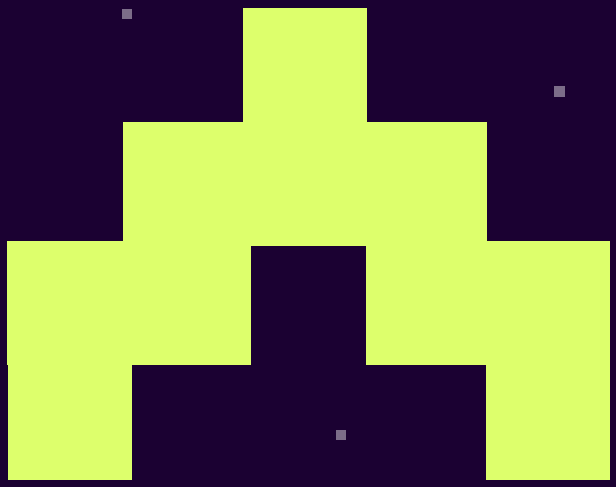
PROJETO DE IAC
CHUVA DE METEOROS

Licenciatura em Engenharia Informática e de Computadores



Grupo 71

Fábio Mata Nº 102802

Pedro Curto Nº 103091

Pedro Sousa Nº 102664

Junho de 2022

ÍNDICE

Instituto Superior Técnico: Licenciatura em Engenharia Informática e de Computadores

Fábio Mata - Pedro Curto - Pedro Sousa

INTRODUÇÃO

OBJETIVO DO JOGO E CONTEXTO

1

O jogo desenvolvido neste projeto, "Chuva de Meteoros", consiste numa simulação na qual um rover deve proteger o seu planeta, destruíndo naves inimigas (a vermelho) através do disparo de mísseis, e recolher energia de meteoros bons (a verde).

O rover tem energia associada, que se inicia a 100%, e decrementa periodicamente; o disparo de cada míssil tem também um custo adicional de energia.

A interface é constituída por um ecrã, onde se desenrola o jogo, um teclado, que permite começar, pausar e terminar o jogo, movimentar o rover e disparar mísseis, e um display de energia, que apresenta em todas as alturas o nível atual de energia do rover.

O objetivo é, portanto, manter o rover em execução o máximo de tempo possível enquanto se destrói o máximo de naves inimigas possíveis e se obtém energia dos meteoros bons, permitindo que embatam no rover.

O término do jogo pode ocorrer de três modos distintos:

- Por colisão com uma nave inimiga;
- Pela depleção completa de energia (quando atinge 0%);
- Por escolha do jogador, que pode terminar a qualquer momento.

PERSPETIVA DO JOGADOR

ECRÃ INICIAL

2

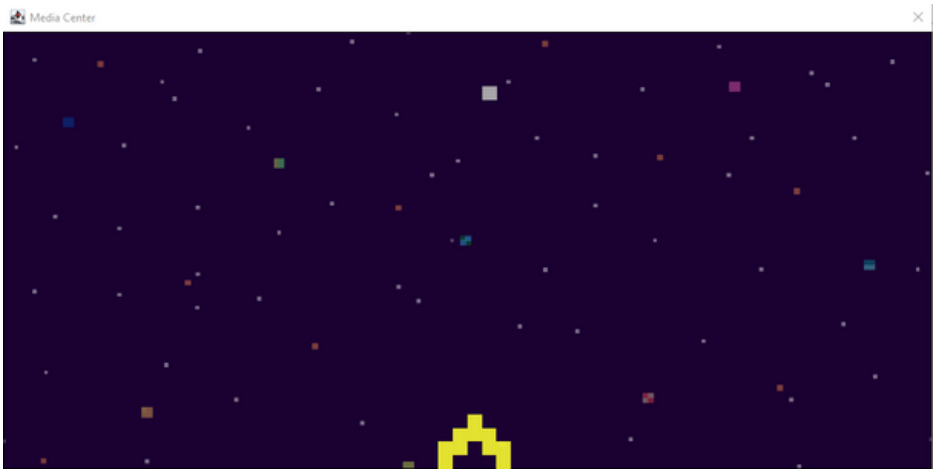
Esta é a primeira janela que se apresenta ao utilizador quando inicia o jogo, com o nome do jogo (em inglês), e com uma indicação da tecla a premir para começar o jogo.
A única hipótese de interação do jogador aqui é mesmo premir "C", para iniciar o jogo.

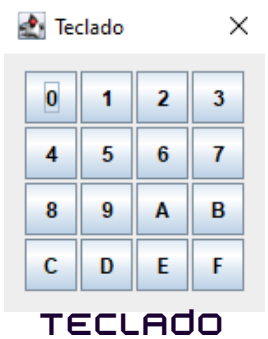
Instituto Superior Técnico ● Licenciatura em Engenharia Informática e de Computadores
Fábio Mata - Pedro Curto - Pedro Sousa

PERSPETIVA DO JOGADOR

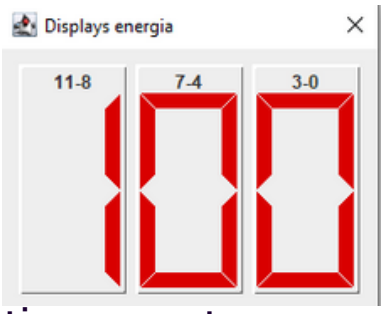
ECRÃ DE JOGO

3





TECLADO



DISPLAY DE ENERGIA

Aqui apresenta-se ao utilizador o ecrã principal, onde o jogo se desenrola. Uma música de fundo irá começar a tocar e o display de energia, que começa a 100, irá decrementar periodicamente de 5 em 5. A nave pode ser movimentada para a esquerda premindo a tecla 0 ou para a direita, premindo a tecla 2. A qualquer altura, o jogo pode ser pausado na tecla "D", ou terminado na tecla "E".

