

Arquitetura de Computadores

IST - Taguspark

Relatório do Projeto – Chuva de Meteoros

Grupo 8

Afonso Pires – 102803

António Silva – 102879

Diogo Marques – 102760

Introdução:

Este trabalho foi desenvolvido para a disciplina de Arquitetura de Computadores, com o objetivo de familiarizarmo-nos com a linguagem de programação *Assembly*. Para tal, foi utilizado um Processador Especial Para Ensino (PEPE16), uma RAM, e periféricos (*Pixel Screen, display, teclado*).

O projeto consiste numa “Chuva de meteoros”, em que o jogador controla uma nave (*Rover*) localizada num planeta X, tendo que proteger o planeta desses meteoros que vão aparecendo no topo do ecrã. Estes meteoros apresentam duas categorias distintas, os meteoros maus (vermelhos) e os meteoros bons (verdes). O jogador consegue eliminar os meteoros, disparando um míssil contra os mesmos.

Numa fase inicial, o *Rover* apresenta a energia de 100% e conforme o tempo vai passando (5 em 5 segundos) a energia vai diminuindo 5%, porém cada vez que disparado um míssil a energia do *Rover* diminui 5%, que podem igualmente ser repostos, caso esse míssil colida com meteoro mau (5%). Caso o meteoro mau colida com o *Rover* o jogo acaba, porém se colidir com um meteoro bom incrementa 10% energia.

O jogo termina quando a energia do Rover estiver a 0.

Concessão e Implementação:

Estrutura Geral:

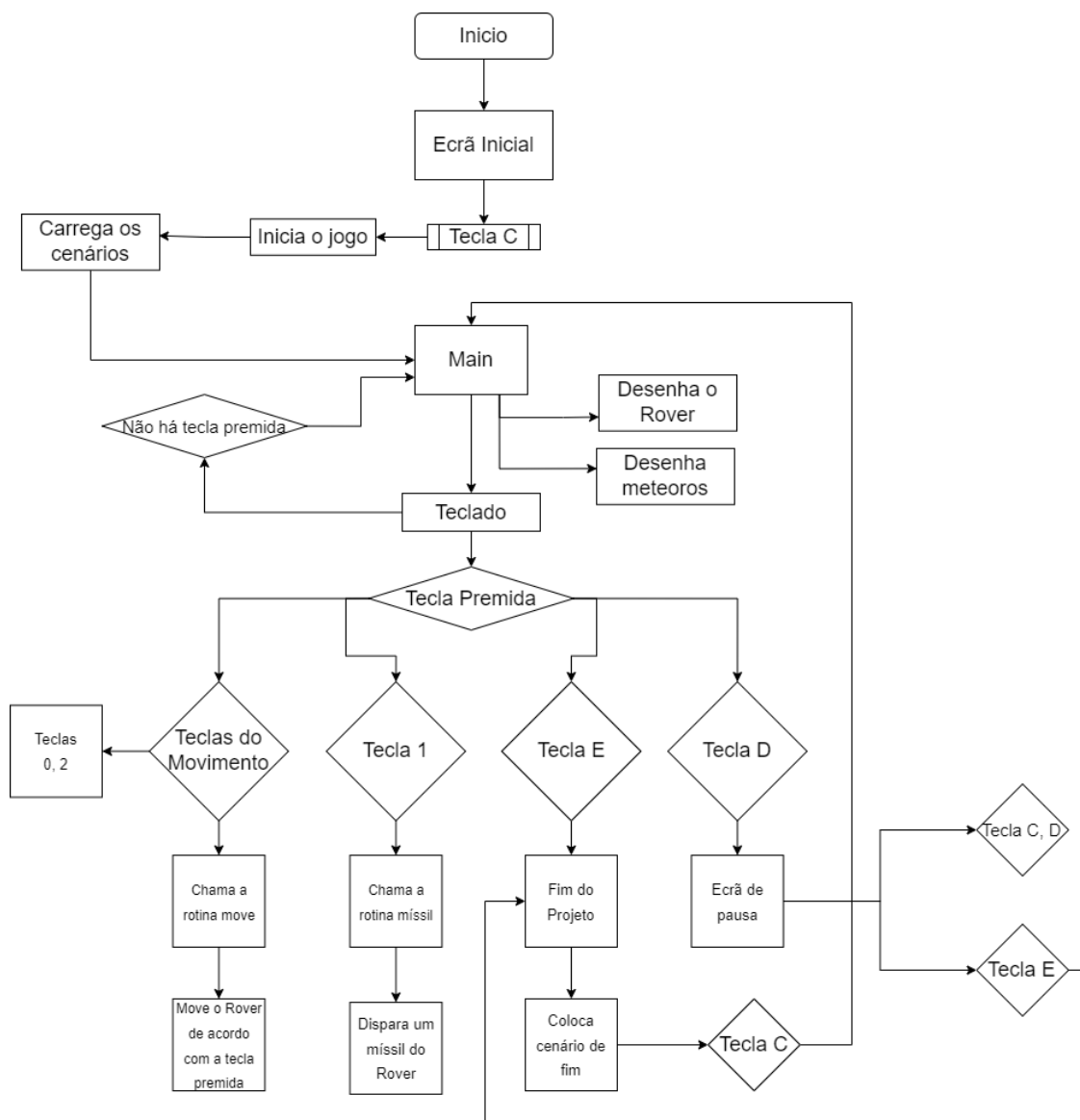


Figura 1 - Fluxograma para compreensão do funcionamento do projeto

Rotinas:

Explicamos de seguida, através de pseudo-linguagem, como funcionam as principais funções que criámos:

→ Teclado – Retorna a tecla premida pelo utilizador no periférico do teclado (POUT-2)

Início

Deteta se há uma tecla premida na linha 4

Se houver uma tecla premida

Identifica a tecla

Guarda num registo a tecla

Se não houver tecla premida

Atualiza a variável da linha

Verifica novamente a linha

Atualiza a tabela TECLA_PREMIDA para 1

Caso contrário, mantém a TECLA_PREMIDA a 0

→ Desenha_Rover - Desenha uma nave (*Rover*) centrado no ecrã com cor e forma definida numa tabela.

