom
mul Xg2
add ax, Yg2
mov zed, ax

mov esi, OFFSET niz
mov eax, esi
sub ax, pom
add ax, zed
mov esi, eax
mov al, [esi]
cmp al, 01h

```
je pad
jmp kraj
```

```
pad: mov al, 00h
      mov[esi], al
      mov eax, esi
      add ax, pom
      mov esi, eax
      mov al, 01h
      mov[esi], al
```

```
dec Xg2
```

```
xor  eax, eax
mov  al, kasnjenje
call Delay
call CRTANJE
mov  al, Xg2
cmp  al, 00h
je   kraj
jmp  poc
```

```
crt: mov al,X
      mov Xg2,al
      mov ax,Y
      mov Yg2,ax
      jmp crtpov
      kraj:call CRTANJE
      ret
```

```
gravitacija_brisanje endp
```

```
brisitenk proc c uses eax esi, Xt:byte, Yt:word
```

```
cmp potez,1h
je uzmi prvi
MOV al, xt2
mov xt, al
mov ax, yt2
mov yt,ax
jmp obrada
```

```
uzmi prvi: MOV al, xt1
            mov xt, al
            mov ax, yt1
            mov yt,ax
```

```
obrada: mov dh, xt
         add dh, 02h
         mov ax, yt
         mov dl, al
         mov eax,LightBlue + (white * 16)
         call SetTextColor
```

```
mov al,0b0h
call Gotoxy
call WriteChar
inc dl
call Gotoxy
call WriteChar
inc dl
call Gotoxy
call WriteChar
dec dl
dec dh
call Gotoxy
call WriteChar
xor eax,eax
mov al, kasnjenje
call Delay
```

```
ret
brisenk endp
```

pisitenk proc c uses eax esi, Xt:byte, Yt:word

```
cmp potez,1h
je uzmi prvi
MOV al, xt2
mov xt, al
mov ax, yt2
mov yt,ax
jmp obrada
```

uzmi prvi: MOV al, xt1

```
    mov xt, al
    mov ax, yt1
    mov yt,ax
```

obrada: mov dh, xt

```
    add dh, 02h
    mov ax, yt
    mov dl, al
    mov eax,gray + (white * 16)
    call SetTextColor
    mov al,04h
    call Gotoxy
    call WriteChar
    inc dl
    call Gotoxy
    call WriteChar
    inc dl
    call Gotoxy
    call WriteChar
    dec dl
    dec dh
    mov al, 01eh
```

```
call Gotoxy
call WriteChar
xor eax,eax
mov al, kasnjenje
call Delay
```

```
ret
pisitenk endp
```

```
pravi proc c uses eax esi edx
mov igra,00h
call clrscr
mov dl,00h
mov dh,00h
call gotoxy
mov edx, OFFSET str13
call writestring
mov edx, OFFSET str14
call writestring
call ReadChar
```

```
ponovi:
call CRTANJE
mov al,X
add al,02h
mov dh,al
mov dl, Y1
call Gotoxy
call ReadChar
cmp ah, 50h
je dole
cmp ah, 4Dh
je desno
cmp ah, 4Bh
je levo
cmp ah, 48h
je gore
cmp al, 1Bh
je napusti
cmp al, 20h
je teren
cmp al, 64h
je brisi
jmp ponovi
```

```
dole:inc X
mov al,X
dec al
cmp al, donja_granica
je smX
jmp ponovi
```

```
smX: dec X
jmp ponovi
```



```
gore: mov al, X
      dec ax
      cmp al, 0ffh
      je poX
      mov X, al
      jmp ponovi
```

```
poX: inc ax
      mov X, al
      jmp ponovi
```

```
levo: dec Y
      dec Y1
      mov al, Y1
      cmp al, 0ffh
      je poY
      jmp ponovi
```

```
poY: inc Y
      inc Y1
      jmp ponovi
```

```
desno: inc Y
        inc Y1
        mov al, Y1
        cmp al, 50h
        je smY
        jmp ponovi
```

```
smY: dec Y
      dec Y1
      jmp ponovi
```

```
teren: mov al, X
        cmp al, 01h
        jle ponovi
        call koordinator
        mov esi, OFFSET niz
        mov eax, esi
        add ax, zed
        mov esi, eax
        mov al, 01h
        mov [esi], al
        call gravitacija_mapa
        jmp ponovi
```

```
brisi: call koordinator
        mov esi, OFFSET niz
        mov eax, esi
        add ax, zed
        mov esi, eax
        mov al, 00h
        mov [esi], al
        call gravitacija_brisanje
        jmp ponovi
```

```
napusti: xor eax, eax
        mov X,al
        mov Y,ax
        mov Y1,al
        mov dh, X
        mov dl, Y1
        call Gotoxy
        call snimanje
```

```
ret
```

```
pravi endp
pozicija1 proc c uses eax esi, Xt:BYTE,Yt:WORD
```

```
        mov ax, Yt1
        mov Yt,ax
        mov al, Xt1
        mov Xt, al
        cmp al,donja_granica
        je kraj
        mov ax, pom
        mul Xt
        add ax , Yt
        mov zed, ax
        mov esi, OFFSET niz
        mov eax, esi
        add ax,zed
        mov esi,eax
        mov al,02h
        mov [esi],al
        inc esi
        mov [esi],al
        inc esi
        mov [esi],al
        dec esi
        mov eax,esi
        sub ax,pom
        mov esi,eax
        mov al,03h
        mov [esi],al
```

```
poc: mov ax, pom
        mul Xt
        add ax , Yt
        mov zed, ax
```

```
        mov esi, OFFSET niz
        mov eax, esi
        add ax, pom
        add ax, zed
        mov esi, eax
        mov al, [esi]
```

```
    cmp al, 0h
    je plus
    jmp kraj
plus: inc Yt
    mov ax,Yt
    sub ax,Yt1
    cmp al,03h
    je pomeraj
```

```
    jmp poc
```

```
pomeraj:
    mov ax,Yt1
    mov Yt,ax
    mov ax, pom
    mul Xt
    add ax , Yt
    mov zed, ax
    mov esi, OFFSET niz
    mov eax, esi
    add ax,zed
    mov esi,eax
    mov al,00h
    mov [esi],al
    inc esi
    mov [esi],al
    inc esi
    mov [esi],al
    dec esi
    mov eax,esi
    sub ax,pom
    mov esi,eax
    mov al,00h
    mov [esi],al
    call brisitenk
    inc Xt
    inc xt1
    mov ax,Yt1
    mov Yt,ax
    mov ax, pom
    mul Xt
    add ax , Yt
    mov zed, ax
    mov esi, OFFSET niz
    mov eax, esi
    add ax,zed
    mov esi,eax
    mov al,02h
    mov [esi],al
    inc esi
    mov [esi],al
    inc esi
    mov [esi],al
    dec esi
    mov eax,esi
    sub ax,pom
```

```

    mov esi,eax
    mov al,03h
    mov [esi],al
    call pisitenk
    mov al,Xt
    cmp al,donja_granica
    je kraj
    jmp poc
kraj:call CRTANJE
ret

