|  |
| --- |
| data segment |
| - | new\_line db 13, 10, "$" |
|  |  |
|  | game\_draw db "\_|\_|\_", 13, 10 |
|  | db "\_|\_|\_", 13, 10 |
|  | db "\_|\_|\_", 13, 10, "$" |
|  |  |
|  | game\_pointer db 9 DUP(?) |
|  |  |
|  | win\_flag db 0 |
|  | player db "0$" |
|  |  |
|  | game\_over\_message db "fasiulhaq", 13, 10, "$" |
|  | game\_start\_message db "samehulhaq", 13, 10, "$" |
|  | player\_message db "PLAYER $" |
|  | win\_message db " WIN!$" |
|  | type\_message db "TYPE A POSITION: $" |
|  | ends |
|  |  |
|  | stack segment |
|  | dw 128 dup(?) |
|  | ends |
|  |  |
|  | extra segment |
|  |  |
|  | ends |
|  |  |
|  | code segment |
|  | start: |
|  | ; set segment registers |
|  | mov ax, data |
|  | mov ds, ax |
|  | mov ax, extra |
|  | mov es, ax |
|  |  |
|  | ; game start |
|  | call set\_game\_pointer |
|  |  |
|  | main\_loop: |
|  | call clear\_screen |
|  |  |
|  | lea dx, game\_start\_message |
|  | call print |
|  |  |
|  | lea dx, new\_line |
|  | call print |
|  |  |
|  | lea dx, player\_message |
|  | call print |
|  | lea dx, player |
|  | call print |
|  |  |
|  | lea dx, new\_line |
|  | call print |
|  |  |
|  | lea dx, game\_draw |
|  | call print |
|  |  |
|  | lea dx, new\_line |
|  | call print |
|  |  |
|  | lea dx, type\_message |
|  | call print |
|  |  |
|  | ; read draw position |
|  | call read\_keyboard |
|  |  |
|  | ; calculate draw position |
|  | sub al, 49 |
|  | mov bh, 0 |
|  | mov bl, al |
|  |  |
|  | call update\_draw |
|  |  |
|  | call check |
|  |  |
|  | ; check if game ends |
|  | cmp win\_flag, 1 |
|  | je game\_over |
|  |  |
|  | call change\_player |
|  |  |
|  | jmp main\_loop |
|  |  |
|  |  |
|  | change\_player: |
|  | lea si, player |
|  | xor ds:[si], 1 |
|  |  |
|  | ret |
|  |  |
|  |  |
|  | update\_draw: |
|  | mov bl, game\_pointer[bx] |
|  | mov bh, 0 |
|  |  |
|  | lea si, player |
|  |  |
|  | cmp ds:[si], "0" |
|  | je draw\_x |
|  |  |
|  | cmp ds:[si], "1" |
|  | je draw\_o |
|  |  |
|  | draw\_x: |
|  | mov cl, "x" |
|  | jmp update |
|  |  |
|  | draw\_o: |
|  | mov cl, "o" |
|  | jmp update |
|  |  |
|  | update: |
|  | mov ds:[bx], cl |
|  |  |
|  | ret |
|  |  |
|  |  |
|  | check: |
|  | call check\_line |
|  | ret |
|  |  |
|  |  |
|  | check\_line: |
|  | mov cx, 0 |
|  |  |
|  | check\_line\_loop: |
|  | cmp cx, 0 |
|  | je first\_line |
|  |  |
|  | cmp cx, 1 |
|  | je second\_line |
|  |  |
|  | cmp cx, 2 |
|  | je third\_line |
|  |  |
|  | call check\_column |
|  | ret |
|  |  |
|  | first\_line: |
|  | mov si, 0 |
|  | jmp do\_check\_line |
|  |  |
|  | second\_line: |
|  | mov si, 3 |
|  | jmp do\_check\_line |
|  |  |
|  | third\_line: |
|  | mov si, 6 |
|  | jmp do\_check\_line |
|  |  |
|  | do\_check\_line: |
|  | inc cx |
|  |  |
|  | mov bh, 0 |
|  | mov bl, game\_pointer[si] |
|  | mov al, ds:[bx] |
|  | cmp al, "\_" |
|  | je check\_line\_loop |
|  |  |
|  | inc si |
|  | mov bl, game\_pointer[si] |
|  | cmp al, ds:[bx] |
|  | jne check\_line\_loop |
|  |  |
|  | inc si |
|  | mov bl, game\_pointer[si] |
|  | cmp al, ds:[bx] |
|  | jne check\_line\_loop |
|  |  |
|  |  |
|  | mov win\_flag, 1 |
|  | ret |
|  |  |
|  |  |
|  |  |
|  | check\_column: |
|  | mov cx, 0 |
