

# PHASER WORLD

FEBRUARY 2018

ISSUE  
115

## ZOKA DICE



THIS WEEK...

ZOKADICE

ANDROID TV APP TUTORIAL

CHASE THE SHADOW

PHASER ON WECHAT

Welcome to Issue 115 of Phaser World

It's a little shocking to think we're in February already! 2018 has flown by so fast. Yet as we gear up for the Phaser 3 release I was pleased to realize that Phaser

will celebrate its 5th birthday on April the 12th. It's amazing to think what has happened in the span of just five years. And still, you guys are creating awesome games and tutorials for it, as this issue demonstrates yet again! Thank you :)

Until the next issue, keep on coding. Drop me a line if you've got any news you'd like featured by simply replying to this email, messaging me on [slack](#) or [Twitter](#).



## The Latest Games



### Game of the Week

#### [ZokaDice](#)

This is the classic five dice game of strategy and luck but with a massively multiplayer twist and top-notch presentation!



### Staff Pick

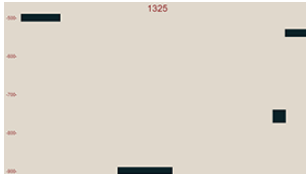
#### [Chase the Shadow](#)

A 1 vs. 1 arena combat game in which two wizards vie to destroy each other with spells, using only their shadows to aim.



### Pulsar

Avoid the bullets, collect the stations and try your best to make the exit in this bullet-hell puzzle game.



### Light Box

An infuriating yet strangely compelling game of leaping from box to box.



### Asteroids

You know the drill - blast those rocks to smithereens and do away with the alien patrols too.

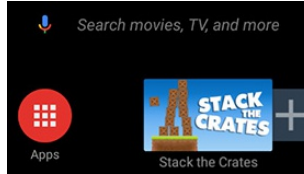


## What's New?



### Turn Based Battle System Tutorial

A comprehensive tutorial and impressive demo showing how to implement your own turn based battle system and visual queue.



## Native Android TV App Tutorial

A tutorial on porting your game to run as an app on Android TV using Android Studio and Cordova.



## SantiagoJS Meetup Phaser Talks

Two great talks on shaders, WebGL and Phaser at the Santiago JavaScript meet-up.

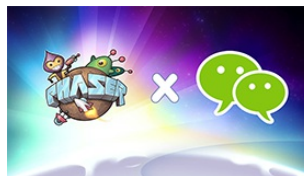
```
game.movedToStop_tk1(4193a70b0c7c, x, y, speed, maxTime, events)
// It takes an event's eventID to ensure that the object will stop at the exact x,y position.

Arguments
  • [displayObjectID, displayID]: any. The display object to move.
  • [x, y]: number. The x coordinate to reach.
  • [x, y]: number. The y coordinate to reach.
  • [speed]: number. [pixels/sec]. The speed it will move, in pixels per second (default is 40 pixels/sec)
  • [duration]: number. [seconds]. The given in milliseconds (1000 = 1 sec). If the speed is adjusted so the object will arrive at destination in the given number of ms.
  • [events]: [object]. List of events to trigger.

Example
game.movedToStop_tk1(4193a70b0c7c, 10, 10, 100, 1000, {
  onComplete: () => {
    console.log('Object moved before reaching position!');
  },
  onEventReached: () => {
    console.log('Object reached!');
  }
});
```

## Move and Stop Plugin

A plugin to allow you to move an item to a given position via physics, and have it stop precisely.



## Running Phaser on WeChat

A guide on setting-up Phaser CE with the WeChat Developer Tools to build your own micro game on their service.



### Official Phaser T-Shirts

A new range of Phaser T-Shirts are now available from Amazon in the US. Available in 5 different colors with male and female fits, look the part at your next game jam or meet-up!



Thank you to our awesome new [Phaser Patrons](#) who joined us this week: **Andrew Styles**, **Brad Dickason**, **Nathaniel Foldan** and **Alexis Eutrope**.

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 #115

A relatively short Dev Log this issue I'm afraid, because we're flat out preparing for launch. And on that note I'm going to copy the post I made to Patreon here, in case you missed it:



# PHASER

If you follow our Dev Logs you'll know how insanely busy we've been recently preparing Phaser 3 for release. If you read Dev Log 114 you'll also know we were meant to release v3 today, but that we weren't quite ready. I had a talk with Mozilla, who awarded us the money to be able to work on v3, to see if there was any flexibility in the release date. They agreed there was. As it actually turns out, as long as we release within 1 year of the contract date, it's all good with them!

