

Problema I (15 puntos)

Escribe un programa que lea continuamente el nombre y la distancia de dos enemigos y determine cuál de ellos está más cerca. Imprime el nombre del enemigo más cercano.

Entrada: enemy1: "EnemyA" dist1: 10 enemy2: "EnemyB" dist2: 20	Salida: EnemyA
--	-------------------

Solucion:

Para realizar esta tarea de tal forma que el programa lea continuamente el nombre y la distancia de dos enemigos agregue una función (randomNum) que genera números random usando math.random y para redondear el número utilice math.floor. Luego de esto para que el código no se quede en un loop infinito, estableci la cantidad de iteraciones que puede tener.

Para crear el loop, agregue la variable enemy1= "Enemy A" y enemy2= "Enemy B" mas las 2 variables de distancia que tendrán como valor lo establecido en la función randomNum, luego imprime cuales son los números random de la iteración usando console.log.

Luego realice una condicional de tal manera que cuando la dist1 sea menor que dist2 nos imprima en consola enemy A, de no ser así, que imprima Enemy B.

Para terminar agregue un sumador de la variable iterations y una condicional que dice que cuando las iteraciones sean mayor o igual al límite que estableci de iteraciones realice un break.

Problema II (15 puntos)

En un juego de disparos, tu objetivo es atacar siempre al enemigo más cercano. Escribe un programa que lea continuamente el nombre y la distancia de dos enemigos y seleccione el objetivo más cercano para atacar. Si ambos enemigos están a la misma distancia, selecciona el que aparece primero.

Entrada: enemy1: "EnemyX" dist1: 15 enemy2: "EnemyY" dist2: 15	Salida: EnemyX
--	-------------------

Solucion:

Para el problema 2 utilice exactamente el mismo código que el problema 1, la única diferencia fue que en la condicional que revisa la distancia entre ambos enemigos, estableci que cuando la distancia1 sea menor o igual a la distancia2, que imprima Atacar a Enemy A, ya que es el primero que aparece..

Problema III (20 Puntos)

En una simulación de radar, los datos de los enemigos y sus distancias cambian dinámicamente. Escribe un programa que lea continuamente el nombre y la distancia de dos enemigos y determine cuál de ellos está más cerca. Asegúrate de que el programa maneje correctamente entradas donde las distancias pueden cambiar rápidamente. Utiliza prompt para ingresar los datos.

Entrada: enemy1: "Enemy1" Enemy2 dist1: 50 enemy2: "Enemy2" dist2: 30	Salida: Enemy2
enemy1: "Enemy1" dist1: 25 enemy2: "Enemy2" dist2: 45	Salida: Enemy1

Solucion:

Utilice la misma base de código que el problema 1, pero realice un prompt para pedirle al usuario las veces que quiere que el radar funcione ya que en mi código ya había generado una función que me devuelve números random.

Problema IV (20 Puntos)

En una situación de combate, además de la distancia, se te proporciona la prioridad de ataque de cada enemigo. Escribe un programa que lea continuamente el nombre, la distancia y la prioridad de dos enemigos y seleccione el enemigo con mayor prioridad para atacar. Si ambos enemigos tienen la misma prioridad, selecciona el más cercano.

Entrada: enemy1: "EnemyA" dist1: 20 priority1: 2 enemy2: "EnemyB" dist2: 15 priority2: 3	Salida EnemyB
--	------------------

Solucion:

Para resolver este problema, utilice la misma base, simplemente agregue la variable priority que genera valores random del 1 al 3 en cada iteración. Para realizar la lógica del problema, cree una condicional de tal manera que cuando la prioridad 1 sea mayor que la prioridad 2, entonces atacará a Enemy 1, de no ser así y prioridad 2 es mayor a 1, entonces atacará a Enemy 2, al no cumplirse ninguna de las dos, lo siguiente sería que prioridad 1 y 2 sean iguales, por lo que realizar otra condición que se ha utilizado en los problemas anteriores para determinar la distancia de cada enemigo.

Problema V (30 puntos)

Desarrolla un programa de simulación de radar que lea continuamente el nombre, la distancia, la velocidad y la prioridad de ataque de dos enemigos. El programa debe seleccionar el enemigo con mayor prioridad para atacar. Si ambos tienen la misma prioridad, selecciona el más cercano. Si las distancias son iguales, selecciona el enemigo con mayor velocidad.

Entrada: enemy1: "Enemy1" dist1: 30 priority1: 2 speed1: 10 enemy2: "Enemy2" dist2: 30 priority2: 2 speed2: 20	Salida: Enemy2
--	-------------------

Solucion:

Para resolver este problema, utilice la misma base del problema anterior, pero agregue la variable speed que genera valores random del 10 al 50 en cada iteración. Para realizar la lógica del problema, cree una condicional de tal manera que cuando la prioridad 1 sea mayor que la prioridad 2, entonces atacará a Enemy 1, de no ser así y prioridad 2 es mayor a 1, entonces atacará a Enemy 2, al no cumplirse ninguna de las dos, lo siguiente sería que prioridad 1 y 2 sean iguales, por lo que realizar otra condición que se ha utilizado en los problemas anteriores para determinar la distancia de cada enemigo, luego cree otra condición que si todas las anteriores no se cumplen, entonces evaluará la velocidad y primero determinará si speed 1 es mayor a speed 2, entonces se atacara al enemigo 1, si es lo contrario, entonces a enemigo 2.