Welcome to Issue 92 of Phaser World

A Summer keeps hotting up, so do all of your amazing releases! Strike Tactics is finally out and it's a complete beast. One of the most advanced and impressive

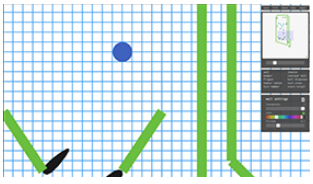
Phaser games in a long, *long* time. Well worth playing. There's also an incredible multiplayer tutorial from PubNub too. Finally, we've got a massive stack of Phaser 3 updates including some beautiful new effects and features.

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

The Latest Games

Strike Tactics >>> **Game of the Week**
A fast-paced steamroller RTS. Harvest, build, attack, defend and conquer your opponents in this epic new game.

Silly ways to Die >>> **Staff Pick**
Can you save the citizens from the hazardous situations they appear to have got stuck in?!



Pinball.Cool

Design your own pinball machines, marble runs, pachinko boards, and pythagora switches.

Bouncy - Jump or Die

Jump as fast as you can, or go slow, and avoid touching anything.

Pyrism

Click the hexes and change their color to control the most territory.

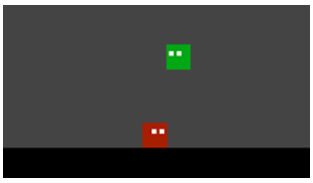
What's New?

Building Your First Multiplayer Game

A fantastic and comprehensive tutorial on creating multiplayer games with PubNub and Phaser.

Spellfall updated to Phaser CE 2.8.2

Updating a 3 year old prototype to take advantage of cached Tweens.

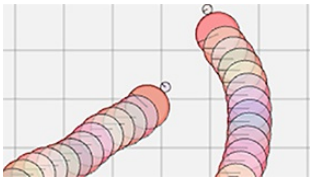


Jump on Enemies Tutorial

A tutorial on managing jumping on enemies using Arcade Physics.

Adding a Toast Message

A tutorial on adding a toast message into your Phaser games.



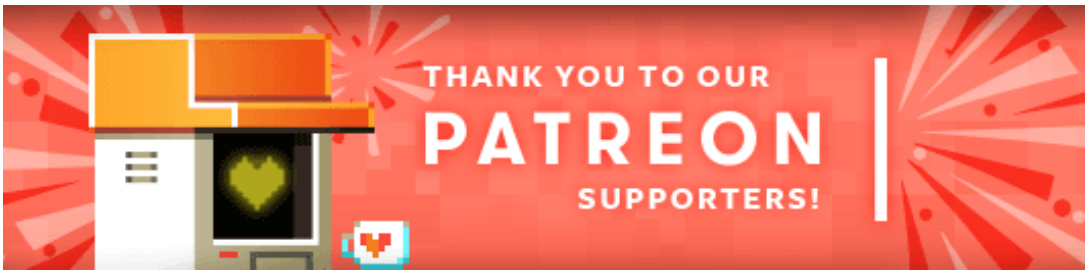
Slither.io Tutorial Part 4

In part 4 of the tutorial series collision detection between snakes is added.

As well as being a cool supporter of Phaser, CrossInstall is also seeking freelancers to join their ranks. You may have seen CrossInstalls work before as they create playable adverts, allowing gamers to get a small taste of an app before downloading it. The playable ads are proving to be far more engaging than traditional banners or videos. CrossInstall is seeking freelancers to join their growing community of diverse and fluent creative engineers to produce these playable ads.

Freelancers should be well versed in JavaScript / HTML5 and Phaser, with experience of using Git and making minor adjustments to Photoshop files (the game assets). The primary responsibility will be to build the playable ads using their custom tools, many of which are built on Phaser.

If this sounds like something you'd be interested in then contact Adam Sheppard and include a portfolio link in your email, showcasing previous related work (Phaser games or other interactive ad work a bonus): adam@crossinstall.com



Welcome and a massive thank you to the new Phaser Patreons who joined us this week: **Joachim Eckert** and **Yaniv Nagar**. Thank you to **Kraig Halfpap** for increasing his pledge. And also thank you to **Axel Martinez** for the donation.

