

Faculdade de Engenharia da Universidade do Porto



Fireboy & Watergirl

Projeto
Laboratório de Computadores
Turma 6 – Grupo 3

Projeto realizado por:

Bruno Miguel de Siuéia Duvane (up202109244)
Carlos Manuel da Silva Costa (up202004151)
Lara Santos Bastos (up202108740)
Rita Gonçalves Pereira (up202108746)

Índice

Introdução.....	4
Instruções para o utilizador.....	5
Estado do Projeto	10
Dispositivos Implementados.....	10
Placa Gráfica	10
Timer.....	11
Teclado	11
Rato	11
Real Time Counter.....	12
Serial Port.....	12
Estrutura do código	13
Módulo do Timer	13
Módulo do Teclado	13
Módulo do Rato	13
Módulo do KBC	13
Módulo da Placa Gráfica.....	13
Módulo do Real Time Counter	13
Módulo Utils.....	14
Módulo do Serial Port	14
Módulo Sprite.....	14
Módulo Sprite_Management.....	14
Módulo Handle Int.....	14
Módulo Draw.....	14
Módulo Element	14
Módulo Game.....	15
Módulo Main.....	15
Function Call Graph	16
Detalhes de Implementação.....	17
Real Time Counter.....	17
Serial Port.....	17
Detecção de Colisões	17
Saltos	17

Animações.....	18
<i>Conclusão.....</i>	<i>19</i>
<i>Referências.....</i>	<i>20</i>

Introdução

O projeto é uma adaptação do jogo *Fireboy & Watergirl* e possui dois modos: jogar apenas num computador ou em dois computadores.

O jogo têm duas personagens, o *Fireboy* e a *Watergirl*, cujo o objetivo é sobreviver aos vários obstáculos existentes no mapa, chegando à porta final. Para saírem do labirinto em que se encontram, é, ainda, necessário que se puxe a alavanca responsável por abrir as portas.

No primeiro modo de jogo, o jogador pode movimentar uma personagem através das setas e a outra através das teclas “A”, “W” e “D”. No segundo modo, cada computador joga com uma das personagens.

Existem vários níveis jogo ao longo dos quais a dificuldade vai incrementando.

Instruções para o utilizador

1. Menu inicial

Ao iniciar o jogo aparece o seguinte menu inicial:



Assim, o utilizador pode escolher o que pretende fazer clicando com a botão esquerdo do rato na opção pretendida:

- **1 computer:** iniciar o jogo no modo de 1 computador,
- **2 computers:** iniciar o jogo no modo de 2 computadores
- **Exit:** sair do jogo.

A qualquer momento o utilizador pode também sair do jogo pressionando o Q ou o ESC.

2. Modo de 1 computador

Ao iniciar o modo de *1 computador*, aparece o mapa do primeiro nível com as duas personagens. Para andar para a frente, para trás e para saltar com a *WaterGirl* deve usar-se a seta para a direita, esquerda e para cima, já com o *FireBoy* as teclas usadas devrão ser 'A', 'D' e 'W', respetivamente.

Para abrir as portas é necessário puxar a alavanca. Para isso deve-se clicar na mesma com o botão do lado esquerdo do rato e fazer um movimento horizontal para o lado pretendido. De modo a ganhar o nível, ambas as personagens devem chegar à sua porta respetiva e, para além disso, esta deve encontrar-se aberta.

Para reiniciar o jogo, basta clicar no botão no canto superior esquerdo

2.1 Nível 1

No primeiro nível, o único obstáculo existente são alguns lagos. Se alguma das personagens tocar num lago que não o da sua cor, morre e o utilizador perde o nível.



2.2 Nível 2

O segundo nível do jogo é semelhante ao primeiro, no entanto tem um obstáculo novo: uma parede que se destrói quando o botão vermelho for pressionado. Tendo em conta que apenas a WaterGirl consegue chegar ao botão, esta deve estar em cima dele para a parede se destruir, permitindo que as personagens avancem no jogo.



