Playstation Graphics

A simulation for retro-inspired effects

Brady Moore
CS: Game Design
UCSC
Santa Cruz CA USA
bjmoore@ucsc.edu

Terrence DuBois
CS: Game Design
UCSC
Santa Cruz, CA USA
tdubois@ucsc.edu

Georgio Klironomos
CS: Game Design
UCSC
Santa Cruz, CA, USA
gklirono@ucsc.edu

ABSTRACT

For our final project, we have created a Unity scene centered around the model of an original Playstation. This model will be affected by several HLSL shaders to create effects ranging from a wireframe of the model to the model taking on the look of a reflective CD.

In the backdrop of the scene is a simulated TV screen, which will display the Playstation One's console intro as well as snippets of gameplay from games released on the console. The screen will be modified by several post-processing effects, such as scan lines and chromatic aberration, to produce the simulacrum of a CRT screen.

The user of the executable will be able to choose one of five games to place into the console. Once placed in the console, footage of the game will begin to play on the backdrop screen, and the console's shader will change to that of an effect representing the game.

CCS CONCEPTS

Computing Methodologies - Computer graphics - Rendering - Non-Photorealistic Rendering

ACM Reference format:

Brady Moore, Georgio Klironomos, and Terrence DuBois. 2019. Playstation Graphics:A simulation for retro-inspired effects. Developed for Comptuer Graphics 163 *(CMPM 163). 2 pages.* https://github.com/terrydubois/CMPM163_FinalProject

1 Model Based Shader Effects

Our Wireframe shader that is applied to the PS1 works by first finding all three edges of the fragment. Then it calculates the distance from one edge to the other and colors in the pixels around the edges minus however big the distance we want to stay away from to the other edges. This creates gaps within out fragments and solid lines along the edges resulting in a manipulatable wireframe shader.

Figure 1: **Wireframe Shader on console** [1]

The Hologram shader works by making a copy of the incoming vertex data for each fragment and then it scales them accordingly based on the normal direction. This is then passed into the fragment shader and each fragment is colored allowing for a solid fill with an outline along the edges. Finally Z-test is turned on to allow transparency and both sides of the object to be seen.

Figure 2: **Hologram shader on console** [2]

Our vertex Displacement shader takes the xyz position of all the fragments and then adds an _Amount value to them. Then when this information is sent to the Vert() function it is displayed with large gaps within the model to show it breaking apart into its components,

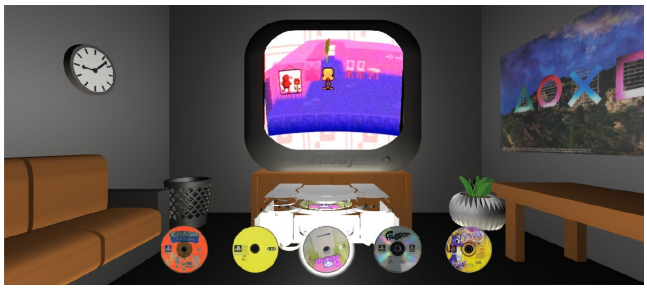

Figure 3: **Vertex displacement shader on console** [3]

The CD-ROM shader produces the effect of iridescent reflections using diffraction grating. The shader mimics the surface pattern of a CD in order to diffract light wavefronts in a way that simulates a rainbow. First the shader creates its own light spectrum, as designed by Alan Zucconi, and produces amplified, circular light waves based on a gating equation.

The Molten material uses the same glitch shader as the post-processing effects put onto the video screen, taking advantage of it's sinusoidal vertex displacement.The material colors it orange and speeds up the effect, mimicking a flickering flame. Added on is two layered flame particle effects, giving the top of the console the whispyness and dimension of a flame.

2 Post Screen Rendering Effects

The TV screen in the center of the scene uses a Render Texture to apply shader effects to an image captured by an alternative scene camera. A camera is placed facing a quad that uses Video Textures to display gameplay discs of five different Playstation games, plus two additional clips for the Playstation logo intro and the static clip. This camera has a C# script that requires another video, that being the source of the noise/glitch effect that will be rendered to the TV. That noise source video becomes one of the two video textures for the TV screen, the other being the gameplay footage. The frag shader for the glitch effect samples these two videos and displaces some of the texels by a subtly random amount, resulting in a distorted, VHS kind of look. The screen has another shader on it, one that curves the appearance of the mesh. The vertices are displaced according to a sine function, resulting in the screen looking curved like a CRT screen from the 20th century. An example of the Render Texture after the noise/glitch effect and the curve effect is pictured below.

REFERENCES

[1]https://github.com/Chaser324/unity-wireframe/tree/master/Assets/Wireframe/Shaders
[2] https://wiki.unity3d.com/index.php/Silhouette-Outlined_Diffuse
[3] https://docs.unity3d.com/Manual/SL-SurfaceShaderExamples.html
[4] https://github.com/hecomi/uCurvedScreen
[5] https://github.com/staffantan/unity-vhsglitch
[6] https://freesound.org/people/newagesoup/sounds/344536/
[7] https://freesound.org/people/kyles/sounds/453549/
[8] https://freesound.org/people/InspectorJ/sounds/376415/
[9] https://www.alanzucconi.com/2017/07/15/cd-rom-shader-1/