Samantha Liu, Hannah Norman
SUNet IDs: szwliu, hnorman
Professor Fatahalian
CS 248A – Computer Graphics
19 March 2024

Final Project: Pixelated Toon Shader

Summary of Functionality

We completed the shader project as described in https://github.com/stanford-cs248/shading, which includes Phong illumination, normal mapping, environment lighting, and shadow mapping from spotlights. Inspired by the game *A Short Hike,* we also implemented some extra features. The first is a pixelation filter, whose strength (from 2x to 16x) can be controlled by the user through keyboard inputs. In addition, we implemented stylized toon shading to replace the Phong reflectance in the original shader.

In order to implement the pixelation filter, we needed to extend the rendering pipeline. Instead of rendering the scene directly to the viewport, we render the scene to a texture buffer. Then, we render the scene inside the texture buffer to a quad that spans the entire viewport. We added simple shaders for this post-processing filter effect, where the vertex shader simply passes the texture coordinates to the fragment shader, which applies a simple downsampling scheme to give the final output a pixelated look. To toggle pixelation, press ‘p’ or ‘P’ and then use the ‘+’ and ‘–’ keys to increase and decrease the pixelation strength. As mentioned above, pixelation begins at 2x strength, and is capped at 16x.

In order to implement the toon shader, we considered what would make the scene look hand drawn, which led us to remove normal mapping and replace the Phong BRDF inside of the shadowed shader that we had built. Instead of having diffuse and specular components, we added reflectance to the BRDF according to the angle between the light direction and the surface normal. We clamped the dot product of the two vectors to fine-tuned thresholds in order to achieve stratified coloring across the surface. We kept the smooth shadows from the spotlights so that the toon shading appears as part of the art style, while the shadows still look realistic.

To demonstrate the pixelated toon shader, we created our own space/pirate scene using 3D assets available for free download online (see *References* section below). We downloaded both asset packs and manually triangulated all of the available `obj` meshes within Blender since their original format utilized quads instead, which do not render correctly in our program. We include the assets that we used in the submission under `media/toon/assets`. But since we spent time triangulating the rest of the packs' assets and recognize they could be of use in future iterations of this course, we will link them here. You can run the toon shader by calling `./render ../media/toon/lowpoly_toon.json.`

References

A Short Hike videogame trailer: https://www.youtube.com/watch?v=usBVx4J4CUM
Toon shading tutorial: https://www.lighthouse3d.com/tutorials/glsl-12-tutorial/toon-shader-version-ii/
3D pirate and space assets: https://quaternius.com/index.html
8-bit pirate theme: https://www.youtube.com/watch?v=hcdgRmbVP_g

Example Assets

Figure 1. *Standard Shader* (Phong illumination, normal mapping, environment lighting, & shadows)