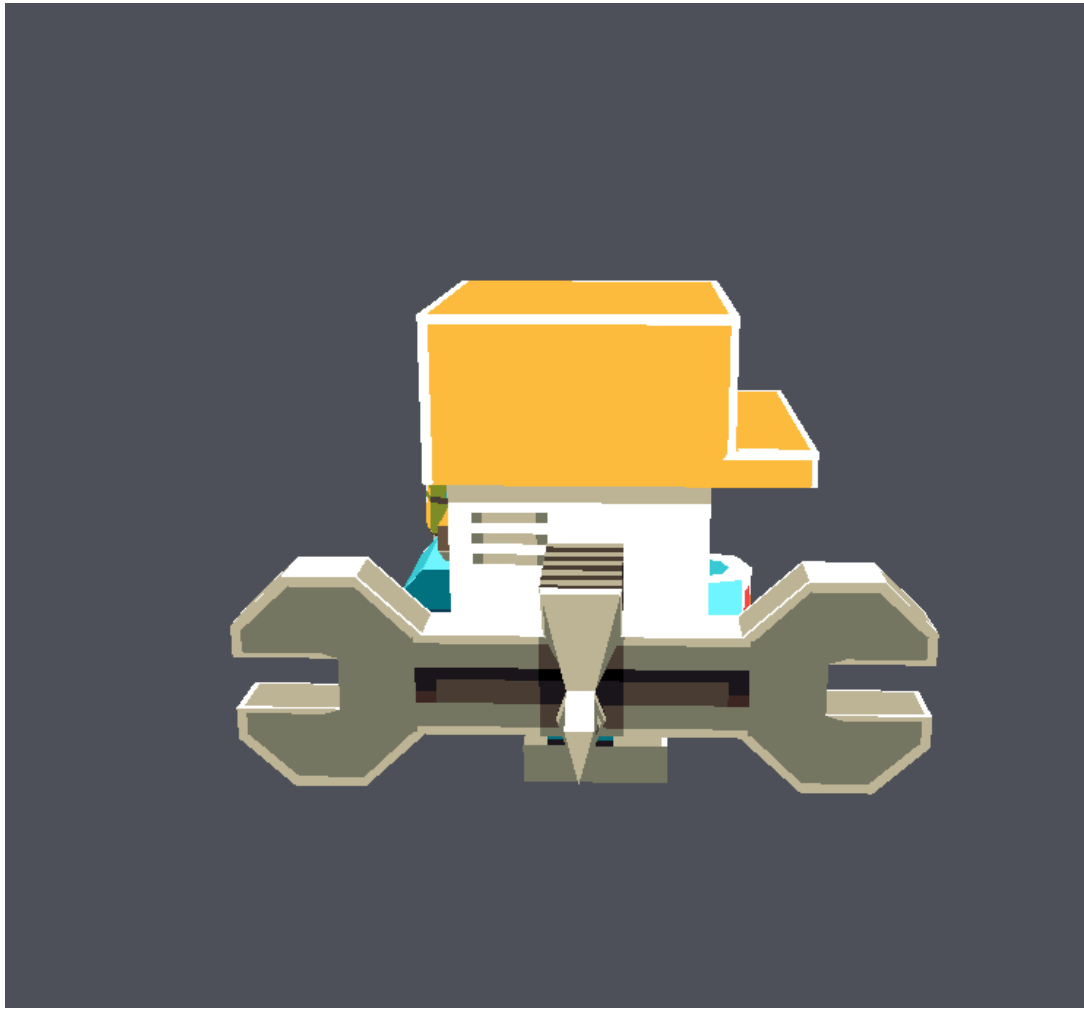
Thankfully, we don't need a year. We just need a couple of weeks. The API is solid, the new WebGL Pipeline code is in, all the internal systems have moved over to plugins and I have been working on the documentation like a man possessed. This week I hit a significant milestone in that regard, in that the docs are now building properly and really coming together! It took some serious jsdoc wrangling, believe me, but it works. It was quite something to see the first build of them in a browser. They feel tangible at long last.

