

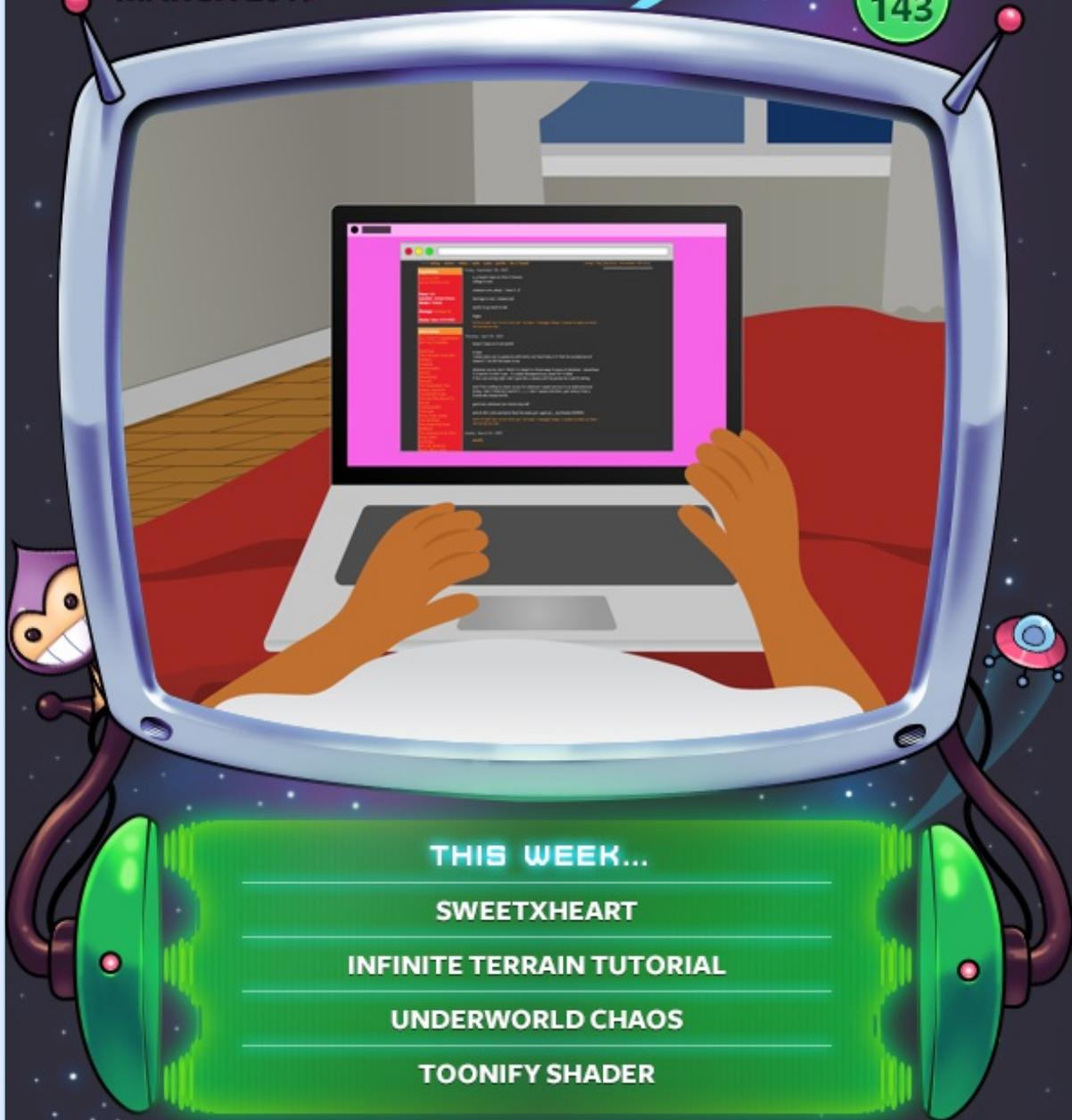
Web Version

Unsubscribe

PHASER WORLD

MARCH 2019

ISSUE
143



THIS WEEK...

