**Introdução à Arquitetura de Computadores**

**Relatório do projeto “Corrida de Bicicleta”**

Segunda-feira, 11:00h, Turno 5

Grupo 44

Eduardo Rodrigues Nº 78431

Alcino Albuquerque Nº 78753

Rui Matos Nº 79100

INTRODUÇÃO

Na cadeira de Introdução à Arquitetura de Computadores foi-nos proposta a realização de um projeto para instrumento de avaliação, na linguagem de programação assembly do P3, cujo objetivo é aplicar os conhecimentos obtidos nas aulas acerca desta linguagem.

Foi-nos pedido que o desenvolvimento de um jogo intitulado “Corridas de Bicicleta”. A evolução e teste do jogo serão realizados no simulador do P3 (p3sim) em que teremos todos os recursos da placa disponíveis, dos quais se destacam a janela de texto, os displays de 7 segmentos, os LEDs, o display LCD e os interruptores. No final, o jogo será apresentado no laboratório, a correr na placa hardware do P3.

O nosso projeto passou por três fases: primeiramente, um estudo e idealização da realização do projeto; depois a fase de escrita de código e teste deste; por fim, a indentação e comentação do projeto, assim como realização de teste finais na placa do P3, que podemos já referir que não revelaram qualquer problema, aquando do primeiro teste.

FUNCIONAMENTO DO JOGO - ENUNCIADO

O funcionamento do jogo é o seguinte:

* O jogo começa ao premirmos I1 e, de seguida, temos uma bicicleta a andar numa pista com limites laterais definidos que não podem ser ultrapassados pela mesma. Apenas podemos mover a bicicleta para a esquerda e para a direita, nessa mesma pista, premindo I0 e IB respetivamente.
* Ao decorrer do jogo, vão aparecendo obstáculos, aleatoriamente, estando no máximo 4 no ecrã. O objetivo do jogo é evitar esses mesmos obstáculos pois quantos mais a bicicleta ultrapassar, maior vai ser a pontuação do jogador.
* Existem 3 níveis no jogo; o jogador começa no nível 1 e quando ultrapassa 4 obstáculos o jogo avança ao próximo nível. Os níveis são caracterizados por dois fatores, a velocidade da bicicleta e o número de LEDs acesos. As velocidades de cada nível são respetivamente 2, 2.3 e 3.3 metros por segundo e os LEDs acesos são, respetivamente e a começar da esquerda, 4, 8, e 12. As pontuações são mostradas nos LCDs, e estas mostram a distância percorrida, a distância máxima percorrida em todos os jogos daquela sessão e o numero de obstáculos ultrapassados.
* Existe um modo especial chamado Turbo que é activado/desativado em qualquer momento do jogo quando premirmos I1. A velocidade da bicicleta muda para 5 metros por segundo e acendem todos os 16 LEDs.
* Podemos pausar o jogo premindo IA e retomar o jogo premindo outra vez IA.
* O jogo termina quando a bicicleta choca com um obstáculo e poderemos recomeçar o jogo premindo I1.

FASES DA REALIZAÇÃO DO PROJECTO

**Numa primeira fase,** foram elaborados fluxogramas para a realização do projecto com as principais funções a serem implementadas, e pensadas as formas como iriamos implementar algumas das funcionalidades. Aquando da apresentação do projecto, na aula de 11 de Novembro, não conseguimos por a bicicleta a andar, o que nos fez repensar que a forma como o trabalho estava a ser desenvolvido não era a mais correta.

É importante referir que há medida que fomos avançando na elaboração do projeto, este foi sendo testado no simulador para verificar se existiam erros e para identificar a sua origem, de forma a corrigi-los. Este processo repetiu-se até o resultado obtido na janela de texto corresponder com o esperado e, só assim, se continuava a codificação, o que nos permitiu obter um código correto e funcional durante todo o projeto.

No início da execução do projeto, começámos a implementar o código no simulador do P3 pelos aspetos mais básicos, ou seja, o desenho do cenário do jogo e o desenho da bicicleta (incluindo as variáveis e contantes associadas a estas rotinas). De seguida, criaram-se as rotinas que fazem o movimento da bicicleta e o movimento do obstáculo; no entanto, estas não funcionavam com as interrupções, pois ainda não existia qualquer tabela de interrupções desenvolvida.

