

PDIH - PRÁCTICA 2

Jorge Sánchez - Carlos Guasp



UNIVERSIDAD DE GRANADA

ÍNDICE

Código .c de las funciones	2
Ejecución de funciones	7

Código .c de las funciones

`hello.c` hace un print de Hello World en el código

```
You, 1 second ago | 2 authors (Jorge Sanchez al  
#include <ncurses.h>  
  
int main() {  
    initscr();  
    printw("Hello World!");  
    refresh();  
    getch();  
    endwin();  
    return 0;  
}
```

`ventana.c` crea una ventana en un terminal que soporta colores. Inicializa pares de colores y muestra un mensaje dentro de un marco en la ventana.

```
Jorge Sanchez, 14 minutes ago | 1 author (Jorge Sanchez)  
#include <stdlib.h>  
#include <ncurses.h>  
  
int main(void) {  
    int rows, cols;  
  
    initscr();  
    if (has_colors() == FALSE) {  
        endwin();  
        printf("El terminal no tiene soporte de color\n");  
        exit(1);  
    }  
  
    start_color();  
    init_pair(1, COLOR_YELLOW, COLOR_GREEN);  
    init_pair(2, COLOR_BLACK, COLOR_WHITE);  
    init_pair(3, COLOR_WHITE, COLOR_BLUE);  
  
    clear();  
    refresh();  
    getmaxyx(stdscr, rows, cols);  
  
    WINDOW *window = newwin(rows, cols, 0, 0);  
    wbkgd(window, COLOR_PAIR(3));  
    box(window, '|', '-');  
  
    mvwprintw(window, 10, 10, "una cadena");  
    wrefresh(window);  
  
    getch();  
    endwin();  
    return 0;  
}
```

`pelotita.c` crea una animación en la que un carácter 'o' se mueve horizontalmente a través de la pantalla. Se invierte la dirección del movimiento al alcanzar los bordes definidos

```
#include <ncurses.h>
#include <unistd.h>

#define DELAY 30000

int main() {
    int x = 0, y = 0;
    int max_y = 50, max_x = 50;
    int next_x = 0;
    int direction = 1;

    initscr();
    noecho();
    curs_set(FALSE);

    while (1) {
        clear();
        mvprintw(y, x, "o");
        refresh();
        usleep(DELAY);
        next_x = x + direction;

        if (next_x >= max_x || next_x < 0) {
            direction *= -1;
        } else {
            x += direction;
        }
    }

    endwin();
}
```

`pong.c` implementa un simple juego de pong en el terminal. Se dibujan dos palas y una pelota que se mueve automáticamente por la pantalla. Las palas pueden ser controladas

por el jugador usando las teclas 'w' y 's' para la pala izquierda, y las teclas de flecha arriba y abajo para la pala derecha. El juego lleva un registro de los puntos de cada jugador y reinicia la posición de la pelota cuando un jugador anota. El programa se ejecuta en un bucle hasta que el jugador presiona la tecla 'q' para salir.

```
#include <ncurses.h>
#include <unistd.h>

#define DELAY 70000
#define WIDTH 80
#define HEIGHT 24
#define PADDLE_HEIGHT 4

void draw_paddle(int y, int x) {
    for (int i = 0; i < PADDLE_HEIGHT; ++i) {
        mvprintw(y + i, x, "|");
    }
}

int main() {
    int ball_x = WIDTH / 2, ball_y = HEIGHT / 2;
    int ball_dir_x = 1, ball_dir_y = 1;

    int left_paddle_y = HEIGHT / 2 - PADDLE_HEIGHT / 2;
    int right_paddle_y = HEIGHT / 2 - PADDLE_HEIGHT / 2;

    int left_score = 0, right_score = 0;
    int ch;

    initscr();
    noecho();
    curs_set(FALSE);
    cbreak();
    nodelay(stdscr, TRUE);
    keypad(stdscr, TRUE);

    while (1) {
        clear();

        // Dibujar palas
        draw_paddle(left_paddle_y, 2);
        draw_paddle(right_paddle_y, WIDTH - 3);

        // Dibujar pelota
        mvprintw(ball_y, ball_x, "o");

        // Dibujar marcador
        mvprintw(0, WIDTH / 2 - 10, "P1: %d   P2: %d", left_score, right_score);

        refresh();
        usleep(DELAY);

        // Mover pelota
        ball_x += ball_dir_x;
        ball_y += ball_dir_y;

        // Rebotar en bordes superior/inferior
        if (ball_y <= 1 || ball_y >= HEIGHT - 1)
            ball_dir_y *= -1;

        // Rebotar con palas
        if (ball_x == 3 && ball_y >= left_paddle_y && ball_y < left_paddle_y + PADDLE_HEIGHT)
            ball_dir_x *= -1;

        if (ball_x == WIDTH - 4 && ball_y >= right_paddle_y && ball_y < right_paddle_y + PADDLE_HEIGHT)
            ball_dir_x *= -1;

        // Punto para el jugador derecho
        if (ball_x <= 0) {
            right_score++;
            ball_x = WIDTH / 2;
            ball_y = HEIGHT / 2;
        }
    }
}
```

```

        ball_dir_x = 1;
        ball_dir_y = (ball_dir_y == 0) ? 1 : ball_dir_y;
    }

    // Punto para el jugador izquierdo
    if (ball_x >= WIDTH - 1) {
        left_score++;
        ball_x = WIDTH / 2;
        ball_y = HEIGHT / 2;
        ball_dir_x = -1;
        ball_dir_y = (ball_dir_y == 0) ? -1 : ball_dir_y;
    }

    // Controles de palas
    ch = getch();
    if (ch == 'q') break;

    if (ch == 'w' && left_paddle_y > 1)
        left_paddle_y--;
    if (ch == 's' && left_paddle_y + PADDLE_HEIGHT < HEIGHT - 1)
        left_paddle_y++;
    if (ch == KEY_UP && right_paddle_y > 1)
        right_paddle_y--;
    if (ch == KEY_DOWN && right_paddle_y + PADDLE_HEIGHT < HEIGHT - 1)
        right_paddle_y++;
}

endwin();
return 0;
}

```

Ejecución de funciones

Ejecución de `hello.c`

```
Hello World!
```

Ejecución de `ventana.c`

```
una cadena
```

Ejecución de `pelotita.c`

```
0
```

```
0
```

Ejecución de `pong.c`