I've still a lot of documentation left to write, but it feels achievable. So much so, that barring any acts of God, we're setting the 3.0.0 release date to be February 12th. A proper fully documented release. Yes, we removed a few features from the API and will add those back in the coming months, but there is so much to get your teeth in to! With the docs in place, you'll be able to use them alongside the examples and really go to town learning what cool new things v3 can do.

Thank you all for bearing with us this past year, and indeed longer. V3 has been such a massive amount of work, so long in the making. I'll be honest, it has been hard to stay positive across all that time and even I doubted myself more than once. It felt like it could have become a bit of a Duke Nukem Forever at some points.

There are elements we have done, re-done, then done again as the gears shifted in to place and we revisited old code. In hindsight, I feel that these are all signs that what you're working on is a true labor of love. That couldn't be truer.

To celebrate the fact we're close to release here is a lovely 3D animated Phaser bot created by our pixel-master, [llkke](#):

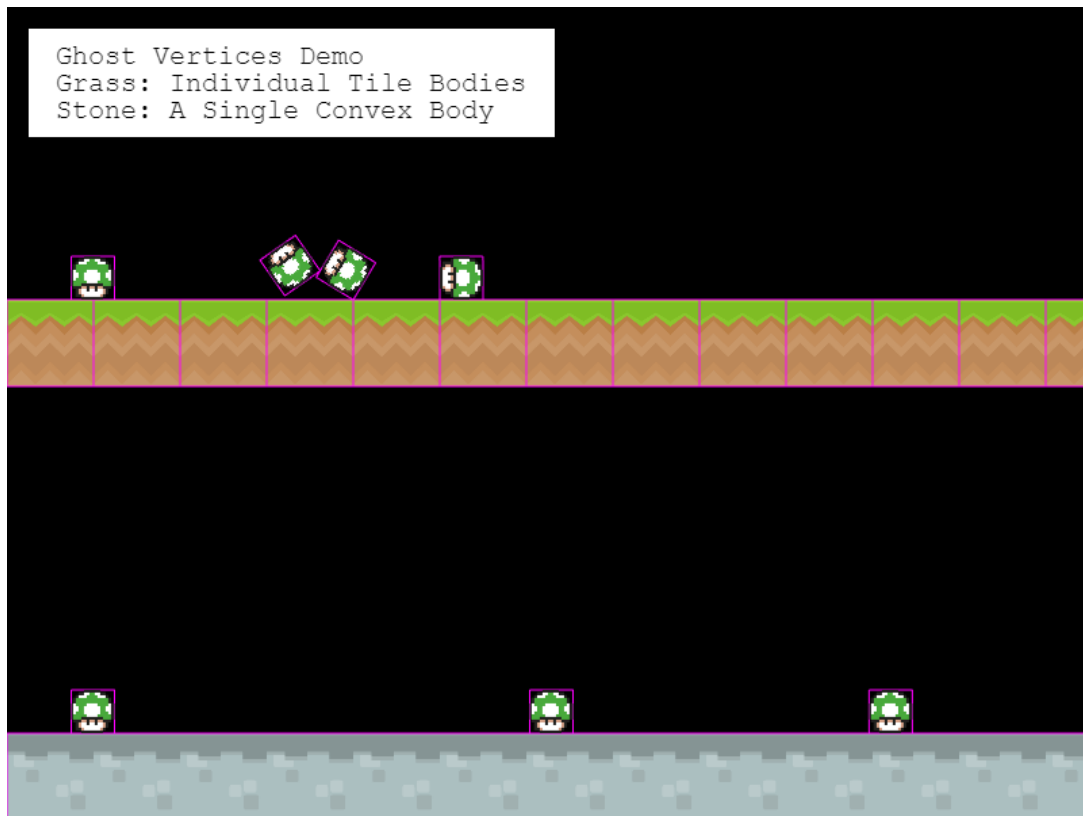


## Tilemap and Physics

In the last issue Michael wrote part 1 of what was new with the Tilemaps API in Beta 20. Here's part 2:

Last issue I showed you how to modify a map in real-time using invisible triggers. I said at the time that the demos were meant to show what is possible, but that isn't necessarily what's best.

