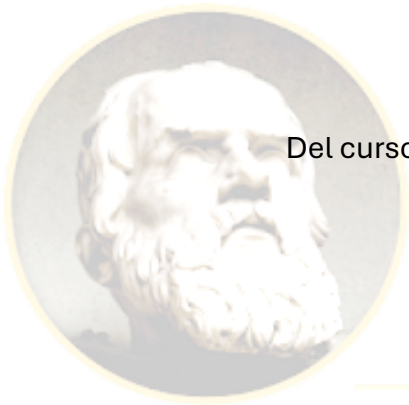


Universidad Galileo
Facultad de ingeniería de Sistemas,
informática y Ciencias de la computación
Diplomado en Desarrollo Fullstack



Tarea 9

Del curso de Introducción a la Programación

Galileo
UNIVERSIDAD
La Revolución en la Educación

Luis Diego Orozco López
24011289
Sección T

Guatemala, 2 de julio de 2024

Tabla de Contenido

Resolución de problemas	3
Problema 1	3
Explicación	3
Código y salida.....	3
Ejercicio 2	4
Explicación	4
Código y salida.....	4
Ejercicio 3	5
Explicación	5
Codigo.....	5
Ejecución	5
Salida	6
Problema 4.....	7
Explicación	7
Código y salida.....	8
Problema 5.....	9
Explicación	9
Código y salida.....	10

Resolución de problemas

Problema 1

Explicación

Se crearon dos objetos con el nombre de cada enemigo. Cada uno contiene los atributos nombre y distancia.

La función masCerca toma dos parámetros x e y que son los dos objetos enemigos. La función compara con un operador ternario cual de las distancias de cada uno de los objetos es menor a la otra. La distancia que sea menor será retornada, ya que eso indica que el objeto con dicha distancia es el mas cercano.

En caso de que las distancias sean iguales. Se retorna el nombre del objeto x, que seria el objeto que se tomó primero.

Con console.log se llama a la función, pasándole los objetos que serán comparados después

Código y salida

```
JS Problema 1.js > ...
1  let enemy1 = {
2    nombre: "Enemy A",
3    dist: 10
4  };
5
6  let enemy2 = {
7    nombre: "Enemy B",
8    dist: 20
9  };
10
11 function masCerca (x, y){
12   return x.dist <= y.dist ? x.nombre : y.nombre;
13 }
14
15 console.log(masCerca(enemy1, enemy2));
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Running] node "c:\Users\do30\OneDrive\Documentos\Galileo\Intro a la progra\Tarea 9\Problema 1.js"

Enemy A

[Done] exited with code=0 in 0.305 seconds

Ejercicio 2

Explicacion

Se crearon dos objetos con el nombre de cada enemigo. Cada uno contiene los atributos nombre y distancia.

La función masCerca toma dos parámetros x e y que son los dos objetos enemigos. La función compara con un operador ternario cuál de las distancias de cada uno de los objetos es menor a la otra. La distancia que sea menor será retornada, ya que eso indica que el objeto con dicha distancia es el más cercano.

En caso de que las distancias sean iguales. Se retorna el nombre del objeto x, que sería el objeto que se tomó primero.

Con console.log se llama a la función, pasándole los objetos que serán comparados después

Código y salida

```
JS Problema 2.js > [?] enemy2
1  let enemy1 = {
2    nombre: "Enemy A",
3    dist: 20
4  };
5
6  let enemy2 = {
7    nombre: "Enemy B",
8    dist: 20
9  };
10
11 function masCerca (x, y){
12   return x.dist <= y.dist ? x.nombre : y.nombre;
13 }
14
15 console.log(masCerca(enemy1, enemy2));
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Running] node "c:\Users\do30\OneDrive\Documentos\Galileo\Intro a la progra\Tarea 9\Problema 2.js"
Enemy A

[Done] exited with code=0 in 0.18 seconds

Ejercicio 3

Explicacion

Este programa es igual al anterior con la diferencia de que el nombre y la distancia del enemigo se toman desde el prompt.

Codigo

```
JS Problema 3.js > enemy2
1  let enemy1 = {
2      nombre: "",
3      dist: 0
4  };
5
6  let enemy2 = {
7      nombre: "",
8      dist: 0
9  };
10
11  enemy1.nombre = prompt("Nombre del enemigo 1");
12  enemy1.dist = parseInt(prompt("Distancia del enemigo 1"));
13  enemy2.nombre = prompt("Nombre del enemigo 2");
14  enemy2.dist = parseInt(prompt("Distancia del enemigo 2"));
15
16  function masCerca (x, y){
17      return x.dist <= y.dist ? x.nombre : y.nombre;
18  }
19
20  console.log(masCerca(enemy1, enemy2));
```

