



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

DAEDALUS MAZE

Laboratório de Computadores

2015/2016

T2G03

José Aleixo Cruz – up201403526@fe.up.pt

Nuno Miguel Cardoso Freitas – up201404739@fe.up.pt

Conteúdo

1.	Introdução	2
2.	Instruções de instalação	2
3.	Instruções de utilização	2
4.	Estado do projeto	5
5.	Organização/Estrutura do código	6
	4.1. proj.C	6
	4.2 Daedalus	6
	4.3 Timer.....	6
	4.4 Maze	6
	4.5 Bitmap	6
	4.6 Keyboard.....	6
	4.7. RTC e Date	6
	4.8 Video_gr	6
6.	Gráfico de chamada de funções	7
7.	Detalhes de implementação	8
	7.1. Desenhar o labirinto	8
	7.2. Desenhar o personagem	8
	7.3. Detetar colisões	8
	7.4. Criar armadilhas	8
	7.5. Controlar periféricos/dispositivos.....	8
	7.6. Conjugação toda a informação.....	9
8.	Conclusões.....	9

1. Introdução

O principal objetivo deste projeto é desenvolver, através da utilização das propriedades do sistema operativo MINIX e linguagem C, um videojogo cujo objetivo é chegar ao fim de um labirinto. Para isso, o grupo recorreu a os conceitos aprendidos durante o semestre incluindo a utilização do teclado, modo gráfico, timer e RTC.

A realização deste projeto permitiu aos elementos do grupo o aprofundamento dos conhecimentos da matéria leccionada durante a totalidade deste semestre e aplicá-los de uma forma mais estruturada do que nos laboratórios práticos. A criação deste projeto construiu também novas bases que irão permitir que no futuro, a construção de *software* seja feita de uma maneira mais eficiente por todos os elementos deste grupo.

2. Instruções de instalação

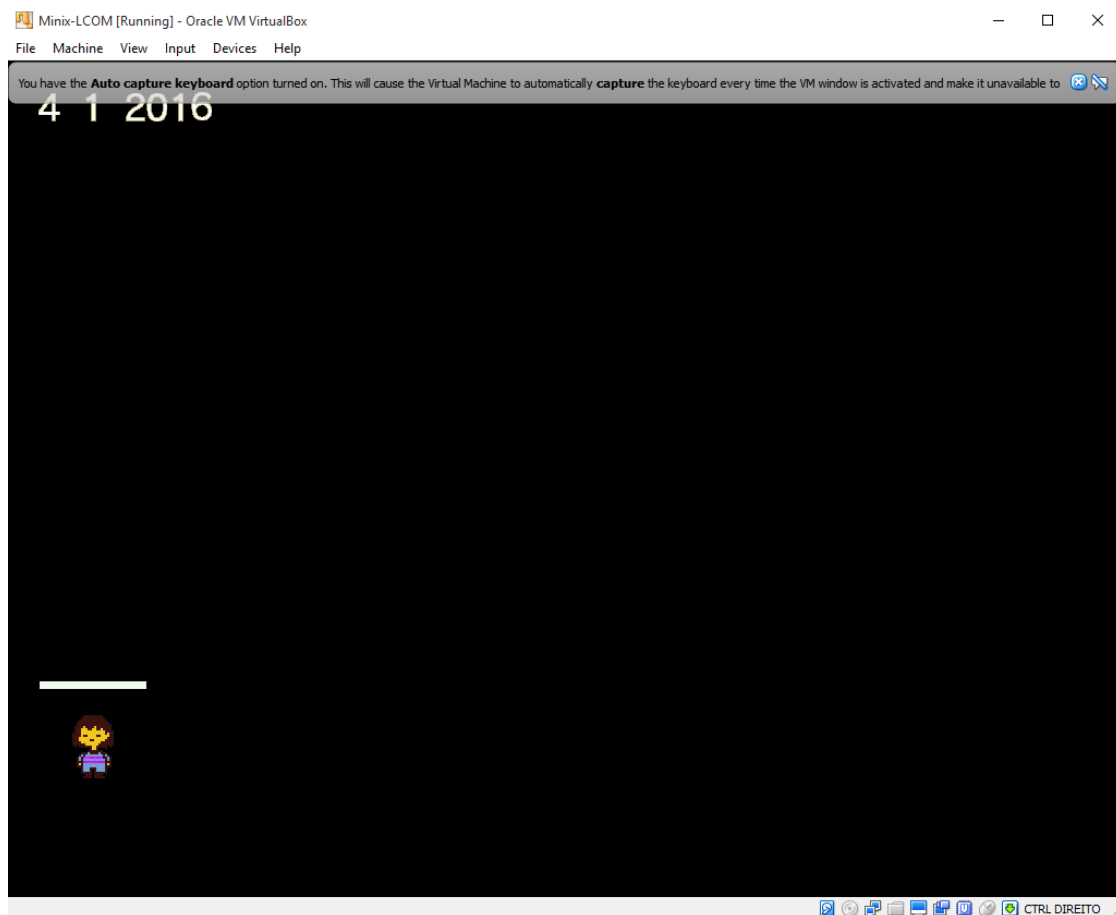
Para instalar o programa, basta navegar até ao diretório que contém o código fonte (lcom1516-t2g03\proj\code) e, em modo *root* executar os seguintes *scripts* por esta ordem: sh install.sh, sh compile.sh, sh run.sh.