```

pozicija1 endp

pozicija2 proc c uses eax esi, Xt:BYTE,Yt:WORD

```

    mov potez,02h
    mov ax, Yt2
    mov Yt,ax
    mov al, Xt2
    mov Xt, al
    cmp al,donja_granica
    je kraj
    mov ax, pom
    mul Xt
    add ax , Yt
    mov zed, ax
    mov esi, OFFSET niz
    mov eax, esi
    add ax,zed
    mov esi,eax
    mov al,02h
    mov [esi],al
    inc esi
    mov [esi],al
    inc esi
    mov [esi],al
    dec esi
    mov eax,esi
    sub ax,pom
    mov esi,eax
    mov al,03h
    mov [esi],al

```

```

poc: mov ax, pom
    mul Xt
    add ax , Yt
    mov zed, ax

```

```

    mov esi, OFFSET niz
    mov eax, esi
    add ax, pom
    add ax, zed
    mov esi, eax
    mov al, [esi]
    cmp al, 0h
    je plus

```

```
    jmp kraj
plus: inc Yt
    mov ax,Yt
    sub ax,Yt2
    cmp al,03h
    je pomeraj
```

```
    jmp poc
```

```
pomeraj:
    mov ax,Yt2
    mov Yt,ax
    mov ax, pom
    mul Xt
    add ax , Yt
    mov zed, ax
    mov esi, OFFSET niz
    mov eax, esi
    add ax,zed
    mov esi,eax
    mov al,00h
    mov [esi],al
    inc esi
    mov [esi],al
    inc esi
    mov [esi],al
    dec esi
    mov eax,esi
    sub ax,pom
    mov esi,eax
    mov al,00h
    mov [esi],al
    call brisitenk
    inc Xt
    inc xt2
    mov ax,Yt2
    mov Yt,ax
    mov ax, pom
    mul Xt
    add ax , Yt
    mov zed, ax
    mov esi, OFFSET niz
    mov eax, esi
    add ax,zed
    mov esi,eax
    mov al,02h
    mov [esi],al
    inc esi
    mov [esi],al
    inc esi
    mov [esi],al
    dec esi
    mov eax,esi
    sub ax,pom
    mov esi,eax
    mov al,03h
```

```

        mov [esi],al
        call pisitenk
        mov al,Xt
        cmp al,donja_granica
        je kraj
        jmp poc
kraj:CALL CRTANJE
ret

```

pozicija2 endp

T1desno proc c uses eax esi, Xt:BYTE,Yt:WORD

```

mov al,xt1
mov Xt,al
mov ax,yt1
mov Yt,ax
inc Yt
inc Yt
    mov ax, pom
    mul Xt
    add ax , Yt
    mov zed, ax

    mov esi, OFFSET niz
    mov eax, esi
    add ax, 01h
    add ax, zed
    mov esi, eax
    mov al, [esi]
    cmp al, 02h
    je kraj
    cmp al, 01h
    je gore
    mov al,Xt
    cmp al, donja_granica
    je desno
    mov eax,esi
    add ax,pom
    mov esi,eax
    dec esi
    dec esi
    mov al, [esi]
    cmp al,02h
    je kraj
    cmp al,01h
    je desno
    inc esi
    mov al, [esi]
    cmp al,02h
    je kraj
    cmp al,01h
    je desno
    inc esi
    mov al, [esi]
    cmp al,02h

```

```

je kraj
cmp al,01h
je desno
mov al,Xt
inc al
cmp al, donja_granica
je dole
mov eax,esi
add ax,pom
mov esi,eax
dec esi
dec esi
mov al, [esi]
cmp al,01h
je dole
inc esi
mov al, [esi]
cmp al,01h
je dole
inc esi
mov al, [esi]
cmp al,01h
je dole
jmp kraj

```

```

gore: mov eax,esi
      sub ax,pom
      mov esi,eax
      mov al,[esi]
      cmp al,01h
      je kraj
      inc ecx
      mov al,cl
      cmp al,03h
      ja kraj
      mov ax, pom
      mul Xt
      add ax , Yt
      mov zed, ax
      mov esi, OFFSET niz
      mov eax, esi
      add ax,zed
      mov esi,eax
      mov al,00h
      mov [esi],al
      dec esi
      mov [esi],al
      dec esi
      mov [esi],al
      inc esi
      mov eax,esi
      sub ax,pom
      mov esi,eax
      mov al,00h
      mov [esi],al
      dec Xt

