

# INTRODUÇÃO À ARQUITETURA DE COMPUTADORES

LEIC

IST-ALAMEDA

2014/2015



# TRON

Realizado por:  
Rafael Belchior, nº80970, LEIC  
Gonçalo Baptista, nº 81196, LEIC  
Jorge Pereira, nº 81498, LEIC

## 1. Introdução

O projeto consiste no desenvolvimento de uma versão do jogo TRON em que o objetivo é bloquear o avanço do adversário. O jogo tem 2 jogadores e decorre na janela de texto do P3. Na janela de texto cada jogador deverá conduzir a sua partícula escolhendo mudanças de direção (esquerda ou direita) de modo a limitar o espaço ao adversário e a evitar colisões com os limites do espaço de jogo e o seu rasto. Se o jogador colidir com o rasto do adversário, com o seu rasto, ou com o limite do jogo, perde. Cada vez que um jogador perde, é somado um ponto ao jogador adversário. Se os jogadores perderem ao mesmo tempo, é declarado um empate. À medida que o tempo passa, a dificuldade do jogo vai aumentando, ou seja, cada mota vai mover-se a velocidades sucessivamente maiores, nomeadamente aos tempos 10s, 20s, 40s e 60s de jogo.

## 2. Organização do programa

O código da versão do jogo TRON encontra-se organizada em quatro partes principais:

- 1) ZONA I: Definição de constantes: As constantes usadas ao longo do programa estão presentes nesta zona. As constantes aqui presentes servem para inicializar a pilha, definir a máscara de interrupções, indicar portos utilizados na conceção do jogo, indicar informações acerca dos níveis, indicar localização das mensagens no espaço de jogo e variáveis de controlo associadas ao funcionamento do relógio.
- 2) ZONA II: Definição de variáveis: Nesta zona encontram-se as variáveis utilizadas ao longo do programa. Estão presentes variáveis relacionadas com o funcionamento do relógio e cadeias de caracteres a imprimir no espaço de jogo, assim como o tamanho do espaço de jogo. Salientar as variáveis:
  1. "Vitorias#" que corresponde ao numero de vitorias do #"
  2. "VitoriasX" que corresponde ao número de vitórias do X
  3. "Jogou#", verifica se foi o J2 (#) que jogou
  4. "JogouX" verifica se foi o J1 (X) que jogou
  5. "TempoMAX" é a variável que guarda o tempo máximo já jogado
  6. "PreGameOverC" pode ter 4 estados: 0 se nenhum jogador perdeu, 1 se perdeu o jogador 1 (X), 2 se perdeu o jogador 2 (#) e 3 se perderam ambos (empate). A constante DezenasJ1 corresponde à posição das dezenas do jogador 1, no LCD (analogamente para as unidades e o J2).

7. “PosicoesIguais” fica a 1 se os dois jogadores estiverem em posições iguais (um dos casos de empate).
- 3) ZONA III: Tabela de interrupções: Nesta zona encontra-se a tabela de interrupções, que corresponde às diversas interrupções a utilizar durante o jogo. As interrupções correspondem ao início do jogo, movimento dos jogadores para a esquerda ou direita e uma interrupção cujo vetor de interrupções é constante, o temporizador.
- 4) ZONA IV: Código: Nesta zona encontra-se o código do programa, comentado e indentado. O código é constituído pelo código principal, pelas rotinas de serviço às interrupções, e pelas rotinas necessárias ao funcionamento correto do jogo.

### **3. Aspetos relevantes da implementação**

#### **1) Jogo (código principal)**

Para a passagem de parâmetros através da pilha ser possível, a pilha teve de ser inicializada.

A representação do espaço de jogo foi feita com recurso a uma matriz.

É de salientar a atribuição do registo R4 à direção do movimento inicial do jogador 1, para cima (X) e a atribuição do registo R5 à direção do movimento inicial do jogador 2, para baixo (#).

Recorre-se a uma variável de estado para esperar pela interrupção que corresponde ao início do jogo. Através do interruptor I1, é acionada a interrupção correspondente à etiqueta “Comeca”, localizada na zona de interrupções. Esta coloca a variável de estado a 1, o que permite o início do jogo.

O ciclo inicial do jogo corresponde à escrita da mensagem de boas vindas, possibilitada pela limpeza da janela, pela escrita do ecrã inicial, correspondente às strings 'Bem-vindo ao TRON', 'Pressione I1 para começar'.

Posteriormente, através do CicloInicio, o relógio é ativado, possibilitando o início do jogo.

Por sua vez, a rotina “Jogo” vai limpar o espaço de jogo, voltar a escrever os limites do espaço de jogo, e verificar o nível correspondente à temporização atual. Dependendo do nível atual, a rotina vai alterar a posição dos jogadores, tendo em conta a direção atual (esquerda, direita, cima ou baixo), de maneira cíclica. Além disso, a função pausa está implementada: para o contador, e imprime uma mensagem de pausa. Depois de iniciado, o jogo vai incrementando o contador e verifica a direção dos jogadores. Quando ocorrer uma colisão, verifica-se qual o jogador que colidiu e se houve empate. Em qualquer dos casos, o LCD é atualizado (pontuação é atualizada,

o tempo máximo é atualizado), limpa-se a memória e espera-se pela interrupção de início de jogo.

## 2) Interrupções

A interrupção I1 tem a função de colocar a variável de estado “Estado” a 1, responsável pelo início do jogo.

As interrupções são responsáveis pela alteração da direção das motas: no caso da direção esquerda do jogador X, correspondente à interrupção I0 (pode ser generalizado para todos). Quando o registo R4 está a 0100h e é chamada a interrupção correspondente a virar para a esquerda (correspondente ao interruptor I0), ocorre uma rotação aritmética para a esquerda. Essa interrupção de 4 bits irá fazer com que a nova direção seja 1000h, (esquerda) ou seja, a interrupção permitiu alterar a direção de cima para esquerda. Através de sucessivas rotações para a esquerda consegue-se fazer com que a mota vá para a direção desejada.

