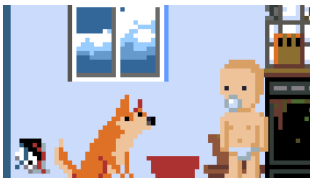
Welcome to Issue 88 of Phaser World

Wow, it's so hot here! I'll be honest, it's pretty hard working at 100% capacity when the air feels so muggy, but there is plenty this issue to take your mind off it.

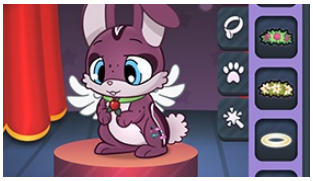
From a great little pixel-art adventure game to a Phaser hackathon with prizes to take part in. Also, some great tutorials which are wonderful to see being created.

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

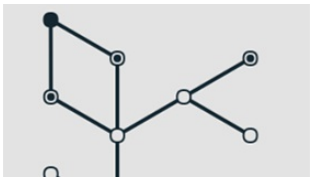


Nobody Puts Baby in the Corner >>> **Game of the Week**
A beautiful point and click pixel art adventure game.



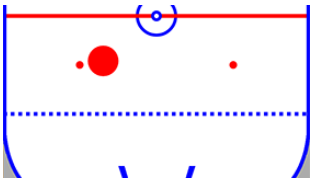
Princess Pet Studio >>> **Staff Pick**
Design your own royal pet in this cute animal dress up game.

Learn Japanese with Games
Play games and learn Japanese as you do it.



Dead Sticks

A minimalistic puzzle game with a nice and simple concept.



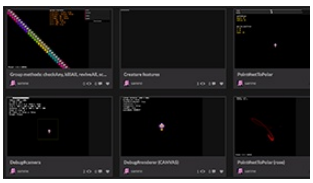
Two Man Hockey

A fun multiplayer version of the classic air hockey.

What's New?

Phaser Hackathon is Live!

Join Zenva's #GameDev Compo – Make a Game and Win Prizes.



Phaser CE Examples Collection

A new collection of Phaser CE specific examples on CodePen.

Creating a Hexagonal Minesweeper

Learn about creating hexagonal tile based games in this comprehensive Tuts+ tutorial.

Creating Games with Phaser Part 7

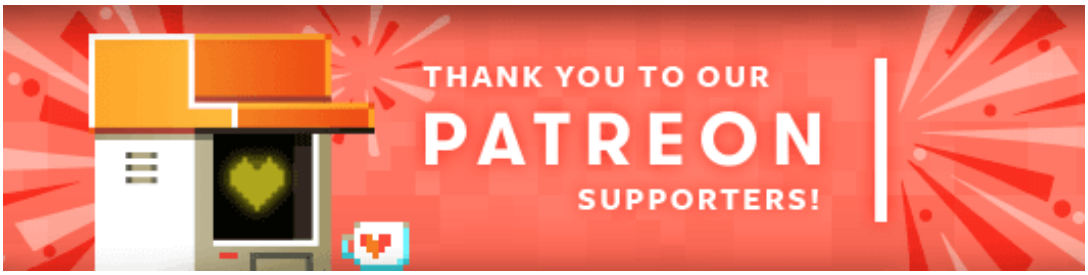
In Part 7 of this new video based tutorial course learn about adding sounds and music into your games.

Phaser Facebook Group

Join a new Phaser specific developers group on Facebook.

Creating a Simple Game Series

A 9 part video series on creating a simple Phaser game from scratch.



Welcome and a massive thank you to the new Phaser Patreon who joined us this week: **Nathan**.

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 #88

Now that the newsletter comes out on a Monday it means the Dev Logs are discussing all the work that took place *last* week. And there was an awful lot of it last week. This is due to the fact we're getting very close to the Phaser 3 Alpha release. Things are getting hectic! and will get more so over the coming weeks. It's a good feeling to know the Alpha is so close, although I'll be honest, it's a mixture of excitement and fear! This Dev Log we're jumping around a lot because development touched nearly every aspect of the API.

Restructuring the Structure

For a few months now I had wanted to restructure both how internal classes were coded and also move all of the State systems into a new location. After a few late nights, both of these tasks were achieved. Every internal class now uses the exact same format and syntax, keeping the API consistent internally. This will make it easier for those who delve into the code. It will also make our transition to ES6 easier next year, as we can swap out our class structure for native classes.

