Deteriorated Radioactive Waste Facility

A scene of nature overtaking industry

Luke Mason†  
 Computer Science: Computer Game Design  
 U.C. Santa Cruz  
 Santa Cruz, CA, USA  
 ljmason@ucsc.edu

Josh Husting  
Computer Science: Computer Game Design  
 U.C. Santa Cruz  
 Santa Cruz, CA, USA  
 jhusting@ucsc.edu

Damen Birtola  
 Computer Science: Computer Game Design  
 U.C. Santa Cruz  
 Santa Cruz, CA, USA  
dbirtola@ucsc.edu

Zack Lawrence  
 Computer Science: Computer Game Design  
 U.C. Santa Cruz  
 Santa Cruz, CA, USA  
zarlawre@ucsc.edu

John Khaw  
Computer Science: Computer Game Design  
 U.C. Santa Cruz  
 Santa Cruz, CA, USA  
 jkhaw@ucsc.edu

David Kirkpatrick  
 Computer Science: Computer Game Design  
 U.C. Santa Cruz  
 Santa Cruz, CA, USA  
dlkirkpa@ucsc.edu

ABSTRACT

This project aims to embody our vision of a deteriorated radioactive waste facility. Areas like this are often riddled with destruction and mysterious chemical reactions allowing us to take a lot of liberty with the types of effects we want to include. The premise of the scene focuses on a radioactive orb hovering in the air above a pedestal. The ceiling, although not shown, can be inferred as very damaged as the rain and sunlight peeks through. In areas the sunlight reaches, grass has started to grow which has a stark contrast with the industrial scene that surrounds it. To top it off, a pipe pours radioactive waste into a river that divides the scene.

The scene was created using Unreal Engine 4, Maya, and Substance Painter. Unreal was chosen due to the power if its material editor in comparison to Unity in addition to the team having prior experience with the engine.

∗Article Title Footnote needs to be captured as Title Note

†Author Footnote to be captured as Author Note

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

1.1 The Plasma Orb

*1.1.1 Orb*

The Orb is a 2D Circular Mesh that is camera facing to give the illusion of being a sphere. There are two panning textures multiplying one another to add motion to the orb. In the particle system we layered the different circular meshes using the camera offset module to put the ‘orb’ circular mesh closer to the camera. Behind that mesh we have the same mesh with another material instance that has the panning texture going in the V direction, rather than the U direction, to have the energy radiate away from the center of the orb. We multiply this noise by a gradient to have it dissipate along the edges of the circular mesh. We pass both the gradient and the noise through a sphere mask to better dissipate than the predictable multiplication of the two textures. We have a third circular mesh behind both of these that is just a gradient glow, using the same mesh but with a material of only a gradient with no noise.

*1.1.2 Electricity*

For the spherical lighting around the mesh, we brought in an icosphere mesh for its convenient UV’s. We have this mesh centered around the center of the circular planes to better fake the depth. The material itself has a two panning textures multiplied passed into the UV’s of our lightni9ng texture to distort the UV’s. This distorted lightning is then passed through alpha threshold-ing to only let a specified range of color through the opacity to bring out the lighting edges in the material.

The Lightning bolts of using a mesh that is generated with a cylinder that is bent in multiple locations to create the arc of lightning. The material uses the same techniques of the orb, but instead of using a texture that seamlessly pans across the orb, swapped out with a straight line texture. Combined with the UV distortion and alpha threshold-ing, this material applied to the meshes generates the lightning arcs we desired. To add more sporadic movement, I used a time sin to offset the lightning textures position to move across the UV’s to not have them look static in the final effect. Finally, these meshes origins are set to the bottom of the lightning, allowing us to random rotate them in the particle system and always having their mesh point out from the center of the orb. Along with the random rotating, they have a random size within a range, and a random rotation rate when spawned. Each mesh sizes change over time to be largest when spawned and then shrink near the half point in their life to add more chaotic motion to the random lightning.

1.2 The River

The scene is divided by a river of toxic waste. In one end of the room a pipe pours the green liquid into the river. It ripples and sends the waste off around a bend and through a grate. The river emits bubbles which pop intermittently throughout its path.

*1.21 The Pouring Pipe*

The Pouring pipe is a mesh composed of a cylinder that was passed through a bend modifier to give its curved L shape that fluid would receive with gravity. The UV’s are normalized for the mesh so that panning textures would be seamless. For the material, we have a panning noise texture for color that is stretched out along the mesh in direction of the ‘flow’ due to the UV’s of the mesh. On top of this, we have two panning textures blended together to generate the world offset positions. To make it more displace more away from the pipe, and less chaotic near the pipe, we generate a linear gradient that we then multiply against the noise, with white values being the most deformation, and black areas as hardly deformed.