2.3 Nível 3

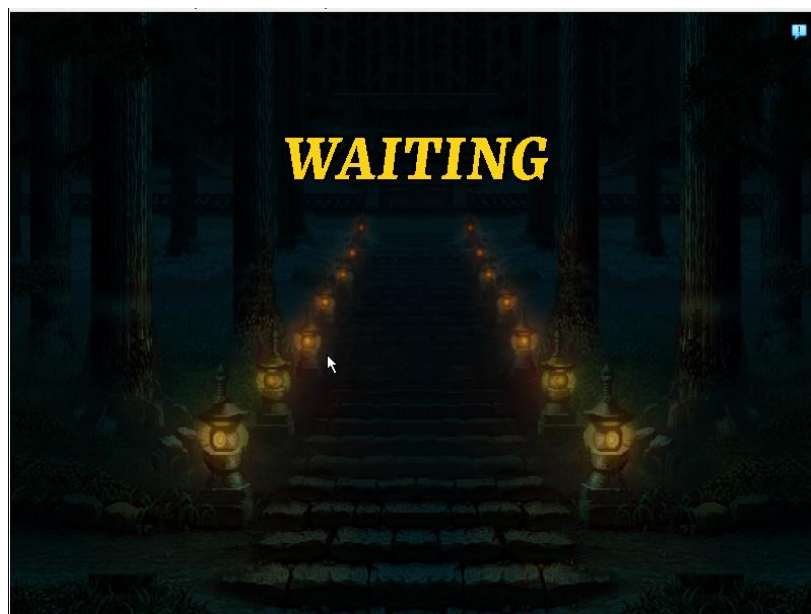
Por fim no último nível, o FireBoy precisa do auxílio de uma plataforma para chegar ao 2ª andar. Para subir a plataforma, a WaterGirl deve, mais uma vez, clicar no botão

3. Modo 2 computadores

Neste modo, as instruções do jogo são semelhantes, a única diferença é que cada utilizador fica alocado a uma personagem e apenas pode movimentar essa. Fora isso, os níveis, os obstáculos e as restantes regras do jogo mantêm-se.



Ao entrar, caso não existam mais nenhum computador conectado, aparece o seguinte ecrã:



4. Menus

Ao jogar cada um dos níveis é possível ganhar ou perder, e em cada um destes casos aparece um menu diferente com várias opções. Para selecionar uma das opções o jogador deve usar o rato.

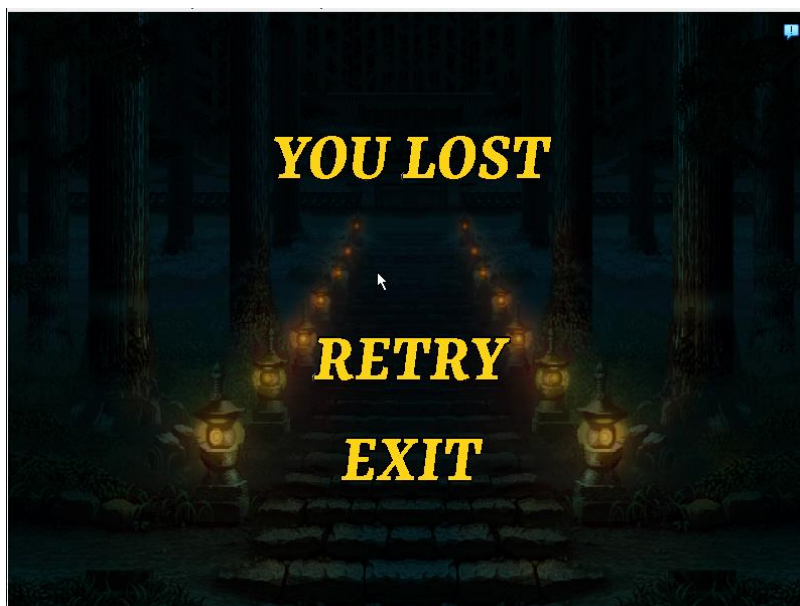
4.1 Menu “GANHOU”



Este menu aparece ao ganhar os níveis 1 e 2 e tem duas opções:

- Passar para o próximo nível
- Sair do jogo

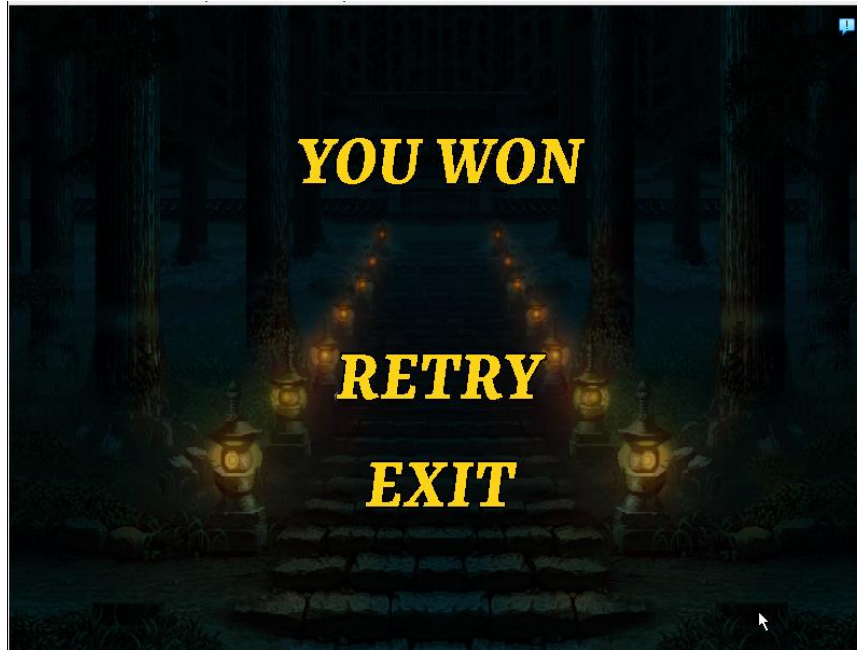
4.2 Menu “PERDEU”



Este menu, por sua vez, aparece quando se perde um qualquer um dos níveis e tem duas opções:

- Repetir o nível
- Sair do jogo

4.3 Menu “FIM DE JOGO”



Este menu aparece quando o utilizador ganha o nível 3, o último. Tem como opções:

- Recomeçar do nível 1
- Sair do jogo

Estado do Projeto

Dispositivos Implementados

Dispositivo	Funcionalidades	Interrupções
Timer	Sincronizar dispositivos, animações e saltos	Sim
Teclado	Controlar os movimentos das personagens	Sim
Rato	Selecionar as opções no menu e puxar a alavanca final para passar o nível	Sim
Placa Gráfica	Mostrar as interfaces do jogo, nomeadamente o mapa, as personagens e os menus	Não
Real Time Counter	Mostrar a data atual (dia e hora de jogo).	Sim
Serial Port	Fazer a comunicação entre dois jogadores, permitindo jogar no modo: 2 computadores	Sim

Placa Gráfica

A placa gráfica serve para mostrar os menus, níveis e as personagens a movimentarem-se.

Ao longo do jogo esta encontra-se no modo de vídeo, modo VBE0x115, com uma resolução de 1024x768 pixels e com as cores no modo 8:8:8.

Utilizamos a técnica de Double Buffering, via copy. O primeiro buffer é alterado apenas com interrupções do timer, enquanto o segundo é atualizado quando os elementos mudam de estado-

O display das várias imagens é feito através de XPMs, nomeadamente os fundos, as personagens, os textos e os obstáculos. Para além disso, implementamos a deteção de colisões (personagens com as paredes laterais, chão, teto, lagos etc.) e animações (ao andar para os lados e a saltar).

Funções relevantes:

- Funções do ficheiro graphics.c, draw.c, sprite_management.c

Timer

O timer foi utilizado para as animações de saltos e desenhar a plataforma no nível a subir a uma velocidade constante. É utilizado também para controlar a leitura das interrupções do RTC, de modo que estas apenas sejam processadas a cada segundo.

Funções relevantes:

- Funções do ficheiro timer.c
- handle_int_timer() – atualiza as Sprites dos elementos caso estes estejam a saltar ou a subir e gere o Double Buffer

Teclado

O teclado recebe input do utilizador durante o jogo, processa os scancodes gerados e move os bonecos de acordo com a tecla que receber. Independentemente do modo de jogo em que se está a jogar, as três setas, correspondentes *WaterGirl*, e as teclas 'A', 'D' e 'W', correspondentes ao *FireBoy*, fazem as personagem movimentarem-se para a frente, trás e saltar, respetivamente. Em qualquer momento do jogo é, também, possível sair pressionando a tecla 'Q' ou 'ESC'.

Funções relevantes:

- Funções do ficheiro keyboard.c
- handle_int_keyboard() – atualiza o jogo de acordo com a tecla pressionada em modo singleplayer
- handle_int_keyboard_multiplayer_transmit() - atualiza o jogo de acordo com a tecla pressionada em modo multiplayer e envia esses dados para o serial port

Rato

O rato é utilizado para selecionar as opções nos menus e puxar a alavanca final para abrir as portas e terminar o jogo. Assim, utilizamos os botões e a posição do rato.

