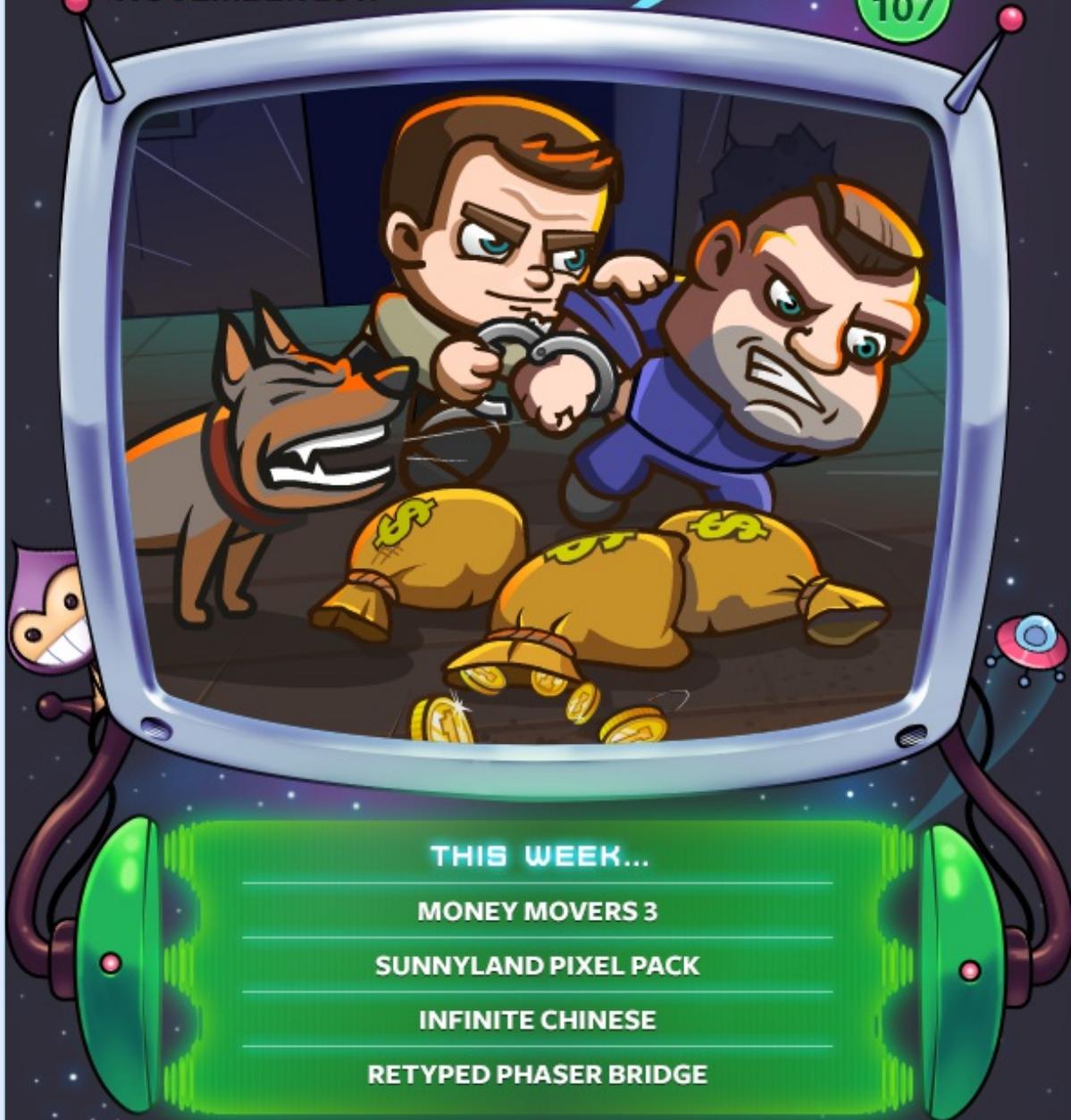


PHASER WORLD

NOVEMBER 2017

ISSUE
107



THIS WEEK...

MONEY MOVERS 3

SUNNYLAND PIXEL PACK

INFINITE CHINESE

RETYPED PHASER BRIDGE

Welcome to Issue 107 of Phaser World

For those of you States-side, I trust you had a happy Thanksgiving. You're probably side and fed-up of Black Friday deals by now, so you'll be pleased to

hear I'm not going to mention a single one in this issue :) However, we do have some awesome games to play, a new huge multi-part tutorial on using Game Analytics and an absolutely rammed Dev Log!

Also, you'll notice I'm no longer linking to the Phaser Discord channel here or on the site. Honestly, while Slack has upwards of nearly 1300 users, Discord was always a bit of a ghost town. Mostly because I didn't have time to hang out there, but also because it's just a generally quieter channel. I'm currently deciding what to do with it and will most likely hand control over to someone else, so it'll become an "Un-official Phaser Discord", but still exist. More info soon.

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

[Money Movers 3](#)

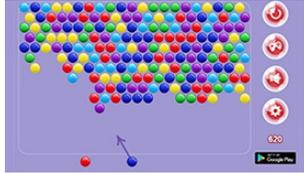
Team up with your guard dog and arrest the robbers in this great action puzzle game.



Staff Pick

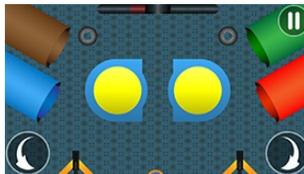
[Infinite Chinese](#)

Learn Chinese by playing fun and interactive games in space! No multiple choice, flashcards, or any other boring stuff!



