Universidad del Istmo de Guatemala
Facultad de Ingenieria
Ing. en Sistemas
Informatica 2
Prof. Ernesto Rodriguez - erodriguez@unis.edu.gt

Examen Final

Tiempo de resolución: 90 minutos

Instrucciones: Responder las preguntas que se presentan a continuación y hacer los ejercicios que se presenten a continuación. El parcial debe ser entregado como un archivo comprimido a traves de blackboard.

Iniciación

Crear una solución llamada "ExamenFinal". Dentro de dicha solución crear dos carpetas: "Examen" y "ExamenTests". Dentro de la carpeta "Examen", crear un proyecto tipo *console* y dentro de "ExamenTests" crear un proyecto tipo "XUnit". Si lo desea, puede utilizar otra herramienta aparte de "XUnit" para crear las pruebas unitarias.

El objetivo de este examen es crear una versión basica del juego de mesa "Battleship". El proceso se llevara a cabo por pasos.

Interfaz IBarco (10%)

Crear una interfaz llamada "IBarco". Dicha interfaz debe tener los siguientes metodos y propiedades:

Metodos:

Nombre	**Tipo**	**Descripción**
Atacar	`int` → `bool`	Intenta colocar un proyectil en el espacio del barco indicado por el parametro numerico. Si la longitud del barco es mayor que el espacio atacado y dicho espacio no ha sido atacado previamente, el barco debe marcar dicho espacio como atacado. Si todos los espacios del barco ya fueron atacados, el barco se considera hundido. Retorna `true` si un espacio del barco fue atacado o `false` de lo contrario.
EstaHundido	`void` → `bool`	Retorna `true` si todos los espacios del barco ya fueron atacados o `false` de lo contrario.

Propiedades

Nombre	Tipo	Accesibilidad	Descripción
Longitud	`int`	get;	Cantidad de especiós que ocupa el barco en el tablaro. También es la candidad de espaciós del barco que deben atacarse para que el barco se hunda.
X	`int`	get;	La coordenada "X" del barco en el tablero.
Y	`int`	get;	La coordenada "Y" del barco en el tablero.
Orientacion	`int`	get;	La dirección en hacia la cual esta orientada el barco. Puede ser horizontal (1) o vertical (2).

Interfaz ITablero (10%)

Crear una interfaz llamada "ITablero". Esta interfaz representa un espacio donde se pueden colocar barcos. El tablero tambien implementa la logica para determinar si un barco es atacado por un proyectil. La interfaz debe tener los siguientes metodos y propiedades:

Metodos:

Nombre	Tipo	Descripción
ColocarBarco	`IBarco` $\rightarrow$ `bool`	Intenta agregar un barco al tablero. Para ello, es necesario verificar que el espacio donde se intenta agregar el barco no este ocupado por otro barco. Tambien se debe tomar en cuenta la longitud y orientación del barco que se esta agregando y los barcos que ya estan en el tablero y asegurarse que no se cruzen. Por ultimo, se debe asegurar que la cantidad de barcos en el tablero no exceda la capacidad del tablero. Retorna `true` si el barco fue agregado exitosamente o `false` de lo contrario.
Atacar	`int` $\otimes$ `int` $\rightarrow$ `bool`	Este metodo intenta lanzar un proyectil a algún barco que se encuentre en el tablero. Este metodo debe considerar todos los barcos, su posición y orientación. Debe llamar al metodo "Atacar" de un barco en caso que dicho barco se encuentre dentro de las coordenadas que se especificaron como parametros. Retorna `true` si algún espacio de algún barco fue atacado o `false` de lo contrario.
EstaConcluido	`void` $\rightarrow$ `bool`	Este metodo revisa todos los barcos en el tablero. Retorna `true` si todos los barcos han sido hundidos o `false` de lo contrario.

Propiedades

Nombre	Tipo	Accesibilidad	Descripción
Capacidad	`int`	get;	La cantidad de barcos que se puede colocar en el tablero.

Implementación de IBarco (20%)

Crear una clase llamada "Barco". Esta clase debe implementar la interfaz "IBarco" adecuadamente. Asegurarse que el comportamiento de sus metodos y propiedades vaya acorde con las reglas.

Implementación de ITablero (30%)

Crear una clase llamada "ITablero". Esta clase debe implementar la interfaz "ITablero" adecuadamente. Asegurarse que el comportamiento de sus metodos y propiedades vaya acorde con las reglas.

Tests "ColocarBarco" (10%)

Crear una prueba unitaria en donde se intenta agregar un barco a un tablero donde dicho barco intersecta al barco siendo agregado. Por ejemplo, el tablero puede tener un barco de logitud 3 en la posición (1,2), orientado horizontalmente y se intenta colocar un barco en la posici'on (2,1), orientado verticalmente de longitud 4. Dicha operación no es permitida y este metodo debe retornar `false`.

Tests "Atacar" (10%)

Crear una prueba unitaria para verificar que el metodo "Atacar" de "ITablero" fue implementado correctamente. En este caso, se colocara un barco en la posición (3,4) orientado verticalmente de longitud 4. Se intentara atacar la posición (3,7), la cual esta ocupada por dicho barco. El metodo atacar debe retornar `true`.

Tests "EstaConcluido" (10%)

Crear una prueba unitaria para verificar que el metodo "EstaConcluido" funciona correctamente. Esta prueba debe iniciar un tablero con barcos. Verificar que "EstaConcluido" retorna `false`, luego debe hundir todos los barcos y verificar que "EstaConcluido" retorna `true`.