

Nuestra empresa de software quiere desarrollar un nuevo videojuego FPS (shooter). Nuestra labor en esta empresa es encargarnos de las físicas y de las interacciones entre objetos y nos han encargado que programemos ciertas funcionalidades.

1. `getDamage(int speed, int armor, int distance) : return int` --> Esta función debe devolver el daño que ha recibido el personaje. La lógica será la siguiente.

- Si la distancia es inferior o igual a 3 metros se considera disparo cercano (ignora armor). $\text{Damage} = (\text{speed}/10)$
- Si la velocidad de la bala es superior a 500 m/s y la distancia inferior a 20 metros se considera disparo cercano (ignora armor). $\text{Damage} = (\text{speed}/10)$
- Si la velocidad de la bala es superior a 500 m/s y la distancia es superior a 20 metros, se aplica la siguiente fórmula $((\text{speed}/10) - \text{armor})$
- Para todos los demás casos $((\text{speed}/10) - \text{armor} + (\text{distance}/10))$

NOTA: Si el damage es negativo, se devuelve 0.

2. `isDeadByBodyShot(int life, int damage) : return boolean` --> Esta función simplemente resta el daño recibido a la vida que le queda al personaje. Si al hacer la resta la vida es 0 o inferior, el personaje muere.

3. `getNumObjects(boolean[][] map) : return int` --> Esta función se le pasa como parámetro un mapa de un escenario. Nuestra labor será la de devolver el número de objetos que se han encontrado en el escenario. El mapa es booleano, esto quiere decir que si es false en esa parte del mapa se encuentra paisaje sin relevancia, en cambio, si es true, en esa parte del mapa se encuentra un objeto con el que se puede interactuar.

4. `isHit(boolean[][] screen, int y, int x) : return boolean`. --> Está función recibe como parámetros la pantalla del jugador y donde ha disparado (coordenadas). La pantalla será parecida a la del ejemplo. Tendremos que devolver si el objetivo ha recibido impacto o no.

5. `isHeadShot(boolean[][] screen, int y, int x) : return boolean` --> Exactamente igual que la anterior, pero en este caso debemos saber si la posición en la que dispara es la columna central de la primera fila del enemigo. En ese caso, el disparo sería en la cabeza y devolveríamos un true.

6. `getEnemyPosition(boolean[][] screen) : return int[]` --> Esta función se le pasa como parámetro la pantalla del jugador cuando está en combate. En cada instante solo puede haber un enemigo ¿Como saber la posición del enemigo en la pantalla? Los enemigos siempre ocuparán un espacio de 3x5 de tal manera que se tiene que devolver el centro de la figura. La función devolverá: Y, X. return del ejemplo: 4, 3
NOTA: Si no existe enemigo en esos momentos, devolverá 0, 0

NOTAS:

1 = true
0 = false

y = número de fila de la matriz
x = número de columna

- Ejemplo nueva matriz: `new boolean[7][9]` --> matriz de 7 filas y 9 columnas de objetos tipo booleano.
- Acceso matriz y = 2, x = 3 `screen[2][3]` --> se accede a la posición de la matriz: fila 3 columna 4.

Ejemplo Screen:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0
0	0	1	1	1	0	0	0	0
0	0	1	1	1	0	0	0	0
0	0	1	1	1	0	0	0	0
0	0	1	1	1	0	0	0	0

Ejemplo Enemy (siempre es 5x3):

1	1	1
1	1	1
1	1	1
1	1	1
1	1	1

Ejemplo Map:

0	0	1	0	0	1
1	0	0	0	1	0
0	0	1	0	0	0
0	0	0	0	0	0
1	0	1	0	1	0
0	0	0	0	0	0
0	0	0	1	1	1