Bubble Shooter Classic

It's the classic game everyone loves, taken right back to the basics. And is all the better for it.



Baked Donuts

Sort-out all color-matching donuts in this puzzle game.

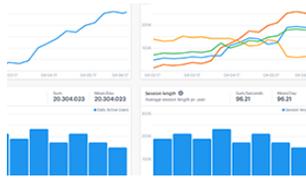


Crazy Presents

A new Christmas themed shuffle game template, easy to customise and fully mobile and social ready.



What's New?



GameAnalytics for Phaser

Want to see everything that happens in your game? Use GameAnalytics in your Phaser games with this tutorial.



Sunnyland Forest Pack

Another great pack of pixel art for use in your games, with a free Phaser example and demo.



Retyped Phaser Bridge

Code Phaser games in C# using Retyped and Bridge.NET.



Phaser Editor Eclipse Feature

A special article about Phaser Editor, as featured in the Eclipse newsletter.



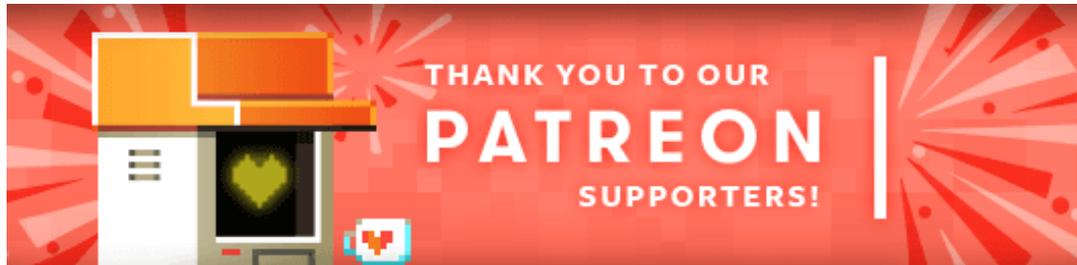
Using Google Fonts in Phaser

Integrate Google fonts into your Phaser games without file bloat.



PlayFab SDK now supports Phaser

Get access to leaderboards, analytics, a/b testing and more from your games with native Phaser support direct from PlayFab.



No new [Phaser Patrons](#) this week, but thanks to Black Friday / Cyber Monday the shop and affiliate sales have been great, which will keep us going for a few weeks.

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



The [Endless Swimmer asset pack](#) is perfect for creating a 2D side-scrolling “endless swimmer” style game (like an endless runner, only it’s underwater!) It contains backgrounds, character sprites, a HUD and two example UI screens.



Dev Log #107

Another week and another huge load of work completed. You can try all new features shown in this Dev Log in [Phaser 3 Beta 12](#). Grab it from GitHub pre-built, or from npm using the beta tag.

Sound API Update

Pavle reports on progress to the Sound API: Although I said that adding support for looping and seeking is next on the list I decided to work on making basic playback options more solid before moving on to more advanced things, so I added support for sound markers first.

A sound marker is part of sound represented by a unique name and a start time and duration. This allows you to bundle multiple sounds together into a single audio file and use markers to jump between them for playback.

Addition in v3 compared to v2 is that marker can contain a config property, the same as the one that you would pass to sound play or add methods, so that each marker can play at different playback rate, at different volume level etc.

I've also exposed two additional properties on sound objects: duration and totalDuration.

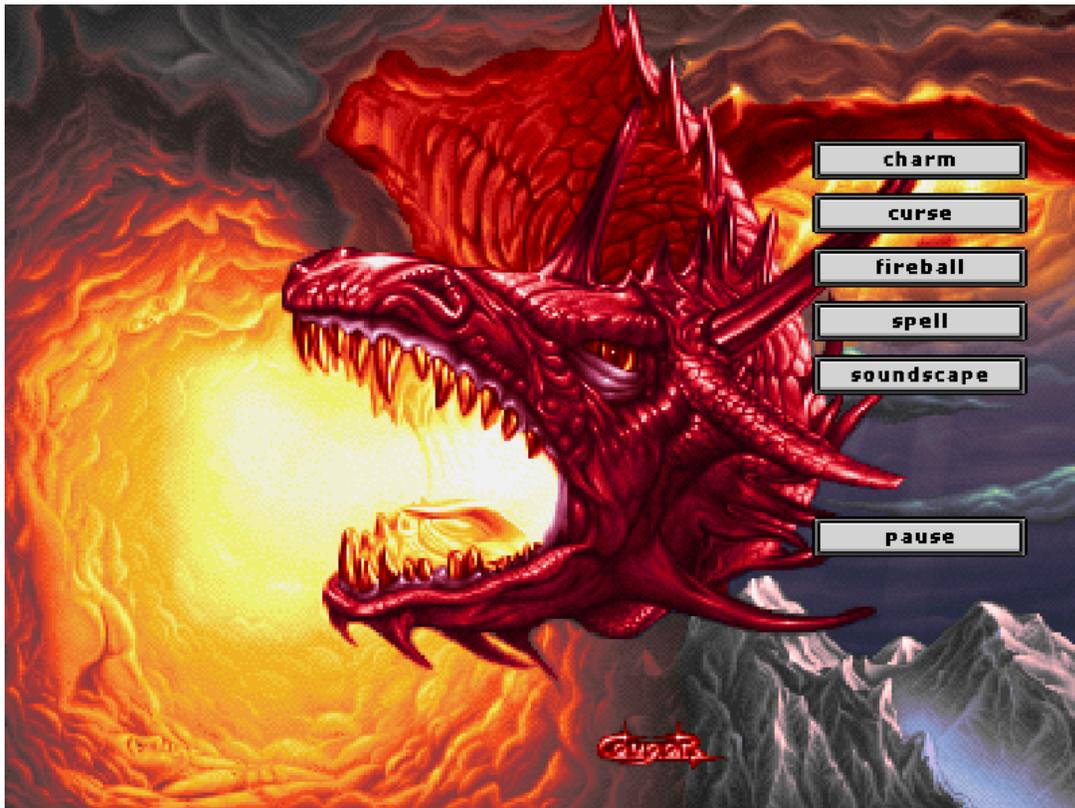
Duration holds the value of currently playing marker duration or duration of the entire sound if no marker is being played, while totalDuration represents the duration of the entire sound - obviously. :D

