

Albert Centeno

1220823

30/10/2023

Laboratorio de introducción a la programación sec. 17

Proyecto no. 2: Batalla naval

- **Nombre y Descripción de las Clases:**

1. “Barco”:

- Descripción: Esta clase representa un barco en el juego de batalla naval.

2. “Tablero”:

- Descripción: Esta clase representa el tablero de juego de batalla naval.

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

Barco:

- “nombre” (str): Tipo de dato string. Representa el nombre del barco.

- “tamaño” (int): Tipo de dato entero. Representa el tamaño del barco en número de celdas.

- “posición” (tuple): Tipo de dato tupla de enteros. Representa la posición del barco en el tablero, donde el primer elemento de la tupla es la fila y el segundo elemento es la columna.

- “orientación” (str): Tipo de dato string. Representa la orientación del barco, puede ser "horizontal" o "vertical".

- “estado” (str): Tipo de dato string. Representa el estado del barco, puede ser "intacto", "golpeado" o "hundido".

Tablero:

- “tamaño” (int): Tipo de dato entero. Representa el tamaño del tablero, es decir, la cantidad de filas y columnas.
- “matriz” (list): Tipo de dato lista de listas. Representa el tablero como una matriz, donde cada elemento de la lista interna puede contener diferentes tipos de datos para representar el estado de una celda del tablero.

- **Nombre, descripción y si retornará algún valor los métodos que contendrá las clases que utilizará:**

“Barco”:

1. “__init__(self, nombre, tamaño)”:

- Descripción: Constructor que inicializa los atributos del barco.
- Parámetros:
 - “nombre” (str): Nombre del barco.
 - “tamaño” (int): Tamaño del barco.
- Retorno: No retorna ningún valor.

2. “colocar_barco(self, fila, columna, orientación)”:

- Descripción: Coloca el barco en la posición especificada en el tablero.
- Parámetros:
 - “fila” (int): Fila en la que se coloca el barco.
 - “columna” (int): Columna en la que se coloca el barco.
 - “orientación” (str): Orientación del barco ("horizontal" o "vertical").
- Retorno: No retorna ningún valor.

3. “verificar_ataque(self, fila, columna)”:

- Descripción: Verifica si el ataque alcanza al barco y actualiza su estado en consecuencia.

- Parámetros:

- “fila” (int): Fila del ataque.

- “columna” (int): Columna del ataque.

- Retorno: Retorna True si el ataque alcanza al barco, False si no alcanza.

“Tablero”:

1. “__init__(self, tamaño)”:

- Descripción: Constructor que inicializa el tablero con el tamaño especificado.

- Parámetros:

- “tamaño” (int): Tamaño del tablero (número de filas y columnas).

- Retorno: No retorna ningún valor.

2. “mostrar_tablero(self)”:

- Descripción: Muestra el tablero en la consola.

- Parámetros: No tiene parámetros.

- Retorno: No retorna ningún valor.

3. “colocar_barco(self, barco)”:

- Descripción: Coloca un barco en el tablero en la posición especificada por el objeto “Barco”.

- Parámetros:

- “barco” (Barco): Objeto de la clase “Barco” que se coloca en el tablero.

- Retorno: No retorna ningún valor.

4. “verificar_ataque(self, fila, columna)”:

- Descripción: Verifica si hay un barco en la posición de ataque y devuelve True si es un acierto, False si es un fallo.

- Parámetros:

- “fila” (int): Fila del ataque.

- “columna” (int): Columna del ataque.

- Retorno: Retorna True si el ataque alcanza un barco, False si no alcanza.

- **Condiciones y Restricciones:**

- Los jugadores deben ingresar coordenadas válidas para los disparos.

- Los nombres de los barcos y los jugadores deben ser únicos.

- Los barcos no pueden superponerse en el tablero.

- El juego debe manejar adecuadamente las situaciones de empate y seguir permitiendo turnos hasta que haya un ganador.

- **Algoritmo para el Juego de Batalla Naval:**

1. Inicio del Juego

- Crear una instancia de la clase “Tablero” para cada jugador, especificando el tamaño del tablero.

- Crear una lista de instancias de la clase “Barco” para cada jugador, especificando nombre y tamaño para cada barco.

- Para cada jugador, colocar sus barcos en el tablero utilizando el método “colocar_barco”.

2. Inicio del Turno:

- Alternar turnos entre los jugadores.

- Para cada turno:

- Mostrar el tablero del jugador actual.
- Solicitar al jugador las coordenadas de disparo (fila y columna).

3. Verificación del Disparo:

- Verificar si el disparo alcanza algún barco en el tablero del oponente utilizando el método “verificar_disparo”.
- Si es un acierto:
 - Marcar la posición del tablero del oponente como "golpeado".
 - Verificar si el barco ha sido completamente hundido.
 - Si el barco está hundido, marcar todas las posiciones del barco como "hundido".
 - Mostrar un mensaje indicando que el barco ha sido hundido.
 - Mostrar un mensaje indicando que el disparo fue un acierto.
- Si es un fallo:
 - Marcar la posición del tablero del oponente como "fallado".
 - Mostrar un mensaje indicando que el disparo fue un fallo.

4. Fin del Turno:

- Verificar si todos los barcos del oponente han sido hundidos.
- Si todos los barcos del oponente están hundidos:
 - Anunciar al jugador actual como ganador.
- Si no todos los barcos del oponente están hundidos:
 - Regresar al paso 2 para el siguiente turno.

5. Fin del Juego:

- Mostrar el tablero final de ambos jugadores con la ubicación de los barcos y los disparos realizados.
- Mostrar un mensaje indicando el resultado final del juego.