Welcome to Issue 91 of Phaser World

We hit a major milestone today with the release of the Phaser 3 Alpha. Check out the Dev Log below for more details. Not to be outdone, Phaser CE had a 2.8.3

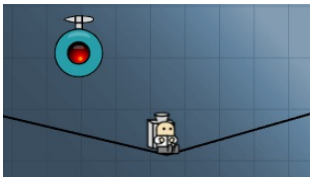
release as well :) On the games front, we've the sublime Wizard in a Bubble and the kid favorite Pets Factor Furry Friends. On the tutorial side there is an excellent nav mesh plugin and more updates to the Slither.io clone.

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

Pets Factor Furry Friends >>> **Game of the Week**
Choose a pet and make sure they are entertained, exercised and well rested in this beautifully animated game.

Wizard in a Bubble >>> **Staff Pick**
Can you blow the bubble to the portal in each level of this massively addictive game?

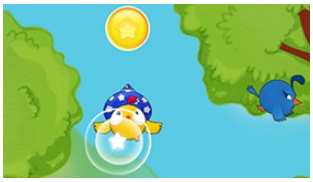


Blast Down!

Test the world's first jetpack while fighting off waves of rogue robots!

The Fearsome Mapinguari

Get rid of the loggers trying to destroy the rain forest.



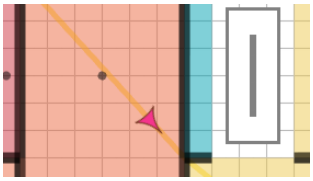
Chickie Can Fly

One day a little Chickie looked high in the sky and dreamed of flying.

What's New?

Phaser CE v2.8.3 Released

This new point release adds a new Ionic template and addresses some TypeScript bugs.



Nav Mesh Plugin

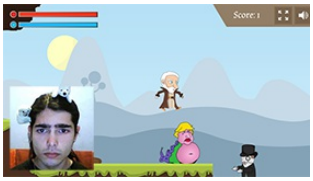
A Phaser plugin for fast pathfinding using navigation meshes.

Streamlining Standalone Game Development

A great tutorial on creating a streamlined workflow for building standalone games.

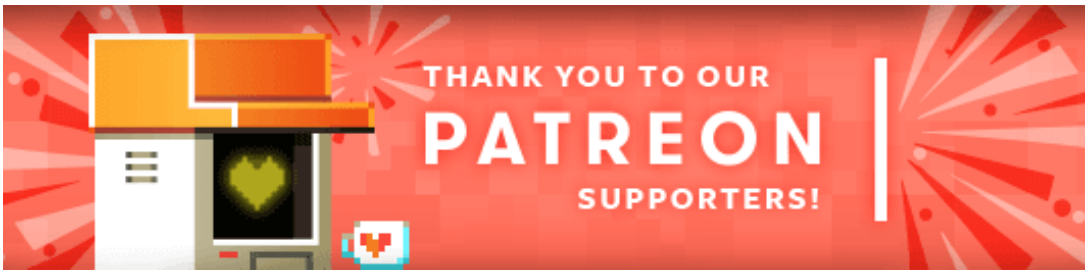
Orange Games Boilerplate

A comprehensive boilerplate used by one of the largest Phaser game publishers in the world.



Student Success with Salvatore Tedde

An interview with the developer of The General Election Game, Salvatore Tedde.



Welcome and a massive thank you to the new Phaser Patreons who joined us this week: **Bagus Aji Santoso** and a huge new patron: **CrossInstall**. We'll have more details about what benefits CrossInstall can offer to Phaser devs next issue.

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 #91

After a lot of work updating our build tool chain, I released Phaser 3.0.0 Alpha today. This is a huge milestone for us and compacts months and months of hard work into a few hundred kilobytes. I would love it if you could help us test it out. Here is how to get involved:

Getting Started with the Alpha

There are several ways to get hold of the V3 alpha:

GitHub Release

Go to the GitHub Alpha Release page and download the build files. There is a phaser.js file and phaser.min.js. The phaser.js is a development build and includes a source map *inside* of it (hence the massive file size!), which is perfect for giving you accurate runtime errors. The min file has the source map and comments stripped out and is more like what a production build of Phaser 3

would be.

Download from NPM

The Phaser 3 Alpha has been published to npm. You can grab it using the following tag:

npm install phaser@3.0.0-alpha.2

Build files are not committed to the repo or in npm, so you'll have to run an npm install and then 'npm run build' command.

Webpack Project Template

We have also prepared a sample project template for you. This uses Webpack to build and is a quick and easy way to get started. Use the following to get the template:

npm install phaser3-project-template

Once it has downloaded go into the folder and issue the command: **npm run build** and it will build the sample project. Open the index.html file into a browser to see the end result. You can modify the source in the src/index.js file to play around with it, or just use this template as a basis for your own. We will expand it over time to add in features like Live Reload.

Build from the source

The final way to get the Alpha is to download the Phaser repo and build it locally. This gives you the most flexibility and allows you to view the Phaser source easily but takes a little more effort.

Full instructions are given on this page.

As an update to those instructions you can perform a build of the dev version now by issuing the command **npm run build** or you can build a distribution version with the command **npm run dist** which will create minified compressed builds.

Alpha Feedback

We would really like to know what you think of v3. I appreciate that because we haven't started the documentation yet it's going to be hard going for you, but there are hundreds of examples to inspect and learn from and things aren't *that*

different to v2.

We need both bug reports and your feedback on the API. Of course, we want to squish as many bugs as possible, so bug reports are really important for us. But we also want to get your feelings on the API changes too, so if you think a method name doesn't make sense, or you have a suggestion for a feature or property then please let us know! There are lots of ways:

GitHub Issues

This is our preferred method of receiving bug reports and suggestions. Please open a new Issue on GitHub and throw in as much detail as you can. Source code snippets are even better.

Phaser 3 Forum

There is a new Phaser 3 Forum running on HTML5 Game Devs. You can attach files or embed source code and it's a good place to get discussion from other devs too. We are monitoring this forum daily.

Email

Old school, but it works. You can email rdavey@gmail.com. Please understand I cannot enter into long support style discussions via email (there just aren't enough hours in the day!) but I'm happy to take v3 feedback here.

Phaser Slack

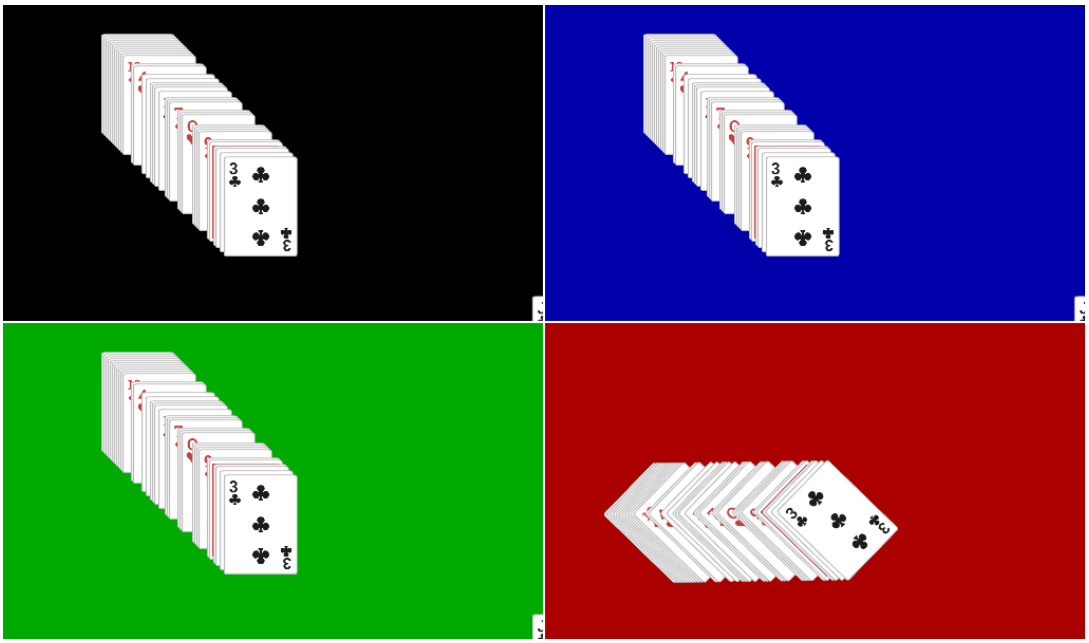
You can also talk to me on the Phaser Slack channel. Just use @rich and I'll get the message when I'm not online (if I'm not there at the time).

