SOLVERA
crafting results

paradigm
CONSULTING GROUP

DEVFACTO

GasBuddy

SASKATCHEWAN
POLYTECHNIC

The Weirdly Practical World of Paint Worklets

Prairie Dev Con Regina — Jeremy Wagner — jeremy.codes — @malchata

What the hell are paint worklets?

✨ Houdini! ✨

(seriously!)

OK though, but what is a worklet?

JavaScript
JavaScript

```
.masthead {
  background-image: paint(amazing);
}
```

```
.masthead {
  ~~background-image: paint(amazing);~~
}
```

```
// Are paint worklets supported?
if (window.CSS.paintWorklet) {
  // If so, register that worklet!
  CSS.paintWorklet.addModule("/amazing-worklet.js").then(() => {
    console.log("Worklet registered!");
  }).catch(error => {
    console.error(error);
  });
}
```

```
// One of those fancy ES6 classes, eh?
class Amazing {
  constructor () {
    // Perform constructor-type duties here
  }

  paint (ctx, geom) {
    // Draw stuff in here
  }
}

// Make the paint worklet available to CSS
registerPaint("amazing", Amazing);
```

4Runner

```
// Make the paint worklet available to CSS
registerPaint("amazing", Amazing);
```

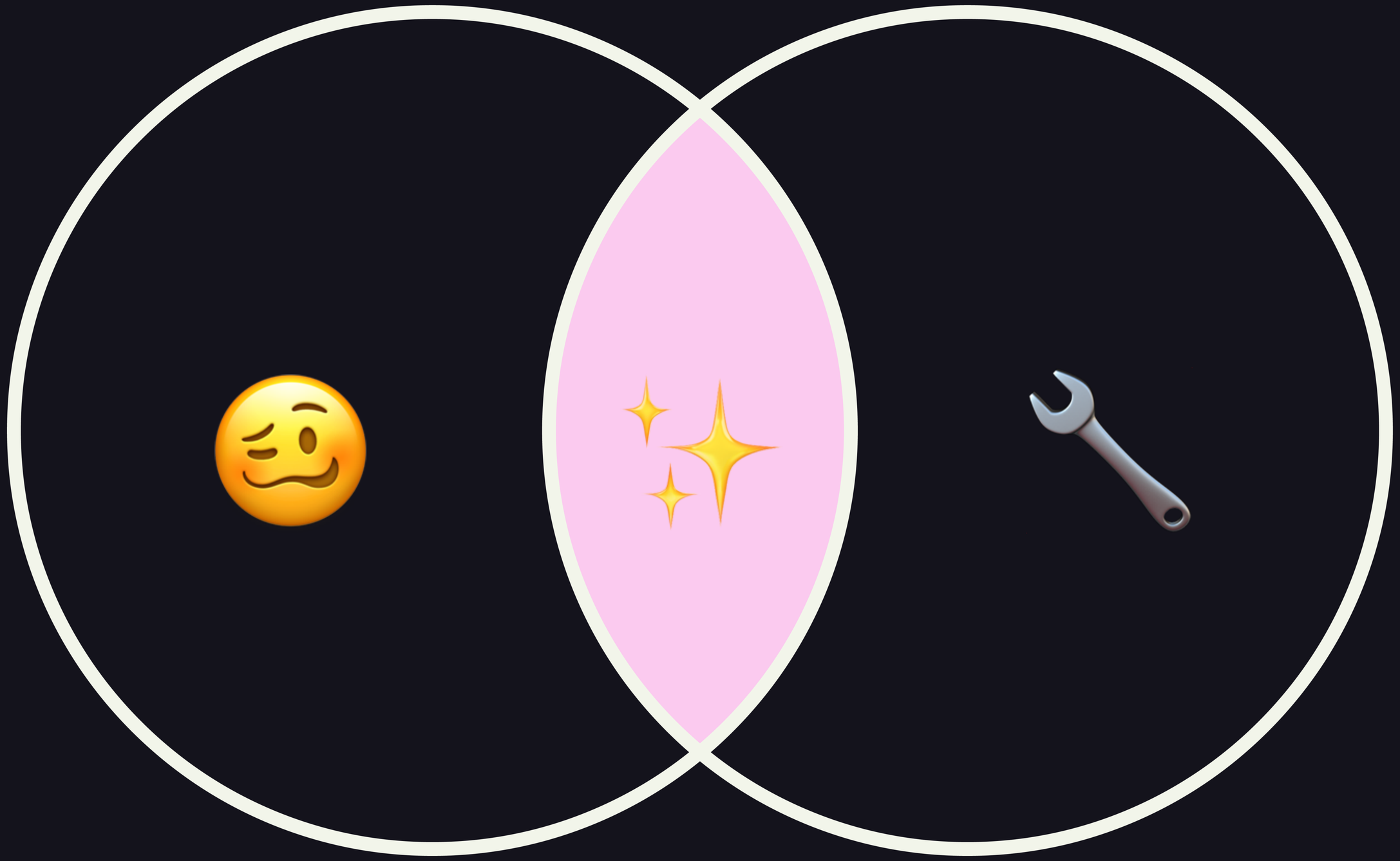
```
paint (ctx, geom) {
  //    ^^^ looks familiar!
}
```

```
paint (ctx, geom) {
  const xPosition = geom.width / 2;
  const yPosition = geom.height / 2;
  const radius = 64;
  const startAngle = 0;
  const endAngle = Math.PI * 2;

  ctx.arc(xPosition, yPosition, radius, startAngle, endAngle);
}
```

```
.masthead {
  background-image: paint(amazing);
}
```

