

6 DE JANEIRO DE 2020



LABORATÓRIO E COMPUTADORES
2019/2020

TERMINIX

TURMA 8 - GRUPO 5

TRABALHO REALIZADO POR:

BERNARDO DA SILVA MOÇO DE SOARES RAMALHO - UP201704334
TIAGO ALEXANDRE PINTO DE FARIA FERREIRA ALVES - UP201603820

ÍNDICE

ÍNDICE.....	1
User intructions.....	2
Menu.....	2
1 Player	2
Co-Op	4
HighScore	5
Project Status	6
Timer.....	6
Teclado.....	6
Rato.....	7
Placa Gráfica.....	7
RTC.....	7
Serial Port.....	8
Code organization	9
Drawing.....	9
Game.....	9
Hero	9
Highscore	10
Keyboard	10
Menu.....	10
Serial Port.....	Error! Bookmark not defined.
Proj	11
RTC.....	11
Serial Port.....	11
Sprite.....	11
Terminix	12
Timer.....	12
Zombie	12
Implementation Detalis	14
Conclusions	15

USER INTRUCTIONS

Menu

Quando se começa o jogo, é se levado para o Menu principal em que as escolhas possíveis são apresentadas em modo de botão. Pode selecionar uma opção clicando sobre a opção pretendida, com o botão esquerdo do rato.



Neste menu existem três opções:

1 Player

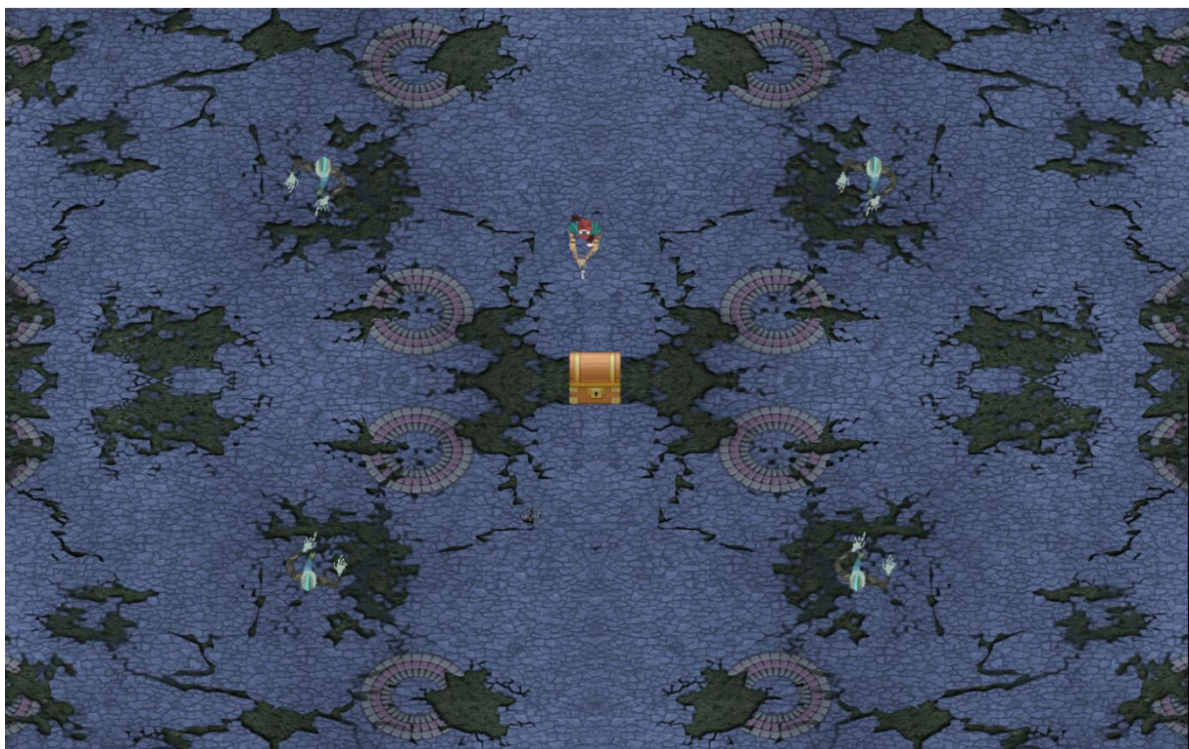
Quando esta opção é escolhida, o jogo principal é iniciado, havendo apenas um jogador em jogo. Mal o jogo comece, o jogador é confrontado com um herói, dois

zombies e um baú, estando os zombies a perseguir os outros dois. O objetivo do jogador é matar ambos os zombies sem que nenhum deles lhe toque a ele ou ao baú.