SWEETXHEART

INFINITE TERRAIN TUTORIAL

UNDERWORLD CHAOS

TOONIFY SHADER

Welcome to Issue 143 of Phaser World

The games this week represent a wide gamut of genres for sure. There is a 16-bit styled pixel art game, a modern-day styled slots game, a casual puzzler and then there is SweetXheart, which is a highly personal visual novel from developer Catt Small. It took her several years to complete the project and I'm glad she did! It's an affecting tale and a wonderful example of the diversity of games created in Phaser.

As usual, be sure to [read this newsletter on the web](#) so you don't miss anything.

Got a game or tutorial you'd like featured? Simply reply to this email, or message me on [Slack](#), [Discord](#) or [Twitter](#). Until the next issue, keep on coding!



The Latest Games



Game of the Week

[SweetXheart](#)

Spread over five fictional days, can you get through a week in the life of a modern black woman?



Staff Pick

[Underworld Chaos](#)

Take control of a wizard, collect the orbs and escape in this cute 16-bit pixel-art styled platform-shooter.



Mow it! Lawn Puzzle

Can you mow the lawn without crossing over your electric cable in this fun and challenging puzzle game.



Wild West Slot Machine

A nice cowboy themed slots game with multiple paylines and bonuses.



Color Hit

Can you hit the right colors quickly enough?



What's New?



Top-down Infinite Terrain Generation Tutorial

A comprehensive guide on using Perlin Noise to generate an infinite terrain that

loads in chunks around the player.



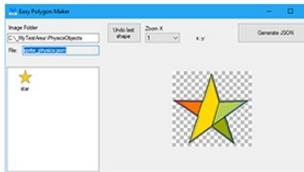
Toonify Shader

A Phaser 3 shader that applies a toon shading effect to a Scene, with edge and color level inputs.



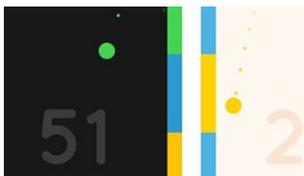
Color Jump Prototype Tutorial

Build a HTML5 game like “Color Jump” using Phaser 3 and Matter physics, introducing some ES6 features.



Easy Polygon Maker

Trace around images, exporting the outlines as P2 Physics JSON data for custom body shapes.



Color Jump Tutorial Part 2

In part 2 of the tutorial creating a game like Color Jump coins and comments are added into the source code.



Phaser 3 Game Development Course

A complete Phaser 3 and JavaScript Game Development package. 9 courses, 119 lessons and over 15 hours of video content. Learn to code and create a huge portfolio of cross platform games.

Help support Phaser

Because Phaser is an open source project, we cannot charge for it in the same way as traditional retail software. What's more, we don't ever want to. After all, it's built on, and was born from, open web standards. The core framework will always be free, even if you use it commercially.

You may not realize it, but because of this, we rely 100% on community backing to fund development.

Your support helps secure a constant cycle of updates, fixes, new features and planning for the future. There are other benefits to [backing Phaser](#), too:

PHASER

SUPPORT PHASER & UNLOCK:

PERKS

- 
BACHERS NEWSLETTER
A NEW REGULAR EDITORIAL STYLE NEWSLETTER. FULL OF INTERESTING AND RELEVANT GAME DEV LINKS
- 
EXAMPLES ARCHIVE
A BACHERS-ONLY FILE ARCHIVE OF PHASER EXAMPLES, GAMES, SNIPPETS AND RESOURCES (COMING DEC. 2019)
- 
PRIVATE DISCORD CHANNEL
YOU'LL GAIN ACCESS TO A BACHERS-ONLY DISCORD CHANNEL, WHERE YOU CAN CHAT WITH FELLOW DEVS
- 
ONE-ON-ONE TECHNICAL SUPPORT
CHAT DIRECTLY WITH THE DEVELOPER OF PHASER! GET BUGS FIXED OR CODE EXPLAINED (\$10+ BACHERS)
- 
EARLY ACCESS
GET EARLY ACCESS TO NEW TUTORIALS, BOOKS, PLUGINS AND VIDEOS. THEN GET A DISCOUNT ON RELEASE
- 
FORUM BADGE
SHOW OFF YOUR BACHER STATUS WITH A SPECIAL BADGE ON THE HTML5 GAME DEVS FORUM

