

1.MINIX GOLF

O minix golf é um jogo em que o jogador tem de tentar introduzir a bola no buraco de forma a não tocar nas paredes, com o auxílio do rato, em que a cada toque a bola segue uma direção perpendicular à direção a que estava.

Conteúdo

1.MINIX GOLF	1
1.1 MENU.....	2
1.2 INSTRUCTIONS.....	3
1.3 GAME.....	4
2. I/O dispositivos	5
2.1 Uso dos dispositivos	5
2.2 Detalhes na utilização dos dispositivos	5
3.Organização do código/estrutura	6
3.1 timer.c – 3%.....	6
3.2 mouse.c – 6%	6
3.3 – KBC_mouse.c - 2%	6
3.4 – keyboard.c -5%	7
3.5 – KBC.c - 2%.....	7
3.6 – video.c -7%	7
3.7 -utils.c – 1%.....	7
3.8 menu.c – 11%.....	7
3.9 instructions.c -11%	8
3.10 game.c – 36%	8
3.11 sprites.c – 5%	8
3.12 main.c -11%.....	8
4 Detalhes da implementação	9
5 Conclusões.....	9

1.1 MENU

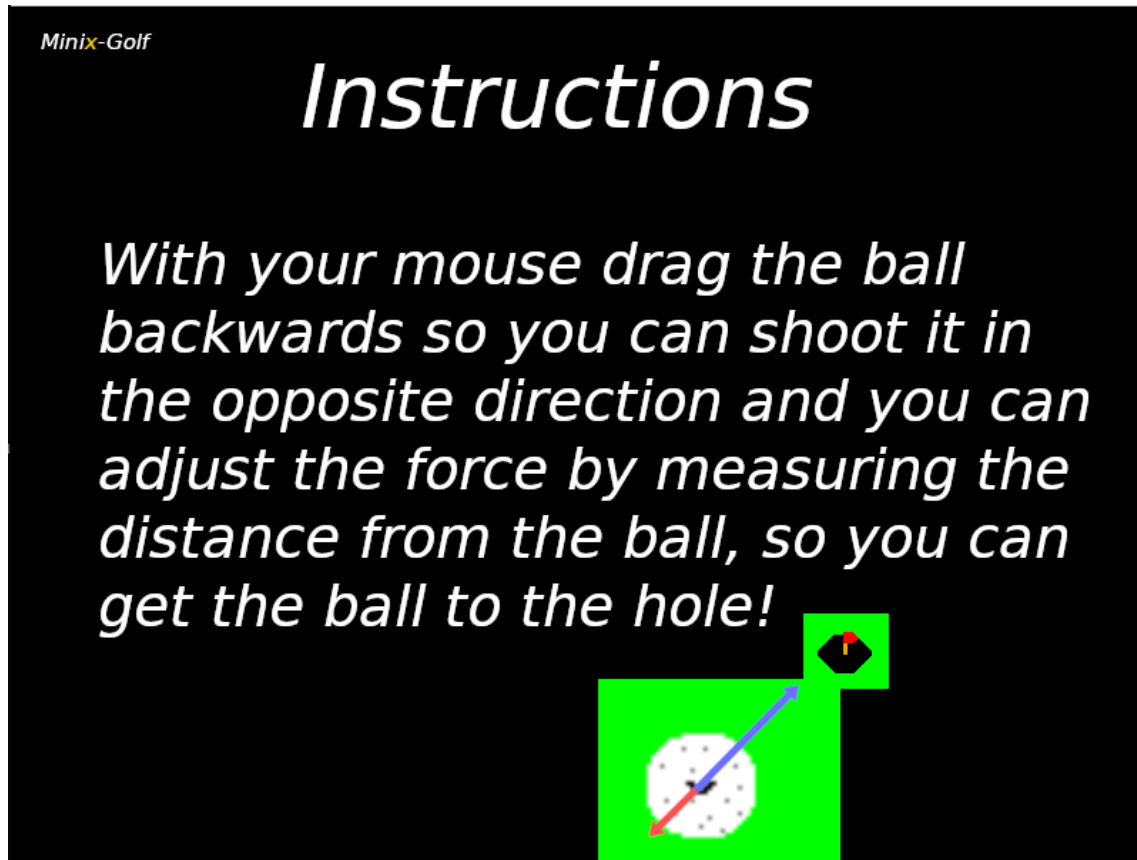
Esta é a nossa página inicial. Nesta mesma, o utilizador, com recurso ao teclado, pode escolher entre 3 opções:

- 1. Entrar dentro do jogo;
- 2. Ver as instruções do jogo;
- 3. Sair;



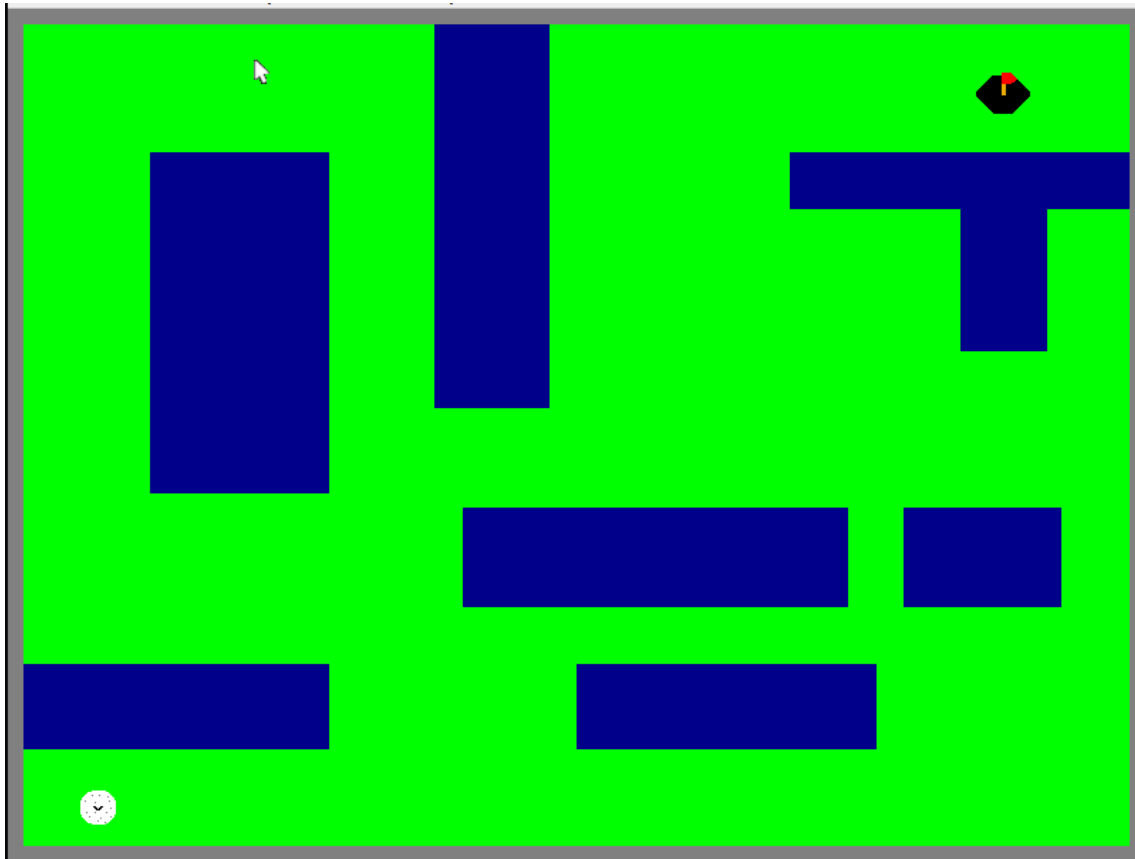
1.2 INSTRUCTIONS

Nesta página, o utilizador pode ver as instruções do jogo, em que lhe é explicado como interagir com o rato de forma a “bater” na bola. Quando o mesmo tiver lido, pode clicar na tecla ESC para voltar ao menu.



1.3 GAME

Aqui é onde a ação acontece. A bola estará numa determinada posição e, tal com diz nas instruções, deve arrastar a bola na direção oposta a que pretende que a bola vá, isto tudo com o menor número de tacadas possível. Nesta página, o utilizador também poderá premir o botão ESC, caso queira voltar ao menu principal. A outra maneira de voltar ao menu é completando o jogo, ou seja, introduzir a bola no buraco.