UHHH
UHHHH
UHHH
UHHH
MAKES MOVING EASIER
BEST
Ride Van
LOWEST COST

Cage me again



Tim Holman—
Generative Art Speedrun

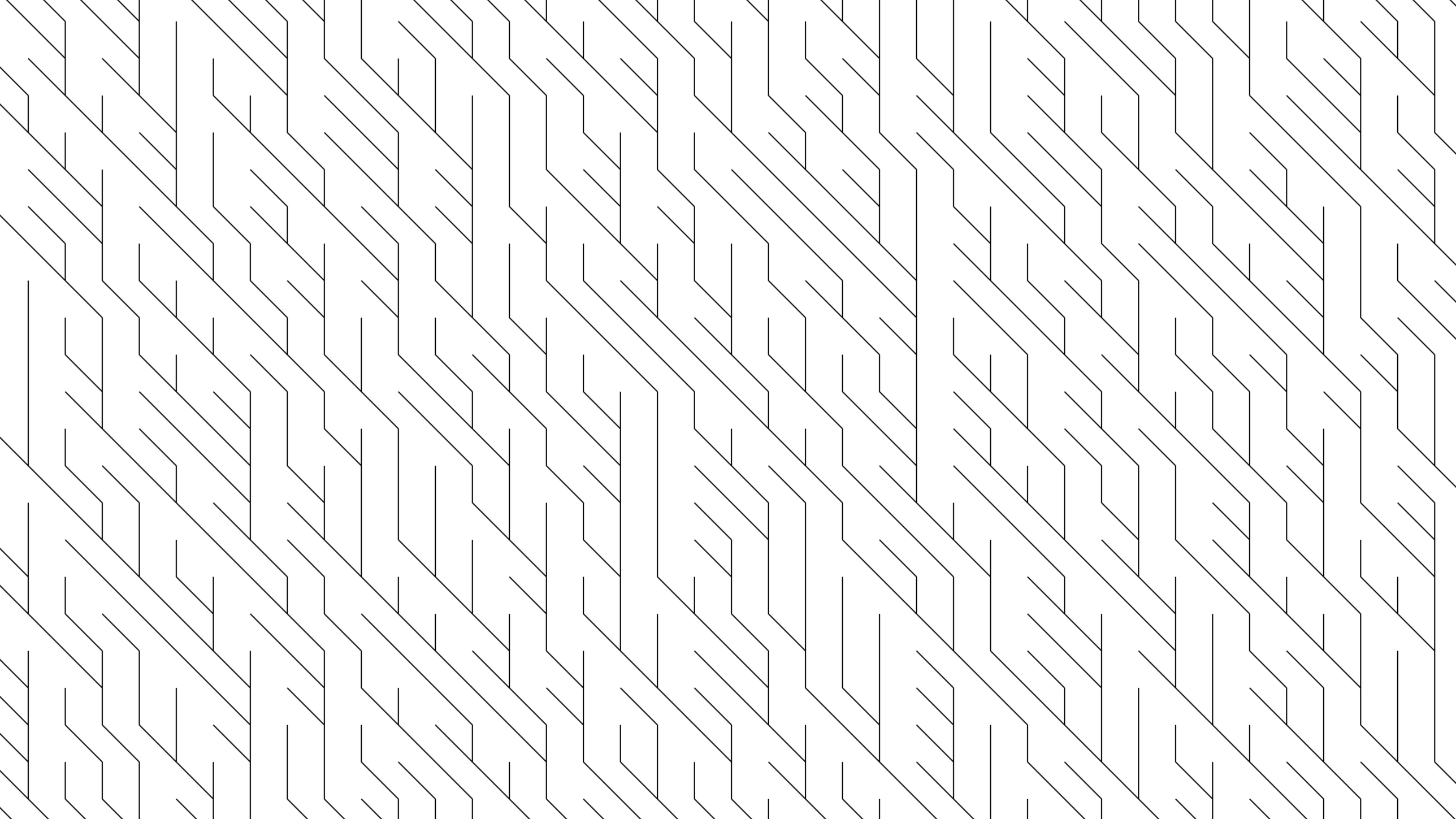
tinyurl.com/timstiles

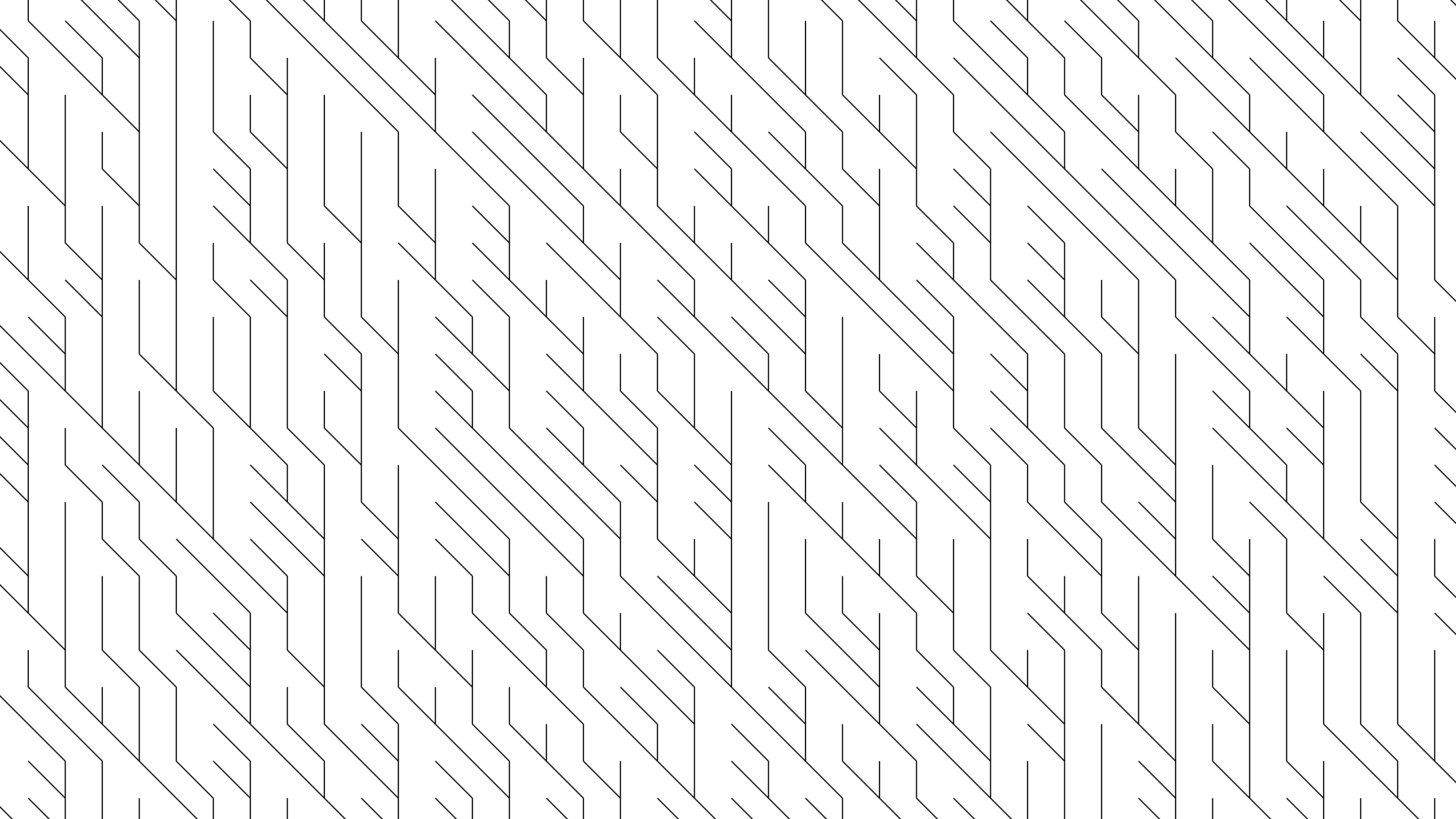
```
paint (ctx, geom) {
  const tileSize = 32;
  const xTiles = geom.width / tileSize;
  const yTiles = geom.height / tileSize;

  for (let y = 0; y < yTiles; y++) {
    const yOffset = y * tileSize;

    for (let x = 0; x < xTiles; x++) {
      const xOffset = x * tileSize;

      ctx.beginPath();
      ctx.moveTo(xOffset, yOffset);

      if (Math.random() >= 0.5) {
        ctx.lineTo(xOffset + tileSize, yOffset + tileSize);
      } else {
        ctx.lineTo(xOffset, yOffset + tileSize);
      }

      ctx.stroke();
    }
  }
}
```

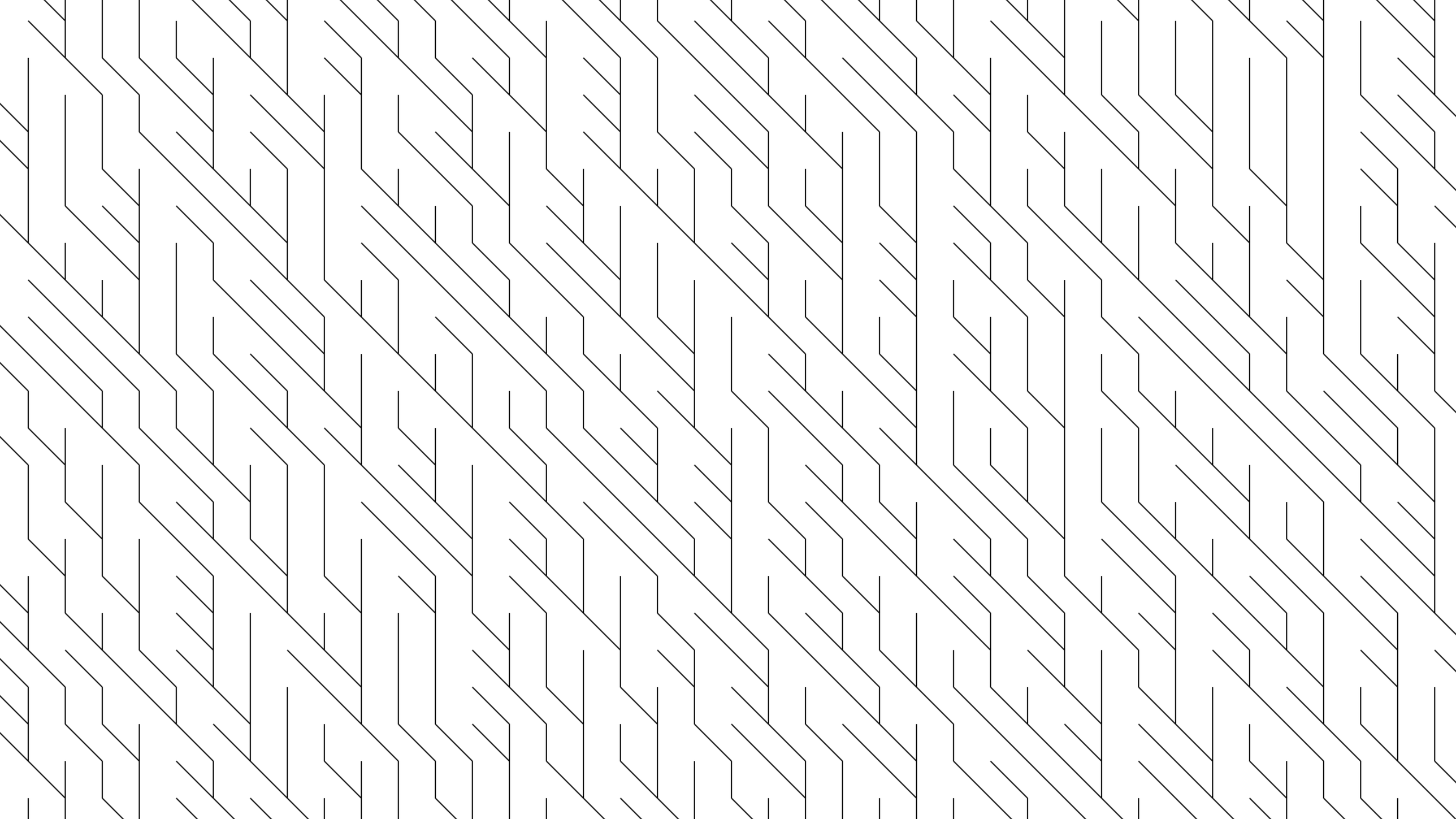
```
paint (ctx, geom) {
  const tileSize = 32;
  const xTiles = geom.width / tileSize;
  const yTiles = geom.height / tileSize;

  // ...
}
```

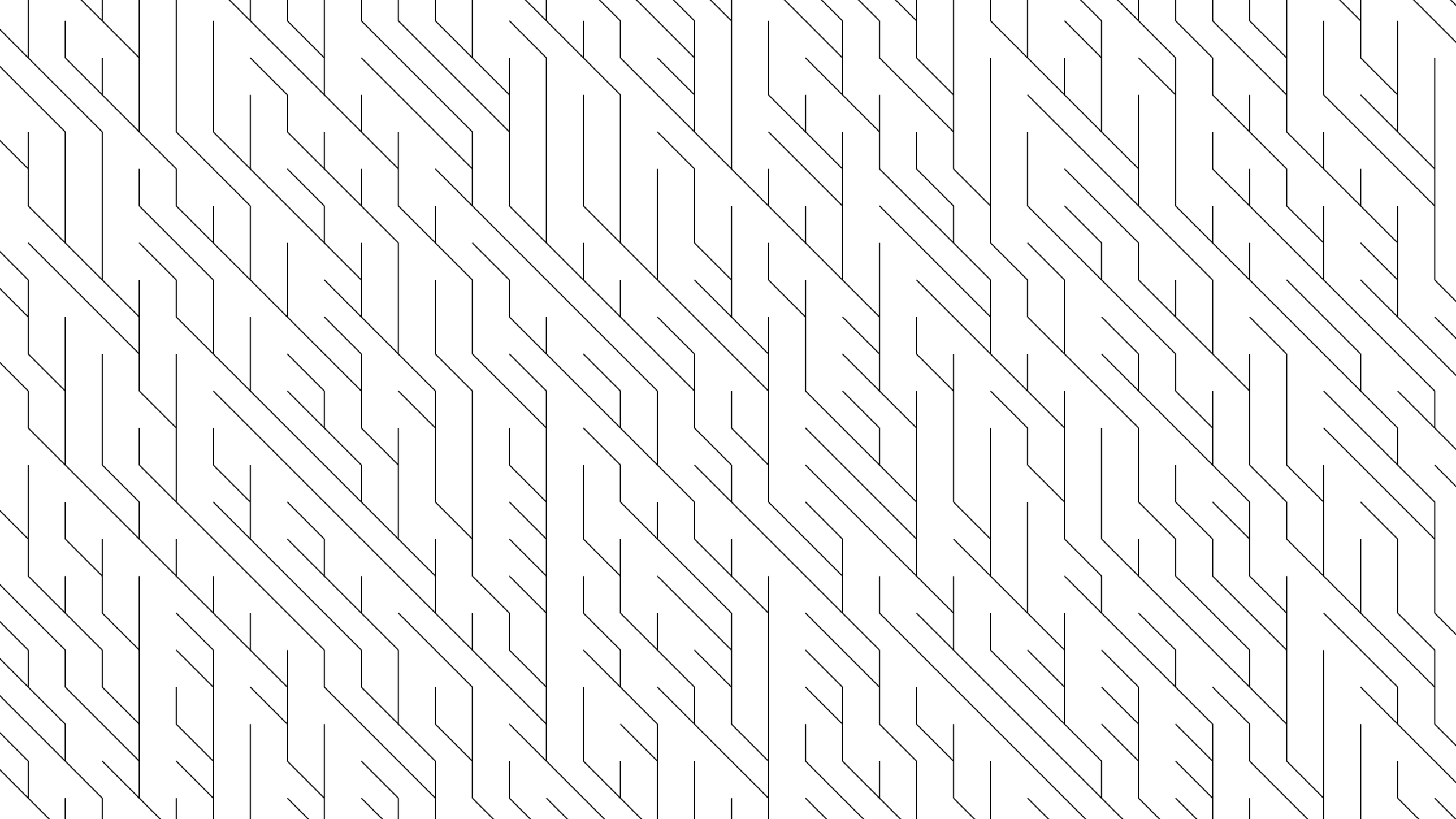
```
paint (ctx, geom) {
  // ...

  for (let y = 0; y < yTiles; y++) {
    const yOffset = y * tileSize;

    for (let x = 0; x < xTiles; x++) {
      const xOffset = x * tileSize;

      // Draw stuff here...
    }
  }
}
```

```
paint (ctx, geom) {
  // ...

  for (let y = 0; y < yTiles; y++) {
    const yOffset = y * tileSize;

    for (let x = 0; x < xTiles; x++) {
      const xOffset = x * tileSize;

      ctx.beginPath();
      ctx.moveTo(xOffset, yOffset);

      if (Math.random() >= 0.5) {
        ctx.lineTo(xOffset + tileSize, yOffset + tileSize);
      } else {
        ctx.lineTo(xOffset, yOffset + tileSize);
      }

      ctx.stroke();
    }
  }
}
```









```
class Blotto {
  paint (ctx, geom) {
    const tileSize = 8;
    const amplitude = .5;
    const geomTileHeight = geom.height / tileSize;
    const geomTileWidth = geom.width / tileSize;
    const fullCircle = Math.PI * 2;

    ctx.fillStyle = "#6369d1";

    for (let y = 0; y < geomTileHeight; y++) {
      const yOffset = y * tileSize;

      for (let x = 0; x < geomTileWidth; x++) {
        const xOffset = x * tileSize;
        const alpha = Math.random() % Math.random();
        const radius = tileSize * Math.random() * amplitude;

        ctx.globalAlpha = alpha;
        ctx.beginPath();
        ctx.arc(xOffset, yOffset, radius, 0, fullCircle);
        ctx.fill();
      }
    }
  }
}

registerPaint("blotto", Blotto);
```

```
paint (ctx, geom) {
  const tileSize = 8;
  const amplitude = .5;
  const geomTileHeight = geom.height / tileSize;
  const geomTileWidth = geom.width / tileSize;
  const fullCircle = Math.PI * 2;

  ctx.fillStyle = "#6369d1";

  // ...
}
```

```
paint (ctx, geom) {
  // ...

  for (let y = 0; y < geomTileHeight; y++) {
    const yOffset = y * tileSize;

    for (let x = 0; x < geomTileWidth; x++) {
      const xOffset = x * tileSize;
      const alpha = Math.random() % Math.random();
      const radius = tileSize * Math.random() * amplitude;

      ctx.globalAlpha = alpha;
      ctx.beginPath();
      ctx.arc(xOffset, yOffset, radius, 0, fullCircle);
      ctx.fill();
    }
  }
}
```

```
registerPaint("blotto", Blotto);
```

```
.some-element {
  background-image: paint(blotto);
  background-color: #fffbfe;
}
```

```
class Flashy {
  constructor () {
    this.degToRad = Math.PI / 180;
  }

  paint (ctx, geom) {
    const radius = 48;
    const rayWidth = 1.25;
    const outerRadius = geom.width *3;
    const x = geom.width * 0.25;
    const y = geom.height * 0.1875;

    ctx.fillStyle = "#fffbfe";
    ctx.beginPath();
    ctx.arc(x, y, radius, 0, Math.PI * 2, false);
    ctx.fill();

    for (let i = 0; i <= 360; i++) {
      if (Math.random() >= 0.75) {
        const radiansEdge1 = this.degToRad * (i - rayWidth);
        const radiansEdge2 = this.degToRad * (i + rayWidth);

        ctx.beginPath();
        ctx.moveTo(x, y);
        ctx.lineTo((Math.cos(radiansEdge1) * outerRadius), (Math.sin(radiansEdge1) * outerRadius));
        ctx.lineTo((Math.cos(radiansEdge2) * outerRadius), (Math.sin(radiansEdge2) * outerRadius));
        ctx.lineTo(x, y);
        ctx.fill();
      }
    }
  }
}

registerPaint("flashy", Flashy);
```

```
paint (ctx, geom) {
  const radius = 48;
  const rayWidth = 1.25;
  const outerRadius = geom.width * 3;
  const x = geom.width * 0.25;
  const y = geom.height * 0.1875;
  const fullCircle = Math.PI * 2;

  // ...
}
```