Without this, you wouldn't be able to make a music player in Phaser, which I mentioned in the previous issue, as you need to know the duration of a sound in order to display its value and position correctly playback slider cursor.

I've recreated two examples from v2 to demonstrate how to use markers in v3, so if you feel a bit nostalgic go check them out and play around with different config values!



Sound Markers working



Pause and Resume markers

Tilemap API Update

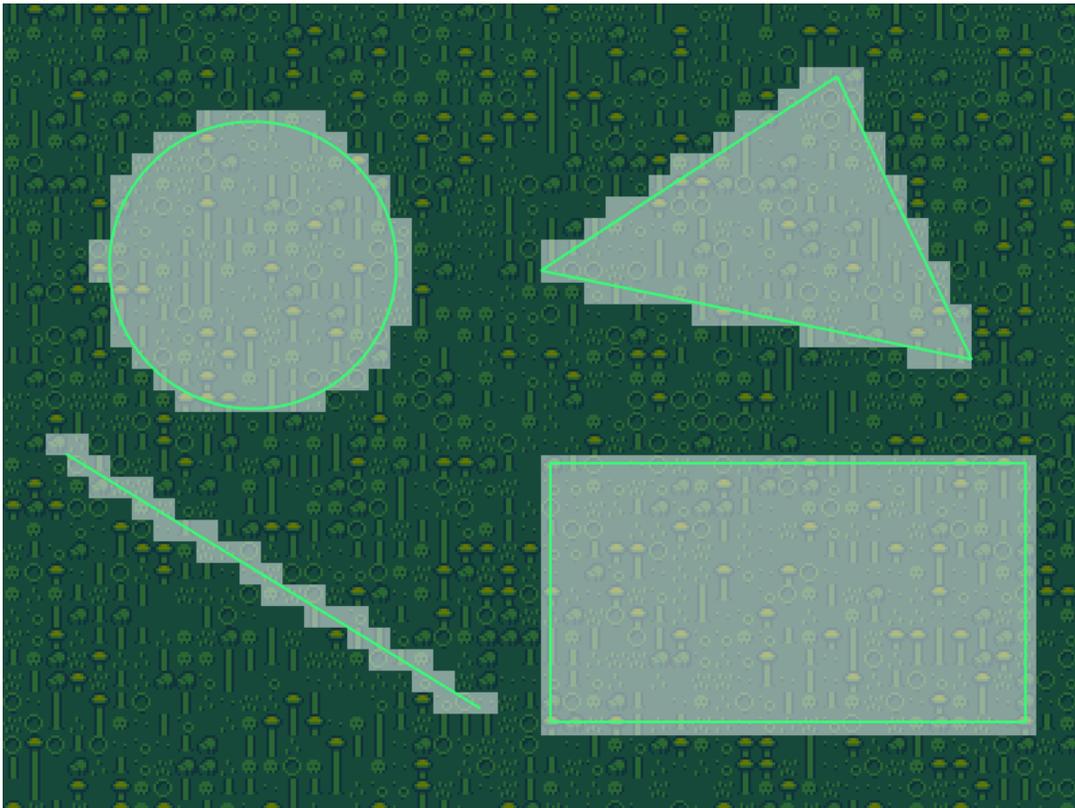
Mike reports on the updates to the Tilemap API: "It was a short week here in the states because of Thanksgiving, but the Tilemap API still marches forward:"

The Tiled parser has been modularized and updated. It can read in layer offsets and properly apply them to objects within layers. It can also parse Tiled's text objects, so you could potentially visually lay out a HUD in Tiled and generate it in Phaser.

Tiles can be set to collide, so everything is in place for physics integration. (A small technical note - recalculating interesting faces is magnitudes faster in a few important cases, so you probably won't often need to think about preventing recalculations like in v2.)

The documentation effort is now well underway.

And last but not least, v3 now has the ability to get tiles that overlap with a geometry shape. You can see this in the screen shot below (it's not a demo yet, so don't try and click it).