3. Instruções de utilização

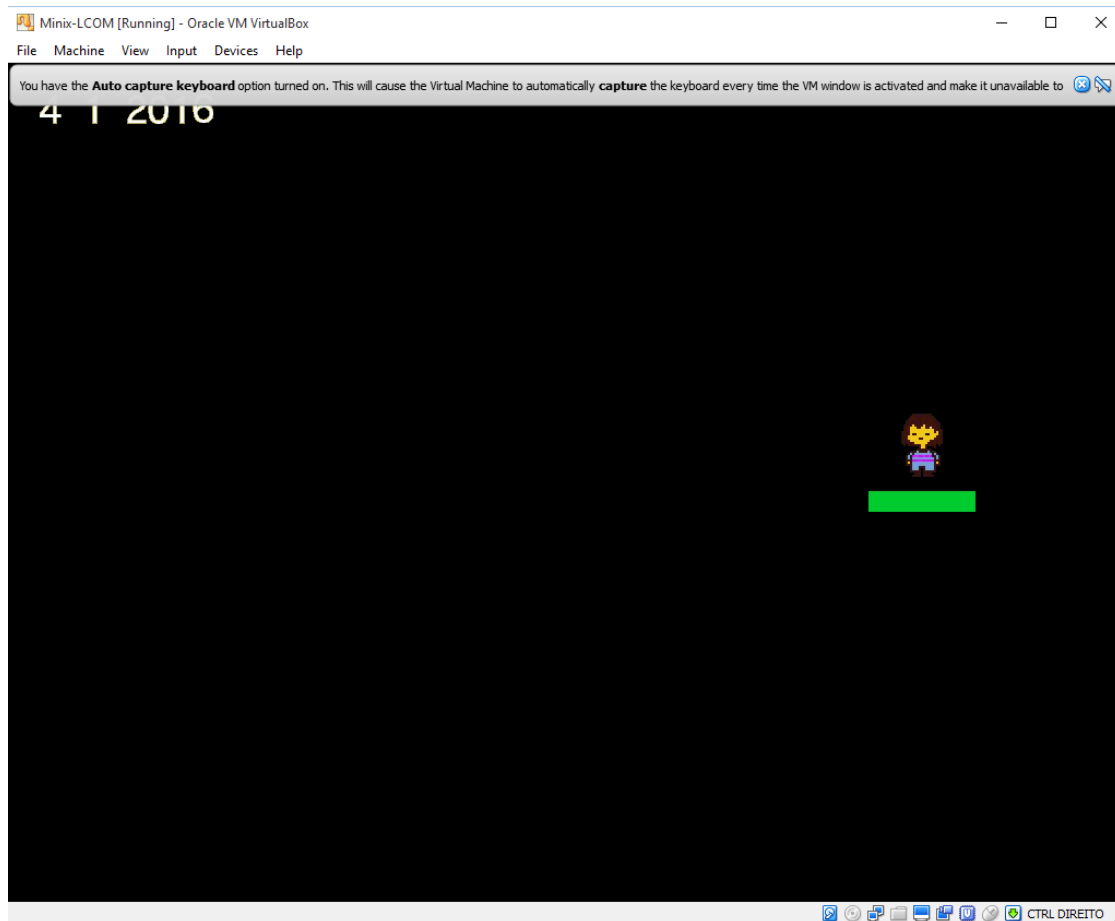
Depois do início do programa o utilizador vai ver um menu inicial, indicando que o programa começou corretamente. Este está demonstrado abaixo.