2. I/O dispositivos

2.1 Uso dos dispositivos

Dispositivos	Implementado?	Interrupções	Uso
timer	Sim	Sim	Usado para controlar a frame-work do jogo
video	Sim	Não	Usado para desenhar o conteúdo de cada mode, bem como para verificar se a bola chega ao buraco
keyboard	Sim	Sim	Usado para selecionar o que quero do menu, bem como sair da página de jogo e da página de instruções
mouse	Sim	Sim	Usado para mudar a direção da bola
RTC	Não	-	-
Serial Port	Não	-	-

2.2 Detalhes na utilização dos dispositivos

2.2.1 Timer

O timer no nosso projeto é utilizado para controlar a frame-work do programa, que, no nosso caso, está a 30 frames por segundo. Também é importante referir que o timer utilizado neste programa é o timer 0. Todas as implementações do timer estão ficheiro timer.c, na pasta timer.

2.2.2 keyboard

O teclado no nosso jogo é usado para mudar os modos de jogo. Isto é, se o jogador estiver no modo menu, poderá utilizar a tecla 1 para ir para o modo game, a tecla 2 para ir para o modo instructions, ou tecla 3 para o modo exit, que fechará o jogo. Tanto na página de instruções como na página do jogo, o user poderá premir o botão ESC para voltar ao menu principal. Todas as implementações do teclado estão no ficheiro keyboard.c e KBC.c, na pasta keyboard

2.2.3 Rato

O rato no nosso jogo será exclusivamente usado para o modo de jogo, para dirigir a bola para o buraco, sem tocar nas paredes. Todas as implementações do rato estão no ficheiro mouse.c e KBC_mouse.c, na pasta mouse.

2.2.4 Video

A placa de video no nosso jogo será usada para desenhar a interface ao utilizador, quer seja do menu, das instruções, ou do game. Neste último game, usufruímos de 2 buffers. Quando no menu pressionamos a tecla para jogar, aí é desenhado para um buffer os elementos estáticos do mapa, como as paredes, o buraco e o relvado. Depois, com recurso às sprites, desenhamos o rato e a bola na posição inicial onde estão definidas, para um frame buffer, onde este é copiado para a placa de vídeo. A partir daí, é só copiar o que está no auxiliar buffer para o frame buffer, e depois atualizar a posição da bola e do rato.

3. Organização do código/estrutura

3.1 timer.c – 3%

Este código é reutilizado do lab2. Neste mesmo ficheiro são implementadas as interrupções do timer, bem como a sua configuração.

Contribuidores:

- Bernardo Pinto
- João Sousa
- Carlos Madaleno

3.2 mouse.c – 6%

Este código é reutilizado do lab4. Neste mesmo ficheiro são implementadas as interrupções do mouse, bem como a sua configuração, e a leitura dos 3 bytes de controlo a cada interrupção, através da chamada para o KBC_mouse.c

Contribuidores:

- Bernardo Pinto
- João Sousa
- Carlos Madaleno

3.3 – KBC_mouse.c - 2%

Este código é reutilizado do lab4. Neste mesmo ficheiro são lidos e escritos os bytes de controlo do/para o buffer.

Contribuidores:

- Bernardo Pinto
- João Sousa
- Carlos Madaleno

3.4 – keyboard.c-5%

Este código é reutilizado do lab3. Neste mesmo ficheiro são implementadas as interrupções do keyboard, bem como a sua configuração, e a leitura da control_word a cada interrupção, através da chamada para o KBC.c.

Contribuidores:

- Bernardo Pinto
- João Sousa
- Carlos Madaleno

3.5 – KBC.c - 2%

Este código é reutilizado do lab3. Neste mesmo ficheiro são lidos e escritos os bytes de controlo do/para o buffer.

Contribuidores:

- Bernardo Pinto
- João Sousa
- Carlos Madaleno

3.6 – video.c -7%

Este código é reutilizado do lab5. Neste mesmo ficheiro é possível configurar o minix para modo gráfico, bem como desenhar no ecrã, recorrendo ao frame_buffer, configurado também neste mesmo ficheiro.

