Instituto Superior técnico

Introdução à Arquitetura de Computadores

Relatório do projeto

Grupo 29
Carolina Inês Maltez Xavier, nº81172
Bernardo Dias Simões, nº81590
Pedro Miguel Carvalho Caldeira, nº81888
Turno 13

No âmbito da cadeira de Introdução à Arquitetura de Computadores desenvolvemos o jogo 'TRON' em Assembly para o processador P3. No código do nosso jogo, criámos várias rotinas que utilizam os diversos periféricos disponíveis no P3 (como os LEDs, o display de 7 segmentos e o LCD).

Na elaboração do projeto começamos por criar os fluxogramas, pois isso dá-nos uma ideia das rotinas que serão necessário criar e da estrutura geral do nosso código, permitindo assim organizar melhor o nosso tempo.

Na escrita do código. A nossa ordem de trabalhos foi a seguinte:

- Definir as rotinas que permitem escrever a moldura do jogo (DesenhaMoldura) e a mensagem de início (Bem_vindo), e implementar a interrupção que dá início ao jogo, colocando as partículas nas suas posições iniciais (Movimentos);

- Criar o temporizador, que irá ser utilizado ao longo do programa para várias funções: como contador do tempo do jogo e para controlar a velocidade das partículas e as mudanças de nível;

- Iniciar o movimento das partículas (Movimentos);

- Criar a tabela onde vão ficar guardadas todas as posições já percorridas por uma partícula – rastos. E definir, numa sub-rotina de Movimentos, a colisão das partículas com os rastos (Colisao) ou de uma partícula com a moldura (ColisaoMoldura);

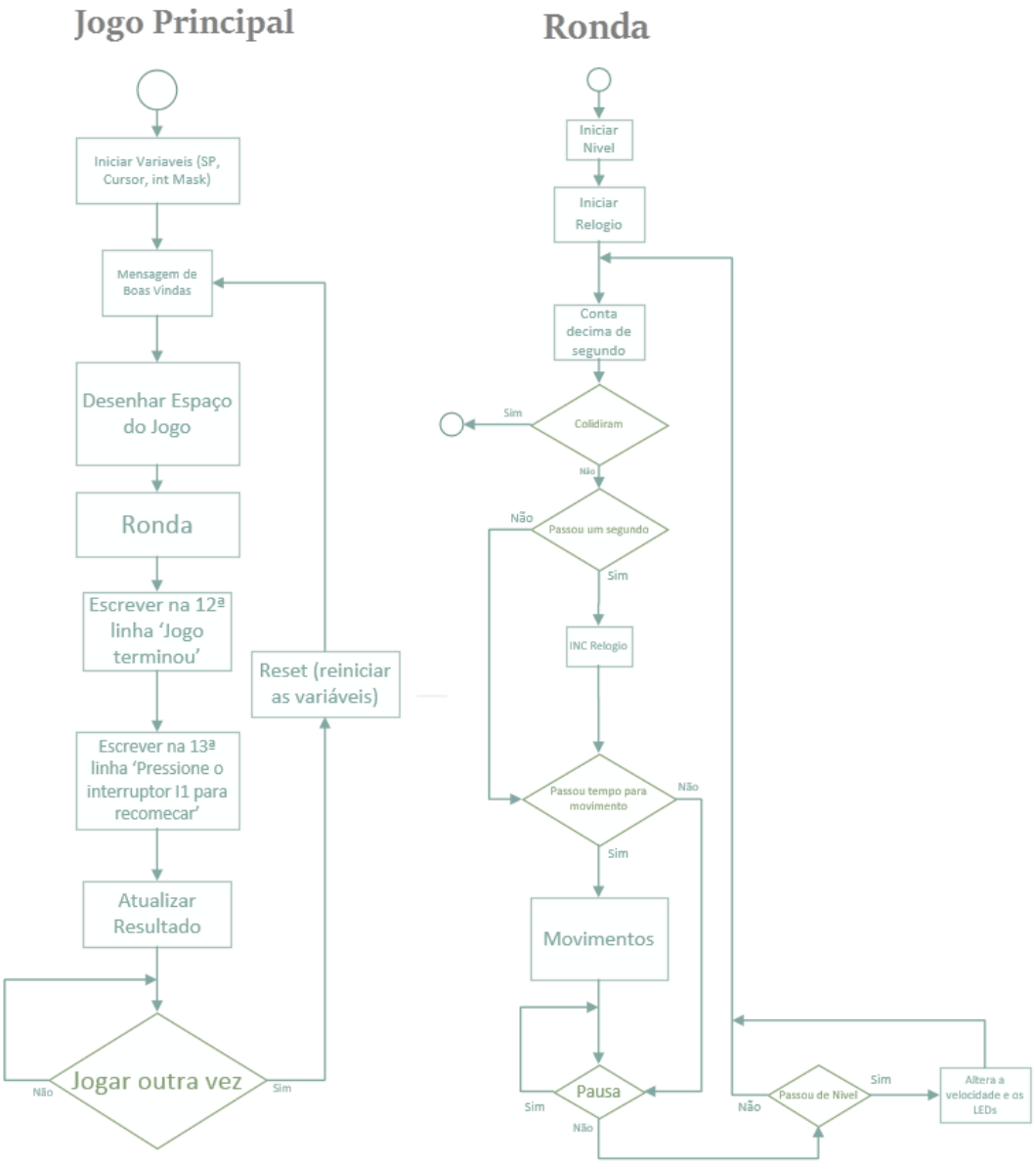
- Definir os movimentos das partículas consoante a interrupção, implementando vetores de sentido que são adicionados à posição da partícula e atualizando os vetores, dependendo do uso de um interruptor, que definem o movimento das partículas (VerificaDireccao, AlteraSentido). Esta rotina atualiza ainda a tabela dos rastos e escreve na janela;

