

PHASER WORLD

DECEMBER 2017

ISSUE
108

Rogue Fable II

THIS WEEK...

ROGUE FABLE II

PHASER BOILERPLATE 1.3

JIGSAW PUZZLE

CHROMA KEY TUTORIAL

Welcome to Issue 108 of Phaser World

And we're in December already. Yikes. Where on earth has 2017 gone?! As Ludum Dare kicks into gear and the GitHub game jam come to a close we'll have

some great new jam games in the coming weeks. This week though, be very careful with Rogue Fable II because it has the ability to cast a Time Drain spell on you! One minute you're happily exploding mushrooms with fireballs, the next it's the small hours of the morning. Dangerous stuff.

There's a monster Dev Log report this issue too, with loads of incredible updates to the Tilemap API in v3 to play with. Dig in and have fun!

So, until the next issue, keep on coding. Drop me a line if you've got any news you'd like featured. You can reply to this email or grab me on the Phaser [Slack](#) channel.



The Latest Games



Game of the Week

[Rogue Fable II](#)

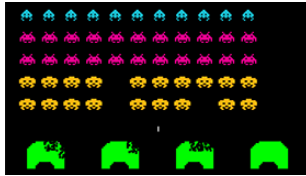
Lose yourself in this superbly playable rogue-like with 6 characters, 7 zones and loads of items to collect.



Staff Pick

[Jigsaw Puzzle](#)

The phenomenally successful jigsaw game comes to Facebook Messenger.



Hobo Digital Space Invaders

A fun 404 page that offers up an accurate rendition of the classic Space Invaders.



Zehkiflorn

A lovely interactive animated web site inspired by 90s screensavers.



pickupon

An infographic bot designed for visualizing your Slack team map.



What's New?



Phaser Boilerplate 1.3

A quick and easy boilerplate for your Phaser projects, now updated with new plugins added.



Mario Platformer Tutorial

A comprehensive tutorial on creating a platformer game using Phaser Editor.



Chroma Key Tutorial

How Phaser was used with chroma key techniques for a video project.



Phaser Tutorial for Beginners

A comprehensive 20 part Phaser beginners tutorial in Polish.



Rubble Games Blog

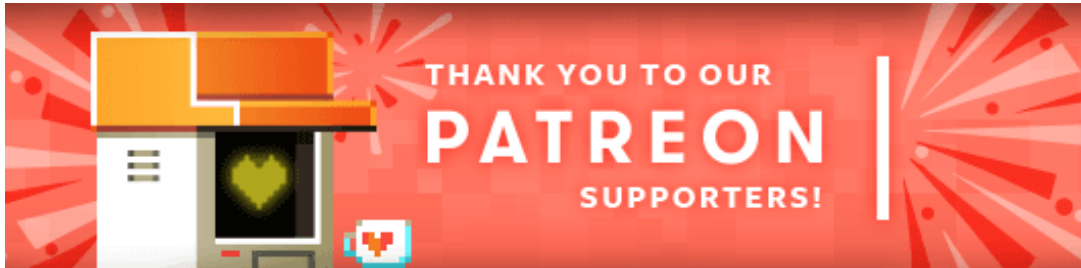
Read about the game development behind Zombie Cannon Attack and other Phaser games.



GameAnalytics for Phaser

Want to see everything that happens in your game? Use GameAnalytics in your

Phaser games with this tutorial.



Thank you to our new **Phaser Patron James**. Also, thank you to **Enriqueto** for increasing his pledge.

Patreon is a way to contribute towards the Phaser project on a monthly basis. This money is used *entirely* to fund development costs and is the only reason we're able to invest so much time into our work. You can also [donate](#) via PayPal.

Backers get forum badges, discounts on plugins and books, and are entitled to free monthly coding support via Slack or Skype.

Game Asset of the Week



Making a Christmas themed game? Check out this [Winter Game GUI pack](#). With a variety of buttons, borders, sliders, stars and icons for your holiday creation.



Dev Log #108

You can try all new features shown in this Dev Log in [Phaser 3 Beta 13](#). Grab it from GitHub pre-built, or from npm using the beta tag.

