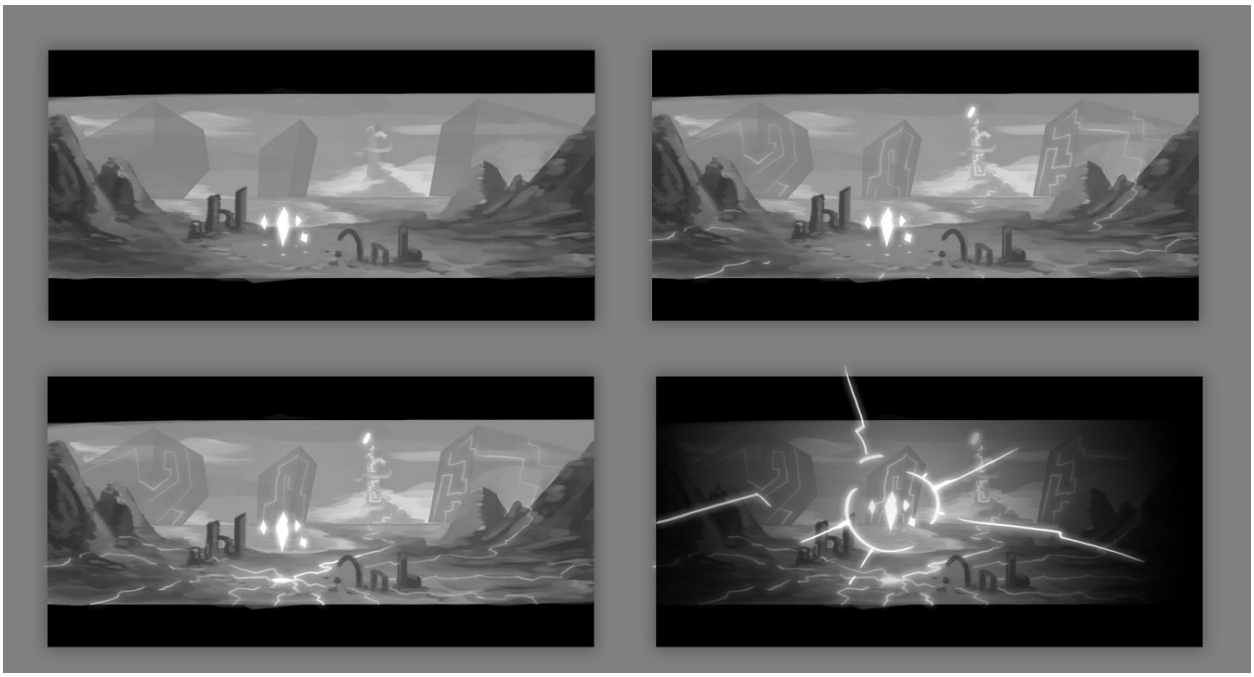
PRUITT-IGOE

A small procedural short

Design Document



CIS700: Procedural Graphics, 2017

Mariano Merchante

INTRODUCTION & GOAL

The motivation of this project is creating an immersive audiovisual experience that takes the audience to the awakening of a magical, cosmic entity inspired by Lovecraft's psychological terror. It is inspired by many different creations from contemporary media, as well as my own concept art.

A raymarched terrain is going to be explored as a flying camera goes by, and different cataclysmic events are going to happen as the audience traverse it. A specific musical piece done by Philip Glass called Pruitt-Igoe is the selected background music, which will guide the choreography of visual stimuli.

In particular, the objective of this project is twofold: one is the creative construction of an immersive audiovisual experience, but a secondary and more technical one is the validation of the plausibility of a specific raymarching optimization that uses temporal coherence, as there's no current method that considers this at the time of writing. Because the length of this project is short, I don't consider necessary to develop a robust method that considers all arbitrary cases; as long as I'm able to validate it as an optimization method for this short I'll consider it a success.

REFERENCES

I've built a moodboard that can be viewed HERE and I'll continue to update as I found more references. It contains a lot of visual references but note that each image may be referenced for mood, style, or a particular thing that sparks an idea, not the image itself. It also contains some illustrations I did a few years ago.

