Batalla Naval

Seminario Microcontroladores

Alumno: Ivanic, Sergio German

El objetivo de este documento es describir el proyecto de *Batalla Naval*; narrando su desarrollo, problemas encontrados durante el mismo y exponiendo el código fuente.

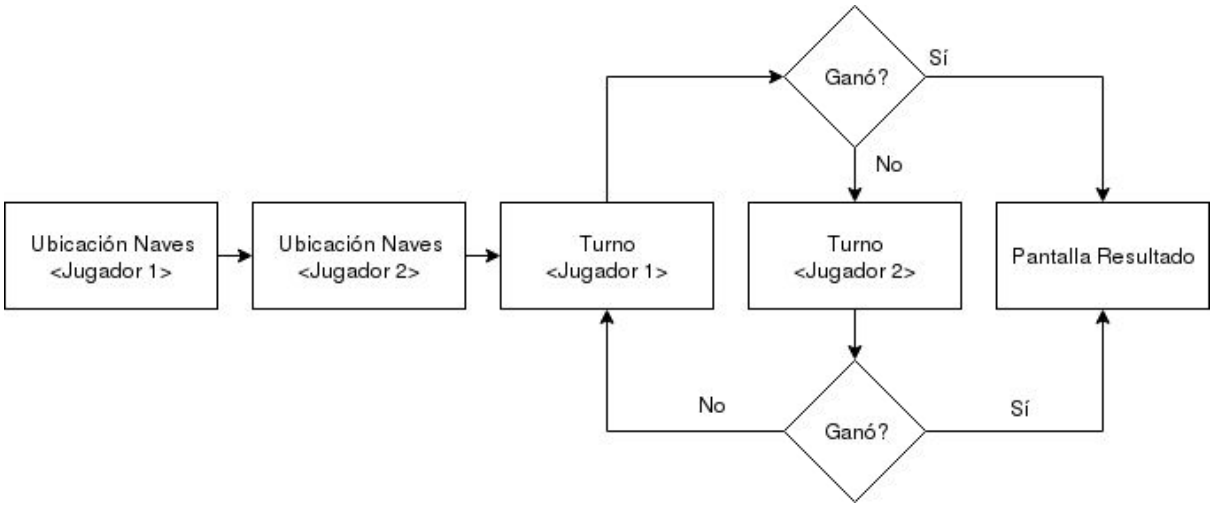
Introducción

La *Batalla Naval* es un juego de mesa para dos jugadores los cuales posicionan distintas naves sobre una matriz propia y proceden a indicar coordenadas donde atacarán en sus respectivos turnos. Cuando un jugador ataca en una coordenada donde el jugador contrario había puesto una de sus naves, esta es destruida. Cuando todas las naves de un jugador son destruidas, este pierde.
Esta mecánica de juego es la que se replicará utilizando un Arduino, dos matrices de leds (una para cada jugador) y botoneras que actuarán de *joysticks* para seleccionar las distintas coordenadas.

Implementación

Diagrama de estados

Durante el juego se tendrán varios estados que cambiarán conforme vaya avanzando la partida; dependiendo del estado que se encuentre activo, el comportamiento y las acciones de cada jugador variarán.
El siguiente diagrama presenta los estados por los que atravesará el juego:



- **Ubicación Naves:** En este estado, el jugador correspondiente ubicará *N* naves en el tablero de juego. Al colocar todas las naves que tenga disponible, se pasará al siguiente estado.
- **Turno:** En el turno de un jugador, ocurrirán distintas cosas dependiendo de quién es el jugador atacante y quién el jugador siendo atacado.
    - **Jugador Atacado:** El jugador atacado visualizará en su pantalla la ubicación de sus naves activas. No podrá realizar ninguna acción.
    - **Jugador Atacante:** El jugador visualizará en su pantalla los lugares en donde atacó en turnos previos, quedando estos deshabilitados para volver a

atacar. El jugador selecciona una ubicación donde atacar, ataca y finaliza su turno.

- **Ganó:** Aunque este no representa un estado por sí mismo ya que se encuentra dentro del estado *Turno*, se representa como la consulta de barcos disponibles para saber si hay un jugador sin naves activas y, por ende, un jugador ganador.
- **Pantalla Resultado:** En este estado simplemente se dibuja en cada pantalla un indicador de victoria o derrota.

Detalles generales del juego

- En las pantallas (matrices de leds), la posición actual seleccionada se indica con la intermitencia del led de la posición en cuestión.
- Una posición seleccionada, ya sea por *ubicación de naves* o por *ataque*, no podrá ser “visitada” nuevamente para el mismo fin.
- Por defecto (para no demorar demasiado en las pruebas), se podrán posicionar 4 naves de una posición cada una.

Materiales

Los materiales utilizados en esta versión de la Batalla Naval son:

- Arduino UNO;
- 2 Módulos de Matrices de Led 8X8 con su correspondiente integrado MAX7219 (comprado en Nubbeo);
- 5 pulsadores 6mmX6mm (comprado en Nubbeo);
- Cables;
- 5 Resistencias 220Ω;
- 1 Plaqueta perforada;

Código Fuente

El código fuente se encuentra versionado en Github, listo para ser forkeado, contribuir o utilizar de fuente de inspiración para algún otro proyecto.