```

```
inc Yt
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,02h
mov [esi],al
dec esi
mov [esi],al
dec esi
mov [esi],al
inc esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,03h
mov [esi],al
call brisitenk
dec xt1
inc yt1
call pisitenk
jmp kraj
```

```
desno:inc ecx
mov al,cl
cmp al,03h
ja kraj
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,00h
mov [esi],al
dec esi
mov [esi],al
dec esi
mov [esi],al
inc esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,00h
mov [esi],al
inc Yt
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
```



```
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,02h
mov [esi],al
dec esi
mov [esi],al
dec esi
mov [esi],al
inc esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,03h
mov [esi],al
call brisitenk
inc yt1
call pisitenk
jmp kraj
```

```
dole:inc ecx
mov al,cl
cmp al,03h
ja kraj
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,00h
mov [esi],al
dec esi
mov [esi],al
dec esi
mov [esi],al
inc esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,00h
mov [esi],al
inc Xt
inc Yt
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,02h
```

```
mov [esi],al
dec esi
mov [esi],al
dec esi
mov [esi],al
inc esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,03h
mov [esi],al
call brisitenk
inc xt1
inc yt1
call pisitenk
```

kraj: ret

T1desno endp

T1levo proc c uses eax esi, Xt:BYTE, Yt:WORD

```
mov al,xt1
mov Xt,al
mov ax,yt1
mov Yt,ax
    mov ax, pom
    mul Xt
    add ax , Yt
    mov zed, ax

    mov esi, OFFSET niz
    mov eax, esi
    sub ax, 01h
    add ax, zed
    mov esi, eax
    mov al, [esi]
    cmp al, 02h
    je kraj
    cmp al,01h
    je gore
    mov al,Xt
    cmp al, donja_granica
    je levo
    mov eax,esi
    add ax,pom
    mov esi,eax
    inc esi
    inc esi
    mov al, [esi]
    cmp al,02h
    je kraj
    cmp al,01h
    je levo
```

```

dec esi
mov al, [esi]
cmp al,02h
je kraj
cmp al,01h
je levo
dec esi
mov al, [esi]
cmp al,02h
je kraj
cmp al,01h
je levo
mov al,Xt
inc al
cmp al, donja_granica
je dole
mov eax,esi
add ax,pom
mov esi,eax
inc esi
inc esi
mov al, [esi]
cmp al,01h
je dole
dec esi
mov al, [esi]
cmp al,01h
je dole
dec esi
mov al, [esi]
cmp al,01h
je dole
jmp kraj

```

```

gore: mov eax,esi
      sub ax,pom
      mov esi,eax
      mov al,[esi]
      cmp al,01h
      je kraj
      inc ecx
      mov al,cl
      cmp al,03h
      ja kraj
      mov ax, pom
      mul Xt
      add ax , Yt
      mov zed, ax
      mov esi, OFFSET niz
      mov eax, esi
      add ax,zed
      mov esi,eax
      mov al,00h
      mov [esi],al
      inc esi
      mov [esi],al

```

```

inc esi
mov [esi],al
dec esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,00h
mov [esi],al
dec Xt
dec Yt
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,02h
mov [esi],al
inc esi
mov [esi],al
inc esi
mov [esi],al
dec esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,03h
mov [esi],al
call brisitenk
dec xt1
dec yt1
call pisitenk
jmp kraj

```

```

levo: inc ecx
      mov al,cl
      cmp al,03h
      ja kraj
      mov ax, pom
      mul Xt
      add ax , Yt
      mov zed, ax
      mov esi, OFFSET niz
      mov eax, esi
      add ax,zed
      mov esi,eax
      mov al,00h
      mov [esi],al
      inc esi
      mov [esi],al
      inc esi
      mov [esi],al
      dec esi
      mov eax,esi

```

```

sub ax,pom
mov esi,eax
mov al,00h
mov [esi],al
dec Yt
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,02h
mov [esi],al
inc esi
mov [esi],al
inc esi
mov [esi],al
dec esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,03h
mov [esi],al
call brisitenk
dec yt1
call pisitenk
jmp kraj

```

dole:

```

inc ecx
mov al,cl
cmp al,03h
ja kraj
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,00h
mov [esi],al
inc esi
mov [esi],al
inc esi
mov [esi],al
dec esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,00h
mov [esi],al
inc Xt

```

```

dec Yt
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,02h
mov [esi],al
inc esi
mov [esi],al
inc esi
mov [esi],al
dec esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,03h
mov [esi],al
call brisitenk
inc xt1
dec yt1
call pisitenk

```

kraj: ret

T1levo endp

T2desno proc c uses eax esi, Xt:BYTE,Yt:WORD

```

mov al,xt2
mov Xt,al
mov ax,yt2
mov Yt,ax
inc Yt
inc Yt
    mov ax, pom
    mul Xt
    add ax , Yt
    mov zed, ax

    mov esi, OFFSET niz
    mov eax, esi
    add ax, 01h
    add ax, zed
    mov esi, eax
    mov al, [esi]
    cmp al, 02h
    je kraj
    cmp al, 01h
    je gore

```

```
mov al,Xt
cmp al, donja_granica
je desno
mov eax,esi
add ax,pom
mov esi,eax
dec esi
dec esi
mov al, [esi]
cmp al,02h
je kraj
cmp al,01h
je desno
inc esi
mov al, [esi]
cmp al,02h
je kraj
cmp al,01h
je desno
inc esi
mov al, [esi]
cmp al,02h
je kraj
cmp al,01h
je desno
mov al,Xt
inc al
cmp al, donja_granica
je dole
mov eax,esi
add ax,pom
mov esi,eax
dec esi
dec esi
mov al, [esi]
cmp al,01h
je dole
inc esi
mov al, [esi]
cmp al,01h
je dole
inc esi
mov al, [esi]
cmp al,01h
je dole
jmp kraj
```

```
gore: mov eax,esi
      sub ax,pom
      mov esi,eax
      mov al,[esi]
      cmp al,01h
      je kraj
      inc ecx
      mov al,cl
      cmp al,03h
```

```

ja kraj
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,00h
mov [esi],al
dec esi
mov [esi],al
dec esi
mov [esi],al
inc esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,00h
mov [esi],al
dec Xt
inc Yt
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,02h
mov [esi],al
dec esi
mov [esi],al
dec esi
mov [esi],al
inc esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,03h
mov [esi],al
call brisitenk
dec xt2
inc yt2
call pisitenk
jmp kraj