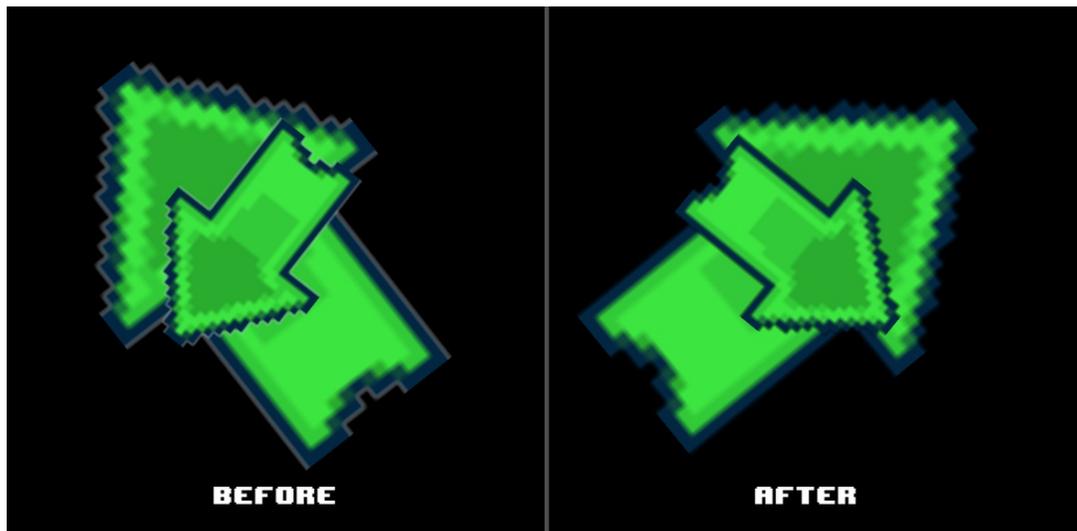


Pre-Multiplied Alpha

Felipe writes: "The past week was mostly fixing already present issues on the WebGL rendering backend. Some really small that only affected specific parts and others that affect large chunks of the code.

The most important issue I fixed was getting the correct colors to appear when you scale and rotate textures by using pre-multiplied alpha. Read this [NVIDIA article](#) for more details about how it all works.

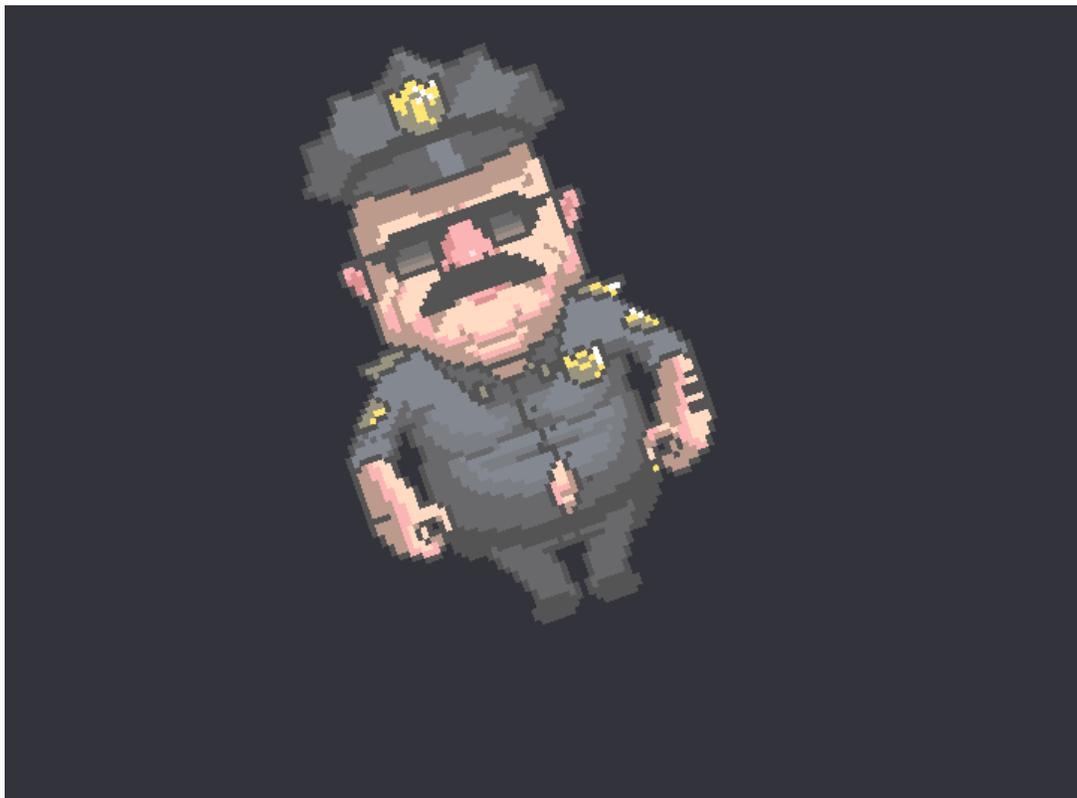
So far the progress looks good:



In the image above you can clearly see the difference between the scaled and rotated textures before and after this work.

Another effect I worked on was simple but very useful - allowing you to set a fill color for a sprite. This would allow for sprites to have an overlay color that only affects the RGB channel, using the Alpha channel as a control for mixing the texture and tint with the fill color.

This would allow for effects the animation below by only modifying a single property on the Game Object:



There are some issues left to fix with regard to masking (the new alpha work has broken masking), so as soon as that's sorted I'll merge the work into master. It means you won't see the above in Beta 12, but keep an eye on the repo to see it land this week.