**Numa segunda fase**, foram implementadas as interrupções necessárias para as rotinas que apresentam as mensagens iniciais e as mensagens finais, e que iniciam/terminam o jogo caso o botão I1 seja premido, e claro, o desenvolvimento do programa principal.

De seguida, desenvolveu-se o movimento da bicicleta, criou-se uma rotina que desenha a bicicleta na posição inicial, depois outra que apaga a posição atual da bicicleta, e chamam-se as rotinas que voltam a desenhar a bicicleta à esquerda ou à direita da posição anterior, verificando primeiro se a bicicleta está nos limites da pista.

Depois, a parte mais difícil, como será referido posteriormente, a implementação dos obstáculos. Criou-se uma rotina que desenha o obstáculo na posição inicial, mas de acordo com o valor que é gerado de forma aleatória pelo algoritmo dado, e colocado numa variável que indica a posição do obstáculo.

A movimentação dos obstáculos, parte difícil de desenvolver, baseia-se no temporizador e no valor que existe numa variável (contatemp), que nos indica se é necessário ou não gerar um novo obstáculo e ir movimentando os restantes.

Apesar da rotina de obstáculos já estar desenvolvida nesta altura, só mais tarde implementámos o funcionamento do temporizador, que contribui muito para o funcionamento “normal” do jogo. Há medida que estas rotinas iam sendo escritas, foi-se atualizando o programa principal, para, entre outros aspetos, que quando o obstáculo chocar contra a bicicleta, o jogo acabar, e desta forma, passar para as mensagens finais.

Posteriormente, implementaram-se os restantes detalhes do jogo que se apresentam na placa, nomeadamente, o funcionamento dos LEDs (que indicam o nível do jogo), do LCD (mostra a distância percorrida e a distância máxima) e dos displays (numero de obstáculos ultrapassados). Após a conclusão da programação dos aspetos do jogo base, dedicámo-nos à programação dos modos avançados do jogo.

Por último, foi implementado um algoritmo que gerava uma sequência pseudoaleatória, que tornava a posição inicial dos obstáculos aleatória, de acordo com o anexo fornecido no enunciado do projeto.

ASPECTOS IMPORTANTES DA IMPLEMENTAÇÃO

O nosso programa está organizado de acordo com a sua realização, começando pelas constantes da janela de texto e as variáveis utilizadas ao longo do projeto. De seguida, apresentam-se as interrupções, ou seja, atribuição de variáveis a essas interrupções e as ordens caso se carregue nos botões que lhes correspondem. Esta rotina contem a interrupção que corresponde à inicialização do jogo (I1), as que fazem a bicicleta mover-se, e que correspondem a I0 para a esquerda e IB para a direita, a interrupção que corresponde ao contador, e as interrupções que correspondem aos modos avançados, pausa (IA) e turbo (I2).

O programa continua com uma rotina que limpa a janela de texto, substituindo todas as posições por espaços, seguida das rotinas que apresentam as mensagens iniciais e finais do jogo, e a rotina que desenha o cenário do jogo, ou seja, os limites da pista. Seguidamente, apresenta-se o código associado à bicicleta e aos obstáculos, relativo ao seu desenho e à sua movimentação, como referido anteriormente.

Por sua vez, nas rotinas referentes aos 4 obstáculos, existe uma sequência de instruções que verifica se ocorreu colisão da bicicleta com o obstáculo (que na verdade, verifica se a posição da bicicleta coincide com alguma das posições do obstáculo), existem as rotinas que desenham, apagam e reescrevem o obstáculo nas várias posições iniciais possíveis, e que fazem a contagem de obstáculos caso estes cheguem à última linha, que calculam e geram uma nova posição para o obstáculo.

De seguida, apresenta-se a rotina que repõe os valores necessários a um novo jogo; esta rotina foi desenvolvida cedo e modificada ao longo do jogo para que fosse possível ir testando todas as implementações que íamos fazendo no código.

Uma rotina que atualiza as informações do LCD, composta por várias outras rotinas, como se verifica também ao longo do código, é que verifica se é necessário atualizar o valor da pontuação máxima atingida pelo utilizador, assim como atualiza os valores ao longo do jogo, e escreve no LCD a distância percorrida e a distância máxima. Para tal, foi necessário criar uma rotina que converta o número de hexadecimal para decimal.

