

## PROYECTO : IMPLEMENTACIÓN HUNDIR LA FLOTA

### VERSIÓN BÁSICA – SEGUNDA PARTE

Vamos a programar la parte del Juego en la que se desarrollo la partida que funcionará de la siguiente manera.

Vamos a programar un proceso iterativo que se repetirá los dos jugadores de la partida tengan puntos.

Dentro de este proceso realizaremos los siguientes pasos:

- Mostrar Mapa del Usuario
- Mostrar Puntos restantes del Usuario y del Ordenador
- Pedimos que el jugador introduzca la tirada que quiere realizar hasta que la tirada realizada sea válida. Esto lo haremos en un método que se describirá más adelante.
- A continuación, actualizaremos el mapa del ordenador, el mapa del ordenador para usuario y los puntos del Ordenador. Esto también lo haremos en un método.
- Visualizamos mapa del Ordenador para usuario en su correspondiente método.
- Si los puntos del ordenador no son 0, se pasa a realizar el disparo del ordenador. Este se realizará aleatoriamente.
- Mostraremos puntos del ordenador
- Generaremos el disparo aleatorio y comprobaremos si el disparo es válido, ambas cosas mediante métodos. Esto lo programaremos en un bucle que se repetirá mientras el disparo no sea correcto.
- A continuación actualizaremos el mapaUsuario y los puntos Usuario.

Descripción de los distintos método.

#### **MÉTODO pedirCasilla**

Método que no recibe parámetros y que devuelve un vector de enteros que declaramos dentro del método y que será de tamaño 2.

El método pedirá que se introduzcan por teclado las coordenadas del disparo que quiere realizar el jugador.

Las coordenadas se introducen mediante un String, del que extraeremos las dos coordenadas. Por ejemplo, se introduce "B4". El método separará B y 4 para comprobar si son coordenadas válidas.

La primera debe ser un char de los válidos en las cabeceras de los mapas y la segunda un valor comprendido entre 0 y TAMANIO. (Sugerencia: utilizar el método de la clase envoltorio `Character` `Character.getNumericValue` para validar la primera coordenada y `Integer.parseInt` para la segunda.

Si ambas coordenadas son correctas se almacenarán sus valores numéricos en el vector de enteros que habremos declarado. Si no lo son se almacenarán sendos -1 en las dos posiciones del vector.

El vector será retornado por el método.

## MÉTODO evaluarDisparo

Método que devuelve un dato de tipo char que determina si el disparo ha acertado en una posición que contiene un Barco(nº), AGUA\_NO\_TOCADO('A') o error ('E') y que recibe como parámetros un array bidimensional de char correspondiente al mapa en el que queremos evaluar el disparo y un vector de enteros (vector disparo).

La comprobación la hará a partir del vector disparo que contiene las coordenadas del mismo.

## MÉTODO generarDisparoAleatorio

Método que genera dos coordenadas aleatoriamente entre las posibles para el mapa del Ordenador y las devuelve mediante un vector de enteros de tamaño 2.

---

```
Proceso DISPARO DEL JUGADOR
    Mientras puntosJugador y puntosOrdenador <>0 y
        //TURNO JUGADOR
        //Visualizar mapaOrdenadorParaUsuario
        Mostrar puntos del ordenador y puntos del usuario.
        mientras !tiroCorrecto
            pedir coordenadas del disparo (pedirCasilla() )
            si disparo[0] j= -1
                resultadoDisparo=evaluarDisparo()
                Según sea el resultado ('A','X','E')
                    Mostrar resultado
                    Actualizar mapaOrdenadorParaUsuario, mapaOrdenador y puntos del Ordenador en su caso
            Sino
                Mostrar mensaje "Coordenadas Incorrectas".
        FinMientras
        Visualizar mapaOrdenadorParaUsuario
        Si puntosOrdenador!=0
            //TURNO ORDENADOR
            Mostrar TURNO ORDENADOR
            Mostrar puntos restantes de ambos
            Mientras !tiroCorrecto
                Disparo=generarDisparoAleatorio()
                resultadoDisparo=evaluarDisparo()
                Según sea el resultado ('A','X','E')
                    Mostrar resultado
                    Actualizar mapaUsuario, mapaOrdenador y puntosUsuario en su caso
            Mostrar mapaUsuario
        FinMientras
    FinProceso
```