Other references include:

- Elevated, by rgba
- Rainforest, by iq
- Journey tribute, by Shakemayster
- Clouds, by iq
- Hell, by iq
- Flames, by iq
- Space curvature, by iq
- Combustible voronoi, by Shane
- Entangled Vines, by Shane

SPECIFICATIONS

The application will be a pure C++ executable that uses SFML as framework for generating a portable OpenGL context. I will also extensively use demoscene code and techniques, such as mercury's HG_SDF library for distance fields.

The main work is going to be done through fullscreen shaders, although I'm still evaluating if the application should precompute some useful information, such as prebuilt Perlin textures or the terrain heightmap.

For the terrain raymarching, considering that it will be static, it would be a nice optimization to have a voxelized approximation data structure that gets rasterized for faster distance checks. Think of this as rendering a Minecraft-like scene, and using the depth buffer as the origin position for the raymarched rays.

Other similar optimizations will be used, as I will continue researching on implementations done by the demoscene.

CHOREOGRAPHY BREAKDOWN

A link to the musical piece can be found HERE, however, a better quality file is preferred. Although the piece is long (~7 minutes), it is built in such a way that additional elements can be introduced to the audience without losing attention.

It has 2:30 minutes of buildup time, in which the camera will fly by different ancient rock artifacts around a fictional world that get activated based on the intensity of the music. The camera will focus a specific one at the end of this section, which will be identifiable by having a more ritualistic design (rock placement, etc)

After this section, an immense pillar of light and magic will shoot into a volumetric vortex in the sky, which the camera will focus. Because there are awakening artifacts around the world, there will be around 1:30 minutes of camera flyby through the previous introduced places, which will also activate in a similar fashion along with music.

At 4:00, gigantic silhouettes of cosmic monsters will start to emerge from the horizon as the dramatic voices come in; details must be as vague as possible to keep the audience thrilled. The camera will continue to show different versions of these on each location.

At 5:30, the camera starts to move away from the locations, showing top down sections of the land while moving away towards space. The fictional planet is then focused at the center of the frame, showing the magical pillars come out of it as strings of light.

At 6:00 the music comes to a halt, and the pillars completely disappear. As the voices calm down, the planet is engulfed in a black atmosphere, while waiting until 6:50, when high frequency noise starts to take in until the piece completely fades off. At this moment, the silhouette and some details of a cosmic face (resembling Marvel's Dormammu) approaching the planet is shown, implying the ritual has been finished and the entity has awoken, maybe to consume the planet.

TECHNIQUES

- Raymarching, sphere tracing and other optimizations.
- Volume marching for clouds, vortex and magic pillars.
- Terrain generation and marching.
- L-Systems for texture generation.
- Post processing effects such as glow or tonemapping.

DESIGN

The application will be structured in four sections:

- Main C++ engine, which handles OpenGL context and shaders
- State machines that will handle all client logic (e.g. the choreography of parameters)
- Shaders that will actually render each frame.
- CPU side procedural generators for precomputed data.

RISKS

The project has some risks that must be considered.

- The music piece is too long: it can be either shortened with an audio editor or switched by another piece, although finding something that matches the mood will be hard.
- The effects are not efficient and cannot run in realtime: either scaling down the quality, or rendering in a reduced resolution and upscaling by some of the modern approaches, such as checkerboard rendering.
- The scope is too big: I'm not looking for a photorealistic style, so there are a lot of design decisions that can reduce work time.

- Initially I plan on having no vegetation, but without it the terrain may look boring. However, vegetation is ***hard***, so I'll need to find alternatives.

TIMELINE

WEEK 1 (11/4)

- Raymarched terrain with camera flyby
- *Some* material design on the terrain
- Code architecture should be able to handle FBOs and passes

WEEK 2 (18/4) - **MILESTONE 1**

- Volumetric clouds and magic pillar
- Activation of artifacts and magical runes.
- Main terrain scene should be finished (no choreography yet)
- Optimization of raymarching ongoing

WEEK 3 (25/4) - **MILESTONE 2**

- Choreography of main sections of the short
- Secondary terrain locations
- Monster silhouette
- Planet rendering

WEEK 4 (3/5) - **DEADLINE**

- Choreography of secondary sections
- Cleanup, debugging, optimization
- Tonemapping
- Post processing effects
- Optimizations