This issue we've updates to the Sound API and massive report on the state of the Tilemap API with loads of new demos and example code. Felipe has been hard at work implementing pre-multiplied alpha support, which we'll cover next issue. And I've been all over the API. From fixing bugs in the matter.js API, to adding features into Group, to adding support for right-to-left Text objects (brand new to V3). It's been quite a week with progress across the whole API. I also had time to create a little game which I'll cover next issue. For now, dig into the two huge updates we've got for you now ...

Sound API Update

Here's the latest on the new Sound Manager from Pavle:

This week I took some time to test what's been done with the Sound API on mobile devices and came across a few issues

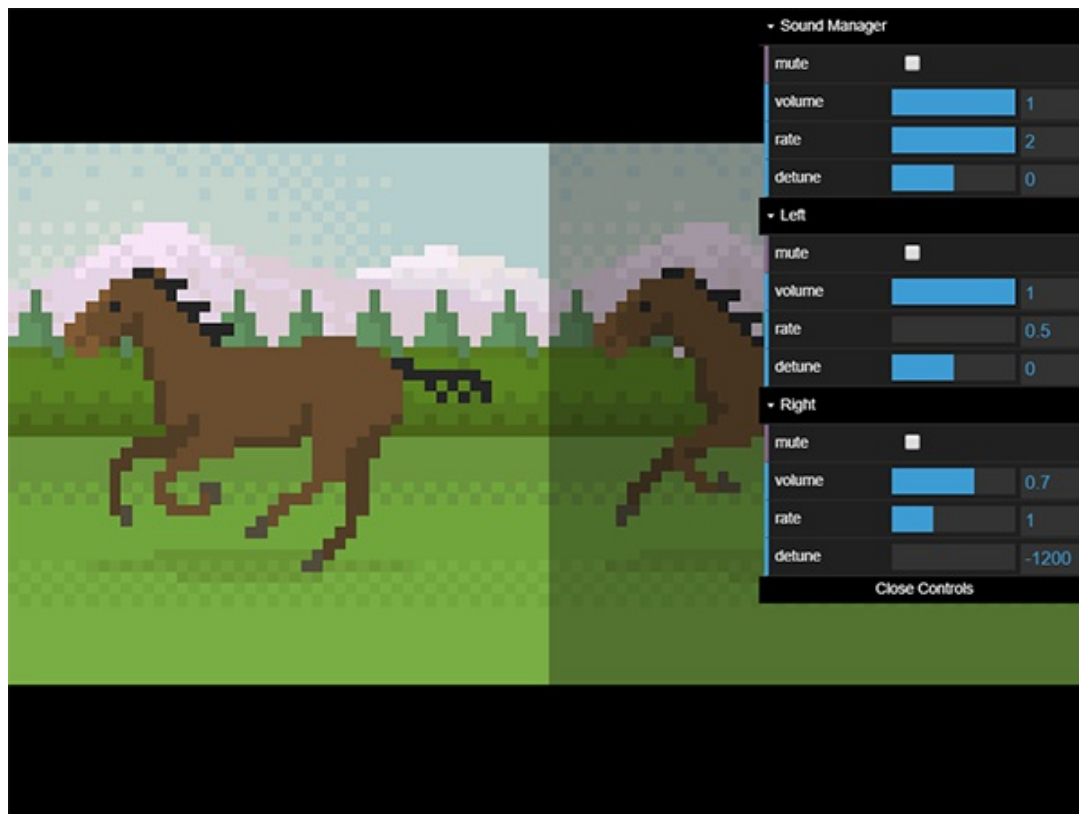
One was related to the detune option - turns out that Webkit implementation of Web Audio API (used mainly in Safari and older browsers) doesn't support detune option natively.

Some of you guys on slack pointed out that my explanation of detuning few issues ago was a bit vague so I'm gonna go into more detail now :)

Both rate and detune settings determine the speed at which the audio will play. For example, setting the rate to 2.0 will have the same effect as setting detune to 1200 and will double the speed (and raise the pitch by one octave). On the other hand setting rate to 0.5 will have the same effect as setting detune to -1200 and will halve the playback speed (and lower the pitch by one octave), and they can be used in combination.

