

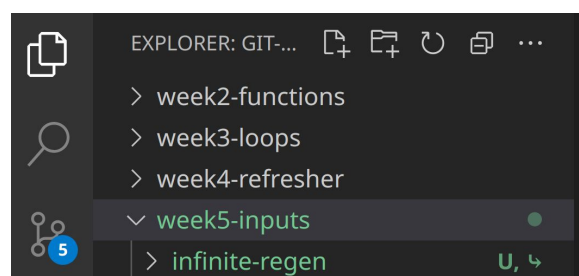
Infinite Regeneration

A JavaScript example operating on the Document Object Model (see https://www.w3schools.com/js/js_htmlDOM.asp). Incorporates randomness, mouse events, and a form element. The title 'Infinite Regeneration' refers to the content of the example - which comprises: an image of a council estate under threat of demolition; references to sham local government consultation (data-gathering) exercises; and word cloud that points to how such exercises can break up and displace predominantly working class communities. A working example can be found at: <http://resources.theanthillsocial.co.uk/experiments/infinite-regen/index.html>



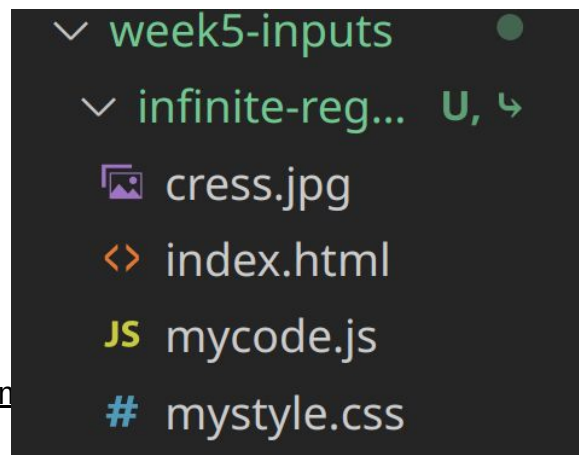
Setup

Open VS Code and create a folder for this weeks work. I've created a folder called **week5-inputs** and in that, another called **infinite-regen** which is the title I gave to the example I created.



Create the file structure

This example combines separate a CSS, JavaScript, and JPG file into an HTML file called index.html. First create empty **mystyle.css**, **index.html**, and **mycode.js** files. Second download the image file from:
<http://resources.theanthillsocial.co.uk/experiments/regen/cress.jpg>



Write index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Infinite Regeneration</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="mystyle.css">
</head>
<body>

  <input type="submit" id="tracker">
  <div id="home">    </div>

  <script src="mycode.js"></script>

</body>
</html>
```

Write mystyle.css

```
html, body {
  height: 100%;
}
body { /* This centers our drawing. */
  margin: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  background: rgb(60,63,60);
}
#home{ /* The HTML id of the drawing. */
```

```

    position:relative;
    height:450px;
    width:800px;
    background-image: url('cress.jpg');
    background-size: 100%;
}
.word{ /* Default style of each word & allows them to be assigned x/y coords. */
    position:absolute;
    background:rgba(200, 200, 200, 0.4);
    width:200px;
    text-align:center;
    font-family: Courier;
    height:20px;
    font-size:14px;
    line-height:20px;
    z-index:1;
    color:rgb(60,63,60);
}
input{ /* Ensures submit remains above all other tags and can move anywhere. */
    position:fixed;
    z-index:2;
    padding:15px;
    width:100px;
    text-transform: uppercase;
}

```

Lets code mycode.js

Global variables

Setup global variables which are available to every function.

```

// Setup global variables
const home = document.getElementById("home");
const submit = document.getElementById("tracker");
submit.setAttribute("onclick", "createShamConsultation(event)");
document.body.setAttribute("onmousemove", "moveMe(event)");
const words = [
    "consultation",
    "financial-viability",
    "displacement",
    "killing-community",
    "social-cleansing",
    "governance",
    "demolition",
    "destroyed homes",
    "for-profit",
    "affordability",
    "party-politics"
];
let intensity = 0;

```

Create div tag

This function automatically generates an HTML tag. Can you see the intentional error? :

```
/* Function to create an HTML tag at
   x/y coordinates relative to #home */
function createTag(word, x, y, d){
  let tag = document.createElement("div");
  tag.innerHTML = word;
  tag.setAttribute("class", "word");
  tag.style.left = x+'px';
  tag.style.top = y+'px';
  tag.style.transform = "rotate("+d+"deg)";
  home.appendChild(tag);
}
```

The HTML tags it generates look like this:

```
<div class="word" style="left:16px;top:452px; transform:rotate(0deg);">consultation</div>
```

Loop through words

This function:

1. Loops through our word array.
2. Generates random x/y coordinates.
3. Generates a random rotation of the body (between -4 and 4 degrees).
4. Keeps repeating (its a recursive function) on a timer until 500 words have been generated.