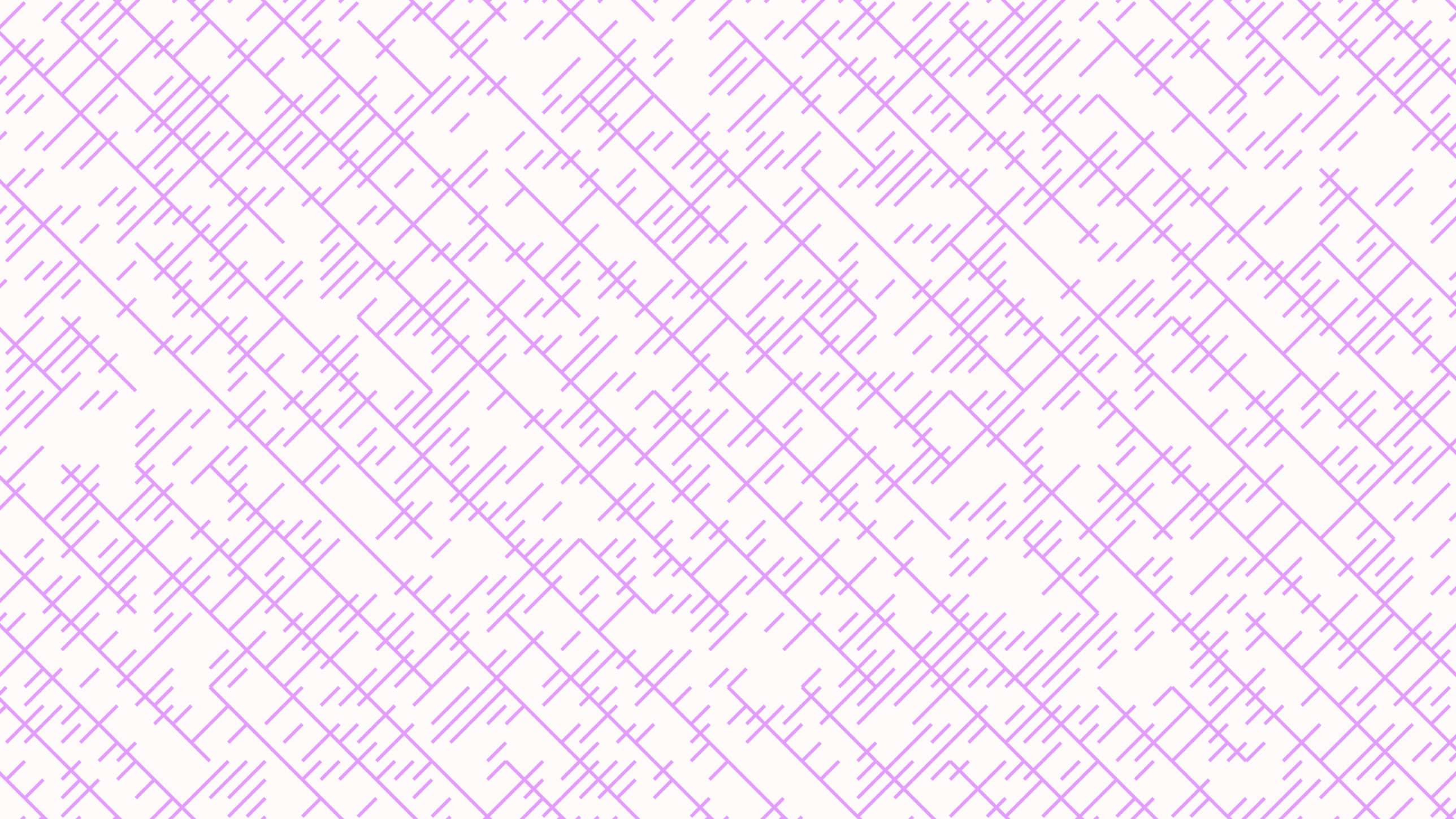
```
paint (ctx, geom) {
  // ...

  ctx.fillStyle = "#fffbfe";
  ctx.beginPath();
  ctx.arc(x, y, radius, 0, fullCircle);
  ctx.fill();

  // ...
}
```

```
paint (ctx, geom) {
  // ...

  for (let i = 0; i <= 360; i++) {
    if (Math.random() >= 0.75) {
      const radiansEdge1 = this.degToRad * (i - rayWidth);
      const radiansEdge2 = this.degToRad * (i + rayWidth);

      ctx.beginPath();
      ctx.moveTo(x, y);
      ctx.lineTo((Math.cos(radiansEdge1) * outerRadius), (Math.sin(radiansEdge1) * outerRadius));
      ctx.lineTo((Math.cos(radiansEdge2) * outerRadius), (Math.sin(radiansEdge2) * outerRadius));
      ctx.lineTo(x, y);
      ctx.fill();
    }
  }
}
```

```
registerPaint("flashy", Flashy);
```

```
.some-element {
  background-image: paint(flashy);
  background-color: #e59eff;
}
```

h
r
tan (θ)
10
5
Blind
V = πr²h
ax² + bx + c = 0
a(x² + b/a x + c/a) = 0
x² + 2 b/2a x + (b/2a)² − (b/2a)² +
(x + b/2a)² − (b² − 4ac)/4a² = 0



Customize!

✨ Custom properties! ✨

(seriously!)