Having both of these options allows you much more control over the playback speed and I didn't want to ditch detuning just because it was not supported by all Web Audio implementations. I did some research and managed to accomplish the same effect, that native detune option provides, by using some clever math. Another good thing is that this approach can be used with HTML5 Audio as well, which of course does not support it natively either, so you can expect the same behavior across all browsers and platforms.

Here is an example that demonstrates how the rate and detune can be used along with mute and volume options:



Da-da-dun-de-dun-dun-deee-dun

Another neat thing is that last week's work done on adding markers made it very easy to add support for [audiosprite tool](#). I have remade one of the markers examples to use audiosprite to show you how to make use of it in your own projects:

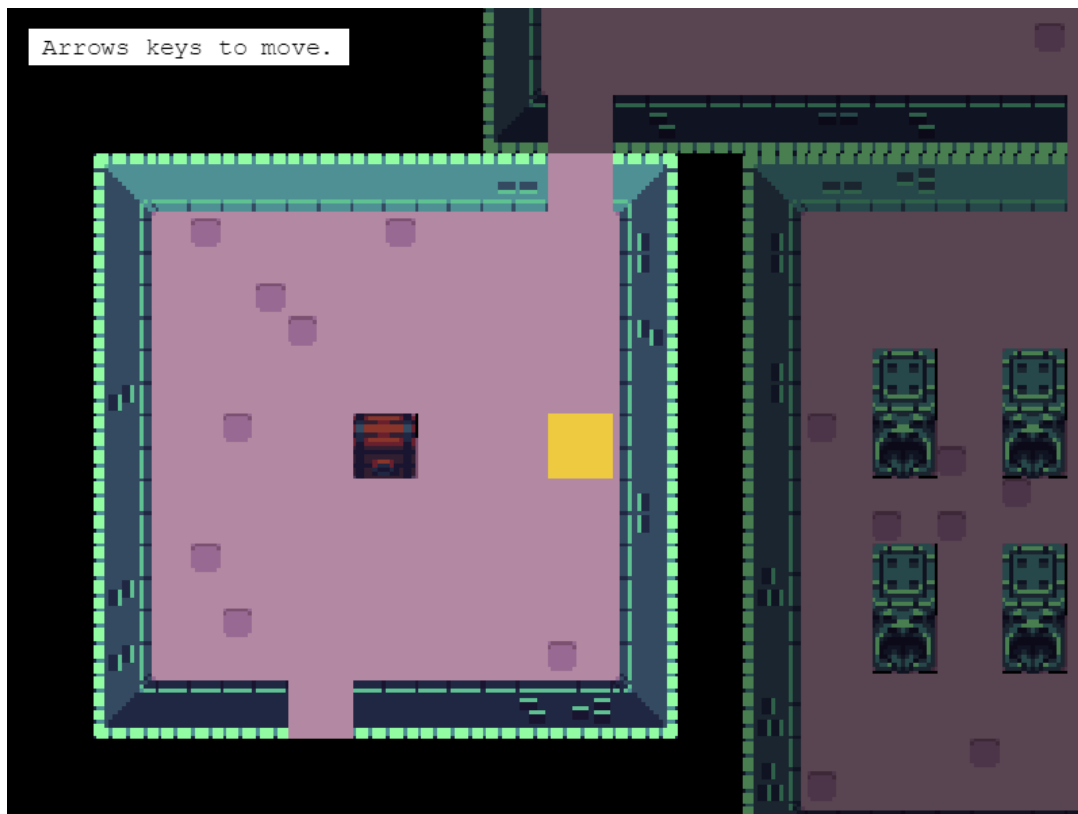


Pew pew

Tilemap API Update

Michael has completed his work on the new Tilemap API. It's looking incredible and here's an extended feature on what you can now find in Beta 13:

This week was focused on getting the final touches on the Tilemap API in place. Aside from Physics integration, the Tilemap API is stable, documented and chock full of examples. You can skip straight to [Phaser Labs](#) and see the 20+ new examples, or jump into the weeds with me below. One particular example worth checking out is the roguelike dungeon generator. It generates a random map with connected rooms and then uses some of the new API features to fill in the details:



Explore a randomly created dungeon

Two weeks ago, I demoed how you can create a map with static or dynamic layers from a JSON file, a CSV file or a plain ol' 2D. Now we can jump into some of the fun things you can do with those maps, once they've been created. A quick refresher first:

```
1 var map = this.make.tilemap({ key: 'map' });
2 var tileset = map.addTilesetImage('tiles');
3
4 // A static layer is built for speed - you can't change the tiles or apply per-tile effects
5 var staticLayer = map.createStaticLayer('Background Layer', tileset, 0, 0);
6
7 // A dynamic layer trades some speed for ability to change tiles and use per-tile effects
8 var dynamicLayer = map.createDynamicLayer('Foreground Layer', tileset, 0, 0);
```

Click the code to get a refresher on loading a map in v3

One area where things have been improved in the new API is with tile operations – getting, putting, removing, etc. tiles. In v2, TilemapLayers and Tilemaps had different methods – you could shuffle the tiles by `map.shuffle(...)`, but you couldn't call `layer.shuffle(...)`. Conversely, you could `layer.getTiles(...)` but not `map.getTiles(...)`. All the tile methods in v3 - `GetTileAt`, `HasTileAt`, `WorldToTileXY`, etc. - can be invoked on a map or on a static/dynamic layer.

```

1 var map = this.make.tilemap({ key: 'map' });
2 var tiles = map.addTilesetImage('tiles');
3 var layer = map.createDynamicLayer('Layer 1', tiles, 0, 0);
4
5 // You can use tile operations on either map or layer:
6 var tileA = map.getTileAt(2, 2); // -> Tile {index: 30, x: 2, y: 2, width: 32, ...}
7 var tileB = layer.getTileAt(5, 2); // -> Tile {index: 30, x: 5, y: 2, width: 32, ...}

```

Click on the code to see an example of using map and layer interchangeably

Note: Tilemaps keep track of the current layer and apply tile operations on that layer, if no explicit layer is specified. There's only one layer in that example, so you can use `map.getTile(...)` and `layer.getTile(...)` interchangeably. With multiple layers, you can set a map's current layer using the `map.layer` property or `map.setLayer(...)`. Check out the example below to see that in action:



Click on the screenshot to see an example showing how to select layers

Of course, there is an important exception to this idea of unifying map and layer methods! Any method that mutates tiles (e.g. `shuffle`, `removeTileAt`) is not available on a `StaticTilemapLayer`.

```
1 // Assuming we have a 'map' with a single 'staticLayer':
2
3 staticLayer.removeTileAt(0, 0);
4 // ^ This would throw an error
5
6 map.removeTileAt(0, 0);
7 // ^ Sneaky, but you'll get a friendly console warning that you can't do that :)
```

I won't bore you with listing all the tile access & manipulation methods, but suffice it to say there's one for each method from v2 and then some. I'll highlight some of the new features.

You can now get tiles within a shape (and optionally filter out non-colliding, empty or non-interesting tiles):



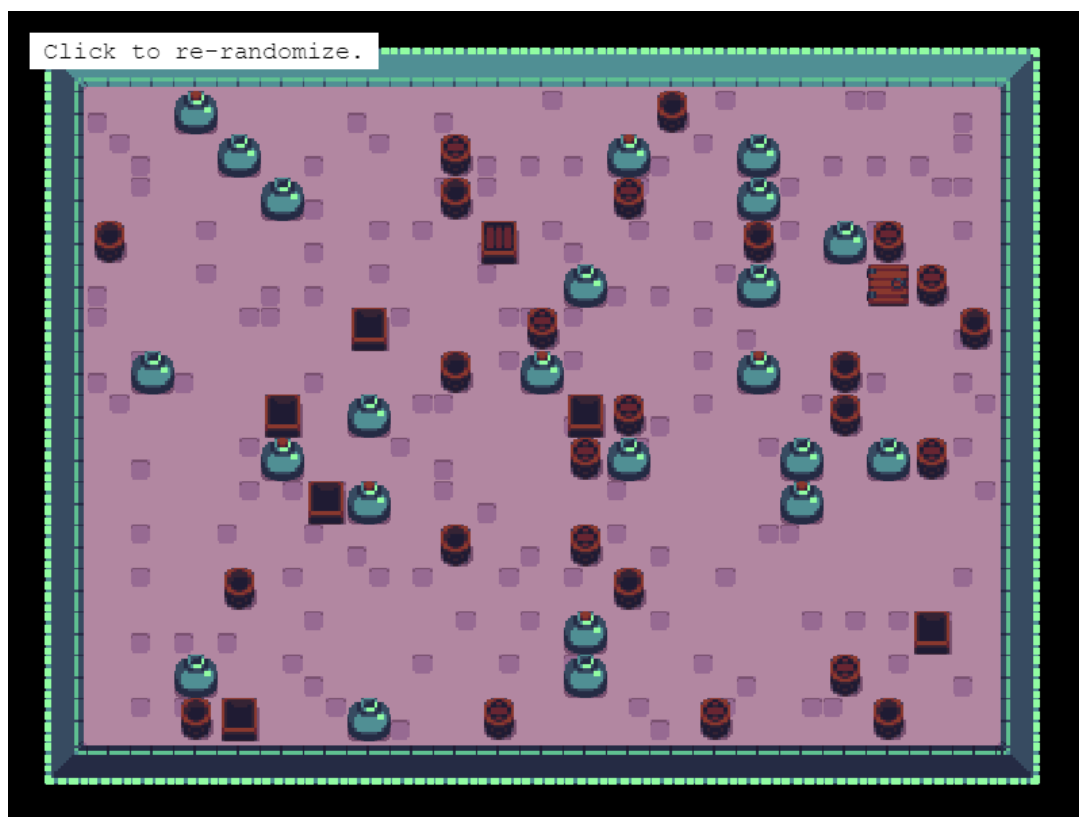
Click on the screenshot to check out the interactive example

If you are a fan of functional programming, there are a few tile access methods that mirror JavaScript's array methods: `filterTiles`, `findTile`, `forEachTile`.



```
1 // Get an array of opaque tiles from anywhere in the layer
2 var opaqueTiles = layer.filterTiles(tile => tile.alpha === 1);
3
4 // Find a specific tile within a 20 x 20 tile area starting at (5, 5)
5 var collidingTrapTile = layer.findTile(
6   tile => tile.collides && tile.properties.isTrap,
7   5, 5, 20, 20
8 );
9
10 // Full parameter listing for both methods:
11 // callback, context, tileX, tileY, width, height, filteringOptions
```

There are some upgrades to tile manipulation. In addition to `randomize`, there's a `weightedRandomize` method that gives you more control over randomizing tiles. There is also a `putTilesAt` method, which gives you the ability to “stamp” out rows or grids of tiles into a map really easily:

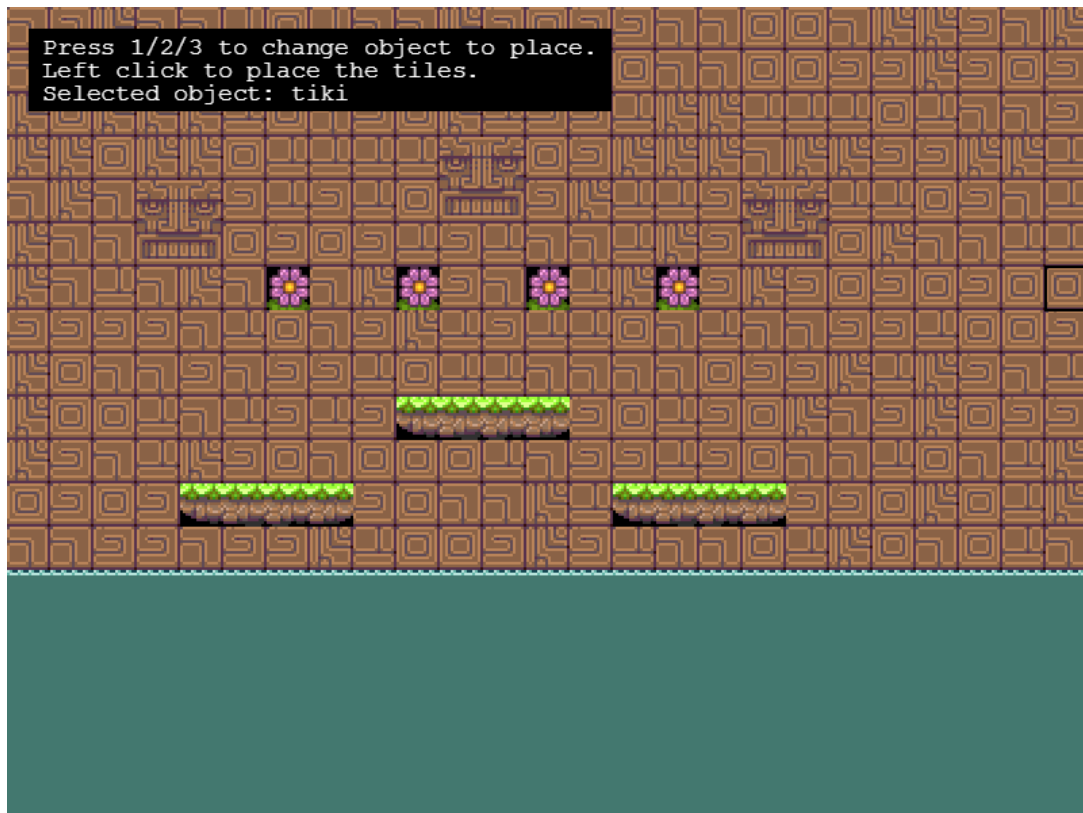


Click on the screenshot & check out the `randomizeRoom` function

```

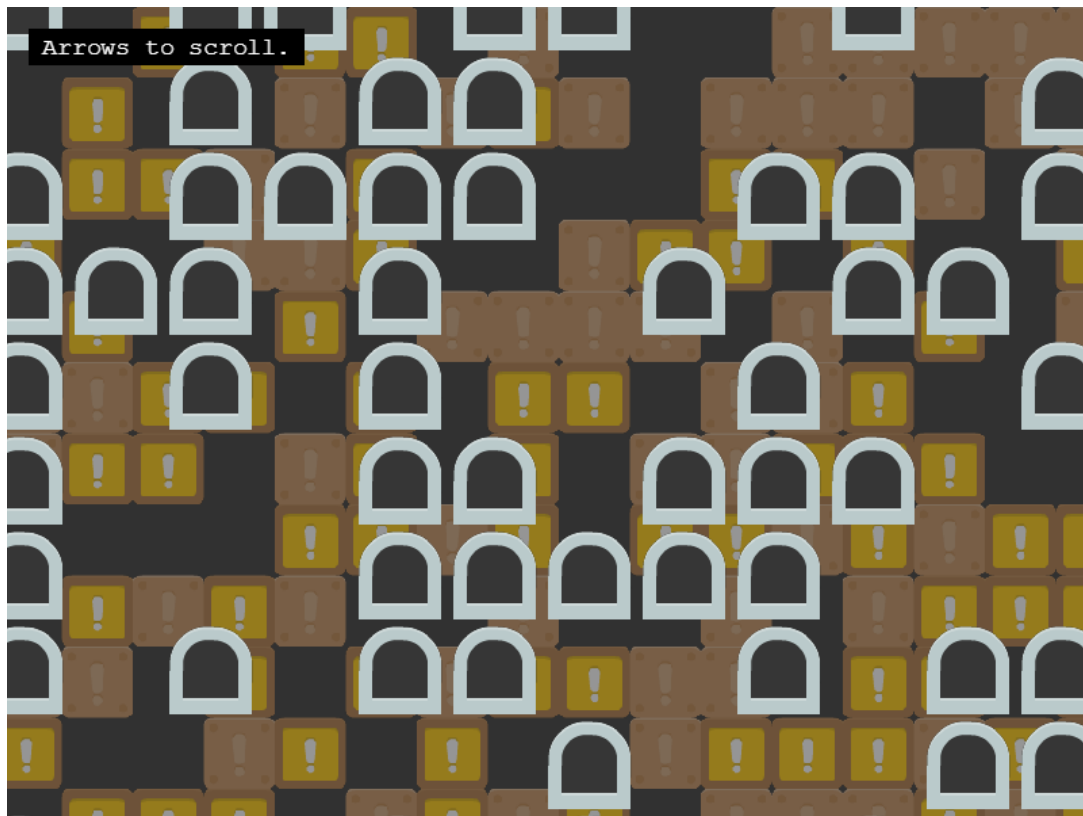
1 // You can place an individual tile by index (or by passing in a Tile object)
2 map.putTileAt(15, 0, 0); // ID 15 at (0, 0)
3
4 // Or, you can place a row of tiles at a given location
5 map.putTilesAt([ 104, 105, 106, 107 ], 2, 0);
6
7 // Or even better, you can also place a 2D array of tiles
8 map.putTilesAt([
9   [ 49, 50 ],
10  [ 51, 52 ]
11 ], 0, 0);