MORE PERKS

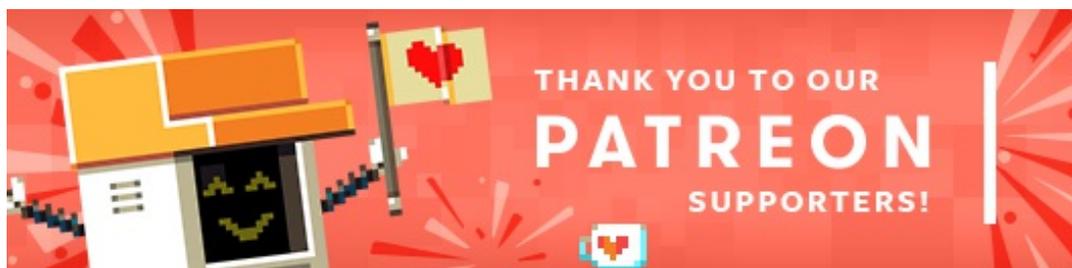
- 
FREE COPY OF INTERPHASE
DOWNLOAD A FREE COPY OF THIS PHASER 2 BOOK AND CODE PACK - PLUS GET EARLY ACCESS TO INTERPHASE 3
- 
PHASER STICKERS PACK
PUMP YOUR RIG WITH THESE GREAT VINYL STICKERS! FREE FOR \$30+ AND DISCOUNTED FOR ALL TIERS BELOW
- 
ROADMAP ROUND-TABLE
HELP STEER THE FUTURE OF PHASER BY JOINING THE MONTHLY ROUND-TABLE DISCUSSING WHAT'S COMING
- 
GAME PROMOTION
WANT SOME MORE EYEBALLS ON YOUR LATEST GAME? WE CAN HELP WITH SITE AND NEWSLETTER PROMOTIONS
- 
ADVERTISING OPPORTUNITIES
FOR HIGH TIER BACHERS - WE CAN RUN BANNER ADS, TWEETS AND MAILING CAMPAIGNS FOR YOUR PRODUCTS
- 
HELP SECURE PHASER'S FUTURE
THE MOST IMPORTANT REASON OF ALL: YOUR BAZING HELPS SECURE THE FUTURE OF PHASER



Click to see the full list of backer perks

I use Patreon to manage the backing and **you can support Phaser from \$1 per month**. The amount you pledge is entirely up to you and can be changed as often as you like.

[Please help support Phaser on Patreon](#)



Thank you to these awesome [Patrons](#) who recently joined and make continued development of Phaser possible:

Anders Rochester
Andy Wigham
Creative Spark Studios
Daniel Cassidy

Gary Stanton
Gnun Ulikhanyan
Moritz Wundke
Paul Martinetti
Proto
Richard Kunkel

Also, thank you to **Mike Carter** and **Dara Sami** for increasing their pledges :)



Dev Log #143

Welcome to Dev Log 143. My work has been split over so many different tasks recently that this Dev Log is going to feel like a shotgun blast of small bits of information. To be honest, this is likely to carry on for a few weeks yet, as lots of ends start being tied together.

My continued focus has been directed on the new docs site, as well as diving in to GitHub issues as well. When I split my time like this it means both tasks take longer to complete, but as I mentioned last week, I feel it's important. Plus, there's only so much php I can take in a day :) The docs



24k GitHub Stars!

During the week, the Phaser GitHub repo hit 24,000 stars. This is quite amazing. Thank you to everyone who has [starred the project](#) (if you haven't, please do so!). I know that these stars don't really mean much, but you have to understand that being an open source project means I have precious few metrics by which to track how things are going.