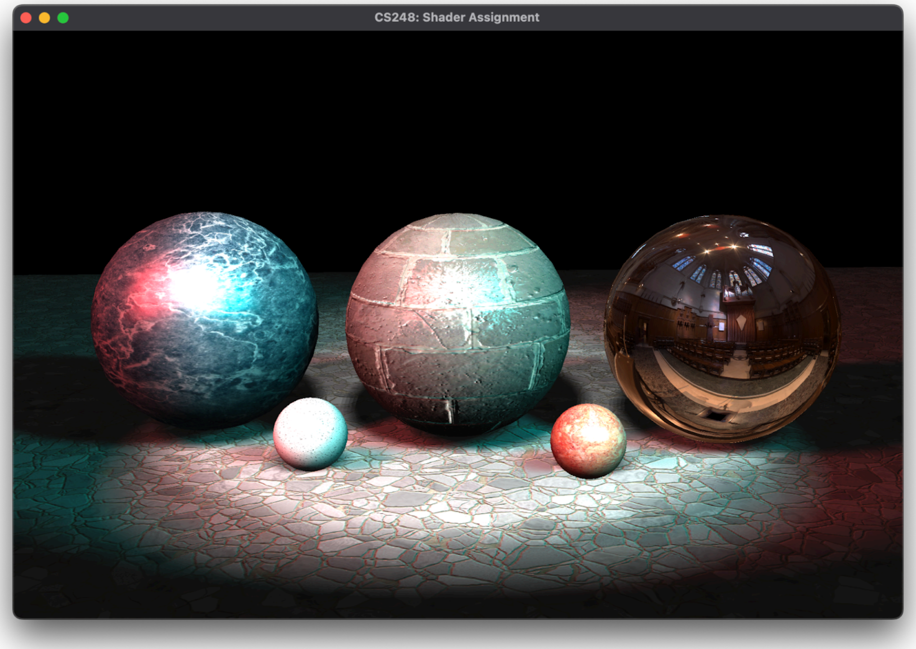


Figure 2. *Toon Shader*

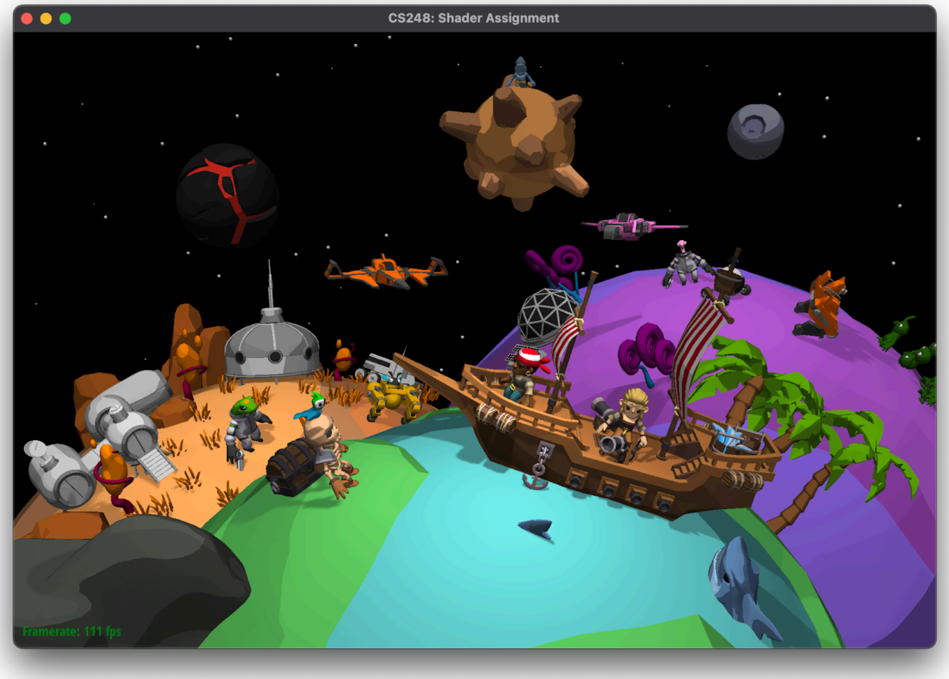


Figure 3. *Toon Shader + Pixelation Filter (2x)*

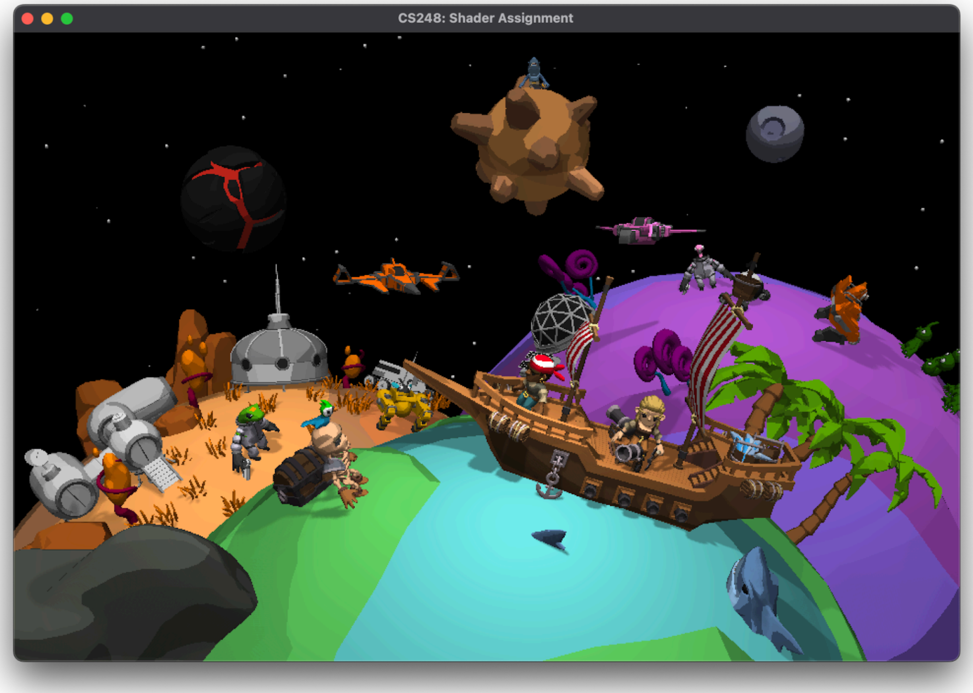