|  |  |
|  | check\_column\_loop: |
|  | cmp cx, 0 |
|  | je first\_column |
|  |  |
|  | cmp cx, 1 |
|  | je second\_column |
|  |  |
|  | cmp cx, 2 |
|  | je third\_column |
|  |  |
|  | call check\_diagonal |
|  | ret |
|  |  |
|  | first\_column: |
|  | mov si, 0 |
|  | jmp do\_check\_column |
|  |  |
|  | second\_column: |
|  | mov si, 1 |
|  | jmp do\_check\_column |
|  |  |
|  | third\_column: |
|  | mov si, 2 |
|  | jmp do\_check\_column |
|  |  |
|  | do\_check\_column: |
|  | inc cx |
|  |  |
|  | mov bh, 0 |
|  | mov bl, game\_pointer[si] |
|  | mov al, ds:[bx] |
|  | cmp al, "\_" |
|  | je check\_column\_loop |
|  |  |
|  | add si, 3 |
|  | mov bl, game\_pointer[si] |
|  | cmp al, ds:[bx] |
|  | jne check\_column\_loop |
|  |  |
|  | add si, 3 |
|  | mov bl, game\_pointer[si] |
|  | cmp al, ds:[bx] |
|  | jne check\_column\_loop |
|  |  |
|  |  |
|  | mov win\_flag, 1 |
|  | ret |
|  |  |
|  |  |
|  | check\_diagonal: |
|  | mov cx, 0 |
|  |  |
|  | check\_diagonal\_loop: |
|  | cmp cx, 0 |
|  | je first\_diagonal |
|  |  |
|  | cmp cx, 1 |
|  | je second\_diagonal |
|  |  |
|  | ret |
|  |  |
|  | first\_diagonal: |
|  | mov si, 0 |
|  | mov dx, 4 ;fasiulhaq |
|  | jmp do\_check\_diagonal |
|  |  |
|  | second\_diagonal: |
|  | mov si, 2 |
|  | mov dx, 2 |
|  | jmp do\_check\_diagonal |
|  |  |
|  | do\_check\_diagonal: |
|  | inc cx |
|  |  |
|  | mov bh, 0 |
|  | mov bl, game\_pointer[si] |
|  | mov al, ds:[bx] |
|  | cmp al, "\_" |
|  | je check\_diagonal\_loop |
|  |  |
|  | add si, dx |
|  | mov bl, game\_pointer[si] |
|  | cmp al, ds:[bx] |
|  | jne check\_diagonal\_loop |
|  |  |
|  | add si, dx |
|  | mov bl, game\_pointer[si] |
|  | cmp al, ds:[bx] |
|  | jne check\_diagonal\_loop |
|  |  |
|  |  |
|  | mov win\_flag, 1 |
|  | ret |
|  |  |
|  |  |
|  | game\_over: |
|  | call clear\_screen |
|  |  |
|  | lea dx, game\_start\_message |
|  | call print |
|  |  |
|  | lea dx, new\_line |
|  | call print |
|  |  |
|  | lea dx, game\_draw |
|  | call print |
|  |  |
|  | lea dx, new\_line |
|  | call print |
|  |  |
|  | lea dx, game\_over\_message |
|  | call print |
|  |  |
|  | lea dx, player\_message |
|  | call print |
|  |  |
|  | lea dx, player |
|  | call print |
|  |  |
|  | lea dx, win\_message |
|  | call print |
|  |  |
|  | jmp fim |
|  |  |
|  |  |
|  | set\_game\_pointer: |
|  | lea si, game\_draw |
|  | lea bx, game\_pointer |
|  |  |
|  | mov cx, 9 |
|  |  |
|  | loop\_1: |
|  | cmp cx, 6 |
|  | je add\_1 |
|  |  |
|  | cmp cx, 3 |
|  | je add\_1 |
|  |  |
|  | jmp add\_2 |
|  |  |
|  | add\_1: |
|  | add si, 1 |
|  | jmp add\_2 |
|  |  |
|  | add\_2: |
|  | mov ds:[bx], si |
|  | add si, 2 |
|  |  |
|  | inc bx |
|  | loop loop\_1 |
|  |  |
|  | ret |
|  |  |
|  |  |
|  | print: ; print dx content |
|  | mov ah, 9 |
|  | int 21h |
|  |  |
|  | ret |
|  |  |
|  |  |
|  | clear\_screen: ; get and set video mode |
|  | mov ah, 0fh |
|  | int 10h |
|  |  |
|  | mov ah, 0 |
|  | int 10h |
|  |  |
|  | ret |
|  |  |
|  |  |
|  | read\_keyboard: ; read keybord and return content in ah |
|  | mov ah, 1 |
|  | int 21h |
|  |  |
|  | ret |
|  |  |
|  |  |
|  | fim: |
|  | jmp fim |
|  |  |
|  | code ends |
|  |  |
|  | end start |