Unlike, say, Construct, or Unity, I have *absolutely no idea* how many developers

are using Phaser. I can infer some guesses, based on what few metrics I do have, such as newsletter subscribers, web site traffic and GitHub stars, but I'll never truly know. Which is kind of sad in a way, yet inevitable. So even tiny things, like a little star counter against the project, can mean more to me than they perhaps should.



Arcade Physics Updates

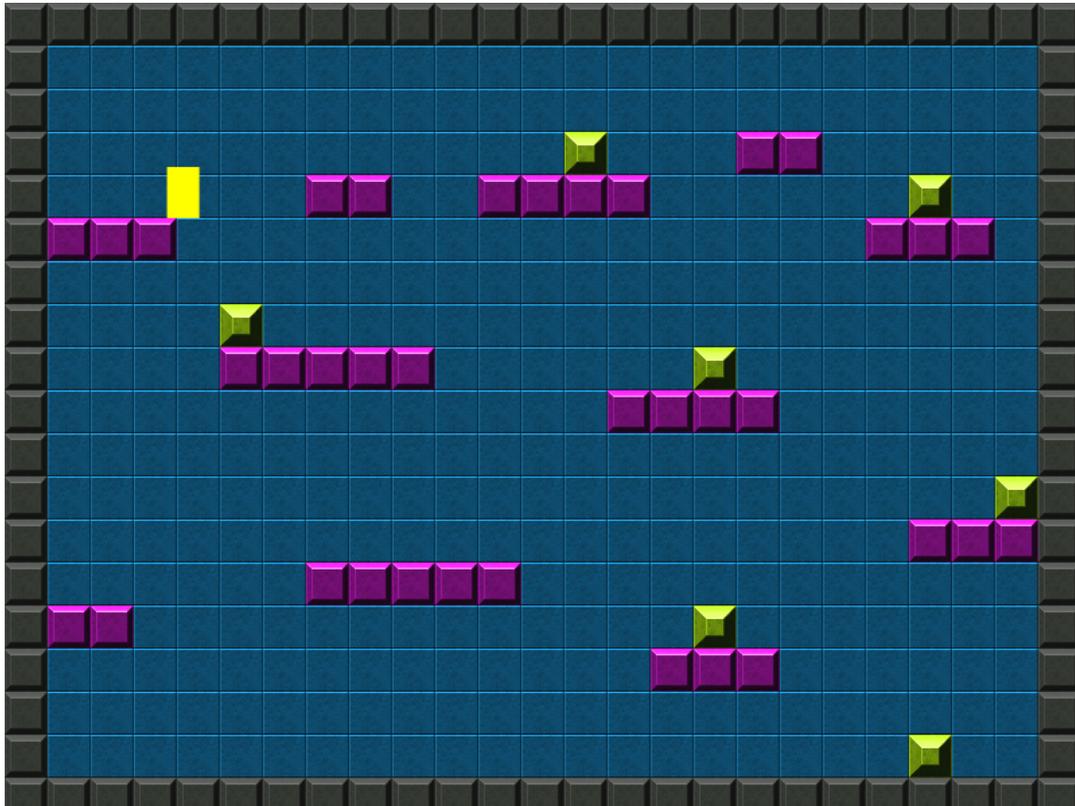
I've recently fixed a number of lingering Arcade Physics issues and changed the way the game and physics step is handled slightly. In a previous update, I moved to a fixed timestep, allowing for deterministic results from Arcade Physics, something that didn't happen previously. I spent some time refining this flow further and ironing out some issues.

The physics body update has been broken down into 3 stages. First is ``preUpdate``. This syncs position data from the parent Game Object and also resets all the internal collision flags. This is run only once per game step. Next, there is the ``update`` method. This now does nothing more than calculate the new velocity and handle bounds collisions. Depending on the physics step, this can be run multiple times per frame. Finally, there is ``postUpdate`` which is responsible for syncing the changes in the body back through to the parent Game Object.

In the previous iteration, all of these tasks were being done at once in the same step. By splitting them up I've been able to tidy up a lot of internal code in the World class, refactor the Body `postUpdate` method and fix a number of bugs at

the same time, including the ability to run your own collision calls during a Scene update.

