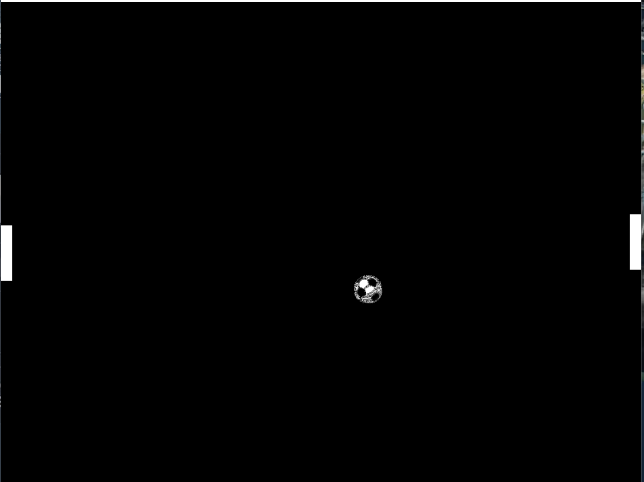
**Pong**

**Secção 1 – Instruções:**

Para o nosso projeto, decidimos fazer uma implementação do jogo Pong, para 2 jogadores, usando o timer, o keyboard e a placa gráfica.

Como mencionado anteriormente, o jogo precisa de 2 jogadores para controlar as 2 paddle’s, um jogador controla a da esquerda com as teclas W e S e o outro jogador controla a da direita com as teclas Arrow Up e Arrow Down.

A bola começa também com uma certa velocidade e á medida que os jogadores vão marcando pontos a velocidade desta aumenta, aumentando consequentemente também o nível de dificuldade do jogo.



1. Aspeto final do jogo

**Secção 2 – Estado do Projeto:**

|  |  |  |
| --- | --- | --- |
| Dispositivo | Funcionalidade | Interrupções |
| Timer | Controlo do frame-rate | Sim |
| Keyboard | Movimento das paddles | Sim |
| Placa Gráfica | Desenhar paddles e carregar xpm da bola | Não |

1. **Timer**

Neste projeto, usámos o timer apenas para controlar o frame-rate do jogo.

1. **Keyboard**

Uma das funcionalidades mais importantes a implementar no projeto era o movimento das paddle’s. Para isso usámos as funções move\_paddle\_arrows() e move\_paddle\_ws(). Estas funções fazem a mesma coisa, que é alterar as coordenadas da posição da paddle da esquerda ou da direita mas tendo em conta a leitura de diferentes scancodes. No caso da função move\_paddle\_arrows() quando se lê o scancode da Arrow Up key, volta-se a desenhar a paddle com as coordenadas da sua posição atualizada, neste caso a coordenada ‘y’ é incrementada dando a ilusão de que a paddle “sobe”, enquanto que quando se lê o scancode da Arrow Down key, acontece o mesmo só que a coordenada ‘y’ é decrementada. O mesmo acontece com a função move\_paddle\_ws(), só que com os scancodes das teclas W e S, respetivamente para cada um dos cenários mencionados anteriormente.

1. **Placa Gráfica**

A placa gráfica foi usada, no modo 0x14c, para desenhar alguns sprites tornando assim o aspeto final do jogo mais apelativo. Para isto usámos as funções draw\_paddles() que utiliza a função que já tínhamos definido no lab.5, draw\_rectangle(), para desenhar as paddles de cada jogador. Utilizámos também a função xpm\_load() para carregar uma imagem para dar um aspeto mais único à bola.

1. **Outras funcionalidades**

Para além das funcionalidades anteriores, implementámos a função check\_collision(), que verifica se houve uma colisão entre a bola e uma das paddles. Para além disso, implementámos também uma função move\_ball(), que gera o movimento autónomo da bola. Nesta função, faz-se a análise das coordenadas da posição da bola, de modo a saber se ela passou do limite da tela e um ponto foi marcado, caso tenha sido marcado um ponto por um dos jogadores a bola volta ao centro da tela. Já no caso de ter sido detetada a colisão da bola com uma das paddles, muda-se a direção da bola ao alterar o sinal da coordenada ‘x’ do vetor de movimento e analisa-se também o ângulo da colisão, de acordo com a seguinte fórmula: (p[i].y + p[i].sprite.height) - b.y, de modo a que bola se comece a mover na direção correta.

**Secção 3 – Organização do código:**

**Módulo Timer**

Neste módulo, temos presente as funções que tinham sido previamente desenvolvidas no lab2 para se configurar o Timer. Ambos contribuíram igualmente para o desenvolvimento do módulo.

**Módulo Keyboard**

Neste módulo, temos presente as funções que tinham sido previamente desenvolvidas no lab3 para se configurar o Keyboard. Ambos contribuíram igualmente para o desenvolvimento do módulo.

**Módulo Video**

Neste módulo, temos presente as funções que tinham sido previamente desenvolvidas no lab5 para se configurar a Placa Gráfica. Ambos contribuíram igualmente para o desenvolvimento do módulo.

**Módulo Game**

Neste módulo, temos presente a função play() que cria a atual instância do jogo, ativando o video mode do Minix no mode especificado e desenhando as paddles de cada jogador e a bola. Para além disso temos as funções que tratam do movimento da bola, move\_ball(), e as funções que tratam do movimento das paddles, move\_paddle\_arrows() e move\_paddle\_ws(), temos também a função que verifica a colisão entre a bola e uma paddle, check\_collision(), e por último temos a função que desenha as paddles, draw\_paddles(). Ambos contribuíram igualmente para o desenvolvimento deste módulo, tendo sido o aluno Ricardo Silva responsável por desenvolver as funções move\_ball() e check\_collision() e o aluno Jorge Costa responsável por desenvolver as funções draw\_paddles, move\_paddle\_arrows() e move\_paddle\_ws().

**Secção 4 – Conclusão:**

Primeiro de tudo, queremos realçar o facto de que o trabalho não foi desenvolvido pelo grupo todo, tendo sido os únicos que contribuíram para o trabalho os alunos Ricardo Silva e Jorge Costa, esperamos então que a nota seja apenas distribuída por esses dois alunos e que seja considerada esta situação na avaliação do trabalho. Tendo em conta este problema o desenvolvimento do trabalho revelou-se ser um bocado difícil, levando a um resultado que não foi o que desejávamos apresentar mas que foi o que se pode desenvolver com o tempo que tivemos. Sentimos que conseguimos aprofundar melhor alguns dos conhecimentos lecionados e outros graças a pesquisa que fizemos para desenvolver o nosso jogo.