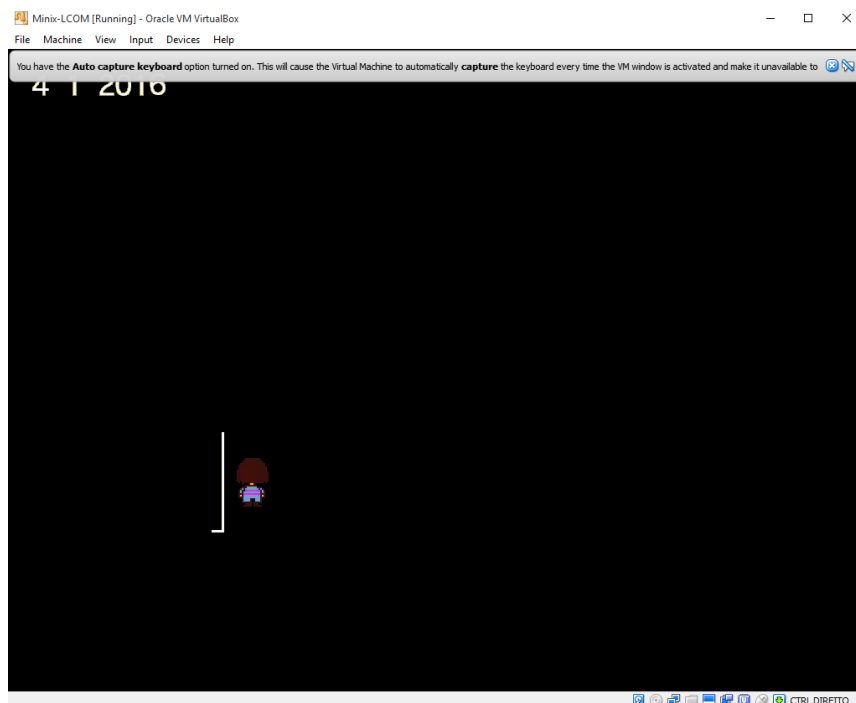
Depois o programa automaticamente irá começar o jogo. O jogador poderá observar a sua personagem, tudo num quadrado à sua volta e a data do dia em que está a jogar, exibida no canto superior esquerdo



