



## Programación 2D

### Práctica 7: Carga de mapas

#### Clase World

En esta práctica de la asignatura, vamos a añadir a la clase World soporte para la carga de mapas de Tiled. Para ello, vamos a añadir tres métodos nuevos a la clase (con las variables miembro que sean necesarias):

- `bool loadMap(const char* filename, uint16_t firstCollid);`

El método `loadMap` debe cargar, utilizando la librería PugiXML, los datos del archivo TMX pasado como parámetro. Este archivo necesita haber sido guardado en Tiled con unas propiedades particulares: en el panel que se muestra en la opción de menú Map / Map Properties..., damos a la propiedad Tile Layer Format el valor XML. Esto guardará el mapa con el formato que hemos visto en las diapositivas del tema.

Cuando leamos el nombre del archivo de imagen del tileset, debemos tener en cuenta la ruta al archivo del mapa. Por ejemplo, si nos dice que el nombre del tileset es "tileset.png", y estamos cargando el mapa "data/map.tmx", entonces debemos de cargar el tileset de la carpeta "data". Podemos hacer una función para extraer la ruta del nombre de un archivo utilizando la librería estándar:

```
inline std::string extractPath(const std::string& filename)
{
    std::string filenameCopy = filename;
    while ( filenameCopy.find("\\") != std::string::npos )
    {
        filenameCopy.replace(filenameCopy.find("\\"), 1, "/");
    }
    filenameCopy = filenameCopy.substr(0, filenameCopy.rfind('/'));
    if ( filenameCopy.size() > 0 ) filenameCopy += "/";
    return filenameCopy;
}
```

Su uso es muy simple, si le pasamos el string "**data/map.tmx**", nos devolverá "**data/**". Debemos poner este valor como prefijo del archivo del tileset antes de cargarlo.

- Vec2 **getMapSize()** const;

El método getMapSize debe devolver un vector con la anchura y altura en píxeles del mapa.

- bool **moveSprite**(Sprite& sprite, const Vec2& amount);

El método moveSprite lo utilizaremos para mover un sprite por el mundo permitiendo que colisione con el mapa (los tiles utilizarán detección de colisiones por rectángulos). El segundo parámetro indica cuántos puntos mover el sprite sobre cada eje. El método hace lo siguiente:

- Movemos el sprite sobre el eje X.
- Si colisiona con algún tile, volvemos a la posición X anterior.
- Movemos el sprite sobre el eje Y.
- Si colisiona con algún tile, volvemos a la posición Y anterior.

El método debe devolver true si se ha producido colisión al mover el sprite, o false en caso contrario.

Modificaremos también el método draw para, justo antes de dibujar los sprites, dibujar todos los tiles del mapa.

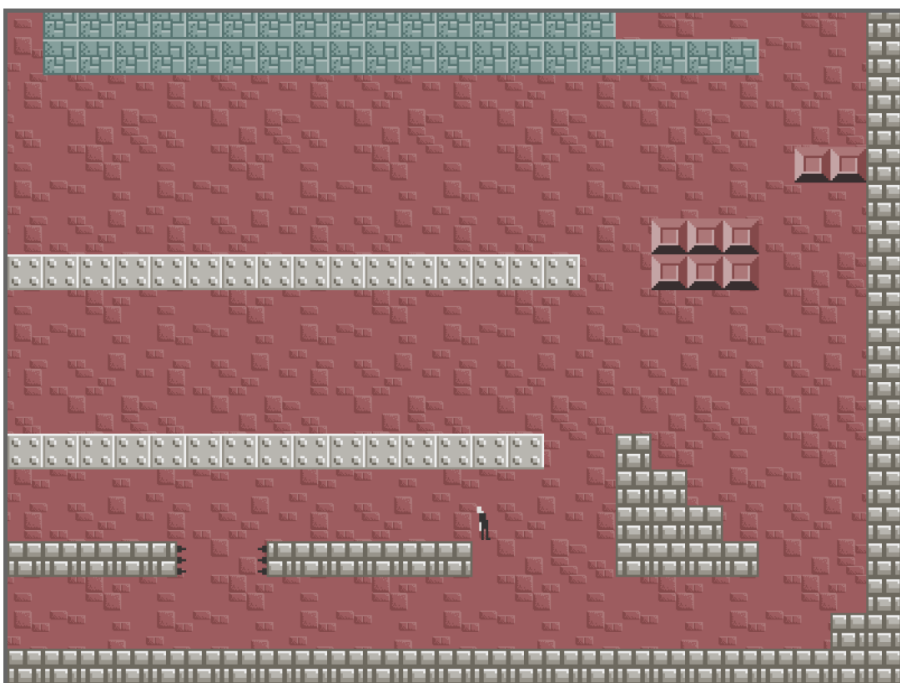
## Programa principal

Crearemos un mundo con el plano de nubes de la práctica anterior de fondo, y cargaremos el mapa "**data/map.tmx**".

Añadimos al mundo un sprite. Podemos mover por el nivel a este sprite utilizando las teclas de cursor izquierdo y derecho, y podemos hacer que salte pulsando espacio.

Cuando el sprite se está moviendo, utiliza la imagen "**data/run.png**" (con 6 frames de animación en horizontal y 1 en vertical). En caso contrario, utiliza la imagen "**data/idle.png**" (con 1 frame de animación).

El sprite debe ser capaz de colisionar con el mapa, tal y como se ve en la práctica resuelta.



[illegible]