Game Dev: TMX loading

Ricard Pillosu - UPC

Data driven level loading

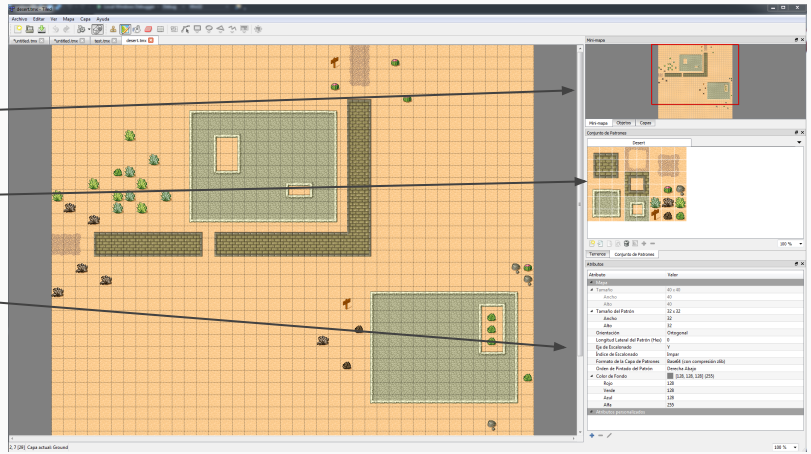
- We will no longer create a .c file for each “scene”
- Instead, we will load **XML** files that describe:
  - Graphics
  - Collisions
  - Other *metadata* like item placement or player start point
- We will use a new Module for loading maps
  - It will load the file and store all important data
  - We can query all map *metadata* to that module
  - We can ask the map module to render itself to screen

Map editor of choice: Tiled

- http://www.mapeditor.org/
- Used in games like Star Command or Shovel Knight
- Used in other engines like Unity3d, Game Maker or Unreal Engine
- Maps are saved in TMX format
- Have http://doc.mapeditor.org/reference/tmx-map-format/ opened

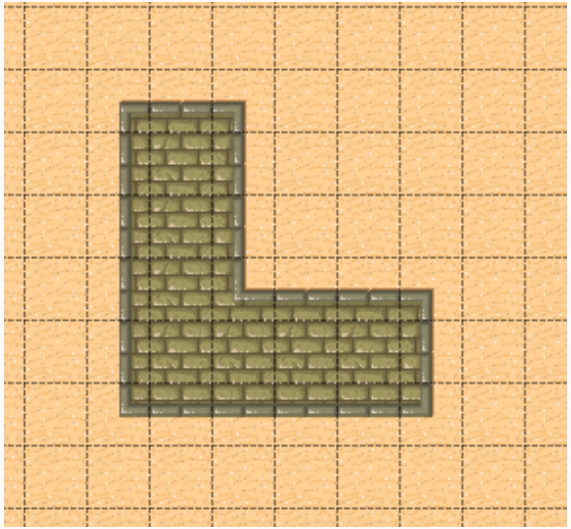
Using Tiled

- Open Tiled
- Open all examples, then stick with **examples/desert.tmx**
- Locate:
  - Minimap / Layers area
  - Patterns and terrains area
  - Attributes area



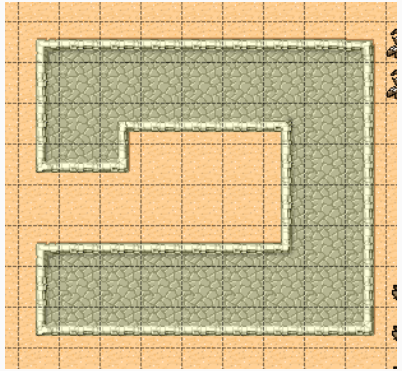
Using Tiled

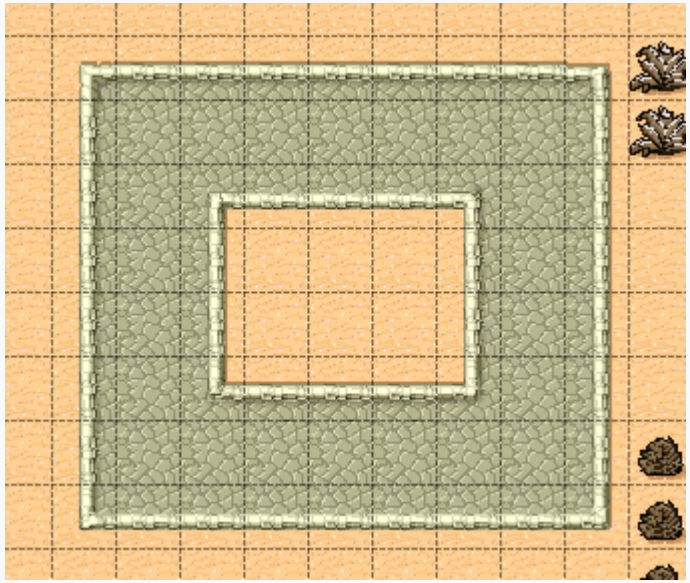
- Duplicate some cactus using **Stamp Tool**
- Right-click to create an area, then paint with it
- Create a closed house of bricks



