

INTRODUÇÃO À ARQUITETURA DE COMPUTADORES

LEIC
IST - TAGUSPARK

RELATÓRIO DO PROJETO NINJAS

Grupo 19:

- Ana Sofia Simões, 90703
- Bernardo Faria, 87636
- Catarina Gonçalves, 90709

1.Introdução

Este trabalho foi realizado no âmbito da cadeira de **Introdução à Arquitetura de Computadores** (IAC), tendo como objetivo praticar e exercitar a matéria lecionada nas aulas teóricas da mesma, como, por exemplo, programar em **linguagem Assembly**, trabalhar tanto com rotinas como interrupções, trabalhar num processador (PEPE), entre outros aspetos.

O projeto consiste na conceção do jogo Ninjas, no qual, de forma sucinta, o jogador tem como objetivo mover os ninjas de forma a capturar os presentes e escapar às estrelas de lâminas (armas) pois se algum dos ninjas for atingido por uma destas morre. Desta forma, no que toca à pontuação, se algum dos ninjas colidir com uma estrela de lâminas o utilizador perde 3 pontos, e caso algum dos ninjas capture um presente o utilizador ganha 3 pontos.

O jogo concebido tem três estados: o estado inicial no qual o jogo tem início, o estado de pausa e o estado final no qual o jogo termina.

Na secção 2, descrever-se-á as estratégias de implementação adotadas, assim como uma revisão detalhada das rotinas implementadas e da comunicação entre processos.

Na secção 3, a conclusão conterá uma reflexão crítica sobre o trabalho realizado ao longo do semestre, referindo os aspetos produzidos no projeto e as dificuldades encontradas ao longo do seu desenvolvimento.

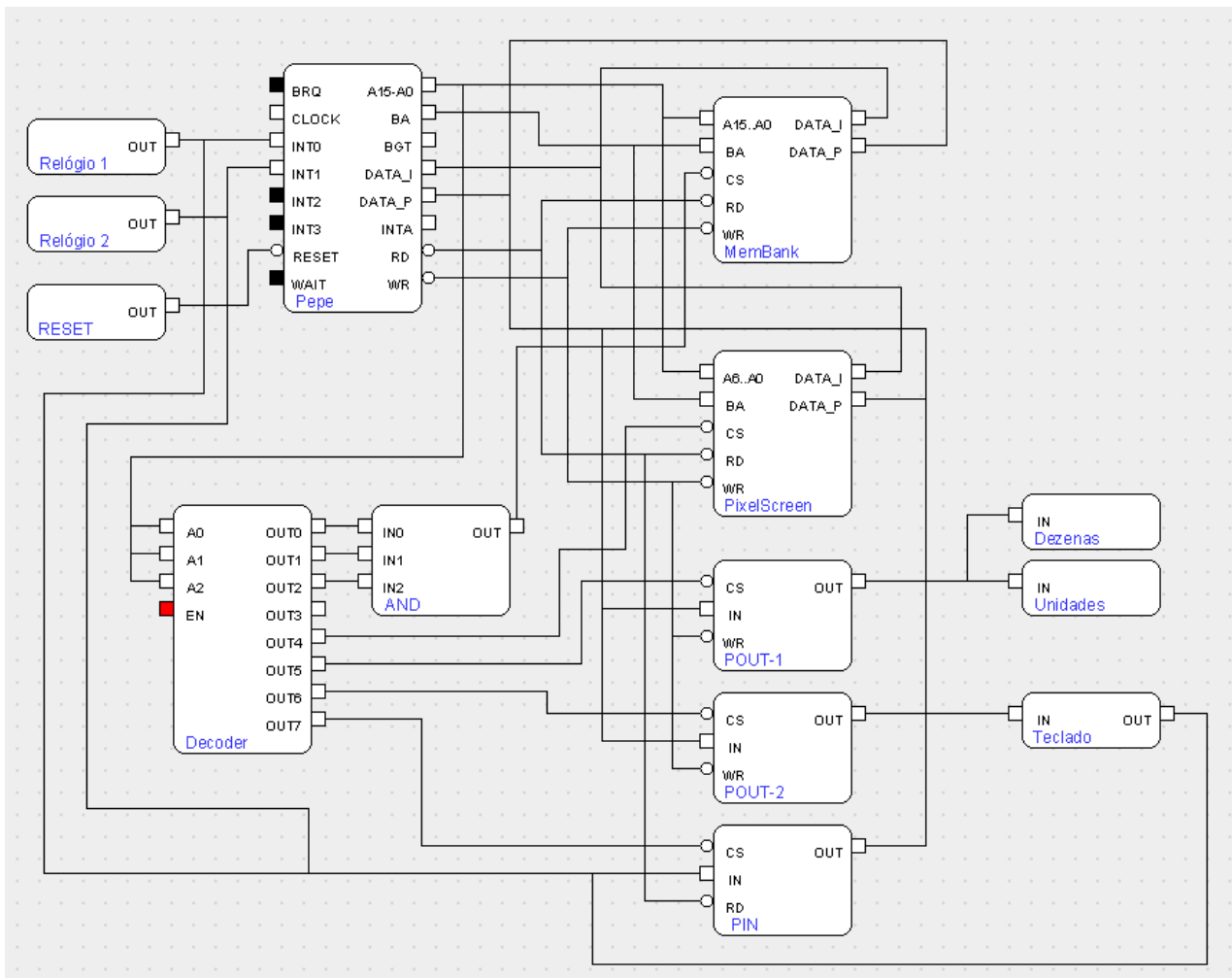
Na 4ª e última secção, o código *Assembly* do jogo estará explicitado.

