

# ISW1-Games

Ahoy! es el nuevo juego de aventuras piratas desarrollado por el estudio de videojuegos ISW1-Games.

Nuestro equipo estará a cargo de pulir el código antes de continuar el desarrollo para tener una buena base para seguir construyendo el juego.

## Ahoy!

Ahoy! Es un juego colaborativo donde una tripulación de piratas unirán fuerzas para enterrar un tesoro, a la vez que detallan en un mapa los pasos a seguir para que el mismo pueda ser encontrado por su Capitán.

El juego se divide en dos partes:

### 1. Enterrar el tesoro

La tripulación formada por diferentes tipos de piratas (ver Piratas) se moverán todos juntos por la isla hasta encontrar un lugar donde enterrar el tesoro.

Cada pirata tendrá una tarea según el tipo de pirata que sea y, en esta primera iteración, la tripulación estará formada siempre por un representante de cada tipo de pirata.

Luego de esta etapa, el resultado es un tesoro enterrado en la isla y un mapa que indica cómo encontrarlo.

### 2. Un Capitán, un mapa y un tesoro

En cualquier momento, un Capitán con un mapa del tesoro en su poder, podrá leerlo para llegar al tesoro y desenterrarlo. Es importante resaltar que ¡Solo un Capitán puede leer mapas de tesoros!

## Piratas

Hay diferentes tipos de piratas, cada unos con habilidades necesarias para esconder un tesoro en una isla:

- Corsarios: ¡son grandes cartógrafos! El corsario se hará cargo de ir dibujando el mapa a medida que se mueven por la isla.
- Filibusteros: ningún obstáculo los detiene. Ayudarán a la tripulación a abrirse paso por la isla, tirando abajo los árboles que se crucen en su camino.
- Bucaneros: serán los encargados de detectar el mejor lugar para enterrar el tesoro.

Volvemos a resaltar que en esta primera iteración del juego, la tripulación estará siempre formada por un corsario, un filibustero y un bucanero.

## Rangos

En nuestro juego simplificamos los rangos a solo dos:

- Capitán: Es el único pirata que puede leer mapas de tesoros.

- Simple pirata: No puede leer mapas de tesoros.

Tener en cuenta que cualquier pirata puede ser Capitán en cualquier momento. Y también un pirata Capitán puede perder su rango y pasar a ser un simple pirata.

## La Isla

La isla está representada por una grilla con un tamaño dado, por ej. 4x5. La misma tendrá celdas ocupadas por árboles (llamadas celdas `Tree`). Otras celdas estarán marcadas como lugares propicios para enterrar el tesoro: las llamaremos celdas `Dig` y los piratas Bucaneros son expertos en detectar estas celdas. Las demás celdas estarán libres.

La tripulación se podrá mover a cualquier celda, salvo a las ocupadas por los árboles que primero deberán ser derribados por el Filibustero de la tripulación y luego una vez que la celda esté libre podrá ser ocupada por la tripulación.

No es necesario controlar movimientos fuera del tablero.

## Trabajo a realizar:

Por suerte todo está desarrollado y hay tests que validan su funcionamiento.

El modelo presentado no tiene un buen diseño por varios motivos vistos en clase.

Se pide mejorar el modelo a partir de:

1. Sacar el código repetido del modelo y hacerlo más declarativo en los lugares que corresponda.
2. Sacar los ifs que puedan ser reemplazados por el uso de polimorfismo utilizando el criterio de decisión visto en clase.
3. Sacar el código repetido en los tests.
4. Mejorar el diseño en todo lo que considere necesario sin alterar el funcionamiento (objetos completos y correctos desde su creación, no romper encapsulamiento, sacar mensajes no usados, borrar variables no usadas, categorizar mensajes, crear jerarquías de clase, usar mensajes de colecciones ya existentes, etc.)

### Importante:

Usar las heurísticas de diseño vistas hasta ahora (buenos nombres, métodos cortos y categorizados, etc.)

Todos los tests deben seguir funcionando.

Recomendamos no modificar funcionalmente los tests (únicamente para refactors)

## Entrega:

1. Entregar por mail el fileout de la categoría de clase **ISW1-2022-2C-1erParcial** que debe incluir toda la solución (modelo y tests). El archivo de fileout se debe llamar: **ISW1-2022-2C-1erParcial.st**
2. Entregar también por mail el archivo que se llama **CuisUniversity-nnnn.user.changes**

3. Probar que el archivo generado en 1) se cargue correctamente en una imagen “limpia” (o sea, sin la solución que crearon) y que todo funcione correctamente. Esto es fundamental para que no haya problemas de que falten clases/métodos/objetos en la entrega.
4. Realizar la entrega enviando mail a la lista de Docentes: [ingsoft1-doc@dc.uba.ar](mailto:ingsoft1-doc@dc.uba.ar) con el **Subject: LU nnn-aa - Solución 1er Parcial 2c2022**
5. Subir a sus repos grupales los archivos **CuisUniversity-nnnn.image** y **CuisUniversity-nnnn.changes**. Debe **zippearlos** previamente para reducir su tamaño o podría dejar sin espacio disponible a sus compañeros.
6. Deberán subirlos al main branch de sus respectivos repos (tenga en cuenta hacer pull antes de ser necesario), y al subdirectorio **/Parcial1/LUnnn-aa/**

### IMPORTANTE:

No retirarse sin tener el ok de los docentes de haber recibido el mail con la resolución.