Christmas Comes To Three.js

Basic setup and scene

Here we set up Three, the ground, lighting and materials to be used.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
</head>
<body>
  <script src="three.min.js"></script>
  <script src="OrbitControls.js"></script>
  <script src="ParticleEngine.js"></script>

  <script  >
    var clock = new THREE.Clock();
    var scene = new THREE.Scene();
    var camera = new THREE.PerspectiveCamera(45, window.innerWidth/window.innerHeight, 0.1, 1000);
    var ambientLight = new THREE.AmbientLight(0x202020);
    scene.add(ambientLight);

    var spotLight = new THREE.SpotLight(0xffffff);
    spotLight.position.set(-60, 300, 60);
    scene.add( spotLight );

    var renderer = new THREE.WebGLRenderer();
    renderer.setClearColorHex(0xEEEEEE);
    renderer.setSize(window.innerWidth * 0.8, window.innerHeight * 0.8);
    renderer.shadowMapEnabled = true;
    document.body.appendChild(renderer.domElement);

    camera.position.x = -60;
    camera.position.y = 80;
    camera.position.z = 60;
    camera.lookAt(scene.position);

    var mtlSnow = new THREE.MeshPhongMaterial({
      color: 0xf0f0ff,
      emissiveColor: 0xf0f0ff,
    });

    var mtlCoal = new THREE.MeshPhongMaterial({
      color: 0x000000,
      emissiveColor: 0x000000,
    });

    var mtlCarrot = new THREE.MeshPhongMaterial({
      color: 0xff0000,
      emissiveColor: 0xff0000,
    });

    var mtlTrunk = new THREE.MeshPhongMaterial({
      color: 0xA56406,
      emissiveColor: 0xA56406,
    });

    var mtlFir = new THREE.MeshPhongMaterial({
      color: 0x00ff00,
    });

    var planeGeometry = new THREE.PlaneGeometry(500,500, 1,1);
    var plane = new THREE.Mesh(planeGeometry, mtlSnow);
    plane.rotation.x = -0.5 * Math.PI;
    plane.position = new THREE.Vector3(15, 0, 0);
    scene.add(plane);

    function makeSnowman(posx, posz){
    };
```

```
    function makeBranch(h, i, dl, d0) {
    };

    function makeTree(posx, posz, k){
    }

    makeSnowman(0, 0);
    makeTree(10,10);

    var controls = new THREE.OrbitControls(camera, renderer.domElement);

    (function animate() {
      requestAnimationFrame(animate);
      renderer.render(scene, camera);
      controls.update();
    })();
  </script>
</body>
</html>
```

Make snowman

Made of three slightly flattened spheres, one on top of the other, with coal eyes and a carrot nose.

```
function makeSnowman(posx, posz){
  // base
  var d = 20;
  var scale = 0.9;

  var sphereGeometry = new THREE.SphereGeometry(d/2, 20, 20);
  var base = new THREE.Mesh(sphereGeometry, mtlSnow);
  base.scale.y = scale;
  var y = d / 2 * scale - 2;
  base.position.set(posx, y, posz);
  base.receiveShadow = true;
  base.castShadow = true;
  scene.add(base);

  // body
  d = 16;
  y = d * scale - 3;
  sphereGeometry = new THREE.SphereGeometry(d/2, 20, 20);
  var sphere = new THREE.Mesh(sphereGeometry, mtlSnow);
  sphere.scale.y = scale;
  sphere.position.y = y;
  base.add(sphere);

  // head
  d = 12;
  y += d * scale;
  sphereGeometry = new THREE.SphereGeometry(d/2, 20, 20);
  sphere = new THREE.Mesh(sphereGeometry, mtlSnow);
  sphere.position.y = y;
  base.add(sphere);