## 3) Rotinas

- 1) As rotinas que atendem o movimento para a direção propriamente dita são as rotinas MoveBaixo#, MoveBaixoX, MoveCima#, MoveCimaX, MoveEsquerda#, MoveDireitaX, MoveDireita#, MoveEsquerdaX. No caso da direção esquerda e da direita, a rotina respetiva vai decrementar ou incrementar a posição de memória que contém a posição da mota num determinado momento. No caso da direção cima e baixo, a rotina respetiva vai subtrair ou somar uma linha da matriz (0100h) à posição de memória que contém a posição da mota num determinado momento.
- 2) “Escstring” efetua a escrita de uma cadeia de caracteres, terminada pelo carácter FIM\_TEXTO, na janela de texto numa posição especificada. Neste caso, o carácter fim texto é a @(arroba). A rotina “EscString” recebe dois parâmetros através da pilha, movendo-os para dois registos. Um dos registos contém a posição de memória que aponta para o início da string, e o outro a localização da posição do primeiro carácter. Utilizando-se o porto de escrita, é possível escrever-se uma mensagem no ecrã.
- 3) “EcrInicial”, “EscCantos”, “EscLinhas”, “EscColunas”, “EscMotas” são todas feitas com o auxílio de outras rotinas: Escolher a string ou carácter a escrever e a posição de escrita. A rotina “EscString” ocupar-se-á da escrita dos caracteres no local apropriado. A rotina “VerificaColisoes” ocupa-se de colocar a flag a 1, para indicar que essa posição de memória corresponde a um extremo do espaço de jogo.

- 4) "LimpaJanela" vai recorrer a várias variáveis: uma para o número de colunas, uma para o número de linhas e uma para permitir apagar a matriz. Para cada coluna, o seu comprimento é decrementado (48 colunas mais 2 da moldura). Quando a variável chegar a zero significa que essa linha já está "limpa". O número das linha é decrementa, até chegar a zero, o que significa que já não há linhas para "limpar". Para limpar a matriz, para cada coordenada da matriz é impressa a constante 0, que indica que já está limpa.
- 5) A zona correspondente ao MudaNível possui várias rotinas: Uma rotina para a alteração de cada nível e uma rotina de controlo. A rotina de controlo, "VerificaNivel" vai comparar o valor que está presente no relógio com o valor necessário para começar um novo nível. Caso se verifique, os LEDs são atualizados e o nível é atualizado.
- 6) A rotina "EscreveCaracterLCD" define a posição na qual se quer escrever no LCD, e move para lá o carácter correspondente em ASCII.
- 7) A rotina "EscreveStringLCD" define a posição na qual se quer escrever no LCD, e move para lá o carácter correspondente em ASCII, por linha, até se chegar ao final da string. Utiliza a rotina "EscreveCaraterLCD" para escrever carater a carater.
- 8) A rotina "Direcoes#" e "DirecoesX" vai chamar a rotina que realiza a direção, dependente do valor que o jogador estava a mover-se.
- 9) As rotinas Move[1][2] vão fazer a partícula mover-se. O valor da posição atual da mota é passado para um registo. Esse valor é submetido a uma operação que fará que tome a direção escolhida. Se a partícula se vai mover para cima, é-lhe subtraída uma linha (0100h), e para baixo é-lhe adicionada uma linha. Se a partícula se for mover para a esquerda, decrementa-se uma posição de memória, e se for para a direita adiciona-se uma posição de memória. Ainda se verifica se ocorreram colisões, através do "VerificaColisoes" LEGENDA: [1] – "Esquerda", "direita", "cima" ou "baixo"; [2] – "# "ou "X".
- 10) A rotina "EscCont" é responsável por escrever o relógio no display. Esta rotina foi implementada baseando-nos no código da atividade laboratorial 5.
- 11) A rotina "VerificaColisoes" obtém as coordenadas da mota, e depois a sua linha e coluna. Posteriormente obtém-se as coordenadas. Compara-se a posição de memória da matriz com 0. Se for 0, a posição está livre e vai ser escrita com 1. Caso contrário, vão ser verificados os casos de fim de jogo, na rotina "PreGameOver".
- 12) A rotina "GameOver" vai escrever a mensagem de fim de jogo e verificar se os dois jogadores se encontram na mesma posição, e em caso positivo, verifica-se um empate. Caso haja um empate, é somado um ponto a cada um dos jogadores. Caso não haja empate, vai-se verificar quem ganhou, com recurso à flag "Jogou#" e "JogouX". Se a flag de um estiver a 1, significa que o jogador jogou e colidiu.

- 13) “RotPausa” escreve a mensagem de pausa, e para o relógio. A rotina “ApagaPausa” limpa as cadeias de caracteres escritas com recurso à rotina “EscString” e o relógio é reiniciado.

#### **4) Conclusão**

Os objetivos iniciais foram cumpridos. Foi criada uma versão do jogo TRON funcional. As partículas mexem-se na direção pretendida, de acordo com a interrupção escolhida. Os níveis de jogo foram implementados de acordo com o enunciado. Os LEDs ligam-se de acordo com o nível e o LCD mostra as pontuações e tempo máximo atualizadas. É feita uma verificação de colisão das partículas com o limite do espaço de jogo, com o adversário e com ela própria. Após ocorrer uma colisão, são registadas pontuações e o novo tempo máximo, se for o caso. Além disso, foi implementada a função extra “Pausa” que permite pôr o jogo em pausa.