

Faculdade de Engenharia da Universidade do Porto



FEUP

ESPECIFICAÇÃO DO PROJETO

Ana Isabel Neves Alves de Sousa, 201108026, ei11068@fe.up.pt

Gabriel Cardoso Candal, 201108021, ei11066@fe.up.pt

Mestrado Integrado em Engenharia Informática e Computação

Laboratório de Computadores

22 de Novembro de 2012

Índice

Índice	2
1.PROJETO	3
2. PERIFÉRICOS	4
2.1. Teclado	4
2.2. Rato	4
2.3. Video Card	5
2.4. RTC.....	5
2.5. Timer	5
7. MÓDULOS.....	6
7.1. Controlo.....	6
7.2. Menu	6
7.3. Timer	6
7.4. Video Card	7
7.5. Controlo posição	7
7.6. RTC.....	7
8. PLANEAMENTO	8

1.PROJETO

Iremos realizar um jogo do género *Space Invaders*, em que o utilizador (representado por uma nave) tem como objectivo fugir de uma chuva de asteróides durante o maior intervalo de tempo possível. Difere da versão original na medida em que os asteróides são gerados de forma aleatória, a nave não está limitada a movimento horizontal, e os asteróides não serão estáticos como no original.

2. PERIFÉRICOS

Pretendemos utilizar os seguintes periféricos:

1. Teclado
2. Rato
3. *Video Card*
4. RTC
5. Timer

Todos irão utilizar interrupções menos o *video card* e não está previsto utilizar código em *assembly*.

2.1. Teclado

Será utilizado para destruir asteróides, para isso será necessário pressionar a tecla espaço, também será utilizado para habilidades especiais, como por exemplo, aumentar o número de vidas ou usar um ataque especial.

2.2. Rato

Será utilizado para o movimento da nave.

2.3. Video Card

Iremos utilizar a placa gráfica em modo gráfico.

2.4. RTC

Será utilizado para a passagem de nível, ou seja, aumento de dificuldade (associado a um aumento da velocidade com que novos asteróides são gerados) após x tempo sem perder.

2.5. Timer

Será utilizado para gerar novos asteróides de forma periódica.

7. MÓDULOS

7.1. Controlo

Responsável pela interacção entre o utilizador e o jogo através do rato e do teclado.

```
+ int control_int_handler();
```

7.2. Menu

Visualizado numa fase inicial, permite ao utilizador escolher as condições iniciais do jogo (escolher a nave e a dificuldade). Integra os restantes módulos.

```
+ void iniciaJogo();  
+ int leFicheiro(ifstream);  
+ int escolheNave();  
+ int escreveFicheiro(ofstream);  
+ void terminaJogo();
```

7.3. Timer

Acompanha o tempo decorrido desde o início do jogo, que será útil para gravação de pontuações máximas e aumento progressivo da dificuldade.

```
+ void timer_int_handler();
```

7.4. Video Card

Mostra informação ao utilizador: posição da nave, asteróides, vidas restantes, tempo decorrido e posição dos disparos.

```
+ void timer_int();
```

```
+ int vg_fill_screen(unsigned long);
```

```
+ int vg_set_pixel(unsigned long, unsigned long, unsigned long);
```

```
+ int vg_get_pixel(unsigned long, unsigned long);
```

```
+ int vg_draw_sprite(struct);
```

7.5. Controlo posição

Gere as colisões entre a nave e os asteróides.

```
+ int checkCol(unsigned long, unsigned long);
```



7.6. RTC

Usa os alarmes para aumento periódico da dificuldade.





```
+ void rtc_int_handler();
```

8. PLANEAMENTO


1ª semana (19.11.2012-25.11.2012)

-  carregar *sprites*
-  movimentar a nave



2ª semana (26.11.2012-02.12.2012)

-  menu
-  gerar asteróides
-  ajuste de dificuldade
-  temporizador


3ª semana (03.12.2012-09.12.2012)

-  colisões

4ª semana (10.12.2012-16.12.2012)

-  colisões
-  destruir asteróides

5ª semana (17.12.2012-23.12.2012)

-  últimas correções