*1.22 The River*

The river is created by panning textures over a curved mesh. The mesh started as a simple plane with sub divisions created a higher poly starting point. It was then stretched to the length of the river and bent with a bend deformer tool which preserves the original UV’s of the plane. This allows us to pan textures over it and have them follow the bend of the mesh.

To add depth to the river and break up the tiling once would notice after multiple passes, two textures are panned at different speeds over each other. This creates the illusion of a “top” and “bottom” texture which helps simulate the light dispersion that happens as depth increases and together with the slower panning gives the illusion of multiple layers of material passing through the river. As a final touch, we add an emissive flare to the river by lerping between the two panning textures based on the sin of the game time to create an emissive map that varies in severity and location throughout the river. We had plans to use the texture as a vertex displacement map to better sell the illusion of flowing water but unfortunately we ran into issues with Unreal that caused flickering which we were not able to work around in time.

*1.23 The Bubbles*

The bubbles were made using two particle systems that would spawn bubbles in an area at a set rate. The particle system would randomize the initial size, rotation, and rotation speed of the spawned bubbles, which used a simple circular mesh. After spawning the bubble, it would slowly increase in size until it reaches a set size, then “pops” by rapidly decreasing in size.

While the particle system is spawning, rotating and resizing the bubbles, I also have a custom shader running on it. The shader takes in two parameters: Alpha Erosion and Mesh Deformation.

Alpha Erosion (0 – 1)

As this parameter approaches 1, it will mask out the image more and more according to a noise texture. A value of 0.5 would mask out everything in the image, except for the areas of the texture that have a r value greater than 0.5.

Mesh Deformation (0 – 1)

As this parameter approaches 1, it will deform the mesh more and more according to the same noise texture as the previous parameter. This deforms the mesh more at the edges of the pop, and less near the solid parts of the mesh.

Both parameters are controlled by the particle system, and are rapidly increased when the bubble pops.

1.3 The Environment

Our environment serves to provide a contrast between the industrial science lab and nature. Although the ceiling is not visible, the viewer can interpret its timely destruction as sunlight and rain find their way through select regions of the scene. To accent this sense of time passed we add vegetation like grass growing at the bottom of the god rays.

*1.31 God Rays*

To make the strands of light, we have a panning texture that we apply UV distortion with another noise function to make it not pan linearly. We then use this panning texture into opacity so that the brightest colors are also the most opaque. To add a smokier, foggy look, we apply a panning smoke noise to a linear gradient so that the smoke is most visible at the bottom and hardly influencing the top. We multiply blend the strands of the light and ground smoke textures for the final textures to drive into color and transparency. We apply this to a cone mesh gives the illusion of a spotlight.

*1.32 Grass*

Our windy grass is done using a few different techniques in its shader. First we displace the vertexes on two different levels. We do “large scale” wind that affects the model more uniformly. Then we do “small scale” wind that affects individual vertexes differently, but is weighted less than large scale wind.

Both textures are panning, and the sample location that determines the displacement is calculated using the vertex’s world position as the textures UV input. This allows us to have wind seem like it’s affecting the world, not just the individual grass models. The color for this shader is just a solid color.

*1.33 Steam*

The steam is done by having two different panning noise textures on a curved cone mesh. Vertex deformation is used to manipulate the shape of the mesh to simulate steam dissipating around the end of the cone. An opacity gradient is used to make sure the steam becomes more transparent at the top of the mesh to allow the steam to appear more wispy.

*1.34 Rain*

We created our rain by using a particle system to stream a simple radial texture downwards. We tint the color of the particle light blue and set the initial scale to randomize in the x and y between two ranges, one twice as large as the other. This helps to get the lengthy stream appearance that most people associate with rain even if it is not exactly true to life. By making sure one is twice the size of the other we also create a variety of raindrop sizes which makes the rain more apparent without being oppressive in the scenes composition.

ACKNOWLEDGMENTS

Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

REFERENCES

[1] FNM Surname (2018). Article Title. Journal Title, 10(3), 1–10.

[2] F.N.M Surname, Article Title, https://www.acm.org/publications/proceedings-template.

[3] F.N.M Surname and F.N.M Surname, 2018 Article Title, The title of book two (2nd. ed.). Publisher Name, City, State, Country.

[4] Ian Editor (Ed.). 2018. *The title of book two* (2nd. ed.). University of XXX Press, City, Chapter 100. DOI: <http://dx.doi.org/10.1000/0-000-00000-0>.

Conference Name:ACM Woodstock conference

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00