```

```

desno:inc ecx
mov al,cl
cmp al,03h
ja kraj
mov ax, pom
mul Xt
add ax , Yt

```



```

mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,00h
mov [esi],al
dec esi
mov [esi],al
dec esi
mov [esi],al
inc esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,00h
mov [esi],al
inc Yt
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,02h
mov [esi],al
dec esi
mov [esi],al
dec esi
mov [esi],al
inc esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,03h
mov [esi],al
call brisitenk
inc yt2
call pisitenk
jmp kraj

```

```

dole:inc ecx
mov al,cl
cmp al,03h
ja kraj
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,00h

```

```

mov [esi],al
dec esi
mov [esi],al
dec esi
mov [esi],al
inc esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,00h
mov [esi],al
inc Xt
inc Yt
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,02h
mov [esi],al
dec esi
mov [esi],al
dec esi
mov [esi],al
inc esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,03h
mov [esi],al
call brisitenk
inc xt2
inc yt2
call pisitenk

```

kraj: ret

T2desno endp

T2levo proc c uses eax esi, Xt:BYTE,Yt:WORD

```

mov al,xt2
mov Xt,al
mov ax,yt2
mov Yt,ax
    mov ax, pom
    mul Xt
    add ax , Yt
    mov zed, ax

```

```
mov esi, OFFSET niz
mov eax, esi
sub ax, 01h
add ax, zed
mov esi, eax
mov al, [esi]
cmp al, 02h
je kraj
cmp al, 01h
je gore
mov al, Xt
cmp al, donja_granica
je levo
mov eax, esi
add ax, pom
mov esi, eax
inc esi
inc esi
mov al, [esi]
cmp al, 02h
je kraj
cmp al, 01h
je levo
dec esi
mov al, [esi]
cmp al, 02h
je kraj
cmp al, 01h
je levo
dec esi
mov al, [esi]
cmp al, 02h
je kraj
cmp al, 01h
je levo
mov al, Xt
inc al
cmp al, donja_granica
je dole
mov eax, esi
add ax, pom
mov esi, eax
inc esi
inc esi
mov al, [esi]
cmp al, 01h
je dole
dec esi
mov al, [esi]
cmp al, 01h
je dole
dec esi
mov al, [esi]
cmp al, 01h
je dole
jmp kraj
```

```
gore: mov eax,esi
      sub ax,pom
      mov esi,eax
      mov al,[esi]
      cmp al,01h
      je kraj
      inc ecx
      mov al,cl
      cmp al,03h
      ja kraj
      mov ax, pom
      mul Xt
      add ax , Yt
      mov zed, ax
      mov esi, OFFSET niz
      mov eax, esi
      add ax,zed
      mov esi,eax
      mov al,00h
      mov [esi],al
      inc esi
      mov [esi],al
      inc esi
      mov [esi],al
      dec esi
      mov eax,esi
      sub ax,pom
      mov esi,eax
      mov al,00h
      mov [esi],al
      dec Xt
      dec Yt
      mov ax, pom
      mul Xt
      add ax , Yt
      mov zed, ax
      mov esi, OFFSET niz
      mov eax, esi
      add ax,zed
      mov esi,eax
      mov al,02h
      mov [esi],al
      inc esi
      mov [esi],al
      inc esi
      mov [esi],al
      dec esi
      mov eax,esi
      sub ax,pom
      mov esi,eax
      mov al,03h
      mov [esi],al
      call brisitenk
      dec xt2
      dec yt2
```

```
call pisitenk
jmp kraj
```

```
levo: inc ecx
      mov al,cl
      cmp al,03h
      ja kraj
      mov ax, pom
      mul Xt
      add ax , Yt
      mov zed, ax
      mov esi, OFFSET niz
      mov eax, esi
      add ax,zed
      mov esi,eax
      mov al,00h
      mov [esi],al
      inc esi
      mov [esi],al
      inc esi
      mov [esi],al
      dec esi
      mov eax,esi
      sub ax,pom
      mov esi,eax
      mov al,00h
      mov [esi],al
      dec Yt
      mov ax, pom
      mul Xt
      add ax , Yt
      mov zed, ax
      mov esi, OFFSET niz
      mov eax, esi
      add ax,zed
      mov esi,eax
      mov al,02h
      mov [esi],al
      inc esi
      mov [esi],al
      inc esi
      mov [esi],al
      dec esi
      mov eax,esi
      sub ax,pom
      mov esi,eax
      mov al,03h
      mov [esi],al
      call brisitenk
      dec yt2
      call pisitenk
      jmp kraj
```

```
dole:inc ecx
      mov al,cl
      cmp al,03h
```

```

ja kraj
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,00h
mov [esi],al
inc esi
mov [esi],al
inc esi
mov [esi],al
dec esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,00h
mov [esi],al
inc Xt
dec Yt
mov ax, pom
mul Xt
add ax , Yt
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax,zed
mov esi,eax
mov al,02h
mov [esi],al
inc esi
mov [esi],al
inc esi
mov [esi],al
dec esi
mov eax,esi
sub ax,pom
mov esi,eax
mov al,03h
mov [esi],al
call brisitenk
inc xt2
dec yt2
call pisitenk

```

kraj: ret

T2levo endp

pucanj1 proc c uses eax esi edx Xp:BYTE,Yp:WORD,Xb:BYTE,Yb:BYTE, ceo_deo:BYTE,
delta_x:Byte , delta_y:Byte

```

mov delta_x,00h

