

Universidad Rafael Landívar
Facultad de ingeniería
Ingeniería industrial
Introducción a la programación
Inga. Damaris Campos



Universidad
Rafael Landívar
Tradición Jesuita en Guatemala

Proyecto 2

Fase 1

Diéguez Martínez, Sofia Elizabeth
Carné: 1092823

Guatemala, 31 de octubre del 2023

ANALISIS Y DISEÑO DEL PROGRAMA

Nombre y descripción de las clases que utilizará.

Tablero:

Descripción: Esta clase representa el tablero de juego en el que se colocan las embarcaciones y se registran los disparos.

Atributos:

Tamaño del tablero (generalmente una cuadrícula)

Matriz para rastrear el estado de cada celda (agua, barco, barco golpeado)

Métodos:

Colocar barcos en el tablero.

Registrar disparos y verificar si un barco ha sido impactado o si el juego ha terminado.

Barco:

Descripción: Esta clase representa un barco en el juego.

Atributos:

Tipo de barco (portaaviones, acorazado, submarino, etc.).

Tamaño del barco.

Posición en el tablero (coordenadas de inicio y dirección).

Estado (intacto, golpeado, hundido).

Métodos:

Verificar si el barco ha sido golpeado o hundido.

Jugador:

Descripción: Representa a un jugador en el juego de Batalla Naval.

Atributos:

Nombre del jugador.

Tablero propio (para colocar barcos y rastrear disparos).

Tablero del oponente (para rastrear disparos realizados contra el oponente).

Métodos:

Colocar barcos en su tablero.

Realizar disparos en el tablero del oponente.

Juego:

Descripción: Gestiona el flujo del juego y controla el turno de los jugadores.

Atributos:

Jugadores involucrados.

Estado del juego (en curso, jugador 1 gana, jugador 2 gana, empate, etc.).

Métodos:

Controlar el turno de los jugadores.

Determinar si un jugador ha ganado.

Main:

Descripción: La clase principal que inicia el juego y gestiona la interacción con los jugadores.

Atributos: Puede incluir configuraciones iniciales, como el tamaño del tablero y la cantidad de barcos por jugador.

Métodos:

Iniciar el juego.

Mostrar el estado del tablero y las instrucciones.

Nombre, descripción y tipo de dato de los Atributos que contendrá las clases que utilizará.

Clase Tablero:

- ****Atributo tamaño:**** Representa el tamaño del tablero como un entero que indica el número de filas y columnas.
- ****Tipo de dato:**** int
- ****Atributo matriz:**** Se encarga de rastrear el estado de cada celda del tablero, indicando si hay agua, un barco o un barco golpeado.
- ****Tipo de dato:**** Puede ser una matriz bidimensional o una estructura de datos adecuada, como una lista de listas.
- ****Atributo barcos:**** Contiene una lista de objetos de la clase Barco que se encuentran en el tablero.
- ****Tipo de dato:**** Una lista de objetos de la clase Barco.

Clase Barco:

- ****Atributo tipo:**** Describe el tipo de barco, por ejemplo, "Portaaviones".
- ****Tipo de dato:**** str
- ****Atributo tamaño:**** Indica el tamaño del barco, es decir, el número de celdas que ocupa en el tablero.
- ****Tipo de dato:**** int
- ****Atributo posicion:**** Representa la posición del barco en el tablero. Puede ser una tupla que incluye las coordenadas de inicio y la dirección.
- ****Tipo de dato:**** Dependerá de la representación de la posición elegida, como una tupla, una lista o un objeto personalizado.
- ****Atributo estado:**** Define el estado del barco, ya sea intacto, golpeado o hundido.
- ****Tipo de dato:**** Puede ser una cadena de texto o valores numéricos para representar los diferentes estados.

Clase Jugador:

- ****Atributo nombre:**** Almacena el nombre del jugador.
- ****Tipo de dato:**** str
- ****Atributo tablero_propio:**** Representa el tablero propio del jugador.
- ****Tipo de dato:**** Es un objeto de la clase Tablero.
- ****Atributo tablero_oponente:**** Indica el tablero del oponente, donde el jugador realiza disparos.
- ****Tipo de dato:**** También es un objeto de la clase Tablero.

Clase Juego:

- ****Atributo jugadores:**** Contiene una lista de objetos de la clase Jugador que están involucrados en el juego.
- ****Tipo de dato:**** Una lista de objetos de la clase Jugador.
- ****Atributo estado_juego:**** Define el estado actual del juego, ya sea que esté en curso, que un jugador haya ganado, que haya empate, etc.

