

# Estancias y Objetos de MundiMap: tipos y dinámica.

Documento:	Módulo-1: MundiMap-EstanciasObjetos
Curso:	SISTEMAS DISTRIBUIDOS - Grado en Ingeniería Informática
Libro:	Enunciado Proyecto 2: Mapa del Mundo - Estancias y Objetos de MundiMap
Autor:	LLAMAS BELLO, CESAR
Fecha:	lunes, 25 de abril de 2024
Versión:	1.0 - original - 04-23

## Descripción

El juego MundiMap se describe mediante *estancias*, *objetos* (como cofres y puertas) y *personajes*. En este documento se presenta con mayor detalle en qué consiste una *estancia* y los tipos de objetos que pueden aparecer en el escenario de MundiMap. Aunque el juego se organiza en torno a estos conceptos, es el creador de cada ***mundo*** quien les da nombre y contenido, por eso no se definen nombres de estancias concretas ni objetos concretos mas que como ejemplo. Es recomendable una lectura atenta de este contenido para poder programar el sistema y poder jugar con éxito después.

## Tabla de contenidos

### Estancias y Objetos de MundiMap: tipos y dinámica.

- Descripción
- Tabla de contenidos
- Generalidades sobre el juego
  - Dinámica del juego
  - Dinámicas de las estancias o lugares
    - 1. Llegada a la estancia:
    - 2. Información sobre la estancia:
  - Dinámicas de las puertas:
  - Dinámica de los objetos (cofres):
    - Reapertura de un objeto o de una puerta
  - Parámetros de estancias y objetos:
- Conclusión

## Generalidades sobre el juego

Una de las cualidades más importantes del sistema que soporta los "mapas" que tiene que ser algo configurable, y llamaremos mapa a cualquier configuración concreta de estancias que se le ocurra a un creador, ya sea una *mazmorra* o un *bosque encantado* o lo que sea. No sólo eso, sino que si pudiéramos avanzar más en el diseño de la aplicación, podríamos hacer que el mapa fuera dinámico según se superen ciertos "eventos" del juego. En nuestra *versión preliminar* no lo vamos a hacer así por simplicidad, pero no lo descarto para la siguiente ocasión.

Dentro de una estancia del mapa aparecen únicamente puertas y cofres. Abrir una puerta o un cofre puede repetirse una y otra vez o convertirse en un *evento de juego* si el cofre o la puerta concreta se mantienen abiertos (en el caso del cofre, *gastado*). Por lo general los cofres y puertas se reinician cuando el personaje sale de la estancia donde se encuentran. Estaría muy bien que pudiéramos definir eventos complejos, pero de momento no va a ser así, y un evento es simple y se extiende a toda la partida del jugador.

El termino *estancia* que se usa en este documento, se refiere a lugares como: **habitación** ( `room` ), **corredor** ( `hallway` ), **vestíbulo** ( `hall` ), **átrio** ( `atrium` ), **plaza** ( `square` ), etc. Aunque también podría referirse a lugares abiertos como en una ciudad, o en la naturaleza. Nos referiremos al término *estancia* mejor por "lugar" en castellano, o en inglés con `Place`.

Los objetos que nos vamos a encontrar en un *lugar* son:

- cofres ( `Chest` ) es preciso abrir para saber qué hay dentro; y
- puertas ( `Door` ) que es preciso abrir para dirigirse a otro lugar.

Una vez accionado un objeto se produce su uso, y no hay vuelta atrás. Si se trata de una puerta se *traspasa el umbral*, y una vez abierto un cofre se *usa su contenido*. (Realmente excitante ¿verdad?.)

Veamos en primer lugar qué acciones

## Dinámica del juego

El jugador debe comportarse en cada jugada según las opciones que le muestra la estancia (lugar) donde se encuentra y puede hacer 3 (TRES) cosas:

- **Preguntar:** `Query`, `Ask`, `Show`, son sinónimos de preguntar y dan información sobre un objeto, lugar o puerta. Por ejemplo:

```
1 > Show Place
2 > Ask Place
3 > Query DarkDoor
4 > Ask DirtySmallChest
5 > Query BigVault
```

`Place` es la forma de referirse al lugar donde nos encontramos, como en la línea (1) y (2). En las siguientes pedimos información sobre una puerta o un cofre.

