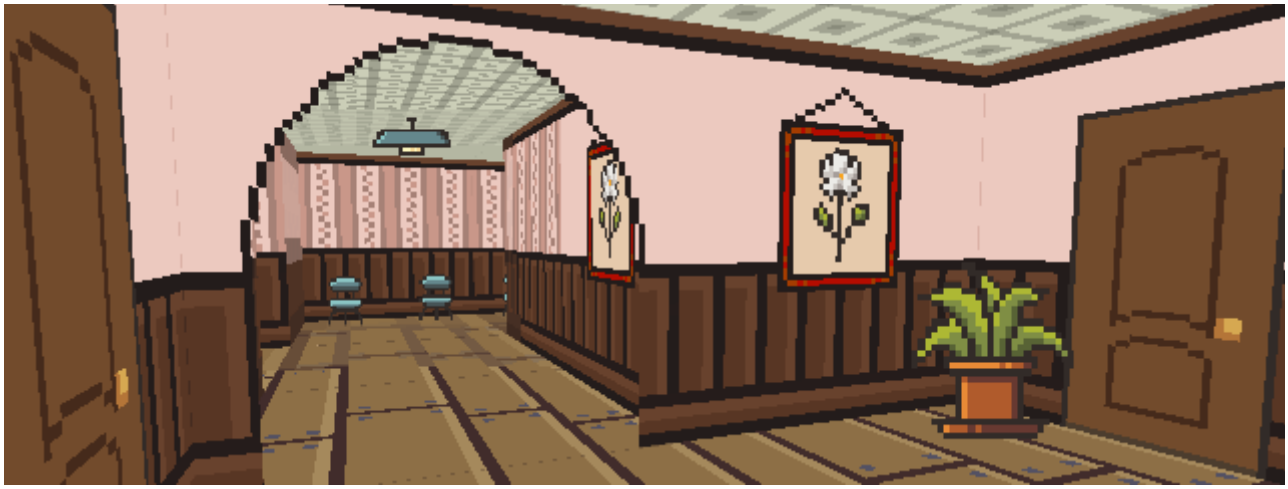
- irrTiled -

These functions are used to load **TMX** files created with the Tiled map editor (www.mapeditor.org). With few lines of code you can show complex maps or if you prefer use all the variables to create complex things. Also, by using part of the *Irrlicht 3D Engine*, it can also be used to create 3D environments using boxes and loading meshes.



Examples:

- **2DBasic** - It contains a very basic example of how a *TMX* file is loaded and how the data is used to draw a map with all its *layers* and *tilesets*.
- **3DBasic** - It contains an example of how to use the 3d functions of Irrlicht Engine to generate a 3D environment with the map data.
- **Animation** - It is a simple example of how you can create animations with tiles, as well as how the tileset is used externally (*TSX* files). It can be done in many ways.
- **Levels** - It shows a way to create different heights in 3D environments. I use positive and negative signs to know if one layer is above another or below.
- **Objects** - It shows how the data of *objectgroup* and *object* is used to use them in anything. They are very useful since they can be added to the player, enemies, create collisions or damage zones.
- **Properties** - This example shows how to extract the properties of the objects. Properties can be present in "layers", "tileset" and many others. They contain information added by the user.
- **Shapes** - The *objects* can have many shapes to be used on the map. The most common is the rectangle but you can use others such as ellipses, points or even create polygons. In this example I visually show what shapes it has and how they could be used.
- **Tilesets** - This example shows how several different *tileset* can be used on the same map to add more images, effects or animations.

Too easy!

[FUNCTIONS]

irrTiled(const c8* filename, irrlicht device*)
Create the environment to be able to load and use tmx files.

- *filename:* File name ".tmx" to load. For now it only accepts Tile layer format in CSV or XML, if it is in compressed or uncompressed Base64 it cannot be used (for now).

- *device:* Here goes the Irrlicht device.

HEX2RGB(stringc value)
Converts hexadecimal to rgb color. The alpha color, if any, is discarded and it will always return 255.

- *value:*Value to enter. It can be a string of the type **#aarrggbb** or **#rrggbb**.

Limitations:
- Only can use the Tile layer format in CSV, XML and Base64 (no compressed).
- There are many variables that cannot be used yet.

[Map]

stringc Version
The TMX format version. Was “1.0” so far, and will be incremented to match minor Tiled releases.

stringc TiledVersion
The Tiled version used to save the file.

stringc Orientation
Map orientation. Tiled supports “*orthogonal*”, “*isometric*”, “*staggered*” and “*hexagonal*”

stringc RenderOrder
The order in which tiles on tile layers are rendered.

dimension2du Size
The map width/height in tiles.

dimension2du TileSize
The width/height of a tile.

s32 Infinite
Whether this map is infinite. An infinite map has no fixed size and can grow in all directions.

s32 NextLayerID
Stores the next available ID for new layers.

s32 NextObjectID
Stores the next available ID for new objects.

