

Lab5 - Modo Gráfico

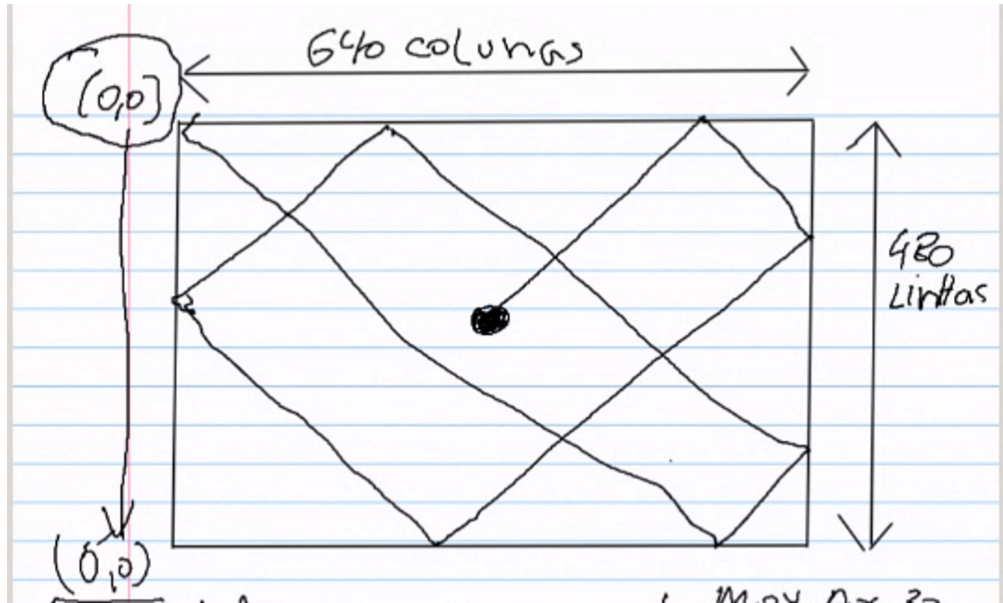
sexta-feira, 16 de outubro de 2020 11:06

A ideia desse laboratório é fazer um círculo bater e repetir por todos os cantos da tela do DOSBOX.

Vamos usar a biblioteca disponibilizada: linec.asm

A primeira coisa é entender como funciona o modo de gráfico do DosBox

Tela do DosBOX - Modo VGA - 16 níveis de cores



São 16 níveis de cores, ou seja, são 4 bits para definir as cores.

A definição das cores está comentado na biblioteca dessa forma:

```
658 ; I R G B COR
659 ; 0 0 0 0 preto
660 ; 0 0 0 1 azul
661 ; 0 0 1 0 verde
662 ; 0 0 1 1 cyan
663 ; 0 1 0 0 vermelho
664 ; 0 1 0 1 magenta
665 ; 0 1 1 0 marrom
666 ; 0 1 1 1 branco
667 ; 1 0 0 0 cinza
668 ; 1 0 0 1 azul claro
669 ; 1 0 1 0 verde claro
670 ; 1 0 1 1 cyan claro
671 ; 1 1 0 0 rosa
672 ; 1 1 0 1 magenta claro
673 ; 1 1 1 0 amarelo
674 ; 1 1 1 1 branco intenso
675
676 preto      equ 0
677 azul      equ 1
678 verde     equ 2
679 cyan      equ 3
```

677	azul	equ	1
678	verde	equ	2
679	cyan	equ	3
680	vermelho	equ	4
681	magenta	equ	5
682	marrom	equ	6
683	branco	equ	7
684	cinza	equ	8
685	azul_claro	equ	9
686	verde claro	equ	10

Onde a variável "preto" recebe o valor 0 em decimal, que corresponde 0000 em binário.

Desenho do círculo colorido.

Para isso, usamos a função full_circle da biblioteca linec.asm.

Para desenhar o círculo basta fazer assim:

```

37 ;pinta o círculo de branco
38     mov     byte[cor],branco_intenso
39     mov     ax,[posicao_x]
40     push    ax
41     mov     ax,[posicao_y]
42     push    ax
43     mov     ax,[raio]
44     push    ax
45     call    full_circle

```

Onde, posicao_x, posicao_y e raio, foram declarados desse jeito:

701	posicao_x	dw	320
702	posicao_y	dw	240
703	raio	dw	10
704	velocidade	dw	50
705	somaY	dw	5
706	somaX	dw	5

Deslocamento pela tela

Para fazer o círculo "andar" na tela, temos que pintar de branco e de preto para dar essa sensação de deslocamento. Se você não pintar de preto, vai sujar toda a tela.

Vale ressaltar também que entre a pintura branca e preta, você deve inserir um delay, que é disponibilizado pelo professor no roteiro do laboratório 5.

O dessa parte se encontra na linha 37 a 61.

```

37 ;pinta o circulo de branco
38     mov     byte[cor],branco_intenso
39     mov     ax,[posicaoax]
40     push    ax
41     mov     ax,[posicaoy]
42     push    ax
43     mov     ax,[raio]
44     push    ax
45     call    full_circle
46
47 ; delay
48     push    cx ; salvando cx
49     call    delay
50     pop     cx
51
52
53 ;pinta o circulo de preto
54     mov     byte[cor],preto
55     mov     ax,[posicaoax]
56     push    ax
57     mov     ax,[posicaoy]
58     push    ax
59     mov     ax,[raio]
60     push    ax
61     call    full_circle
62

```

Encontrando limite da tela

Para fazer isso, você deve verificar pelo comando **cmp**.

Exemplo da extensão de Y.

Sabemos que o máximo valor de Y é: **480**. Então a lógica fica assim:

```

89 ; logica para rebate do y
90     cmp     word[posicaoy],470
91     je      inverte_Y
92     cmp     word[posicaoy],10
93     je      normal_Y
94     jmp     continuaY
95
96 inverte_Y:
97     mov     word[somaY],-5
98     jmp     continuaY
99
100 normal_Y:
101     mov     word[somaY],5
102     jmp     continuaY
103
104 continuaY:
105     mov     ax,word[somaY]
106     add     word[posicaoy],ax ;posicaoy = posicaoy+5
107 ; fim da logica do rebate do y
108

```

Finalização do programa

Isso, na verdade, é a primeira coisa que se programa.

Você vai fazer que sempre que o programa rode, ele verifique se algo no teclado foi apertado.

Se algo tiver sido apertado, você vai verifica se foi a tecla 's'

Esse código foi disponibilizado pelo professor.

```
23 volta:
24     ; forma de sair
25     mov ah,0bh
26     int 21h ; Le buffer de teclado
27     cmp al,0 ; Se AL =0 nada foi digitado. Se AL =255 então há algum caracter na STDIN
28     jne adelante
29     jmp segue ; se AL = 0 então nada foi digitado e a animação do jogo deve continuar
30 adelante:
31     mov ah, 08H ;Ler caracter da STDIN
32     int 21H
33     cmp al, 's' ;Verifica se foi 's'. Se foi, finaliza o programa
34     jne segue
35     jmp sai
36 segue:
```