

Proyecto de recuperación

Los Barcos (Hundir la Flota)



Darel Martínez Caballero

Clase Hundir:

Esta clase controla la interactividad de cada jugador llamando a los métodos del mundo del jugador.

Métodos a los que llama →

rellenar_tablero() (rellena el tablero de espacios),

visualizar() (visualiza el tablero con los barcos del jugador)

desvelar() (visualiza el tablero de disparos)

disparar() (Coloca una en una posición una x o * si toca un barco)

Clase Jugador

Esta clase se compone de varios métodos y se encarga del control de las acciones del jugador.

Método colocar_barcos() → si el jugador es la maquina se llama a este método y trata de colocar cada uno de los barcos de forma aleatoria en el tablero.

Método colocar_barcos_manual() → El el jugador es humano, se llama a este método y se encarga de ir colocando cada barco en una posición específica que se le solicita al usuario.

Método disparar() → Se encarga de realizar un disparo coloca una x o * si a tocado un barco, en el tablero del contrincante.

Método pedir_posicion() → solicita una posición cuando la posición sea válida retorna la misma en un arreglo.

Método eje_x() → Al llamar a este método se le pasa una letra por parámetro y devuelve la posición de la misma.

Método pedir_opcion() → solicita al usuario en que posición desea colocar el barco. Solo es llamado cuando el barco es un Yate o Transatlantico, y existen diferentes opciones (Arriba, abajo, izquierda, derecha y las 4 diagonales)

aleatorio() → devuelve un numero alea torio entre 0 y 9

Clase Movimiento

En esta clase se tratará de colocar un barco en una posición del tablero.

Método posición barco → Se le pasa por parámetros todo lo necesario para poder colocar un barco, (opción, el barco, el tablero visible, columna y fila)

Según la opción que se le pase llamara a los distintos métodos de posiciones.

vertical_arriba()

Trata de colocar el barco en una posición vertical hacia arriba desde la posición que se le pasa por parámetro. Recibe como argumentos el tablero visible, el barco a colocar, las columnas y las filas de la posición desde donde quiere el usuario que se coloque.

Método vertical_abajo()

Lo mismo que el anterior, pero esta vez en posición vertical_abajo

Método horizontal_izquierda()

Lo mismo que el anterior, pero esta vez en posición horizontal izquierda

Método horizontal_derecha()

Lo mismo que el anterior, pero esta vez en posición horizontal derecha

Método diagonal_arriba_izquierda

Lo mismo que el anterior, pero esta vez en posición diagonal superior izquierda

Método diagonal_abajo_izquierda

Lo mismo que el anterior, pero esta vez en posición diagonal abajo izquierda

Método diagonal_arriba_derecha

Lo mismo que el anterior, pero esta vez en posición diagonal arriba derecha

Método diagonal_arriba_izquierda

Lo mismo que el anterior pero esta vez en posición diagonal arriba izquierda

Método compruebaAlrededor()

Este método llamará a los métodos de la clase Comprueba para saber si la siguiente posición es ocupada, y si este barco se puede colocar en la posición que se le pasa por parámetro.

Método generar_tablero_temporal ()

Este método genera un tablero temporal con el que se va a intentar realizar la colocación del barco.

Clase Mundo

Esta clase contiene los barcos de cada jugador, controla si se a tocado o hundido un barco

Método getTransatlantico() → devuelve los transatlánticos de este mundo

Método getYates() → devuelve los yates de este mundo

Método getSubmarinos → devuelve los submarinos de este mundo

Método isColocado() → Cuando se llama a este metodo este a su vez llamará a Movimiento con la posición del barco e intentará colocarlo. Si el barco se a generado por completo devolverá true.

Método disparo() → Este método coloca una x o * si se a tocado un barco. Y llamará a los métodos **compruebaYate()** **compruebaSubmarino()** y **compruebaTransatlantico()** para saber si está tocado o hundido y quien.

Método compruebaTransatlantico() → comprueba si a sido tocado un transatlantico o si se a hundido

Método compruebaYate() → comprueba si a sido tocado un yate o si se a hundido

Método compruebaSubmarino() → comprueba si a sido tocado un submarino o si se a hundido

Método todos_hundidos() → comprobará si se a llegado a la totalidad de barcos hundidos, si es a si devuelve true.

Método desvelar() → Se encarga de mostrar el tablero de disparos del contrincante.

Método visualizar() → Muestra el tablero del jugador donde se encuentran visible los disparos y los barcos.

Método rellenar_tablero() → Se llama al principio para que genere un tablero relleno de espacios

Método getTablero() → devuelve el tablero visible.

Clase comprueba

Esta clase se encarga de comprobar las posiciones alrededor de donde se quiere colocar el barco.

Método **compruebaArriba()** → se mueve una fila hacia arriba de la posición que se le pasa por parámetro, y comprueba si se puede generar el barco hacia esta posición. Si el barco mide mas de una casilla y ya se a generado la primera casilla, ignorará la posición contraria de a donde se va a generar.