- Definir o fim de jogo (RESULTADO), em que se verificam as colisões para ver qual o jogador vencedor, ou se houve empate, atualizando-se as pontuações e o tempo máximo. E escrita da mensagem de fim de jogo;

- Definir a rotina que permite voltar a jogar (Reset), que reinicia todas as variáveis, como as posições das partículas, reinicia os LEDs, o temporizador, a tabela e o movimento das partículas. Voltando-se de seguida à parte da rotina de estrutura do jogo que permite reiniciar o jogo (OutroJogo);

- Por fim definimos rotina de pausa (Pausa) – função adicional – que permite pausar o jogo se o interruptor 7 estiver para cima.

Funções que servem de base ao nosso jogo



Conclusão

Divergências:

No nosso jogo são evidentes alguns desvios do enunciado, pois fizemos algumas alterações que achámos que melhorariam o nosso jogo:

- Optamos por atribuir uma representação diferente aos rastos das partículas, para que, durante o jogo, seja mais fácil aos jogadores distinguir os rastos já existentes da posição atual das suas partículas (o rasto da partícula X fica representado com um '+' e o da partícula # com um 'O').

- Alterámos ainda a mensagem de fim de jogo, para que, para além da mensagem "Prima o interruptor I1 para recomeçar", informasse também qual o jogador vencedor dessa ronda, ou se houve empate.

- Como funcionalidades da versão avançada acrescentámos ao nosso projeto a opção de pausar o jogo (quando o interruptor 7 estiver para cima o jogo fica em pausa).

Devido à maneira como definimos a moldura, não criámos a opção de alterar do tamanho do espaço de jogo, porque para isso era necessário mudar integralmente o código. Pois a criação da nossa moldura consiste na escrita linha a linha através de strings de toda a moldura, o que facilita no reinício do jogo, pois não precisamos de apagar nada do que está escrito na janela de texto, apenas é necessário voltar a escrever a moldura.

Apesar de todos os desafios que enfrentamos para realizar o projeto (tendo sido um dos principais desafios para nós a criação da tabela com os rastos e a sua implementação no jogo) achamos que o resultado final foi bastante satisfatório e que conseguimos realizar o jogo com sucesso.

A partir do momento que conseguimos estruturar as ideias e à medida que nos fomos familiarizando com o projeto, fomos também tendo mais facilidade na sua realização. Podendo assim concluir que este projeto contribui, principalmente, para uma melhoria das nossas capacidades de programação em *Assembly*.