Matter.js Physics

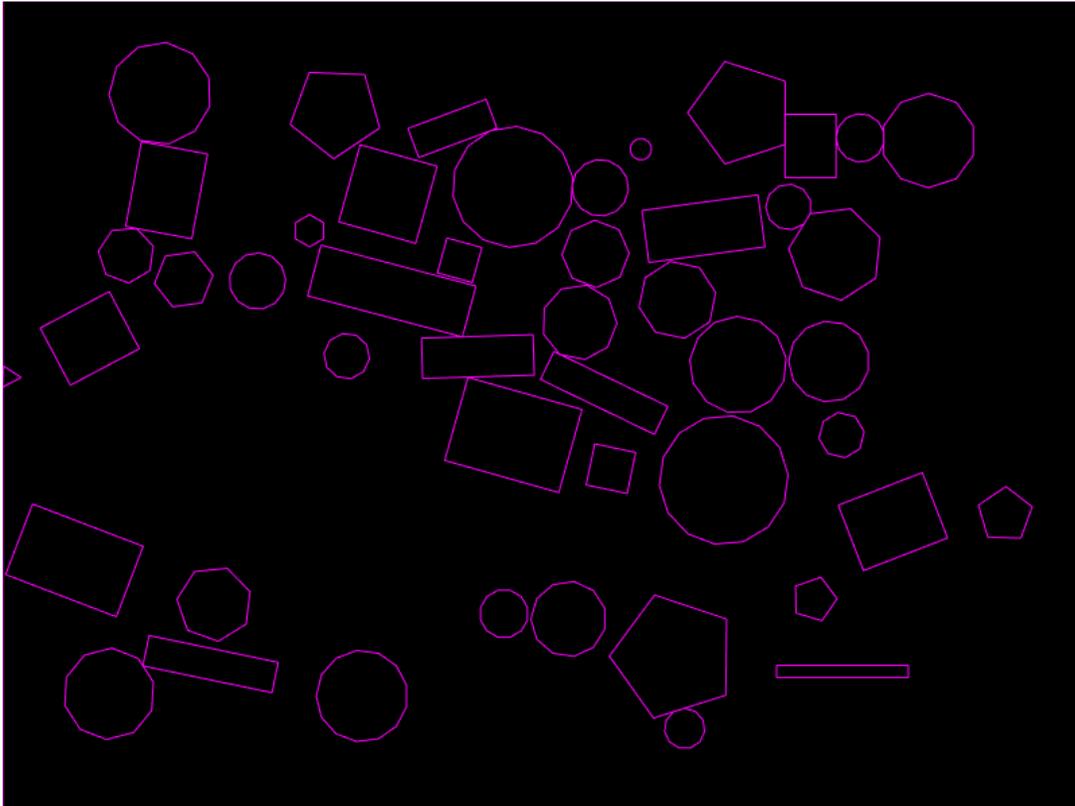
I spent last week working hard on getting [Matter.js](#) integrated with V3. It's one of the final areas of the API and brings us a massive step closer to feature completion.

For those who have not come across Matter before, it's a rigid body physics system with a host of powerful features: Compound and Convex bodies, concave and convex hulls, physical properties (mass, area density), restitution, collision filters, stacking, resting and a whole load of other really nice things. It replaces the use of P2 Physics from Phaser 2 and is our new default full-body physics engine. There are several reasons for picking it over P2. The first, and most important, is that it's still actively developed. The main developer is available and responsive, which is great! It also has some superb features missing from P2, such as the ability to translate, scale or rotate a body without modifying its velocity at all. Perfect if you want to tween a physics body and still have it respond accurately. And finally, and it's a small one but still great: everything is done in pixels. There is no meters to pixels conversion needed anywhere, which makes your code simpler, the API code simpler and much easier to debug as well.

Integration was pretty smooth and I've done my best to make it as friendly as possible, following the same conventions as the rest of the V3 API. I still expose the whole of the Matter API to you for when you just want to get in there and play with it directly. There are also lots of helper methods and classes, which you'll see some of in a moment, but just know that they're optional and you can hit the metal directly if you wish and roll your own.

I'm going to cover the use of Matter in more detail next issue. For now feel free to play with the API (as it's all present in Beta 12), look at the stacks of examples in the labs and check out these demos. As usual, click the screenshots to run the examples in browser.

Mixed Bodies

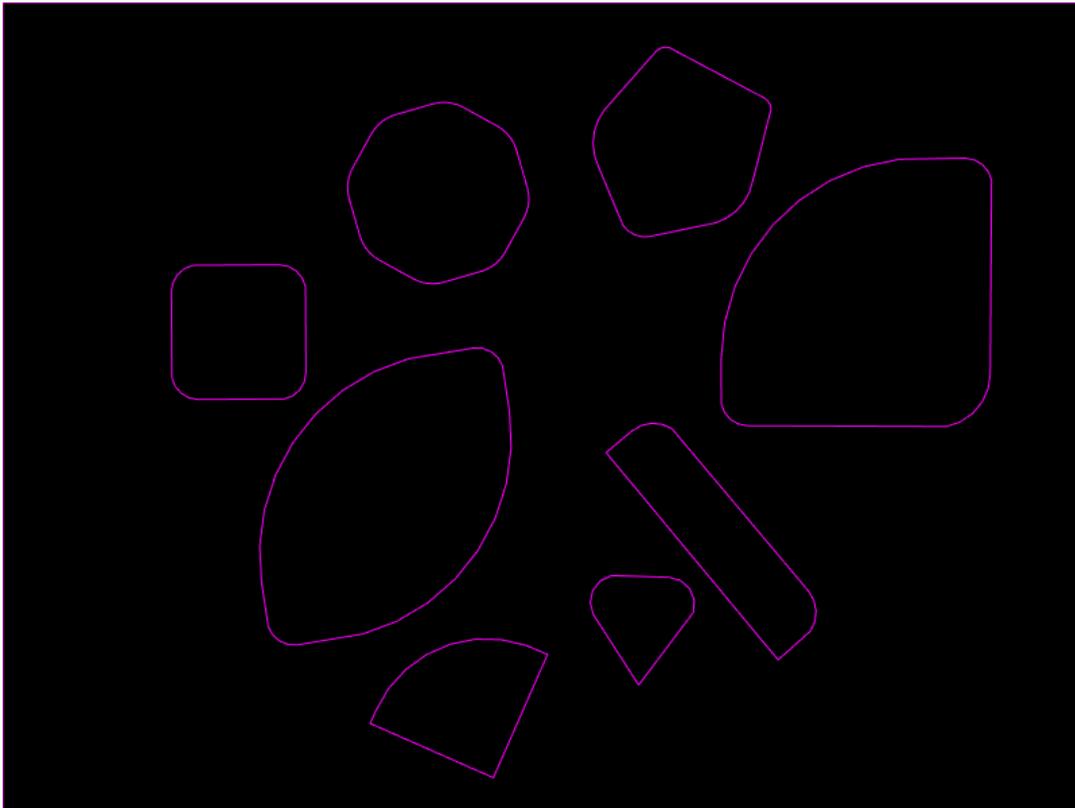


Use the mouse to drag the bodies around

Creating bodies in Matter is really simple. Using the Factory API you can call upon methods like **Add.rectangle** or **Add.polygon** and provide the required parameters. For a polygon you specify a radius in pixels and then the number of sides it should have. Ask for 3 sides and you'll get a triangle but it can increase from there. The more sides you add the more like a circle it becomes.

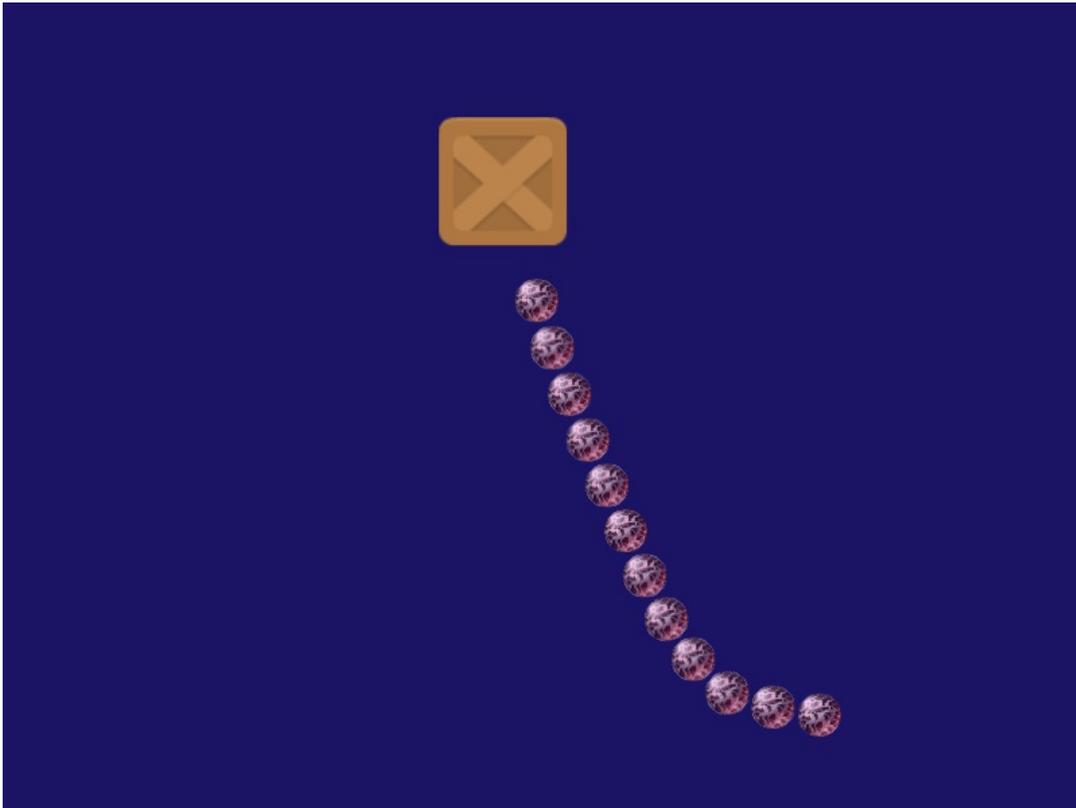
There are 5 built-in body types: Circle, Rectangle, Polygon, Trapezoid and one called FromVertices, which takes a series of points and if needed decomposes it into concave hulls and gives you a single body as the end result.