Para mover o herói deve premir as seguintes teclas:

- **W** – Andar para cima;
- **A** – Andar para a esquerda;
- **S** – Andar para baixo;
- **D** – Andar para a direita;
- **ESC** – Sair do jogo;
- **Botão esquerdo do rato** – Disparar.

Através do movimento é possível ainda rodar o herói, estando este sempre apontado na direção do rato.



O jogo tem vários níveis, sendo o critério de passagem matar todos os zombies antes que eles atinjam ou o tesouro, ou o herói, sendo que em cada nível a dificuldade é aumentada e são incrementados o numero de zombies.

No caso de o jogador ganhar o jogo, é apresentada a imagem à esquerda, e caso perca, à direita.



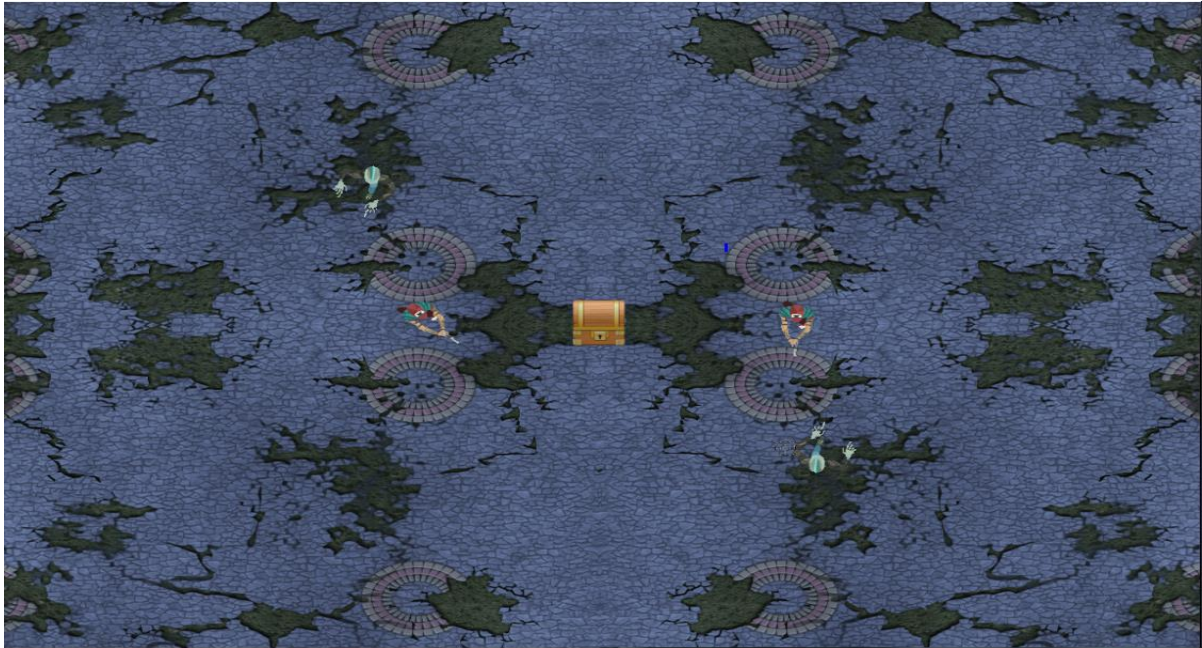
Co-Op

Quando é seleccionada a opção de Co-Op, o utilizador é confrontado com uma janela de espera, até que o outro jogador se junte na outra máquina e a comunicação ter sido estabelecida.



Neste modo, existem dois heróis, sendo que cada um tem controladores diferente. O objetivo de ambos é igual ao modo de 1 jogador, e os controlos também, sendo

que a única diferença é um terceiro nível escondido, quase impossível de se ganhar sozinho.

