**Faculdade de Engenharia da Universidade do Porto**

****

**Projeto de Laboratórios de Computador**

**Jogo: Head Football**

**Licenciatura em Engenharia Informática e Computação**

**Grupo 2LEIC02\_G03**

**Realizado por:**

Guilherme Teixeira [**up20224875@up.pt**](mailto:up20224875@up.pt)

Guilherme Silva [**up202205298@up.pt**](mailto:up202205298@up.pt)Valentina Cadime [**up202206262@up.pt**](mailto:up202206262@up.pt)

**Índice**

[Introdução 2](#_Toc168075342)

[1. Instruções de Utilizador 3](#_Toc168075343)

[**1.1 Menu** 3](#_Toc168075344)

[**1.2 Modo de Treino (Um Jogador)** 3](#_Toc168075345)

[**1.3 Modo de Jogo (Multijogador)** 3](#_Toc168075346)

[2. Estado do Projeto 4](#_Toc168075347)

[**2.1 Funcionalidades Implementadas** 4](#_Toc168075348)

[**2.2 Dispositivos Utilizados** 5](#_Toc168075349)

[**2.2.1 Timer** 5](#_Toc168075350)

[**2.2.2 Teclado** 6](#_Toc168075351)

[**2.2.3 Rato** 6](#_Toc168075352)

[**2.2.4 Placa de vídeo (GPU)** 6](#_Toc168075353)

[**2.2.5 Real-Time Clock (RTC)** 6](#_Toc168075354)

[**2.2.6 Porta-série (UART)** 7](#_Toc168075355)

[3. Estrutura e Organização de Código 8](#_Toc168075356)

[4. Detalhes de Implementação 9](#_Toc168075357)

[5. Conclusão 10](#_Toc168075358)

Introdução

O **Head Football** é a nossa implementação em C do jogo de telemóvel [Head Soccer](https://play.google.com/store/apps/details?id=com.dnddream.headsoccer.android&hl=en&pli=1).

Consiste num jogo de futebol de vista lateral, para, no máximo, 2 pessoas, uma de cada lado, em frente à respetiva baliza, e o objetivo é marcar o maior número de golos possível em 5 minutos.

Para além disso, também existe um Modo de Treino, para apenas 1 jogador, que o utilizador pode jogar para se familiarizar com os controlos do jogo. Este modo de jogo termina automaticamente ao fim de 60 minutos, sendo possível sair antecipadamente através do menu de pausa.

1. Instruções de Utilizador

**1.1 Menu**

Ao ser iniciado, o programa mostra o título do jogo com as 4 opções que é possível selecionar. Esta seleção pode ser feita através com o teclado, com as setas para cima (↑) e para baixo (↓), sendo a escolha confirmada ao clicar no Enter (↵), ou com o rato.

**1.2 Modo de Treino (Um Jogador)**

Após selecionar a opção “Singleplayer”, o jogo começa. Estando presente apenas um jogador, o utilizador pode praticar os controlos do jogo e dominar a movimentação da personagem e da bola.

Os controlos do jogo são:

* Barra de espaços: O jogador salta;
* Seta para a esquerda (←): O jogador move-se para a esquerda;
* Seta para cima (→): O jogador move-se para a direita;
* Escape (Esc): O jogo para e surge o menu de pausa;

**1.3 Modo de Jogo (Multijogador)**

Ao selecionar a opção “Multiplayer”, o jogo

2. Estado do Projeto

**2.1 Funcionalidades Implementadas**

|  |  |  |
| --- | --- | --- |
| Funcionalidade | Dispositivo | Implementado? |
| Navegação entre menu e jogo | Placa Gráfica, Rato e Teclado | Sim |
| Rato com ponteiro | Placa Gráfico e Rato | Sim |
| Leaderboard | Placa Gráfica e Teclado | Não |
| Seleção de personagem | Placa Gráfica, Rato e Teclado | Não |
| Cenário de fundo dinâmico | Placa Gráfica e RTC | Sim |
| Temporizador | Placa Gráfica e Timer | Sim |
| Mover, saltar, chutar e cabecear | Placa Gráfica e Teclado | Sim |
| Modo multijogador | Porta-Série | Sim |

