TÉCNICO LISBOA

INTRODUÇÃO À ARQUITETURA DE COMPUTADORES

LEIC

IST-TAGUSPARK

RELATÓRIO DO PROJETO

GRUPO 26

André Patrício 87631

Carolina Carreira 87641

João Antunes 87668

Índice

1. Introdução

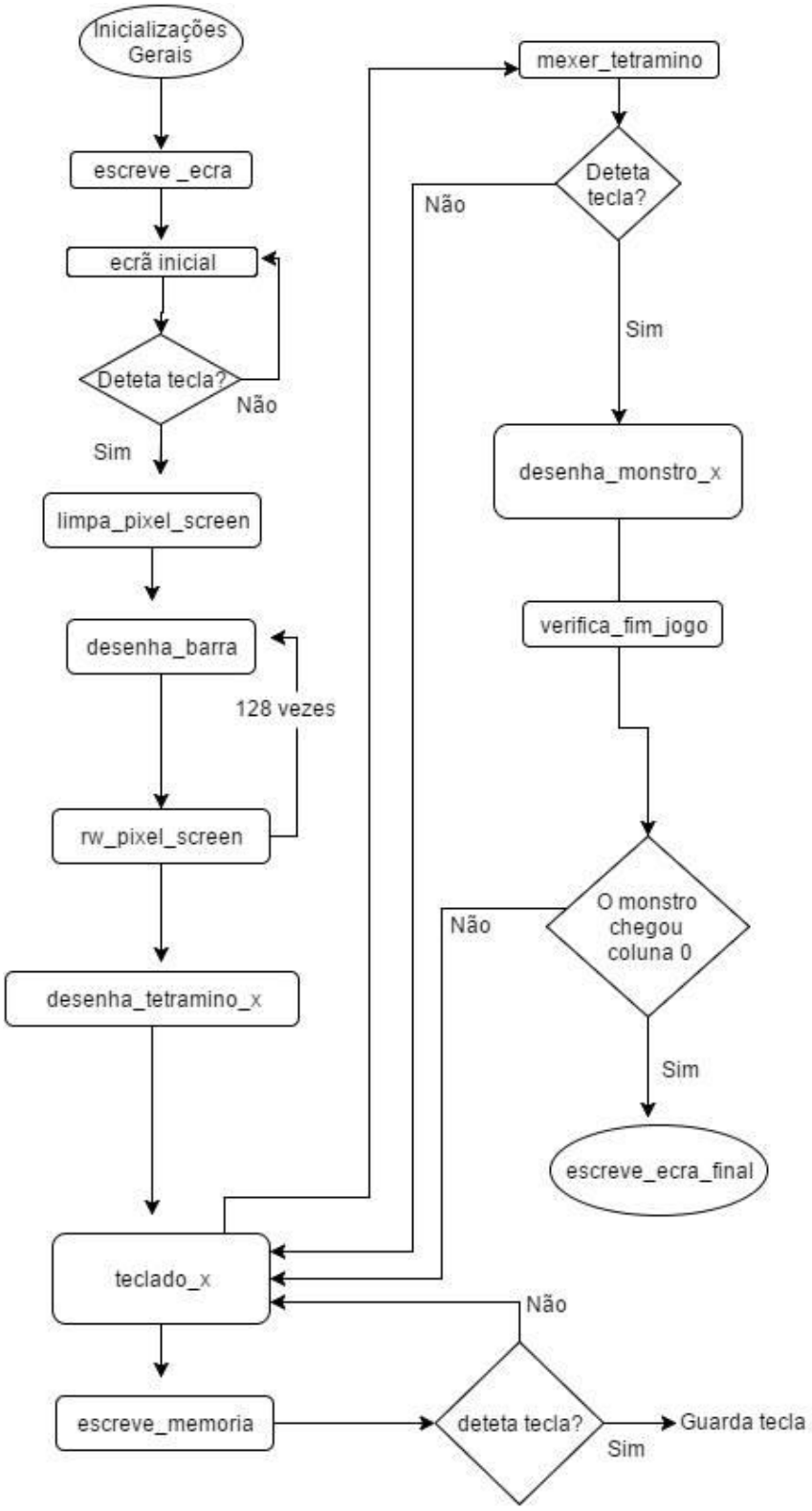
Este projeto está inserido no âmbito da *cadeira* de Introdução à Arquitetura de Computadores, tendo como objetivos primários a familiarização dos alunos com a linguagem de programação assembly, periféricos e interrupções. Este fim é alcançado através da elaboração do jogo 'Tetris Invaders' em assembly no simulador PEPE.

