

PHASER WORLD

MARCH 2018

ISSUE
117



THIS WEEK...

PHASER 3.2.0 RELEASED

FIVE NATIONS

MAKECODE EDITOR

LICH BEASTMASTER

Welcome to Issue 117 of Phaser World

This issue we're back to our regular format, with features games and tutorials, the dev log and geek links rounding it out. There has been a real flurry of

incredible Phaser games released recently, each one of them a worthy newsletter feature / cover game - so look forward to some really exciting games in the coming weeks!

Phaser 3 development continues to consume the dev team and you can read all about the latest version in this week's Dev Log.

Until the next issue, keep on coding. Drop me a line if you've got any news you'd like featured by simply replying to this email, messaging me on [slack](#) or [Twitter](#).



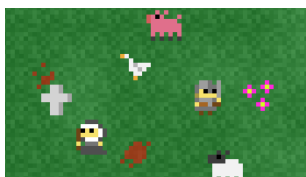
The Latest Games



Game of the Week

[Five Nations](#)

An open source real-time strategy game that narrates the events of a galactic battle between five distinct races.



Staff Pick

[Lich Beastmaster](#)

Be a good lich and send in the chickens to do battle for you in this soul captivating game.



Color Pin

Shoot the pins into the spinning colored wheel, making sure the colors match and your aim is perfect.



Pop Soap

A colorful looking match-3 bubble popper with 10 levels, endless mode and achievements.



Dossier the Ninja Mouse

Take control of Dossier the mouse and fight his way through the sewers of Paris, collecting cheese and wine and defeating the rats.



What's New?



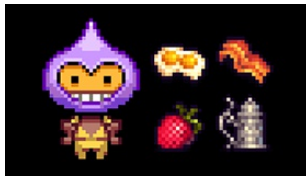
GameFromScratch Phaser 3 Video

A new 40 min video from Game From Scratch focusing on getting started with Phaser 3.



MakeCode Phaser CE Editor

A sample Phaser CE based MakeCode Editor published by Microsoft.



Brunch with Phaser 3

A Brunch skeleton for quickly getting started with Phaser 3 projects.



Ballz Prototype Tutorial Part 2

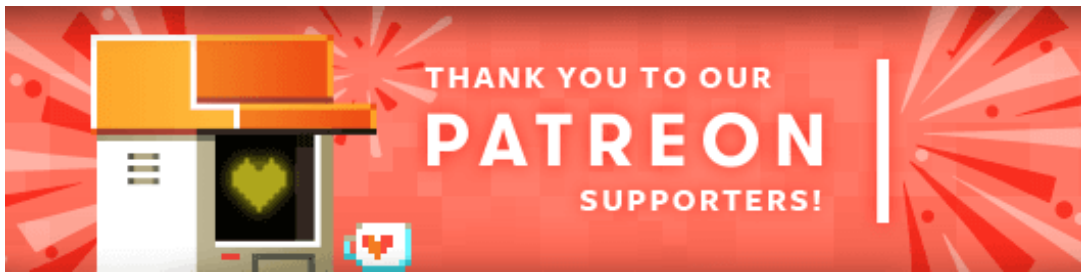
In the second part of this tutorial the ability to add and destroy blocks are put into the game.

```
// Creating a container (options are optional)
container = this.add.container({
  x: 100,
  y: 50,
  alpha: 0.5,
  rotation: 1.3,
  watch: ['x', 'y', 'rotation', 'alpha']
});

// Create a sprite directly on the container, just as you would
sprite1 = container.createChild(200, 200, 'spaceship');
sprite2 = this.add.sprite(200, 200, 'spaceship');
container.addChild(sprite2);
```

Phaser 3 Interim Containers

Basic containers for Phaser 3 until they officially land.



Thank you to our awesome new [Phaser Patrons](#) who joined us this week:

Adrianus Kleemans

Samuel Johnson Stoever

David Ang

Ray Martone

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.



Dev Log #117

Since the release of v3, it has been a bit of a rollercoaster, to say the least. We have carried on prioritizing bug fixes and acting upon community feedback, which has lead us up to the 3.2.0 release. That may sound like quite a large jump considering we're barely a month old but it's only because we're semver'ing properly. Even so, 3.2 is a pretty huge update. It was never planned that way, we actually wanted to keep the updates quite small, but being a point release we had the ability to include API changes, which a few features had been waiting for. This combined with lots of different things coming together at the same time that we'd been working on.