Using Tiled

- Select Terrains from the middle right panel
- Pick Cobblestone and draw a donut
- Now pick Desert terrain and make a hole



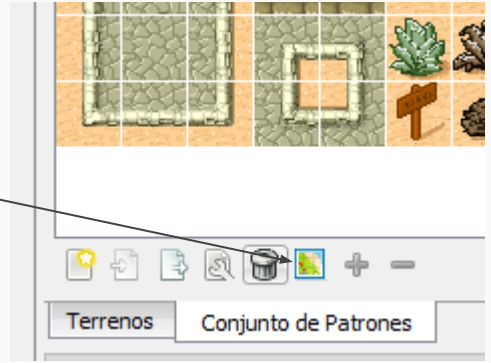


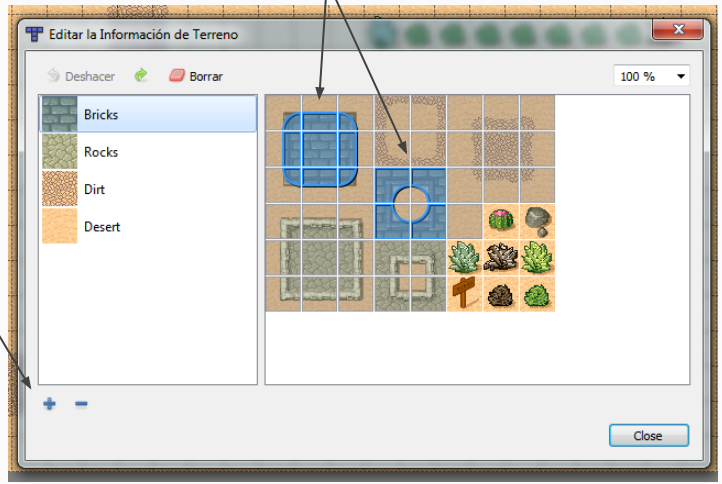
Using Tiled

- Create a new map (File > New …)
- Pick Orthogonal orientation
- Make it 50 tiles wide by 15 high
- Tiles should be 32x32
- Add a new pattern, pick tw_desert_spacing.png
- It is 32x32 with 1x1 margin!

