Prácticas Lua: Tablas

Práctica 1

En el esqueleto suministrado hay una función definida sin cuerpo. Esta función recibe un solo parámetro y debe pintar un punto en la pantalla, cuya posición está definida en ese parámetro.

El objetivo de la práctica es rellenar esa función con el código necesario para implementar la funcionalidad pedida y utilizar esta función en el segmento de código dentro de la función onDraw para pintar un cuadrado relleno.

Práctica 2

El objetivo de esta práctica es crear la función addCreature (entre los comentarios) para que sin usar condicionales se pinte una criatura distinta en función del nombre de la criatura que pase como parámetro. La llamada a addCreature está escrita ya en el código, así que si ejecutáis el código sin modificar nada os dará un error porque no encuentra la función que debéis definir. Para la implementación de esta función se dispone de la función addImage(nombre_imagen), a la que hay que pasarle el nombre de la imagen.

La textura que corresponde a cada criatura la tenéis aquí:

Grifo	creatures/gryphon.png
Mago	creatures/mage.png
Grunt	creatures/grunt.png
Peon	creatures/peon.png
Dragon	creatures/dragon.png

Práctica 3

Ahora vamos a extender la práctica anterior para definir un mapa de criaturas. De esta forma podemos pintar varias criaturas en pantalla, especificando su nombre y su posición.

Para ello se dispone de la función addImage(nombre_imagen, posicionX, posicionY), a la que hay que pasarle el nombre de la imagen y la posición de la criatura en la pantalla.

Queremos especificar sólo el nombre y no el nombre del fichero, porque así el mapa es genérico, pero podríamos cambiar las texturas independientemente del mapa para un mod, para un DLC, para otro nivel distinto, etc.

En el mapa sólo está el nombre de la criatura, con lo que la asociación entre nombre y fichero con la textura tendrá que estar en otro sitio, por suerte ya habéis hecho la práctica anterior y sabéis cómo hacerlo.

Práctica 4

Extiéndelo ahora para gestionar el tamaño de cada tipo de criatura, ya que un dragón debería ser más grande que un peón.

Para ello se dispone de la función addImage(nombre_imagen, posicionX, posicionY, tamañoX, tamañoY), a la que hay que pasarle el nombre de la imagen, la posición y el tamaño de la criatura.

Crea un mapa con al menos 3 dragones, 2 magos, 2 grunts, 1 grifo y 1 peon.

En este caso no hay llamada para pintar el mapa, así que tendréis que añadirla vosotros también.

Práctica 5

Vamos a terminar de mejorar lo que estamos haciendo. Para hacer el programa más realista, los datos de las criaturas y el mapa no pueden estar metidos dentro del código. Es mejor tenerlo fuera del código para tratarlo de forma independiente, por ejemplo en ficheros .xml.

El objetivo de esta práctica es definir los ficheros xml que sean necesarios, con su formato. Todos los datos deben estar en ficheros .xml, ningún dato debe estar definido en código lua.

Después utilizar estos ficheros para pintar criaturas en la pantalla de igual forma que hemos hecho en las prácticas anteriores. El resultado final debe ser el mismo que en la práctica 4, pero obteniendo todos los datos de xml.

Para cargar un xml de disco tenéis disponible la función readXML que lee un fichero xml cuyo nombre se le pasa como parámetro y devuelve una tabla con los valores almacenados en el fichero xml. El formato de la tabla es el siguiente:

Índice	Valor
tag	Nombre del tag
attr	Tabla de atributos
1	Primer sub elemento
2	Segundo sub elemento
...	
n	Último sub elemento

La tabla de atributos tiene este formato:

Índice	Valor
1	Nombre del atributo 1
2	Nombre del atributo 2
...	
n	Nombre del último atributo
Nombre del atributo 1	Valor del atributo 1
Nombre del atributo 2	Valor del atributo 2
...	
Nombre del último atributo	Valor del último atributo

Cada sub elemento tiene la misma estructura de nuevo.

Por ejemplo este fichero .xml:

```
<elemento>
    <sub_elemento1 attr1="valor1">
        <sub_sub_elemento1> valor2 </sub_sub_elemento1>
        <sub_sub_elemento2> valor3 </sub_sub_elemento2>
    </sub_elemento1>
</elemento>
```

Tendría esta tabla:

<table>
<tr><td>tag</td><td colspan="3">elemento</td></tr>
<tr><td>attr</td><td colspan="3">{}</td></tr>
<tr><td rowspan="7">1</td><td>tag</td><td colspan="2">sub_elemento1</td></tr>
<tr><td rowspan="2">attr</td><td>1</td><td>attr1</td></tr>
<tr><td>attr1</td><td>valor1</td></tr>
<tr><td rowspan="3">1</td><td>tag</td><td>sub_sub_elemento1</td></tr>
<tr><td>attr</td><td>{}</td></tr>
<tr><td>1</td><td>valor2</td></tr>
<tr><td rowspan="3">2</td><td>tag</td><td>sub_sub_elemento2</td></tr>
<tr><td>attr</td><td>{}</td></tr>
<tr><td>1</td><td>valor3</td></tr>
</table>

Debes realizar la práctica en dos versiones:

1. Usando atributos en los ficheros xml
2. Sin usar atributos en los ficheros xml