```



Click on the screenshot & check out the update function

Instead of creating layers from 2D arrays/CSV/JSON, you can also create blank layers in v3 and then use any of these handy tile manipulation methods to create your map on-the-fly. Although blank layers must be dynamic, there's a way to convert a layer from dynamic to static, so you get the benefits of the dynamic map methods when initially creating a level, but then the zippy performance of a static map when it comes to rendering. Check out these examples:



Click on the screenshot to see how to create & fill blank layers

```

1 // Creating a blank tilemap with the specified dimensions
2 var map = this.make.tilemap({ tileWidth: 16, tileHeight: 16, width: 20, height: 20 });
3 var tiles = map.addTilesetImage('tiles');
4 var layer1 = map.createBlankDynamicLayer('layer1', tiles);
5
6 // Fill with random tiles by specifying tile IDs
7 layer1.randomize(0, 0, map.width, map.height, [ 44, 45, 46, 47, 48 ]);
8
9 // ... or use shuffle, putTileAt, putTilesAt, etc.
10
11 // Once you are done making modifications, optimize it by converting the layer to static
12 // Note: you can't change the individual tiles after this point!
13 layer1 = map.convertLayerToStatic(layer1);

```

Click on the code to see how to convert dynamic layers to static

Another improved area of v3 is tile position world position conversions. Any method that has “world” in its name is referring to pixel position with layer scale & position and camera scroll all factored in. (v2 had a couple different meanings of “world” mixed together.) You can check out an updated previous [example](#) to see that in action with the new `Pointer#positionToCamera` method.

