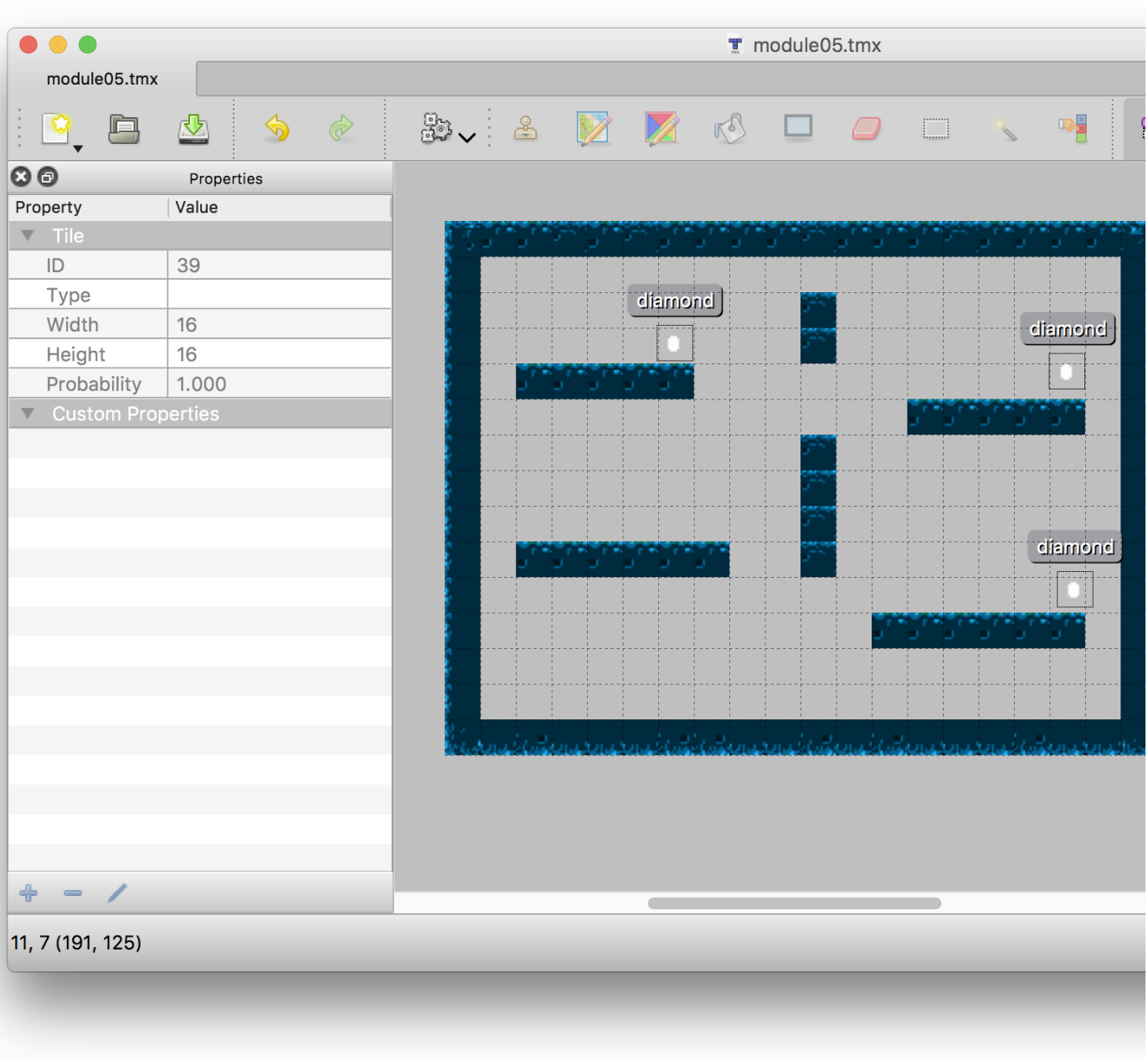
Week 7: All About Spritesheets + Texture Atlases

Going Deep into Spritesheets + Texture Atlases

In last week's materials, we looked at the basics of sprites. In this week, we go deeper into the capabilities that the Phaser.js JavaScript library offers us for creating complex game worlds filled with sprites. We will explore building complex game levels using spritesheet or texture atlas editors, such as the free Tiled, available for Mac OS and Windows.



