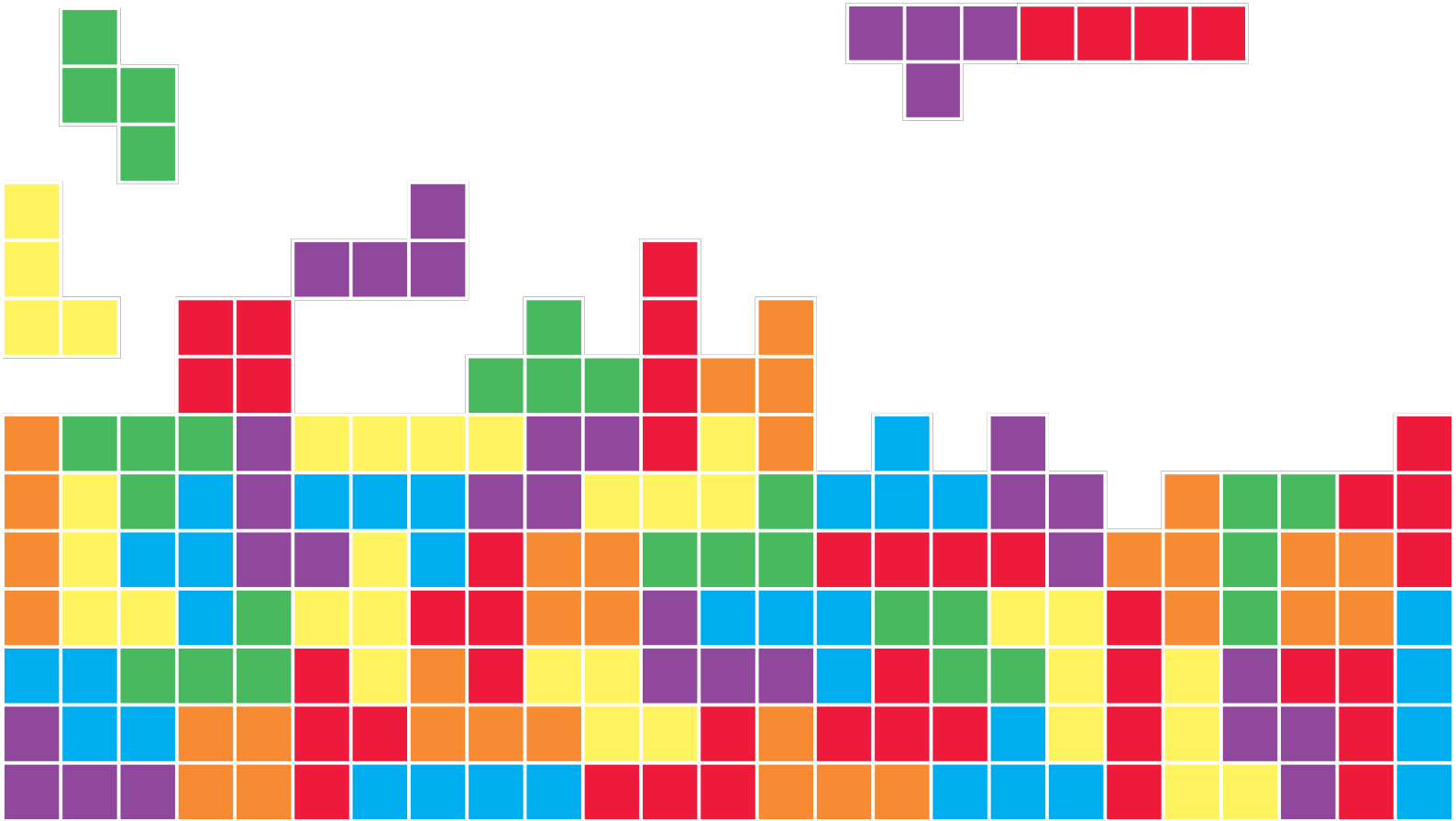


# TETRIS

**Roberto Pitarch Carrasco**  
**48753014-Y**

**Sergio Pardo García**  
**49194382-G**

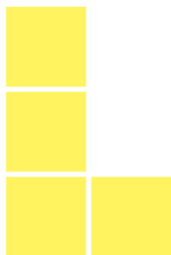


# ÍNDICE

## Traducción

## Implementación

- Marcador de puntuación
- Final de la partida
- Completando líneas
- Eliminando líneas
- Mostrar la siguiente pieza



# TRADUCCIÓN

Hemos realizado la traducción de todas las funciones incompletas de `tetris.c`:

- `imagen_set_pixel`
- `imagen_clean`
- `imagen_init`
- `imagen_copy`
- `imagen_dibuja_imagen`
- `imagen_dibuja_imagen_rotada`
- `nueva_pieza_actual`
- `intentar_movimiento`
- `intentar_rotar_pieza_actual`
- `bajar_pieza_actual`

# IMPLEMENTACIÓN

Hemos realizado la implementación de las 5 funciones para contar la puntuación, terminar la partida, aumentar los puntos cuando se completa una fila, eliminar la fila y tener una previsualización de la siguiente pieza.

## Marcador de puntuación

Cada nueva pieza que aparezca sumará 1 punto. La puntuación hasta el momento “puntuacion\_actual” se muestra sobre el campo de juego. Hemos utilizado la función “integer\_to\_string” que implementamos en la práctica 2, la cual recibe como parámetros un valor entero y una dirección de memoria a partir de la cual deja la cadena de caracteres que contiene la representación de los dígitos del entero.

Para dibujar dicha cadena, hemos implementando una función llamada “imagen\_dibuja\_cadena”, la cual recibe como parámetros la dirección de la imagen donde se dibujará la cadena, las coordenadas dentro de la imagen en las que comenzar a escribir y la dirección donde se encuentra la cadena que se desea escribir.

## Final de la partida

Para comprobar si hemos llegado a la altura total del campo utilizamos la función “probar\_pieza”, la cual devuelve 1 si la nueva pieza que va a entrar al campo puede entrar y 0 si la nueva pieza no puede entrar. Esta comprobación la realizamos llamando a la función “game\_over” desde “actualizar\_pantalla”. En caso de devolver 0, creamos un cuadro en el centro del campo con los mensajes “GAME OVER” y “Pulsa una tecla!”. Cuando el usuario pulsa una tecla, vuelve al menú principal.

```
+ - - - - - +
|   GAME OVER   |
|               |
|   Pulsa una   |
|   tecla!      |
+ - - - - - +
```

## Eliminando líneas

Cada vez que la función “comprobar\_linea” encuentra una línea horizontal, ésta llama a “eliminar\_linea”, la cual se encarga de eliminar la línea completa y bajar el resto de piezas. Para ello recorre el campo de abajo a arriba y copia los caracteres de la línea anterior a donde los copia.

## Mostrar la siguiente pieza

Una forma de hacer más fácil nuestro juego, es conocer cuál será la próxima pieza que caerá. De esta manera podemos seleccionar dónde poner la actual para encajar mejor la siguiente. Queremos que la siguiente pieza se muestre en la parte superior del campo de juego, a la derecha del campo actual y rodeada por un recuadro.

Para ello hemos modificado la función “nueva\_pieza\_actual”, la cual había sido traducida previamente. Dicha función hace uso de la función “imagen\_copy” para copiar la pieza aleatoria creada anteriormente de “pieza\_next” a “pieza\_actual”. A su vez, dentro de la función “actualizar\_pantalla” hemos creado el cuadro en el cual aparecen las piezas siguientes, haciendo uso de la función creada “imagen\_dibuja\_cadena”.