Even though v3 is very function driven it is not exclusively function orientated and we still use classes internally where it makes sense. Don't forget that all modern browsers optimise heavily for this pattern and while it's easy to say everything ought to be a function, in practice you have to balance it carefully. It's vastly different to how Phaser 2 handled it though. For example, take the Point class. In v2 this class had no less than 30 methods defined on the class instance itself

and another 19 on the prototype. This is because Point doubled-up as both a geometry class, which is where it lived in the name space and was also used internally where normally a vec2 would have been. Even the Circle class, which didn't have any such math overlap still had 18 methods defined on it.

In Phaser 3 we've been careful to avoid the feature set of one object spilling across multiple areas. The Point object is still a class, so browsers can recognise it and compile it internally, yet it now has just 1 method. Related Point functions like GetCentroid, Perp and Clone still available of course but not on the actual class itself. It's a shift from v2 and will take a little getting used to but we're seeing it pay dividends in performance.

A Tint of #ff0000

Tint support has been part of v3 since pretty much the beginning, exposed in the shaders and batches. However after some restructuring a few months ago, it hasn't been working, that is until last week when I went through and fixed it all. Tinting is a WebGL only feature for now and a tint can be applied as either a single color, or given as 4 separate colors allowing you to tint from each corner of the game object:

Click to change the tint

Click to change the tint

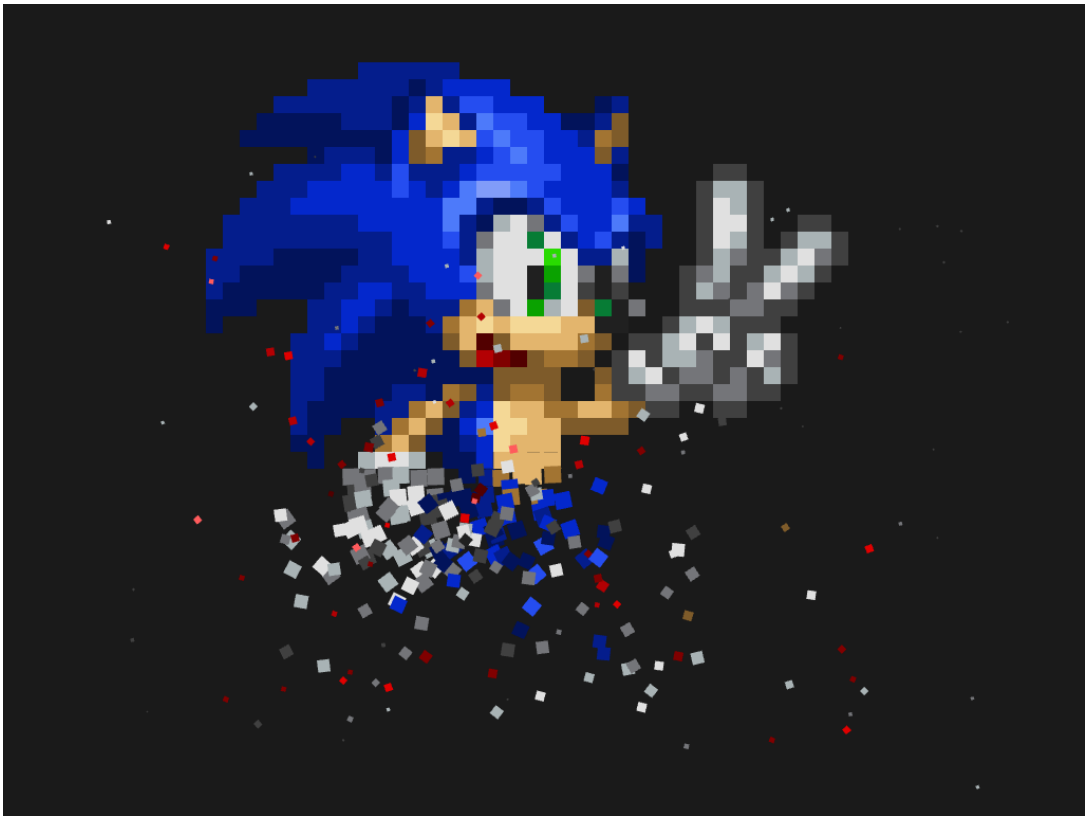
Tinting can be done either directly by modifying one of 4 properties, or by using the setTint method. Using **setTint()** allows you to chain calls to Game Objects together, for example, the following is perfectly valid code now:

```
this.add.image(400, 300, 'face').setAlpha(0.5).setOrigin(0).setTint(0xff9955);
```

Finally, you can also set the objects tint via a configuration object. This '3 tiered' setting approach applies to virtually all Game Object features in v3: direct via properties, via chainable setters and via JSON configuration. You can now pick your favourite style or just mix and match them together like I do.

