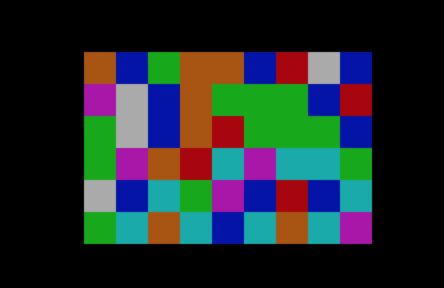
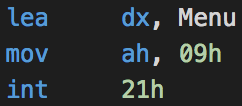
TAC

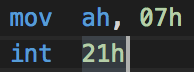


Eduardo Barros – 21270614

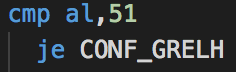
Henrique Dias – 21260023

MENUS:

* Foram criados menus e vários submenus, usando strings (de tamanho byte), delimitadas por 10,13
* Imprimimos em ecrã através de um LOAD EFETIVE ADRESS da string antes criada:
* Usando para isso a interrupção 09h (escreve uma string para STDOUT)



* Espera que o utilizador introduza um carater com interrupção 07h (leitura direta do STDIN)



* Depois de introduzido, faz varias comparações para saber a qual submenu/instrução aceder (sendo que para o exemplo ‘51’ é o numero 3)

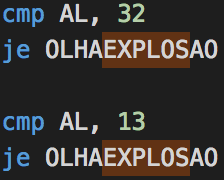
VETOR:

* Colocamos o vetor de memoria de vídeo em um vetor, para que seja possível trabalharmos no data segment apenas com um array unidimensional, isto através do procedimento:

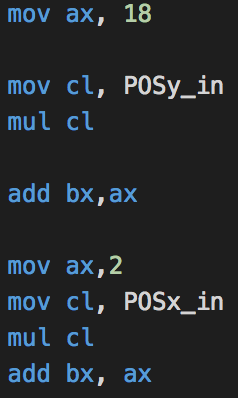
LIMITES DO CURSOR:

* Depois de inicializado o tabuleiro, e de colocado o cursor (fontes do professor), temos de limitar o alcance do cursor conforme as dimensões do tabuleiro

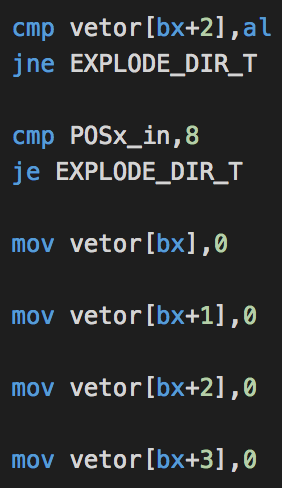
EXPLOSÕES:



* A função é ativada se for pressionado ENTER ou SPACE (ver tópico dos menus)

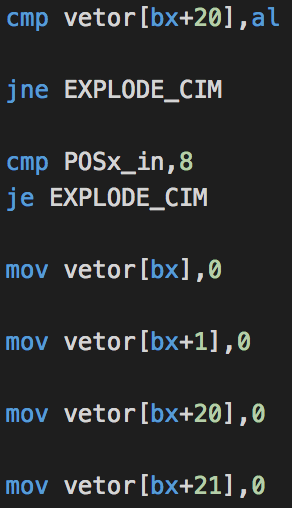


* É colocado o bx na posição em que o cursor estava quando pressionado ENTER ou SPACE, para posteriormente acedermos a essa posição no vetor.
* De linha para linha existe uma diferença de 18 (pois são 9 colunas sendo que cada uma constitui dois bytes)
* De coluna para coluna existe uma diferença de 2 (pois cada coluna = 2 bytes)
* Fazendo assim a conta bx = (posy\*18)+(posx\*2)
* Agora podemos saltar para cada posição que nos interessa, começando pela posição á frente



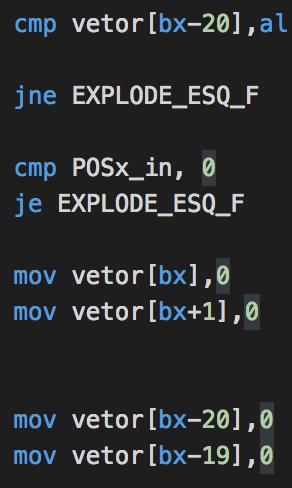
* Vetor[bx+2]=posição á frente | al=cor carregada
* Se não for igual salta para a posição seguinte (direita-topo)
* Se estivermos na 8ºcoluna não explodimos á frente, dessa forma iria explodir a linha de baixo
* Colocamos o primeiro byte do meio a preto
* Colocamos o segundo byte do meio a preto
* Colocamos o primeiro byte da posição a frente a preto
* Colocamos o segundo byte da posição a frente a preto