O jogo 'Tetris Invaders' é a fusão de dois outros jogos: 'Tetris' e 'Space Invaders'. Especificamente une a função de encaixe e empilhamentos de tetraminós do clássico jogo 'Tetris' com um monstro que aparece aleatoriamente e que deve determina o fim do jogo ao chegar ao lado esquerdo do ecrã. O jogador deve então eliminar este deixando cair uma peça em cima dele.

Os objetivos deste projeto consistem em elaborar este jogo dentro das especificações acima referidas aliado com uma boa estrutura de código.

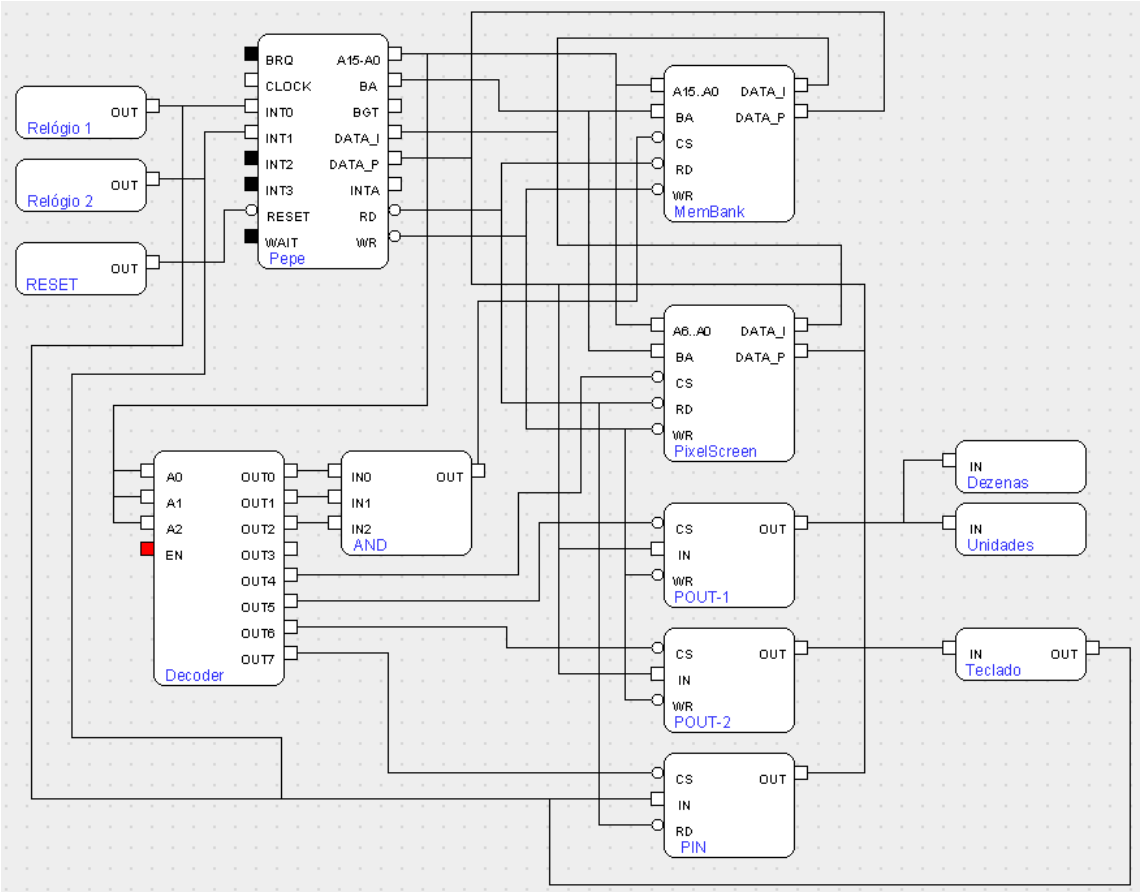
2. Conceção e Implementação

2.1 Descrição Geral



2.1.1 Circuito utilizado

Foi utilizado o seguinte circuito com o simulador PEPE.



2.1.2 Mapa de endereçamento

Respetivo ao simulador PEPE.