  // eyes
  d /= 2; d -= 0.5;
  var offInRadsHor = 15 * (2 * Math.PI / 360);
  var offInRadsVer = 15 * (2 * Math.PI / 360);

  var offX = Math.sin(offInRadsHor);
  var offY = Math.sin(offInRadsVer);
  var offZ = Math.cos(offInRadsVer);
  sphereGeometry = new THREE.SphereGeometry(0.5, 20, 20);
  sphere = new THREE.Mesh(sphereGeometry, mtlCoal);
  sphere.position.y = y + d * offY;
  sphere.position.x = d * offX;
  sphere.position.z = d * offZ;
  base.add(sphere);

  sphereGeometry = new THREE.SphereGeometry(0.5, 20, 20);
```

```
  sphere = new THREE.Mesh(sphereGeometry, mtlCoal);
  sphere.position.y = y;
  offZ = Math.cos(-offInRadsHor);
  offX = Math.sin(-offInRadsHor);
  sphere.position.x = d * offX;
  sphere.position.y = y + d * offY;
  sphere.position.z = d * offZ;
  base.add(sphere);

  // CylinderGeometry(radiusTop, radiusBottom, height, radiusSegments, heightSegments, openEnded)
  var cylinderGeometry = new THREE.CylinderGeometry(0.5, 1, 6);
  var carrot = new THREE.Mesh(cylinderGeometry,mtlCarrot);
  base.add(carrot);
  carrot.position.z = d;
  carrot.position.y = y;
  carrot.rotation.x = 90 * (2 * Math.PI / 360);
};
```

Make tree

The trunk is easy – a simple cone. The branches are four sets of boxes spiralling up the trunk, each offset by 45°.

```
function makeBranch(h, i, dl, d0) {
  var boxGeometry = new THREE.BoxGeometry(1, 1, 1);
  var branch = new THREE.Mesh(boxGeometry, mtlFir);
  branch.scale.x = 40-(dl * i);
  branch.rotation.y = d0;
  branch.position.copy(new THREE.Vector3(0, i - h/2 + 10, 0));
  return branch;
};
```

```
function makeTree(posx, posz){
  var h = 50;
  var cylinderGeometry = new THREE.CylinderGeometry(0.5, 1, h);
  var trunk = new THREE.Mesh(cylinderGeometry, mtlTrunk);
  trunk.position.x=posx;
  trunk.position.y=h/2;
  trunk.position.z=posz;

  scene.add(trunk);

  var d0 = (90 / h) * (2 * Math.PI / 360);
  var dl = 40 / h;
  var branches = new THREE.Geometry();
  for(var i = 0; i < h-2; i++) {
    var branch = makeBranch(h,i,dl,i*d0); branch.updateMatrix();  branches.merge(branch.geometry,
branch.matrix);
    branch = makeBranch(h,i,dl,i*d0 + 45 * (2 * Math.PI / 360)); branch.updateMatrix();
branches.merge(branch.geometry, branch.matrix);
    branch = makeBranch(h,i,dl,i*d0 + 90 * (2 * Math.PI / 360)); branch.updateMatrix();
branches.merge(branch.geometry, branch.matrix);
    branch = makeBranch(h,i,dl,i*d0 + 135 * (2 * Math.PI / 360)); branch.updateMatrix();
branches.merge(branch.geometry, branch.matrix);
  }
  var allBranches = new THREE.Mesh(branches, mtlFir);
  trunk.add(allBranches);
}
```

Make trees

Place a tree at regular interval, then randomly agitate its position. Skip it if it's too close to the snowman.

```
for(j = -200; j < 200; j+=90) {
  for(i = -200; i < 200; i+=90) {
    var ii = i + Math.random() * 75;
    var jj = j + Math.random() * 75;
    if(Math.abs(ii) > 30 || Math.abs(jj) > 30)
      makeTree(ii, jj);
  }
}
```

Make snow

Three's particle system isn't as comprehensive as Babylon's so I grabbed an engine off the electro-net.

```
var src="..."; // for full string see addendum at end of document
var snow =
{
  positionStyle    : Type.CUBE,
  positionBase     : new THREE.Vector3( 0, 200, 0 ),
  positionSpread   : new THREE.Vector3( 500, 0, 500 ),

  velocityStyle    : Type.CUBE,
  velocityBase     : new THREE.Vector3( 0, -60, 0 ),
  velocitySpread   : new THREE.Vector3( 50, 20, 50 ),
  accelerationBase : new THREE.Vector3( 0, -10,0 ),

  angleBase               : 0,
  angleSpread             : 720,
  angleVelocityBase       :  0,
  angleVelocitySpread     : 60,

  particleTexture : THREE.ImageUtils.loadTexture( src ),

  sizeTween    : new Tween( [0, 0.25], [1, 10] ),
  colorBase   : new THREE.Vector3(0.66, 1.0, 0.9), // H,S,L
  opacityTween : new Tween( [2, 3], [0.8, 0] ),

  particlesPerSecond : 200,
  particleDeathAge   : 4.0,
  emitterDeathAge    : 60
}

var engine = new ParticleEngine();
engine.setValues( snow );
engine.blendStyle = THREE.AdditiveBlending;
engine.initialize();
```

And we need to update the render loop to animate it.

```
(function animate() {
  requestAnimationFrame(animate);

  var dt = clock.getDelta();
  if(dt < 0.1)
  engine.update( dt * 0.5 );

  renderer.render(scene, camera);
  controls.update();
})();
```

Addendum

```
var src =
"
UAAAMAUExURQAAAAMDAQYFAQkIAgwKAw8MBBEOBBQRBRYTBhkVBhwXBx4ZCCAbCSMdCSUfCichCykjDCwlDC0mDTAoDjEqDzQsDzUtEDc
vETkwEjoyEj00Ez42FEA3FUE4FUM6FkU7F0Y9GEg+GEk/GUtBGkxCG01DHE9FHVBGHVFHHlNIH1RJIFVLIFZMIVdNIlhOI1pPJFtQJVxR
JV1SJl5TJ19UKGBVKWJXKWNYKmRZK2VaLGZbLWdcLmhdLmleL2pfMGtgMWxhMm1iM25jNG9kNXBlNXFmNnJnN3NoOHNoOXRpOnVqOnZrO
3dsPHhtPXluPnpvP3twQHxxQX1yQn1zQ350RIB2RYF3RoJ4R4N5SIR5SYV6SoZ7S4d8TIh9TYl+TYl/ToqAT4uBUIyCUY2DUo2DU46EVI
+FVZCGVpGHV5KIWJOJWZSKWpWLW5aMXJeNXZiOXpiPX5mQYJqRYZuSYpySY52TZJ6UZZ+VZp+WZ6CXaKGYaaKZaqKZa6OabKSbbaScbqW
db6aecKeecaifcqmgc6qhdKqhdauidqyjd6ykeK2lea6meq+me7CnfLGofbGpfrKqgLOrgbOrgrSsg7WthLauhbevhriwh7iwiLmxibmy
irqzi7u0jLy0jb21jr62kL63kb+4kr+4k8C5lMG6lcK7lsO8l8S9mMS9mcW+m8W/nMbAncfBnsjBn8nCoMnDocrEosvFo8vFpMzGpc3Hp
87IqM/Jqc/JqtDKq9DLrNHMrtLNr9LNsNPOsdTPstXQs9XQtNbRtdfSt9fTuNjUudnUutrVu9rWvNvXvtzYv9zYwN3Zwd7awt/bxN/bxe
DcxuDdx+HeyOLfyePfy+TgzOThzeXizuXi0Obj0efk0ujl0+jl1Onm1enn1+ro2Ovp2ezp2u3q3O3r3e7s3u7s3+/t4fDu4vHv4/Hv5PL
w5vLx5/Py6PPy6fTz6/X07Pb17fb17vf28Pf38fj48vn48/r59fr69vv79/v7+fz8+v39+/7+/P7+/v///wAAAAAAFX2SBoAAAAJcEhZ
cwAADsMAAA7DAcdvqGQAAAAYdEVYdFNvZnR3YXJlAHBhaW50Lm5ldCA0LjAuNWWFMmUAAAksSURBVFhH7ZhpWFNXGscNZCEJWQghCRAW2
QKETQFZBBFRXCgKgqgIjFo3BHdcBm2p4zY6tk7dQGvVugzIdOrouNFSRQvZ9/3mXqBuiAsCBdl9njk3UuuHdhJlPvL/AIHk/PK+//Oec9
5zx4xqVKMa1R8KM6zhP0cuwLKzs7MHsrO3+7+AAdAei8XhHVAR8HgcFgUPv/lhAkgsjkAkUah0upOTE51KcSQR8NiRYDEYeyzBwZHGcGG
7cT08PT24bhyWM51CJODsPxQLoiSQqAyWu5cvLyg0NCw8PCSY5+fNZTtTyR+IBWHiSRSGq6c/Pzw6LnHylJSUKZMT46IjQvy93JhUMmrC
8EdtFsYORyAz2F4BoVEJKbPmZM/PXZSXOz97TtrUxJhwnhfHmULEvS8VMIlUJtcvNCZp5tzcpYXrS7aVlm7bvL7o40VZs5LjwgI8WDQis
GD44zYJpE5huPmFxU3LzCss2Xnw+OkLVdX/rLp4pvyLXVuLCrKmT4wI4DIphPexAGOHd3Tm8sYnfbRw1dZ95Rcu36qrF4rFwvq6miv/OH
GgtGhRenJUoAdKtRkKcgfMwKjkzMUbdpdX3bwnVeuNkNkMGfUaWX1N9cl9m5dlpUwI9mRS8DY7gME6OLnzoqZmL9926Pz1eqXBjLwVbFQ
LblUe2VGYMy0myMPZEW+jAaihbgGRKfNWfnrs2ztyA4zAZpNBD2QwATxsVN69fOIvRfOnRQdyGSTbDACGkpk+45KzVn5acaVBY0Zgk16r
kstkUplcqdGbwHfohNdO7SqalxLp70q1LVSQPN09JGHOsh0VV4V6BIH0arlEJBQAgcmSqXQQghgkN07tXJWVFD7WhYSzAQoCpbB8YmYWb
DlyWWhAEKNGIRY2AAkE6E+hWKY2IohJfK1i+9L0+GAulWBDqBgs0cmDPyVn7cFLP+nAaLUMIEGMEqlUIhYBsFCqAjYbBN99uSl3WqQvw5
ZQQTmxfSPTl5edrVXDiEkpARyRVKHR6fU6jVIGsAKJElC1dRf3FGYm8F0p1l3F2DvQufyEBRsPX5VCIE7AFEoUWhPS/PP95kZIp5IKGwR
i4IBZcaNiW15qpAeDaLUAMFgyc2zE1GU7z9XpELNWBphSjanp4ZNnz188a33UDGmBHQKpBkKM9VX7CjPi/Vhkq/nbESicgNj0tX+/IjMj
BoUIANTQ/da2jq7u7q7OtqcPYJQqkuthWHnjWMmC5EA3mrWpwtgT6O78Sdmlp2q1iFkjaWgQq6EHzzq6e/v6+/t6X3U+f2hGLZGoIUR/9
1zZkhlhnk5EK2sVzD3DK2Lqot2VDUZLoCK58eennT39A0NAg/09vzy/DylFDUK5AYHE/zqwOiPSh2ltVWGwJBefyJlLP/+3DIZ1UkGDWI
M8aX81MDj0GmhoaKCno7VRKxUIJFozorp+ZH12jD/b0YqpGByZ5R+TvvLoTRXIXoxGdP95V98gikQ11P/qxUOTHLiqghBt7cmtC+MDORS
c3fDw3xeoUg4vPqP4BLDUhKaphB639wxY4kQ1NNDb0QKrwBsKI6KvO7M9PzHYzdqiAmvUNXDi3DVf39EjJgUaEPKks/dtoK9fD/b98rTp
TQqI8afzZX9KCnGn/e/yx4CKcg1KyFp3pg6MQaHqxtbO37JH8+961qyVWKCmhoufLU7iu9McbIKuPW0b9IItUOApBXiauearH3UWT0Uqu
KW9983cWzTY29naqAZQ4Knx3jefFCQGu1vbUzFgOwmInbO6/HsNYlYD6xSmh22v+t+ZqJ6Xj83otylNiP7O139elBDkCiZqePjvCyx9ll
902vIvrylgGK1Hqa75KZipYerQYF/Xs/t6UFJitRnR1JRvyonjcRytlBQofmfvcakF+78VQ4jBUo/mR21dfWBBASRgdr9sgUH2ApkOhuV
XDq2ZG+3LIltbUWDn8wxNXlB27i6oKZUYHY08ftnV2z8wODjQ39vd/qRJj+4oSmBpQ9We5R+N83a2ukzt8FS3oImZJcdvqmCzDh2u0De2
vOjs7unt7XnV2dbSbASOCqRglWprT5XmTeNzaQRrhz+6Tn2jZhburxaYLKECqg552PqivaOjve3poyaDUiywBApJLh9akz0pgGN960e3K
c+wyQWlJ2tUMKyXiwBVroWaHjxuaXn8oAnSKVAmcBTR3j5b9vGsCWOZ1ixF8ydQXXkxWcX7q+v1iNlCFcqUWgNkhiGjVoUeg0IZuvGLvj
u0Lic53N169pb8mV7hM/K3HvuP1NQIDhRAFYgkMoVSpZBLRYApAsxGs+LWV9uXpMXyWKDzGR76xwLzT3PjJWas2Hm6RmFuNOuVEvTYFwp
FIqEQPaHFCi3UCKtvX9i7OispzJvuYDV7IAyOxPAan5JT/NcLP6oA1aCWi1EaKnD+y1R60Alp6qoObliYGuXvSrapoUY3FXZgzIzcDQcu
1spNCAz6HoVMLAISS+VqHWimIFVd9aEtBWlxwR4MB9uaSbRH8eTHp+VvPHC+RqIHDRlk0GpUKqVKrTWg/ZlBVlv5xZYlsxNDvV2s7fq/C
u2jWWNDE9Lz1+89ffWe0giD/tEMAZnRV0Z1w/Vv/layOCMp3Jdtbdf7TaBBpbF9wibOyi3+7GjlzXoFmvKblhfSKwXfV5fvWleQPinCz5
VmvTt5K4w9kcYZGxo3fd7yzfsqKq/dFshUGq1Oo5KJ6m5cOnlg26oFMxPCfV3pJBuTR4XB4Ig0tjd/wpTZeau37j18uvLytVs//FBz48q
ls0f3lRYXZE6NDfHhoEybA0VtxTlQXTx44xJSM/NWbNix++Dh4xUnjh/5fM8nmwrzs6ZPigzyYoOL1PswLVQCxdnNhx+ZkDp7Xv6yonUb
S0o2rSteUTB/zvTECSG+7kyqwwdc+cBtl+7i4csfH5eUmpaRNS9nfk52Rtr0yfFRIX6ebCfH92cCASweXKLZXB8ePyIyOiY2LnZCZHgIz
8eDw6SR3+uy944s130SxcmFw/Xy8fMPCPD38/HmclycqKQRXPhRLA7vQKbSnZguLDabzWIy6FQyET8CJCr0EQqO4EAkkciOjo5kEpEAgh
zpQxQUC7josxkgLNbyWGakyDcCHJT95tfw/0Y1qlGNalTvasyY/wKvHdSvIHaxxwAAAABJRU5ErkJggg==";
```