PERSPETIVA DO JOGADOR

ECRÃ DE PAUSA

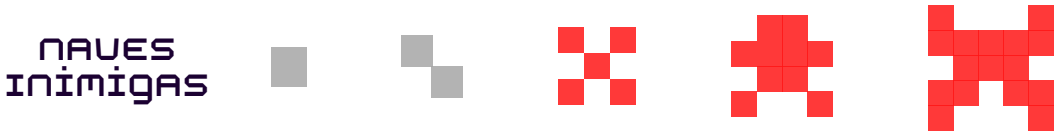
4

Este é o ecrã apresentado aquando do pressionamento da tecla "D". Nele, a música de fundo cessa, e o utilizador apenas pode retomar o jogo, clicando novamente na tecla "D", ou terminá-lo, se assim desejar, clicando na tecla "E". O display de energia manter-se-á naturalmente inalterado neste ecrã, e ao retomar, a evolução do jogo e a música continuam de onde foram parados.

PERSPETIVA DO JOGADOR

EVOLUÇÃO DO JOGO E INTERAÇÕES

5



Os meteoros bons e as naves inimigas são gerados numa coluna aleatória (75% naves, 25% meteoros), podendo coexistir no máximo 4 no ecrã ao mesmo tempo. Cada um tem 5 fases distintas, variando de tamanho entre 1x1 até 5x5 (fotografias na lateral), aumentando com a proximidade ao rover. Podem ser destruídos através de um míssil, premindo a tecla 1, com um custo de 5 de energia que é devolvida aquando da destruição de uma nave inimiga mas não de um meteoro, provocando ambos um som e efeito de explosão. O rover, ao colidir com um meteoro, ganha 10 de energia.

Instituto Superior Técnico ● Licenciatura em Engenharia Informática e de Computadores
Fábio Mata - Pedro Curto - Pedro Sousa

PERSPETIVA DO JOGADOR

ECRÃS DE GAME OVER

6

As figuras acima representam os três ecrãs para os quais o utilizador pode ser encaminhado quando o jogo termina, dependendo do que o provocou:

- na figura 1, o utilizador deliberadamente terminou o jogo, premindo a tecla "E";
- na figura 2, a energia do rover chegou a 0%, depletando-se por completo;
- na figura 3, o rover colidiu com uma nave inimiga.

Em qualquer um destes ecrãs, o utilizador está limitado na sua interação com o teclado, e apenas pode premir "C" para recomeçar o jogo, se assim o entender.

COMENTÁRIOS FINAIS

OPÇÕES, OBJETIVOS E RECURSOS EXTRA

7

Quanto às opções de projeto, decidimos por exemplo que, aquando da colisão de um míssil com duas naves sobrepostas, este destruíria ambas; teríamos um ecrã de *game over* distinto para a depleção de energia e para quando o utilizador decide voluntariamente terminar o jogo através da tecla "E" (além do ecrã de colisão com nave inimiga); e que cada meteoro seria desenhado no seu próprio ecrã, permitindo portanto que existam sobreposições.

Relativamente aos objetivos do enunciado, julgamos não haver nenhum que não tinha sido alcançado.

Em termos de *features* adicionais, o programa conta com as seguintes:

- Os meteoros *spawnam* com delays uns em relação aos outros, para tentar diminuir sobreposições e permitir ao jogador interagir com todos os meteoros e/ou naves;
- No ecrã de jogo, existe uma música de fundo a tocar em *loop* enquanto o jogo está ativo;
- O efeito sonoro de recolher um meteoro verde e colidir com uma nave inimiga é distinto.

COMENTÁRIOS FINAIS

DIFICULDADES E REFLEXÃO CRÍTICA

8

Naturalmente, encontrámos alguns problemas durante o desenvolvimento do projeto, que melhoraram a nossa capacidade de identificar e resolver um problema. Por exemplo, percebemos que, apesar de ativarmos as interrupções antes do spawn dos processos, estas não se encontravam ativas em cada um (contra o esperado), e portanto tivemos de ativar individualmente; além disso, também tivemos de inicializar o stack pointer no início de cada processo, algo que esperávamos que fosse feito automaticamente. Alguns dos bugs persistiram, como os meteoros aparecerem sobrepostos ao ecrã de pausa, apesar de estar selecionado como cenário frontal; e um bug muito raro, no qual a meio da execução do programa, o PC passava para zero e voltava a executar o código todo do início, duplicando todos os processos, algo que achamos que a nova versão do simulador corrigiu.

Fazendo um balanço do desenvolvimento do projeto, julgamos que foi uma boa experiência que permitiu entender muito melhor e solidificar alguns conceitos fundamentais da UC, como processos, rotinas, interrupções, registos vs acessos à memória, *stacks*, variáveis *lock*, e também como funciona e de que forma conseguimos coordenar a interação entre o processador e os diversos periféricos que compõem o circuito.