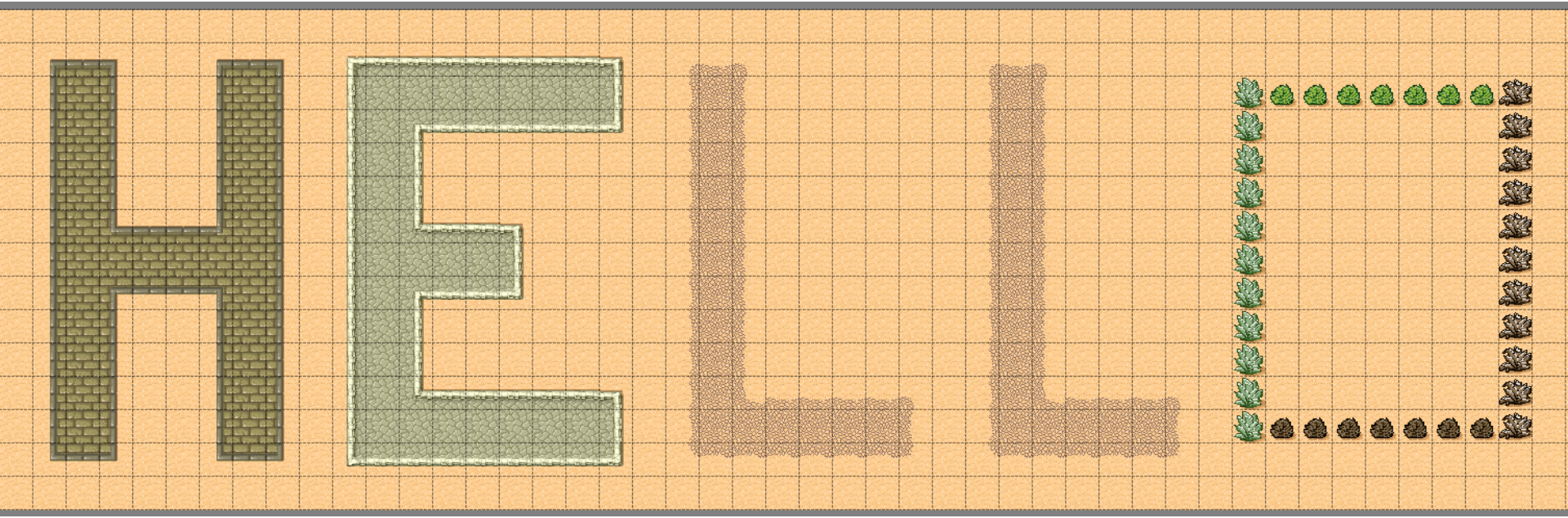
Using Tiled

- Click on Terrain Editor
- Add a new terrain and its tiles





Create this map



The TMX (Tile Set XML) Format

- Save the previous map as *hello.tmx* on *Game/maps/*
- You also need to copy the pattern file: *tmw_desert_spacing.png*
- Open hello.tmx
- Have opened http://doc.mapeditor.org/reference/tmx-map-format/

The TMX Format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map SYSTEM "http://mapeditor.org/dtd/1.0/map.dtd">
<map version="1.0" orientation="orthogonal" renderorder="right-down" width="50" height="15"
tilewidth="32" tileheight="32" nextobjectid="1">
 <tileset firstgid="1" name="Desert" tilewidth="32" tileheight="32" spacing="1" margin="1">
  <image source="../../../../../../Program Files (x86)/Tiled/examples/tmw_desert_spacing.png"
width="265" height="199"/>
  <terraintypes>
   <terrain name="Bricks" tile="9"/>
   <terrain...
  </terraintypes>
  <tile id="0" terrain="3,3,3,0"/>
  <tile...
 </tileset>
 <layer name="Capa de Patrones 1" width="50" height="15">
  <data>
   <tile gid="30"/>
   <tile...
  </data>
 </layer>
</map>
```

In the end all the information is stored in an XML we can parse and store for later rendering.

The TMX Format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map SYSTEM "http://mapeditor.org/dtd/1.0/map.dtd">
<map version="1.0" orientation="orthogonal" renderorder="right-down" width="50" height="15"
tilewidth="32" tileheight="32" nextobjectid="1">
 <tileset firstgid="1" name="Desert" tilewidth="32" tileheight="32" spacing="1" margin="1">
  <image source="../../../../../../Program Files (x86)/Tiled/examples/tmw_desert_spacing.png"
width="265" height="199"/>
  <terraintypes>
   <terrain name="Bricks" tile="9"/>
   <terrain...
  </terraintypes>
  <tile id="0" terrain="3,3,3,0"/>
  <tile...
 </tileset>
 <layer name="Capa de Patrones 1" width="50" height="15">
  <data>
   <tile gid="30"/>
   <tile...
  </data>
 </layer>
</map>
```

Standard XML header with version and character encoding.

The TMX Format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map SYSTEM "http://mapeditor.org/dtd/1.0/map.dtd">
<map version="1.0" orientation="orthogonal" renderorder="right-down" width="50" height="15"
tilewidth="32" tileheight="32" nextobjectid="1">
 <tileset firstgid="1" name="Desert" tilewidth="32" tileheight="32" spacing="1" margin="1">
  <image source="../../../../../../Program Files (x86)/Tiled/examples/tmw_desert_spacing.png"
width="265" height="199"/>
  <terraintypes>
   <terrain name="Bricks" tile="9"/>
   <terrain...
  </terraintypes>
  <tile id="0" terrain="3,3,3,0"/>
  <tile...
 </tileset>
 <layer name="Capa de Patrones 1" width="50" height="15">
  <data>
   <tile gid="30"/>
   <tile...
  </data>
 </layer>
</map>
```

Where they store the DTD file for validating this XML against the TMX format.

The TMX Format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map SYSTEM "http://mapeditor.org/dtd/1.0/map.dtd">
<map version="1.0" orientation="orthogonal" renderorder="right-down" width="50" height="15"
tilewidth="32" tileheight="32" nextobjectid="1">
 <tileset firstgid="1" name="Desert" tilewidth="32" tileheight="32" spacing="1" margin="1">
  <image source="../../../../../../Program Files (x86)/Tiled/examples/tmw_desert_spacing.png"
width="265" height="199"/>
  <terraintypes>
   <terrain name="Bricks" tile="9"/>
   <terrain...
  </terraintypes>
  <tile id="0" terrain="3,3,3,0"/>
  <tile...
 </tileset>
 <layer name="Capa de Patrones 1" width="50" height="15">
  <data>
   <tile gid="30"/>
   <tile...
  </data>
 </layer>
</map>
```

The root node map stores the general configuration of the map as attributes *(similar to the dialog when creating a new map)*.