2. Conceção e Implementação

2.1. Estrutura Geral

2.1.1 Hardware

O circuito utilizado para o jogo foi o circuito fornecido juntamente com o enunciado do projeto, não se efetuando quaisquer alterações no hardware. O seguinte esquema de blocos mostra o circuito em causa.



Legenda:

- **PEPE (16 bits)** – Componente responsável pela lógica, aritmética e ainda pela interpretação do código e controlo de todos os componentes
- **Relógios de tempo real:**
 - Relógio 1 – utilizado como base para a temporização do movimento das armas.

- Relógio 2 - utilizado como base para a temporização da queda dos ninjas.
- **Matriz de pixels (PixelScreen)** – (ecrã de 32 x 32 pixels acedido como se fosse uma memória de 128 bytes) Tem como finalidade ser o ecrã do jogo, ou seja, onde visualizamos o movimento dos ninjas, dos presentes e das armas. Os pixels a vermelho correspondem a um bit a 1 e os pixels a cinzento correspondem a um bit a 0.
- **Dois displays de 7 segmentos** – mostram a pontuação obtida durante o jogo. Existem dois displays, um que marca as unidades e outro que marca as dezenas. A pontuação é sempre iniciada a 00, é incrementada em 3 unidades sempre que um ninja colide com um presente, e retiram-se 3 unidades sempre que um ninja colide com uma estrela de lâminas.
- **Teclado** – serve de interação entre o utilizador e o jogo permitindo, assim, que o utilizador consiga controlar o movimento dos ninjas.
- **MemBank** - é o bloco correspondente à memória onde são guardadas as variáveis necessárias ao funcionamento do jogo (por exemplo: a pontuação, tecla premida, etc...)
- **POUT** – têm como função principal a escrita da informação.

O mapa de endereçamento a seguir representado demonstra quais os endereços em que os dispositivos podem ser acedidos pelo PEPE.

Dispositivo	Endereços
RAM (MemBank)	0000H a 5FFFH
PixelScreen	8000H a 807FH
POUT – 1 (Periférico de saída de 8 bits)	0A000H
POUT – 2 (Periférico de saída de 8 bits)	0C000H
PIN (Periférico de entrada de 8 bits)	0E000H

