

## **Práctica 2 PDIH**



**UNIVERSIDAD  
DE GRANADA**

**Enrique González López  
Aaron Rivet**

En esta práctica se nos ha pedido desarrollar el juego Pong, haciendo uso de C y de la librería ncurses. Para llevarla a cabo hemos hecho un código el cual nos muestra inicialmente un menú que nos presenta el juego y los controles de éste. Tras continuar con el programa se inicia el programa y este termina cuando uno de los dos jugadores alcanza una puntuación de 5 puntos.

Vamos a analizar el código, empezando por el menú:

```

28 if (has_colors() == FALSE) {
29     endwin();
30     printf("El terminal no tiene soporte de color \n");
31     exit(1);
32 }
33
34 start_color();
35 init_pair(1, COLOR_YELLOW, COLOR_BLACK);
36 clear();
37
38 refresh();
39 getmaxyx(stdscr, rows, cols);
40
41 WINDOW *window1 = newwin(rows,cols,0,0);
42 wbkgd(window1, COLOR_PAIR(1));
43 box(window1, '|', '-');
44
45 mvwprintw(window1, 3, cols/2-42, "PPPPPPPPPPPPPPPPPPPP          000000000          NNNNNNNN          NNNNNNNN          GGGGGGGGGGGG");
46 mvwprintw(window1, 4, cols/2-42, "P:PPPPPPPPPPPPPPPPPPPP          00:00000000          N:00000000          N:000000          GGG:0000000000");
47 mvwprintw(window1, 5, cols/2-42, "P:PPPPPPPPPPPPPPPPPPPP          00:00000000          N:00000000          N:000000          GG:0000000000");
48 mvwprintw(window1, 6, cols/2-42, "PP:PPPPPP          P:PPPPPP          00:00000000          0N:00000000          N:000000          G:0000000000");
49 mvwprintw(window1, 7, cols/2-42, " P:PPPP          P:PPPP          00:000000          0N:00000000          N:000000          G:0000000000");
50 mvwprintw(window1, 8, cols/2-42, " P:PPPP          P:PPPP          00:000000          0N:00000000          N:000000          NG:0000000000");
51 mvwprintw(window1, 9, cols/2-42, " P:PPPPPPPPPPPP          P:000000          0:000000          0N:00000000          N:000000          NG:0000000000");
52 mvwprintw(window1, 10, cols/2-42, " P:PPPPPPPPPP          PP:000000          0:000000          0N:00000000          N:000000          NG:0000000000");
53 mvwprintw(window1, 11, cols/2-42, " P:PPPPPPPPPP          P:000000          0:000000          0N:00000000          N:000000          NG:0000000000");
54 mvwprintw(window1, 12, cols/2-42, " P:PPPP          P:000000          0:000000          0N:00000000          N:000000          NG:0000000000");
55 mvwprintw(window1, 13, cols/2-42, " P:PPPP          P:000000          0:000000          0N:00000000          N:000000          NG:0000000000");
56 mvwprintw(window1, 14, cols/2-42, " P:PPPP          P:000000          0:000000          0N:00000000          N:000000          NG:0000000000");
57 mvwprintw(window1, 15, cols/2-42, "PP:PPPPPP          PP:000000          0:000000          0N:00000000          N:000000          G:0000000000");
58 mvwprintw(window1, 16, cols/2-42, "P:PPPPPPPP          P:000000          00:000000          00          N:000000          N:000000          GG:0000000000");
59 mvwprintw(window1, 17, cols/2-42, "P:PPPPPPPP          P:000000          00:000000          00          N:000000          N:000000          GGG:0000000000");
60 mvwprintw(window1, 18, cols/2-42, "PPPPPPPPPP          000000000          NNNNNNNN          NNNNNNNN          GGGGGG          GGGG");
61
62 mvwprintw(window1, rows/2, cols/2-16, "EL JUGADOR (1) SE MUEVE CON W Y S");
63 mvwprintw(window1, rows/2+1, cols/2-16, "EL JUGADOR (2) SE MUEVE CON O Y L");
64 mvwprintw(window1, rows/2+5, cols/2-10, "PRESS 'INTRO' TO PLAY");
65
66 wrefresh(window1);
67
68 getch();

```

Primero se comprueba si nuestra máquina puede trabajar con colores, luego creamos una ventana que con el fondo negro y con el resto de elementos que escribamos en ella de amarillo y finalmente escribimos el mensaje, dejando el programa a la espera de que pulsemos alguna tecla del teclado para continuar. Este sería el resultado:



- El campo

```
//Dibujar el campo
WINDOW *window2 = newwin(30,100,0,0);
wbkgd(window2, COLOR_PAIR(1));
box(window2, '|', '-');

wrefresh(window2);

for(int i = 1; i < 29; i++)
{
    mvprintw(i, 50, "|");
}
```

El campo consiste en una ventana de 30 de altura y 100 de longitud, que sigue el mismo patrón de colores anterior. Con el último for dibujamos la línea central del campo y al hacerlo después de recargar la ventana esta es de un color distinto, para poder diferenciarla con más facilidad.

