Final Project Report

Project Link

Team Name: Red Volcano

Team representative: Ryan Brennan

Ryan Brennan
rpbrennan@g.ucla.edu
UID: 605-347-597
Github: ryanpbrennan

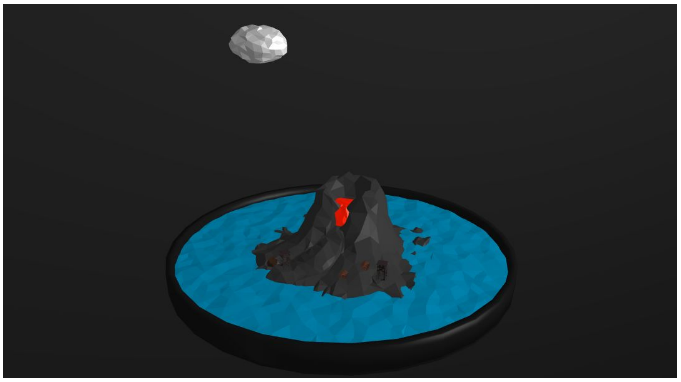
Ahmad Habibollahzadeh
ahmadhz@g.ucla.edu
UID: 405-347-485
Github: Ahmadahz

Irfan Sattar
irfansattar@ucla.edu
UID: 605-181-709
Github: irfansattar

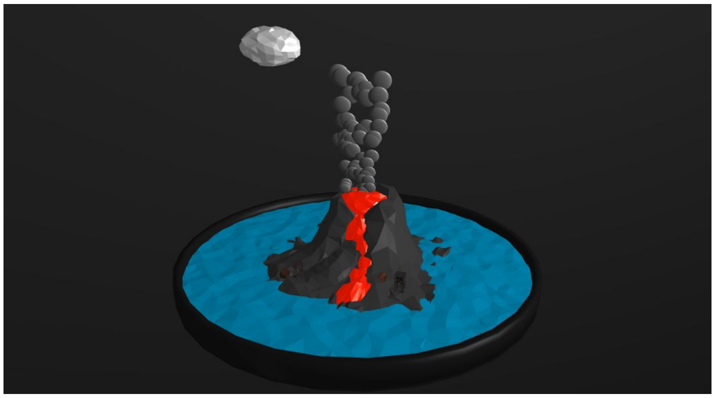
Summary

The idea behind our project is an interactive, visuals-focused scene of a volcano spewing smoke and lava with environmental additions like buildings and a body of water. In addition to the volcano and environment, a user-operated cloud is present with the ability to rain down on the scene. The advanced features we chose to implement were a particle system for the smoke from the volcano and the rain from the cloud, as well as collision detection between the rain and smoke: when the rain drops come in contact with the smoke particles, the smoke particles disappear. We chose to use these particular features because it made sense for creating a story in our scene and we thought it would be both interesting and challenging to have the two advanced features interact with one another.

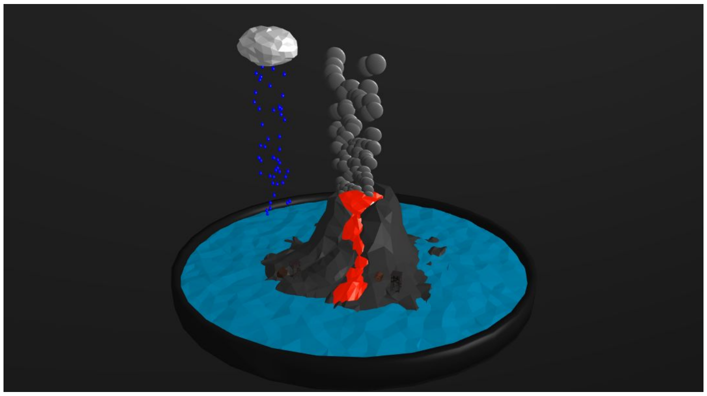
Our scene starts off here: an island floating in the night sky with a volcano surrounded by water, a few buildings at the base, and a cloud overhead. Many of these objects were modelled using blender, imported to our project and displayed using webGL and tiny-graphics.



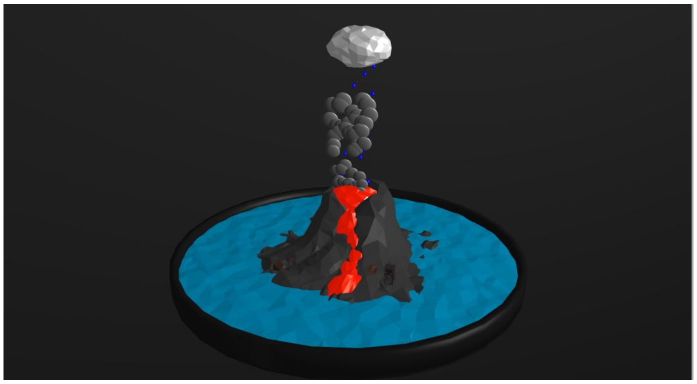
Included is a control panel (not shown), which allows the user to toggle features on and off, as well as control the cloud motion and the camera’s position. The screen below shows the scene after the smoke has been toggled on.



Here is the same scene, but showing the rain feature toggled on. We can also see here how the lava has made its way down the side of the volcano and stopped at the sea, consuming the small shack on the side of the volcano in the process. The lava was implemented as a series of solid objects rising out of the side of the mountain. This was an example of where our implementation displayed creativity: due to the complexity of our other advanced features, we used course concepts like translation, rotation, and scaling to move solid objects in a way that simulated the flow of liquid for our project instead of implementing complex physic-based liquid simulation.



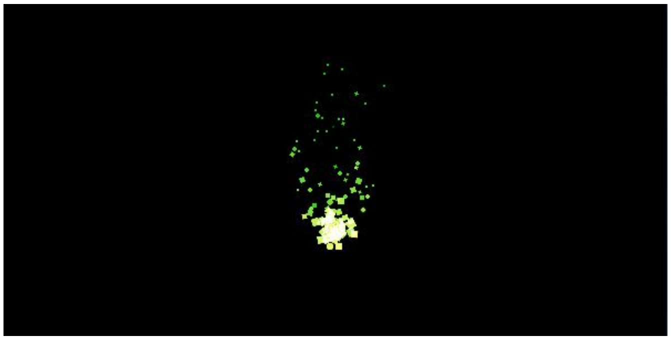
In the following image, we can see both features on and interacting through collision detection. As the rain drops come in contact with the smoke particles, the smoke particles disappear. The implementation of both particles systems and the collision detection between them is described below in the advanced features section.



Advanced Features

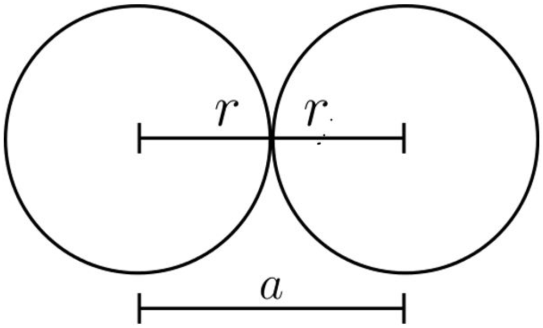
Particle System

We used a particle system to create the smoke that comes out of the volcano and the raindrops that come out of the cloud. This system helped to use a high number of particles in our project and that helped us a lot to make the scene more realistic. For implementation of it we first created a simple black circle. Then we specified a velocity to it and we observed that it worked for one particle. Then we created a table of velocity and generated different velocities in the specified range and stored them in that table. After that we used our particle and each time stored the particle with a different velocity in an array. Then we used loops and other similar tools to create the particle system. For making the program smoother we gave a lifetime to each particle so they get removed after they get to some point. For creating this life time we used the technique from hw4 which was using module and animation time so that the number increases smoothly but it can't go to infinity.



Basic Collision Detection

We used this feature to remove the smoke particles when they contact the cloud. It was challenging at first because doing collision detection for a complex shape was hard. However, we tried to use approximations and created simpler invisible shapes to make the detection easier. The shape of the cloud was so close to a circle, and the smoke particles had the shape of a circle, so we used a circle to make things easier.



For this feature, we used some simple codes to constantly check the distance between the center of the two objects that we want(cloud and smoke). If the distance is equal to the summation of radiuses, then we can guarantee that they have collided. And clearly, if the distance is larger, they have not collided yet. After the implementation, we realized that if we make the invisible radius larger, it is going to look like that the raindrops are removing the smoke particles, so we changed previous code to make it more interesting.

One more interesting feature (not very advanced):

For creating the lava first, we wanted to create an object that works completely like a liquid. However, we found out that it was challenging to do especially given time constraints after having implemented our other features. As a result, we tried the technique currently in our project. In this technique, we used different shapes with different sizes on the surface of the volcano. Then we gave them movement with the help of animation time so that when we load the scene, the y component of those objects increases. We first weren't sure about doing this, but the result looked excellent, and the movement of the lava felt like a real liquid.

Contributions

Irfan

Added tiny-graphics
Added volcano obj file
Restructured code, created volcano.js
Integrated new changes with master branch
Moved particle system to separate branch
Improved models, added buildings

Merged scene-wip
created separate branches for smoke and particle effects
Added background texture (EC)
Fixed smoke particle to work with new background
Code cleanup
Optimized rain drops

Ahmad

Created a basic form of the Scene
Created a better Scene
Added an unoptimized and the first particle system
Optimized the smoke particle system
Finished cloud with rain particles
Basic collision detection
Added the first try for the lava
Added readme

Ryan*

First test commit
Added inn, modified lava
Moved and added structures, modified rain
Merged lava-motion branch and associated changes
Changed default smoke state
Added buggy branch for camera pan
Finished rough pan for camera angle change when switching background
Final merge before submission: integrate rough camera pan

*In the early stages of the project when we were figuring out how we wanted to implement some of the bigger pieces in the scene I was just making them on my local machine and sharing the results in the group chat. We ultimately wound up using blender and going with Irfan's version, which looked better, so some of the work that I did won't show up as git commits because it was never committed. A similar thing happened with the smoke: I worked on a particle system, but Ahmad finished first so we wound up going with that one. - Ryan

Sources of some objects used

https://free3d.com/3d-model/watch-tower-made-of-wood-94934.html
https://www.turbosquid.com/FullPreview/Index.cfm/ID/911748
https://www.turbosquid.com/FullPreview/Index.cfm/ID/1199170