Faculdade de Engenharia da Universidade do Porto

BrickBlitz

2023/2024

Laboratório de Computadores

Projeto Grupo 4 – Turma 18

Realizado por:

- Afonso Castro (up202208026)
- José Martins (up202204857)
- Pedro Santos (up202205900)
- Rodrigo de Sousa (up202205751)

Índice

1. Introdução

O nosso jogo é um *remake* do conhecido *Breakout* de 1976.

O contexto consiste em várias camadas de blocos posicionadas em fileiras na parte superior do ecrã que necessitam de ser destruídas pelo jogador antes que estas desçam e atinjam a secção inferior do ecrã. É dado ao jogador o controlo de uma prancha localizada na secção inferior com a capacidade de deslocamento horizontal, que este terá de utilizar para inicialmente lançar (tecla ENTER) e depois refletir os movimentos de uma bola que será usada para eliminar os blocos quando em contacto com estes.

O ângulo de lançamento inicial será escolhido pelo jogador através do rato, e o movimento da prancha é comandado através das teclas A e D.

Caso o jogador falhe na receção e reflexão da bola principal e deixe que esta passe os limites inferiores da tela, este perderá uma das 3 vidas que lhe são dadas.

Para fomentar um jogo mais dinâmico, ao destruir cada bloco existe uma probabilidade de ganhar um *powerup* que poderá ser uma bola extra que entra imediatamente em jogo (e que caso perdida, não afeta as vidas do jogador), ou então um míssil que quando disparado (tecla SPACE), voa verticalmente e poderá ter um rasto de destruição de até 6 blocos (misseis são acumuláveis).

O jogo acaba com a vitória, aquando da destruição de todos os blocos, ou com a derrota, aquando da perda das 3 vidas ou do sucesso dos blocos em chegar à secção inferior do ecrã.

2. Instruções para o Utilizador

(a) Menu Inicial

Quando o jogo é inicializado, aparece o seguinte menu inicial:

Figura 1 – Menu Inicial

Para iniciar uma partida do jogo, o utilizador deverá clicar ENTER.

Clicando ESC nesta fase, levará ao text-mode do MINIX.

É um simples SplashScreen com o logótipo do nosso jogo, com a palette das cores proeminentes do resto dos Sprites.

O jogador pode retornar a qualquer altura para este ecrã clicando na tecla ESC.

(b) Jogo

Os blocos aparecem e começam a descer após 30 segundos.

Quando o jogo começa, o jogador escolhe o ângulo de lançamento da bola com o rato (indicado pelo cursor vermelho e verde acima da prancha). Para confirmar o ângulo, este pressionará a tecla ENTER.

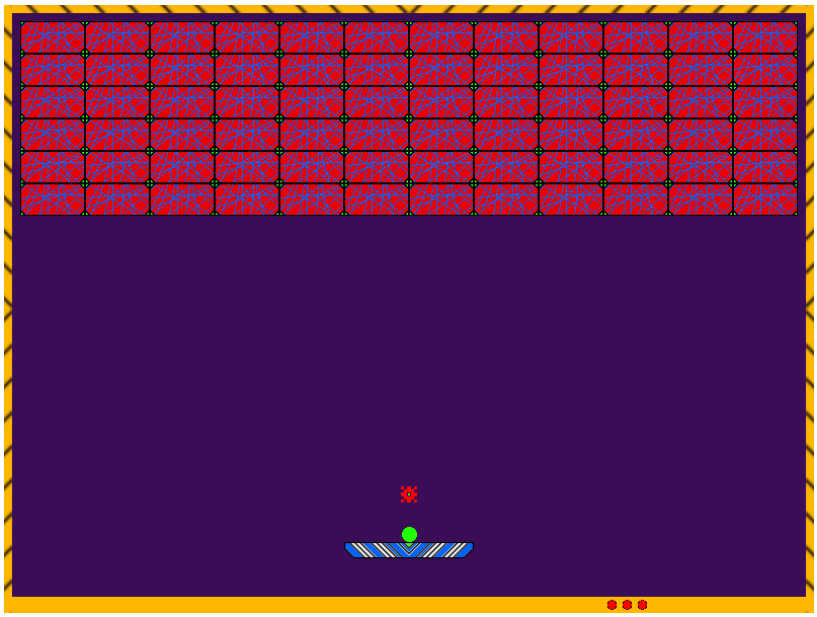

Figura 2 – Escolha do Ângulo de Lançamento

O jogador tem que então eliminar todos os blocos para atingir a vitória, utilizando a prancha para lançar e refletir a bola, e usando os *powerups* eficientemente (tal como explicado acima na secção da ***Introdução***).