Ejemplo:

```
public static boolean compruebaArriba(int col, int fil, String[][] tableroTemporal, int tamaño,
String direccion) {
    boolean me_puedo_colocar = false;
    if (fil - 1 >= 0) {
        if (tableroTemporal[col][fil - 1].equals(" ")) {
            me_puedo_colocar = true;
        } else if (direccion.equals("vertical_abajo") && tamaño > 0) {
            me_puedo_colocar = true;
        }
    } else {
        if (fil == 0 && tableroTemporal[col][fil].equals(" ")) {
            me_puedo_colocar = true;
        }
    }
    return me_puedo_colocar;
}
```

Método compruebaAbajo() → Igual que el método anterior, pero comprobando una posición abajo e ignorando la superior si el barco mide mas de una casilla.

Método compruebaIzquierda() → Igual que el anterior ignorando la posición derecha, y comprobando la posición izquierda.

Método compruebaDerecha() → Igual que el anterior ignorando la posición izquierda, y comprobando la posición derecha.

Método `compruebaDiagonalAbajoDerecha()` → Igual que el anterior ignorando la posición diagonal arriba izquierda, y comprobando la posición diagonal inferior derecha.

Método `compruebaDiagonalAbajoIzquierda()` → Igual que el anterior ignorando la posición diagonal arriba derecha, y comprobando la posición diagonal inferior izquierda.

Método `compruebaDiagonalArribaDerecha()` → Igual que el anterior ignorando la posición diagonal inferior izquierda, y comprobando la posición diagonal superior derecha.

Método `compruebaDiagonalArribaIzquierda()` → Igual que el anterior ignorando la posición diagonal inferior derecha, y comprobando la posición diagonal superior izquierda.

Clase yate

Posee varios atributos

- **int cantidad** → Número de yates que se pueden generar en el tablero
- **int tamaño** → El tamaño del barco en este caso 2 casillas
- **String letra** → la letra del barco en este caso y
- **int[][]** → posiciones las posiciones que ocupa ese barco en el tablero
- **boolean hundido** → cuando es hundido el barco pasa a true
- **int tocado** → veces que a sido tocado

Clase Submarino

Posee varios atributos

- **int cantidad** → Número de submarinos que se pueden generar en el tablero
- **int tamaño** → El tamaño del barco en este caso 1 casilla
- **String letra** → la letra del barco en este caso s
- **int[][]** → posiciones las posiciones que ocupa ese barco en el tablero
- **boolean hundido** → cuando es hundido el barco pasa a true
- **int tocado** → veces que a sido tocado

Clase Transatlantico

Posee varios atributos

- **int cantidad** → Número de Transatlanticos que se pueden generar en el tablero
- **int tamaño** → El tamaño del barco en este caso 3 casillas
- **String letra** → la letra del barco en este caso t
- **int[][]** → posiciones las posiciones que ocupa ese barco en el tablero
- **boolean hundido** → cuando es hundido el barco pasa a true
- **int tocado** → veces que a sido tocado

Interfaz Barco

Me provee de la estructura que tendrá cada barco.

```

int getTamano();
int getCantidad();
char getLetra();
int[][] getPosiciones();
void setPosiciones(int[][] posiciones);
void setHundido(boolean hundido);
boolean getHundido();
void tocado();
int getTocado();

```

Pantallazos de ejecución del juego:

```

-----
a|      |
b|      |
c|      |
d|      |
e|      |
f|      |
g|      |
h|      |
i|      |
j|      |
-----
Vamos a colocar el submarino 0

Posición:

```

```

En que posición deseas que se mueva
0.- Vertical abajo
1.- Vertical arriba
2.- horizontal izquierda
3.- horizontal derecha
4.- diagonal arriba izquierda
5.- diagonal abajo izquierda
6.- diagonal arriba derecha
7.- diagonal abajo derecha

```

Opcion: 0

Colocado!

```

-----
a|s s      s|
b|      |
c|      y  |
d|      y  |
e|      |
f|      |
g|      |
h|      |
i|      |
j|      |
-----

```

Todos los barcos en posición...
0123456789

```

-----
a|      |
b|      |
c|      |
d|      |
e|      |
f|      |
g|      |
h|      |
i|      |
j|      |
-----
a|s s      s|
b|      |
c|      y  |
d|      y  |
e| t      |
f| t      |
g|t      |
h|      y  |
i|      y  |
j|      |
-----

```

Preparados para realizar el disparo.

Posición:

Preparados para realizar el disparo

Posición: a7

Tocado

Barco hundido

Todos los barcos en posición...
0123456789

```

-----
a|xxxxxxx* |
b|      |
c|xx      |
d|      |
e|      |
f|      |
g|      |
h|      |
i|      |
j|      |
-----

```

```

-----
a|s sx      *|
b|      x  |
c| x  y  |
d|  x y  |
e| t      |
f| t x  |
g|t      |
h|      y  |
i|x  y x  |
j|      x  |
-----

```