3.2 also contains a massive amount of contributions from the community. Both in

the form of merged pull requests, bug reports and feature suggestions we have acted upon. There's a nice organic process evolving, where some requests and issues are resolved directly by the community, landing as PRs, before we even get a chance to do them, which is just brilliant to see!

With 3.2 and this newsletter out of the way, I'm going to spend the rest of the week working on the documentation. I've been improving the docs piece by piece as I worked on 3.2. Fixing missing data types and other errors I spotted. So it has been a continuous process. But it really needs some dedicated time. I have to confess, it's not going to be particularly exciting. Yet it's essential. Once I get stage 2 of the docs process complete we can finally get some TypeScript defs out, because it appears some people are just unable to work without code-completion ;) That aside, let's dive into this Dev Log.



As is custom by now the 3.2 Change Log has been split into 'New Features', 'Updates' and 'Bug Fixes'. I'm not going to include everything here because you can read the full log for yourself on GitHub and the website. Let's pick a few interesting entries though:

Render Textures

These are now available for both Canvas and WebGL in 3.2. A Render Texture is a special type of Game Object which you can 'draw' to. You can literally draw any texture frame to it, fill it or clear it. We included the option to set tint (WebGL only) and alpha and other canvas style methods like scale, rotate and translate. Render Textures are nice and fast - there is no gpu thrashing going on and they're pretty versatile too, even capable of being applied as masks.

Render Textures differ from those in v2 because you don't apply them to another Game Object. They are a Game Object in their own right. This means you can position them in the world, rotate them, scale them and so on like you would any other object. This perhaps means the name of them isn't ideal but we honestly couldn't think of a better one, because they literally are textures that you can render to! At the moment we only have 1 example of their use in the Labs but could really do with your help creating some more - so if you make a fun Render Texture example, please do submit a PR for it.

Scene Manager Updates

Several things happened with the Scene Manager in 3.2. First of all, I added in some helper methods like 'moveAbove' and 'moveBelow' to help you with positioning a Scene in the scene list. I also added 'remove' to let you clear a Scene from the manager entirely.

Perhaps more importantly, I recoded the way the boot process works. When the Scene Manager now boots it handles the way the initial batch of Scenes it is given are processed. It will create them all in turn and only then will it start those that need starting. This means that issues such as a Scene not being able to get a reference to another Scene while the game was booting is now resolved. So if you've got some whacky Scene chaining going on, where you jump from one Scene to another quickly at the start, it should now merrily go on its way without needing any custom timeout hacks or preloading.

Resize Handler

There were some important new additions made which allows for you to handle game resizing events more elegantly and easily than before. There was also a nasty offset fix made in the Input Manager, which caused hit areas to fall out of sync in some situations if the canvas changed position on the page.

Game.resize can be called directly and it will now pass on the new dimensions to the renderer, the input manager and all of the Scenes in turn. Scenes will now emit a 'resize' event you can listen to and act upon, such as adjusting sprite positions or camera bounds. We still intend to add a Scale Manager into v3 but you can now do an awful lot of what is needed without it. We've built a fully responsive game as a test case for these changes, where one Scene is kept at 100% browser size containing the background elements of the game, and the gameplay scene is adjusted in scale using Camera baseScale and placement. When we're able we will share the tips we learned in making this game with you all.



Drag Events and Drop Zones

Someone kindly reported that the drop zone examples were all broken. I went in to fix them, expecting to just change the event names, and instead found that drop zones didn't work properly at all, due to a change made in the Input API with regard to only returning the top-most object interacted with.

It took some serious unpicking but I was finally able to restore zones to their previous capabilities, fix all the examples and add in a nice new feature too: the ability for *any* Game Object to now be a drop zone. This means you can set a Sprite as being an interactive drop zone and it'll be a valid drop target for other dragging objects. It may sound small but it's a handy thing to have!

And More!

There are lots of equally important updates in 3.2. The ability to pass any Group to Arcade Physics collide and have it work (before only Physics Groups worked). The ability to pass data into a Scene launch. Fixes to the Text component when passing in non-string values, and so much more.

Please see the full [Change Log](#) and update your builds to 3.2 :)



PNG Animations

Someone asked on the forum if it was possible to create an animation in Phaser 3 that used frames from different PNG files, rather than a sprite sheet or texture atlas. Or would they have to wire together some kind of Timer based system to swap the textures out.