Hay un tipo especial de "pregunta" pensada para obtener información sobre nuestro personaje ( `Me` ), y tiene una única forma como esta:

```
1 > Query Me
2 Hello César, you are a
3
4      _ _ _ _
5      | _ _ | | ( _ ) _ / _ |
6      | | | ' \ | / - _ ) _ |
7      | _ | | _ | _ | _ | _ |
8 and your current stats are HP: 30 and MP: 10
```

- **Actuar:** `Activate`, `Open`, `Disclose`, son sinónimos de abrir, ya sea un cofre o una puerta. Por ejemplo:

```

1 > Disclose DarkDoor
2 > Open RustyGate
3 > Activate LittleBox

```

En cualquier caso, con la información que tenemos, se intenta actuar sobre un objeto o puerta, y si se tiene éxito, se activa su contenido o se traspasa el umbral a una nueva estancia.

- **Terminar:** `Exit`, `Abandon`, `Surrender`, son sinónimos de terminar la partida. Por ejemplo:

```

1 > Surrender
2 Goodbye!

```

También terminas el juego cuando tu `HP` o tu `MP` llegan a `0`, en este caso **mueres** y ¡toca comenzar el juego de nuevo!

No se me ha olvidado que hay que hay una condición que nos permite terminar la partida con éxito, que es cuando abrimos el **cofre del tesoro** (evento `Youwin!`). El **cofre del tesoro** nos da una gran cantidad de energía física y magia y termina el juego con un pequeño mensaje de “elogio”. Puede haber más de un cofre del tesoro pero no es recomendable. También podríamos ponerlo en una sala imposible de abrir, al comienzo del juego, o en lo más profundo del mapa. Depende de tí.

## Dinámicas de las estancias o lugares

### 1. Llegada a la estancia:

Llegamos a una estancia al traspasar una puerta o justo cuando comenzamos el juego. Una vez en ella, y por la razón que sea vengamos de donde vengamos, se nos muestra un mensaje informativo. Este cartel puede incluir nombre de la estancia o cualquier texto que desee el diseñador del mapa. Mensajes válidos son, por ejemplo:

```

1 < Place >
2 "Estás en un lugar sumamente oscuro con varias puertas a derecha e
  izquierda."
3
4 < Gran Domo del Palacio >
5 "Luminoso lugar tan grande como un circo."

```

Como puedes ver, el juego nos muestra una línea donde pone `< Place >` si el lugar es “anónimo”, y otro texto si tiene nombre propio.

### 2. Información sobre la estancia:

Al llegar a una estancia nueva deberíamos investigar su contenido. `Query Place` mostrará todos los objetos y puertas visibles (al menos en ese momento), con su nombre propio y un texto. Por ejemplo:

```

1 > Query Place
2 LittleDoor <nada>
3 BigDoor Gran puerta metálica.
4 LittleBox Una pequeña caja de madera en el extremo de la estancia.
5 DirtyChest Cofre común

```

En la primera columna aparece un nombre “sin espacios” que podemos usar como identificador para hacer una nueva pregunta o interactuar sobre el elemento, tanto una puerta como un cofre.

## Dinámicas de las puertas:

Con el nombre de una puerta (acceso, paso o umbral) podemos:

- **obtener información** sobre ella, como por ejemplo:

```

1  > Ask BigDoor
2
3  | _ |_) _ _ | \ _ _ _ _
4  | _ \ / _ ` | |) / _ \ _ \ ' _ |
5  | _ / _ \ , | _ \ _ \ _ \ _ / _ |
6      | _ /
7  HP: 50, MP: 0, closed
8
9  > Show LittleBox
10 ..little box..
11 HP: 0, MP: 0, Just opened
12
```

En el cualquier caso se muestra un mensaje arbitrario, y a continuación la cantidad de "energía física" (HP) y "energía mágica" (MP) necesaria para accionar la puerta o el objeto. Estos números son fijos para esa puerta independientemente del personaje, y en el caso de un Thief deberán contabilizarse de modo distinto que con un Fighter o a un Wizard. Cuando una puerta necesita ser abierta se indica con la línea `Closed`. Si no, pondrá `Just opened` y no tendremos que gastar energía para traspasarla

- **Actuar** sobre ella, como por ejemplo:

```

1  > Open BigDoor
2  "Sorry, you don't have enough MP or HP!!"
3
4  > Disclose LittleDoor
5  "As you wish!..."
6
```

En el primer ejemplo anterior (1) vemos que la puerta no se abre porque el personaje no dispone de suficiente poder mágico (MP), y la puerta no ha sido abierta antes (Closed) mientras que en el segundo caso (4) se ha traspasado el umbral a una nueva estancia. Se restan los puntos de HP y MP del personaje según indiquen los valores requeridos para accionar esta puerta si la puerta estaba `Closed`, pues si la puerta estaba `Just opened` no se necesita gastar energía.

### Cálculo de los valores de apertura y nuevos valores del personaje:

Al abrir la puerta, o traspasar el umbral, se recalculan los valores del personaje con los puntos de magia o energía que roba la transición, de la misma forma para cualquier tipo de personaje Wizard, Thief o Fighter.