```
// Loops through words array
function createShamConsultation(){
  let x, y;
  let body = document.body;
  for(word=0; word<words.length; word++){
    x = createRandom(-100,700);
    y = createRandom(-50,500);
    d = createRandom(-4,4);
    createTag(words[word], x, y, d);
    intensity++; // # Num of words
  }
  body.style.transform = "rotate("+d+"deg)";
  console.log('generated words');
```

```

// Keep generating until 500 words
setTimeout(function(){
  if(intensity>=500){
    home.innerHTML = "";
    intensity=0;
  }else{
    createShamConsultation();
  }
}, 50); // Timer
}

```

Can you work out what this conditional is doing?

```

if(intensity>=500){

```

Follow mouse

The `moveMe()` function is triggered by the 'onmousemove' attribute of the body tag and the `createShamConsultation()` function is triggered by the 'onclick' attribute of the input tag:

```

function moveMe(event){
  submit.style.left=(event.pageX-50)+'px';
  submit.style.top=(event.pageY-30)+'px';
}

```

Generate number

This function generates a random number within a range.

```

function createRandom(from, to){
  let r1 = [0.0,1.0];
  let r2 = [from,to];
  let rand = Math.random();
  return Math.floor((rand-r1[0])*(r2[1]-r2[0])/(r1[1]-r1[0])+r2[0]);
}

```

Finally

A handy command at the end of the script - if this prints "all OK" to the console then we know there are no JavaScript errors.

```
console.log('all OK');
```

For the complete **mycode.js** script take a look at:

<http://resources.theanthillsocial.co.uk/experiments/infinite-regen/mycode.js>

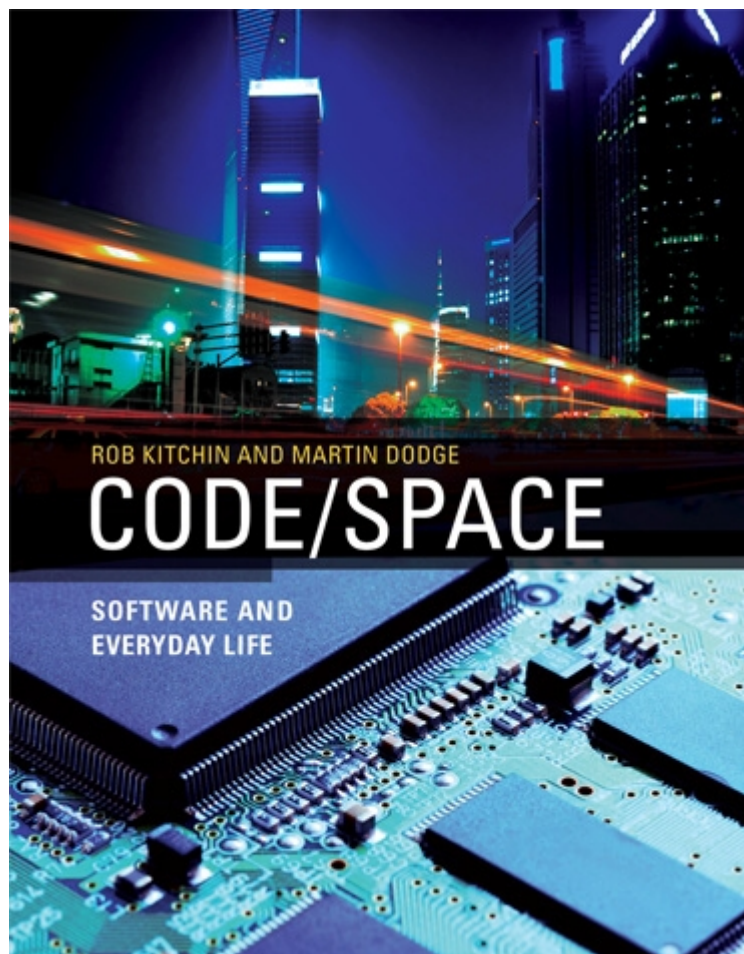
And for the fully working example see:

<http://resources.theanthillsocial.co.uk/experiments/infinite-regen>

References

Code/Space: Software and Everyday Life. 2011. Rob Kitchin, Martin Dodge. MIT Press.

“An analysis of the ways that software creates new spatialities in everyday life, from supermarket checkout lines to airline flight paths”



Wirelessness. 2010. Adrian Mackenzie. MIT Press.

“matching the disordered flow of wireless networks, meshes, patches, and connections with felt sensations”.



JavaScript Perlin Noise Library

Terrain demo:

<https://joeiddon.github.io/zengine/demos/terrain.html>

The library:

<https://joeiddon.github.io/projects/javascript/perlin.html>

Colour demon:

<https://joeiddon.github.io/perlin/demo>