If you use Arcade Physics in your game, please can you [help test the new build](#) and report if anything breaks.



World.overlapTiles

As part of the work I was doing inside Arcade Physics, I took the time to implement a feature I had been wanting in there for months: the ability to check for overlaps with specific tiles from a Tilemap, regardless if they have been flagged for collision or not.

The new method, accessible via the Arcade Physics World instance, is called `overlapTiles`` and it takes a single physics-enabled Game Object, such as a Sprite, and then performs overlap detection against any tile you provide in the input array. The tiles don't have to have been set for collision, either in your map editor, or via code. They don't even have to be on the same Tilemap layer, the method doesn't care, it will run overlap tests against them all in turn, firing your provided callback when an overlap occurs.

I put together a demo to show how it works. You can see it in the screen shot above, click it to test it out. Use the cursors to move and jump. All you need to do is collect the yellow tiles.

The method works by taking an array of tiles. So you need to build this up first. The Tilemap API has a method called `filterTiles` which can be used to do this. For example, in our test map, the yellow tiles have an index of 82. So we build an array of just those tiles:

```
var pickups = map.filterTiles(function (tile) {  
    return (tile.index === 82);  
});
```

We then pass this array to the `overlapTiles` method, along with our player sprite and a callback to invoke, should an overlap occur:

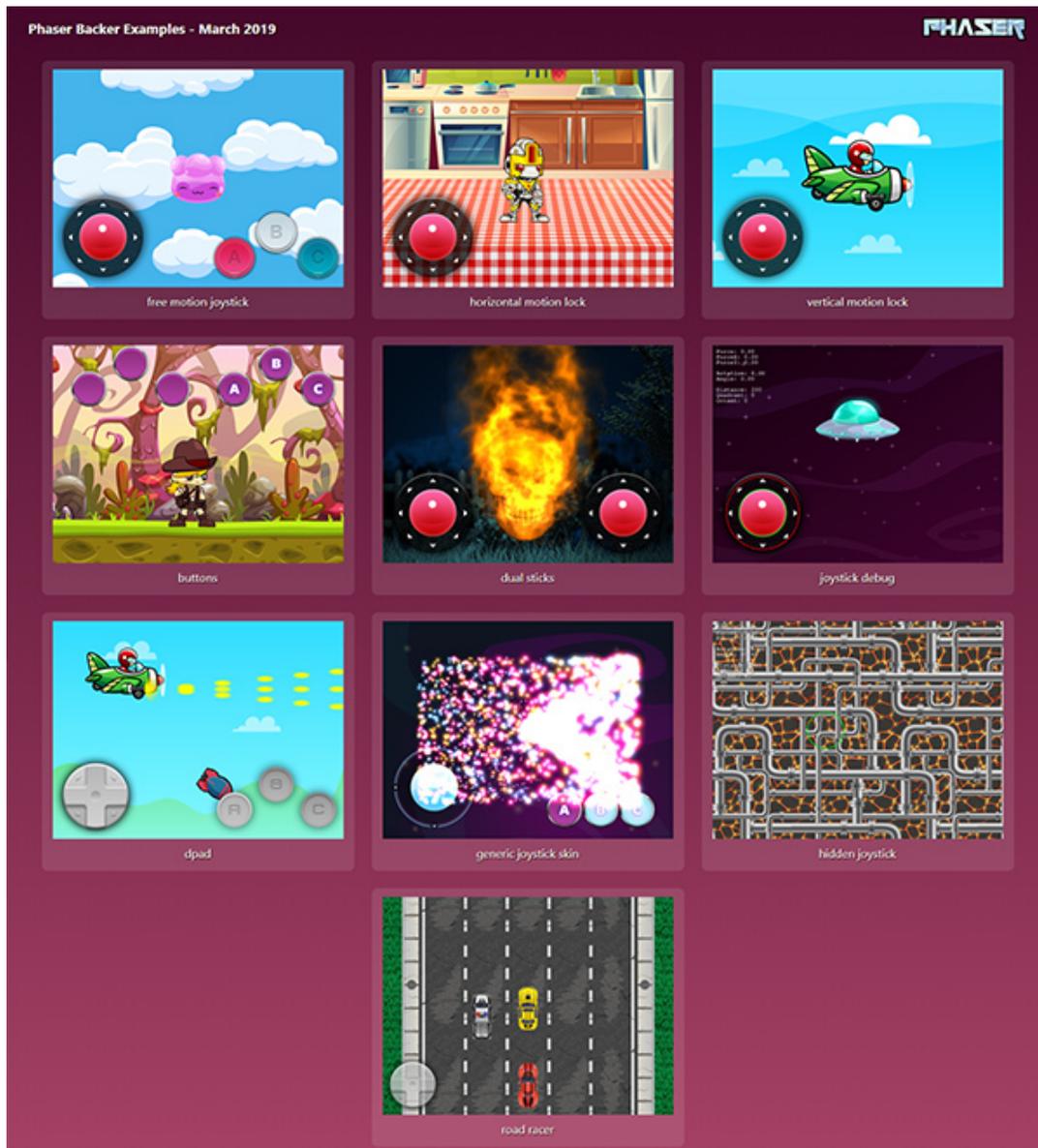
```
this.physics.world.overlapTiles(player, pickups, hitPickup, null, this);
```