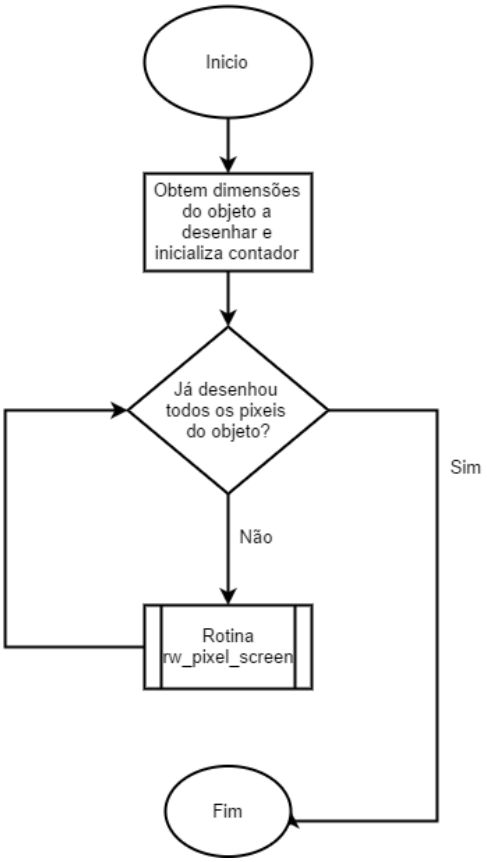
Dispositivo	Endereços
RAM (MemBank)	0000H a 5FFFH
PixelScreen	8000H a 807FH
POUT-1 (displays)	0A00H
POUT-2 (teclado saída)	0C00H
PIN (teclado entrada)	0E00H

2.2 Descrição principais rotinas

Rotina *desenha_tetramino_x*

Desenha um qualquer tetraminó/objeto.

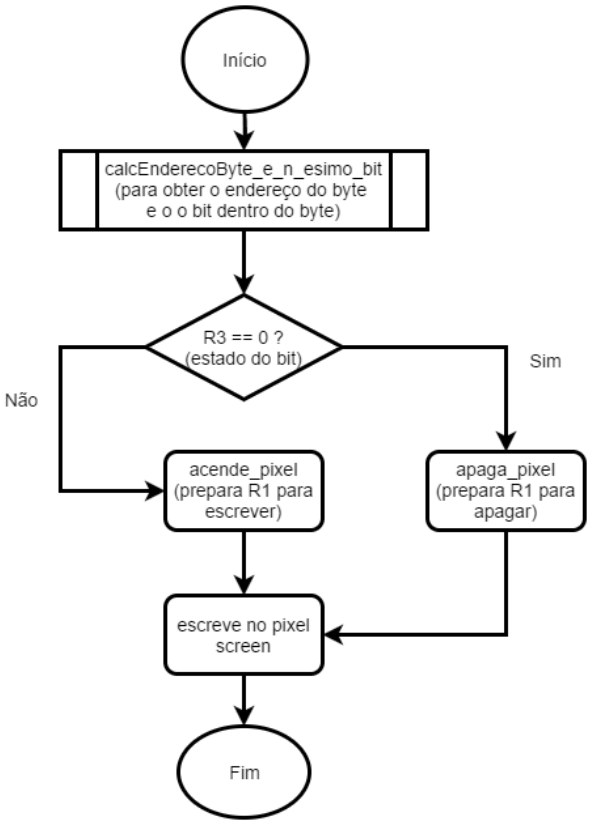
Input	**R0: endereço do objeto a desenhar**
Output	ND (chama a rotina rw_pixel_screen tantas vezes quantos os pixéis do objeto)



Rotina *rw_pixel_screen*

Escreve/apaga um pixel no pixel screen.

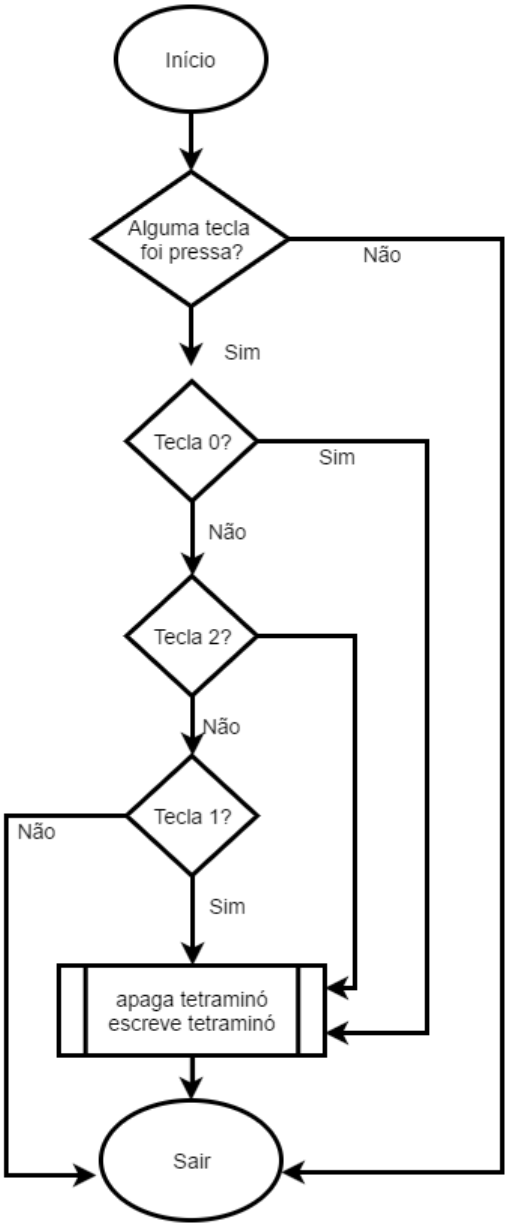
Input	**R1: linha do pixel (0…31)** **R2: coluna do pixel (0…31)** **R3: estado do bit (apagar: 0, ou escrever:1)**
Output	ND (chama a rotina rw_pixel_screen tantas vezes quantos os pixéis do objeto)