Ejecución

This page says

Nombre del enemigo 1

OK

Cancel

This page says

Distancia del enemigo 1

OK

Cancel

This page says

Nombre del enemigo 2

OK

Cancel

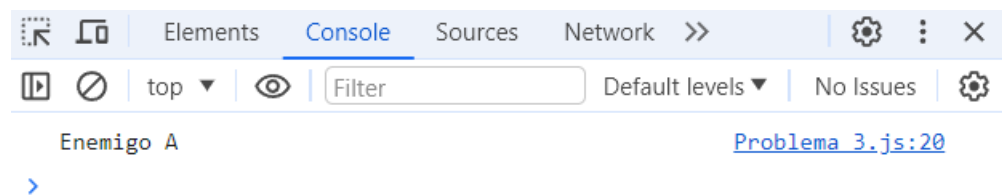
This page says

Distancia del enemigo 2

OK

Cancel

Salida



Problema 4

Explicación

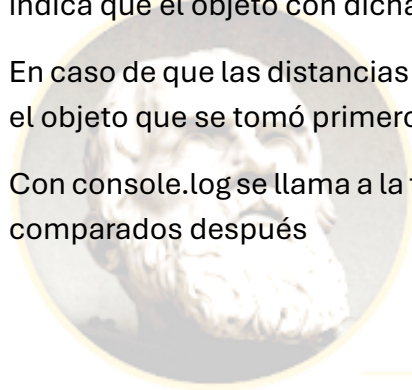
Se crearon dos objetos con el nombre de cada enemigo. Cada uno contiene los atributos nombre, distancia y prioridad.

La función llamada `masImportante` recibe dos parámetros `x` e `y` que son los dos objetos enemigos. La función utiliza un operador ternario compuesto para comparar las prioridades. Si la prioridad de `x` es mayor a la prioridad de `y`, se devuelve `x` y viceversa. Sin embargo, si las prioridades son iguales, se pasan los dos objetos a la función `masCerca`.

La función `masCerca` toma dos parámetros `x` e `y` que son los dos objetos enemigos. La función compara con un operador ternario cuál de las distancias de cada uno de los objetos es menor a la otra. La distancia que sea menor será retornada, ya que eso indica que el objeto con dicha distancia es el más cercano.

En caso de que las distancias sean iguales. Se retorna el nombre del objeto `x`, que sería el objeto que se tomó primero.

Con `console.log` se llama a la función `masImportante`, pasándole los objetos que serán comparados después



La Revolución en la Educación

Código y salida

```
JS Problema 4.js > masImportante
1  let enemy1 = {
2      nombre: "Enemy A",
3      dist: 10,
4      prioridad: 2
5  };
6
7  let enemy2 = {
8      nombre: "Enemy B",
9      dist: 20,
10     prioridad: 3
11 };
12
13 function masImportante(x,y){
14     return x.prioridad > y.prioridad ? x.nombre
15           : x.prioridad < y.prioridad ? y.nombre
16           : masCerca(x,y);
17 }
18
19 function masCerca (x, y){
20     return x.dist <= y.dist ? x.nombre : y.nombre;
21 }
22
23 console.log(masImportante(enemy1, enemy2));
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Running] node "c:\Users\do30\OneDrive\Documentos\Galileo\Intro a la progra\Tarea 9\Problema 4.js"
Enemy B

[Done] exited with code=0 in 0.164 seconds

Problema 5

Explicación

Se crearon dos objetos con el nombre de cada enemigo. Cada uno contiene los atributos nombre, distancia y prioridad y velocidad (speed).

La función llamada masImportante recibe dos parámetros x e y que son los dos objetos enemigos. La función utiliza un operador ternario compuesto para comparar las prioridades. Si la prioridad de x es mayor a la prioridad de y, se devuelve x y viceversa. Sin embargo, si las prioridades son iguales, se pasan los dos objetos a la función masCerca.

La función masCerca toma dos parámetros x e y que son los dos objetos enemigos. La función compara con un operador ternario cuál de las distancias de cada uno de los objetos es menor a la otra. La distancia que sea menor será retornada, ya que eso indica que el objeto con dicha distancia es el más cercano. En caso de que las distancias sean iguales, se procede a llamar a la función masVeloz.

La función masVeloz compara el atributo speed de cada objeto para determinar cual es mayor. La función retornará el objeto cuyo atributo speed sea mas grande ya que eso indica que es el mas veloz. Si son iguales, retornara el primero.

Con console.log se llama a la función masImportante, pasándole los objetos que serán comparados después

Código y salida

```
JS Problema 5.js > masImportante
1  let enemy1 = {
2      nombre: "Enemy A",
3      dist: 30,
4      prioridad: 2,
5      speed: 10
6  };
7
8  let enemy2 = {
9      nombre: "Enemy B",
10     dist: 30,
11     prioridad: 2,
12     speed: 20
13 };
14
15 function masImportante(x,y){
16     return x.prioridad > y.prioridad ? x.nombre
17           : x.prioridad < y.prioridad ? y.nombre
18           : masCerca(x,y);
19 }
20
21 function masCerca (x, y){
22     return x.dist > y.dist ? y.nombre
23           : x.dist < y.dist ? x.nombre
24           : masVeloz(x,y);
25 }
26
27 function masVeloz(x,y){
28     return x.speed >= y.speed ? x.nombre : y.nombre;
29 }
30
31 console.log(masImportante(enemy1, enemy2));
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Running] node "c:\Users\do30\OneDrive\Documentos\Galileo\Intro a la progra\Tarea 9\Problema 4.js"

Enemy B

[Done] exited with code=0 in 0.164 seconds

La Revolución en la Educación