```

```
mov delta_y,00h
cmp ugao_znak1,0h
jne skok
mov x_bilo_0,0h
jmp skok1
skok:mov x_bilo_0,01h
skok1:mov al,xt1
sub al,01h
mov Xp,al
mov ax, yt1
add ax,03h
mov Yp,ax
mov al,Xp
mov xgr,al
mov ax,Yp
mov ygr,ax
mov dh,xgr
mov ax,ygr
mov dl,al
add dh,02h
call Gotoxy
MOV al,0F8h
call WriteChar
```

```
mov ax,ugao1
div step
mov Xb,al
mov al,09h
sub al,Xb
mov Yb,al
```

```
mov ax, pom
mul Xp
add ax , Yp
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax, zed
mov esi, eax
mov al, [esi]
cmp al, 02h
je kraj
cmp al, 01h
je kraj
cmp snaga1,01
jne poc
add Yb,01h
add Xb,01h
poc: xor eax, eax
mov al, Xb
cmp al, Yb
ja xvecey
```

```

xor eax,eax ;yvecex
xor ecx,ecx
inc ecx
mov al, Yb
cmp Xb,00h
je x_je_0
div Xb
MOV ceo_deo,al
mov al,ah
cmp al, 01h
ja ceoinc2
dalje2: inc Yp
        inc delta_y
        mov al, Yb
        cmp delta_y,al
        je provera_x
        cmp cl, ceo_deo
        jae pomeranjemanjeg2
provera2:inc cl
        cmp xgr,00h
        jle preskok
        mov dh,xgr
        mov ax,ygr
        add dh,02h
        mov dl,al
        mov eax,LightBlue + (white * 16)
        call SetTextColor
        call Gotoxy
        MOV al,0b0h
        call WriteChar
preskok:mov al,Xp
        mov xgr,al
        mov ax,Yp
        mov ygr,ax
        call udar
        jmp dalje2

```

```

xvecey: xor eax,eax
        xor ecx,ecx
        inc ecx
        mov al, Xb
        div Yb
        MOV ceo_deo,al
        mov al,ah
        cmp al, 01h
        ja ceoinc1
dalje1:cmp x_bilo_0,01h
        je rast
        dec Xp
        jmp nastavi
rast: inc Xp

```



```

nastavi:inc delta_x
        mov al, Xb
        cmp delta_x, al
        je provera_y
        cmp cl, ceo_deo
        jae pomeranjemanjeg1
provera1:inc ecx
        cmp xgr,00h
        jle preskok1
        mov dh,xgr
        mov ax,ygr
        mov dl,al
        add dh,02h
        mov eax,LightBlue + (white * 16)
        call SetTextColor
        call Gotoxy
        MOV al,0b0h
        call WriteChar
preskok1:mov al,Xp
        mov xgr,al
        mov ax,Yp
        mov ygr,ax
        call udar
        jmp dalje1

provera_y: mov al, Yb
        cmp delta_y,al
        je smanji_brzinu
vrt:inc Yp
        INC delta_y
        cmp xgr,00h
        jle preskok2
        mov dh,xgr
        mov ax,ygr
        mov dl,al
        add dh,02h
        mov eax,LightBlue + (white * 16)
        call SetTextColor
        call Gotoxy
        MOV al,0b0h
        call WriteChar
preskok2:mov al,Xp
        mov xgr,al
        mov ax,Yp
        mov ygr,ax
        call udar
        mov al, Yb
        cmp delta_y,al
        je smanji_brzinu
        jmp vrt

provera_x: mov al , Xb
        cmp delta_x,al
        je smanji_brzinu
vrt1:cmp x_bilo_0,01h
        je rast1

```

```

        dec Xp
        jmp nastavi1
rast1: inc Xp
nastavi1: INC delta_x
        cmp xgr,00h
        jle preskok3
        mov dh,xgr
        mov ax,ygr
        mov dl,al
        add dh,02h
        mov eax,LightBlue + (white * 16)
        call SetTextColor
        call Gotoxy
        MOV al,0b0h
        call WriteChar
preskok3:mov al,Xp
        mov xgr,al
        mov ax,Yp
        mov ygr,ax
        call udar
        mov al,Xb
        cmp delta_X,al
        je smanji_brzinu
        jmp vrt1

x_je_0:mov x_bilo_0,01h
        inc Yp
        inc delta_y
        mov al, Yb
        cmp delta_y,al
        jne skok_dalje
        dec Yp
skok_dalje: cmp delta_y,al
        je smanji_brzinu
        cmp xgr,00h
        jle preskok4
        mov dh,xgr
        mov ax,ygr
        mov dl,al
        add dh,02h
        mov eax,LightBlue + (white * 16)
        call SetTextColor
        call Gotoxy
        MOV al,0b0h
        call WriteChar
preskok4:mov al,Xp
        mov xgr,al
        mov ax,Yp
        mov ygr,ax
        call udar
        jmp x_je_0

smanji_brzinu: cmp x_bilo_0,01h
        je rast3
        dec Xb
        jmp nastavi3