As well as just tinting images we also extended it to Text objects, Tile Sprites, Tilemap Tiles and this week we'll add it to Bitmap Fonts. Being able to tint Text can create some lovely effects (click it to see the color animation in action):

You can also tint animations or texture atlas frames and unlike v2 it won't change every single instance of that frame, only the Game Object you specifically targetted. Tinting is extremely fast in WebGL, costing next to no extra computation time, allowing you to do cool things like this (where each 'pixel' is a single tinted image)

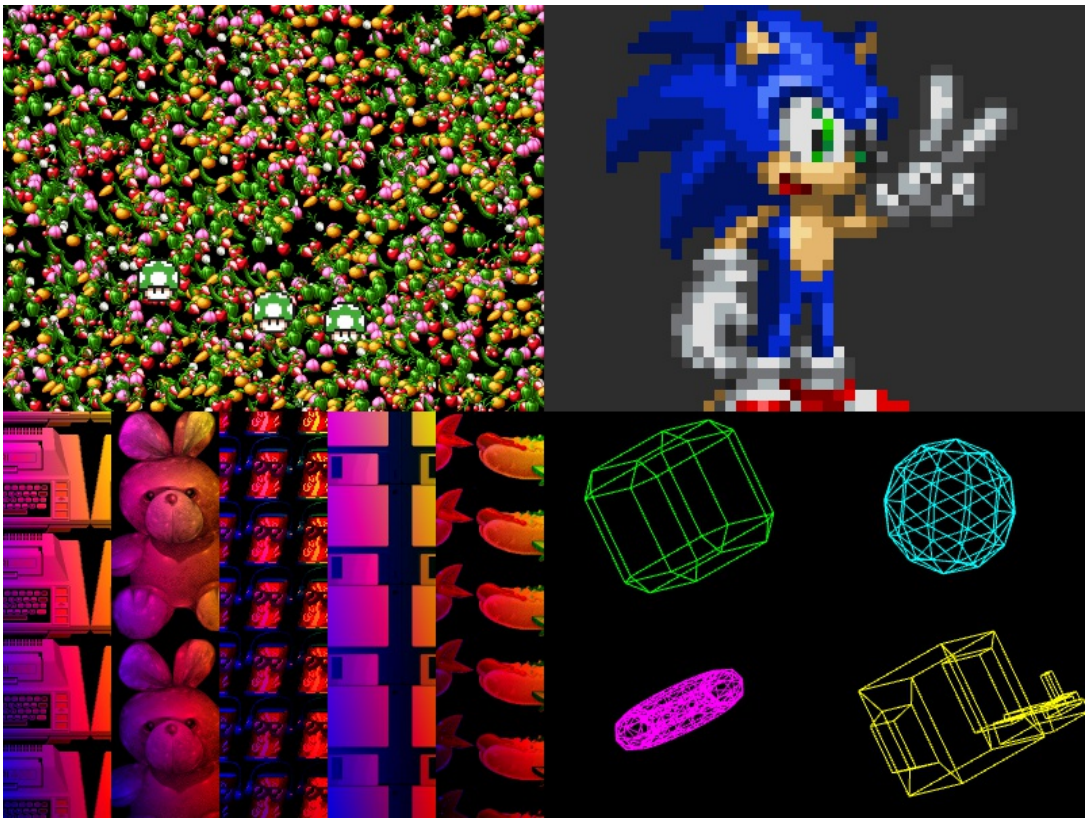


Multi-Camera Demo

If you hang out on the Phaser slack channel then you may have already seen this demo, as I shared it there last week after completing it. This demo is an example of the power of the Phaser 3 cameras and multiple States running together. As I mentioned last issue states in Phaser 3 can run in parallel. Each State controls its own unique set of systems, such as tweens and input handling, and also manages its own display list. The display list for one State is not shared by another, allowing you to run completely independent things per State.

States has their own Cameras too. The camera has a default position, size, scale and rotation, but you can change it in real time. This allows you to do things like move a camera around the screen, or zoom it into a specific part of your game, and again this is all unique per State.

Click the image below to see the demo in action and let it run for a while so you can see all of the effects:



First of all, it's worth noting that each of the 4 scenes is a unique State, the code for which is stored in their own files as well. Demo A contains 2000 images being depth sorted in real time (as the mushrooms move around). Demo B shows the tint pixel particle demo from above. Demo C has 5 tinted Tile Sprites animating and finally, Demo D has 4 pieces of 3D line art being rendered by the Graphics renderer.