The TMX Format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map SYSTEM "http://mapeditor.org/dtd/1.0/map.dtd">
<map version="1.0" orientation="orthogonal" renderorder="right-down" width="50" height="15"
tilewidth="32" tileheight="32" nextobjectid="1">
 <tileset firstgid="1" name="Desert" tilewidth="32" tileheight="32" spacing="1" margin="1">
  <image source="../../../../../../Program Files (x86)/Tiled/examples/tmw_desert_spacing.png"
width="265" height="199"/>
  <terraintypes>
   <terrain name="Bricks" tile="9"/>
   <terrain...
  </terraintypes>
  <tile id="0" terrain="3,3,3,0"/>
  <tile...
 </tileset>
 <layer name="Capa de Patrones 1" width="50" height="15">
  <data>
   <tile gid="30"/>
   <tile...
  </data>
 </layer>
</map>
```

We will have one tileset node per each pattern *(very similar to the dialog when creating a new pattern)*.

The TMX Format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map SYSTEM "http://mapeditor.org/dtd/1.0/map.dtd">
<map version="1.0" orientation="orthogonal" renderorder="right-down" width="50" height="15"
tilewidth="32" tileheight="32" nextobjectid="1">
 <tileset firstgid="1" name="Desert" tilewidth="32" tileheight="32" spacing="1" margin="1">
  <image source="../../../../../../Program Files (x86)/Tiled/examples/tmw_desert_spacing.png"
width="265" height="199"/>
  <terraintypes>
   <terrain name="Bricks" tile="9"/>
   <terrain...
  </terraintypes>
  <tile id="0" terrain="3,3,3,0"/>
  <tile...
 </tileset>
 <layer name="Capa de Patrones 1" width="50" height="15">
  <data>
   <tile gid="30"/>
   <tile...
  </data>
 </layer>
</map>
```

The texture source file. Remove the path since we will have the file in the same folder as the map.

The TMX Format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map SYSTEM "http://mapeditor.org/dtd/1.0/map.dtd">
<map version="1.0" orientation="orthogonal" renderorder="right-down" width="50" height="15"
tilewidth="32" tileheight="32" nextobjectid="1">
 <tileset firstgid="1" name="Desert" tilewidth="32" tileheight="32" spacing="1" margin="1">
  <image source="tmw_desert_spacing.png" width="265" height="199"/>
  <terraintypes>
   <terrain name="Bricks" tile="9"/>
   <terrain...
  </terraintypes>
  <tile id="0" terrain="3,3,3,0"/>
  <tile...
 </tileset>
 <layer name="Capa de Patrones 1" width="50" height="15">
  <data>
   <tile gid="30"/>
   <tile...
  </data>
 </layer>
</map>
```

Mind that is already tells you the image size.

The TMX Format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map SYSTEM "http://mapeditor.org/dtd/1.0/map.dtd">
<map version="1.0" orientation="orthogonal" renderorder="right-down" width="50" height="15"
tilewidth="32" tileheight="32" nextobjectid="1">
 <tileset firstgid="1" name="Desert" tilewidth="32" tileheight="32" spacing="1" margin="1">
  <image source="tmw_desert_spacing.png" width="265" height="199"/>
  <terraintypes>
   <terrain name="Bricks" tile="9"/>
   <terrain...
  </terraintypes>
  <tile id="0" terrain="3,3,3,0"/>
  <tile...
 </tileset>
 <layer name="Capa de Patrones 1" width="50" height="15">
  <data>
   <tile gid="30"/>
   <tile...
  </data>
 </layer>
</map>
```

Terrain Types duplicates the terrains we have created.