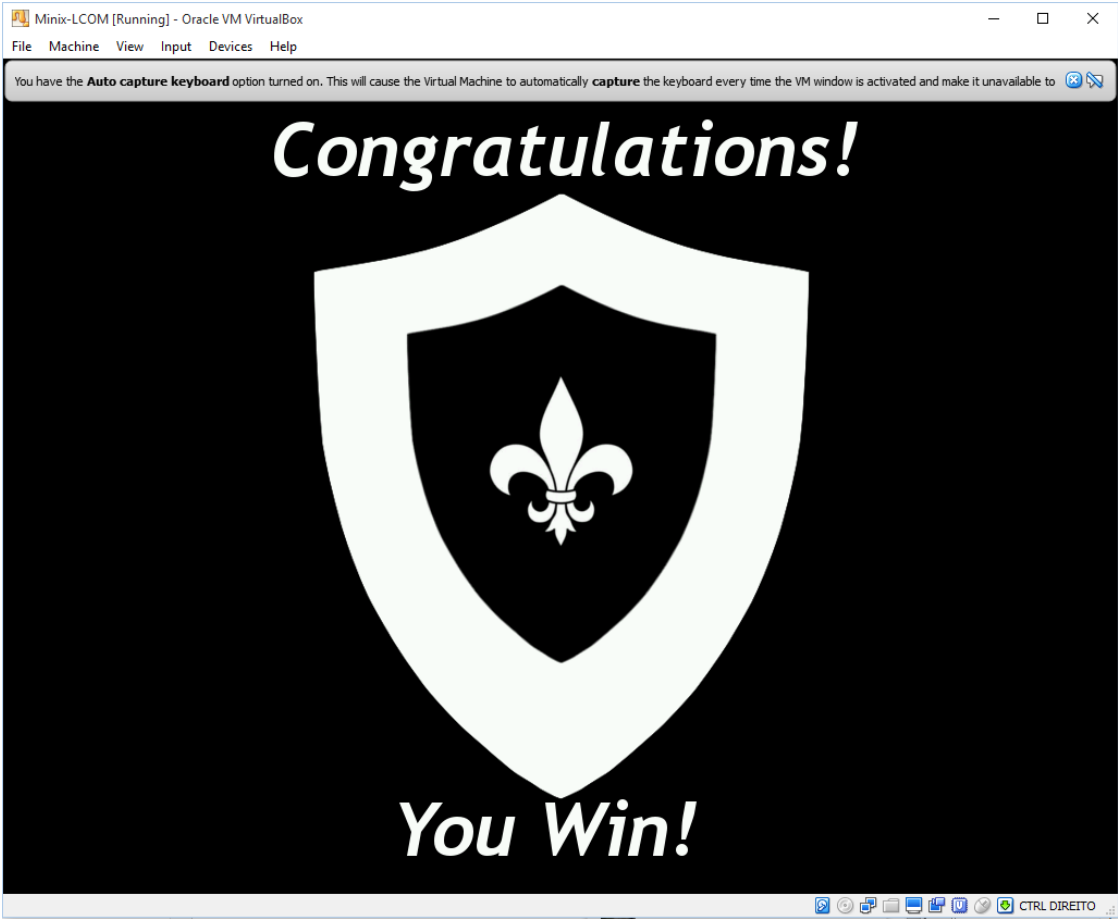
Utilizando as teclas do teclado "W", "A", "S", "D" o jogador poderá mover a personagem pelo labirinto pelo que não deve tocar nas paredes, o que faz com que tenha de recomeçar o labirinto mais uma vez. Para alcançar vitória o utilizador tem de encontrar a zona verde do labirinto, como demonstrado abaixo.



Tendo atingido a zona de vitória o jogador é automaticamente levado para uma imagem do menu inicial para se preparar para o próximo labirinto. Este tem paredes mais finas, pelo que a dificuldade aumenta pois navegar neste labirinto é mais difícil do que no anterior, mas o objetivo é o mesmo.



Encontrando de novo a zona verde, o jogador vence o jogo e é mostrado um ecrã de vitória.



4. Estado do projeto

Dispositivo	Função	Interrupções ou <i>polling</i> ?
Timer (temporizador)	Controlar a <i>frame-rate</i>	Interrupções
Teclado	Comandar o personagem	Interrupções
<i>Real-Time Clock</i> (RTC)	Demonstração de data	Interrupções
Placa Gráfica	Mostar os estados de jogo	Polling

- Timer – Permite controlar a frame-rate do jogo.
- Teclado – Permite mover a personagem pelo labirinto, podendo assim levá-la à vitória
- Real Time Clock – Deteta a data e utilizando bitmaps, o programa mostra a no canto superior esquerdo.
- Placa Gráfica – Utilizada para conseguir todos os aspectos gráficos do jogo.

5. Organização/Estrutura do código

José Aleixo foi o membro do grupo encarregado pela realização dos módulos proj, Timer, Maze e Keyboard, e nesses módulos o membro Nuno Freitas ficou responsável por ajudar o José em tudo o necessário e testar o código para que não haja erros no código. O Nuno ficou encarregado de desenvolver os módulos Date e RTC além de criar a maior parte dos aspetos gráficos do jogo. Os outros módulos, sendo estes maze, Daedalus Video_gr e Bitmap foram criados a partir de uma colaboração dos dois membros do grupo, permitindo assim um desenvolvimento de código mais eficiente e rápido.

4.1. proj.C

Este módulo é um dos mais importantes no projeto sendo responsável por iniciar o programa, chamando Daedalus.

4.2 Daedalus

Daedalus é o módulo principal do projeto, sendo o responsável pela junção e coordenação de todos os outros módulos, permitindo a função plena do projeto.

4.3 Timer

Utilizado para controlar a *framerate* do jogo.

4.4 Maze

Módulo encarregado de desenhar o labirinto.