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 #92

After releasing Phaser 3 Alpha last Monday it has been encouraging to see some of you already starting to use it. From animated gifs on twitter to forum posts, it's great that even without documentation yet you're still able to pick it up and make something.

Thanks to your feedback in the forum we've already acted upon some suggestions and v3 is definitely growing stronger as a result. So please, keep the ideas and observations coming!

Last week I spent some time looking at how we're going to handle documentation for v3. I was never very happy with the templated used in v2 but choices were limited and indeed they still are. However, the tooling itself has evolved a lot. After a couple of days doing research and testing it looks as if jsdoc is still the standard way to mark-up the source. However, when it comes to generating the actual docs from that we've far more choices than ever before. Right now I'm trying to decide between using Sphinx JS, which is a JavaScript port of the absolutely fantastic Sphinx documentation system from the Python world, or documentation.js which looks like it will give us far more control over the output of the docs. I need to spend some more time evaluating them both before deciding but I spent a day cleaning up the v3 source code in preparation for this. It will be great to get docs added, even if only for the more stable and fixed classes as it will help you all test a lot more when it comes to the first Beta release at the end of August.

Custom Blend Modes

As part of my tidying-up work last week I noticed that the blend modes weren't working in Canvas. Amazingly they worked fine in WebGL, but not canvas (usually it's the other way around!). So I debugged it and resolved the issue,

enabling all of the canvas blend modes again. Here's an example of the Difference blend mode running in canvas:

Difference blend mode

As a result of working in this part of the API I also knocked off a few tasks from our trello board including the addition of custom blend modes. This is a WebGL only feature but a damned powerful one and allows you to create your own blend modes within the renderer. All you do is call **renderer.addBlendMode** and pass it the mix values and the blend equation to use and it will return a blend mode index you can apply to any Game Object that supports blend modes.

The blend mode can use either 2 or 4 values along with the equation. Here's an example re-creating the classic Mr. Doob blend mode test in v3. Use the drop-down menus to alter the src and destination mix:

Mix the blend modes

There is also an updated version using the blend separate function. This allows you to specify both the source and destination alpha as well as color, allowing for more complex blending. In the right hands, you can make some really neat effects using this feature including effects such as 'knock out' or source-atop quite easily.

Introducing the Light Layer

For the past couple of weeks, while I've been busy preparing the Alpha, Felipe has been working on a new v3 renderer feature in isolation from the rest of the API. After concluding his tests and evaluating the features we decided to merge it into the core API and last week a brand new Game Object arrived called the Light Layer.

Lights are something I've personally wanted in v3 for a long time. Ever since the original renderer tests we did with Pete way back in the mists of 2015 we had lights in there. And I'm glad to say they're back and better than ever before.

We took inspiration from several sources but Felipe coded it all from scratch to best fit in with the v3 API. He has built both forward rendering path and deferred rendering path pipelines for the lights as well as soft shadow support. Currently in v3 is the forward rendering implementation, with deferred and shadows arriving

this week (so we'll cover them next issue).

For a great technical explanation of the difference between forward and deferred rendering please read this Tuts+ article.

