

Reporte de laboratorio 2

Profesor: Luis Campos Duarte

Estudiante: Leonardo Solano

Carnet: C4k103

La clase JuegoBatalla se encarga de gestionar el juego de batalla entre robots. Dentro de esta clase, se define un atributo llamado robots, que es un arreglo de objetos de la clase Robot. Este arreglo almacena los robots creados por el jugador. En el constructor, se recibe un parámetro cantidadDeRobots que indica cuántos robots se crearán al inicio del juego, pero el código tiene un error en esta parte, ya que no se está inicializando correctamente el arreglo de robots.

El método iniciarCreacionRobots permite que el usuario cree los robots. En este método, se llama a otro método, creacionRobots, donde se pide al jugador que ingrese la cantidad de robots que desea crear. El programa luego crea un arreglo de robots y, para cada uno, solicita el nombre del robot. Sin embargo, el código tiene algunos errores, como el uso incorrecto de la clase Scanner para leer los valores del usuario. También falta la implementación de la lógica de batalla, lo que significa que, por ahora, solo se están creando robots sin que puedan interactuar entre ellos.

Aunque la parte de la batalla no está implementada, la estructura general de la clase es clara y proporciona una base para crear robots y posiblemente añadir más funcionalidad en el futuro, como la mecánica de combate entre ellos.

La clase Robots define el comportamiento y los atributos de un robot en el juego. Cada robot tiene tres atributos principales: nombre, puntos de vida y poder de ataque. Estos atributos son inicializados en el constructor de la clase, el cual también asegura que los valores de puntos de vida estén dentro de un rango válido (entre 50 y 100) y que el poder de ataque esté entre 10 y 20. Si los valores ingresados por el jugador son menores o mayores que estos rangos, el constructor los ajusta automáticamente para que estén dentro de los límites definidos.

Además de los atributos, la clase incluye métodos Set y Get para modificar y obtener los valores de los atributos de cada robot. Los setters permiten cambiar los valores de nombre, puntos de vida y poder de ataque después de que el robot

ha sido creado, mientras que los getters permiten obtener estos valores para ser utilizados en otras partes del código, como al mostrar información del robot o al hacer cálculos en la batalla.

Uno de los métodos importantes en esta clase es atacar, que permite a un robot atacar a otro. Al llamar a este método, el robot atacante reduce los puntos de vida del robot objetivo en función de su poder de ataque. Si la vida restante del robot atacado es menor que 0, se ajusta a 0 para evitar que un robot tenga vida negativa. Este método actualiza la vida del robot atacado y es un componente clave para la mecánica de batalla del juego.

Finalmente, el método `estaVivo` verifica si un robot sigue con vida, basándose en si sus puntos de vida son mayores que 0. Si la vida es mayor que 0, el robot está vivo, de lo contrario, se considera muerto.

Se ha intentado, pero definitivamente no he logrado el objetivo, el código no está completo, pero esto me ayudó a entender el GitHub y la forma de crear los arrays.

Link de GitHub: <https://github.com/Leonardo-Solano/CI-0112.git>