Esta conversão baseia-se na divisão inteira do número por 10 e consequente aproveitamento do resto obtido. Foi necessário, também, converter algarismo a algarismo o número obtido para ASCII, o que se faz pela operação *ADD R1(primeiro algarismo), ‘0’.* Existe uma rotina que escreve o número de obstáculos ultrapassados pelo utilizador no display de 7 segmentos, realizando as mesmas operações de conversão da anterior.

Existe, ainda, um conjunto de instruções que criam a sequência aleatória de coordenadas para escrever o obstáculo, baseadas nas indicações dadas pelo corpo docente. Esta rotina contém também código que permite gerar uma coordenada dentro dos limites, procedemos dividindo o número obtido por 20 e por conseguinte somando-lhe 31 para que ele fica-se dentro dos limites pretendidos para a posição do obstáculo.

Já na última parte do código, existe uma rotina que altera os LEDs e as velocidades do jogo, de acordo com o Nível, dependo do número de obstáculos ultrapassados.

De seguida apresentam-se as sequências de instruções que traduzem o modo pausa e o modo turbo.

**Pausa -** Esta rotina, correspondente a um modo avançado, permite não só parar o jogo como escreve também no LCD a mensagem ‘-PAUSA-‘ mantendo para o utilizador a distância máxima percorrida até ao momento, de forma a proporcionar uma melhor estética. De seguida, ou seja, após o botão IA, o jogo decorre sem qualquer problema deste o ponto em que parou. Esta rotina funciona desligando o temporizador e de seguida voltando-o a ativar com os valores anteriores de velocidade, caso o botão seja novamente pressionado.

**Turbo -** Esta rotina, correspondente a um modo avançado, permite colocar o jogo com uma velocidade de actualização de obstáculos superior à normal. Esta rotina funciona movendo um valor para uma variável TURBOONOFF que serve para indicar a rotina que altera os leds e as velocidades que não pode fazer qualquer alteração ao níveis, pois estes são definidos e alterados pelo Turbo. Por sua vez, caso o turbo seja desligado, a variável é novamente alterada sendo os valores repostos pela responsável pela VEL e Led.

No final, encontra-se uma grande rotina que corresponde ao funcionamento geral do jogo em si, utilizando e chamando as rotinas anteriormente referidas, com as suas variáveis e constantes, que mostra as mensagens iniciais do jogo, e o inicia se o botão I1 for acionado, desenhando o cenário, e fazendo as movimentações dos obstáculos e da bicicleta (consoante os botões IB e I0, finalizando o jogo caso haja colisão, o que é indicado por uma variável o COLISAOCHECK que é alterada aquando da escrita dos obstáculos e que caso eles choquem contra a bicicleta é posteriormente alterada e faz o programa principal indicar que existe fim do jogo. Por fim, inicia-se um novo jogo, com os valores necessários repostos.

O programa principal possui, apesar de não dever, pelo menos do ponto de vista estético pois não altera em nada o óptimo funcionamento do programa, o código que faz os obstáculos moverem-se continuamente. Tentámos ainda, mudar esta situação, no entanto, não tivemos tempo pois demos mais importância a implementar todas as funcionalidades avançadas.

**Numa terceira fase,** passou-se a uma fase de revisão, indentação e comentação do projeto, para que a compreensão deste fique acessível e explícita, tendo sido, para isso, necessária a mudança do nome de algumas variáveis, contantes e rotinas.

CONCLUSÃO

O projeto finalizado realizado, contém todas as funcionalidades previstas em conformidade com o protocolo do projeto, incluindo as funcionalidades avançadas, nomeadamente o modo pausa e o modo turbo, não tendo por isso divergido do enunciado base, uma vez que não tem funcionalidades adicionais implementadas, nem funcionalidade que não estejam implementadas.

Este trabalho permitiu-nos obter conhecimento acerca de técnicas de programação em geral e particularmente, da programação em Assembly do P3, bem como a desenvolver métodos lógicos para resolução problemas, o que contribui não só para o nosso desenvolvimento na presente cadeira mas também para o nosso desenvolvimento como futuros Engenheiros.