It was awesome to read this because I immediately knew the answer was "Yes!" The Animation Config object is really quite powerful, allowing for fine-grained control over every aspect of the sequence. Something like this is beautifully simple though. Just load the images you want and use the cache keys in the config:

```

function preload ()
{
    this.load.path = 'assets/animations/';

    this.load.image('cat1', 'cat1.png');
    this.load.image('cat2', 'cat2.png');
    this.load.image('cat3', 'cat3.png');
    this.load.image('cat4', 'cat4.png');
}

function create ()
{
    this.anims.create({
        key: 'snooze',
        frames: [
            { key: 'cat1' },
            { key: 'cat2' },
            { key: 'cat3' },
            { key: 'cat4', duration: 50 }
        ],
        frameRate: 8,
        repeat: -1
    });

    this.add.sprite(400, 300, 'cat1').play('snooze');
}

```

Lovely and simple!

The animation system is actually even more powerful than that. You could technically create an animation that contained frames from image files, then into a sprite sheet, then into a texture atlas and back to image files again, and it will let you do this without complaint. I would not recommend it though. Each time you swap texture like this it's going to cause a batch flush in WebGL. Phaser 3 uses a lot of internal batching to keep gl operations down - but splitting animations up like this works against that batching. So use it with caution, but at least know you *can* do it if you want to!

Feature Vote Results

Last issue, and on Patreon, I asked you to vote on what features you'd like to see us work on next. 209 of you voted (thank you!) and we have picked the next features to work on [based on the results](#).

In the first place, with 85 votes, was Containers. No real surprise there. It was always going to be popular, probably more due to their familiarity being in v2 than anything else. Once you're used to doing something a certain way it's hard to change. We will start work on them this week. However, please understand we're not going to compromise the display system we worked hard to build into v3. Don't expect all Game Objects to suddenly gain 'addChild' methods - this will absolutely not be happening.

What we do plan to do is offer a v3 version of the Godot Engines [RemoteTransform](#) node. This allows you to link together multiple nodes (or Game Objects in our case) and choose which properties are remotely controlled via it. You can elect to have rotation, scale, and position influenced by the pairing, meaning an object could control the scale of another, but leave its rotation alone. We can expand this to include visual attributes like alpha, flip and visibility too. It will take a little re-learning from the old ways, but you'll end up with a far more flexible control as a result.

In second place, with 34 votes, was Multi-Touch support. This one actually surprised me. I've built a *lot* of html5 games over the years and played thousands (while adding them to the Phaser site) and I've never once seen a game that used multi-touch support to control it. Phaser v2 had it, so it was there, just rarely employed. Yet, lots of you want it, so it's time to add it back in. Thankfully this is a relatively simple task. When I built the new Input Manager I knew it would be needed eventually, so lots of the internal systems are already set-up for it. Next week I'll start adding it back in fully.

There was a joint second place which was support for Spine Animations. This is a non-trivial request, to say the least. We understand the importance though. Thanks to a discussion on Twitter the Spine team are ready to help us implement support directly in v3. We will be taking them up on this and dev will start once Containers have landed. Realistically that feels like being a couple of weeks away.

As I said before we will revisit the other vote options. It was just to help us prioritize. We've got on-going work patching v3 as well, yet it was a handy guide to let us deliver what was most wanted next.

This Dev Log is getting pretty long so it's time to wrap things up. I will carry on with the v2 migration guide soon and hope to have another nice little exclusive tutorial next issue as well. Until then, once more unto the docs dear friends. Hopefully, you'll see me again on the other side.



The beautiful 16-color pixel art image above is called [The Den](#). Read about the stages involved in drawing it (in French, but Google Translate does a good job)

A great video on [writing NES games in assembly](#).

Better node.js console dumps with [console-probe](#).

Phaser Releases

Phaser 3.2.0 released March 5th 2018.

Phaser CE 2.10.1 released February 18th 2018.

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.



©2018 Photon Storm Ltd | Unit 4 Old Fleece Chambers, Lydney, GL15 5RA, UK

[Web Version](#)

[Preferences](#)

[Forward](#)

[Unsubscribe](#)

Powered by [Mad Mimi](#)®
A GoDaddy® company