Para selecionar a opção pretendida nos menus, deve ser clicado o botão do lado esquerdo do rato na posição correta. Para além disso, durante o jogo, quando as duas personagens chegarem à porta final, o utilizador deve, ainda, clicar na alavanca com o botão esquerdo do rato e fazer um movimento horizontal para o lado que pretende mover a alavanca

A cada interrupção do rato, os três bytes gerados são guardados na estrutura MouseInfo de modo a conhecer a posição atual do rato, as teclas pressionadas e o deslocamento horizontal.

Funções relevantes:

- Funções do ficheiro mouse.c

- `handle_int_mouse_click()` - atualiza o jogo de acordo com as interrupções recebidas do rato

Real Time Counter

O RTC foi usado para ler a data e hora a que o utilizador está a jogar. O display desta informação é feita em dia/mes/ano hora:minuto.

Funções relevantes:

- Funções do ficheiro `rtc.c`
- `handle_int_rtc()` - preenche a struct a cada 60 interrupções do timer
- `displays_time()` – atualiza a data no ecrã do jogo de cada vez que uma interrupção do rtc é processada

Serial Port

A Serial Port foi utilizado para enviar e receber dados entres os dois computadores permitindo que haja comunicação entre estes. Para isto, usamos interrupções, e enviamos 8 bits para o outro computador de cada vez que há uma interrupção do teclado ou do rato durante o jogo. Estes 8 bits podem representar o scancode enviado pelo KBC ou um comando que representa que a alavanca foi aberta ou fechada.

Funções relevantes:

- Funções do ficheiro `sp.c`
- `handle_sp_data_multiplayer()` – processa os dados recebidos através do serial port e atualiza o jogo de acordo com estes
- `handle_ind_sp()` – inicializa os jogadores ao entrar ao modo multiplayer

Estrutura do código

Módulo do Timer – 8%

As funções do timer são idênticas às funções desenvolvidas no Lab2 e serve, para o configurar e receber as suas interrupções.

Módulo do Teclado –8%

As funções do teclado são idênticas às funções desenvolvidas no Lab3, tendo como principal objetivo configurar o teclado, receber scancodes do KBC e escrever comandos no mesmo e, para além disso, receber as interrupções do teclado

Módulo do Rato – 8%

As funções do rato são idênticas às funções desenvolvidas no Lab4, com algumas alterações. Servem para receber os 3 bytes derivados das 3 interrupções que cada movimento no rato provoca, escrever comandos no KBC e receber interrupções do rato. No entanto, alteramos a estrutura MouseInfo para que apenas tivesse as informações necessárias à implementação do jogo, nomeadamente clique da tecla esquerda, posição dos cursores, movimento no eixo dos x e sinal deste movimento.

Módulo do KBC – 5%

As funções do KBC são idênticas às do ficheiro KBC.c do Lab3 e do Lab4 e auxiliam a utilização do rato e do teclado. Leem o estado do KBC, o seu output e escrevem comando neste.

Módulo da Placa Gráfica – 5%

As funções da placa gráfica são idênticas às funções desenvolvidas no lab5, tendo como objetivo configurar a placa gráfica e desenhar o todos os menus, personagem e obstáculos no ecrã.

Ao contrário do Lab, no nosso projeto, a placa gráfica apenas trabalha com XPMs, pelo que retiramos as funções que não eram necessárias.

Módulo do Real Time Counter – 5%

As funções do RTC servem para configurá-lo, ler o seu output e organizá-lo na struct `real_time_info` para poder ser mostrada a data e a hora durante o jogo.

Módulo Utils – 2%

Este módulo contém as funções presentes no ficheiro `utils.c` do Lab2 e que serviram de auxílio para os restantes Labs. Implementa funções para aceder ao byte mais ou menos significativo de uma variável de 18 bits e `util_sys_inb`, que permitem ao utilizador aceder à informação de um registo usando uma variável de 8 bits, em detrimento de uma de 32 bits.

Módulo do Serial Port –10%

Este módulo, por sua vez, contém as funções relativas à implementação da porta série, ou seja, verifica se existem dois jogadores prontos a jogar, inicializa o jogo em caso de resposta afirmativa, envia e recebe dados e, por fim, subscreve as interrupções do dispositivo.

Módulo Sprite – 2%

Neste módulo encontra-se as funções utilizadas para alocar memória e posteriormente libertá-la. Esta classe foi retirada dos power points das aulas teóricas.

Módulo Sprite_Management – 3%

Este módulo, possui funções para interagir e gerir as Sprites do jogo, ou seja criá-las todas inicialmente, desenhá-las quando necessário e, por fim, destruí-las.

