

GuillemSalas

M06UF1-Map en C++

-Explicad cómo funciona "map" en C++

Map en c++ es un contenedor que almacena pares clave-valor donde cada clave es única y está asociada al valor.

-Cómo lo utilizaríais en nuestro proyecto para almacenar las texturas de tal manera que sea fácil relacionarlas con las letras que tienen asignadas.

```
#include <map>
```

```
map<char, string> texturesMap;
```

Primero ponemos el include map para poderlo utilizar después podemos especificar de qué tipo quieres que sea la clave y el valor en nuestro caso será un char para clave y string para el valor

```
ZEN;0.5;;  
TITLE;Cucuruchos Mentirosos;;  
TEXTURES;4;  
G;grass.png;;  
R;rock.png;;  
B;brick.png;;
```

Ahora para leer el archivo nos quedamos en el 4 lo primero que haremos será hacer un salto de línea para saltar los dos ;

```
getline(file, tmp, '\n');  
  
for (int i = 0; i < textureNumber; i++) {  
    getline(file, textureLetter, delemiter);  
    getline(file, textureFile, delemiter);  
  
    texturesMap[textureLetter[0]] = textureFile;  
  
    cout << "Letter: " << textureLetter[0] << " File: " << textureFile << endl;  
  
    getline(file, data, '\n');  
}
```

Y después hacemos un for que se ejecute las veces que texturas tengamos en nuestro texto en nuestro caso 4.

```
for (int i = 0; i < textureNumber; i++) {  
    getline(file, textureLetter, delemiter);  
    getline(file, textureFile, delemiter);  
  
    texturesMap[textureLetter[0]] = textureFile;  
  
    cout << "Letter: " << textureLetter[0] << " File: " << textureFile << endl;  
  
    getline(file, data, '\n');  
}
```

Lo que hacemos dentro del for es coger la letra y la guardamos en texture letter y después cogemos la textura y la guardamos en texture File.
Por último guardamos en el map textureMap la letra con su textura y la imprimimos por pantalla finalmente hacemos un salto de línea para la siguiente textura.