***Alumno**: Kevin Gómez Codina*

***Módulo**: M06 UF2*

***Repositorio**: https://github.com/kegoco/DAWM06UF2_tetris*

1. Objetivos cumplidos

- Se ha realizado todas las tareas propuestas en el enunciado de la práctica.

2. Ampliaciones realizadas

- Las piezas se muestran con color por pantalla.
- La pieza se rota en sentido antihorario sin tener que ejecutar tres veces la función de rotar.
- Se puede iniciar otra partida después de otra sin tener que recargar la página.

3. Explicación del código

A continuación se explicará las partes más “engorrosas” que encontramos en el código, ampliando información que no encontraremos en los comentarios.

3.1. Uso de constantes

En el fichero *index.html*, antes de cargar los *scripts* para el videojuego se carga un *script* en el que tienen almacenadas las constantes de la aplicación.

```
<!-- CARGA DE SCRIPTS -->
<script src="./js/constants.js"></script>
<script src="./js/objects/Piece.js"></script>
<script src="./js/objects/Tetris.js"></script>
<script src="./js/main.js"></script>
```

3.2. Comprobar posición válida

En el momento en que programé esta función me hice un lío respecto a las posiciones del tablero y a la forma de la pieza.

Mi idea era mostrar la pieza por pantalla tal y como la defino en la constante, es decir, que la pieza “T” (por ejemplo) se mostrase tal cual y no invertida. A causa de esta idea opté por mostrar el mapa con un *for* de *rows* hasta cero (tal y como se muestra en la imagen).

```
192     paintScreen: function () {
193         // Muestra el videojuego por pantalla
194         var table = $("#tetris");
195         table.empty();
196         var content = "";
197         for (var y = ROWS - 1; y >= 0; y--) {
198             content += "<tr>";
199             for (var x = 0; x < COLUMNS; x++) {
200                 var color = (this.board[y][x] != 0) ? this.board[y][x] : "#A9A9A9";  // Gris: #A9A9A9
201                 content += "<td style='background-color: " + color + "; width: 20px; height: 20px;'></td>";
202             }
203             content += "</tr>";
204         }
205         table.append(content);
206     },
```

A causa de esta idea y del lío que me hice salió el código de la siguiente imagen:

```
127 Piece.prototype.checkPosition = function (board, position, shape) {
128     // Comprueba si la posición es válida
129     var can_move = true;
130     var x_pos = 0, y_pos = 3;
131
132     for (var y = position[0]; y < position[0] + 4 && can_move; y++) {
133         x_pos = 0;
134         for (var x = position[1]; x < position[1] + 4 && can_move; x++) {
135             if (shape[y_pos][x_pos] != 0) {
136                 if (board[y] == undefined) {
137                     // Si la posición es "undefined" o hay alguna pieza en la posición del tablero entonces no se moverá
138                     can_move = false;
139                 }
140                 else if (board[y][x] == 0) {
141                     // Si la posición del tablero está libre entonces la pieza se podrá mover
142                     can_move = true;
143                 }
144                 else if (board[y][x] != 0) {
145                     // Si la posición del tablero está ocupada entonces la pieza ya no se podrá mover más
146                     can_move = false;
147                 }
148             }
149             x_pos++;
150         }
151         y_pos--;
152     }
153
154     return can_move;
155 }
```

Como podemos ver, los dos *fors* que hay tienen como objetivo recorrer la posición actual sumando el índice de la forma de la pieza. Es por eso que he creado dos variables aparte llamadas *x_pos* e *y_pos*, para así con estas poder acceder al *array* de la forma de la pieza y así hacer las comprobaciones necesarias.

Mi idea era optimizar esta parte del código, y hacer que los *fors* sean de cero al número máximo de índices del *array* de forma, así podría ahorrarme las variables *x_pos* e *y_pos*, pero resulta que

al hacer esto las piezas me salían invertidas en el tablero (cosa que quería evitar desde el principio). No he encontrado solución, así que he optado por dejar el código de la imagen anterior.