Sprite Sheets

Sprite sheets are essential for creating multi-frame animation sequences we can attach to sprites.

> "Texture Packer is an essential part of our daily workflow. Every bit of GPU memory helps when dealing with mobile html5 games, so intelligent packing of assets is a must. And Texture Packer has all the features we need to effortlessly create atlases for our games."
> *Richard Davey (@photonstorm) - Creator of Phaser*

There are at least two choices for composing spritesheets and texture atlases: Texture Packer and Tiled. Both applications can be found online:

- Texture Packer is found at **https://www.codeandweb.com/texturepacker (https://www.codeandweb.com/texturepacker)**
- Tiled is found at **http://www.mapeditor.org/ (http://www.mapeditor.org/)**

With a sprite sheet, we cut down on server requests. Rather than hitting the server ten times for ten different images, only one request is sent. That one file may have a larger relative file size, but it will likely be smaller than the sum of all the individual files. Smaller overall size and fewer HTTP requests makes for better performance.

Once we have the full sprite sheet loaded into memory, we then need to display only the section of the image we need. When we load a sprite sheet asset in Phaser, we specify the dimensions of each individual sprite frame within the sheet.

```
game.load.spritesheet('myasset', 'assets/asset.png', 40, 30);
```

When we use the asset, we must let Phaser know which frame to display for the sprite.

```
mysprite = this.game.add.sprite(100, 300, 'myasset');
mysprite.frame = 2;
```

The rendering of sprites can be sped up dramatically by using sprite sheets.

This topic is discussed in textbook section 6.8.1.

Fixed to Camera

If you need to keep a sprite visible within the view of the camera in your game, consider fixing the sprite to the camera's position.

```
mysprite.fixedToCamera = true;
```

This topic is discussed in textbook section 6.8.2.

Texture Atlases

In Phaser, texture atlases are a variation on a sprite sheet. While individual frames in a sprite sheet must all be the same dimensions, so that a sprite sheet is effectively a uniform grid of individual frames, the frames in a texture atlas can each possess their own unique dimensions. Because the metadata about the individual frame dimensions is not uniform, an additional data structure must be created to contain the frame dimensions. This metadata must be stored in a separate file in either the Starling XML format (from Adobe Flash) or using JavaScript Object Notation (JSON).

This topic is discussed in textbook section 6.9.

Tile Sprites

Whenever you wish to create a scrolling environment composed of sprites, you will face the challenge of managing a seamless scrolling scene. Moving chunks of the scene from one side of the screen to the other as they go off-stage becomes a necessary task.

Tile sprites are Phaser's answer to providing you support. You can still position the overall sprite using its **x** and **y** properties. But tile sprites also support the management of the scrolled position of the sprite sheet used with a **tilePosition** property that contains both **x** and **y** values.

```
mybackground.tilePosition.x -= 4;
```

This topic is discussed in textbook section 6.10.

Maps

Maps are a solution for composing complex game world environments – levels if you will – filled with a rich collection of sprites. Trying to write code to place and position every sprite in a complex level is exhausting and an unnecessary challenge, thanks to maps.

Phaser supports two approaches for defining game world maps: scenes and graphs. Scenes can be defined in JSON files and refer to loaded assets that can be used for sprites defined by the scene's JSON data. Graphs are more rigid than scenes, more like sprite sheets in their fixed width and height.

This topic is discussed in textbook section 6.13.

Tilemaps

A tilemap makes it possible to define an entire level's map in a very compact amount of memory space. By breaking up the environment into a grid of reused assets, a tilemap makes defining a complex and rich game world a straightforward and structured task.

The Tiled map editor – found at **http://www.mapeditor.org/** **(http://www.mapeditor.org/)** – provides an easy-to-use tool for composing maps to use with Phaser.

This topic is discussed in textbook section 6.13.2.