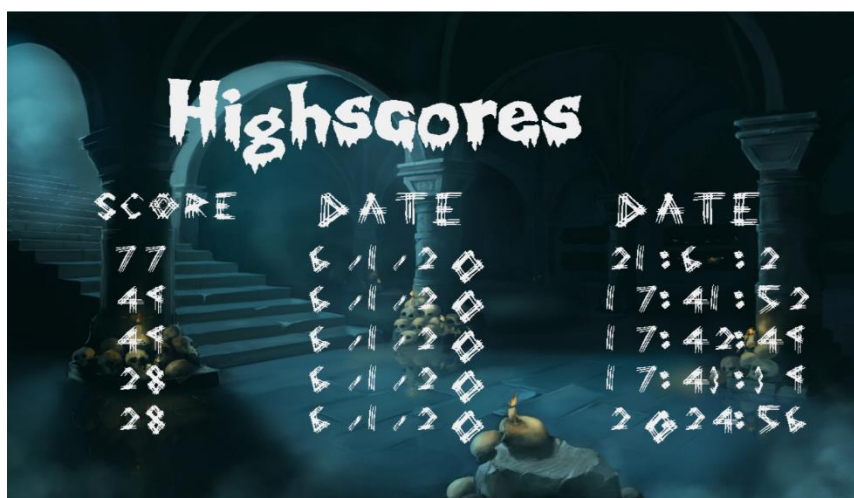


HighScore

No Menu Highscore, são apresentados os 5 melhores resultados de sempre no jogo, estando cada um associado a uma data e hora, de quando foram feitos.

Estes estão ordenados por ordem decrescente de pontuações, estando o jogador com mais pontos acima de todos os outros.

Para regressar ao menu anterior o jogador deve premir o botão esquerdo do rato.



PROJECT STATUS

Periférico	Função	Interrupção
Timer	Atualização de <i>frames</i>	Sim
Teclado	Movimento dos heróis	Sim
Rato	Orientação do herói, mira e seleção de opções	Sim
Placa Gráfica	Display do Jogo	Não
Real Time Clock	A	Não
Serial Port	Comunicação entre os dois pcs para ser possível jogar em Co Op	Sim

Timer

Primeiramente, o timer é utilizado para controlo do frame rate do jogo, tendo usado as interrupções do timer para desenhar todas as imagens numa frequência de 60 Hz.

Além disso, este é utilizado para controlar o movimento dos Zombies e das balas por exemplo, sendo estas incrementadas de uma determinada velocidade sempre que a cena é desenhada.

Teclado

No projeto o teclado é essencialmente usado para mover o herói, movendo-o em determinada direção sempre que recebe um determinado scan-code.

É também utilizado para interromper o ciclo dos interrupts e sair do jogo, caso o utilizador deseje.

Rato

O rato é utilizado tanto nos menus, como durante o jogo em si, utilizando em ambos os casos o botão direito do rato, seja para disparar ou para escolher uma determinada opção no menu, e o movimento para, por exemplo, determinar a rotação do jogador.

Placa Gráfica

Neste projeto optámos por usar o modo 0x14C, com uma resolução 1152x864 e um modelo de cor direto. Foi ainda implementado um double-buffer, de modo a termos uma imagem consistente e uma aparência melhorada.

Durante o correr do jogo o jogador é confrontado com variadíssimas animações, sendo algumas feitas automaticamente, como o exemplo dos zombies que se deslocam na direção do herói a uma determinada velocidade a cada frame, ou algumas controladas pelo teclado, como é o caso do herói que é animado premindo as teclas já referidas.

O nosso jogo é muito à base de colisões, sendo que o resultado do jogo é determinado por colisões de imagens, como o herói com o zombie ou uma bala com o tesouro.

Optámos no nosso trabalho por usar o modelo Hit-Box, optando por uma opção mais eficiente que, na nossa opinião, pouco afeta a jogabilidade.

RTC

Ao longo do trabalho apenas consideramos útil para o nosso trabalho, o uso das interrupções de alarme e a leitura de data.