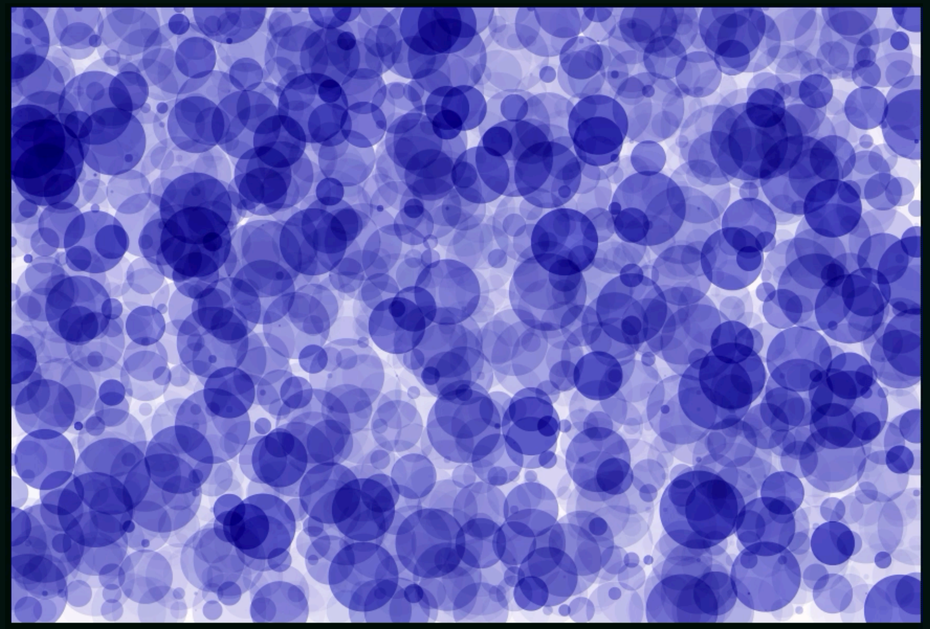
```
.some-element {
  background-image: paint(flashy);
}
```

```
.some-element {
  background-image: paint(flashy);
  --flashy-radius: 48;
}
```

```
if ("registerProperty" in CSS) {
  CSS.registerProperty({
    name: "--flashy-radius",
    syntax: "<integer>",
    inherits: false,
    initialValue: 48
  });
}
```

```
static get inputProperties () {
  return [
    "--flashy-radius"
  ];
}
```

```
paint (ctx, geom, properties) {
  const radius = parseInt(properties.get("--flashy-radius"));

  // ...
}
```



--blotto-tile-size:

8

--blotto-color:

#6060d1

--blotto-amplitude:

2.375

--blotto-max-opacity:

1.0

--blotto-blend-mode:

multiply

Resiliency!

Is Houdini ready yet?

	Google Chrome	Mozilla Firefox	Opera	Microsoft Edge	Samsung Internet	Apple Safari	Spec
Layout API (Explainer Demos)	Partial support (Canary) Details	Intent to implement Details	No signal	No signal	No signal	No signal	First Public Working Draft Spec
Paint API (Explainer Demos Article)	Shipped (Chrome 65) Details	Intent to implement (Servo) Details	Shipped (Opera 52) Details	No signal	Shipped (Internet 9.2) Details	In Development Details	Candidate Recommendation Spec
Parser API (Explainer)	No signal	No signal	No signal	No signal	No signal	No signal	No spec
Properties & Values API (Demos)	Partial support (Canary) Details	In Development Details	No signal	No signal	No signal	Partial support (Safari TP 67) Details	Working Draft Spec
AnimationWorklet (Explainer Demos Article)	Partial support (Canary) Details	No signal	No signal	No signal	No signal	No signal	First Public Working Draft Spec
Typed OM (Explainer Article)	Shipped (Chrome 66) Details	Intent to implement Details	Shipped (Opera 53) Details	No signal	Shipped (Internet 9.2) Details	In Development Details	Working Draft Spec
Font Metrics API (Explainer)	No signal	No signal	No signal	No signal	No signal	No signal	No spec

Maintained by **Surma**, source code & license in the **repository** on GitHub, last updated on Mon Jun 03 2019 22:47:49 GMT+0100 (British Summer Time)

```
.masthead {
  background-image: paint(flashy);
}
```

```
.masthead {
  ~~background-image: url("fallback.svg");~~
  background-image: paint(flashy);
}
```

```
.masthead {
  background-image: url("fallback.svg");
  background-image: paint(flashy);
}
```

Hi, I'm Jeremy.

I'm a web developer, writer, and speaker. I work to make the web faster for everyone, everywhere.

Hi, I'm Jeremy.

I'm a web developer, writer, and speaker. I work to make the web faster for everyone, everywhere.

```
if (window.CSS.paintWorklet) {
  CSS.paintWorklet.addModule("/amazing-worklet.js").then(() => {
    console.log("Worklet registered!");
  }).catch(error => {
    document.body.classList.add("worklet-unavailable");
  });
}
```

```
.masthead {
  ~~background-image: url("fallback.svg");~~
  background-image: paint(flashy);
}

.masthead.worklet-unavailable {
  background-image: url("fallback.svg");
}
```