Along with all that, you'll likely run into "fun" ghost collisions when working with Matter + Tilemaps. Check out the example below to see what I'm talking, and then check out the comments in the code for some suggestions on how to deal with them:



Next up, improvements on flagging tiles as colliding. Using indices can be a bit tedious. Two new Tilemap methods will help ease that: [setCollidingByProperty\(...\)](#) & [setCollidingFromCollisionGroup\(...\)](#). Those links will take you to examples. They make use of Tiled's [tileset editor](#). This is where you can define different pieces of information per tile in a tileset: custom properties, collision shapes, animations, etc.

```
1 // If a tile has any of the given properties & values it will be set to collide:
2 map.setCollidingByProperty({ collides: true });
3
4 // You can also pass in an array of values - if the type is either rock or grass, it collides:
5 map.setCollidingByProperty({ type: [ 'rock', 'grass' ] });
6
7 // Using a different method, you can set any tile with collision data to collide:
8 map.setCollidingFromCollisionGroup();
```

### A few Impact updates:

Last week, I demoed how you could use the Impact engine with Tiled maps, which you might want to do if you need slopes or Arcade physics isn't cutting it. To load a tilemap in Impact, you need to specify a mapping from tile indices to slope IDs (see the [slope reference tileset](#)), and there are a couple new ways to do that:



```

1 // If you have slope IDs defined as Tiled properties:
2 this.impact.world.setCollisionMapFromTilemapLayer(layer, { slopeProperty: 'slope' });
3
4 // If you'd rather specify a custom mapping (tile index -> slope ID):
5 var slopeMap = { 32: 1, 77: 1, 82: 1, 95: 24, 137: 2, 140: 24, 36: 2 };
6 this.impact.world.setCollisionMapFromTilemapLayer(layer, { slopeMap: slopeMap });
7
8 // Or, if you have a map that you want to load that only has solid blocks (no slopes):
9 this.impact.world.setCollisionMapFromTilemapLayer(layer, { defaultCollidingSlope: 1 });

```

Arrow keys to move. Press "up" to jump.



This update also makes it really easy to load maps without slopes using Impact: see this [example](#).

The last set of updates is around the general Tilemap API. A few things have been added to make working with physics a bit easier. Tiles now have `tileset`, `tilemapLayer` and `tilemap` (getter) properties. If you need to access the Tiled data for a tile, there are `getCollisionGroup(...)` and `getTileData(...)` helper methods. There are also a few methods to get the world position of a Tile, like `getBounds(...)`. They all make it a bit easier to get at whatever data you need directly from a Tile.

Since you may find yourself mapping out triggers & colliding shapes within Tiled's object layers (which allow you to draw out shapes on top of your level), the Tiled object parser got a few upgrades. Along with under the hood upgrades, there are a few new helpers for finding objects:

```
1 const trigger = map.findObject('Trigger Layer', obj => obj.name === 'tripwire');
2 const switches = map.filterObjects('Collision Layer', obj => obj.properties.type === 'switch');
```

And that's all for this week. Fingers crossed, next week should be quite a special issue :)

## Phaser 3 Labs

[Phaser 3 Beta 20](#) is out and ready for testing.

Visit the [Phaser 3 Labs](#) to view the 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!



Neat HTML5 re-creations of famous [C64 cracker intros](#) from the likes of Fairlight, Laxity, Alphaflight, EagleSoft, Hotline, Level 99 and many more.

If you're still using `console.log()` to find and fix JavaScript issues, you might be spending more time debugging than you need to. [This tutorial](#) shows you how you can debug your JavaScript as quickly as possible.

[Threed.io](#) allows you to create cool mock-ups of things running on mobile devices. It's easier to play with than explain, so go check it out :)

---

## Phaser Releases

**Phaser CE 2.10.0** released January 18th 2018.

**Phaser 3 Beta 20** released January 30th 2018.

Please help [support](#) Phaser development

Have some news you'd like published? Email [support@phaser.io](mailto:support@phaser.io) or [tweet us](#).

Missed an issue? Check out the [Back Issues](#) page.



©2018 Photon Storm Ltd | Unit 4 Old Fleece Chambers, Lydney, GL15 5RA, UK

[Web Version](#)

[Preferences](#)

[Forward](#)

[Unsubscribe](#)

Powered by [Mad Mimi®](#)

A GoDaddy® company