A principal funcionalidade do RTC, é para apresentar os melhores resultados de sempre, do Terminix, sendo que cada pontuação está associada a uma data e hora de quando foi feita.

Além disto, decidimos ainda implementar um pequeno easter EGG no nosso jogo, em que caso o jogador esteja no menu do highscore, num determinado período de tempo, é confrontado por uma mensagem secreta.

Apesar de nem os update como os periodic interrupts terem sido utilizados, foi apenas porque considerámos que eram desnecessárias ao jogo, tendo as funções para as implementar, sido criadas de qualquer maneira.

Serial Port

O serial port é usado quando se joga em Co-Op. Neste modo, o serial port (usado com interrupções) serve para sincronizar o jogo de ambos os jogadores. Isto faz-se enviando mensagens quando alguma coisa importante acontece (por exemplo, o movimento do herói). Esta mensagem é depois analisada de forma a ter ambos os jogos iguais.

CODE ORGANIZATION

Durante a realização deste trabalho, tivemos muito em conta a organização do nosso projeto, assim como a legibilidade para futuros programadores que possam vir a utilizar o nosso programa. Optámos por uma divisão bastante modular do projeto, sendo cada módulo responsável por uma coisa muito específica.

Os diagramas de funções foram gerados pelo Doxygen.

Drawing

Este módulo é responsável por tratar de todos os ficheiros .xpm que temos e desenhá-los. Todas as funções usadas foram feitas, por nós, para os labs e adaptadas ao projeto.

Contribuição: Bernardo Ramalho / Tiago Alves (50 / 50)

Game

Este módulo é responsável por toda a lógica do jogo. É neste módulos que se encontram as funções que tratam dos estados do jogo, das colisões dos objetos, das atualizações das posições e análise das mensagens enviadas pelo serial port.

Todas as funções deste módulo foram feitas por nós.

Contribuição: Bernardo Ramalho / Tiago Alves (50 / 50)

Hero

Este módulo é responsável por tudo o que é relacionado com o herói. É neste módulo que se calcula a direção do herois, a nova posição e onde se cria/destrói o herói. Toda a informação do hero é guardada numa struct Hero (criada por nós).

Todas as funções deste módulo foram feitas por nós.

Contribuição: Bernardo Ramalho / Tiago Alves (25 / 50)

Highscore

Neste módulo foram realizadas todas as operações relacionadas com as pontuações, nomeadamente o uso do RTC e a escrita de texto para ao ecrã

Keyboard

Este módulo é responsável pelas funcionalidades do teclado. É neste módulo que se encontra as funções para subscrever e para analisar a informação proveniente do keyboard.

As funções foram as que utilizamos no lab do teclado mas foram adaptadas para este projeto.

Contribuição: Bernardo Ramalho / Tiago Alves (25 / 75)

Menu

Este módulo é responsável por criar e destruir os menus e, também, por desenhar e limpar as várias imagens do menu. Toda a informação do menu é guardada numa struct Menu (criada por nós).

As funções foram feitas por nós para este projeto.

Contribuição: Bernardo Ramalho / Tiago Alves (25 / 75)

Mouse

Este módulo é responsável pelas funcionalidades do rato. É neste módulo que se encontra as funções para subscrever e para analisar a informação proveniente do rato. Este módulo é mais complexo que o do teclado pois tem

como base uma maquina de estados. Toda a informação do rato é guardada numa struct Mouse (criada por nós).

As funções foram as que utilizamos no lab do rato mas foram adaptadas para este projeto.

Contribuição: Bernardo Ramalho / Tiago Alves (75 / 20)

Proj

É neste módulo que o jogo corre, apenas contendo um ciclo de interrupções. A base deste ficheiro foi dada pelo professor mas nós tivemos que adaptar ao nosso projeto.

Contribuição: Bernardo Ramalho / Tiago Alves (50 / 50)

RTC

Neste módulo encontramos a base para o funcionamento do RTC. Neste módulo criámos todas as funções relativas a este, incluindo as funções de subscrição, de configuração, de envio e onde lidamos com as interrupções deste periférico.

Contribuição: Bernardo Ramalho / Tiago Alves (25 / 75)

Sprite

Este módulo é bastante simples tendo só uma função para criar e outra para destruir. A informação do sprite é guardada numa struct Sprite.