Whichever method you use, please *do* use one! All comments are useful, even if it's just to let us know that something works, that's good too. Bugs are best suited to GitHub Issues as you will then know when we fix them because the issue will be linked to a commit but if you'd rather not register on there we'll happily take them via any other means listed.

Input Manager Updates

I spent last week working on the Input Manager and made some fantastic progress with it. It's now very nearly feature complete and even has the start of the mobile touch support added as well. The following demos show some of the new ways to use it - remember you can click the 'Edit' button on any demo to view the source code. All of these demos are running with the v3 Alpha release.

First here is a quick demo showing pointer events working across multiple cameras, even if they're rotated and scaled:



Click a card, any card.

Input Zones

The next example shows off a new feature in v3, which is the ability to make zones input enabled:

Mmm!

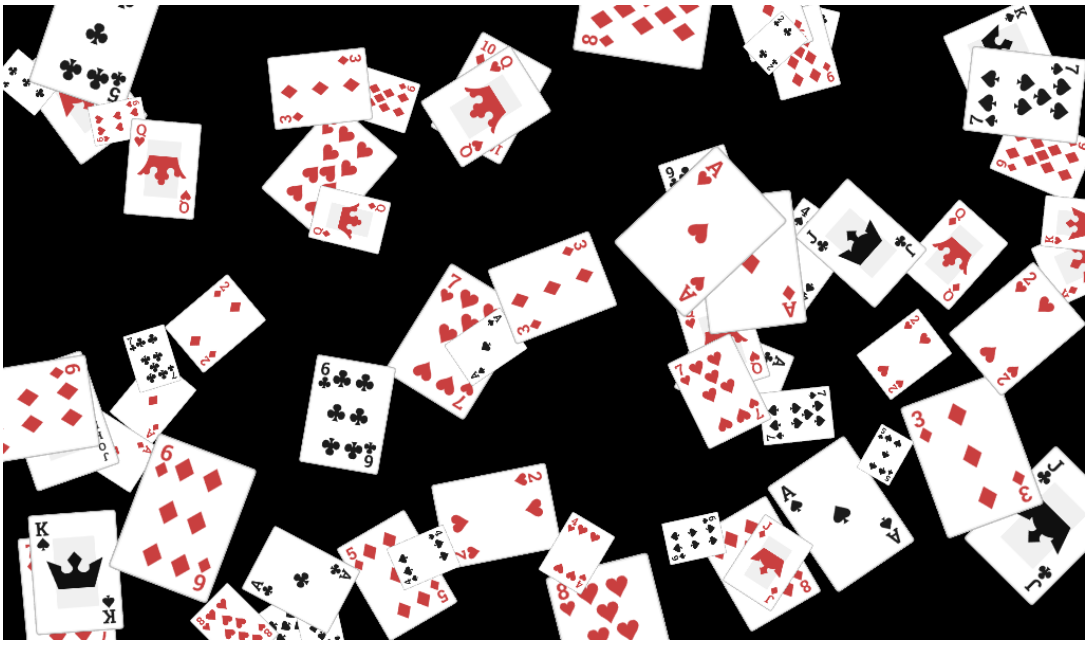
The above example (which makes me hungry each time I load it!) shows how you can use a Zone as an Interactive Object. Zones are game objects that have

