gui

  .add(parameters, "count")

  .min(100)

  .max(100000)

  .step(100)

  .onFinishChange(generateGalaxy);

gui

  .add(parameters, "size")

  .min(0.001)

  .max(0.1)

  .step(0.001)

  .onFinishChange(generateGalaxy);

Adding gui tweaks – but issues cause we are not removing old galaxies

const parameters = {};

parameters.count = 1000;

parameters.size = 0.02;

let geometry = null;

let material = null;

let points = null;

const generateGalaxy = () => {

  //geometry

  if (points !== null) {

    geometry.dispose();

    material.dispose();

    scene.remove(points);

  }

  geometry = new THREE.BufferGeometry();

  const positions = new Float32Array(parameters.count \* 3);

  for (let i = 0; i < parameters.count; i++) {

    const i3 = i \* 3;

    positions[i3 + 0] = (Math.random() - 0.5) \* 3;

    positions[i3 + 1] = (Math.random() - 0.5) \* 3;

    positions[i3 + 2] = (Math.random() - 0.5) \* 3;

  }

  geometry.setAttribute("position", new THREE.BufferAttribute(positions, 3));

  //material

  material = new THREE.PointsMaterial({

    size: parameters.size,

    sizeAttenuation: true,

    depthWrite: false,

    blending: THREE.AdditiveBlending,

  });

  //points

  points = new THREE.Points(geometry, material);

  scene.add(points);

};

generateGalaxy();

//add tweaks

gui

  .add(parameters, "count")

  .min(100)

  .max(100000)

  .step(100)

  .onFinishChange(() => generateGalaxy());

gui

  .add(parameters, "size")

  .min(0.001)

  .max(0.1)

  .step(0.001)

  .onFinishChange(() => generateGalaxy());

Here’s what the code looks like to get a basic galaxy