The above code is run in our Scene `update` method in this demo, but you could call it from anywhere. If the Game Object overlaps with a tile, both items are sent to your callback, where further action can be taken. In the example demo above the tile is removed from the map and the array is refreshed.

There are several benefits to using this method, combined with a couple of drawbacks. The most obvious benefit is that you can dynamically create the array of tiles to be tested against, as your game requires it. Each different array could redirect to a different callback too. What's more, because it's just a tile instance it doesn't care if the tile can collide or not, meaning you don't need any extra steps in your level set-up or code.

The downside is that it's an $O(N)$ method. That means, for every single tile you pass to the method, each one of them will be checked against the Game Object. No filtering or spatial culling is performed. This makes it perfect for small quantities of tiles, but increasingly more expensive the larger the numbers go. As with most things like this, test it for yourself to see the impact on your game. If the array is small enough you may not even notice it.

This new method can be found in the master branch on GitHub, along with all other changes targetted for 3.16.3.



March Backers Examples Bundle with free Plugin!

Earlier this week I released the March Backers Examples Bundle to everyone who [supports Phaser via Patreon](#) or PayPal. I release a bundle of examples each month to backers and they're slowly building into quite a handy resource, especially as each bundle contains all previous months, too.

This month was a little different. As well as the examples I gave all backers a free copy of the brand new Phaser 3 Virtual Joystick Plugin. This plugin was originally released several years ago for Phaser 2 and has proved popular ever since. It allows you to easily add a virtual joystick and buttons to your game, perfect for mobile users.

I took some time to recode the plugin from scratch, adding features as I went. The new plugin was written fully in ES6 and packaged using Parcel and Babel. It