Sin embargo, en el caso de un **Thief** debemos tener en cuenta que los puntos de activación de la puerta son distintos, porque es especialmente hábil, de modo que no sólo necesitará menos puntos para activarla, sino que se le deducirán menos puntos al traspasar el umbral. Pongamos un ejemplo:

### Ejemplo de puerta para un **Thief**:

Cierto ladrón (**Thief**) tiene 20 puntos de **HP** y 10 puntos de **MP**. Al enfrentarse a una puerta, contará como que tiene más puntos:

$$[HP = 20, MP = 10] \rightsquigarrow [HP = 20 \times 1.5 = 30, MP = 10 \times 1.5 = 15]$$

puesto que se corrige su potencial por 1.5. En este caso podría traspasar la puerta.

Tras traspasar el dintel de la puerta la cantidad de puntos que se retraen es menor también. Si ponemos que la puerta tiene un umbral de **HP** de 8 y un umbral de **MP** de 5, se retraen  $\text{round}(8/1.5) = 5$  puntos reales de **HP** y  $\text{round}(5/1.5) = 3$  puntos reales de **MP**.

Visto de otro modo, nuestro "ladrón" podría pasar por una puerta "equivalente" pero con umbrales más bajos de **HP=5** y **MP=3** que se restan de sus *stats* al pasar por ella.

## Dinámica de los objetos (cofres):

Los cofres funcionan de modo parecido a las puertas. Más allá del rótulo que aparece cuando preguntamos por los contenidos de la estancia podemos averiguar cuánto **HP** y **MP** se necesita para actuar con éxito sobre ellos.

- Podemos **obtener información** sobre el objeto, como por ejemplo:

```
1 > Ask LittleBox
2 --little box--
3 HP: 0, MP: 0, Just opened
```

En el cualquier caso se muestra un mensaje arbitrario, y a continuación la cantidad de "energía física" (**HP**) y "energía mágica" (**MP**) necesaria para accionar la puerta o el objeto. En el ejemplo, de arriba se nos dice que no necesitamos ningún punto de **HP** ni **MP** para abrirlo, y que además el cofre ya ha sido abierto (**Just opened**).

- Actuar** sobre el objeto, como aquí:

```
1 > Open LittleBox
2 "As you wish!... HAHAAHAHA!!!! This cursed Food will make you perish."
3 Food: -25
4
5 > Show Me
6 Hello César, you are a
7
8      _ _ _ _ _
9      | _ _ | | _ ( ) _ _ / _ |
10     | | | ' \ | / - ) _ |
11     | _ | | _ | _ \ _ _ | _ |
12 and your current stats are HP: 5 and MP: 10
13
14 > Open DirtyChest
15 "As you wish!"
16 Potion: +20
```

```

16 Food:    +15
17
18 > Ask Me
19 Hello César, you are a
20
21      _ _ _ _
22     | _ _ | | _ ( _ ) _ _ / _ |
23     | | | ' \ | / - _ ) _ |
24     | _ | | _ | | _ \ _ _ | _ |
25 and your current stats are HP: 20 and MP: 30

```

## Cálculo de los nuevos valores del personaje:

Al abrir la puerta, se recalculan los valores del personaje con el contenido. Hay que tener en cuenta que un `Wizard` o un `Fighter` tienen un factor de corrección sobre los puntos recibidos de `Food` y `Potion`, de modo que en el caso anterior, si fuera un mago, habría recibido del `DirtyChest` una poción con un valor de `30`, en lugar de `20`.

## Reapertura de un objeto o de una puerta

Uno de los puntos interesantes que tiene el juego es la posibilidad de dejar abierta una puerta para un uso posterior. Si este es el caso, debemos añadir al juego esta posibilidad, que veremos en el punto posterior. A este tipo de fenómeno lo denominamos **Evento de partida** (o de juego, según el caso), y por eso cada vez que preguntamos por una puerta o un cofre, aparece la coletilla (`Closed` o `Just opened`). Añadiremos información adicional a estos elementos para tener en cuenta que ciertas puertas y ciertos cofres permanecen abiertos cuando ya han sido *accionados*.