Estas funções foram feitas por nós.

Contribuição: Bernardo Ramalho / Tiago Alves (50 / 50)

Serial Port

Este módulo é responsável pelas funcionalidades do rato. É neste módulo que se encontra as funções para subscrever e para analisar a informação proveniente do rato. Este módulo é mais complexo que o do teclado pois tem como base uma maquina de estados. Toda a informação do rato é guardada numa struct Mouse (criada por nós). As funções foram as que utilizamos no lab do rato mas foram adaptadas para este projeto. Contribuição: Bernardo Ramalho / Tiago Alves (75 / 20)

Terminix

Este módulo é responsável por agregar toda a informação do jogo numa struct chamada Terminix. É aqui que criamos os inimigos, criamos o tesouro e criamos o segundo herói.

Todas as funções foram feitas por nós.

Contribuição: Bernardo Ramalho / Tiago Alves (75 / 25)

Timer

Este módulo tem como função subscrever o timer e lidar com as suas interrupções, incrementando um counter.

Estas funções foram feitas para o lab do timer e adaptadas para o projeto.

Contribuição: Bernardo Ramalho / Tiago Alves (75 / 25)

Zombie

Este módulo tem como função criar e destruir zombies. A informação relativa a um zombie é guardada numa struct Zombie (criada por nós).

Todas as funções foram criadas por nós.

Contribuição: Bernardo Ramalho / Tiago Alves (75 / 25)

Módulo	Peso Relativo
Drawing	10%
Game	30%
Hero	5%
Highscore	2%
Keyboard	10%
Menu	3%
Mouse	10%
Proj	5%
RTC	5%
Serial_port	5%
Sprite	2%
Terminix	8%
Timer	3%
Zombie	2%

IMPLEMENTATION DETAILS

Todo o jogo está assente num único loop while. Neste loop encontram-se todas as subscrições. Quando há uma interrupção de um periférico, essa interrupção é analisada. Cada handler é responsável por atuar sobre o periférico segundo a interrupção. O render do jogo é feito quando há interrupções do timer

Neste render, usa-se uma maquina de estados (usando o estado do jogo) para fazer as várias coisas. Quando se está no estado “level” temos que é tudo feito através de uma função `update_position`. Neste função, são atualizados todos os objetos e vai se fazendo as colisões entre os objetos. Para aumentar a fluidez da representação dos objetos usamos double buffering. Isto também ajuda a prevenir frame stuttering.

Uma coisa a notar é o calculo do movimento dos inimigos. Os zombies movem-se em direção ao objeto mais próximo (sem contar com outros zombies). É algo que torna o jogo mais fluido e menos monótono, visto que o movimento dos zombies nunca é igual.

De modo a encapsular o nosso código e de ter uma abordagem orientada a objetos, implementámos várias structs, sendo que quase todos os módulos, sem ser dos periféricos têm a sua. De modo a não repetir código, criámos também uma struct `Sprite`, que contém a imagem, velocidade e posição correspondente ao xpm.

O serial port (UART) foi configurado com 8 bits por char, paridade ímpar e 1 bit de paragem. Foi utilizado através de interrupts visto que precisamos de enviar informação de ambos os pcs (seria muito ineficiente fazer por polling).

CONCLUSIONS

Em conclusão, achamos que este projeto foi bastante interessante de desenvolver. Perceber como todos os periféricos funcionam e analisar as informações diretamente deles é bastante gratificante e dá-nos um bom conhecimento para o futuro.

Numa perspetiva de melhoramento futuro, consideramos que no início da cadeira poderia existir mais apoio aos alunos, no que se trata de entrar no esquema da disciplina e perceber bem os interrupts e as subscrições.

Também achamos, no entanto, que não houve informação suficiente para desenvolver certas coisas. Muitas vezes nós sabemos o que temos de fazer mas não sabemos como o fazer. Isto é algo bastante frustrante visto que muitas vezes essas informações ou não se encontram ou estão explicadas de uma forma pouco esclarecedora.

Apesar de tudo, ambos vemos este projeto como uma mais valia para o nosso futuro.