

Faculdade de Engenharia da Universidade do Porto



Universidade do Porto

FEUP Faculdade de
Engenharia

Laboratório de Computadores
2022/23

T04G5

up202005832 - Luís Sousa
up202006479 - Rúben Esteves
up2021087831 - Tiago Gouveia
up202108878 - Rui Silveira

Índice

Instruções para o Utilizador.....	4
Menu.....	5
Personagem principal.....	6
Durante uma partida.....	7
Fim do jogo.....	8
Estado do Projeto.....	9
Timer.....	10
Keyboard.....	10
Mouse.....	10
Graphics Card.....	10
RTC (real time clock).....	10
Organização e estrutura de código.....	11
main.c - 45%.....	11
i8254.h - 2%.....	11
utils.c - 2%.....	11
timer.c - 5%.....	11
keyboard.c - 8%.....	11
mouse.c - 17%.....	12
vg.c - 18%.....	12
rtc.c - 3%.....	12
Conclusões.....	14
Sobre o jogo.....	14
Durante o desenvolvimento.....	15

Instruções para o Utilizador

O Minix Surfers é um emocionante jogo *multiplayer*, onde dois jogadores se envolvem em uma divertida disputa. Um jogador assume o controle da personagem principal, movendo-a habilidosamente para desviar dos obstáculos deixados pelo segundo jogador, representado pelo cursor.



Figura 1 - menu inicial

Menu

Ao iniciar o jogo, os jogadores são recebidos por um menu envolvente, que apresenta todas as opções necessárias para iniciar a partida, em conjunto com instruções e informações de jogo para ambos os jogadores, garantindo uma experiência fluida e imersiva desde o início.