Início

Define a cor da nave

Seleciona o pixel a desenhar no ecrã

Chama a função ESCREVE_PIXEL

Preenche os pixels da linha onde se encontra (consoante foi definido)

Se chegar à última coluna

Atualiza a linha, para a próxima, e volta à primeira coluna

Fim

→ Apaga_Boneco – Apaga a nave (*Rover*) do ecrã.

Início

Define a cor da nave = 0

Seleciona o pixel a desenhar no ecrã

Chama a função ESCREVE_PIXEL

Preenche os pixels da linha onde se encontra a zero

Se chegar à última coluna

Atualiza a linha, para a próxima, e volta à primeira coluna

Fim

→ Move_Nave – Move a nave no ecrã (*pixel screen*)

Início

Verifica se o jogo já começou

Caso contrário, sai da rotina

Verifica para que direção tem que mover

Se for para a direita (segundo o eixo do 'X')

Verifica o limite direito

Caso esteja na última coluna (63)

Não move

Caso contrário

Incrementa 1 à variável da coluna

Se for para a esquerda (segundo o eixo do 'Y')

Verifica o limite esquerdo

Caso esteja na primeira coluna (0)

Não move

Caso contrário

Decrementa 1 à variável da coluna

Fim

→ Cria_missil – Cria o míssil a ser disparado

Início

Verifica se já existe míssil

Caso não exista ainda

Dá-se as coordenadas do missil e a cor

Chama-se a função ESCRIVE_PIXEL

Caso já exista

Sai da rotina

Dá-se o som de disparo como argumento

Chama-se a função SOM

Fim

→ ROTINA_MISSIL – Move o míssil no ecrã

Início

Verifica-se se já existe um míssil

Se já existir

Sai da rotina

Verifica-se se o já começou

Se ainda não

Sai da rotina

Chama-se a função APAGA_MISSIL

Verifica-se se já foram atingidos o máximo de movimentos do míssil

Se já se tiver atingido

Sai da rotina

Se ainda não tiver

Atualiza a variável da linha do meteoro

Chama a função ESCRIVE_PIXEL

Fim

→ Colisão – Verifica se o míssil embateu no meteoro, desenhando uma explosão

Início

Verifica-se se o míssil e o meteoro estão na mesma linha

Se estiverem na mesma linha

Verifica-se a coluna onde está o míssil é maior ou igual à do meteoro

Se for maior ou igual

Soma-se se 5 à coluna do meteoro

Verifica-se se a coluna do míssil é menor ou igual à do meteoro

Se for menor ou igual

Verifica-se qual coluna tem o míssil

Chama-se a função APAGA_METEORO

Chama-se a função DESENHA_COLISAO

Chama-se uma função de atraso

Chama-se a APAGA_COLISAO

Se não estiver na mesma linha

Sai da rotina

Fim

Conclusões:

Acreditamos ter cumprido o desafio que foi este projeto, tendo exercitado a utilização da linguagem *Assembly*, aprofundando assim os nossos conhecimentos da mesma e também da cadeira.

O movimento da nave é controlado pelo jogador através do teclado, que programámos conforme recomendado. As teclas 0, 2, dizem respeito ao movimento da nave, a tecla 1 dispara um míssil, a tecla C começa o jogo, a tecla D coloca o jogo em pausa e num segundo clique tira da pausa, ou até mesmo pode-se terminar ou reiniciar o jogo, clicando nas respetivas teclas, por último a tecla E termina o jogo, tendo a opção de reiniciar o mesmo.

O display controla energia Rover que começa a 100 e vai diminuindo gradualmente com o passar do tempo ou com o embate de um meteoro mau no fim do ecrã. A destruição de um meteoro recupera essa energia. Caso a energia chegue a zero, o jogo acaba tal como, o programa.

Através da compreensão das funcionalidades do *Media Center* adicionámos vários cenários e sons ao projeto. Para além disso aprendemos ainda a desenhar e apagar nesse mesmo ecrã, através de tabelas com a definição de cada *pixel*. Para além disso adicionamos cenários especiais de espera para transmitir uma sensação de maior autenticidade, para efetivamente parecer-se com um jogo.

Na secção da criatividade decidimos acrescentar 3 níveis distintos de jogo, de modo a oferecer uma sensação de progressão ao jogador e assim despertar mais interesse no mesmo. Para mudar de nível implementámos um sistema em que basta destruir três meteoros, para progredir no jogo. Estes níveis são constituídos por cenários de fundo diferentes, acompanhados de 3 modelos de *Rover* também diferentes.

Assumimos também que o facto de não termos utilizado ecrãs diferentes para os diferentes meteoros, quando os desenhos se sobrepõem apagam-se mutuamente, sendo que voltam a ser desenhados quando se movimentam para a linha seguinte, sempre sobrepondo-se um ao outro. Importante ainda referir, que quando ocorre a colisão do míssil com o meteoro, detetamos que esse meteoro apesar de ser destruído, é novamente desenhado, algo que acontece apenas uma vez por nível. Não detetámos o que poderia provocar este erro.