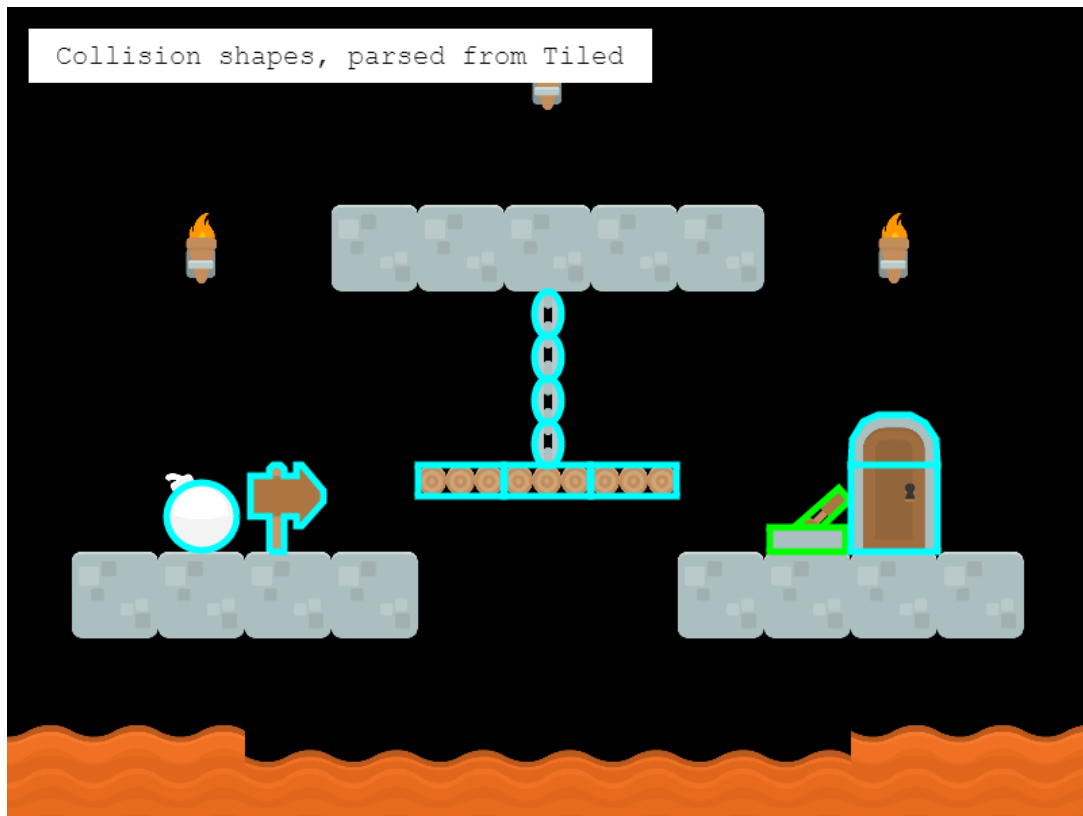
While tilemaps aren’t integrated with the shiny new Physics systems yet, you can still mark tiles for collision. There’s a new ``renderDebug`` method that will allow you to visualize colliding tiles and edges:



Click on the screenshot to see an example

```
1 var debugGraphics = this.add.graphics();
2
3 // Pass in null for any of the style options to disable drawing that component
4 map.renderDebug(debugGraphics, {
5   tileColor: new Phaser.Display.Color(105, 210, 231, 200), // Non-colliding tiles
6   collidingTileColor: new Phaser.Display.Color(243, 134, 48, 200), // Colliding tiles
7   faceColor: new Phaser.Display.Color(40, 39, 37, 255) // Interesting faces, i.e. colliding edges
8 });
9
10 // Or omit one of the style options and you'll get a nice default color
```

The Tiled parser has gotten some upgrades along the way – in particular, it can parse text objects and tile collision shapes.



Click on the screenshot & check out the corresponding code to see the parsing in action.

Whew! If you read all that, feel free to have a look at the [source code](#) on GitHub to see what else is new.

Phaser 3 Labs

[Phaser 3 Beta 13](#) is out and ready for testing.

Visit the [Phaser 3 Labs](#) to view the new API structure in depth, read the FAQ, previous Developer Logs and contribution guides.

You can also join the [Phaser 3 Google Group](#) or post to the [Phaser 3 Forum](#) - we'd love to hear from you!





[CRTC³](#) is a new Amstrad 6128+ cartridge demo that came first at Alchimie 2017. It's a stunning production with some really impressive sections. Remember it's running on an 8-bit machine with a 4 Mhz CPU created in 1985.

Related to the above this video is a [Making of CRTC³](#). It's in French, so enable subtitles if you need, but you can see how each of the scenes was created and it's an amazing trip into 8-bit mastery.

Finally, the [Mobile Gaming Industry Analysis 2017](#) is out. This free PDF (registration required) looks at the top 20 genres present in a dataset of 40K+ games to learn how popularity is changing over time, and which genres have stronger KPIs.

Phaser Releases

Phaser CE [2.9.2](#) released November 10th 2017.

Phaser 3 [Beta 13](#) released December 4th 2017.

Please help [support](#) Phaser development

Have some news you'd like published? Email support@phaser.io or [tweet us](#).

Missed an issue? Check out the [Back Issues](#) page.



©2017 Photon Storm Ltd | Unit 4 Old Fleece Chambers, Lydney, GL15 5RA, UK

[Web Version](#)

[Preferences](#)

[Forward](#)

[Unsubscribe](#)

Powered by **Mad Mimi®**
A GoDaddy® company