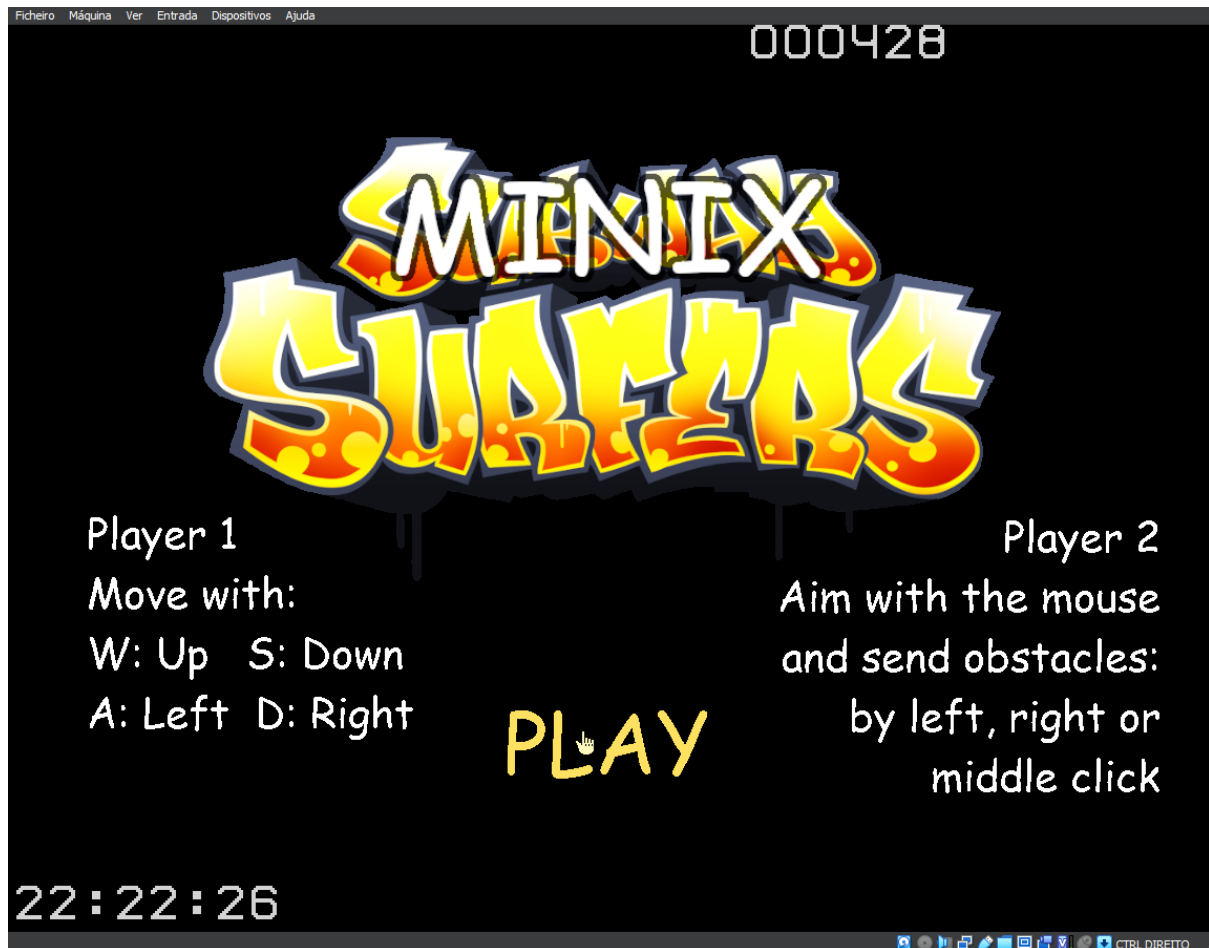


Figura 2 - tela inicial com os controles

Personagem principal

No que diz respeito aos gráficos, a personagem principal do Minix Surfers destaca-se como um elemento carismático e animado. A sua aparência e animações cuidadosamente projetadas criam uma personagem cativante, capaz de estabelecer uma ligação emocional com os jogadores. Através desse vínculo, os jogadores sentem-se mais envolvidos e imersos na experiência do jogo, aumentando a diversão dos mesmos.



Figura 3 - partida do jogo

Durante uma partida

No que diz respeito ao *design*, o *background* é alterado consoante a altura do dia, oferecendo uma variedade visual encantadora. Os ambientes apresentam planos de fundo distintos para representar os períodos de dia e noite. Essa atenção aos detalhes é possível graças à implementação do RTC (*Real-Time Clock*), que permite a sincronização precisa do tempo no jogo.

Na interface do jogo, a hora exata do dia é exibida com clareza no canto inferior esquerdo da tela, proporcionando uma imersão ainda maior para os jogadores. Além disso, no canto superior direito, o *score* da partida atual é exibido, juntamente com o *highscore*, posicionado à esquerda. Essa visualização facilita o acompanhamento do desempenho e a busca por superar recordes pessoais.



Figura 4 - layout de uma partida

Fim do jogo

Quando a personagem principal é capturada por um dos obstáculos ao longo do jogo, o primeiro jogador sofre a derrota. No entanto, essa derrota é acompanhada de uma imagem memorável protagonizada pela famosa personagem. Essa abordagem adiciona um toque de humor e entretenimento ao jogo, aliviando a frustração da derrota e mantendo um ambiente descontraído e divertido. Essa combinação da derrota e da imagem envolvente permite que os jogadores aproveitem o jogo mesmo em momentos de desafio, incentivando-os a persistir e tentar novamente para alcançar melhores resultados.

Aquando a aparição do *Game Over* a escarlate no ecrã, basta clicar na tecla R para recomençar.



Figura 5 - derrota do jogador um

Estado do Projeto

Dispositivo	Função	Interrupção
Timer	Conta os frames e os pontos	Sim
Keyboard	Movimento do jogador 1	Sim
Mouse	Movimento do jogador 2	Sim
Graphics Card	Desenhos dos menus, obstáculos, personagem e plano de fundo.	N/A
RTC	Visualizar as horas e a altura do dia (dia/noite).	Não

Timer

O *Timer* é utilizado no código para contar os frames e os pontos. Os obstáculos aumentam a sua velocidade quanto mais tempo durar o jogo.

Keyboard

O *Keyboard* é usado no jogo para o primeiro jogador movimentar o boneco, esquivando-se assim dos obstáculos dispensados pelo adversário.

Este periférico está restringido à exclusiva utilização das teclas W, A, S e D, para, respetivamente, a personagem andar para cima, esquerda, baixo e direita, à tecla ESC (Escape) para fechar o jogo, e à tecla R para reiniciar o jogo após um “Game Over”.

Mouse

O *Mouse* é utilizado no programa para o segundo jogador controlar os obstáculos que irão atrapalhar o jogador um. Para isso, ele aponta com o cursor do rato onde quer que a barreira apareça, escolhendo entre três (utilizando o botão esquerdo, o direito e o do meio do rato). Cada botão coloca um obstáculo diferente, sendo que o botão esquerdo coloca um obstáculo comprido e lento, o direito coloca um obstáculo alto e rápido, e o do meio coloca um obstáculo muito pequeno e muito rápido.

Graphics Card

A *Graphics Card* é utilizada para mostrar os gráficos do jogo, entre eles o plano de fundo, a personagem principal, os obstáculos e os menus, lendo os sprites guardados em imagens do formato *xpm*.

RTC (real time clock)

O *Real Time Clock* é usado na impressão das horas e na demonstração da fase do dia ao utilizador (sendo estas dia ou noite).

Organização e estrutura de código

main.c - 45%

Ficheiro principal do código, com a função `main` que contém o nosso game loop, assim como `saveHighScore` e `readHighScore`, para ler e guardar os highscores. Inclui todos os ficheiros mencionados abaixo e utiliza-os para compor as diferentes componentes do jogo.

i8254.h - 2%

Ficheiro com as macros usadas pelas diversas drivers.

utils.c - 2%

Ficheiro com as funções de utils para simplificar o left shift e right shift, junto com o `sys_inb`.

timer.c - 5%

Ficheiro com as funções para controlar o timer, nomeadamente:

- `timer_set_frequency` - altera a frequência do timer;
- `timer_subscribe_int` - subscreve os interrupts do timer;
- `timer_unsubscribe_int` - remove a subscrição dos interrupts;
- `timer_int_handler` - aumenta o counter do tempo;
- `timer_get_conf` - prepara o timer para ser lido e lê-o;
- `timer_display_conf` - dá display à informação do timer previamente recolhida;

keyboard.c - 8%

Ficheiro com as funções que recebem e processam inputs do keyboard, nomeadamente:

- `kbc_ih` - input handler do teclado, quando há um interrupt do tipo correto é chamado;
- `kbd_subscribe` - subscreve os interrupts do keyboard;
- `kbd_unsubscribe` - remove a subscrição dos interrupts;
- `kbd_status_reg_reader` - lê o status byte do keyboard;
- `kbd_output_buffer_reader` - verifica se o interrupt é do tipo pretendido e lê o buffer;

mouse.c - 17%

Ficheiro com as funções para processar os movimentos do mouse, assim como os cliques, nomeadamente:

- `mouse_subscribe` - subscrive os interrupts do mouse;
- `mouse_unsubscribe` - remove a subscrição dos interrupts;
- `mouse_ih` - input handler do mouse;
- `readBytes` - lê um byte do mouse;
- `toPacket` - tranforma o byte lido numa structure packet;
- `mouse_write` - envia um comando para o mouse;
- `get_KBD_stat` - lê o status do keyboard;
- `get_KBD_out` - lê o output do keyboard;
- `KBD_cmd` - envia um comando para o keyboard

vg.c - 18%

Ficheiro com as funções para processar a gráfica e desenhar no ecrã:

- `vg_init` - inicializa a gráfica no modo fornecido;
- `map_vram` - dá map à vram;
- `vg_draw_pixel` - desenha um pixel;
- `vg_draw_pixel_to_background` - desenha um pixel no background;
- `vg_draw_hline` - desenha uma linha;
- `vg_draw_rectangle` - desenha um retângulo;
- `print_xpm` - desenha uma imagem XPM no ecrã;
- `loadBackground` - carrega o background a partir de XPM;;
- `allocateBuffer` - aloca espaço na memória para o background e o buffer;
- `showBuffer` - copia o buffer para a vram;
- `clearBuffer` - mete todos os elementos do buffer a 0 (preto);
- `drawBackground` - desenha o background no buffer;
- `freeBuffer` - liberta o espaço do buffer da memória;

rtc.c - 3%

Ficheiro com as funções para ler do RTC

- `readRtc` - lê do RTC;
- `rtcUpdating` - verifica se o RTC está a dar update;
- `rtcBinary` - verifica se o RTC está em binário;
- `toBinary` - converte um número para binário;
- `rtcReadHours` - lê as horas no RTC;
- `rtcReadMinutes` - lê os minutos no RTC;
- `rtcReadSeconds` - lê os segundos no RTC;

Conclusões

Sobre o jogo

Em conclusão, o Minix Surfers é um jogo divertido e envolvente que proporciona uma experiência emocionante para os jogadores. A dinâmica de dois jogadores, em que um desvia dos obstáculos criados pelo outro, acrescenta um elemento estratégico e competitivo ao jogo.

O *design* do jogo é bem cuidado, com planos de fundo distintos para cada fase do dia, trazendo variedade visual à experiência de jogo. A personagem principal é cativante e animada, contribuindo para a diversão e proporcionando momentos engraçados quando é capturada pelos obstáculos.

Além disso, o Minix Surfers apresenta um sistema de pontuação que incentiva os jogadores a superarem seus próprios recordes. A exibição do *score* atual e do *highscore* no canto superior direito mantém os jogadores engajados e motivados a alcançarem pontuações cada vez melhores.

Em suma, o Minix Surfers é um jogo viciante e bem projetado, com elementos visuais atrativos, jogabilidade envolvente e um toque de competição entre jogadores. É uma opção divertida para os fãs de jogos de corrida infinita, oferecendo entretenimento garantido por longas sessões de jogo.

Durante o desenvolvimento

A nossa experiência ao escrever o código do jogo Minix Surfers foi desafiadora e gratificante ao mesmo tempo. Como desenvolvedores, enfrentamos várias dificuldades ao longo do processo, mas também aprendemos muito.

A atipicidade do sistema minix mostrou-se um desafio, não tínhamos experiência prévia a programar drivers de qualquer tipo, em especial de um sistema antigo como o mencionado. Foi um desafio que tivemos prazer em superar, estimulando a nossa vontade de trabalhar mais junto do *back end*.

Essa experiência também foi uma oportunidade de aplicar os conhecimentos adquiridos na disciplina. Ao trabalharmos juntos no código, pudemos consolidar conceitos de programação que aprendemos em sala de aula, fazendo uso das drivers criadas em aula, criando uma ligação extra entre as aulas práticas e o trabalho de grupo.

No final, a experiência de escrever o código do Minix Surfers foi recompensadora. Ver o jogo a ganhar vida, com sua jogabilidade fluida e elementos visuais atraentes, foi gratificante. Além disso, deixou-nos mais confiantes nas nossas habilidades de programação e motivados a explorar novos projetos no futuro.