Módulo Handle Int – 15%

Este módulo contém as funções que interagem com o jogo a cada interrupção recebida por qualquer um dos dispositivos. De cada vez que, no loop principal do projeto, é recebida uma interrupção é chamado o respetivo interrupt handler e, posteriormente, a função neste ficheiro relativa a esse dispositivo. Estas interpretam a interrupção recebida e, caso importante para o jogo, atualiza a posição das personagens, o estado do jogo, o nível etc.

Módulo Draw – 8%

Neste módulo vão ser desenhados todos os menus e os mapas de jogo. Recorrem ao array do nível para saber quais das Sprites é suposto dar display e em que posição.

Módulo Element – 8%

Este módulo atualiza os vários elementos do jogo (personagens, mapa, obstáculo) de acordo com o que está a acontecer e com as interrupções que estão a ser recebidas. Para este efeito, criamos variáveis globais que nos indicam o estado atual destes vários elementos do jogo.

Módulo Game – 8%

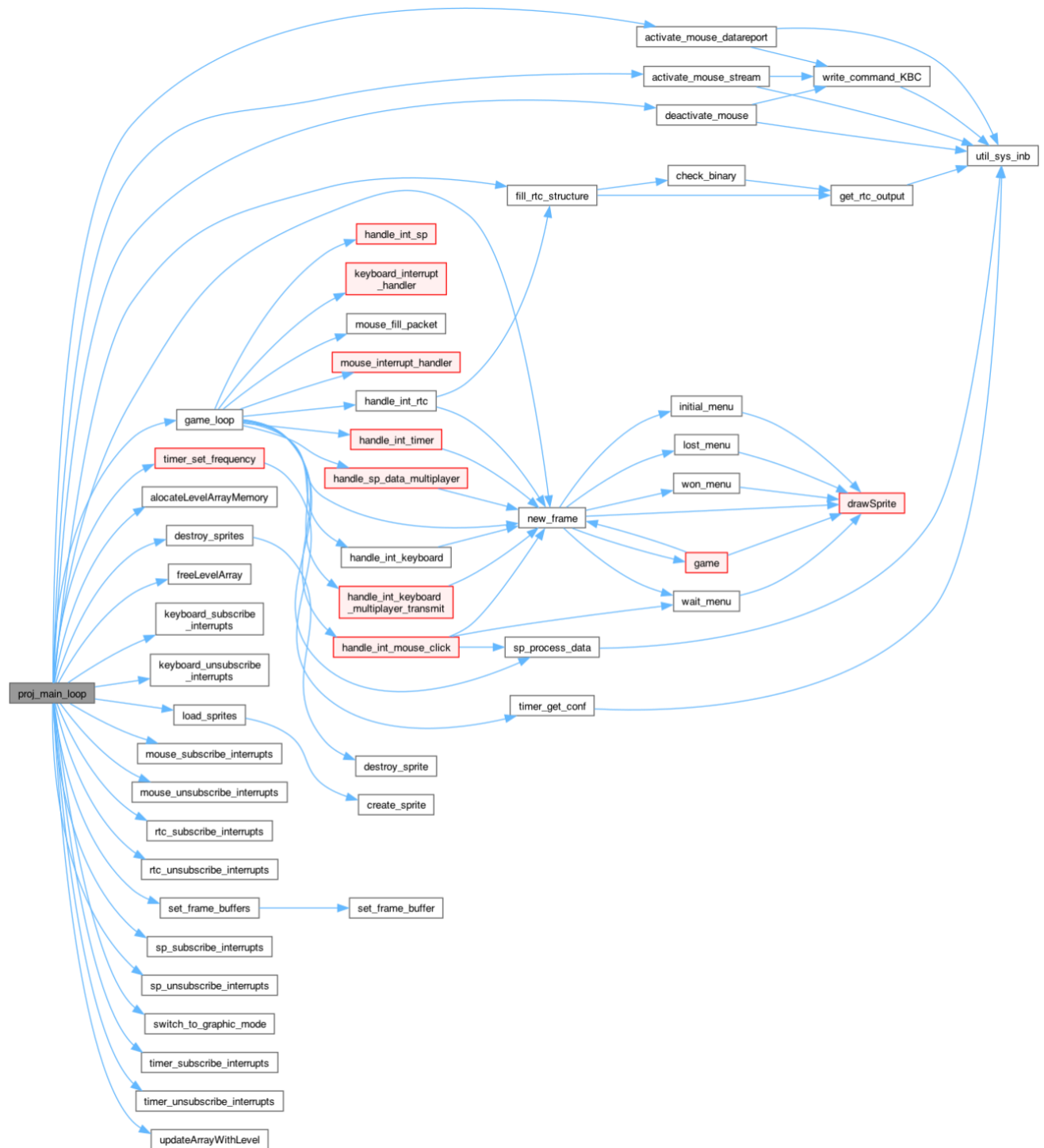
Neste módulo, está o loop principal do jogo, onde são recebidas as interrupções de cada dispositivo, chamados os interrupt handlers e as funções que interagem com o jogo de acordo com a informação recebida.