4.5 Bitmap

Módulo que lê e mostra bitmaps que tenham 16 bits de profundidade de cor. Este módulo foi desenvolvido pelo Henrique Ferrolho, que também é estudante do MIEIC. Cremos que se terá baseado no código presente na seguinte página de fórum:

<http://forums.fedoraforum.org/archive/index.php/t-171389.html> .

4.6 Keyboard

O módulo keyboard tem o objetivo de detetar as teclas que permitem o movimento da personagem.

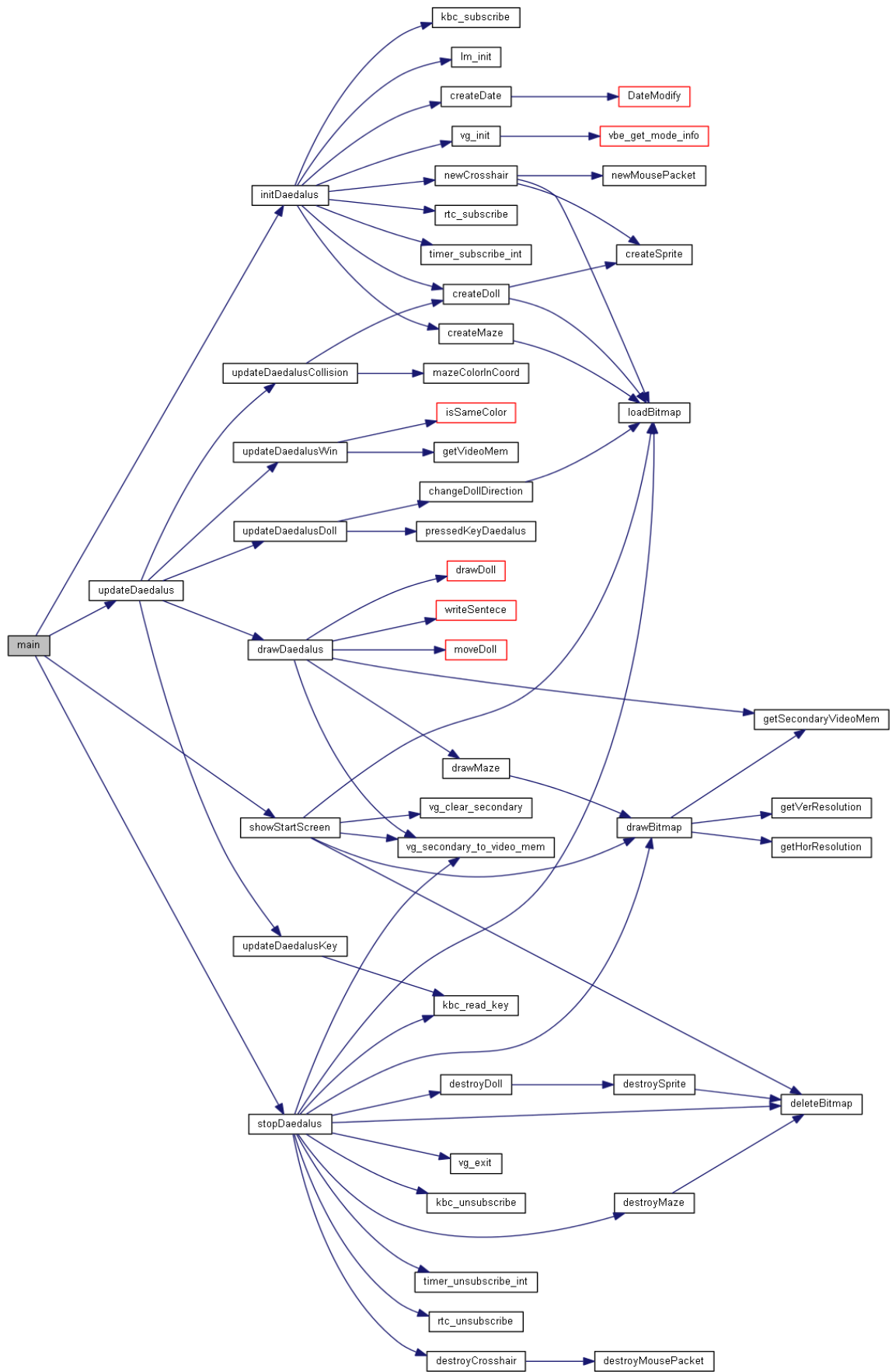
4.7. RTC e Date

Módulo que permite a atualização e obtenção da data, utilizados na sua demonstração no canto superior esquerdo.