Performance! ⚡

INTERNET
BROWSING
INTERNET B

Babel

+

paint worklets

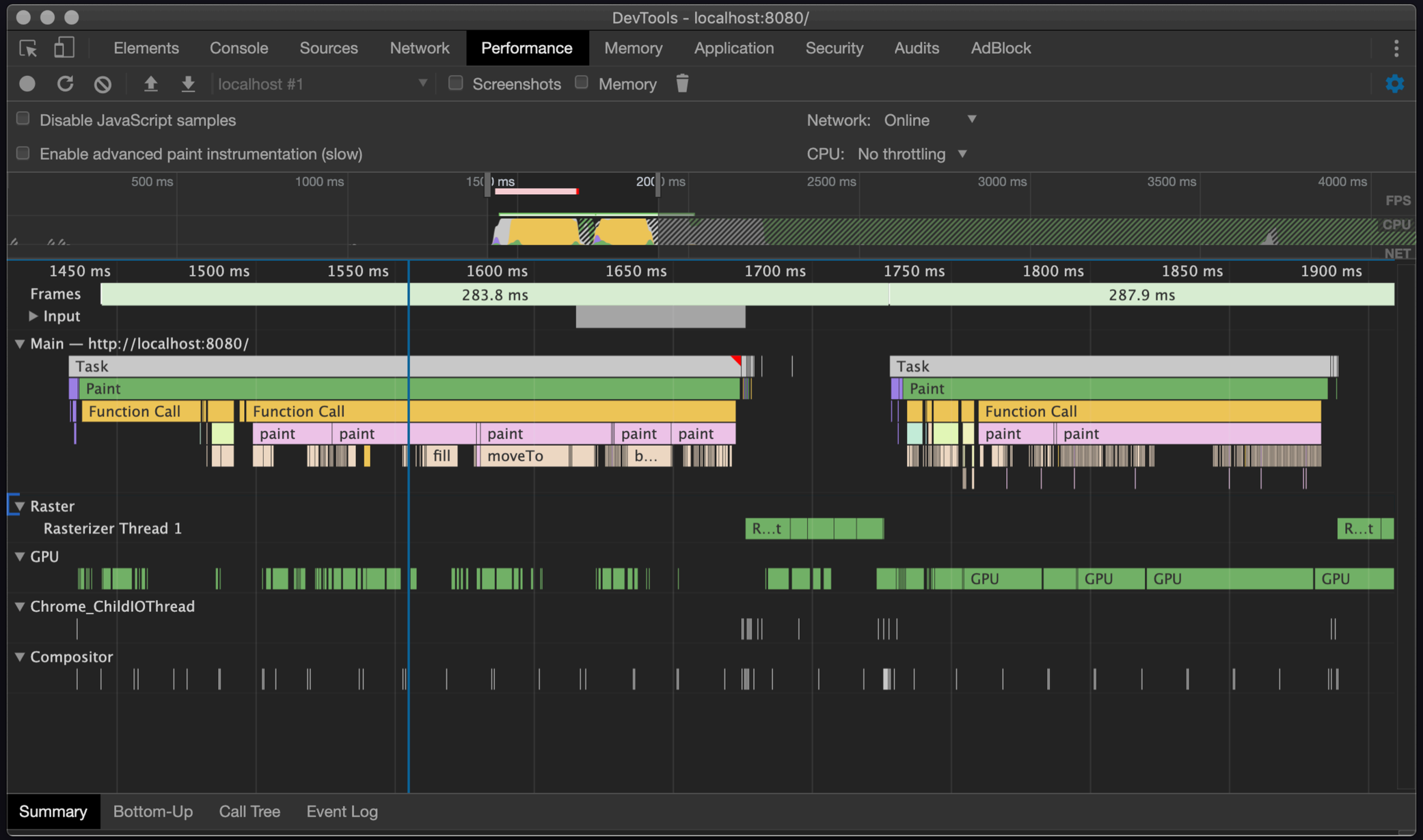
=

no-nos

```
class Amazing {
  paint (ctx, geom) {
    const tileSize = 32;
    const xTiles = geom.width / tileSize;
    const yTiles = geom.height / tileSize;

    for (let y = 0; y < yTiles; y++) {
      const yOffset = y * tileSize;

      for (let x = 0; x < xTiles; x++) {
        const xOffset = x * tileSize;

        ctx.beginPath();
        ctx.moveTo(xOffset, yOffset);

        if (Math.random() >= 0.5) {
          ctx.lineTo(xOffset + tileSize, yOffset + tileSize);
        } else {
          ctx.lineTo(xOffset, yOffset + tileSize);
        }

        ctx.stroke();
      }
    }
  }
}

registerPaint("amazing", Amazing);
```

```
function _classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor)) { throw new TypeError("Cannot call a class as a function"); } }

function _defineProperties(target, props) { for (var i = 0; i < props.length; i++) { var descriptor = props[i]; descriptor.enumerable = descriptor.enumerable ||
false; descriptor.configurable = true; if ("value" in descriptor) descriptor.writable = true; Object.defineProperty(target, descriptor.key, descriptor); } }

function _createClass(Constructor, protoProps, staticProps) { if (protoProps) _defineProperties(Constructor.prototype, protoProps); if (staticProps)
_defineProperties(Constructor, staticProps); return Constructor; }

var Amazing =
/*#__PURE__*/
function () {
  "use strict";

  function Amazing() {
    _classCallCheck(this, Amazing);
  }

  _createClass(Amazing, [{
    key: "paint",
    value: function paint(ctx, geom) {
      var tileSize = 32;
      var xTiles = geom.width / tileSize;
      var yTiles = geom.height / tileSize;

      for (var y = 0; y < yTiles; y++) {
        var yOffset = y * tileSize;

        for (var x = 0; x < xTiles; x++) {
          var xOffset = x * tileSize;
          ctx.beginPath();
          ctx.moveTo(xOffset, yOffset);

          if (Math.random() >= 0.5) {
            ctx.lineTo(xOffset + tileSize, yOffset + tileSize);
          } else {
            ctx.lineTo(xOffset, yOffset + tileSize);
          }

          ctx.stroke();
        }
      }
    }
  }]);

  return Amazing;
}();

registerPaint("amazing", Amazing);
```

Be <u>considerate</u>!


DevTools - localhost:8080/
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Audits
AdBlock
localhost #1
Screenshots
Memory
Disable JavaScript samples
Enable advanced paint instrumentation (slow)
Network: Online
CPU: No throttling
500 ms
1000 ms
2500 ms
3000 ms
3500 ms
4000 ms
FPS
CPU
NET
1450 ms
1500 ms
1550 ms
1600 ms
1650 ms
1700 ms
1750 ms
1800 ms
1850 ms
1900 ms
Frames
283.8 ms
287.9 ms
Input
Main — http://localhost:8080/
Task
Paint
Function Call
paint
fill
moveTo
b...
Raster
Rasterizer Thread 1
R...t
GPU
Chrome_ChildIOThread
Compositor
Summary
Bottom-Up
Call Tree
Event Log

```
// Check for Device Memory API support.
// Default to 8 GiB if the API is absent.
const deviceMemory = navigator.deviceMemory || 8;

// Only register a paint worklet if the
// CSS Paint API is available and the memory
// on the device is at least 4 GiB.
if (window.CSS.paintWorklet && deviceMemory >= 4) {
  CSS.paintWorklet.addModule("/js/blotto.js");
} else {
  document.body.classList.add("no-paint-worklets");
}
```