* Saltamos no fim para a posição seguinte (direita baixo)
* Sendo que a logica será a mesma
* Vetor[bx+20]=posição á frente-baixo| al=cor carregada
* Se a cor não for igual saltamos para uma nova posição
* Se estivermos na 8ºcoluna não explodimos á frente, dessa forma iria explodir a linha de baixo
* Colocamos o primeiro byte do meio a preto
* Colocamos o segundo byte do meio a preto
* Colocamos o primeiro byte da posição a frente-baixo a preto
* Colocamos o segundo byte da posição a frente-baixo a preto

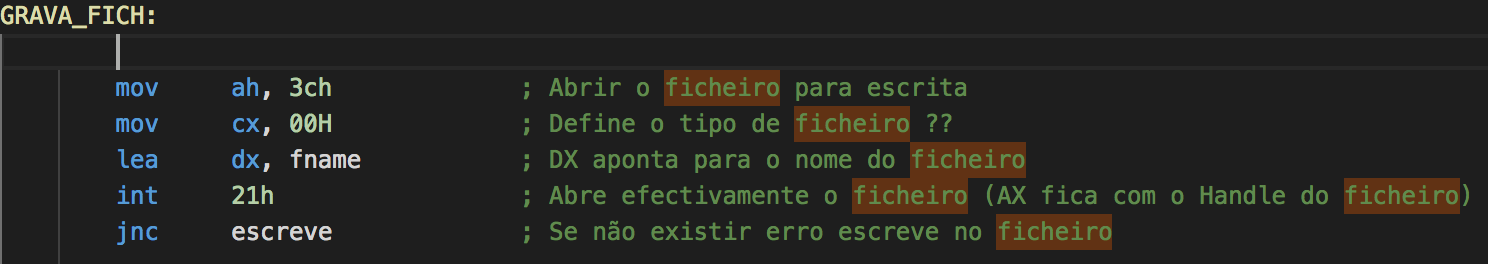


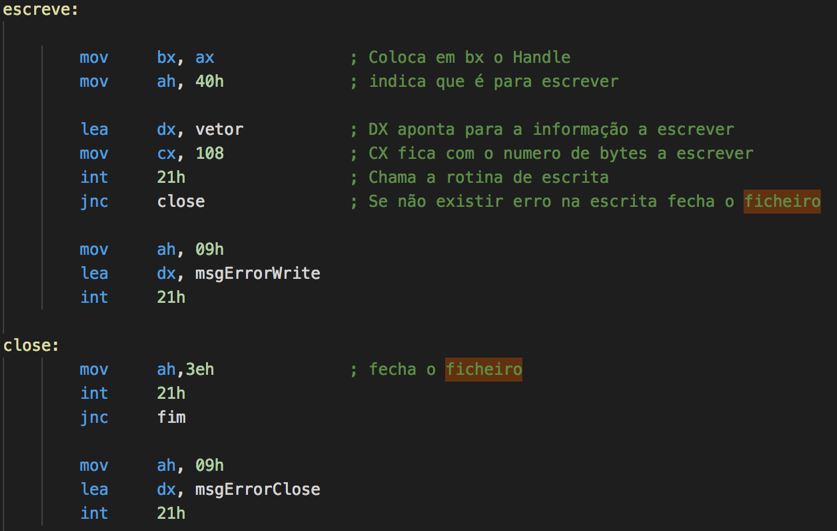
* É de notar que nestes procedimentos incrementamos uma variável para ser mais tarde usada na pontuação
* Usamos o mesmo procedimento em todos as posições desejadas, deixamos mais uma em exemplo:

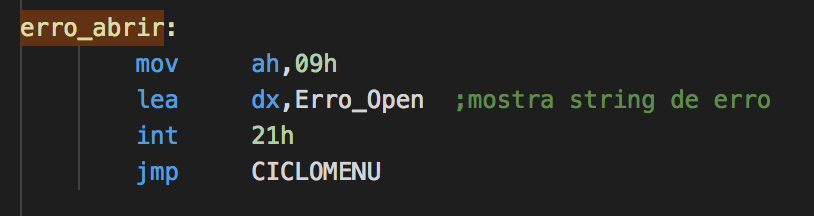


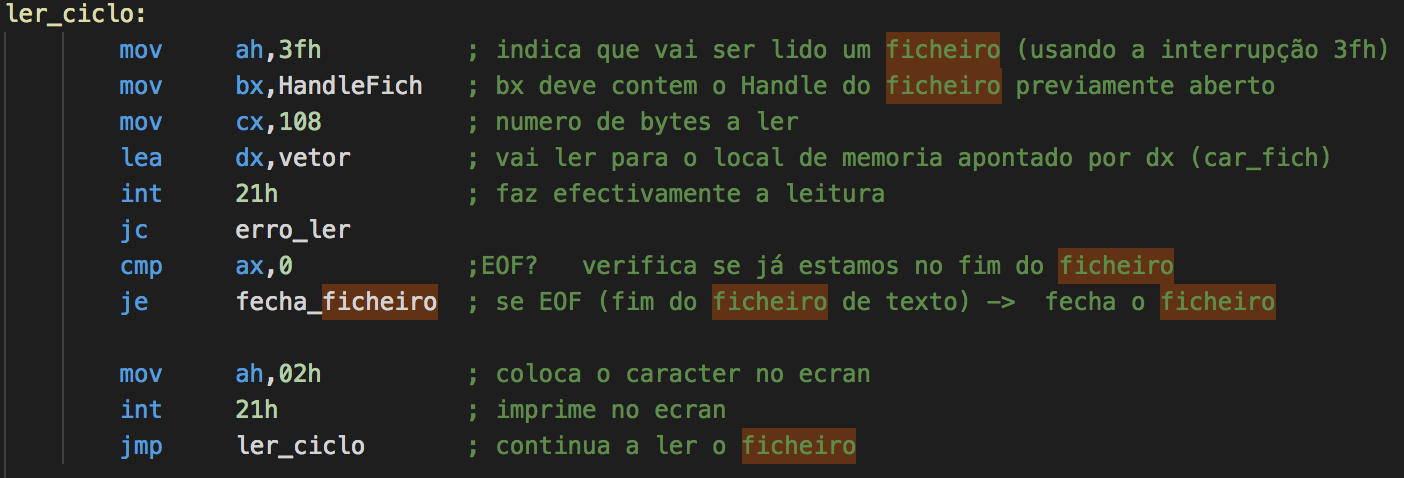
* Vetor[bx-20]=posição á esquerda-cima| al=cor carregada
* Se não for igual passamos á próxima
* Se estivermos na 1ºcoluna não explodimos atras, dessa forma iria explodir a linha de cima
* Colocamos o primeiro byte do meio a preto
* Colocamos o segundo byte do meio a preto
* Colocamos o primeiro byte da posição atras-cima a preto
* Colocamos o segundo byte da posição atras-cima a preto

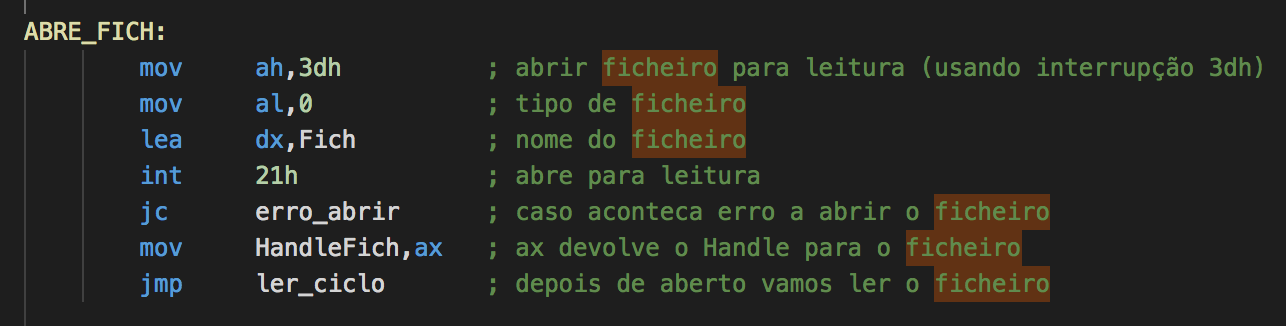
FICHEIRO:

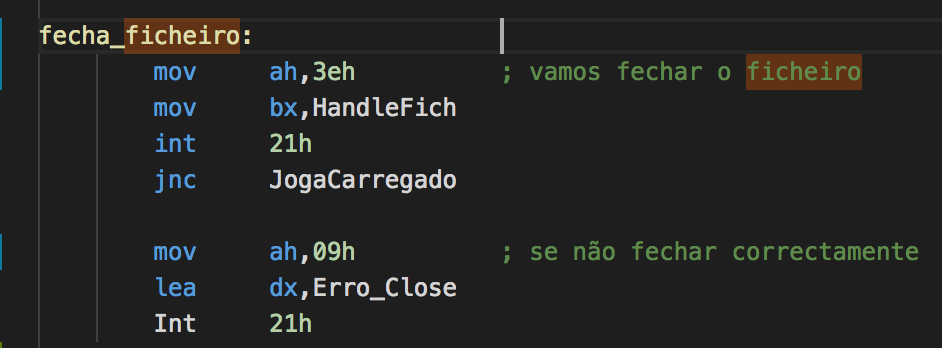








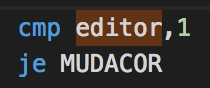
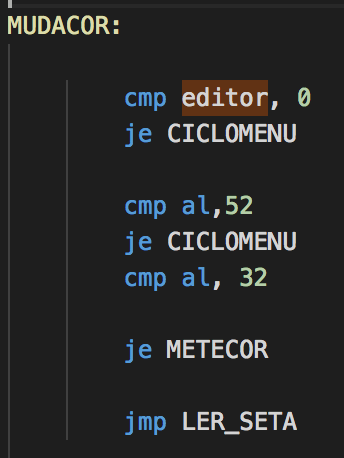




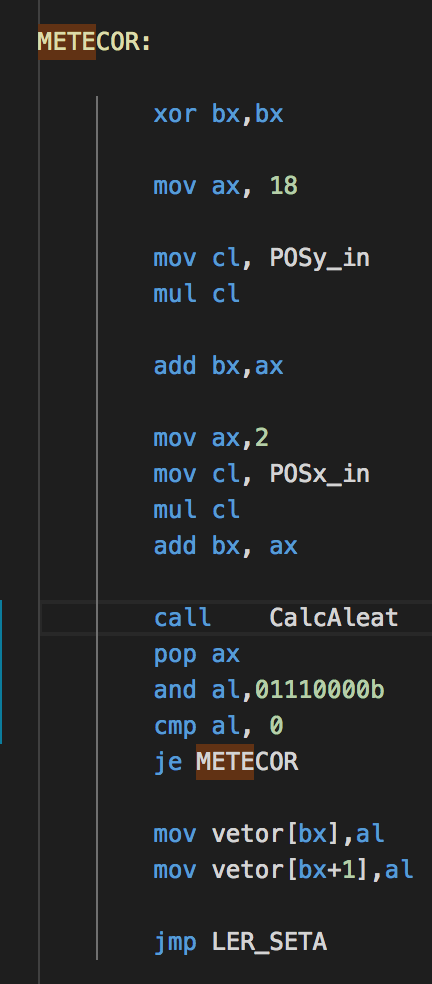
CRIAR GRELHA:



* CONF\_GRELH, é chamado no menu
* Se for para o jogo, editor=0, se for para configurar a grelha, editor=1
* Vai para o display onde está o tabuleiro e o cursor
* Vai para o menu inicial



* Volta para o menu se o editor passar a ser 0
* Se for carregado o ‘4’, retrocede
* Se for carregado o SPACE salta para METECOR
* Se carregar noutra tecla continua o ciclo



* Vai colocar o valor em bx, da atual posição do cursor, para dessa forma conseguirmos aceder á exata posição do array onde ele se encontra
* Colocando em bx = (posx\*2)+(posy\*18), como já averiguamos antes
* Chamamos a função de calculo aleatório para preencher a atual posição onde nos encontramos
* Verificamos se a cor gerada é preta para não a colocarmos
* Preenchemos o primeiro byte de atual posição
* Preenchemos o segundo byte de atual posição
* Voltamos ao ciclo

Tempo:

Pontuacao:

Guardar grelha:

ESTRUTURA, OBJETIVOS, FUNCIONAMENTO do programa e o ALGORITMO subjacente a cada função desenvolvida, bem como as principais opções de desenvolvimento.