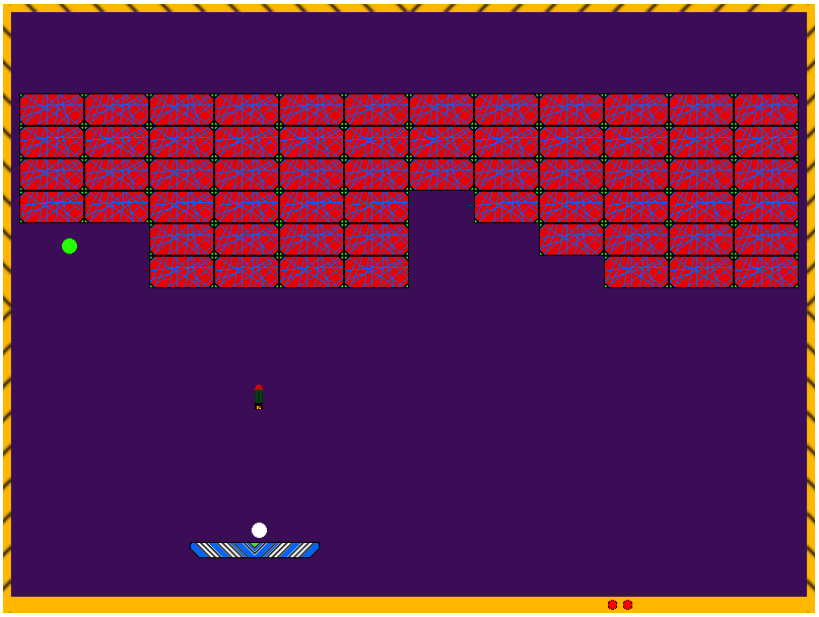

Figura 3 – Momento Após Lançamento do Míssil

As vidas restantes do jogador encontram se assinaladas na barra inferior (bolas vermelhas). 

Os misseis disponíveis para disparar também se encontrarão na mesma barra. 

Quando um bloco é destruído, existe 10% chance de ganhar uma bola extra (bola branca), ou 10% chance de ganhar um míssil.

Acertar nos cantos verdes dos blocos aumentará as hipóteses de destruir múltiplos blocos de uma só vez.

(c) Ecrã de Vitória

Após a vitória do jogador, o seguinte ecrã aparecerá, podendo o jogador clicar ENTER para voltar ao Menu Inicial, ou ESC para voltar ao text-mode do MINIX.

Figura 4 – Ecrã de Vitória feito no GIMP

(d) Ecrã de Derrota

Após a derrota do jogador, o seguinte ecrã aparecerá, podendo o jogador clicar ENTER para voltar ao Menu Inicial, ou ESC para voltar ao text-mode do MINIX.

Figura 5 – Ecrã de “Game Over” feito no GIMP

3. Estado do Projeto

Dispositivos Implementados

Dispositivo	***Funcionalidades***	***Interrupções***
Timer	Contagem dos frames e do tempo de inicialização da descida dos blocos.	Sim
KBD	Controlo da prancha. Funcionalidade do Menu. Lançamento da bola e dos mísseis.	Sim
Mouse	Escolha do ângulo de lançamento da bola.	Sim
Video Card	Desenho dos Sprites (XPMs).	Não

i. Timer

O Temporizador foi utilizado no modo 60hz na contagem dos frames e na gestão do tempo da inicialização da descida dos blocos.

Funções utilizadas:

- Utilização de interrupts na função *run()* do *game.c*
- Funções do ficheiro *timer.c*

ii. KBD

O Teclado é utilizado no modo interrupts na seleção das funcionalidades de menu, no controlo da prancha, confirmação de lançamento da bola, e na ativação dos mísseis.