Creating Lights

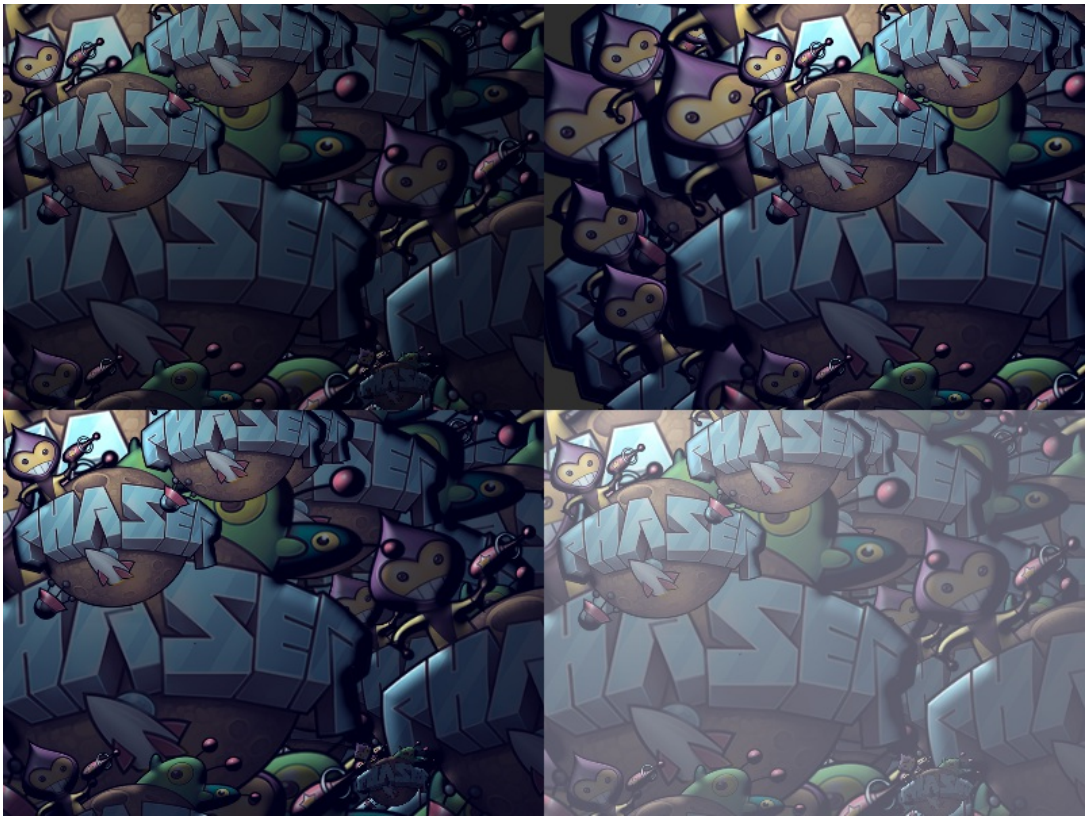
The Light Layer works in a very similar manner to the Effect Layer. You create a layer instance and then use local methods to add lights and sprites to it. With forward rendering we hard code the limit to 10 lights maximum, to avoid it ballooning into millions of fragment operations per frame. With deferred lighting, there will be no maximum.

When Lights are added to a Light Layer you can then control their position and z-depth, as well as their color (per color channel) and attenuation. These values can be directly modified, perhaps via a tween, or by hooking a light to a player sprite or pointer input. When adding a Sprite into a Light Layer you can provide a normal map, which is used to calculate how the light is projected across the texture. In the following example move the mouse around and click to drop a different color light:

Click to drop a light

As you can see the normal map used for the sprite allows the sprite to look almost beveled as the light passes over it. Once you drop several lights the

colors all start to blend together, creating some lovely effects. We used the software Sprite Illuminator to create the normal maps for all our demos, however, there are free online web based tools too. In the following example you can see the lights working even in a multi-camera set-up:



Multi-camera lights

Depending on the source texture you can create some beautiful effects. Here we tween a light across the surface of a brick texture with a smoothed normal map providing the depth effects. The light color is also tween slightly, changing the look of the whole scene as it passes across:

Stone lights!

There is still more work to be done with the lights. For example, I spent most of Friday overhauling the Loader so we can easily pass in a normal map without needing to break out into another Loader call. Plus there are some issues with the display list getting corrupted with multiple light layers. These will all be ironed out this week and deferred implemented. Personally, I'm most excited about the shadow effects we've built and can't wait to show them to you in the coming weeks :) They can add such stunning atmosphere to a scene. It's a great feature to have in there for launch for sure.

Phaser 3 Labs

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. The group is for anyone who wishes to discuss what the Phaser 3 API will contain.

These retro Lego technology kits are insanely cool!

This 1hr session video from Siggraph 2017 had 10 speakers talking about a WebGL2 feature each. Including the guys from ShaderToy, Babylon.js and more. Well worth a watch!

JS13K 2017 starts in 5 days time - feel like making something tiny? Get involved! We're sponsoring two prizes this year with Amazon gift vouchers :)

Further Reading ...

Phaser Facebook Group
GameDev.js Weekly Newsletter
HTML5 Game Development
Lostcast

Phaser Releases

The current version of Phaser CE is 2.8.3 released on July 24th 2017.

The current version of Phaser is 3.0.0 Alpha released on July 31st 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.