SColor BackgroundColor
The background color of the map.

array<TILESET> Tileset
It contains an array with all the tilesets that the map contains.

array<LAYER> Layer
It contains an array with all the layers that the map contains.

array<OBJECTGROUP> ObjectGroup
It contains an array with all the objects groups that the map contains.

array<PROPERTY> Properties
Contains an array with all the properties. To use correctly you must know the name and type of the variable to use.

[Tileset]

s32 FirstGID
The first global tile ID of this tileset (this global ID maps to the first tile in this tileset).

stringc Name
The name of this tileset.

dimension2du Size
The (maximum) width/height of the tiles in this tileset.

s32 TileCount
The number of tiles in this tileset

s32 Columns
The number of tile columns in the tileset. For image collection tilesets it is editable and is used when displaying the tileset.

stringc Source
The reference to the tileset image file.

IMAGE Image
This is the already loaded texture that will be used to display the tiles.

array<recti> SubRects
It contains an array with all rectangles with a specific position and size to select a portion of the image to use.

array<Tile> Tile
It contains an array with all Tiles with a specific id, frame and object group.

[Tile]

`s32` **ID**
The local tile ID within its tileset.

`array<FRAME>` **Animation**
It contains an array with all the frames of the animation that the tile has.

`array<OBJECTGROUP>` **ObjectGroup**
It contains an array with all the groups of objects that the tile contains.

[Frame]

`s32` **`TileID`**
The local ID of a tile within the parent tileset.

`s32` **`Duration`**
How long (in milliseconds) this frame should be displayed before advancing to the next frame.

[Image]

`s32` **`ID`**
Unique ID of the layer. Each layer that added to a map gets a unique id.

`stringc` **`Source`**
Unique ID of the layer. Each layer that added to a map gets a unique id.

`ITexture` **`Texture`**
Unique ID of the layer. Each layer that added to a map gets a unique id.

[Layer]

`s32` **`ID`**
Unique ID of the layer. Each layer that added to a map gets a unique id.

`stringc` **`Name`**
The name of the layer.

`dimension2du` **`Size`**
The width/height of the layer in tiles.

`bool` **`Visible`**
Whether the layer is shown (*true*) or hidden (*false*).

`SColor` **`Tintcolor`**
A color that is multiplied with any tile objects drawn by this layer.

`array<s32>` **`Data`**
It contains a matrix with all the tile IDs and their position on the map.

`array<PROPERTY>` **`Properties`**
Contains an array with all the properties. To use correctly you must know the name and type of the variable to use.

[Object Group]

`s32` ID
Unique ID of the layer. Each layer that added to a map gets a unique id.

`stringc` Name
The name of the object group.

`bool` Visible
Whether the object group is shown (*true*) or hidden (*false*).

`SColor` Tintcolor
A color that is multiplied with any tile objects drawn by this objectgroup.

`array<OBJECT>` Object
Contains an array with all the objects in the group.

`array<PROPERTY>` Properties
Contains an array with all the properties. To use correctly you must know the name and type of the variable to use.

[Objects]

`s32` **ID**
Unique ID of the object. Each object that is placed on a map gets a unique id.

`position2di` **Pos**
The x/y coordinate of the object in pixels.

`s32` **Rotation**
The rotation of the object in degrees clockwise around (x, y).

`dimension2du` **Size**
The width/height of the object in pixels.

`SHAPE` **Shape**
The object can have several shapes and each one has different variables:

ellipse: Used to mark an object as an ellipse. Used to mark an object as an ellipse. The existing *x*, *y*, *width* and *height* attributes are used to determine the size of the ellipse.

point: Used to mark an object as a point. The existing *x* and *y* attributes are used to determine the position of the point.

polygon
- points: A list of *x*,*y* coordinates in pixels.

Each polygon object is made up of a space-delimited list of *x*,*y* coordinates. The origin for these coordinates is the location of the parent object. By default, the first point is created as 0,0 denoting that the point will originate exactly where the object is placed.

`array<PROPERTY>` **Properties**
Contains an array with all the properties. To use correctly you must know the name and type of the variable to use.

[PROPERTIES]

stringc Name
The name of the property.

stringc Type
The type of the property. Can be string, int, float, bool, color, file or object.

bool BoolValue
Boolean properties have a value of either “true” or “false”.

s32 IntValue
Integer value of the property.

f32 FloatValue
Float value of the property.

stringc FileValue
File properties are stored as paths relative from the location of the map file.

stringc StringValue
String value of the property.

s32 ObjectValue
Object properties can reference any object on the same map and are stored as an integer (the ID of the referenced object, or 0 when no object is referenced).

SColor ColorValue
Color properties are stored in ARGB.

LICENSED

Copyright (c) 2020-2022 TheMrCerebro

irrTiled - Zlib license.

This software is provided 'as-is', without any express or
implied warranty. In no event will the authors be held
liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute
it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

If you think this tool is useful, you can help me in various ways: help me improve it, advertising where you think is convenient or making a donation.

-To contact me or DONATION$-

or

themrcerebro@gmail.com

THANKS!!!