

LEIC-A
2014/2015
1º Semestre



TÉCNICO
LISBOA

Introdução à Arquitetura
de Computadores

TRON

Relatório do projeto

Grupo 16

Rui Ventura	81045
Joel Almeida	81609
Nádia Mourão	82259

Introdução

Este projeto consiste na conceção do conhecido videojogo TRON, através da linguagem de programação *Assembly* do P3, utilizando o simulador do processador P3 para desenvolver e testar o projeto para ultimamente correr no processador físico.

Sendo um jogo com dois jogadores, o objetivo é bloquear o avanço do adversário e não colidi, seja contra o rasto deixado pelos jogadores ou as fronteiras do campo. O jogo decorre na janela de texto do P3. Nesta, os jogadores conduzem a partícula correspondente podendo mudar a direção da mesma (utilizando botões de interrupção) com o intuito de provocar a colisão do adversário sem colidir no processo.

Os limites do jogo estão representados pelos caracteres “+” (cantos), “|” (paredes), e “-” (tecto/chão). As partículas são representados pelos caracteres “X” e “#”, correspondendo ao jogador 1 e 2, respectivamente, que começam nas posições (8,10) e (40,10) do espaço de jogo e o rasto deixado por cada um é representado pelo mesmo caracter.

O jogo tem 5 níveis que vão aumentando periódica e não linearmente (representados nos LEDs na placa do P3), segundo a correspondência seguinte:

	Intervalo/mov (s)	Tempo no jogo (s)
Nível 1	0.7	0-10
Nível 2	0.5	10-20
Nível 3	0.3	20-40
Nível 4	0.2	40-60
Nível 5	0.1	> 60

Quando um jogador colide contra a fronteira do campo de jogo ou contra um rasto, quer seja o seu ou não, é dada uma vitória ao adversário. Se colidirem entre si ou na mesma jogada, é considerado empate e as pontuações dos jogadores mantêm-se inalteradas. As pontuações dos jogadores, bem como o tempo máximo de jogo alcançado ao longo das partidas, são representados no LCD na placa do P3.

O Jogo

A implementação do TRON em código Assembly pode ser dividida em três partes principais:

- Ecrã de boas vindas, em que o utilizador é solicitado para premir um botão de interrupção (I1) para iniciar o jogo;
- Campo de jogo, podendo os utilizadores jogar propriamente, estando exposto o campo e as partículas;
- Ecrã de *game over*, que é apresentado após uma colisão.

Ecrã de boas vindas

O ecrã de boas vindas é apresentado apenas a primeira vez que o jogo é iniciado. Esta implementação é feita pela rotina Main, que só é chamada uma única vez no decurso do programa. É aí que são inicializados a pilha, a máscara de interrupções apenas com a interrupção 1, os elementos da placa (LEDs, 7seg e LCD) e o cursor da janela, sendo seguidamente apresentada a mensagem inicial do programa.

As rotinas com maior importância na escrita de informações na janela são a WriteStrW, que depende da rotina WriteCharW para escrever uma cadeia de caracteres numa determinada posição na janela (e em memória, uma vez que WriteCharW aproveita o cursor na janela para facilitar o processo para a deteção de colisões) e a SetupBoard, que utiliza as rotinas Write7Seg, VeriLevel e WriteLCD para imprimir informação no display de 7 segmentos (tempo decorrido), nos LEDs (nível actual) e no LCD (tempo máximo atingido e pontuações) da placa do P3.

A única interrupção ativa neste ecrã é a interrupção 1 (I1). Enquanto esta não for accionada, o programa mantém-se no ecrã de boas vindas. Quando ativada, o jogo começa e é exposto o campo do jogo.

Campo do jogo

Neste ecrã o programa chama primeiro a rotina ClearScreen para limpar na totalidade a janela de texto, substituindo todos os valores em memória e os valores na janela de texto por espaços (' '), escrevendo depois as fronteiras do tabuleiro recorrendo às rotinas DrawWall e DrawCorners. De seguida, as partículas de cada jogador são desenhadas nas suas posições respetivas, com a rotina DrawParticles, e dá-se então início ao jogo, que a cada ciclo de relógio atualiza o nível, o relógio, os LEDs e o LCD com as respetivas informações, move a partícula dos jogadores para posição seguinte consoante a sua direção, e por fim verifica se existiu uma colisão. De notar que apenas atualiza a informação se for necessário (atualizará o display de 7 segmentos se tiver passado 1 segundo, verificando se transitou de nível nesse mesmo instante; moverá as

partículas se o contador do intervalo entre movimentos corresponder ao do nível atual; escreverá o tempo máximo no LCD se o atual ultrapassar o anterior).

Cada partícula é movida de acordo com a sua direção, que é controlada com as respetivas interrupções. As tabelas seguintes representam as ações que o programa faz para o movimento das partículas (utilizando a rotina MoveParticles):

Interrupção	Ação
INT0	Muda a direção do jogador 1 para a esquerda
INTB	Muda a direção do jogador 1 para a direita
INT7	Muda a direção do jogador 2 para a esquerda
INT9	Muda a direção do jogador 2 para a direita

Direção	Ação
Esquerda	Soma 1h à posição atual (coluna seguinte) e escreve a partícula
Direita	Soma FFFFh (ou subtrai 1h) à posição atual (coluna anterior) e escreve a partícula
Cima	Soma FE00h (ou subtrai 0100h) à posição atual (linha anterior) e escreve a partícula
Baixo	Soma 0100h à posição atual (linha seguinte) e escreve a partícula

A rotina Collision é chamada sempre que uma partícula é movida, verificando se a partícula colidiu contra um jogador ou uma fronteira. Determina o vencedor, se não for caso de empate, e indica o estado do jogo como estando terminado uma vez detetada uma colisão.

Existe ainda um interruptor que aciona a rotina Paused, que pausa o jogo (interruptor 7, o interruptor “mais significativo”, no extremo esquerdo da placa).

Ecrã de game over

Após uma colisão, é apresentada uma mensagem indicativa de que o jogo terminou. Neste ecrã, tal como no de boas vindas, apenas é possível utilizar a interrupção I1 para o recomeço do jogo. Este ecrã é chamado pela rotina GameOver quando o estado do jogo indicar que este foi terminado.

Conclusão

O projeto de criar um jogo em Assembly foi interessante pela sua temática. Pudemos finalmente aplicar os conhecimentos adquiridos nas aulas e nos laboratórios, tendo oportunidade de utilizar todas as funcionalidades e ferramentas (se não, uma grande parte das mais “expostas”) do processador P3, entre as quais a janela de texto, o display de 7 segmentos, o LCD, os LEDs, as interrupções e os interruptores, bem como o temporizador. Apesar do projeto ser cativante, encontrámos grandes dificuldades na sua execução, especialmente no que toca às diferenças entre o simulador e a placa, mas tendo conseguido superar as dificuldades com sucesso no final.

Além dos problemas com a placa, as tarefas que se mostraram ser mais complexas foram a escrita de dados no LCD e a forma como o programa lida com as interrupções (por exemplo, impedir que utilizador mude a posição da partícula mais do que uma vez num mesmo ciclo de movimento, por intervalo correspondente ao nível actual, e a forma como o programa lida com interrupções pendentes, havendo a necessidade de precaver a situação em que um dos jogadores prime a interrupção para recomeçar o jogo durante o mesmo, pré-ecrã de game over).

O projeto segue à risca o melhor possível todas as instruções dadas no enunciado, contendo ainda a funcionalidade de colocar o jogo em pausa utilizando um interruptor. A implementação de outras funcionalidades ou características extra para além desta foi comprometida devido a restrições temporais.