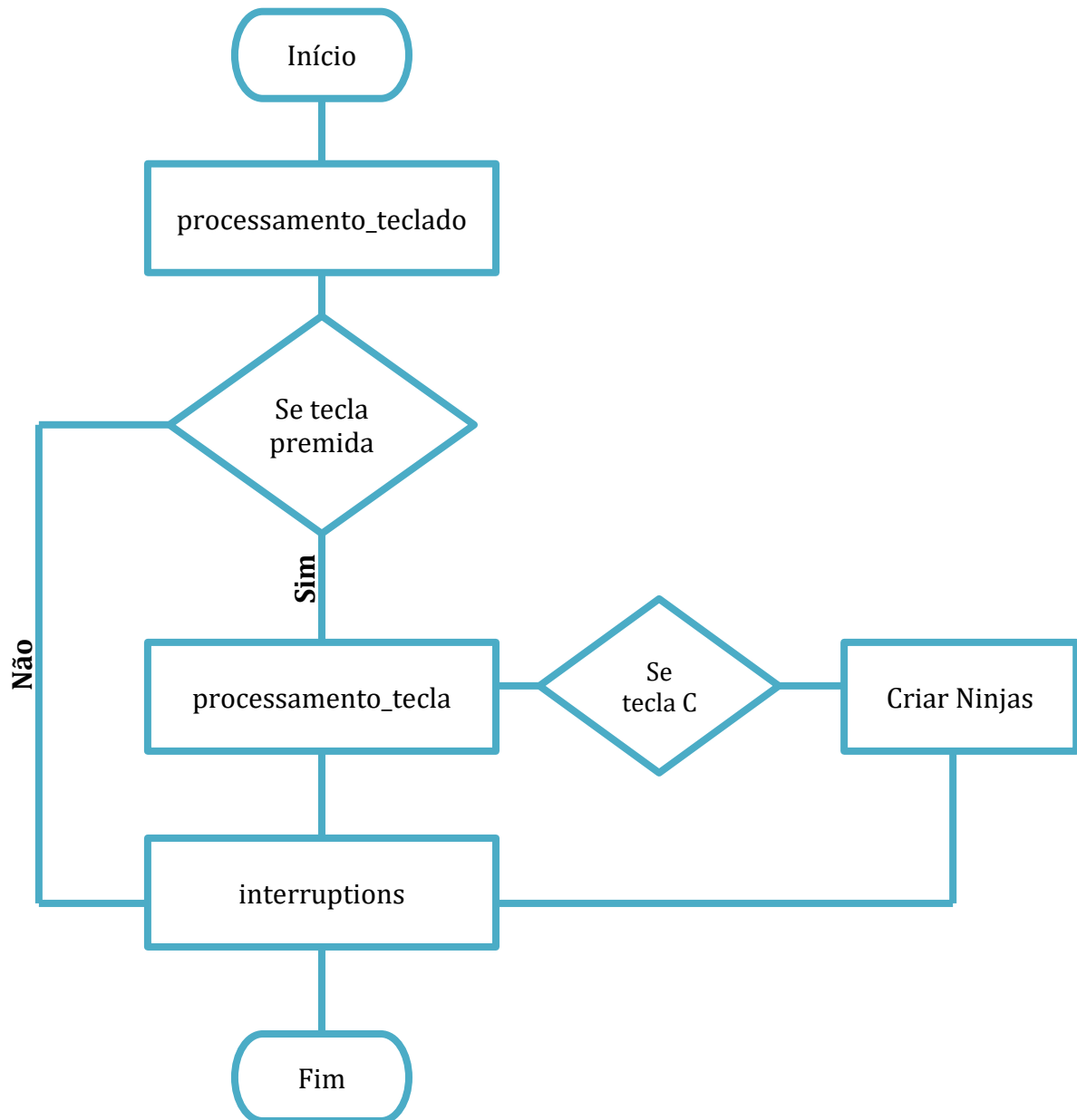
2.1.2 Software

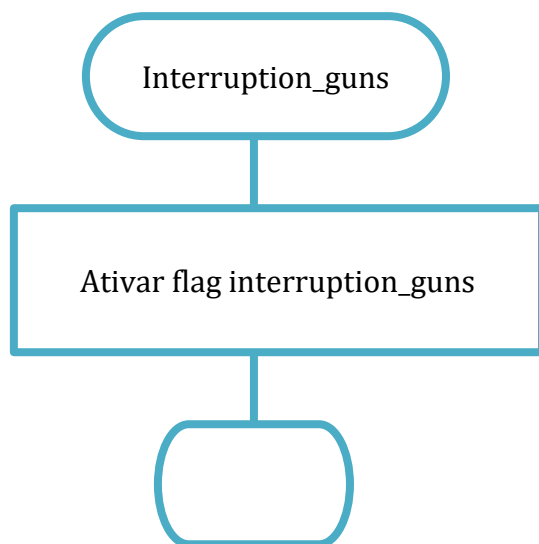
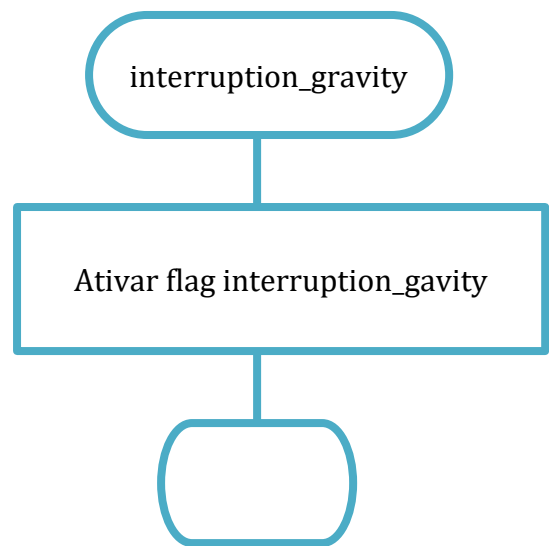
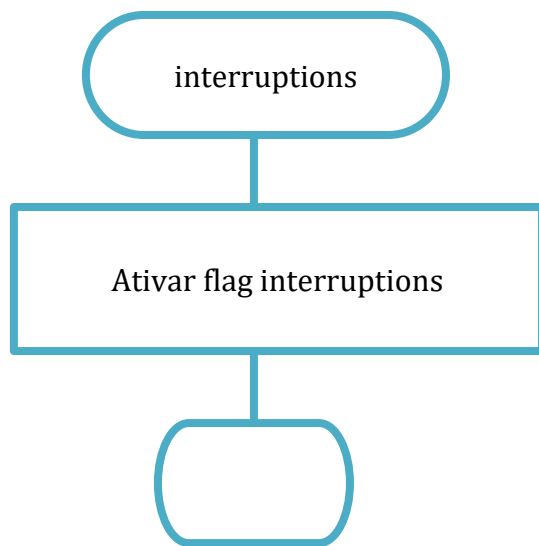
De modo a implementar um código eficaz recorreremos a chamadas às rotinas necessárias. Estas são:

- **keyboardcheck**: rotina que tem como função verificar qual a tecla premida;
- **processamento_tecla**: executa a função da tecla premida;
- **subir_ninja1**: rotina que tem como função apagar o ninja 1, atualizar as posições e tornar a pintar o ninja 1 nas novas posições;
- **descer_ninja1**: rotina que tem como função apagar o ninja 1, atualizar as posições e tornar a pintar o ninja 1 nas novas posições;
- **subir_ninja2**: rotina que tem como função apagar o ninja 2, atualizar as posições e tornar a pintar o ninja 2 nas novas posições;
- **descer_ninja2**: rotina que tem como função apagar o ninja 2, atualizar as posições e tornar a pintar o ninja 2 nas novas posições;
- **subir_ninja3**: rotina que tem como função apagar o ninja 3, atualizar as posições e tornar a pintar o ninja 3 nas novas posições.
- **descer_ninja3**: rotina que tem como função apagar o ninja 3, atualizar as posições e tornar a pintar o ninja 3 nas novas posições;
- **subir_ninja4**: rotina que tem como função apagar o ninja 4, atualizar as posições e tornar a pintar o ninja 4 nas novas posições;
- **descer_ninja4**: rotina que tem como função apagar o ninja 4, atualizar as posições e tornar a pintar o ninja 4 nas novas posições;
- **paintxy**: rotina que tem como função pintar ou limpar no pixelscreen nas coordenadas de um ponto (x,y);
- **paint_object**: rotina que tem como função pintar os diferentes objetos;
- **move_object**: rotina que tem como função mover os objetos;
- **gerador**: rotina que tem como função gerar os ninjas e verificar o estado de jogo;

- **clean_pixelscreen**: rotina que tem como função limpar o pixelscreen;
- **create_shuriken**: rotina que tem como função gerar a estrela de lâminas (shuriken);
- **create_present**: rotina que tem como função gerar o presente;
- **gravity**: rotina que tem como função mover os ninjas para baixo;
- **move_guns**: rotina que tem como função ativar as flags e mover as armas para a esquerda;
- **interruptions**: rotina que tem como função ativar as flags;
- **interruption_gravity**: rotina que tem como função ativar a flag da gravidade;
- **interruption_guns**: rotina que tem como função ativar a flag das armas;
- **comecar_jogo**: rotina que tem como função alterar o estado do jogo para começar o jogo;
- **suspender_jogo**: rotina que tem como função alterar o estado do jogo para pausar o jogo;
- **paint_screen_begining**: rotina que tem como função pintar o pixelscreen de acordo com as etiquetas de inicialização.

2.2 Fluxogramas





3. Conclusões

Com o desenvolvimento desde projeto, o grupo conseguiu expandir os seus conhecimentos tanto na parte relacionada com o software (linguagem Assembly) como na parte relacionada com o hardware.

Durante a conceção do projeto deparámo-nos com certas dificuldades, tais como: conseguir escrever um código correto e eficaz para as colisões entre os ninjas e os diferentes objetos, colocar o jogo em estado de pausa (ou seja, colocar a tecla D funcional) e mostrar a pontuação obtida durante o jogo em ambos os displays.

Sendo o objetivo principal do projeto desenvolver o jogo Ninjas em Assembly, podemos concluir, em termos de funcionamento e desempenho, que o nosso projeto está funcional, apesar de não executar todas as funcionalidades pedidas.