## Parámetros de estancias y objetos:

Esta es una propuesta de valores por defecto para una estancia cualquiera, puesto que siempre es conveniente que al crear una estancia, tengo valores aceptables para el sistema.

```

1  DefaultPositiveMessage = "As you wish!";
2  DefaultNegativeMessage = "Sorry, you don't have enough power or magic!";
3
4  DefaultPlaceMessage    = "<...>";
5                          /* say something, please, like "Random dark place." */
6  DefaultPlaceName       = "Here";
7
8  DefaultChestName       = "Chest";
9                          /* si ya existe el nombre se agrega una 'x' y se prueba de
nuevo */
10 DefaultChestHP          = 0;
11 DefaultChestMP          = 0;
12 DefaultChestEvent       = "recloses"; /* vuelve a cerrarse al salir del lugar
*/
13
14 DefaultDoorName         = "Door";
15                          /* si existe el nombre se agrega una 'x' y se prueba de nuevo
*/
16 DefaultDoorHP           = 0;
17 DefaultDoorMP           = 0;
18 DefaultDoorEvent        = "recloses"; /* vuelve a cerrarse al salir del lugar
*/

```

Las estancias se describen mediante un diccionario para cada una de ellas. Contienen además otros dos diccionarios, uno de ellos para las puertas (`Doors`) y otro para los cofres (`Chests`). Cada lugar tiene un identificador enteror a efectos de programación del que justo después del siguiente ejemplo:

```

1 {
2   "Id" : "221",
3   "Name" : "GranDomoDelPalacio",
4   "Message" :
5     "Estas en un lugar sumamente oscuro con varias puertas a derecha e
6     izquierda.",
7   "Doors" : [
8     { "Name": "BigDoor", "HP": 5, "Id": 1432 },
9     { "HP": 10, "MP": 1, "Id": 202, "Event": "reopens"},
10    { "HP": 1, "MP": 10, "Id": 202 }
11  ],
12  "Chests" : [
13    { "NegativeMessage": "Don't try to open me again!!", HP: 50, "Potion":
14      -5},
15    { "Name": "TinyShinyBox", "HP": 2, "MP": 2, "Food": 20, "Potion": 15}
16  ]
17 }
```

En ejemplo en notación al estilo de `Json` pueden verse los siguientes campos:

- `Id`: es un identificador numérico con el índice de la estancia. Es único y nos dirige al "Gran Diccionario de Estancias" (que llamaremos `GDE`). Este diccionario es muy importante porque es donde iremos a buscar cuando traspasemos una puerta.
- `Name`: puede ser un lugar anónimo, o tener un nombre rimbombante, que ponga al jugador en situación. Se mostrará en primer lugar cuando el jugador se encuentre en ese lugar y cuando pregunte dónde está (`Ask Place`).
- `Message`: Una peculiaridad de las mazmorras es su capacidad de suscitar impresiones en el jugador, y el mensaje da mucho ambiente al juego. Es supérfluo a nivel informático pero, *sin pasarse*, hace el juego más divertido.
- `Doors`: Es un pequeño diccionario que contiene puertas, con el nombre de la puerta (si es que lo tiene), sus umbrales de `HP` y `MP` y el índice de la estancia a la que conduce y que se encuentra en el `GDE`. Está indexado (¡jojo!) por el campo `Name`, que es lo que aparece cuando hacemos un (`Show Place` o un `Query Place`). Usaremos ese nombre para viajar por esa puerta.
- `Chests`: Es otro pequeño diccionario que contiene los objetos y, lo mismo que `Doors` está indexado por el campo `Name`. Contiene toda la información con los valores que pedimos que tenga el aventurero para abrir el cofre y también los valores de `Food` y `Potions` que se va a encontrar dentro.
- `Event`: Cuando abrimos una puerta o un cofre y salimos del lugar por una puerta, lo normal es que se vuelvan a cerrar para que tengamos que abrirlos otra vez. Los cofres se recargan y podemos recuperarnos con ellos, pero otros cofres y puertas deberían reabrirse porque si no el juego se vuelve repetitivo. Este caso se indica con `reopens`, que es lo contrario de `recloses` que es el valor por defecto, por lo general. Además existe un evento especial, que es el evento `Youwin!` que marca un cofre como el evento que hay que cumplir para ganar la partida.

## Conclusión

---

Es probable que se te ocurran mil formas de aumentar la diversión del juego. La mayoría de ellas pasan por hacer el sistema multijugador, donde el número de variantes se expande fácilmente:

- Multijugador de mundos separados.
- Multijugador de mundos separados pero con eventos de Mapa. Es decir donde algunos eventos de apertura afectan al resto de jugadores.
- Multijugador de mundo compartido, separado por equipos.
- Multijugador de mundo compartido, separado por equipos, con eventos de Mapa.
- Multijugador de mundo compartido, compartido por equipos.

Además, podemos añadir otros tipos de cofres, desafíos, etcétera. De momento lo más importante para nosotros va a ser conseguir que esto funcione de modo distribuido, donde los lugares y los personajes ocurran en máquinas distintas, y los usuarios interaccionen con el sistema mediante clientes remotos.