```

```
    rast3:inc Xb
nastavi3:mov delta_x,00h
        mov delta_y,00h
        jmp poc
```

```
pomeranjanjeg1: inc Yp
                inc delta_y
                xor ecx,ecx
                jmp provera1
pomeranjanjeg2:cmp x_bilo_0,01h
                je rast2
                dec Xp
                jmp nastavi2
rast2: inc Xp
nastavi2:inc delta_x
        xor ecx,ecx
        jmp provera2
```

```
ceoinc1: inc ceo_deo
jmp dalje1
ceoinc2: inc ceo_deo
jmp dalje2
```

```
kraj:call udar
ret
```

pucanj1 endp

pucanj2 proc c uses eax esi edx Xp:BYTE,Yp:WORD,Xb:BYTE,Yb:BYTE, ceo_deo:BYTE,
delta_x:Byte , delta_y:Byte

```
    mov delta_x,00h
    mov delta_y,00h
    cmp ugao_znak2,0h
    jne skok
    mov x_bilo_0,0h
    jmp skok1
skok:mov x_bilo_0,01h
skok1:mov al,xt2
    sub al,01h
    mov Xp,al
    mov ax, yt2
    sub al,01h
    mov Yp,ax
    mov al,Xp
    mov xgr,al
    mov ax,Yp
    mov ygr,ax
```

```
mov dh,xgr
mov ax,ygr
mov dl,al
add dh,02h
call Gotoxy
MOV al,0F8h
call WriteChar
```

```
mov ax,ugao2
div step
mov Xb,al
mov al,09h
sub al,Xb
mov Yb,al
```

```
mov ax, pom
mul Xp
add ax , Yp
mov zed, ax
mov esi, OFFSET niz
mov eax, esi
add ax, zed
mov esi, eax
mov al, [esi]
cmp al, 02h
je kraj
cmp al, 01h
je kraj
cmp snaga2,01
jne poc
add Yb,01h
add Xb,01h
poc: xor eax, eax
mov al, Xb
cmp al, Yb
ja xvecey
```

```
xor eax,eax ;yvecex
xor ecx,ecx
inc ecx
mov al, Yb
cmp Xb,00h
je x_je_0
div Xb
MOV ceo_deo,al
mov al,ah
cmp al, 01h
ja ceoinc2
dalje2: dec Yp
inc delta_y
mov al, Yb
cmp delta_y,al
```

```

        je provera_x
        cmp cl, ceo_deo
        jae pomeranjemanjeg2
provera2:inc cl
        cmp xgr,00h
        jle preskok
        mov dh,xgr
        mov ax,ygr
        add dh,02h
        mov dl,al
        mov eax,LightBlue + (white * 16)
        call SetTextColor
        call Gotoxy
        MOV al,0b0h
        call WriteChar
preskok:mov al,Xp
        mov xgr,al
        mov ax,Yp
        mov ygr,ax
        call udar
        jmp dalje2

xvecey: xor eax,eax
        xor ecx,ecx
        inc ecx
        mov al, Xb
        div Yb
        MOV ceo_deo,al
        mov al,ah
        cmp al, 01h
        ja ceoinc1
dalje1:cmp x_bilo_0,01h
        je rast
        dec Xp
        jmp nastavi
        rast: inc Xp
nastavi:inc delta_x
        mov al, Xb
        cmp delta_x, al
        je provera_y
        cmp cl, ceo_deo
        jae pomeranjemanjeg1
provera1:inc ecx
        cmp xgr,00h
        jle preskok1
        mov dh,xgr
        mov ax,ygr
        mov dl,al
        add dh,02h
        mov eax,LightBlue + (white * 16)
        call SetTextColor
        call Gotoxy
        MOV al,0b0h

```

```
    call WriteChar
preskok1: mov al, Xp
    mov xgr, al
    mov ax, Yp
    mov ygr, ax
    call udar
    jmp dalje1
```

```
provera_y: mov al, Yb
    cmp delta_y, al
    je smanji_brzinu
vrt: dec Yp
    INC delta_y
    cmp xgr, 00h
    jle preskok2
    mov dh, xgr
    mov ax, ygr
    mov dl, al
    add dh, 02h
    mov eax, LightBlue + (white * 16)
    call SetTextColor
    call Gotoxy
    MOV al, 0b0h
    call WriteChar
preskok2: mov al, Xp
    mov xgr, al
    mov ax, Yp
    mov ygr, ax
    call udar
    mov al, Yb
    cmp delta_y, al
    je smanji_brzinu
    jmp vrt
```

```
provera_x: mov al, Xb
    cmp delta_x, al
    je smanji_brzinu
vrt1: cmp x_bilo_0, 01h
    je rast1
    dec Xp
    jmp nastavi1
rast1: inc Xp
nastavi1: INC delta_x
    cmp xgr, 00h
    jle preskok3
    mov dh, xgr
    mov ax, ygr
    mov dl, al
    add dh, 02h
    mov eax, LightBlue + (white * 16)
    call SetTextColor
    call Gotoxy
    MOV al, 0b0h
    call WriteChar
preskok3: mov al, Xp
    mov xgr, al
```

```

        mov ax,Yp
        mov ygr,ax
        call udar
        mov al,Xb
        cmp delta_X,al
        je smanji_brzinu
        jmp vrt1

x_je_0:mov x_bilo_0,01h
        dec Yp
        inc delta_y
        mov al, Yb
        cmp delta_y,al
        jne skok_dalje
        inc Yp
skok_dalje:cmp delta_y,al
        je smanji_brzinu
        cmp xgr,00h
        jle preskok4
        mov dh,xgr
        mov ax,ygr
        mov dl,al
        add dh,02h
        mov eax,LightBlue + (white * 16)
        call SetTextColor
        call Gotoxy
        MOV al,0b0h
        call WriteChar
preskok4:mov al,Xp
        mov xgr,al
        mov ax,Yp
        mov ygr,ax
        call udar
        jmp x_je_0

smanji_brzinu: cmp x_bilo_0,01h
        je rast3
        dec Xb
        jmp nastavi3
rast3:inc Xb
nastavi3:mov delta_x,00h
        mov delta_y,00h
        jmp poc

pomerañjemanjeg1: dec Yp
        inc delta_y
        xor ecx,ecx
        jmp provera1
pomerañjemanjeg2:cmp x_bilo_0,01h
        je rast2
        dec Xp
        jmp nastavi2
rast2: inc Xp
nastavi2:inc delta_x
        xor ecx,ecx
        jmp provera2

```

```
ceoinc1: inc ceo_deo
jmp dalje1
ceoinc2: inc ceo_deo
jmp dalje2
```

```
kraj:call udar
ret
```

```
pucanj2 endp
```

```
udar proc c uses eax esi edx
    cmp xgr,00h
    jle kraj
    mov al,xgr
    dec al
    mov al,xgr
    mov dl, donja_granica
    inc dl
    cmp al,dl
    jae sledeci
    cmp ygr,50h
    jae sledeci
    mov ax, pom
    mul xgr
    add ax ,ygr
    mov zed, ax
    mov esi, OFFSET niz
    mov eax, esi
    add ax, zed
    mov esi, eax
    mov al, [esi]
    cmp al, 00h
    je nije_udar
    cmp al,02h
    je smanji_zivot
    cmp al,03h
    je smanji_zivot
    cmp al,01h
    je udar_brda

    jmp kraj
```

```
nije_udar:cmp xgr,00h
    jle preskok
    mov dh,xgr
    mov ax,ygr
    mov dl,al
    add dh,02h
```



```

    call Gotoxy
    mov eax,black + (white * 16)
    call SetTextColor
    MOV al,0F8h
    call WriteChar
preskok: jmp kraj

```

```

udar_brda: mov ax,pom
            mul xgr
            add ax ,ygr
            mov zed, ax
            mov esi, OFFSET niz
            mov eax, esi
            add ax, zed
            mov esi, eax
            mov al,00h
            mov[esi],al
            call gravitacija_brisanje
            cmp potez,01h
            jne dalje
            call pozicija2
            mov potez,01h
            jmp dalje2
dalje: call pozicija1

```

```

dalje2: cmp potez, 01h
        je igra_drugi
        call igrac1

```

```

smanji_zivot: cmp potez, 01h
               je z2
               dec zivot1
               cmp zivot1,0h
               jne produzi
               call krajigre
               produzi: call igrac1
z2: dec zivot2
    cmp zivot2,0h
    jne produzi1
    call krajigre
produzi1: call igrac2
sledeci: cmp potez, 01h
         je igra_drugi
         call igrac1
igra_drugi: call igrac2

```

```

kraj: xor eax,eax
mov al, kasnjenje
call Delay
ret
udar endp
igrac1 proc c uses eax ecx ebx

```

```

xor ecx,ecx
mov dl,28h
mov dh,01h
call gotoxy
mov eax,lightblue + (white * 16)
    call SetTextColor
mov al,31h
call writechar
mov potez, 01h
poc: mov dl,00h
mov dh,00h
call gotoxy
mov eax,lightblue + (white * 16)
    call SetTextColor
mov edx,OFFSET str1

call WriteString

mov al, ugao_znak1
cmp al, 0h
je pisi_poz

mov ax, 02Dh
jmp pisi_kon
pisi_poz: mov al , 00h

pisi_kon: call WriteChar
mov ax,ugao1
mov dl, 0Ah
div dl
add al, 030h
call WriteChar
mov al, 030h
call WriteChar ; završava se pisanje "UGAO TENKA"

mov dl,00h
mov dh,01h
call Gotoxy
mov edx,OFFSET str3
call WriteString

xor ebx,ebx
mov dl,00h
mov dh,01h
call Gotoxy
mov eax,red + (white * 16)
    call SetTextColor
mov al,03h
mov bl,zivot1
opet:cmp bl,00
je drugacije
call WriteChar
inc dl
call Gotoxy
dec bl
jmp opet

```

```
drugacije:call ReadChar
cmp ah, 4Dh
je desno
cmp ah, 4Bh
je levo
cmp al, 20h
je puc
cmp ah, 50h
je dole
cmp ah, 48h
je gore
cmp al, 2bh
je snagagore
cmp al, 2dh
je snagadole
jmp poc
levo:mov ax,yt1
cmp al,0h
je poc
call T1levo
jmp poc
desno:mov ax,yt1
cmp al,79h
je poc
call T1desno
jmp poc
puc: call pucanj1
call igrac2
jmp poc
gore: cmp ugao1,00h
      je obrni
      cmp ugao_znak1,0
      je obrni
      mov ax,ugao1
      sub al,0Ah
      mov ugao1,ax
      cmp ugao1,00h
      jne poc
      mov ugao_znak1,00h
      jmp poc
obrni:mov ax,ugao1
      cmp al,50h
      je poc
      add al,0Ah
      mov ugao1,ax
      mov ugao_znak1,00h
      jmp poc
dole: cmp ugao1,00
      jne obrni1
obrni2:mov ax,ugao1
      cmp al,50h
      je poc
      add al,0Ah
      mov ugao1,ax
      mov ugao_znak1,01h
      jmp poc
```

```

obrni1:cmp ugao_znak1,0
      jne obrni2
      mov ax,ugao1
      cmp al,0h
      je poc
      sub al,0Ah
      mov ugao1,ax
      jmp poc

snagagore:mov snaga1,01h
      mov dl,0bh
      mov dh,01h
      call gotoxy
      mov eax,lightblue + (white * 16)
      call SetTextColor
      mov al,0afh
      call writechar
      jmp poc
snagadole:mov snaga1,00h
      mov dl,0bh
      mov dh,01h
      call gotoxy
      mov eax,lightblue + (white * 16)
      call SetTextColor
      mov al,00h
      call writechar
      jmp poc

      ret
igrac1 endp
igrac2 proc c uses eax ecx ebx

      xor ecx,ecx
      mov dl,28h
      mov dh,01h
      call gotoxy
      mov eax,lightblue + (white * 16)
      call SetTextColor
      mov al,32h
      call writechar
      mov potez, 02h
poc: mov dl,42h
      mov dh,00h
      call gotoxy
      mov eax,lightblue + (white * 16)
      call SetTextColor
      mov edx,OFFSET str2

      call WriteString

      mov al, ugao_znak2
      cmp al, 0h
      je pisi_poz

      mov ax, 02Dh
      jmp pisi_kon
pisi_poz: mov al , 00h

