Tutorial – Loading a Level with JSON

The lecture slides for this week cover creating a level using Tiled Studio and exporting it in JSON format. This tutorial assumes you have already created and exported your level, and will only be concerned with adding new code to your code from last week's tutorial.

Drawing the Level:

1. Make sure you have modified your level1.json file (as per the description in the slides) so that the text 'level1 = ' is added to the very start of the file.

   The first few lines of level1.js should look similar to this:

```
var level1 = { "height":15,
 "layers":[
        {
         "data":[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ...
         "height":15,
         "name":"Background Images",
         "opacity":1,
         "type":"tilelayer",
```

2. Include a reference to the level1.js file in the HTML file

```
<html>
  <body>
    <canvas id="gameCanvas" width="640" height="480">
          This text is displayed if your browser does not support HTML5.
    </canvas>
    <script src="level1.js"></script>
    <script src="keyboard.js"></script>
    <script src="player.js"></script>
    <script src="main.js"></script>
  </body>
</html>
```

3. Open main.js and copy the drawMap() function from the slides into your main.js file (just before the run() function is a good place to put it)

4. To get the drawMap function to work, we need to add some variables that specify how the tile set and level have been set up.

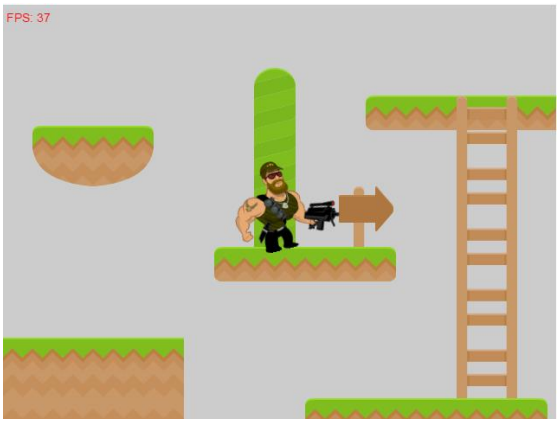
   Add the following variables to your game. If you are using your own tileset, or have made a different level to the one used in the slides, modify the values accordingly:

- var LAYER_COUNT = 3;
  The number of layers in your map. In the sample from this week's lesson we're using a background layer, a layer for the platforms, and a layer for the ladders. (We'll add more layers in a later lesson)
- var MAP = { tw: 60, th: 15 };
  Specifies how big your level is, in tiles. The sample level from the lesson is 60 tiles wide by 15 tiles high.
- var TILE = 35;
  The width/height of a tile (in pixels). Your tiles should be square. These dimensions refer to the map grid tiles. Our tileset tiles (the images) can be different dimensions.
- var TILESET_TILE = TILE * 2;
  The width/height of a tile in the tileset. Because the images are twice as big as the grid in our map we need to be careful (but it allows us a bit more flexibility when designing the level)
- var TILESET_PADDING = 2;
  How many pixels are between the image border and the tile images in the tilemap
- var TILESET_SPACING = 2;
  how many pixels are between tile images in the tilemap
- var TILESET_COUNT_X = 14;
  How many columns of tile images are in the tileset
- var TILESET_COUNT_Y = 14;
  How many rows of tile images are in the tileset

5. Finally, load the image to use for the map's tileset. (Make sure this comes before the drawMap() function.

```
      // load the image to use for the level tiles
var tileset = document.createElement("img");
tileset.src = "tileset.png";
```

6. You should now be able to run your program and see the map loaded.



Exercises:

If you haven't already, make your own level. We are going to add side-scrolling in a later tutorial, so feel free to make your level wider than one screen.

There are numerous resources on the web where you can download tilesets. Some useful ones are:

- http://opengameart.org/
- http://www.lostgarden.com/search/label/free%20game%20graphics