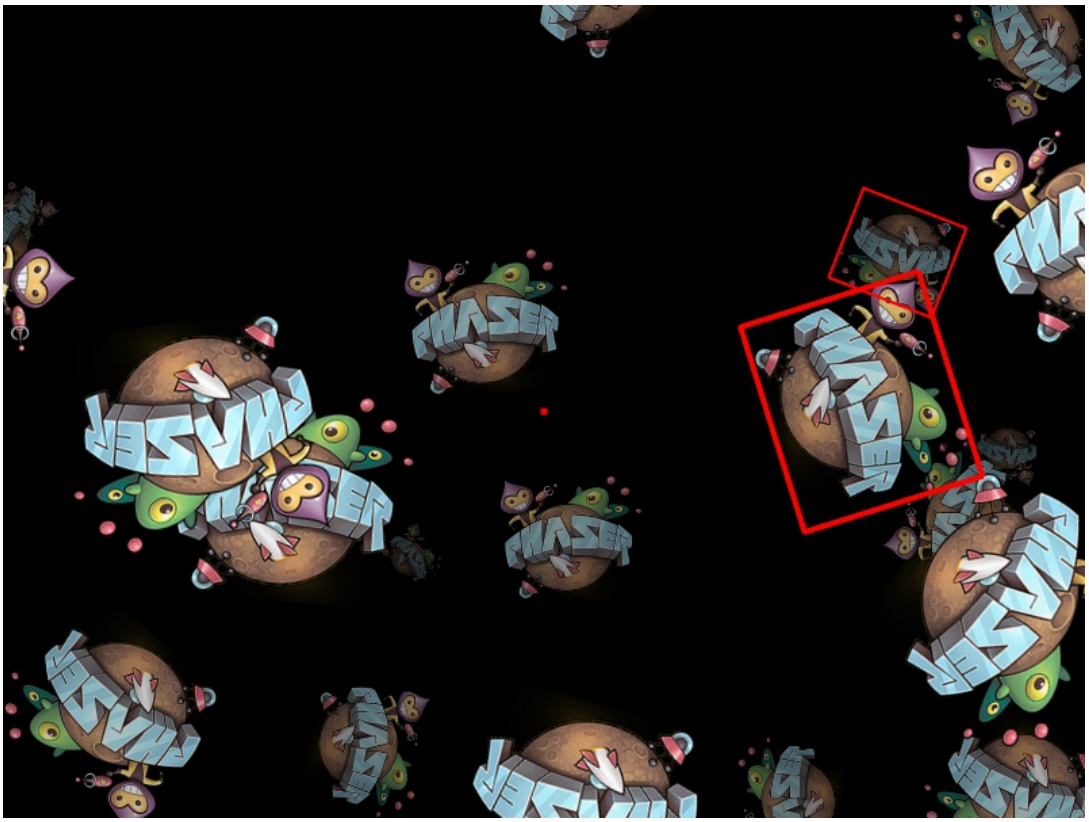
There is also a Controller State which doesn't render at all, it just grabs a reference to the 4 cameras from each of the demos and then is responsible for tweening them, so that they change position, scale and so on. All in all, it's a pretty intensive test, touching 3 different internal WebGL batches, tinting, lots of display list management and render states. I'm really excited that the cameras in v3 are powerful enough to do this without batting an eyelid. Some of the guys on Slack even got it running on their phones and tablets without any issue, which is quite something for a demo that was definitely only built with desktop in mind when I made it!

Camera Input Functions

As well as tints and cameras we've also been cracking on with the Input classes. Last week that took the form of Felipe coding a set of "screen to world" point conversion functions, allowing us to map mouse and pointer input coordinates down through the camera and transformed display list, getting a list of interacted

Game Objects back again.

Use the cursor keys to move the camera in the example below and the mouse to mouse-over the objects:



Felipe also implemented Camera culling, so a camera will no longer render a Game Object that isn't within its field of view. As v3 uses a single level display list this is fast to calculate, without needing deep recursion or iteration. Even so, we have made it optional - so you can disable camera culling if you don't require it in your game.

With the core input functions being nearly finished we can get the Mouse and Touch classes into v3 and working to full capacity. This is the final element required before we can start preparing the Alpha release. It's damned exciting and we can't wait to share it with you soon :)

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.

Winamp's woes: How the greatest MP3 player undid itself.

Check out the High Definition Earth Viewing (HDEV) experiment aboard the ISS for some beautiful HD shots of the earth.

What if NPM was a parcel delivery company? :)

Further Reading ...

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

Phaser Releases

The current version of Phaser CE is 2.8.1 released on June 20th 2017.

Phaser 3.0.0 is in active development in the GitHub v3 folder.

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