no texture or display properties. They have a position and size and can be transformed, but they never render or have any related components. This makes them extremely tiny and also useful for hit areas. This example demonstrates adding 6 zones over the top of a single image and it is the zones that dispatch the input events, not the image. This could be extremely useful if you wanted to add interactive areas over the top of a video or an image that constantly changes without having to use 'blank' sprites to do it. The following code shows the creation of the zones and the related event handler:

```
//  One zone for each donut in the picture

this.add.zone(0, 0, 345, 300).setName('Plain').setInteractive();
this.add.zone(345, 0, 310, 300).setName('Chocolate').setInteractive();
this.add.zone(655, 0, 369, 300).setName('Coffee\nand cream').setInteractive();

this.add.zone(0, 300, 330, 300).setName('Chocolate\nSprinkles').setInteractive();
this.add.zone(330, 300, 350, 300).setName('Strawberry').setInteractive();
this.add.zone(680, 300, 344, 300).setName('More\nSprinkles').setInteractive();

//  The event handler

this.input.events.on('GAME_OBJECT_DOWN_EVENT', function (event) {

    fork.x = event.pointer.x;
    fork.y = event.pointer.y;

    label.setText(event.gameObject.name);
    label.x = event.gameObject.x;
    label.y = event.gameObject.y;

});
```

Drag and Drop

Any Game Object can be enabled for dragging by calling the **setDraggable** method of the Input Manager. It can accept a single object or an array of objects and they'll all be flagged as being draggable. Here you can see a bunch of draggable cards, each rotating and a different scale, but all draggable:



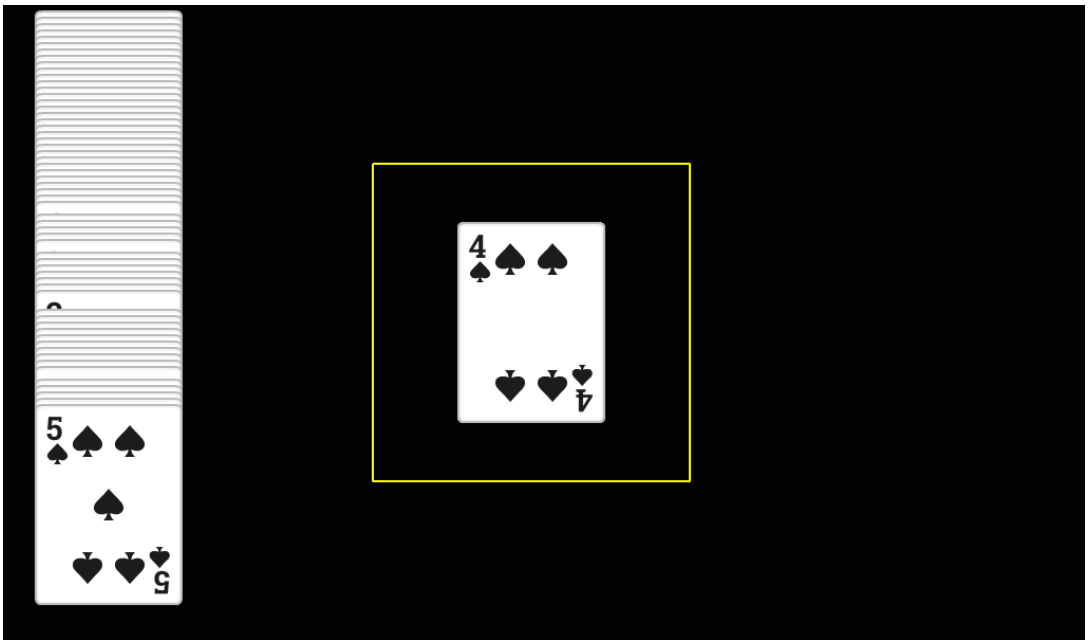
Drag the cards. Arrows to move the camera.

In the above example, you can use the arrow keys to move the camera as well, showing that drag input works even across scroll factors and z-depths.

Objects set to be draggable emit their own special events, such as DRAG_START, DRAG and DRAG_END. Also available are the options to set 'drag time' and 'drag distance' thresholds. These are values (given in ms and pixels respectively) that once exceeded a drag is considered as being invoked. For example, you could say that a sprite isn't being dragged unless it has been moved at least 10 pixels, or held down for at least 2 seconds.