Rotina *mexer_tetraminó*

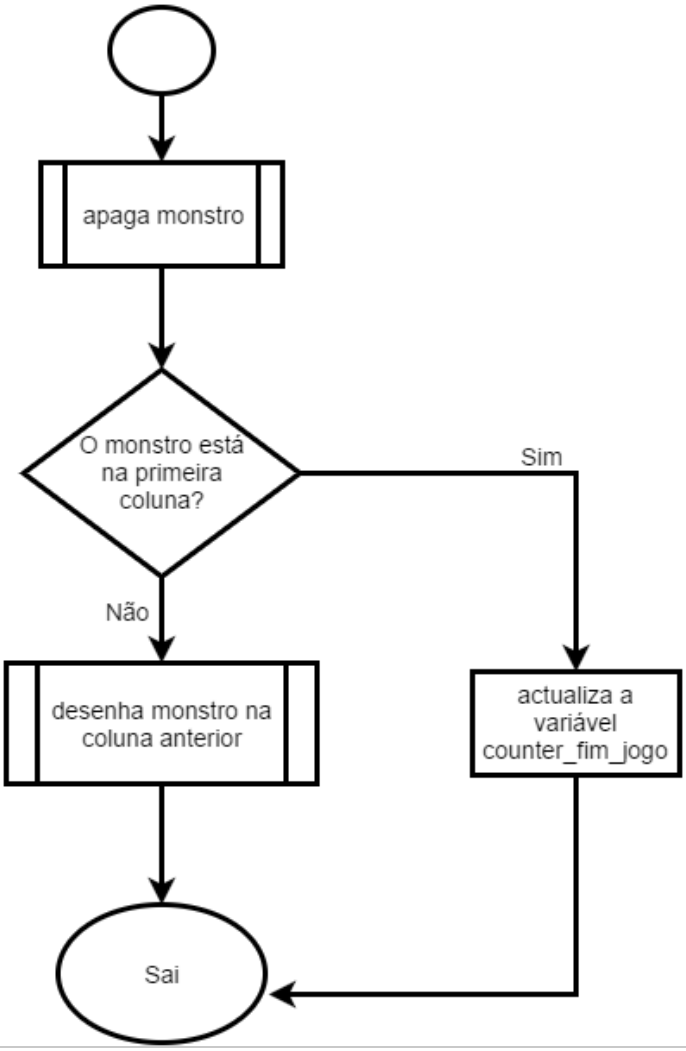
Descola tetraminó horizontalmente e roda-o.

Input	ND (tecla_pressa da rotina teclado)
Output	ND (atualiza variáveis em memoria: coluna_tetramino, tetramino_forma)



2.3 Interrupções

O projeto tem duas rotinas de interrupção ligadas aos dois clocks.

Int0	Int1
Desce a peça do tetraminó	Desenha e move o monstro
	

3. Conclusões

Em termos da compreensão do funcionamento da linguagem de programação *assembly* penso que conseguimos alcançar um bom nível de assimilação. Não conseguimos, porém, alcançar o pleno funcionamento do jogo.

Funcionalidade implementadas:

- Apresentação do ecrã de início e fim de jogo;
- Interrupções:
  - tetraminó descer;
  - monstro.
- Deteção da tecla premida;
- Movimento tetraminó:
  - movimento horizontal (esquerda/direita)
  - rotação.

Porém as interrupções e as rotinas de movimento do tetraminó entram em conflito quando em simultâneo.

Funcionalidades não implementadas:

- Estrutura de processos;
- Destruição do monstro pelo tetraminó

A respeito dos aspetos a melhorar:

- Implementar a rotina *random* (gerador de números aleatórios) (temo-la em código, as respetivas variáveis em memória, e a tabela de escolha do tetraminó)

- Desenho de tetraminó aleatório (recorrendo à rotina random)