Lendsum

# Prueba técnica programador senior

Disponemos de un api web para controlar un terreno de juego tipo cuadricula donde el **player** deberá “cazar” los **targets.** Además, el escenario también puede contener **bloques** que impiden avanzar al player hacia la casilla bloqueada.

La capacidad de movimiento del player viene dada por los cuatro ejes cardinales:

-arriba (y=y-1)

-derecha (x=x+1)

-abajo (y=y+1)

-izquierda (x=x-1)

El API está publicada en la dirección facilitada por correo. Está disponible un front-end generado por swagger para facilitar la tarea de comprensión del problema en el endpoint

<http://94.23.84.144:54818/swagger>

El Api se compone de tres llamadas:



Recibe la información del escenario con el Id dado. El output es un json con:

-una entrada llamada **ítems**, de tipo array que contiene una lista de elementos de tipo **ScenarioItem**

-El ScenarioItem contiene las siguientes propiedades:

-x: coordenada X donde está el ítem

-y: coordenada Y donde está el ítem

-itemType:

- “10” para el ítem de tipo player

- “20” para el ítem de tipo block

- “30” para el ítem de tipo target

Ejemplo -: En el siguiente json podemos ver los dos bloqueos, el player (p) y el target (x) representado gráficamente.

{

"items": [

{

"x": 1,

"y": 2,

"itemType": 20

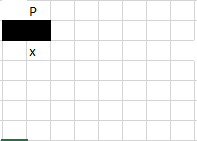
},

{

"x": 2,

"y": 2,

"itemType": 20

 },

{

"x": 2,

"y": 3,

"itemType": 30

},

{

"x": 3,

"y": 1,,

"itemType": 10

}

],

"player": {

"x": 2,

"y": 1,

"itemType": 10

},

"score": 0,

"movements": 1,

"uid": "00000000-0000-0000-0000-000000000001",

"version": 3,

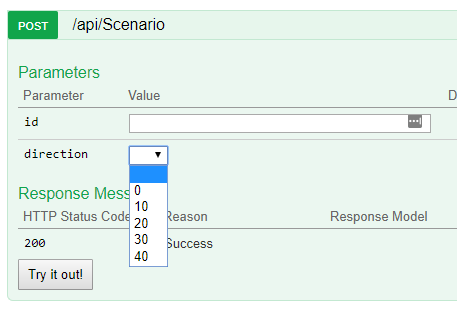
"creationDate": "2017-09-28T14:11:03.3030821Z",

"readyForSnapShot": false,

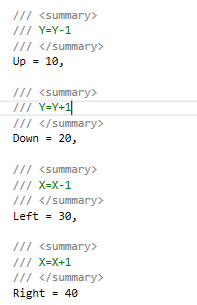
"snapShotTypeVersion": 0

}

Un escenario solo tiene un player pero puede tener multiples bloqueos y targets.

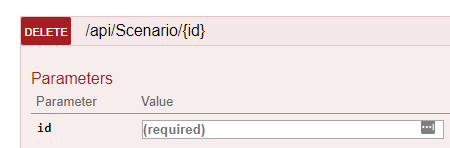


El post lo usaremos para mover al player. El significado del enum es el siguiente:



En el ejemplo anterior, si queremos que el player atrape al target, tenemos que enviar 4 posts (Right, Down, Down, Left).

El parámetro Id es el scenario.



Sirve para resetear el escenario, volverlo a la posición inicial tanto el player como los targets:

# Puntos de la prueba

## Punto 1 (nivel I’m too Young to die)

Se pide realizar una aplicación de consola/desktop/mobile/web que **represente** el escenario dado un Id, pintando el **player**, los **bloqueos** y los **targets**, así como el **score actual** y **numero de movimientos** realizados por el player. Nos vale ASCII ART en una aplicación de consola, Windows Form, WPF, ASP.NET, Ionic, Xamarin Forms, un plugin del Excel o lo que mejor domines.

## Punto 2 (nivel Hurt me plenty)

Poder mover el player de algún modo (con botones, ordenes de voz, ReadLine de consola)…

### Punto 3 (nivel Ultra-violence)

Programar los movimientos del player para que capture automáticamente los targets

## Punto 4 (nivel Nightmare¡)

Programar los movimientos del player para que cada vez sea más eficiente

Tú decides hasta dónde quieres llegar, para los puntos 1 y 2 calculamos que máximo lleva un par de horas a un programador experimentado, el 3 es algo que dejamos al candidato como reto personal. Si consigues el 4 en un tiempo razonable el puesto es tuyo. Entregamos un ejemplo de cómo consumir el API en C#

Que valoramos: Código limpio, que funcione, que no tenga fallos, que las excepciones se hayan tratado (la red puede estar caída, por ejemplo) y que se entregue a tiempo.

Suerte¡¡

Ayuda:

