Welcome to Issue 109 of Phaser World

Regular readers will know this issue is a week and a half late. I can't go into the details, but suffice to say it was for reasons completely outside of my control.

However, we're back, and just before Christmas too!

There is finally an official Phaser Twitter account! Unlike the others out there, this one is actually run by us. We'll use it to tweet news items, examples and retweet cool Phaser things we see - so if you're on twitter give us a follow and send your games our way!

Issue 110 of Phaser World will be out January 1st 2018. I really can't believe it's almost 2018 but I'm really excited about what it will bring. I'll try and do my yearly round-up of what happened to Phaser in 2017 and what I see for the year ahead. I'll also take a look back at last years predictions and see how many of them came true.

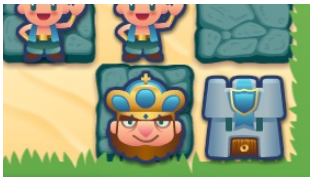
So, until the next issue, keep on coding and have a fantastic holiday time. Drop me a line if you've got any news you'd like featured. You can reply to this email or grab me on the Phaser Slack channel.

The Latest Games

Game of the Week

Pirates! The match-3

It's a rip-roaring high seas adventure with gems to match, buried treasure to uncover, sea monsters to defeat and chests to plunder in this top-quality game.



Staff Pick

Chess Challenges

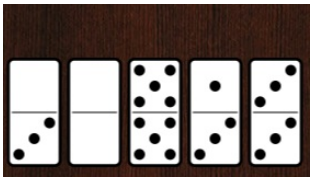
Can you destroy the castle and defeat the King by solving these chess puzzles in this great game?

Christmas Gifts

A candy-cane, gingerbread, Christmas stocking filled match 3 game.

Mystic Sea Treasures

Deep beneath the sea, lies a magnificent treasure waiting to be found in this match 3 hexagon adventure!



Chickenfoot Dominoes

A browser based version of the classic game Chickenfoot Dominoes. Can you win the round by scoring the lowest?

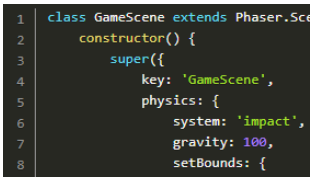
What's New?

Phaser CE v2.9.3 Released

The new Phaser CE release includes BitmapData polygon drawing, lots of new Point methods, fixes, updates and TS and doc enhancements.

DragonBones Phaser Plugin

The Phaser runtimes for this top-end skeletal animation system get an update.



```
class GameScene extends Phaser.Sce
    constructor() {
        super({
            key: 'GameScene',
            physics: {
                system: 'impact',
                gravity: 100,
                setBounds: {
```

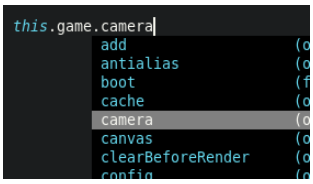
Phaser 3 Tutorials

3 new tutorials on creating physics enabled sprites in Phaser 3.

Isometric Worlds and Real Time Games

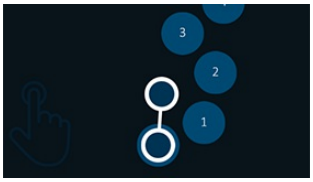
All of the slides and code from a recent talk and workshop on creating real-time

games.



Using Phaser with Vim

A visual tutorial on how to set-up Vim for Phaser development.



Risky Steps Updated

The Risky Steps game is reskinned and updated with the full source code available.

Thank you to our awesome new Phaser Patrons **Daniel Klava**, **Nikolas Lotz**, **Paul Boudreaux** and **Jared Smith**. It's really great to have so many join at once :)

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

This week we're featuring the Complete Fantasy Game UI Kit. It contains backgrounds, borders, buttons, icons and more. Vector files included so you can scale to whatever size you need.

Dev Log #109

You can try all new features shown in this Dev Log in Phaser 3 Beta 14. Grab it from GitHub pre-built, or from npm using the beta tag.

Sorry this Dev Log is a week late. The reasons are not something I can go into in depth here, suffice to say it involves being a Police witness to the near-death of two relatives and managing the aftermath. It wasn't exactly something you could ever plan for and as you can appreciate it took priority over work. I'm very glad to now be back in the world of Phaser though!

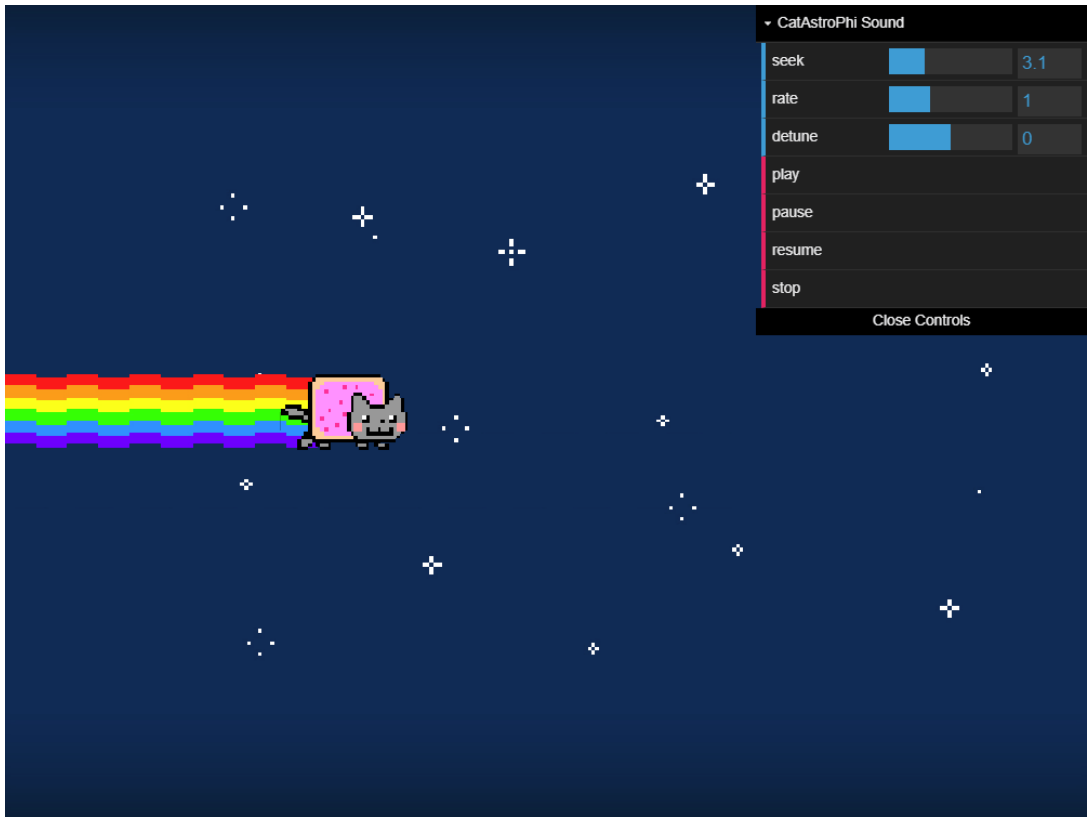
Sound Manager Update

Pavle writes: Last week was all about implementing a seeking feature for the v3 Sound API. It allows you to read the current position from a sound and set it to a desired value.

It was quite a challenge since the Web Audio API does not support seeking natively, so a lot of logic had to be added on top to make it work. Also, another tricky part was taking into account the playback rate and detune changes as they too very much affect calculating the current position of the sound.

Now that we checked seeking off the list, we are one step closer to being able to create a full-blown music player with Phaser v3.

The work was all very technical, so in order to lighten things up a bit, I made this example featuring Nyan Cat. Have fun dragging the cat around! :D



Drag nyan cat!

Pointer Lock

After his awesome work on the Tilemap API, we asked if Mike could work on a couple of other features on the todo list. One of them was adding support for Pointer Lock into the V3 Input Manager:

I integrated the pointer lock API in v3. The Pointer Lock API in the browser allows you to hide the mouse cursor and makes it so that the cursor is no longer limited

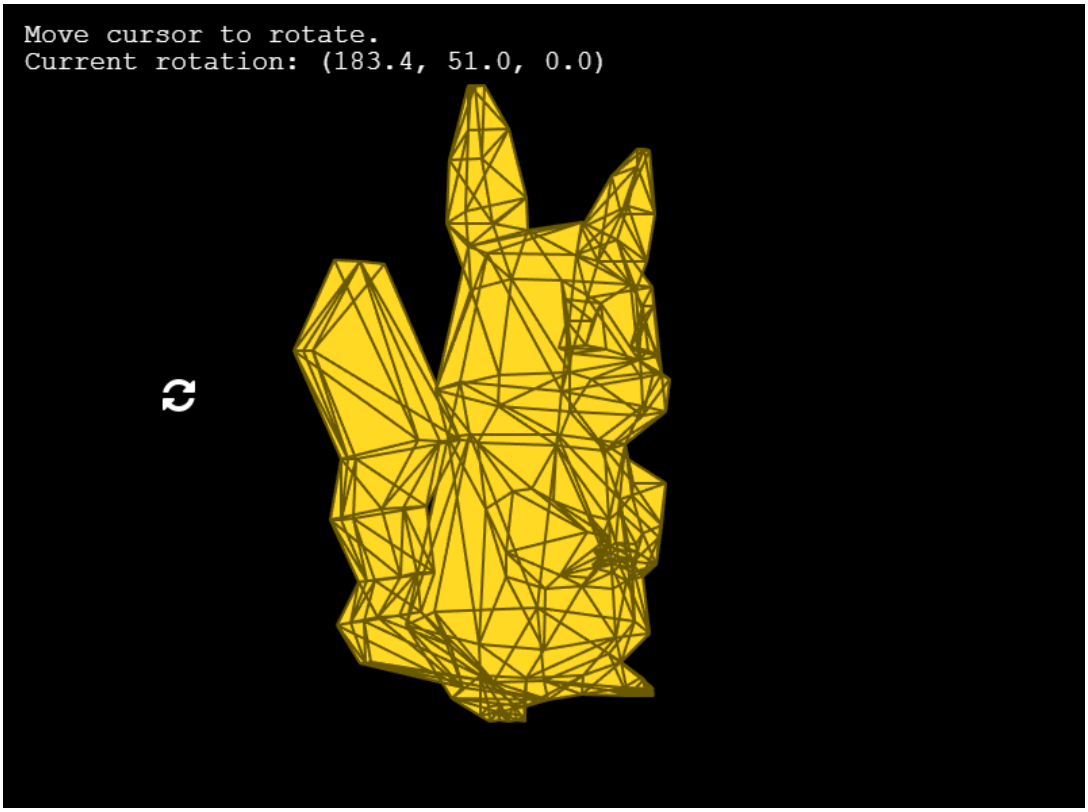
by the browser/screen boundaries. You might be familiar with this from first person 3D games, but it can also be useful when you want to create mouse-based controls in a 2D game. Check out the clever use of pointer lock in the multiplayer game cursor.io where the player is literally a mouse cursor.

Once you successfully initiate a pointer lock in v3, the active pointer will have a movementX and movementY property that tells you how much the mouse has moved since the last update:

```
var sprite = this.add.sprite(400, 300, 'ball');

// Pointer lock will only work after an 'engagement gesture', e.g. mousedown, keypress, etc.
// We are using the a vanilla DOM event to initiate the lock.
game.canvas.addEventListener('mousedown', function () {
  game.input.mouse.requestPointerLock();
});

// When locked, you will have to use the movementX and movementY properties of the pointer
// (since a locked cursor's xy position does not update)
this.input.events.on('POINTER_MOVE_EVENT', function (event) {
  if (this.input.mouse.locked)
  {
    sprite.x += event.pointer.movementX;
    sprite.y += event.pointer.movementY;
  }
}, 0, this);
```

Click to see demo

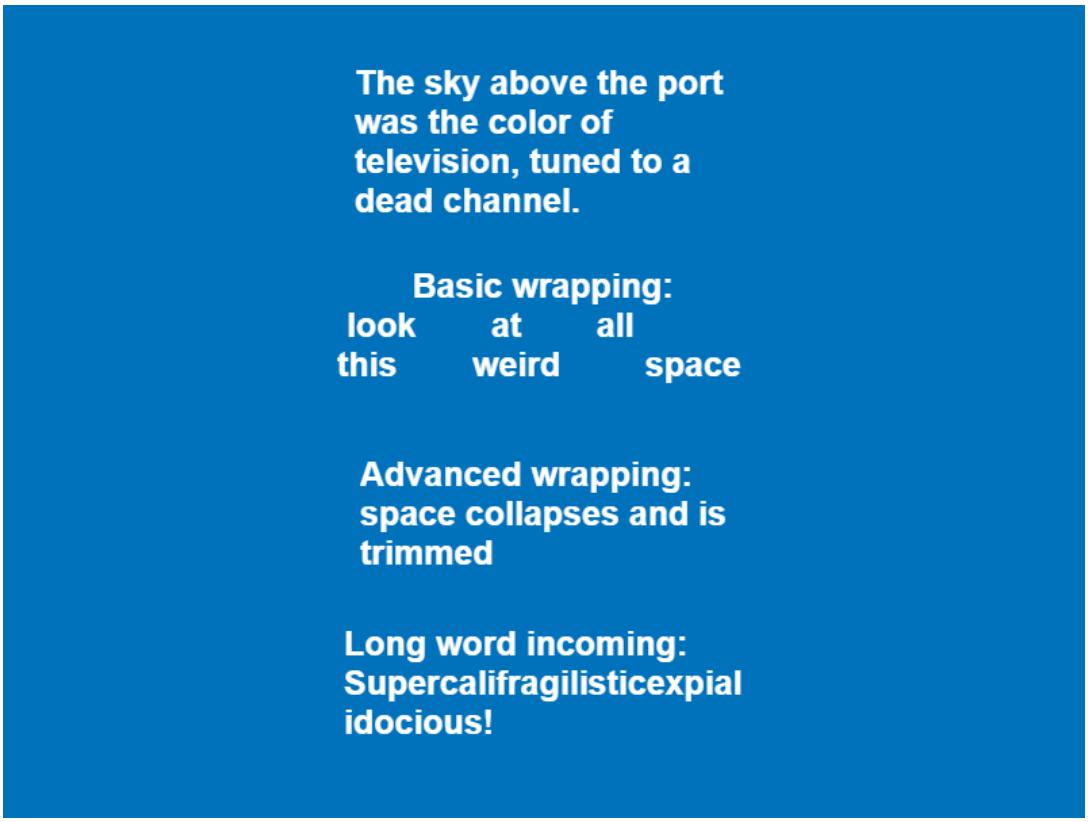
The annotated example in phaser labs also shows how to unlock the pointer and how to listen for changes in the pointer lock state. You can also use pointer lock with the new 3D mesh features:



Click on the screenshot to rotate a low poly Pikachu.

Text Wrapping

Mike also worked on word wrapping this week, which comes with a couple of options. We've got basic and advanced wrapping as in v2 - click on the screenshot below to see the differences:



Word Wrapping options

You can specify the wrapping in the style option or via methods on a Text instance:

```
this.add.text(0, 0, 'The quick brown fox jumps over the lazy dog.', {
  font: 'bold 25px Arial',
  fill: 'white',
  wordWrap: { width: 300, useAdvancedWrap: true }
});

this.add.text(0, 0, 'Glib jocks quiz nymph to vex dwarf.')
  .setWordWrapWidth(300, true); // Parameters: max width of a line, useAdvancedWrap
```

Maybe you need some language-specific wrapping, or just want to inject your own line breaking logic? In that case, you can specify your own custom wrapping function:

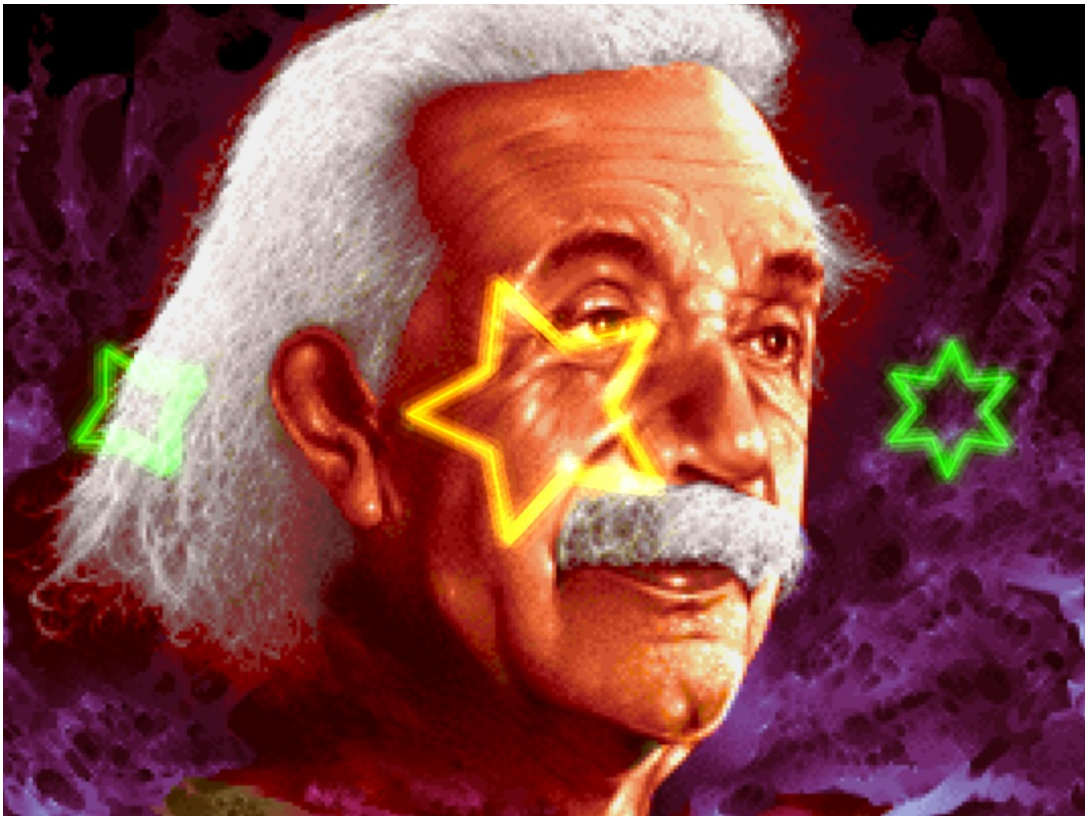
```
this.add.text(0, 0, 'How vexingly quick daft zebras jump!', {
    font: 'bold 25px Arial',
    fill: 'white',
    wordWrap: { callback: customWrap, callbackScope: this }
});

function customWrap (text, textGameObject) {
    // You can return an array of lines or a string with line break
    // characters (e.g. '\n') where you want them
    return text.split(' ');
}
```

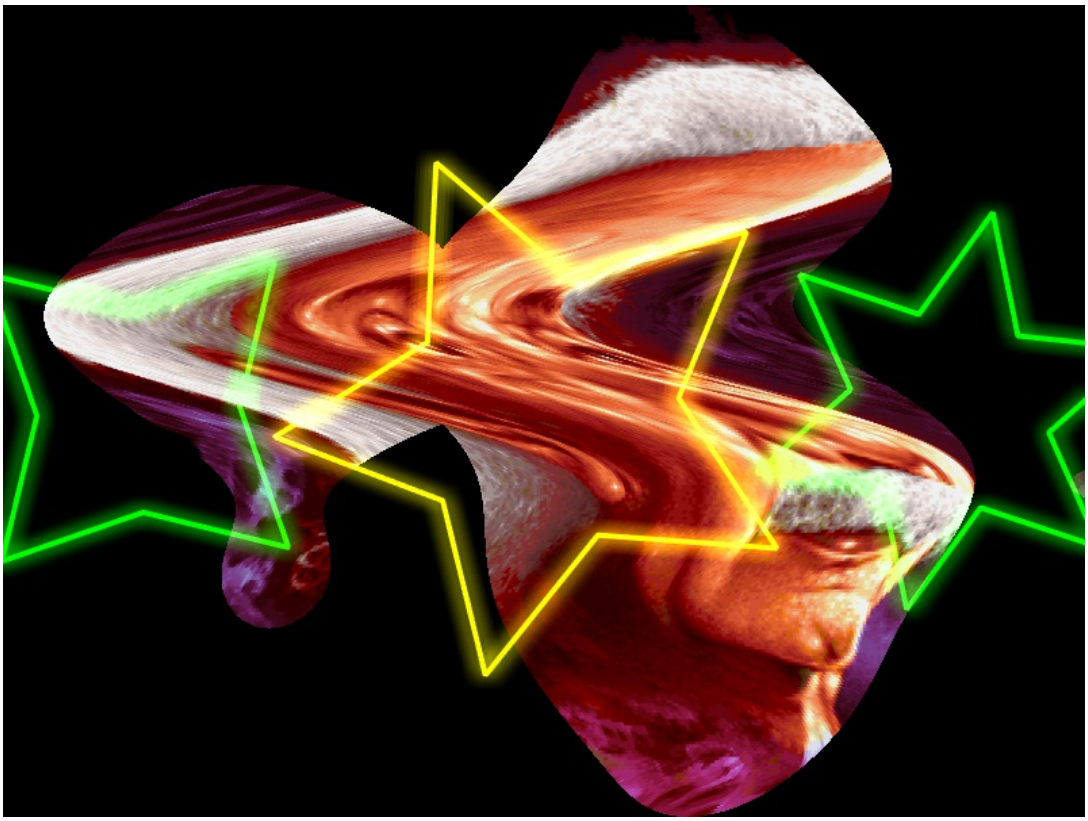
My thanks to Mike and Pavel for their help with Phaser 3 this year. As we're now in holiday season their next contributions will hopefully be with Beta 15 in January as we head towards a final release.

Glow Shader

This was something I asked Felipe to work on because I had seen it requested so many times on the forum and in chat: the ability to add a glow to a sprite. The first stage of this was to create a shader that could be used by a RenderPass. He set to work exploring a few different techniques and in the end settled on two different shaders: the first used a fast gaussian blur technique (originally devised by Jam3). This used just one RenderPass and was the quickest option, in terms of rendering. The second method was a soft glow effect that used two shaders, one for the horizontal blur and one for the vertical blur.

You can see the difference in the following two examples:





Although the demos are slightly different you can see the subtle differences in the glow effects as well. In these examples, they're just being applied to a Graphics object but they can actually be used on anything renderable.

The plan is to introduce a range of Special Effects game objects that allow you to

easily control effects like glow and manage the display list updating for you. These will be post-launch of 3.0 but are in planning already. In the meantime, the curious among you can dig around with the source in the examples :)