- ****Tipo de dato:**** Puede ser una cadena de texto o un valor numérico que representa el estado del juego.

Clase Main:

- ****Atributo tamano_tablero:**** Representa el tamaño inicial del tablero, es decir, el número de filas y columnas.
- ****Tipo de dato:**** int
- ****Atributo cantidad_barcos:**** Indica la cantidad de barcos que se deben colocar en el tablero de cada jugador.
- ****Tipo de dato:**** int
- ****Otros atributos:**** Pueden existir otros atributos según las configuraciones iniciales del juego que desees definir.

Nombre, descripción y si retornará algún valor los métodos que contendrá las clases que utilizará.

Clase `Tablero`:

1. ****colocar_barco()****

- Descripción: Este método se utiliza para posicionar un barco en el tablero en una ubicación específica.

- Retorna: Puede no necesariamente devolver un valor, ya que su principal función es actualizar la matriz del tablero.

2. ****registrar_disparo()****

- Descripción: Registra un disparo en una posición determinada del tablero y verifica si impacta a un barco.

- Retorna: Puede proporcionar información sobre si el disparo afectó a un barco y el estado actual del barco afectado.

Clase `Barco`:

1. ****verificar_estado()****

Clase `Jugador`:

1. ****colocar_barcos()****

- Descripción: Permite al jugador colocar sus barcos en su tablero personal.
- Retorna: Puede no necesariamente retornar un valor, ya que su principal función es actualizar el tablero propio del jugador.

2. ****realizar_disparo()****

- Descripción: Permite al jugador llevar a cabo un disparo en el tablero del oponente.
- Retorna: Puede proporcionar información sobre si el disparo impactó a un barco y el estado actual del barco afectado.

Clase `Juego`:

1. ****iniciar_juego()****

- Descripción: Inicia el juego y gestiona su flujo, incluyendo los turnos de los jugadores.
- Retorna: Puede devolver información sobre el resultado del juego, como el ganador o si hay empate.

Clase `Main`:

1. ****configurar_juego()****

- Descripción: Este método se utiliza para establecer las configuraciones iniciales del juego, como el tamaño del tablero y la cantidad de barcos por jugador.
- Retorna: Puede devolver una instancia de la clase `Juego` con la configuración inicial.

Condiciones y restricciones que debe tener su programa, Algoritmo que implementará en el programa.

1. ****Tamaño del Tablero****: Es fundamental que el tamaño del tablero sea adecuado, lo que significa que debe ser lo suficientemente grande para albergar los barcos y

permitir un juego interesante, pero no tan grande como para ser abrumador. Una cuadrícula típica podría ser de 10x10.

2. ****Cantidad y Tamaño de Barcos****: Debes definir cuántos barcos se permiten en el juego y sus tamaños. Por ejemplo, podrías permitir un barco "Portaaviones" de 5 celdas, un "Acorazado" de 4 celdas, un "Submarino" de 3 celdas, y así sucesivamente.

3. ****Restricciones de Colocación de Barcos****: Es importante que los barcos no se superpongan ni se salgan del tablero. También, asegúrate de que no se coloquen en posiciones diagonales, lo que podría dificultar el juego.

4. ****Orden de los Turnos****: Establece un orden de turnos entre los jugadores para determinar quién juega primero y asegúrate de que los jugadores se alternen en sus turnos de manera justa.

5. ****Reglas de Disparo****: Define las reglas para realizar disparos, como el formato de entrada de coordenadas (por ejemplo, "A3" o "2,5") y si un jugador puede disparar repetidamente a la misma celda.

6. ****Detección de Impactos y Hundimientos****: Implementa la lógica para detectar cuándo un disparo impacta un barco y cuándo un barco está completamente hundido. Debes llevar un registro del estado de cada barco.

7. ****Finalización del Juego****: Define las condiciones para determinar cuándo el juego termina, generalmente cuando todos los barcos de un jugador han sido hundidos.

8. ****Mensaje de Resultado****: Cuando el juego finaliza, muestra un mensaje de resultado que indique quién ganó o si el juego terminó en empate.

9. ****Validación de Entrada del Jugador****: Asegúrate de que los jugadores ingresen datos válidos al colocar barcos o realizar disparos. Valida las coordenadas y garantiza que estén dentro de los límites del tablero.

10. ****Interfaz de Usuario****: Considera cómo presentar el juego a los jugadores, ya sea a través de una interfaz gráfica o en la línea de comandos, y cómo recopilar las entradas del usuario.