4.8 Video_gr

Módulo que permite a utilização do modo gráfico. Também permite a obtenção de informações que ajudam na criação de colisões e criação de *transparência* nos bitmaps.

6. Gráfico de chamada de funções



7. Detalhes de implementação

7.1. Desenhar o labirinto

A ideia inicial para concretizar o labirinto era fazê-lo utilizando *sprites*. Criávamos um bloco-modelo, a partir do qual se construiria uma parede. No entanto isto revelou-se um obstáculo para a criação inicial de labirintos, pois não conseguíamos desenhar diretamente o labirinto que visualizamos.

Decidimos, então, usar *bitmaps* puros para representar o labirinto. Tudo o que basta é um editor de imagem, desenhar o labirinto com cores distintas e implementá-lo.

As cores que caracterizam cada parte do labirinto (parede, zona final, caminho) são identificadas pelo seu código RGB no *header* dos gráficos, e, obtendo a cor de um determinado pixel do mapa sabemos a que parte essa posição corresponde.

7.2. Desenhar o personagem

Tal como usámos bitmaps com o mapa, percebemos que usar bitmaps com a personagem também nos facilitaria o trabalho de desenhar a personagem.

No entanto, não nos apercebemos que usando bitmaps e não pixmaps, não tínhamos forma considerar uma determinada cor transparente. Desenvolvemos, para esse fim, uma função derivada da função que desenha bitmaps. Esta nova função ignoraria uma certa cor ao desenhar o bitmap, tornando essa cor transparente.

Para dar a ilusão de movimento ao personagem, incluímos bitmaps com o boneco a virar-se para 4 direções, e alternamos esses bitmaps à medida que ele se move.

7.3. Detetar colisões

Visto que tínhamos acesso às cores do bitmap do labirinto e àquilo que representavam, as nossas colisões, em vez de se basearem em sobreposição de objetos, baseiam-se em sobreposição de cores. Sempre o branco que representa as paredes do labirinto era sobreposto pelas cores do personagem, sabíamos que existia uma colisão.

7.4. Criar armadilhas

Um dos pilares da nossa implementação seria criar “armadilhas” que o jogador enfrentaria à medida que percorria o labirinto. Uma das ferramentas indispensáveis para isso era o rato. Apesar de termos conseguido usar e obter informação do rato em modo de texto, quando tentámos implementar o rato a par com o jogo, este não era detetado, por causa de um erro de escrita no KBC que acabámos por não descobrir. Desta forma, o nosso jogo está desprovido de armadilhas.

7.5. Controlar periféricos/dispositivos

Como não foi possível implementar o rato, só é necessário controlar o teclado, o temporizador e o RTC.

Para o teclado e para o temporizador baseámo-nos em funções desenvolvidas nos laboratórios das aulas práticas. Para o RTC, usámos o conhecimento divulgado pelo professor nas aulas teóricas.

A ideia original para implementar o RTC era gerar um ficheiro com as pontuações mais altas de cada jogador, mas devido a conflitos que essa funcionalidade causava no nosso código, usámos o RTC para mostrar a data (dia, mês e ano) no topo do ecrã.

7.6. Conjugação da informação

Aquilo que conjugação a informação obtida por todos os módulos é o objeto da estrutura “Daedalus”, que é criado no início do programa. Nessa estrutura, criámos variáveis que armazenariam a informação de cada dispositivo quando este a enviasse. No fundo, este objeto representa o estado em que o jogo se encontra a cada instante.

8. Conclusões

Apesar de uma cadeira com um grau de dificuldade elevado, não temos muitos pontos a criticar, visto que as nossas dúvidas foram sempre retiradas com paciência. Além disso, Emails que foram enviados aos professores foram respondidos com rapidez e eficácia. Em geral sentimos que houve um ensino eficaz da cadeira.

Talvez a nossa única crítica será relacionada com o nosso horário. O facto de termos a única aula por semana no dia anterior à única tarde que temos sem aulas de LCOM, fez com que as práticas coincidissem muito frequentemente com o dia anterior a muitas avaliações, acabando por dificultar o estudo.

Os membros do grupo ambos contribuíram 50% deste relatório.