Funções utilizadas:

- Utilização de interrupts na função *run()* do *game.c*, sendo processados na função *handle_keyboard()*
- Funções do ficheiro *keyboard.c*

iii. Mouse

O Rato é utilizado na seleção do ângulo de lançamento da bola principal. É apenas utilizado o movimento horizontal do rato (delta X).

Funções utilizadas:

- Utilização do struct *packet* do *lab4.h*, proveniente dos ficheiros de LCOM do MINIX.
- Processamento na função *handle_mouse()* do ficheiro *game.c*
- Funções do ficheiro *mouse.c*

iv. Video Card

A Gráfica é utilizada no modo 0x115 (*800x600 Direct color 24 (8:8:8)*) para desenhar todos os sprites e animações no ecrã.

Funções utilizadas:

- Single Buffering com as funções *draw_init()* e *draw_frame()* do ficheiro *game.c* e as funções do ficheiro *draw.c*
- Sprites no ficheiro *sprite.c* com a utilização da struct *Sprite* lecionada em aula.
- Deteção de colisões com as funções *move* do ficheiro *game.c*
- Utilização de ficheiros *XPM* na pasta xpm.

4. Organização do Código

Módulos Implementados

i. graphics.c

Neste Módulo estão incluídas as funções desenvolvidas no Lab 5 e são as que desenham todas as componentes gráficas no ecrã.

ii. kbc.c

Neste Módulo estão incluídas as funções desenvolvidas no Lab 3 e servem para ler os inputs do rato e do teclado.

iii. mouse.c

Neste Módulo estão incluídas as funções desenvolvidas no Lab 4 e permitem o programa receber o input do rato.

iv. timer.c

Neste Módulo estão incluídas as funções desenvolvidas no Lab 2 e têm as funcionalidades para preparar e correr o timer, para que seja possível contar os frames do programa.

v. utils.c

Neste Módulo estão apenas funções auxiliares de MSB, LSB e leitura de bytes, realizado no Lab 2.

vi. sprite.c

Neste Módulo estão incluídas todas as configurações para a criação e utilização de Sprites, classe presente nos slides teóricos.

vii. draw.c

Neste Módulo estão incluídas as funcionalidades de desenho para cada objeto, usando a sua posição e sprite.

viii. game.c

Neste Módulo estão presentes todas as funções que fazem parte das mecânicas do jogo, como as colisões, controlos e movimento de objetos, interrupt loops, retirado dos slides teóricos, e os respetivos handlers e os estados.

ix. objects.c

Neste Módulo estão incluídas todas as funções de criação de objetos e os seus structs.

