

DOTS & LINES

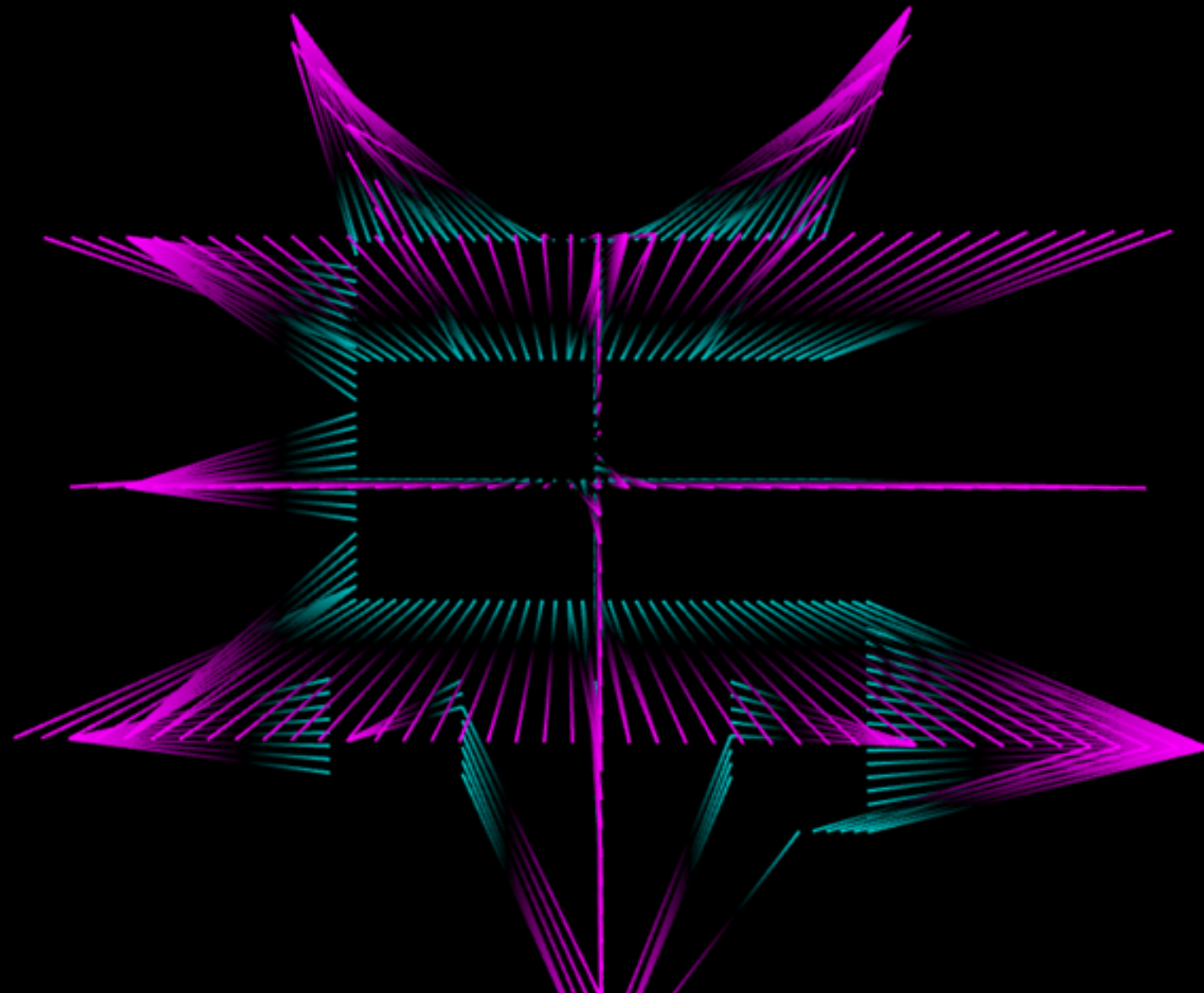
A PROJECT ABOUT
CONNECTION AND TRANSFORMATION
BY JAM ZHANG

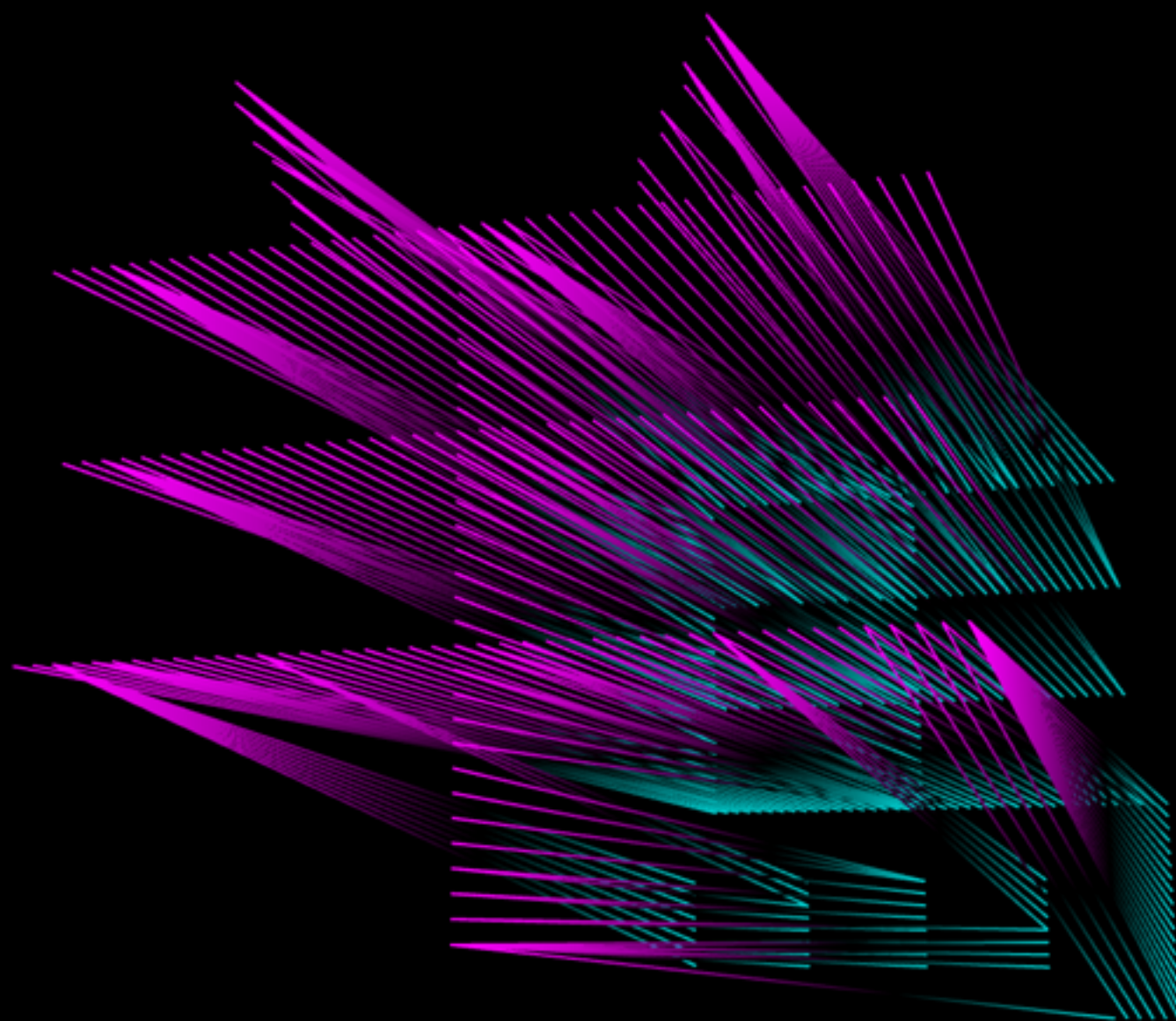
FEB 2015

THE CONCEPT

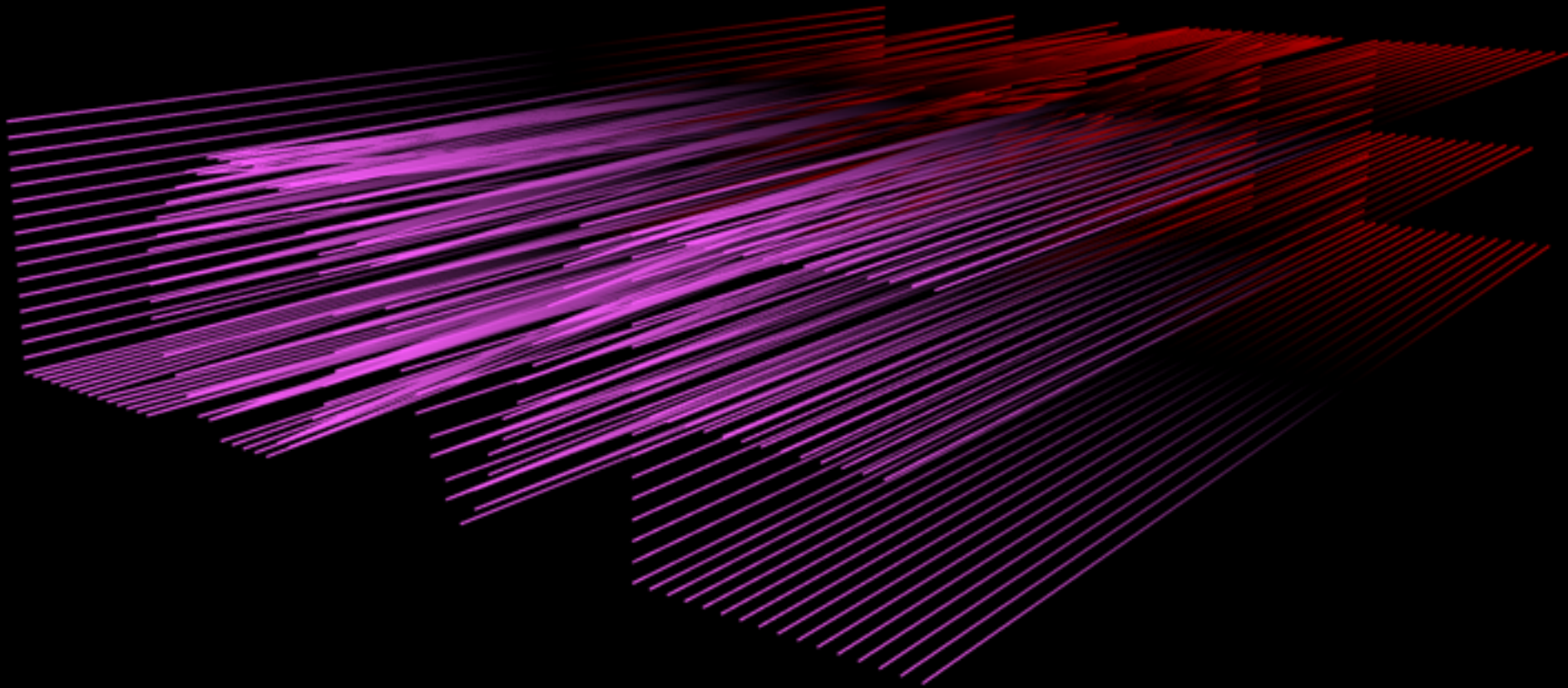
- › THINGS ARE CONNECTED
- › THINGS WE SEE NOW ARE TRANSFORMED FROM THE THINGS EXISTED BEFORE
- › TRANSFORMATION IS A CONNECTION THROUGH THE TIMELINE

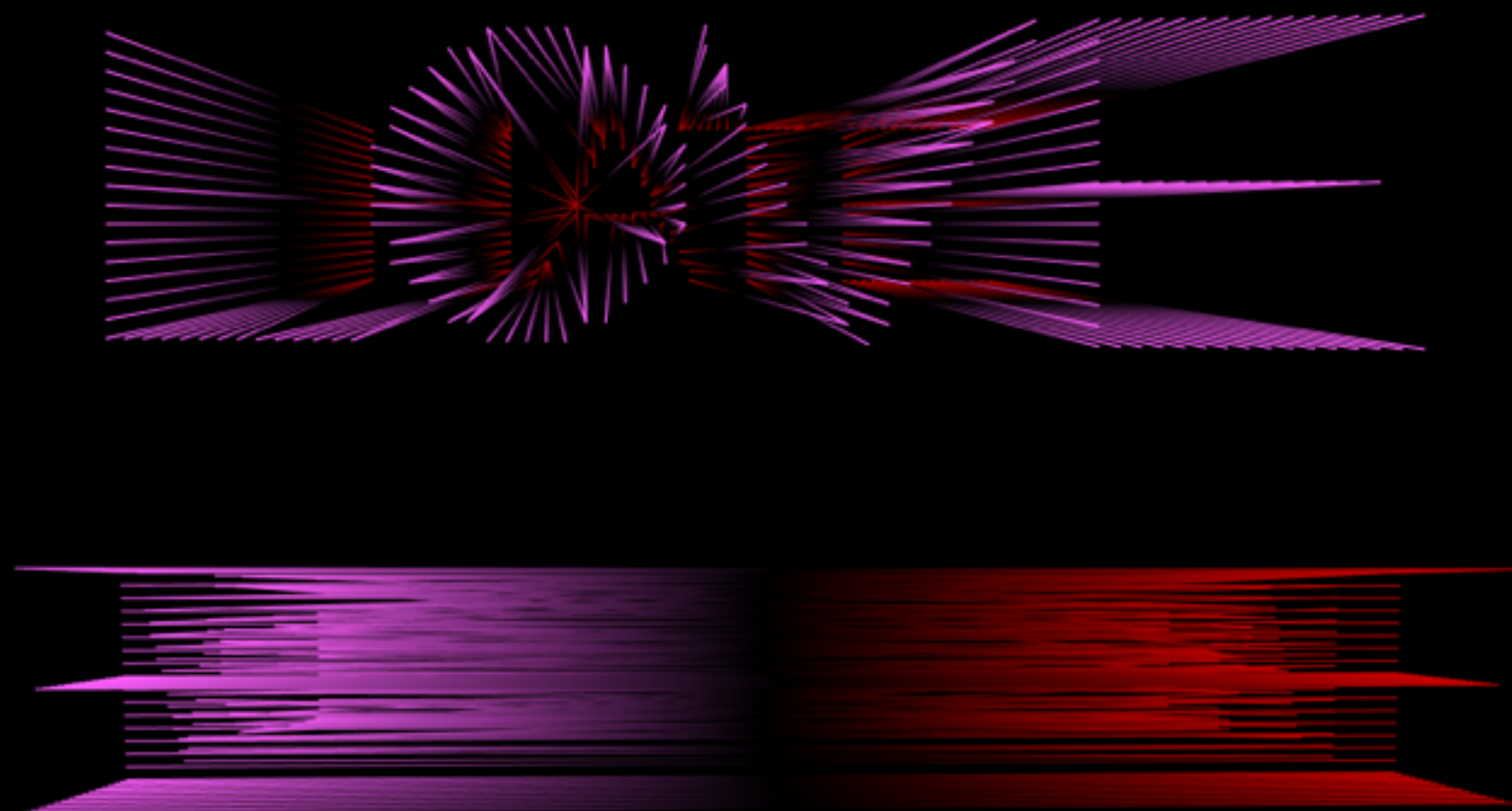
SHEEP AND HORSE

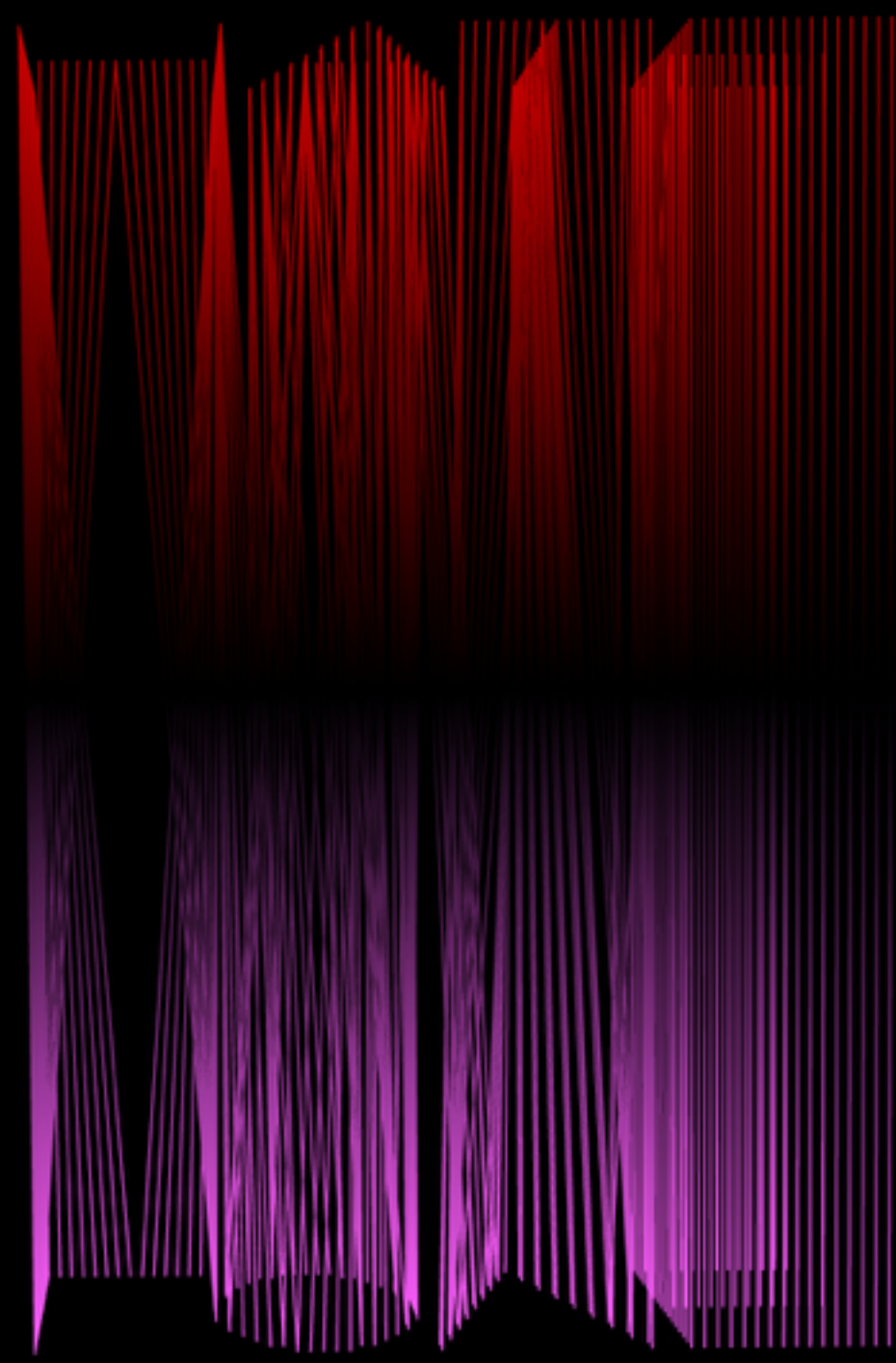




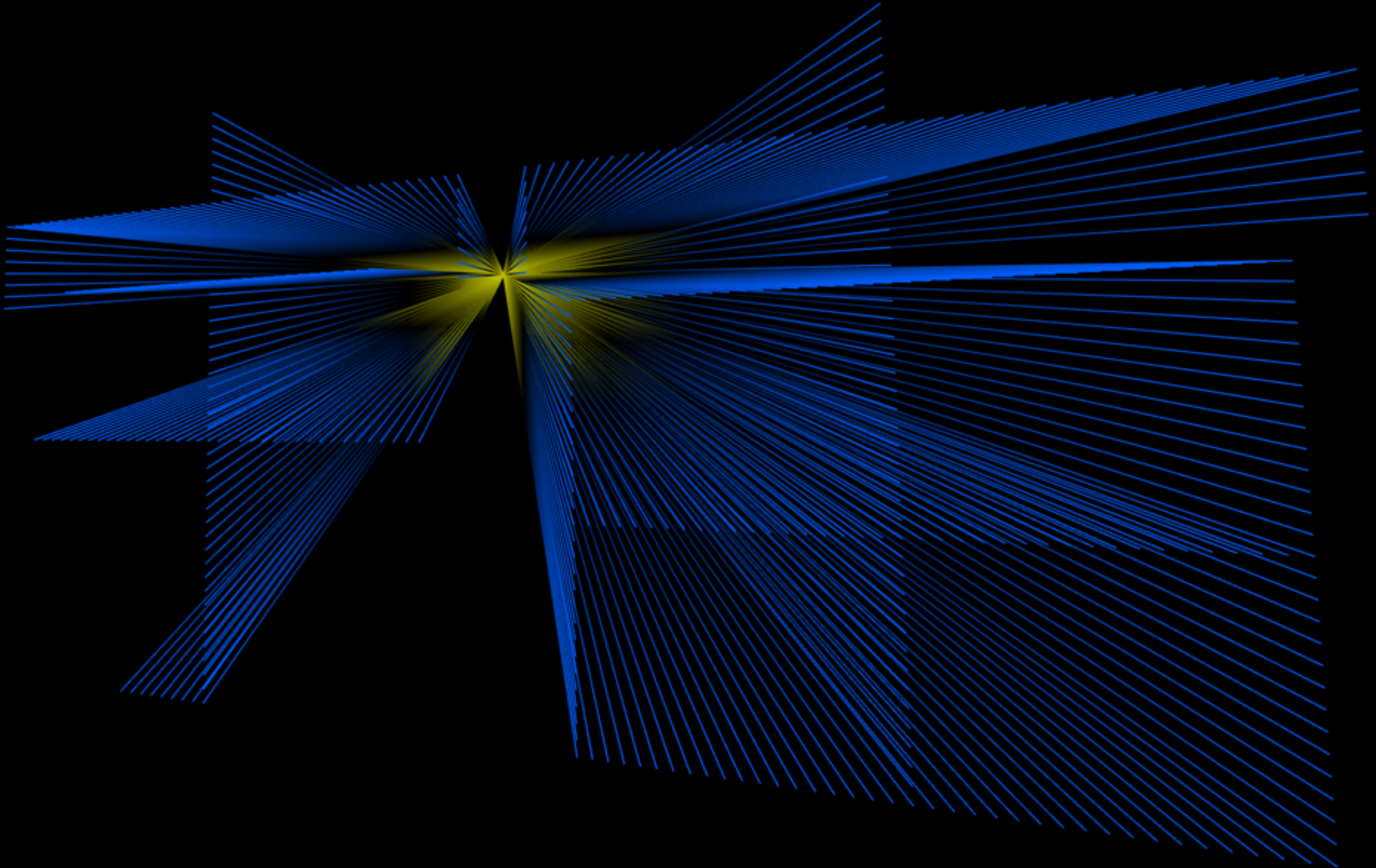
LOVE AND HATE

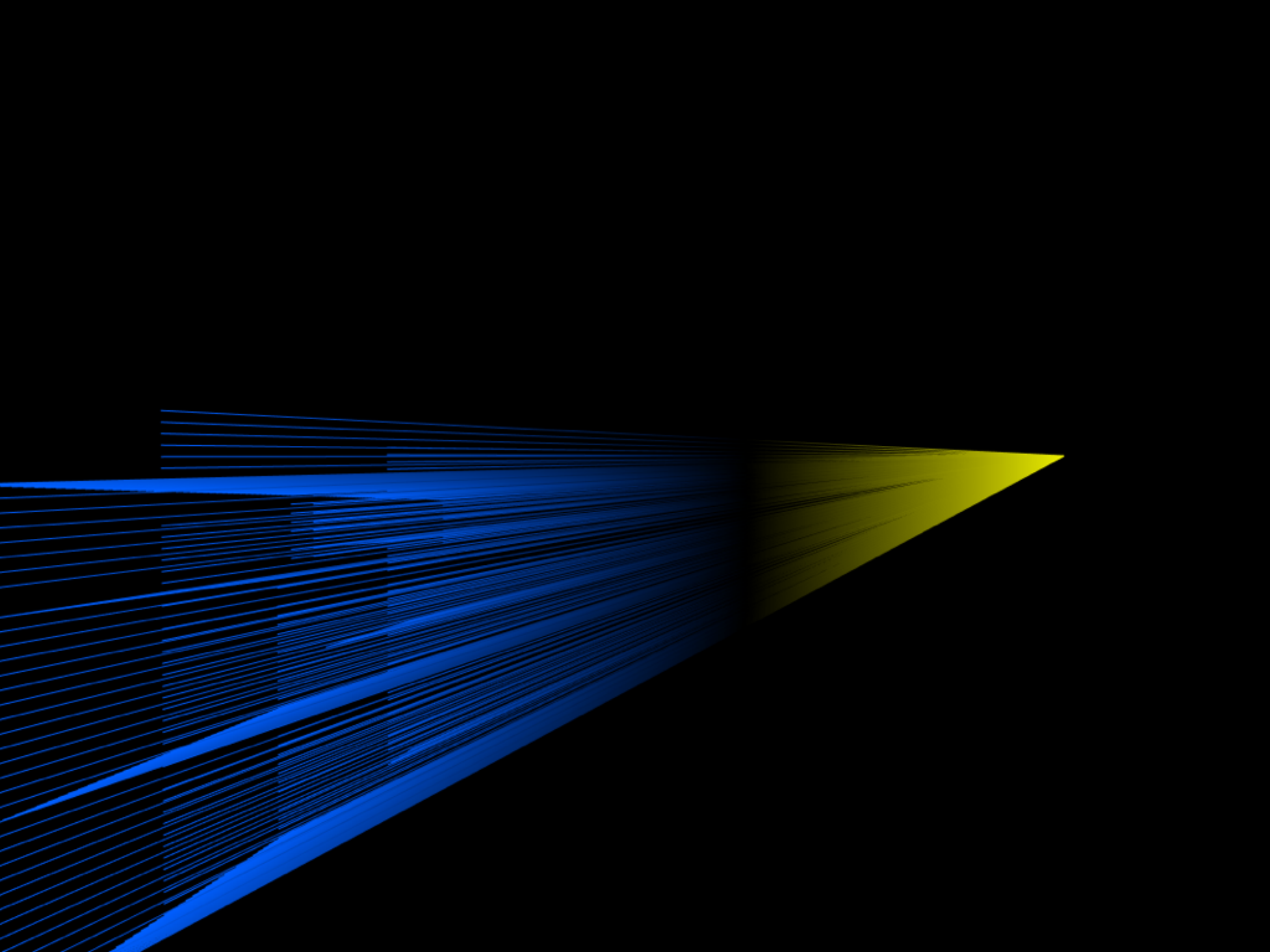




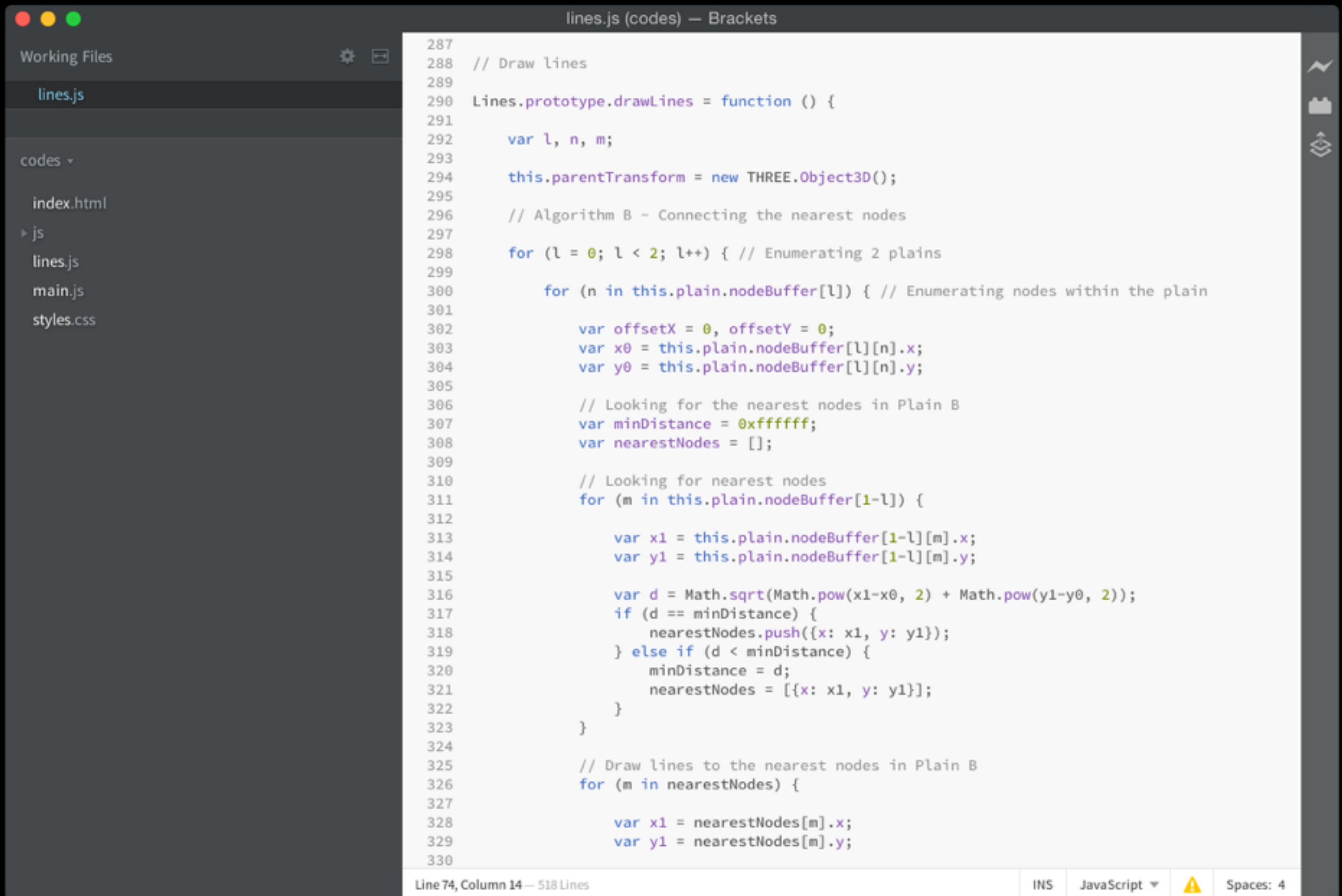


THE UNIVERSE





CONCEPT MADE WITH HTML5 AND THREE.JS



The image shows a code editor window titled "lines.js (codes) — Brackets". The left sidebar displays a file tree with "Working Files" containing "lines.js", and a "codes" folder containing "index.html", "js", "lines.js", "main.js", and "styles.css". The main editor area shows the following JavaScript code:

```
287
288 // Draw lines
289
290 Lines.prototype.drawLines = function () {
291     var l, n, m;
292
293     this.parentTransform = new THREE.Object3D();
294
295     // Algorithm B - Connecting the nearest nodes
296
297     for (l = 0; l < 2; l++) { // Enumerating 2 plains
298         for (n in this.plain.nodeBuffer[l]) { // Enumerating nodes within the plain
299
300             var offsetX = 0, offsetY = 0;
301             var x0 = this.plain.nodeBuffer[l][n].x;
302             var y0 = this.plain.nodeBuffer[l][n].y;
303
304             // Looking for the nearest nodes in Plain B
305             var minDistance = 0xffffffff;
306             var nearestNodes = [];
307
308             // Looking for nearest nodes
309             for (m in this.plain.nodeBuffer[1-l]) {
310
311                 var x1 = this.plain.nodeBuffer[1-l][m].x;
312                 var y1 = this.plain.nodeBuffer[1-l][m].y;
313
314                 var d = Math.sqrt(Math.pow(x1-x0, 2) + Math.pow(y1-y0, 2));
315                 if (d == minDistance) {
316                     nearestNodes.push({x: x1, y: y1});
317                 } else if (d < minDistance) {
318                     minDistance = d;
319                     nearestNodes = [{x: x1, y: y1}];
320                 }
321             }
322
323             // Draw lines to the nearest nodes in Plain B
324             for (m in nearestNodes) {
325
326                 var x1 = nearestNodes[m].x;
327                 var y1 = nearestNodes[m].y;
328
329
330
```

The status bar at the bottom indicates "Line 74, Column 14 — 518 Lines", "INS", "JavaScript", a warning icon, and "Spaces: 4".

THE END = THE BEGINNING