Rounded Bodies (Chamfering)



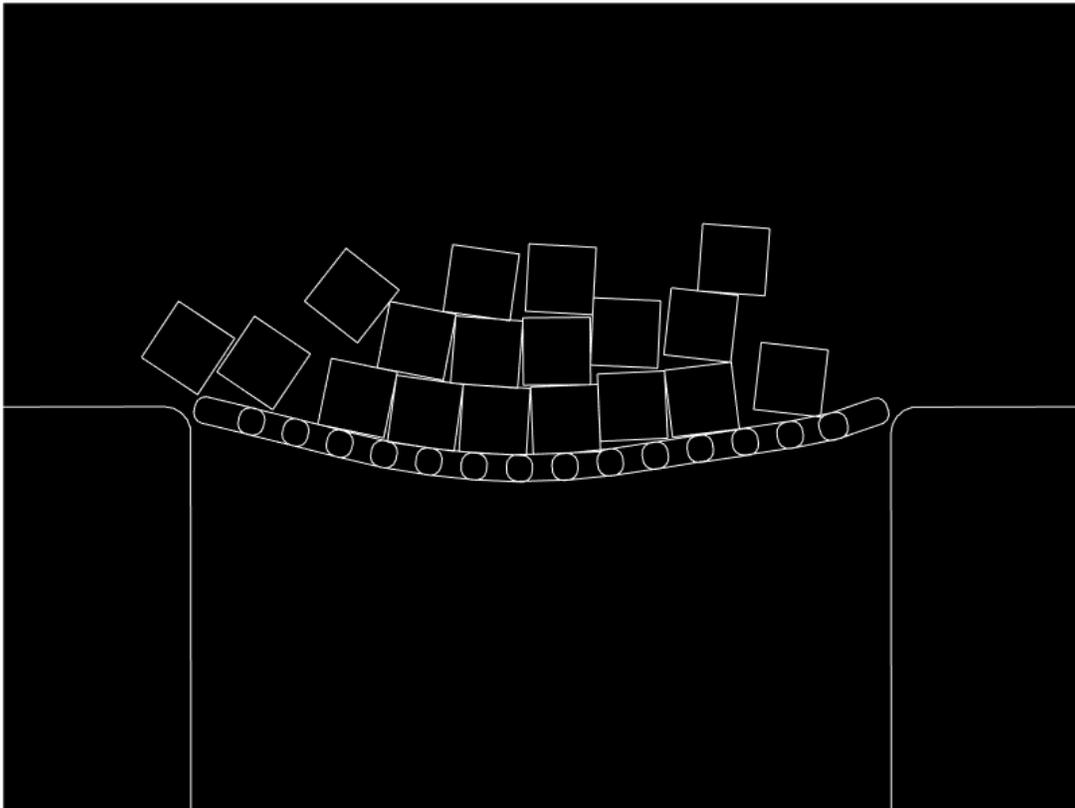
Bodies also have an optional **chamfer** property during creation. It allows you to set the number of points used for the edges of the body. In effect, you can create smooth edged bodies really quickly, such as in the example above. You can control the number of points used for the whole shape, or per corner.

Chains



Cursors to move and mouse drag

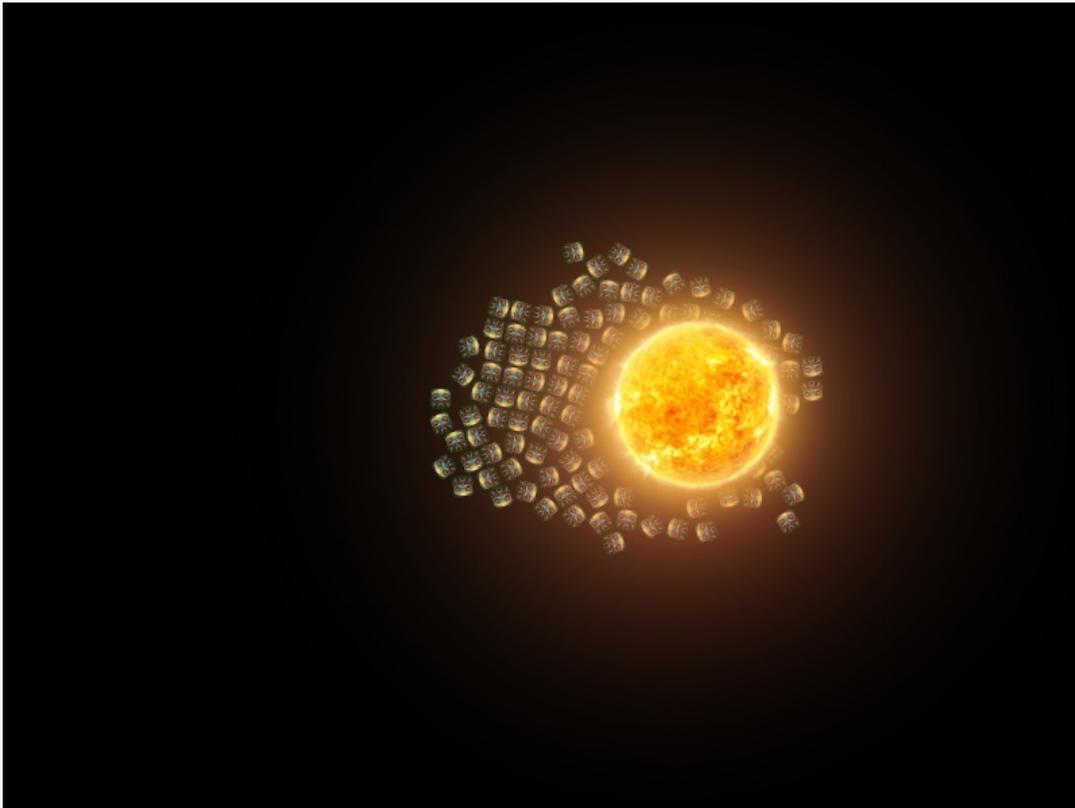
The Matter Factory has built-in support for creating a chain between a group of bodies. It works by creating a constraint tween each body. You can control the properties of the constraint, such as the resting length and stiffness, allowing for some complex behaviors. The example above also shows the use of sprites with a bound body. When you start combining constraints you automatically open-up the ability to create springy bridges like this:



Drag the bridge!

Attractors and World Wrap

Matter has a small but nice plugin ecosystem going and we've taken a couple of them and merged them into our default build. The Attractors plugin allows for the creation of continual forces, such as wind, gravity or magnetism. You can see it in action in the demo below:



Attractive!

The World Wrap plugin allows you to wrap a body around the bounds of a world should it stray off the edge. Think of the classic game Asteroids, and how the ship and rocks appeared on the other side of the screen? It basically allows you to do this, without accidentally teleporting your body right through something else. Both of these available by default and you can enable them with a single line in your game config or game code.

Simple API

As you'd expect from us by now, the API is dead simple to use. Just specify that you want to use Matter as your physics engine in the game config and then you can call the Factory functions directly. Have a look at the following code:

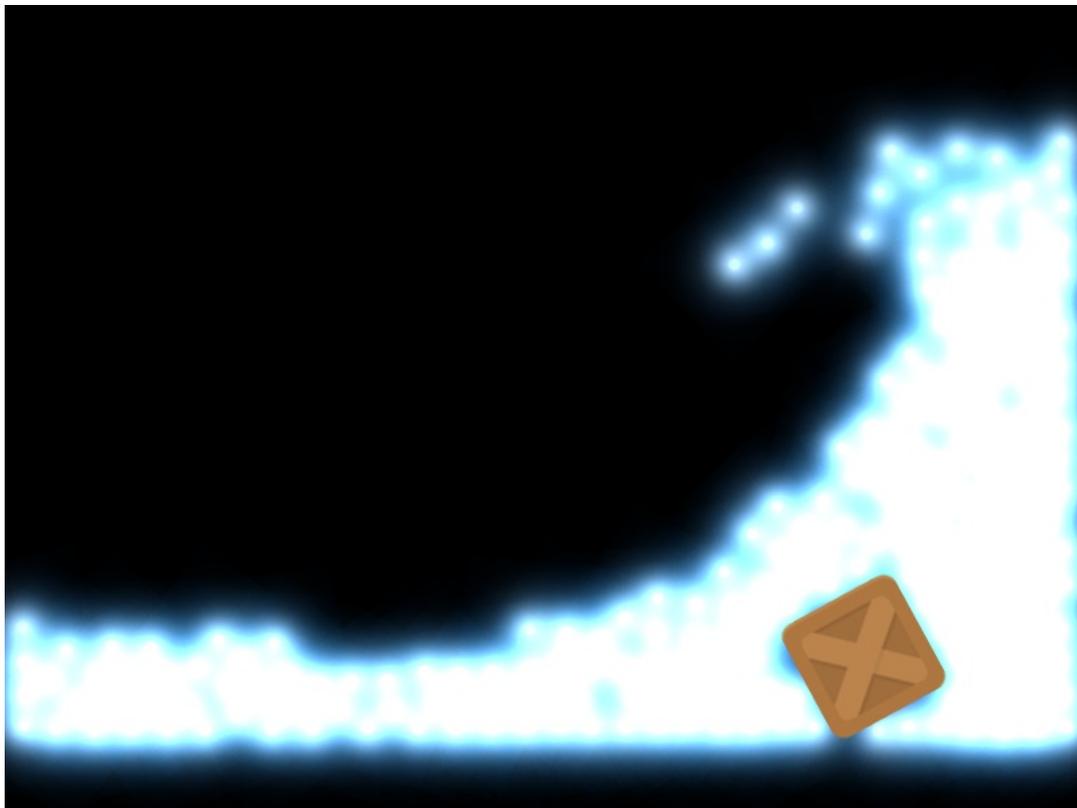
```
function create ()
{
  this.physics.add.image(325, -100, 'block');
  this.physics.add.image(400, 300, 'block');
  this.physics.add.image(450, 50, 'block');

  this.physics.add.image(400, 550, 'platform', null, { isStatic: true });
}
```

This code, makes this ...

Literally, one line of code to create a full rigid physics body bound to an image. It will automatically size the body to match the image texture, assigning it the correct mass and density. You can pass in additional properties as well, such as giving it your own mass, or telling it to use a different body shape, or a static body, or any number of other options.

There are lots of helper methods on our Matter classes, such as `setBodyShape`, `setVelocity`, `thrustLeft` and `thrustRight`, `setBounce` and helpers to deal with collision groups and collision filtering. We wanted it to be as easy to use as possible. For it to be really quick to get something up and running, but flexible enough that you can really go to town creating physics games with it, or just fun demos like this:



Drag that crate!

I'm nearly done with integration of the API, there are a few elements left to test but it's 90% there already. I'll show some more demos next week, after which we will be moving on to the new Phaser 3 Scale Manager. It's the final system to build for V3. There are more features to complete, such as Pointer Lock support and Tilemap collision, but the Scale Manager is the final system to actually develop. With that in place it will give us the rest of the year to hammer through tests, docs and examples, ready for the New Year.

Phaser 3 Labs

[Phaser 3 Beta 12](#) 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!



This WebGL experiment is just fantastic - pull the [elastic face](#) around and enjoy the squishy sound effects!

[Games on Demand](#) is a wonderful idea - a game jam all about creating a game with a concept submitted by a child who is currently stuck in a hospital.

A very sobering post-mortem: [I thought I could ship at least 700 units to stay in business](#). Game dev is hard you guys!

Phaser Releases

Phaser CE 2.9.2 released November 10th 2017.
Phaser 3 Beta 12 released November 27th 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