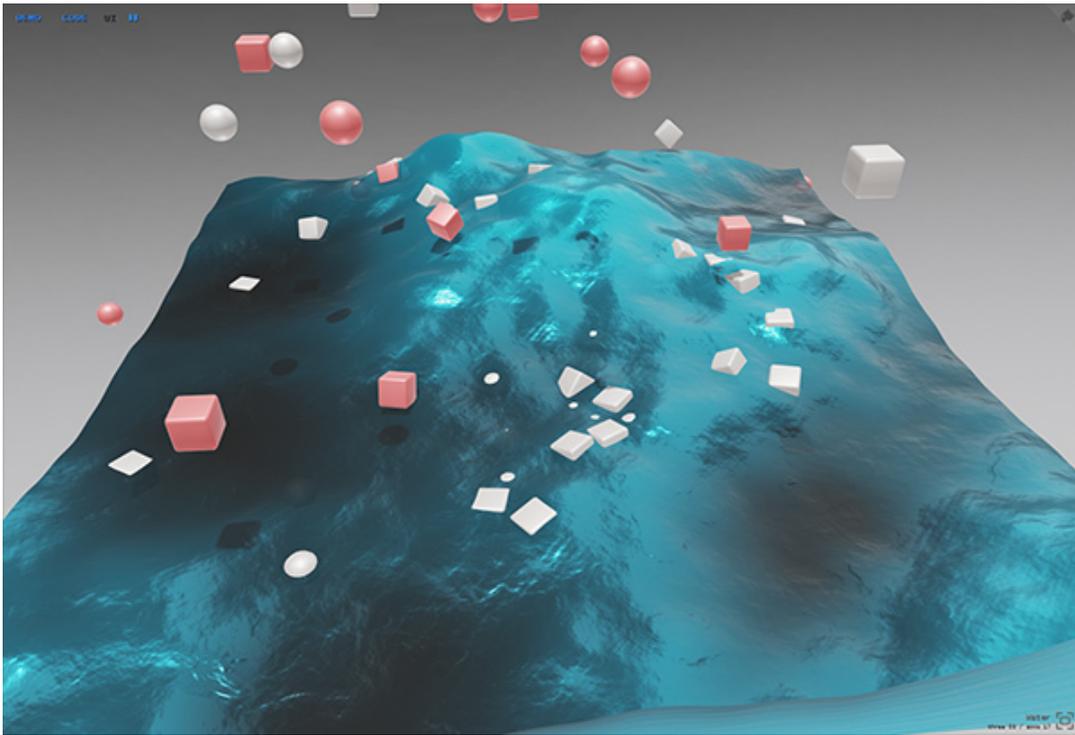
was my first time using Parcel and I have to admit the experience was a bit mixed. The main benefit of Parcel is that it's "zero configuration". This is the polar opposite to Webpack, which has a configuration file complexity akin to learning a second language. This aspect of Parcel was fantastic. It really did bundle out of the box with nothing more than a single command-line call.

However, because I needed a slightly different output than its defaults are used to, it took several hours of frustrating trial and error before I finally stumbled upon a semi-related GitHub issue, in which the Parcel author had posted a link to a tiny, easy to miss repo, that finally demonstrated a specific way of running Parcel that provided the bundle output I required. With this new found knowledge in place I was able to get it to package the plugin and it worked within Phaser 3 perfectly.

There's definitely a potential tutorial-in-the-making from my experience with this! After solving that issue I was able to code merrily away in ES6 and test it instantly. The new joystick plugin uses fully native Phaser 3 features, as you'd expect, includes some updated PSDs, 3 different skins, 10 examples, a getting started guide and full documentation generated using ESDoc. This was another first. Previously I had used JSDoc for everything, but I figured as I was coding in ES6 anyway, why not use ESDoc instead? As it transpires, ESDoc is even more opinionated than JSDoc! Too much for me to ever consider using it for Phaser itself, but it just about managed the plugin with a few tweaks.

Anyway, overall I was very pleased with the end result, even if the new tooling experience required to get there was a painful learning curve with plenty of dead-ends at times. All Backers will now have a copy of the plugin. If you become a backer after reading this, you can find the post on my Patreon page and download it from there! For everyone else, I will be releasing the plugin onto the Phaser site for sale in the coming days.





[Ammo.lab](#) is a great playground site for ammo physics tests. Some of the demos are superb!

[iOS 12.2 beta removes gyro support from Safari](#) and puts it behind a privacy setting. If your game relies on this, be aware of this coming change.

[JScrambler 6.0](#) is out and it's a major update packed with new client-side protection and a new Threat Monitoring dashboard.

Phaser Releases

Phaser 3.16.2 released February 11th 2019.

Phaser CE 2.12.0 released 6th February 2019.

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.



©2019 Photon Storm Ltd | Unit 4 Old Fleece Chambers, Lydney, GL15 5RA, UK

[Preferences](#)

[Forward](#)

Powered by **Mad Mimi**®
A GoDaddy® company