Drop Zones

Another new feature is drop zones. *Any* interactive object can become a drop zone and they have a whole bunch of associated events all modelled on the Drag and Drop DOM API. This example shows a rectangular drop zone:



Drop zone

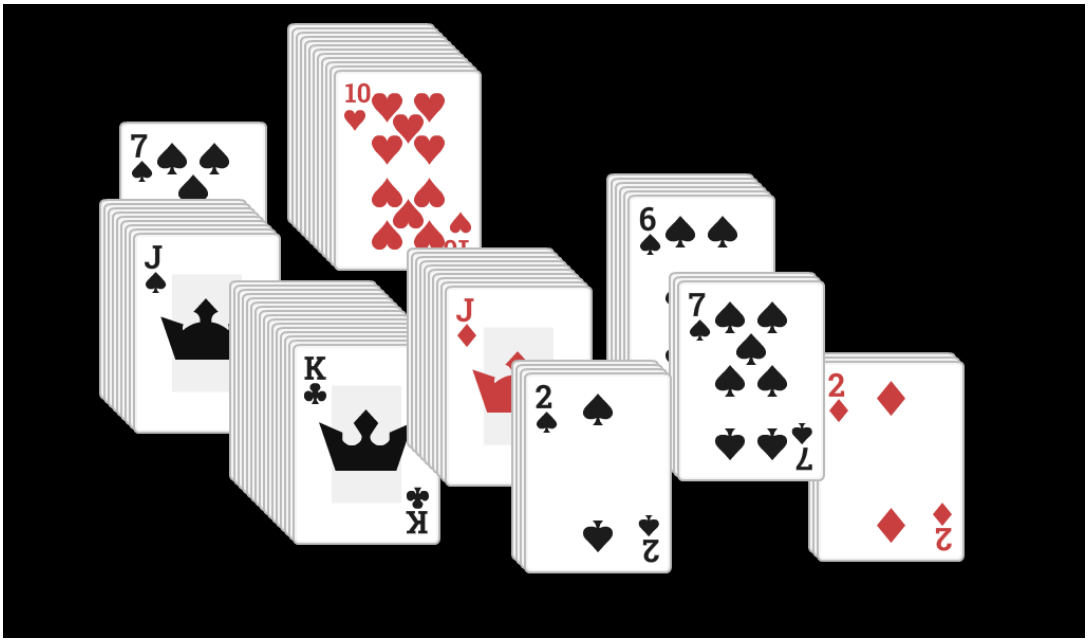
Drag the cards into the drop zone. As long as the pointer is within the zone the card will remain there when dropped, otherwise, it is returned to its start position. Drop zones don't have to be rectangles either. Any shape that a hit area can use is valid.

If you drag a game object across one that has been flagged as being a drop zone then you get lots of useful events such as 'DRAG_ENTER', 'DRAG_LEAVE' and the essential 'DROP' event. What you do inside of these events is entirely up to you of course but it gives you a lot of flexibility out of the box.

Top Only

Another feature that landed is the ability to control what the Input Manager will return. If you had a stack of sprites on-top of each other and clicked the top-most one, then by default events will be dispatched for the top sprite only. This is exactly how the DOM works and we emulate it inside of Phaser too.

However, it's often useful for you to be able to get all objects below the pointer, not just the top-most one. So you can now set the **topOnly** boolean to 'false' and it'll work across all interactive objects below the pointer, from the top down. You can observe the difference in this example:



topOnly

As you can see above, you can drag all cards below the pointer at once. If you're very careful and click to the edges of a stack of cards you can peel them away one by one :)

It's really handy feature though, allowing you to have a lot more control over your interactive objects than you can in the DOM.

Per-Vertex Alpha Support

The last feature to mention this issue is that you can now set the alpha of each vertex of a game object. Just like you could tint each vertex you can now control the alpha directly via the properties: alphaTopLeft, alphaTopRight, alphaBottomLeft and alphaBottomRight. This allows you to make some really lovely transition effects without using any custom shaders or masks, such as this:

Fade to black

The above example is just two images with a tween controlling the alpha of one. Naturally per-vertex alpha is a WebGL only feature. The original 'alpha' property still exists and works in exactly the same way as before too.

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.

The Stranger Things season 2 trailer, released at Comic Con, looks so insanely good! I cannot wait :)

Low Res Jam starts tomorrow and runs until August 17th if you fancy making a game in 64x64 pixels or less.

Adobe announces Flash distribution and updates to end. We all knew it was coming but it's still a shame.

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.