The TMX Format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map SYSTEM "http://mapeditor.org/dtd/1.0/map.dtd">
<map version="1.0" orientation="orthogonal" renderorder="right-down" width="50" height="15"
tilewidth="32" tileheight="32" nextobjectid="1">
 <tileset firstgid="1" name="Desert" tilewidth="32" tileheight="32" spacing="1" margin="1">
  <image source="tmw_desert_spacing.png" width="265" height="199"/>
  <terraintypes>
   <terrain name="Bricks" tile="9"/>
   <terrain...
  </terraintypes>
  <tile id="0" terrain="3,3,3,0"/>
  <tile...
 </tileset>
 <layer name="Capa de Patrones 1" width="50" height="15">
  <data>
   <tile gid="30"/>
   <tile...
  </data>
 </layer>
</map>
```

If a tile is part a terrain, it will be here (normally a long list). The *terrain* attribute holds the id of the terrain at each corner.

The TMX Format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map SYSTEM "http://mapeditor.org/dtd/1.0/map.dtd">
<map version="1.0" orientation="orthogonal" renderorder="right-down" width="50" height="15"
tilewidth="32" tileheight="32" nextobjectid="1">
 <tileset firstgid="1" name="Desert" tilewidth="32" tileheight="32" spacing="1" margin="1">
  <image source="tmw_desert_spacing.png" width="265" height="199"/>
  <terraintypes>
   <terrain name="Bricks" tile="9"/>
   <terrain...
  </terraintypes>
  <tile id="0" terrain="3,3,3,0"/>
  <tile...
 </tileset>
 <layer name="Capa de Patrones 1" width="50" height="15">
  <data>
   <tile gid="30"/>
   <tile...
  </data>
 </layer>
</map>
```

After all tilesets, we have the lists of layers (one for now), with basic name and size.

The TMX Format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map SYSTEM "http://mapeditor.org/dtd/1.0/map.dtd">
<map version="1.0" orientation="orthogonal" renderorder="right-down" width="50" height="15"
tilewidth="32" tileheight="32" nextobjectid="1">
 <tileset firstgid="1" name="Desert" tilewidth="32" tileheight="32" spacing="1" margin="1">
  <image source="tmw_desert_spacing.png" width="265" height="199"/>
  <terraintypes>
   <terrain name="Bricks" tile="9"/>
   <terrain...
  </terraintypes>
  <tile id="0" terrain="3,3,3,0"/>
  <tile...
 </tileset>
 <layer name="Capa de Patrones 1" width="50" height="15">
  <data>
   <tile gid="30"/>
   <tile...
  </data>
 </layer>
</map>
```

Each layer holds all their data inside the data tag. Each *tile* element holds the *id* of a tile in the pattern (starts with 1).

TODO 1

“Create a struct needed to hold the information to Map node”

- Based on the XML create a struct with the right type of vars

```
<map version="1.0"
     orientation="orthogonal"
     renderorder="right-down"
     width="50"
     height="15"
     tilewidth="32"
     tileheight="32"
     nextobjectid="1">
```

TODO 2

“Create a struct to hold information for a TileSet. Ignore Terrain Types and Tile Types for now, but we want the image!”

- Same as with map properties (TODO 1)

```
<tileset
      firstgid="1"
      name="Desert"
      tilewidth="32"
      tileheight="32"
      spacing="1"
      margin="1">
```

TODO 3

“Create and call a private function to load and fill all your map data”

- As with config files or save files, just read the attributes of map and fill your struct
- We want everything! :)

TODO 4

“Create and call a private function to load a tileset. Remember to support more any number of tilesets!”

- As with map data, first make a function to load a single tileset
- Then make sure you loop all possible tilesets and repeat the process
- We want everything! :)

```
for(pugi::xml_node tileset = map_file.child("map").child("tileset"); tileset; tileset = tileset.next_sibling("tileset"))
```

TODO 5

“LOG all the data loaded iterate all tilesets and LOG everything”

- We need to be extremely careful when loading external data
- LOG everything we read so far, including all tilesets

```
Successfully parsed map XML file: hello2.tmx
width: 50 height: 15
tile_width: 32 tile_height: 32
Tileset ----
name: Desert firstgid: 1
tile width: 32 tile height: 32
spacing: 1 margin: 1
```

TODO 6

“Iterate all tilesets and draw all their images in 0,0 (you should have only one tileset for now)”

- To achieve the results of solution.exe, draw the tileset image
- Iterate all, but we should have only one
- Blit to 0,0

TODO 7

*“Set the window title to **Map:%dx%d Tiles:%dx%d Tilesets:%d**”*

- To achieve the results of solution.exe
- Last is the amount of tilesets loaded

Homework

- Load all layers
- As with tilesets you need to support any number of them
- First create the structs based on the data from the XML
- Then the private functions to fill the structs
- Then LOG everything