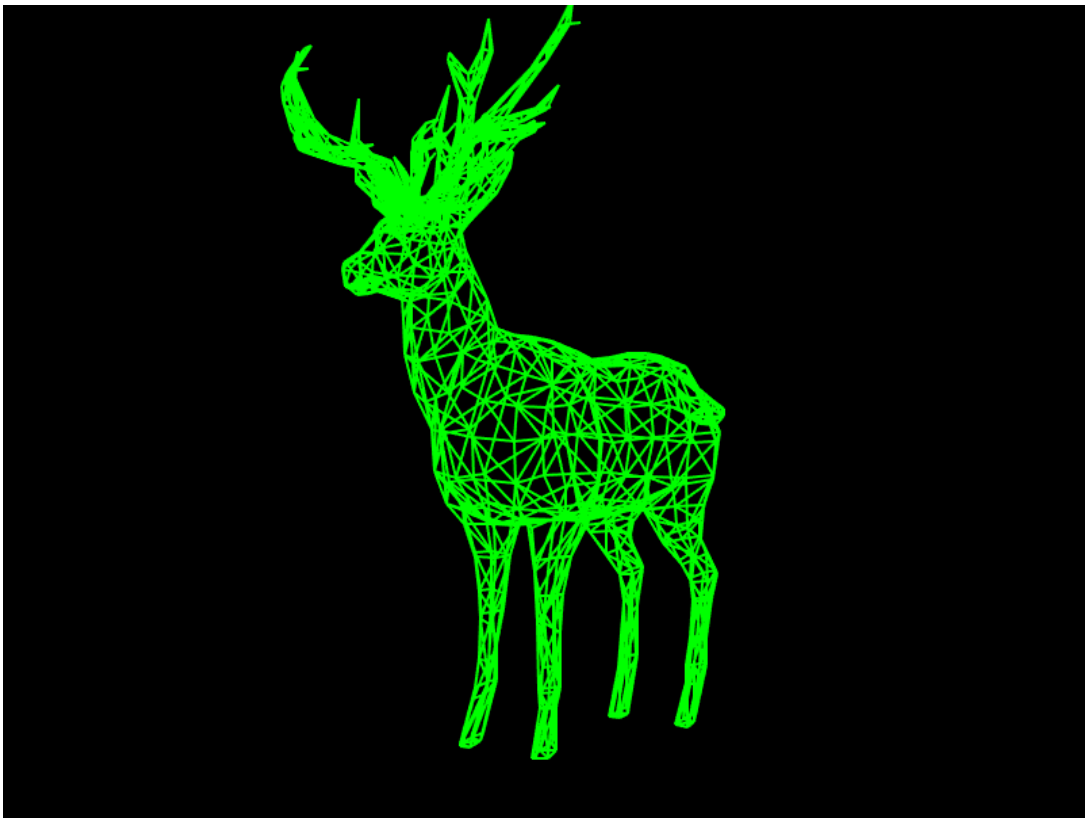
Graphics Mesh

If you've been following the Dev Logs for a while you may remember ages ago when we introduced the new Graphics object and I show-cased doing some fast line-drawing in it by spinning a few 3D shapes around. At the time this was done with a rudimentary object parser and some really early math code from Babylon.js :) But that didn't matter because all it was showing off was the speed of line drawing in the Graphics API.

The demo was finished, it worked and we moved on to other tasks. Recently though I revisited it as part of a different bit of work and figured it could be improved considerably. Taking the code I'd already built I refactored it, vastly updated the Wavefront OBJ parser, created a new Mesh geometry object and built it into the Graphics object. It means you can now natively load a model into v3 and it'll parse all of the mesh and texture data out, and then add it to a Graphics object. Here's an example:

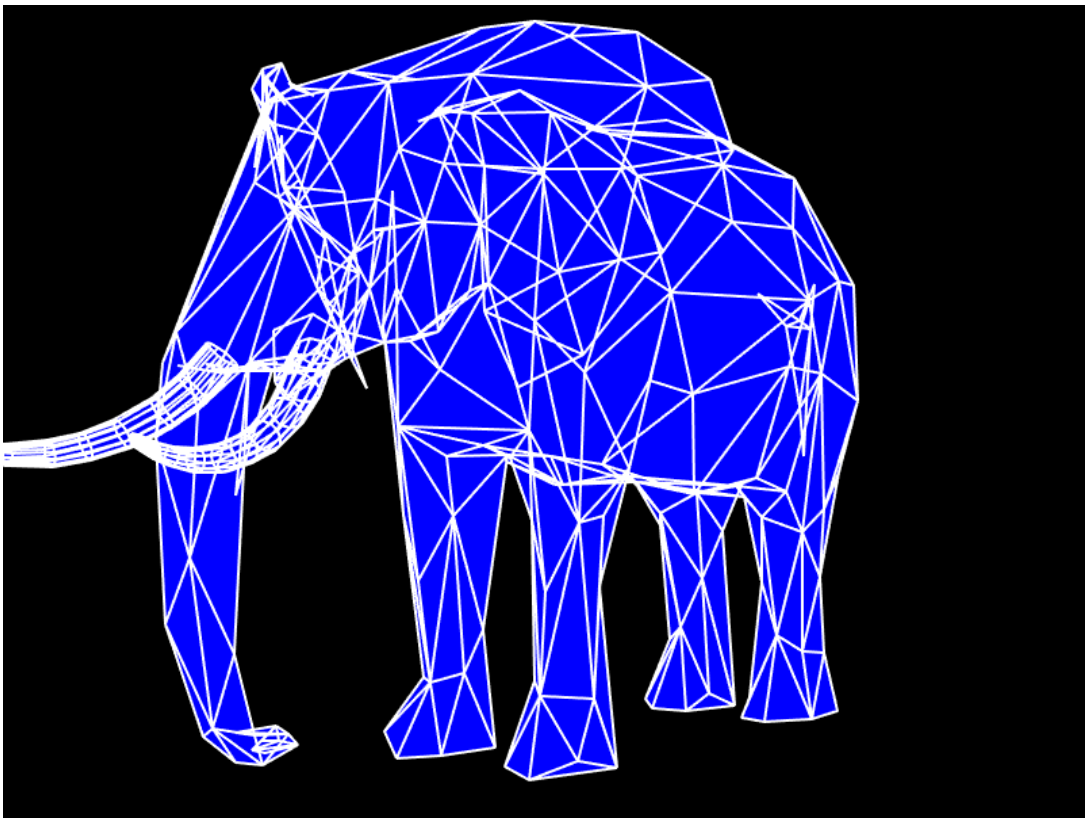
```
function preload ()
{
    this.load.obj('deer', 'assets/obj/deer.obj');
}

function create ()
{
    graphics = this.add.graphics(0, 0);

    mesh = graphics.createMesh('deer', 0, 700, 4000);

    mesh.rotation.x = Phaser.Math.DegToRad(180);

    mesh.thickness = 2;
}

function update ()
{
    mesh.rotation.y += 0.01;

    graphics.clear();

    graphics.strokeMesh(mesh);
}
```

As you can see we're loading a model, creating a mesh which we can influence (via rotation, position, scale, color, etc) and drawing it. This will work in Canvas or WebGL. This is the result:



Oh deer, oh deer!

Pretty neat for such a small amount of code, right? While I was at it I implemented backface culling. I also wanted to be able to fill the shape with a solid color, which leads to me finding a bug and missing feature in Graphics, and soon both were done:



Backface cull and fill

If you mix this in with the new Glow shaders, it can have some interesting results ... :)

It's nuts in space!

While I appreciate this has somewhat limited uses it was fun to work on. And importantly the code mostly already existed, is nicely self-contained so is easily excluded from custom builds and opens up the opportunity to create some real retro styled wireframe vector stuff should you wish.

And that is what Phaser 3 is really all about: the ability to explore and create. Even if that means making glowing squirrels in space.

Most of us have the next week or two off. Family commitments, events and the holiday period eat into the available time, *as it should*, and honestly it's good to step away and recharge your mental batteries. I'll be spending a few days going through the v3 issues and tidying things up. We're now just a single system (the Scale Manager) and finished documentation away from a feature complete 3.0 release.

January is going to be an exciting and very busy month for sure! :)

Phaser 3 Labs

Phaser 3 Beta 14 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!

The Game Chasers is my latest YouTube addiction :) Filmed in the style of

shows like Storm Hunters it follows a group of guys who tour around flea markets and garage sales hunting down retro games. Strangely more fascinating than it sounds to watch, mostly because the hosts are so entertaining!

Here is an impressively low-level article that we hope gives you a good idea about what happens in V8 when it comes to optimization.

Nvidia has two Star Wars-themed graphics cards on the way and they look pretty sweet!

Phaser Releases

Phaser CE 2.9.4 released December 20th 2017.
Phaser 3 Beta 14 released December 21st 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.