Instituto Superior Técnico
INTRODUÇÃO À ARQUITETURA DE COMPUTADORES
Turno de Segunda-feira, 11:00 – 12:30
Grupo 13

Relatório do Pássaro Bamboleante

Estrutura geral do jogo

O objetivo do jogo é movimentar um pássaro na vertical e impedir que o mesmo colida contra os obstáculos que se vão movimentando da direita para a esquerda. O jogo decorre numa janela de texto e o pássaro está sempre localizado na parte esquerda do ecrã.

O programa divide-se em várias rotinas: as que são executadas no ciclo principal e as que servem como apoiantes dessas rotinas. Das rotinas implementadas destacamos as que implementam as seguintes funcionalidades:

- Gravidade no jogo (DescerPassaro);
- Verificação das colisões (CoordTemObs);
- Movimentação dos obstáculos (DeslocaObs, DesObs, DesX, ApagaUltCol);
- Escrita do número aleatório do espaço dos obstáculos para a memória (InsereVetObs, MemObsFrente, RandomN).

Variáveis principais do jogo

R1 = Contador da dificuldade.

R3 = Ni (Número aleatório atual).

R4 = Posição do cursor.

R5 = (y0) Posição da última vez que o pássaro subiu.

R6 = (t) Tempo desde a última vez que o pássaro subiu (em ciclos do temporizador).

R7 = Interrupções ativadas (T, I0, I1, I2,…) (primeiro bit para T e por aí em diante).

Distancia = Distancia percorrida.

Nível = Nível invertido (1-16) (16-N).

PosPassaro = Posição do pássaro em vírgula fixa (0C,00).

PosAPassaro = Posição anterior do pássaro (em vírgula fixa).

LinhasASubir = Linhas que faltam subir.

EspObstaculos = (8043) Número máximo de obstáculos.

FaseObstaculos = Fase, de 0 a 5, da movimentação dos obstáculos.

NumObstaculos = Número de obstáculos existentes na janela.

EstadoLEDs = Conteúdo dos LEDs.

Gravidade no jogo

A cada ciclo do temporizador, enquanto o I0 não é pressionado, é calculada uma nova posição para o pássaro utilizando a fórmula do movimento uniformemente acelerado com uma aceleração predefinida. Para tal usamos um contador em R6 que nos diz o tempo decorrido desde a última subida. A cada ciclo é verificado se a personagem do jogo atinge o limite inferior e se o mesmo o atingir, o jogo termina.

Verificação das colisões

Esta rotina baseia-se no facto de termos chegado à conclusão que, devido à sua posição, o pássaro apenas pode colidir com coluna 9 (a décima coluna a contar da direita).

Concluímos que para qualquer coluna n,

83449 Duarte Galvão;
83539 Pedro Caldeira;
85080 João Pina.

$x_{Col_n} \in [LE - 1 - n \times (IO + 1) - IO,\ LE - 1 - n \times (IO + 1)]$, em que x_{Col_n} é a coordenada x da coluna n, LE é a largura do ecrã (79) e IO é o intervalo entre os obstáculos (5). Para obter o n a partir da coordenada, devemos ter em conta que o n deve cumprir a seguinte condição:

$$\frac{LE - 1 - x_{Col_n}}{IO + 1} \leq n \leq \frac{LE - 1 - x_{Col_n} - IO}{IO + 1}$$

Desta fórmula, retiramos que $8{,}8333 \leq n \leq 9{,}333$, e como $n \in \mathbb{N} \Rightarrow n = 9$.

Uma vez tendo verificado a coluna, é necessário verificar se a linha está no espaço da coluna dos obstáculos ou não. A verificação é feita de acordo com a seguinte condição:

$$y_{primeiro\ espaço} - y_{atual} \leq 0 \leq y_{primeiro\ espaço} - y_{atual} + Esp\_Obs$$

Movimentação dos obstáculos/ Escrita do número aleatório do espaço dos obstáculos para a memória

Este conjunto de rotinas tem como base o contador da dificuldade, gravado em R1, que varia entre 0 e Nível-1. Inicialmente, o ecrã está sem obstáculos e a cada ciclo do contador são executadas as rotinas referentes à criação e consequente movimentação dos obstáculos.

O deslocamento dos obstáculos deriva de um sistema de 6 fases distintas (de 0 a 5). A primeira fase corresponde à criação de um novo obstáculo. As seguintes refletem-se na movimentação do mesmo. Depois do primeiro obstáculo se mexer 5 posições para a esquerda, a fase é reiniciada e é criado um novo obstáculo. Este processo repete-se até ao fim do jogo.

Esta movimentação é feita através do deslocamento para a direita dos elementos do vetor de memória que guarda os números aleatórios correspondentes ao espaço de cada coluna e a inserção de um novo número aleatório no início do vetor

É de notar que o nível de dificuldade do jogo está invertida. Isto é, na memória, o nível está guardado como 16 – Nível.

Conclusões

Em suma, neste projeto foi necessária a criação de várias rotinas auxiliares de modo a “servir” a rotina principal.

As variáveis foram inicializadas no início do programa para permitir o reiniciar do programa no simulador.

De modo a melhorar a experiência de jogo, definimos o número de espaços de abertura das colunas como 6.

A grande dificuldade sentida ao longo do projeto foi o facto de termos ficado sem registos. Em muitas ocasiões, precisamos de guardar os dados na pilha e em memória.

No entanto, achamos que o projeto tem um balanço satisfatório sendo que conseguimos realizar uma versão final funcional tanto na placa como no simulador e que cumpre todos os requisitos descritos no enunciado.

83449 Duarte Galvão;
83539 Pedro Caldeira;
85080 João Pina.