**2.2 Dispositivos Utilizados**

|  |  |  |
| --- | --- | --- |
| Dispositivo | Utilização | Interrupções |
| *Timer* | Controlo da *frame-rate* e dos temporizadores do jogo | Sim |
| Teclado | Navegação das opções dos menus e movimentação da personagem | Sim |
| Rato | Navegação das opções dos menus e ativação de poder especial | Sim |
| Placa de vídeo | Apresentação dos menus e do jogo no ecrã | Não |
| *Real-Time Clock* | Modificação da imagem de fundo do jogo de acordo com a realidade | Sim |
| Porta-série (UART) | Comunicação ponto-a-ponto que permite a modo multi-jogador | Sim |

**2.2.1 Timer**

Através das interrupções geradas pelo contador (*counter*) 0 do *timer*, tal como foram configuradas pelo MINIX (60Hz – 60 interrupções por segundo), conseguimos definir a *frame-rate*, ao modificar o conteúdo que estar a ser apresentado no ecrã a cada 2 interrupções (total de 30 *frames* por segundo). Isto é feito através de uma variável que guarda o número de interrupções e uma simples condição no *loop* de interrupções - função **interrupt\_loop()** no ficheiro **proj.c** – quando é recebida uma interrupção do *timer*.

Utilizando a mesma variável que guarda o número de interrupções, enquanto o utilizador está em modo de jogo (Um-Jogador ou Multi-Jogador), de cada vez que se passam 60 interrupções (1 segundo), a variável que guarda o tempo restante é atualizada (função **remove\_time()** no ficheiro **timeboard.c**).

Os detalhes de implementação podem ser encontrados no ficheiro **timer.c**.

**2.2.2 Teclado**

Tanto para a seleção de opções como para a movimentação da personagem, o teclado foi utilizado.

Através do processamento dos *makecodes* e *breakcodes* gerados pelo teclado, identificamos a tecla pressionada e realizamos a transformação necessária.

Os detalhes de implementação podem ser encontrados nos ficheiros **keyboard.c** e **kbc.c**.

**2.2.3 Rato**

Nos menus, para a detetar a movimentação e botões do rato, de forma a mostrar o respetivo ponteiro e interagir com os botões no ecrã são interpretados os pacotes de dados

Os detalhes de implementação podem ser encontrados nos ficheiros **mouse.c** e **kbc.c**.

**2.2.4 Placa de vídeo (GPU)**

Utilizamos a placa de vídeo para mostrar toda a interface gráfica do jogo no ecrã.

Estando a placa em modo gráfico (em oposição a modo de texto), a configuração escolhida foi o modo 0x115 do *Standard VBE*, com uma resolução de 800x600 pixéis, um modo de cor direto com 3 bytes por pixel (1 byte para cada canal, vermelho, azul e verde).

Com um total de, aproximadamente, 16.8 milhões de cores que este modo permite, todos os objetos mostrados no ecrã, incluindo personagens, botões, números, texto e até imagens de fundo foram usados a partir de ficheiros em formato XPM. A conversão foi feita para este formato usando o *software* **GNU Image Manipulation Program (GIMP)** após as *sprites* e imagens de fundo terem sido criadas.

Para evitar artefactos visuais como *screen tearing*, foi utilizada a técnica de *Double Buffering* com cópia, sendo um segundo *buffer* guardado em RAM para montar o novo *frame* e copiado para o *buffer* em VRAM no final (função **draw\_frame\_end()**). Adicionalmente, foi criado um terceiro buffer para guardar partes estacionárias do jogo (imagens de fundo, balizas, *scoreboard*) e evitar que fossem carregadas novamente para um buffer de cada vez que são necessárias (a cada frame). Assim, de cada vez que é necessário desenhar um frame (no segundo *buffer*), começa-se por copiar o conteúdo do terceiro *buffer* para o segundo (função **draw\_frame\_start()**).

**2.2.5 Real-Time Clock (RTC)**

Asdasdasdasda

Apresentação da imagem de fundo do céu coincidente com o tempo real de onde esta aplicação está a ser executada

**2.2.6 Porta-série (UART)**

Asdasdasdasda

Comunicação entre as duas máquinas (quer no mesmo computador, ou em computadores distintos) virtuais para permitir a interação multi-jogador

3. Estrutura e Organização de Código

4. Detalhes de Implementação

5. Conclusão