Contribuidores:

- Bernardo Pinto
- João Sousa
- Carlos Madaleno

3.7-utils.c – 1%

Este código é reutilizado do lab2. Neste mesmo ficheiro são implementadas funções que são reutilizadas pelos dispositivos acima referidos.

Contribuidores:

- Bernardo Pinto
- João Sousa
- Carlos Madaleno

3.8 menu.c – 11%

Ficheiro onde está implementada a interface do menu, bem como as funcionalidades do teclado, de modo a poder mudar de modo de jogo.

Contribuidores:

- Bernardo Pinto
- João Sousa
- Carlos Madaleno

3.9 instructions.c-11%

Ficheiro onde está implementada a interface das instruções, bem como as funcionalidades do teclado, de modo a poder mudar de modo de jogo.

Contribuidores:

- Bernardo Pinto
- João Sousa
- Carlos Madaleno

3.10 game.c – 36%

Ficheiro onde está implementada a interface do jogo, bem como as funcionalidades do teclado, de modo a poder mudar de modo de jogo, e também do rato, de modo a poder dar uma tacada na bola.

Contribuidores:

- Bernardo Pinto
- João Sousa
- Carlos Madaleno

3.11 sprites.c – 5%

Ficheiro onde estão implementadas os “desenhos” do jogo, como a bola, o mapa, o menu e as instruções.

Contribuidores:

- Bernardo Pinto
- João Sousa
- Carlos Madaleno

3.12 main.c-11%

Ficheiro onde está implementado o main loop do jogo, em que se trata de qualquer interrupção.

Contribuidores:

- Bernardo Pinto
- João Sousa
- Carlos Madaleno

4 Detalhes da implementação

Como referido acima, usamos 4 modos de jogo diferente: o MENU_MODE, o INSTRUCTIONS_MODE, O GAME_MODE e o EXIT_MODE, sendo que este último termina o nosso programa, voltando de novo ao modo de texto. Esta troca de modos acontece com as interrupções do teclado, à exceção do GAME_MODE, onde se pode voltar para o MENU_MODE se terminarmos o jogo, ou seja, se metermos a bola no buraco.

Também de modo a melhorar a eficiência do nosso projeto, utilizamos um buffer extra para aquilo de que tínhamos imaginado, de modo a melhorar a eficiência do mesmo. Para este buffer, desenhámos 1 única vez os elementos estáticos do jogo, quando saímos do MENU_MODE e vamos para o GAME_MODE, como as paredes, o buraco e o relevo, e guardamos lá para que depois só se tenha que copiar este buffer para o main_buffer, de modo que só tenhamos de desenhar a cada interrupção uma nova posição da bola e do rato, melhorando assim a performance do minix.

Outro detalhe da implementação tem a ver com as colisões da bola com a parede, em que, a cada frame, vemos se a cor da bola não “embate” numa cor azul ou cinzenta, e, caso “embata”, existem 2 possibilidades. Caso a colisão com a parede tenha sido com a parte vertical de uma parede, o x da bola passa a ser o contrário do mesmo; caso a colisão seja numa parte horizontal, o y da bola passa a ser o contrário do mesmo.

5 Conclusões

Como referido acima, usamos o rato para arrastar e fazer com que a bola vá numa determinada direção. O problema que nos causou isto foi de o rato estar demasiado lento e não saber ao certo se esta era a velocidade da bola que nós queríamos, e tivemos de usar um triple buffer para isso, consumindo mais memória do computador, mas desta maneira o desenho do rato é fluído.

Infelizmente, não conseguimos implementar o RTC e a serial port, tal como estava previsto, bem como as colisões com os objetos, por não termos planeado corretamente o nosso tempo necessário para este projeto, bem como a desistência de um elemento do nosso grupo. Embora não tenhamos feito isso, noutras funcionalidades, nomeadamente o rato, fizemo-lo de forma mais complexa, com o arrastar da bola para a direção pretendida e medir a distância entre o ponto em que agarramos e o ponto onde soltamos a bola.

Se pudéssemos desenvolver o projeto para um futuro próximo, gostaríamos que fosse possível que mais jogadores jogassem, através da serial port, bem como a implementação de mais mapas, para melhorar a experiência do utilizador.

Mesmo assim, aprendemos bastante com este trabalho, ao utilizar tantos dispositivos num só programa, e a conseguir coordená-los de forma efetiva.