O loop das interrupções foi retirado dos slides das aulas teóricas.

Módulo Main – 5%

Por fim, neste módulo, são subscritas as interrupções de todos os dispositivos que vão se utilizados, é chamado o loop principal do jogo e, por fim, é feito unsubscribe das interrupções.

Function Call Graph



Detalhes de Implementação

Real Time Counter

De modo a ler a data do jogo, a cada interrupção é enviado um comando para cada uma das informações que pretendemos (ano, mês, dia, hora e minutos) que indica qual destas vai ser lida para o registo 0x70, e posteriormente lida a informação do registo 0x71. Para cada uma destas leituras, é colocada numa variável da estrutura RTC_structure a informação na posição correspondente. Algumas considerações importantes durante este processo são verificar se o modo de contagem está em binário ou BCD e se o RTC não está a atualizar os valores.

Tivemos o cuidado de ler o RTC apenas a cada 3600 interrupções do timer, de modo a atualizá-lo apenas uma vez por minuto e não fazer leituras desnecessárias.

Serial Port

O serial port foi implementado para que houvesse a oportunidade de existir um modo de jogo entre dois computadores. De cada vez que, no modo de jogo multiplayer, era recebida um interrupção quer do rato quer do teclado, os dados eram enviados para o outro computador através de uma variável de 8 bits . Esta pode ser ou o scancode da tecla pressionada no teclado ou um comando que indica que a alavanca foi aberta/ fechada. A cada interrupção do serial port, ou seja, de cada vez que é recebida informação do outro jogador, esta é lida e o jogo é alterado.

Deteção de Colisões

De modo a permitir que as personagens não saiam do ecrã, possam estar em cima de plataformas em várias alturas diferentes e morram quando pisam um lago que não for da sua cor, foi necessário implementar a deteção de colisão destas com os vários elementos do jogo.

Para isto, sempre antes de movimentar uma personagem, verificamos no array do nível, se a posição para a qual ela vai está livre. Caso não esteja, ou o movimento não é feito, nomeadamente ao colidir com paredes, ou perde o nível, caso toque num lago que não pode.

Saltos

Durante o jogo, além de andar para frente e para trás, as personagens podem também saltar, quer para cima quer horizontalmente. Utilizamos assim o timer para coordenar o movimento do Sprite das personagens e garantir que passa em todas as posições pretendidas ao longo de um salto. Implementamos também uma função que verifica a gravidade, ou seja, se a personagem não estiver em cima de uma plataforma vai cair, mais uma vez a um frame rate constante coordenado pelo timer.

Animações

Por fim, implementamos também algumas animações durante o jogo. A Sprite de cada personagem muda quando esta está a andar ou a saltar. Este efeito foi obtido com o auxílio de variáveis que nos indicam qual a direção atual de cada uma bem como um contador das frames de animação que indica a sprite da animação a ser mostrada.

Conclusão

A maior dificuldade surgiu da implementação dos saltos das personagens, uma vez que é uma feature nunca antes explicada ou implementada na UC. A implementação da porta séria foi também um desafio, uma vez que não foi praticada em nenhum dos Labs, no entanto, conseguimos fazê-lo sem problemas. Acreditamos assim que estes são as duas maiores conquistas do projeto.

Efetivamente, conseguimos implementar todos os dispositivos e funcionalidades planeadas e mencionadas no formulário inicial. Na nossa opinião, a única melhoria que poderia ser feita no futuro e que acrescentaria valor ao projeto seria a criação de mais níveis, mais complicados e com obstáculos diferentes.

Como consequência do trabalho feito ao longo do desenvolvimento do projeto, conseguimos compreender melhor como trabalhar com interrupções, como são programados os dispositivos de hardware num computador e como coordená-los de modo a poderem ser utilizados em simultâneo.

Referências

- Classe Sprite.h /Sprite.c retirada dos slides das aulas teóricas
- Loop das interrupções retirado dos slides das aulas teóricas