- Jugadores, bola y rebotes

```
//Jugador 1
int xj1 = 5, yj1 = 11;

//Jugador 2
int xj2 = 95, yj2 = 11;
```

```
//Crear los objetos
for(int i = 0; i < 8; i++)
{
    mvprintw(yj1+i, xj1, "#");
}

for(int i = 0; i < 8; i++)
{
    mvprintw(yj2+i, xj2, "#");
}
```

```

// Mover las barras
if(ch == 'W' || ch == 'w')
{
    if(yj1 != 1)
    {
        yj1 -= 1;
    }
}
else if(ch == 'S' || ch == 's')
{
    if(yj1 != 21)
    {
        yj1 += 1;
    }
}

if(ch == 'O' || ch == 'o')
{
    if(yj2 != 1)
    {
        yj2 -= 1;
    }
}
else if(ch == 'L' || ch == 'l')
{
    if(yj2 != 21)
    {
        yj2 += 1;
    }
}

```

Con las dos primeras partes del código, inicializamos las variables xj1,yj1,xj2 y yj2, y las mostramos en la pantalla. Para darles movimiento usamos el código de la última captura establecemos con que teclas se van a mover los jugadores y les ponemos límites para que no salgan de la pantalla.

```

//Bola
int xb = 50;
int yb = 15;

int next_x = 0;
int directionx = 1;

int next_y = 0;
int directiony = 1;

```

```
mvprintw(yb, xb, "0");
```

Primero inicializamos las coordenadas iniciales de la bola(el centro del campo), las siguientes posiciones de esta y la dirección en la que se va a mover.

```
next_x = xb + directionx;
next_y = yb + directiony;

//Rebote con las paredes
if (next_x >= 100 || next_x < 0) {
    xb = 50;
    yb = 15;
    directionx *= -1;
    yb += directiony;
    if(next_x >= 100)
    {
        marcador1 +=1;
    }
    else if(next_x < 0)
    {
        marcador2 +=1;
    }
}
else
{
    xb += directionx;
}

if (next_y >= 29 || next_y < 1) {
    directiony *= -1;
} else {
    yb += directiony;
}
```

```
//Rebote con los jugadores
if ((next_x == xj1 && (next_y == yj1 || next_y == yj1+1 || next_y == yj1+2 || next_y == yj1+3 || next_y == yj1+4 || next_y == yj1+5 || next_y == yj1+6 || next_y == yj1+7)) || (next_x == xj2 && (next_y == yj2 || next_y == yj2+1 || next_y == yj2+2 || next_y == yj2+3 || next_y == yj2+4 || next_y == yj2+5 || next_y == yj2+6 || next_y == yj2+7))) {
    directionx *= -1;
}
```

Para los rebotes con las paredes hemos usado un código que depende de cual vaya a ser la siguiente posición de la bola. En el caso de que alcance los límites laterales, la bola volverá al centro del campo, invirtiendo su dirección del eje X y se sumará un punto al marcador de jugador del lado contrario. Si la bola alcanza los límites verticales esta invertirá su dirección del eje Y. En cualquier caso que la bola no alcance ninguno de estos límites simplemente seguirá su dirección.

Para el rebote de los jugadores simplemente comprobamos que si la bola va a encontrarse en alguna de las coordenadas de los jugadores, se invierte la dirección del eje X

- **El marcador**

```
//Puntuación
int marcador1 = 0;
int marcador2 = 0;
int ganador1 = 0;
int ganador2 = 0;

while(ganador1 == 0 && ganador2 == 0) {
    clear();
    refresh();
```

Todos los movimientos ocurren dentro de un bucle do while controlado por el marcador. Una vez gana alguno de los jugadores este se termina y finaliza el programa. Como se ha visto antes con los rebotes laterales controlamos la puntuación y la mostramos con el siguiente código:

```
if(marcador1 == 0)
{
    mvprintw(32, 24, "0");
}
else if(marcador1 == 1)
{
    mvprintw(32, 24, "1");
}
else if(marcador1 == 2)
{
    mvprintw(32, 24, "2");
}
else if(marcador1 == 3)
{
    mvprintw(32, 24, "3");
}
else if(marcador1 == 4)
{
    mvprintw(32, 24, "4");
}

if(marcador2 == 0)
{
    mvprintw(32, 74, "0");
}
else if(marcador2 == 1)
{
    mvprintw(32, 74, "1");
}
else if(marcador2 == 2)
{
    mvprintw(32, 74, "2");
}
else if(marcador2 == 3)
{
    mvprintw(32, 74, "3");
}
else if(marcador2 == 4)
{
    mvprintw(32, 74, "4");
}
```

Cuando tenemos un ganador, cambiamos la variable ganador correspondiente, lo que finaliza el do while y mostramos un mensaje indicando quién ha ganado

```
        if(marcador1 == 5)
        {
            ganador1 = 1;
        }

        if(marcador2 == 5)
        {
            ganador2 = 1;
        }
    }

    if(ganador1 == 1)
    {
        mvprintw(32, 24, "5");
        mvprintw(31, cols/2-8, "GANA EL JUGADOR 1");
    }
    else if(ganador2 == 1)
    {
        mvprintw(32, 74, "5");
        mvprintw(31, cols/2-8, "GANA EL JUGADOR 2");
    }
}
```

Y tras este mensaje, el programa queda a la espera de que pulsemos alguna tecla y termina.

Para acabar estas son las capturas del juego en funcionamiento:



