5. 파이프에서 쏟아지는 물 예제 따라 하기

계정 비쥬얼 시뮬레이션 연구실

파이프에서 쏟아지는 물을 만들어 보는 것이 이번 예제의 목표이다.  
이번 예제는 다음과 같은 순서로 진행할 것이다.

1. 필요한 js파일 연결하기.
2. Three.js의 기본적인 요소들 만들기
3. 파이프 만들기.
4. 물을 표현할 파티클 시스템 설정하기.
5. 물을 파이프에서 생성하기.

**Step 1. 필요한 js파일 연결하기**

|  |  |
| --- | --- |
| <code1> | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <title>Pipe Water</title>  </head>  <body>  <script src="./lib/three/build/three.min.js"></script>  <script src="./lib/three/examples/js/controls/OrbitControls.js"></script>  <script src="./lib/three/examples/js/loaders/OBJLoader.js"></script>  <script src="./ParticleSystem.js"></script>  </body>  </html> |

파이프에서 물을 만들기 위해서 필요한 js파일은 <code1 >의 8째줄에서 11번째 줄에 나타나 있는 4가지 이다. 이번에 사용하는 다양한 함수들은 js파일에 미리 정의되어 있기 때문에 스크립트 태그로 연경을 해주는 것만으로 더 쉽게 코드를 작성할 수 있게 된다.

**Step 2. Three.js의 기본적인 요소들 만들기**

|  |  |
| --- | --- |
| <code2> | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54 | <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <title>Pipe Water</title>  </head>  <body>  <script src="./lib/three/build/three.min.js"></script>  <script src="./lib/three/examples/js/controls/OrbitControls.js"></script>  <script src="./lib/three/examples/js/loaders/OBJLoader.js"></script>  <script src="./ParticleSystem.js"></script>  <script>      var container;      var camera, scene, renderer;        function init() {          container = document.createElement( 'div' );          document.body.appendChild( container );            camera = new THREE.PerspectiveCamera( 35, window.innerWidth / window.innerHeight, 1, 15 );            camera.position.y = 0.8;          camera.position.x = 2.2;          camera.position.z = 3;            controls = new THREE.OrbitControls(camera);            scene = new THREE.Scene();            var light = new THREE.DirectionalLight( 0xffffff, 1.35 );          light.position.set( 1, 1, 1 )          scene.add( light );            light = new THREE.DirectionalLight( 0xffffff, 1.35 );          light.position.set( 2, 3, -1 )          scene.add( light );            renderer = new THREE.WebGLRenderer( { antialias: true } );          renderer.setSize( window.innerWidth, window.innerHeight );            container.appendChild( renderer.domElement );            window.addEventListener( 'resize', onWindowResize, false );      }        function onWindowResize() {          camera.aspect = window.innerWidth / window.innerHeight;          camera.updateProjectionMatrix();            renderer.setSize( window.innerWidth, window.innerHeight );      }  </script>  </body>  </html> |

위의 <code2>와 같이 기본적인 요소들을 설정한다. 13번째 줄과 14번째 줄에서 다양한 함수에서 쓰이는 변수들은 전역변수로 선언했다. 선언한 변수들은 16번째 줄에 나타나 있는 init함수에서 초기화 했다. 파이프를 볼 수 있게 하기 위해서 init함수에서 light를 만들었다.

**Step 3. 파이프 만들기**

|  |  |
| --- | --- |
| <code3> | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93 | <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <title>Pipe Water</title>  </head>  <body>  <script src="./lib/three/build/three.min.js"></script>  <script src="./lib/three/examples/js/controls/OrbitControls.js"></script>  <script src="./lib/three/examples/js/loaders/OBJLoader.js"></script>  <script src="./ParticleSystem.js"></script>  <script>      var container;      var camera, scene, renderer;        init();      loop();        function init() {          container = document.createElement( 'div' );          document.body.appendChild( container );            camera = new THREE.PerspectiveCamera( 35, window.innerWidth / window.innerHeight, 1, 15 );            camera.position.y = 0.8;          camera.position.x = 2.2;          camera.position.z = 3;            controls = new THREE.OrbitControls(camera);            scene = new THREE.Scene();            var light = new THREE.DirectionalLight( 0xffffff, 1.35 );          light.position.set( 1, 1, 1 )          scene.add( light );            light = new THREE.DirectionalLight( 0xffffff, 1.35 );          light.position.set( 2, 3, -1 )          scene.add( light );            renderer = new THREE.WebGLRenderer( { antialias: true } );          renderer.setSize( window.innerWidth, window.innerHeight );            container.appendChild( renderer.domElement );            window.addEventListener( 'resize', onWindowResize, false );            var loader = new THREE.OBJLoader();          loader.load('./models/pipe.obj', function(object){                var material = new THREE.MeshPhongMaterial( { color: 0x101010, specular: 0xaaaaaa, shininess: 30 });              material.side = THREE.DoubleSide;                object.traverse( function ( child ) {                    if ( child instanceof THREE.Mesh ) {                        child.material = material;                      child.castShadow = true;                      child.receiveShadow = true;                  }              });                object.position.y = -0.5;              object.scale.set(0.02, 0.02, 0.02);              object.rotation.x = -Math.PI\*0.5;                scene.add(object);          });      }        function onWindowResize() {          camera.aspect = window.innerWidth / window.innerHeight;          camera.updateProjectionMatrix();            renderer.setSize( window.innerWidth, window.innerHeight );      }        var fps = 40;        function loop() {          setTimeout(loop, 1000/fps);          controls.update();            requestAnimationFrame(render);      }        function render() {          renderer.render( scene, camera );      }  </script>  </body>  </html> |

파이프 모델을 가져와서 화면에서 보기 위해서 위의 <code3>처럼 init함수에 코드를 추가했다. 48번째 줄에서 69번째 줄에 나온 것처럼 OBJLoader을 이용해서 미리 정의 되어 있는 pipe.obj에서 파이프 모형을 가져왔다. 특히, 51번째 줄에서 모델의 색, 반사색, 반사 정도를 설정했다. 이렇게 설정한 material을 58번째 줄에서 적용시킨다. 62, 63, 64번째 줄에서는 가져온 모델의 위치와 크기를 설정했다.

이렇게 만든 파이프를 확인하기 위해서 loop함수와 render함수를 만들고 16, 17번째 줄에서와 같이 init함수와 loop함수를 실행시키면 아래의 <그림1>과 같은 화면을 볼 수 있을 것이다.

|  |
| --- |
|  |
| <그림 1> |

**Step 4. 물을 표현할 파티클 시스템 설정하기**

|  |  |
| --- | --- |
| <code4> | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109 | <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <title>Pipe Water</title>  </head>  <body>  <script src="./lib/three/build/three.min.js"></script>  <script src="./lib/three/examples/js/controls/OrbitControls.js"></script>  <script src="./lib/three/examples/js/loaders/OBJLoader.js"></script>  <script src="./ParticleSystem.js"></script>  <script>      var container;      var camera, scene, renderer;        var particleSystem = new ParticleSystem();        particleSystem.initialize(10000);        particleSystem.setParameters({          seedVelDir: new THREE.Vector3(0,0,1),          seedVelMag: 2.0,          globalForce: new THREE.Vector3(0, -10,0),          windStrength: 0.05,          seedSize: 0.6,          seedLife: 2.0,          texFile : "./textures/gradient.png",          particleColor: new THREE.Color(0.1, 0.1, 1.0),          alpha : 0.5      });        init();      loop();        function init() {          container = document.createElement( 'div' );          document.body.appendChild( container );            camera = new THREE.PerspectiveCamera( 35, window.innerWidth / window.innerHeight, 1, 15 );            camera.position.y = 0.8;          camera.position.x = 2.2;          camera.position.z = 3;            controls = new THREE.OrbitControls(camera);            scene = new THREE.Scene();            var light = new THREE.DirectionalLight( 0xffffff, 1.35 );          light.position.set( 1, 1, 1 )          scene.add( light );            light = new THREE.DirectionalLight( 0xffffff, 1.35 );          light.position.set( 2, 3, -1 )          scene.add( light );            renderer = new THREE.WebGLRenderer( { antialias: true } );          renderer.setSize( window.innerWidth, window.innerHeight );            container.appendChild( renderer.domElement );            window.addEventListener( 'resize', onWindowResize, false );            var loader = new THREE.OBJLoader();          loader.load('./models/pipe.obj', function(object){                var material = new THREE.MeshPhongMaterial( { color: 0x101010, specular: 0xaaaaaa, shininess: 30 });              material.side = THREE.DoubleSide;                object.traverse( function ( child ) {                    if ( child instanceof THREE.Mesh ) {                        child.material = material;                  }              });                object.position.y = -0.5;              object.scale.set(0.02, 0.02, 0.02);              object.rotation.x = -Math.PI\*0.5;                scene.add(object);          });            scene.add(particleSystem.getMesh());      }        function onWindowResize() {          camera.aspect = window.innerWidth / window.innerHeight;          camera.updateProjectionMatrix();            renderer.setSize( window.innerWidth, window.innerHeight );      }        var fps = 40;        function loop() {          setTimeout(loop, 1000/fps);          controls.update();            requestAnimationFrame(render);      }        function render() {          renderer.render( scene, camera );      }  </script>  </body>  </html> |

파티클 시스템을 전역변수로 선언하고 setParameters함수를 이용해서 입자의 성질을 설정했다. 그리고 init함수에서 scene에 particleSystem의 mesh를 추가했다.

**Step5. 물을 파이프에서 생성하기**

|  |  |
| --- | --- |
| <code5> | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120 | <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <title>Pipe Water</title>  </head>  <body>  <script src="./lib/three/build/three.min.js"></script>  <script src="./lib/three/examples/js/controls/OrbitControls.js"></script>  <script src="./lib/three/examples/js/loaders/OBJLoader.js"></script>  <script src="./ParticleSystem.js"></script>  <script>      var container;      var camera, scene, renderer;        var particleSystem = new ParticleSystem();        particleSystem.initialize(10000);        particleSystem.setParameters({          seedVelDir: new THREE.Vector3(0,0,1),          seedVelMag: 2.0,          globalForce: new THREE.Vector3(0, -10,0),          windStrength: 0.05,          seedSize: 0.6,          seedLife: 2.0,          texFile : "./textures/gradient.png",          particleColor: new THREE.Color(0.1, 0.1, 1.0),          alpha : 0.5      });        init();      loop();        function init() {          container = document.createElement( 'div' );          document.body.appendChild( container );            camera = new THREE.PerspectiveCamera( 35, window.innerWidth / window.innerHeight, 1, 15 );            camera.position.y = 0.8;          camera.position.x = 2.2;          camera.position.z = 3;            controls = new THREE.OrbitControls(camera);            scene = new THREE.Scene();            var light = new THREE.DirectionalLight( 0xffffff, 1.35 );          light.position.set( 1, 1, 1 )          scene.add( light );            light = new THREE.DirectionalLight( 0xffffff, 1.35 );          light.position.set( 2, 3, -1 )          scene.add( light );            renderer = new THREE.WebGLRenderer( { antialias: true } );          renderer.setSize( window.innerWidth, window.innerHeight );            container.appendChild( renderer.domElement );            window.addEventListener( 'resize', onWindowResize, false );            var loader = new THREE.OBJLoader();          loader.load('./models/pipe.obj', function(object){                var material = new THREE.MeshPhongMaterial( { color: 0x101010, specular: 0xaaaaaa, shininess: 30 });              material.side = THREE.DoubleSide;                object.traverse( function ( child ) {                    if ( child instanceof THREE.Mesh ) {                        child.material = material;                  }              });                object.position.y = -0.5;              object.scale.set(0.02, 0.02, 0.02);              object.rotation.x = -Math.PI\*0.5;                scene.add(object);          });            scene.add(particleSystem.getMesh());      }        function onWindowResize() {          camera.aspect = window.innerWidth / window.innerHeight;          camera.updateProjectionMatrix();            renderer.setSize( window.innerWidth, window.innerHeight );      }        var fps = 40;      var time;      function loop() {            setTimeout(loop, 1000/fps);            var now = new Date().getTime();          var dt = (now - (time||now))/1000;          time = now;            var cen = new THREE.Vector3(-0.57,0.24,0.24);          var rad = 0.11;            particleSystem.addParticlesFromSphere(50, cen, rad);          particleSystem.updateParticles(dt);          controls.update();            requestAnimationFrame(render);      }        function render() {          renderer.render( scene, camera );      }  </script>  </body>  </html> |

설정한 파티클 시스템을 이용해서 물을 만들 것이다. loop함수에서 addParticlesFromSphere을 통해서 입자를 생성하는 것은 앞에서의 예제와 같지만 이번에 특히 신경 써야 하는 것은 물이 생성되는 위치를 잘 잡는 것이다. 여기에서는 (-0.57, 0.24, 0.24)에서 물을 만들어 냈다. 다만 파이프의 크기와 위치가 달라진다면 생성되는 위치를 다시 설정해야 한다. <code5>를 브라우저에서 실행 시켜보면 아래의 <그림2>와 같이 파이프에서 물이 만들어지는 모습을 확인할 수 있을 것이다.

|  |
| --- |
|  |
| <그림 2> |