Grafo de Chamada de Funções

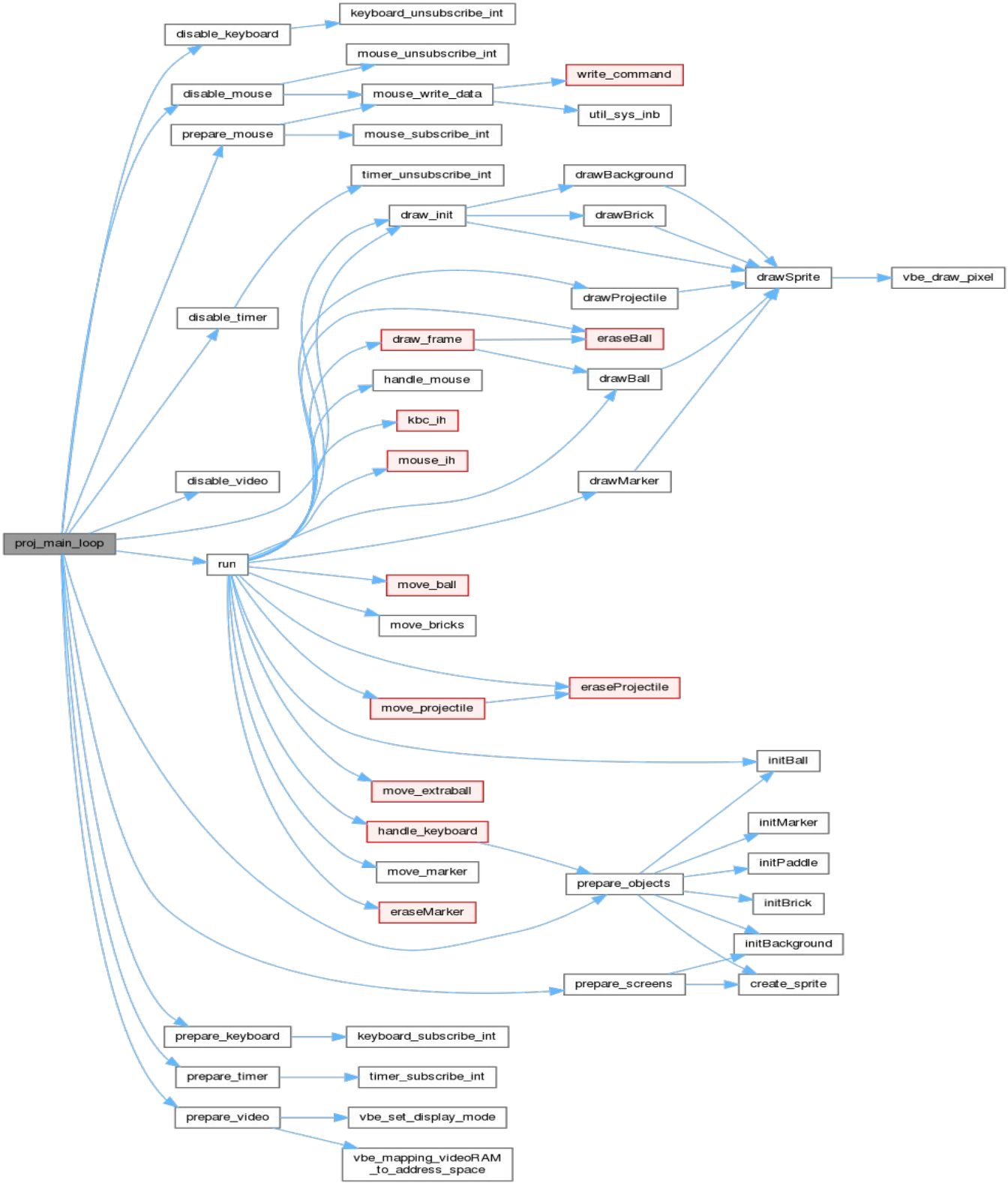

Figura 6 – Grafo Gerado pelo Doxygen

5. Detalhes de Implementação

Estados

Para controlar a mecânica de Menu, Ecrã de Vitória, Ecrã de Derrota e Jogo, foi implementado um struct State, que é alterado para mudar o estado do jogo. Assim, certos controlos e ações só funcionam em certos estados e dá fluídez ao jogo.

Deteção de Colisões

O nosso jogo funciona à base de colisões, logo foi necessário implementá-las para que a bola ao colidir com as paredes não saísse do jogo e refletisse.

Assim, cada vez que a bola bate numa parede lateral a sua velocidade em x é invertida e cada vez que bate no teto, a sua velocidade em y é invertida, da mesma forma. Para partir os blocos, também é necessário verificar as posições da bola e de cada um dos blocos para verificar se coincidem e, caso se confirme, definir esse bloco como “destroyed” e deixar de o desenhar, invertendo também a velocidade em y da bola.

Para o jogador conseguir controlar a bola, existem também colisões na prancha com a bola, em que dependendo da zona em que bate na prancha, altera a velocidade em x e em y para valores definidos, para proporcionar um elemento desafiador no jogo, e com as paredes laterais para que a prancha não saia da zona de jogo.

Os mísseis têm uma dinâmica mais complicada pois destroem mais blocos do que aquele que toca, por isso necessita de confirmar que esses blocos existem por perto e só aí definí-los como “destroyed”.

6. Conclusões

A realização deste projeto ajudou a cimentar melhor os conhecimentos nas interações inter-dispositivo e pessoa-máquina. Também foi mais satisfatório que a realização dos labs pois era mais simples de dar debug a algo que somos nós a visualizar a ideia. Não houve grandes dificuldades e o trabalho foi realizado sem o encontro de obstáculos, tendo sido bem planeado. Não implemetamos o RTC, nem Serial Port pois não conseguimos encontrar um sítio para os incluir no nosso jogo sem que parecesse aldrabado.