parallelowow.js

```
43          for (let y = -1; y < geomTilesY; y++) {
44            const yOffset = y * tileHeight;
45
46            for (let x = -1; x < (geomTilesX + y); x++) {
47              const xOffset = (x * tileWidth) - (y * tileHeight);
48
49  0.1 ms      if (Math.random() > probability) {
50                // Helpers!
51                const upperLeftX = xOffset;
52                const upperLeftY = yOffset;
53                const upperRightX = xOffset + tileWidth;
54                const upperRightY = yOffset;
55                const lowerRightX = xOffset + (tileWidth - tileHeight);
56                const lowerRightY = yOffset + tileHeight;
57                const lowerLeftX = xOffset - tileHeight;
58                const lowerLeftY = lowerRightY;
59
60                // 1. Draw shape on the right side of the parallelogram
61  2.3 ms        ctx.fillStyle = colors[1];
62                ctx.beginPath();
63                ctx.moveTo(upperRightX, upperRightY);
64                ctx.lineTo((Math.cos(this.radians) * outerRadius), (Math.sin(this.radians) * outerRadius));
65                ctx.lineTo(lowerRightX, lowerRightY);
66  0.1 ms        ctx.lineTo(upperRightX, upperRightY);
67  0.1 ms        ctx.fill();
68
69                if (strokeWeight > 0) {
70                  ctx.stroke();
71                }
72
73                // 2. Draw shape on the lower left side of the parallelogram
74  0.2 ms        ctx.fillStyle = colors[2];
75                ctx.beginPath();
76                ctx.moveTo(lowerRightX, lowerRightY);
77  0.2 ms        ctx.lineTo((Math.cos(this.radians) * outerRadius), (Math.sin(this.radians) * outerRadius));
78                ctx.lineTo(lowerLeftX, lowerLeftY);
79                ctx.moveTo(lowerLeftX, lowerLeftY);
80                ctx.fill();
81
82                if (strokeWeight > 0) {
83                  ctx.stroke();
84                }
85
```

Wrapping up

Una Kravets Follow
http://una.im :) front-end developer & brunch enthusiast

New horizons in CSS: Houdini and the Paint API

January 9, 2019 · 7 min read

tinyurl.com/unapaints

Cssconf

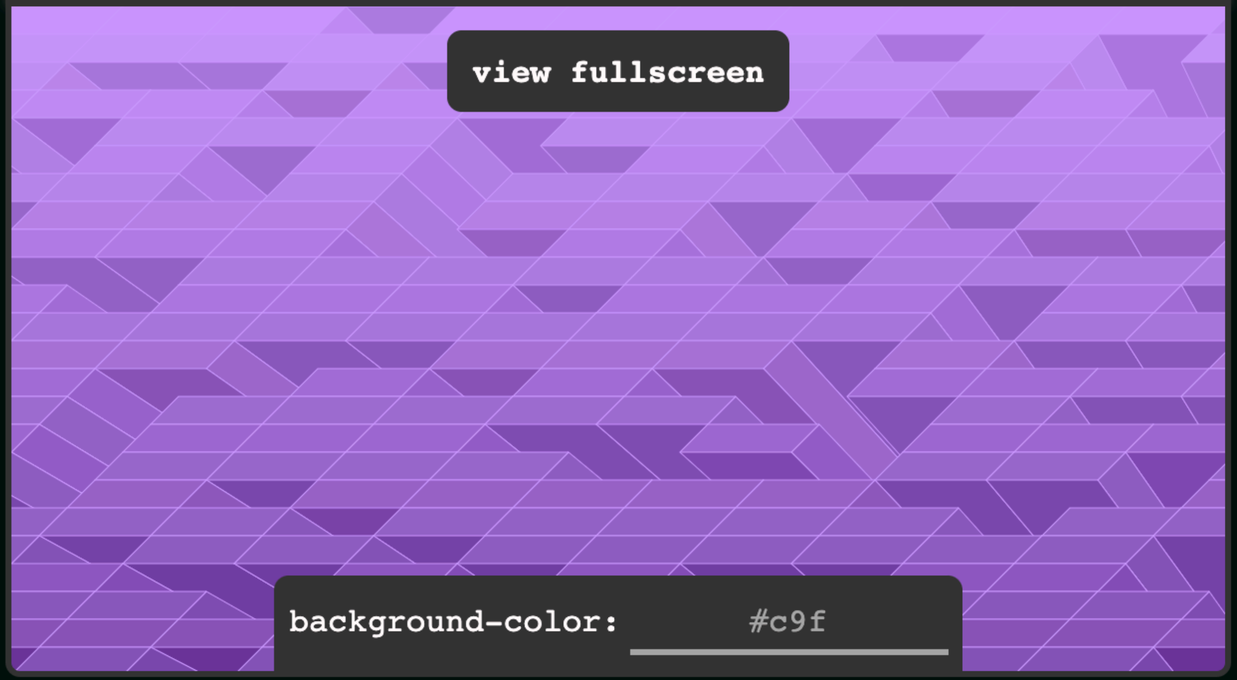
Sam Richard—
Magic tricks with Houdini

tinyurl.com/magicsammy

paintlets!

a gallery of tweakable and downloadable paint worklets!

parallelowow by malchata



bytemare by malchata

--parallelowow-tile-width:

56

--parallelowow-base-color:

#c9f

--parallelowow-color-step:

--bytemare-tile-size:

16

--bytemare-gap:

1

--bytemare-color:

tinyurl.com/yeahpaintlets

malchata / paintlets

Watch 0 Star 6 Fork 0

Code Issues 0 Pull requests 0 Security Insights

A gallery of tweakable and downloadable paint worklets! https://paintlets.herokuapp.com/

30 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Find File Clone or download

malchata Renaming paintlets, tweaking server side stuff. Latest commit 35b4aae 22 hours ago

build-helpers	Bunch of changes.	last month
src	Renaming paintlets, tweaking server side stuff.	22 hours ago
.editorconfig	First (big) commit.	4 months ago
.eslintrc.js	- Updated styles.	3 months ago
.gitignore	Tweaking heroku config.	3 months ago
Procfile	Tweaking heroku port.	3 months ago
README.md	Bunch of changes.	last month
babel.config.js	- Updated styles.	3 months ago
package-lock.json	Renaming paintlets, tweaking server side stuff.	22 hours ago
package.json	Added fullscreen toggle.	2 days ago

tinyurl.com/gitpaintlets

A CSS **Houdini** library for making your site a little more **#extra**.

tinyurl.com/extrauna

Thank you!

slides: jlwagner.net/talks/csspaint