

Relatório do projeto de IAC – “DINO”

Introdução

O projeto de Introdução à Arquitetura de Computadores tinha como objetivo criar, em Assembly do P4, um jogo que imitasse o jogo Dino do Google Chrome. Nesse sentido, desenvolvemos um código com diversos módulos, cada um tratando de uma parte específica do jogo em si.

Estrutura do programa

O nosso código está dividido em duas partes principais: a preparação do terreno de jogo e a inicialização do jogo e as funções que manipulam as diferentes partes do jogo durante o seu decorrer. Assim, podemos separar em dois tipos as funções que usamos: o primeiro é o do tipo estático, que apenas é usado uma vez por cada iteração do jogo e o segundo tipo é o dinâmico pois estão constantemente a ser chamadas e os seus valores a serem utilizados

- **Funções Estáticas**

Primeiramente usamos uma função INICIOJOGO de modo a limpar a área de jogo bem como as posições em memória onde armazenamos a altura dos cactos do ecrã, de seguida usamos a função ESC_TERR para desenhar o terreno do jogo. Ambas estas funções são estáticas pois apenas são chamadas no início de cada jogo. Outra função estática é a função GAMEOVER, que apenas tem como intuito escrever GAME OVER quando os jogadores perdem, pois apenas é usada uma vez. Para a inicialização do jogo usamos um ciclo e a variável RESTART, permitindo apenas o código principal correr caso esta variável tivesse o valor de 1, o que apenas era possível caso fosse pressionado o botão 0.

- **Funções dinâmicas**

O segundo tipo de funções, as dinâmicas, são as que estão constantemente a ser chamadas para alterar o estado do jogo. Para atingir este efeito usamos o temporizador do simulador do P4, com um intervalo de tempo entre cada interrupção de 0,1s. Em cada interrupção são chamadas duas funções principais: a função SET_SCORE, que aumenta em 1 a pontuação do jogador de cada vez que é chamada e a mostra no display de 7 segmentos, a segunda função é a função ATUALIZAJOGO, cujo objetivo é atualizar o terminal e fazer os cactos do terreno de jogo andarem uma posição de cada vez para a esquerda sempre que é chamada. Estes cactos estão representados como números aleatórios em posições de memória correspondentes com o terreno de jogo. Além disso, por cada vez que a função ATUALIZAJOGO é chamada são chamadas as funções ESCRIVE_DINO e ESCRIVE_CACTO que têm como função fazer aparecer no ecrã o dino e os cactos nas posições desejadas.

- Decisões relativas ao desenho da aplicação

Nós decidimos agrupar assim as funções devido à independência entre os dois tipos, pois um deles é usado durante o jogo e outro é usado antes e depois do jogo. Consequentemente decidimos chamar as funções `ESCREVE_DINO` e `ESCREVE_CACTO` dentro da `ATUALIZAJOGO` devido à sua natureza que só é útil se forem atualizadas continuamente. Além disso a `ESCREVE_DINO` tem também um sub-ciclo que permite ao dino saltar cada vez que é pressionada a tecla UP do simulador do P4, do mesmo modo a `ESCREVE_CACTO` tem um sub-ciclo que permite detetar se houve colisões entre o cacto e o dino e terminar o jogo caso isso seja verdade.

Conclusão

Na realização deste trabalho tivemos algumas dificuldades quanto a organização e a disposição do código do jogo, bem como, na correção de pequenos bugs durante a escrita do projeto. Entretanto, consideramos que fizemos um bom trabalho e que as diversas adversidades que encontramos ao longo do caminho foram superadas, o que de certa forma nos fez ganhar mais conhecimento sobre a linguagem assembly (especialmente a do P4) e a sua interligação com os periféricos e o respectivo hardware, tornando-nos programadores mais experientes e abertos.