Dependencia

Para poder utilizar el código fuente correctamente, se debe descargar la *library* para controlar los módulos leds via el integrado MAX7219. Dicha library puede ser encontrada en su repositorio en Github.
Para poder utilizar el código descargado, se deben colocar los archivos MaxMatrix.h y MaxMatrix.cpp dentro de <carpeta instalación de Arduino IDE>/libraries/MaxMatrix/.

Entradas por defecto

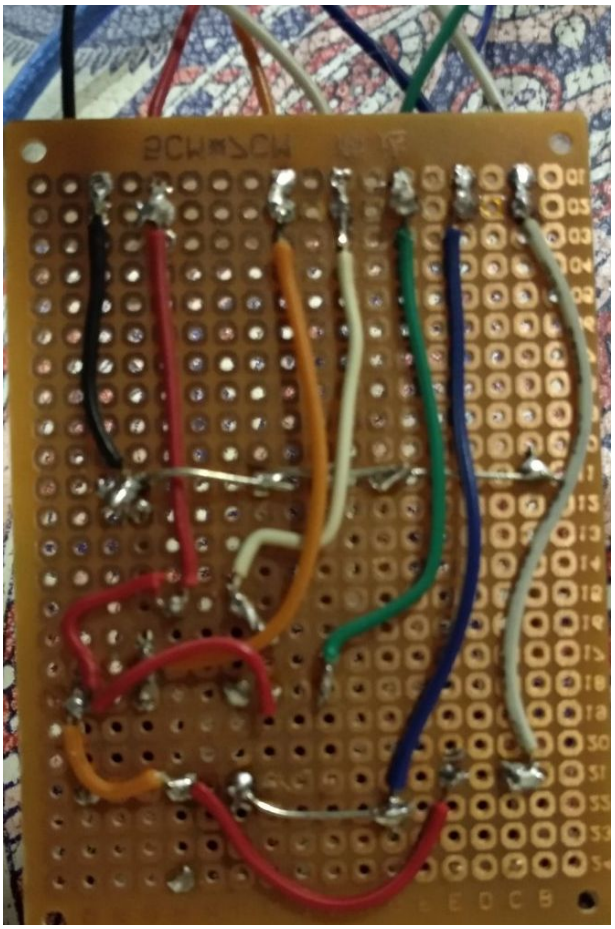
Por defecto, las entradas del arduino están distribuidas de la siguiente manera:

- Pin 1,2,3: CS, CLK, DIN respectivamente del módulo led para jugador 1;
- Pin 4,5,6: CS, CLK, DIN respectivamente del módulo led para jugador 2;
- Pin 8,9,10,11,12: Entradas del mando.

Mando

Para realizar el mando se construyó un circuito muy pequeño y simple utilizando 5 pulsadores los cuales activan 5 salidas distintas, usando resistencias en distribución *pull-down* para poder mantener un estado de reposo con nivel bajo de tensión.

Frente del mando



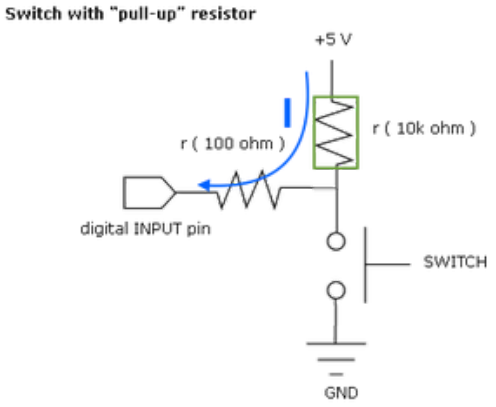
Reverso del mando

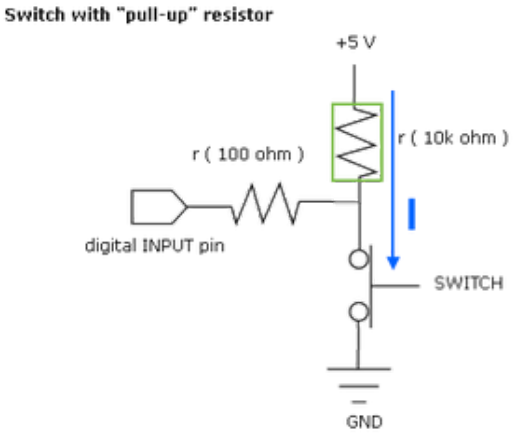
En las imágenes se pueden observar 6 cables que salen de la plaqueta. Los cables son, de izquierda a derecha, negativo, positivo, dato pulsador izquierdo, dato pulsador superior, dato pulsador derecho, dato pulsador inferior, dato pulsador selección.

Sobre pull-up y pull-down

Son denominaciones de distintas distribuciones de resistencias y pulsadores. Estas distribuciones son comúnmente utilizadas para evitar falsas lecturas ocasionadas por posibles variaciones en la tensión recibida en las entradas.
A continuación se explicará muy brevemente la diferencia entre estos dos esquemas:

- **Pull-up:** Esta distribución nos permite tener un estado en alto en la salida digital en reposo, como se puede observar de los siguientes esquemas.





- Pull-down: Por otro lado, esta distribución nos permite tener un estado en bajo en la salida digital en reposo, el cual se volverá alto una vez pulsado el pulsador.