```

```

pisi_kon: call WriteChar
        mov ax,ugao2
        mov dl, 0Ah
        div dl
        add al, 030h
        call WriteChar
        mov al, 030h
        call WriteChar
        mov dl,46h
        mov dh,01h
        call Gotoxy
        mov edx,OFFSET str3
        call WriteString
        xor ebx,ebx
        mov dl,4fh
        mov dh,01h
        call Gotoxy
        mov eax,red + (white * 16)
            call SetTextColor
        mov al,03h
        mov bl,zivot2
opet:cmp bl,00
        je produzi
        call WriteChar
        dec dl
        call Gotoxy
        dec bl
        jmp opet

```

```

produzi:call ReadChar
        cmp ah, 4Dh
        je desno
        cmp ah, 4Bh
        je levo
        cmp al, 20h
        je puc
        cmp ah, 50h
        je dole
        cmp ah, 48h
        je gore
        cmp al, 2bh
        je snagagore
        cmp al, 2dh
        je snagadole
        jmp poc
levo:mov ax,yt2
        cmp al,0h
        je poc
        call T2levo
        jmp poc
desno:mov ax,yt2
        cmp al,4dh
        je poc
        call T2desno
        jmp poc

```

```
puc:call pucanj2  
call igrac1
```

```
jmp poc
```

```
gore: cmp ugao2,00h  
      je obrni  
      cmp ugao_znak2,0  
      je obrni  
      mov ax,ugao2  
      sub al,0Ah  
      mov ugao2,ax  
      cmp ugao2,00h  
      jne poc  
      mov ugao_znak2,00h  
      jmp poc
```

```
obrni:mov ax,ugao2  
      cmp al,50h  
      je poc  
      add al,0Ah  
      mov ugao2,ax  
      mov ugao_znak2,00h  
      jmp poc
```

```
dole: cmp ugao2,00  
      jne obrni1
```

```
obrni2:mov ax,ugao2  
      cmp al,50h  
      je poc  
      add al,0Ah  
      mov ugao2,ax  
      mov ugao_znak2,01h  
      jmp poc
```

```
obrni1:cmp ugao_znak2,0  
      jne obrni2  
      mov ax,ugao2  
      cmp al,0h  
      je poc  
      sub al,0Ah  
      mov ugao2,ax  
      jmp poc
```

```
snagagore:mov snaga2,01h  
          mov dl,44h  
          mov dh,01h  
          call gotoxy  
          mov eax,lightblue + (white * 16)  
          call SetTextColor  
          mov al,0aeh  
          call writechar  
          jmp poc
```

```
snagadole:mov snaga2,00h  
          mov dl,44h  
          mov dh,01h  
          call gotoxy  
          mov eax,lightblue + (white * 16)  
          call SetTextColor  
          mov al,00h
```

```
    call writechar  
    jmp poc
```

```
    ret  
igrac2 endp  
krajigre proc c uses eax  
    call clrscr  
    mov dl,00h  
    mov dh,00h  
    call gotoxy  
    cmp potez,01h  
    jne preskok  
    mov edx,OFFSET str4  
    call writestring  
    jmp dalje  
preskok:mov edx,OFFSET str5  
    call writestring  
dalje:xor eax,eax  
    mov ax,0fffh  
    call delay  
    call main  
    ret  
krajigre endp  
main proc  
    mov eax,LightBlue + (white * 16)  
    call SetTextColor  
    mov xt1, 1h  
    mov yt1, 0ah  
    mov xt2, 1h  
    mov yt2, 46h  
    mov xgr, 0h  
    mov ygr, 0h  
    mov x_bilo_0, 0h  
    mov ugao1, 0h  
    mov ugao2, 0h  
    mov zivot1, 0Ah  
    mov zivot2, 0Ah  
    mov snaga1, 0h  
    mov snaga2, 0h  
    mov potez,01h  
    call ucitavanje
```

```
biranje: call Clrscr  
    mov dl,00h  
    mov dh,00h  
    call gotoxy  
    mov edx, OFFSET str7  
    call writestring  
    mov edx, OFFSET str8  
    call writestring
```

```

call ReadChar
cmp al, 70h
je pra
cmp al, 69h
je igr
cmp al, 1Bh
je kraj
jmp biranje
pra: call pravi
xor eax, eax
jmp biranje

```

```

igr:mov igra,01h
call clrscr
mov dl,00h
mov dh,00h
call gotoxy
mov edx, OFFSET str9
call writestring
mov edx, OFFSET str10
call writestring
mov edx, OFFSET str11
call writestring
mov edx, OFFSET str12
call writestring
call readchar
call učitavanje
call CRTANJE
call pozicija1
call pozicija2

```

```

mov potez, 01h
mov dl,42h
mov dh,00h
mov eax,lightblue + (white * 16)
call SetTextColor
call gotoxy
mov edx,OFFSET str2
call WriteString
mov al,20h
call Writechar
mov al,30h
call Writechar
call Writechar
mov dl,46h
mov dh,01h
call gotoxy
mov eax,red + (white * 16)
call SetTextColor
xor ecx,ecx
mov cl,0ah
mov al,03h
opet:cmp cl,00
je preskok

```



```

    call writeChar
    dec cl
    jmp opet
preskok:mov dl,00h
    mov dh,00h
    call gotoxy
    mov eax,lightblue + (white * 16)
        call SetTextColor
    mov edx,OFFSET str1
    call WriteString
    mov dl,00h
    mov dh,01h
    call gotoxy
    xor ecx,ecx
    mov eax,red + (white * 16)
        call SetTextColor
    mov cl,0ah
    mov al,03h
opet1:cmp cl,00
    je preskok1
    call writeChar
    dec cl
    jmp opet1
preskok1:mov eax,lightblue + (white * 16)
    call SetTextColor

    call igrac1

kraj:invoke ExitProcess,0

main endp
end main

```