

# INTRODUÇÃO À ARQUITETURA DE COMPUTADORES

IST – LEIC

## RELATÓRIO DO PROJETO CHUVA DE METEOROS

### 1. Manual de utilizador

O projeto foi desenvolvido e testado tendo em vista a versão do simulador disponibilizada no dia 17/06/2022.

Uma vez no **ecrã inicial**, o jogador tem acesso às seguintes ações:

- **C** – começar o jogo

No **jogo**:

- **C** – deslocar a nave para a esquerda
- **E** – deslocar a nave para a direita
- **9** – disparar um míssil
- **3** – pausar o jogo
- **2** – terminar o jogo

No **menu de pausa**:

- **3** – voltar ao jogo

Finalmente, no **ecrã de final de jogo** (que ocorre na colisão com um inimigo, quando acaba a energia ou no clique para terminar o jogo):

- **C** – jogar um jogo novo

## 2. Comentários

Além de todos os objetivos oficiais alcançados, também desenvolvemos algumas funcionalidades extra para **promover o envolvimento do jogador com o jogo**:

- Designs novos de inimigos e meteoros bons
- Sons adicionais de explosão e recuperação de energia
- Animação de explosão dos inimigos

Passando às práticas adotadas no desenvolvimento do programa, consideramos que podem ser divididas nas seguintes categorias:

### Clareza de código:

- Utilização extensa da diretiva *EQU*, evitando a utilização direta de constantes no corpo do programa
- Abstração procedimental, recorrendo nomeadamente ao uso de rotinas que são reutilizadas ao longo de todo o programa, como é o caso das rotinas *desenha\_objeto* e *apaga\_objeto* (utilizadas na renderização e movimento dos meteoros, tiros e nave). Para além disso, recorreremos também ao suporte do PEPE para processos cooperativos, respeitando o limite da funcionalidade intencionada para cada um (exemplo: o processo energia é o único que interage diretamente com a mesma).
- Comentários abundantes para facilitar a interpretação do código assembly

### Desempenho:

- O uso de processos apresenta uma vantagem em relação às rotinas cooperativas
- Utilização da diretiva *WAIT* no processo de leitura do teclado, que neste contexto apresenta uma vantagem face ao uso da diretiva *YIELD*

Uma funcionalidade que gostaríamos de ter implementado, mas não tivemos oportunidade de o fazer, seria o suporte para múltiplas instâncias do processo *missil*. Sabendo que cada tiro percorre no máximo 12 pixéis e cada movimento demora 200 ms, é possível determinar que um tiro vai durar, no máximo, 2,4 s. Restringindo o intervalo entre disparo de mísseis a, por exemplo, 600 ms, teríamos no máximo 4 mísseis em jogo simultaneamente.