Figure 4. *Toon shader + Pixelation Filter (4x)*

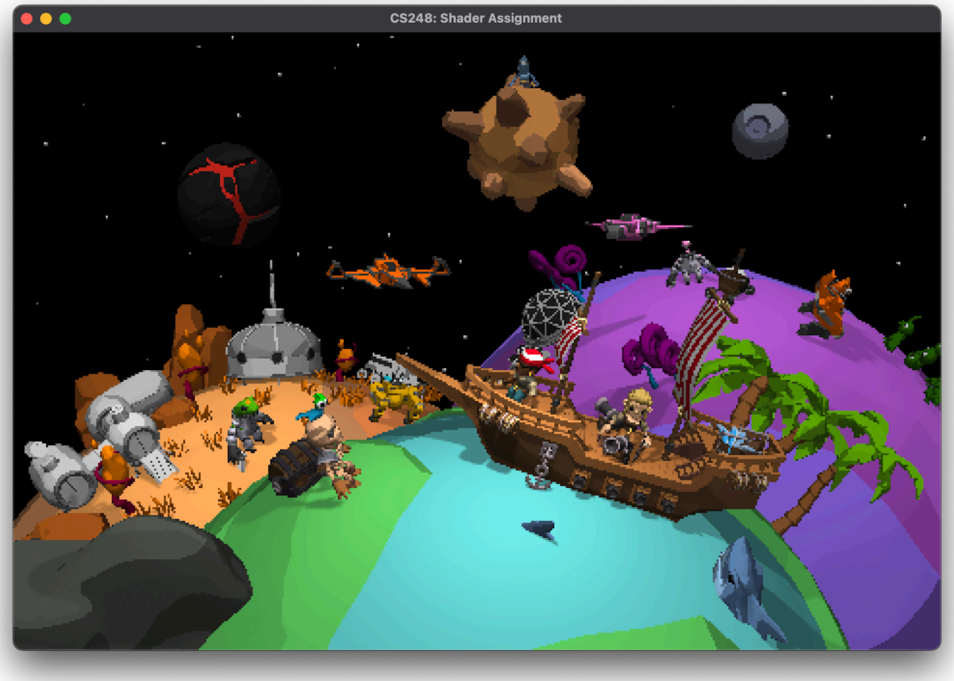


Figure 5. *Toon Shader + Pixelation Filter (8x)*

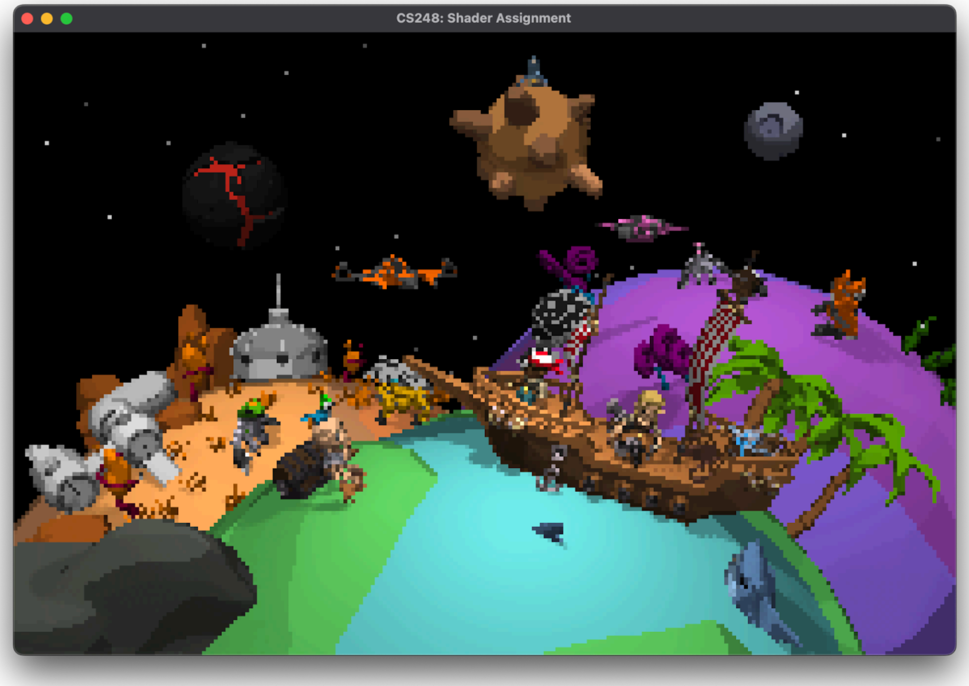
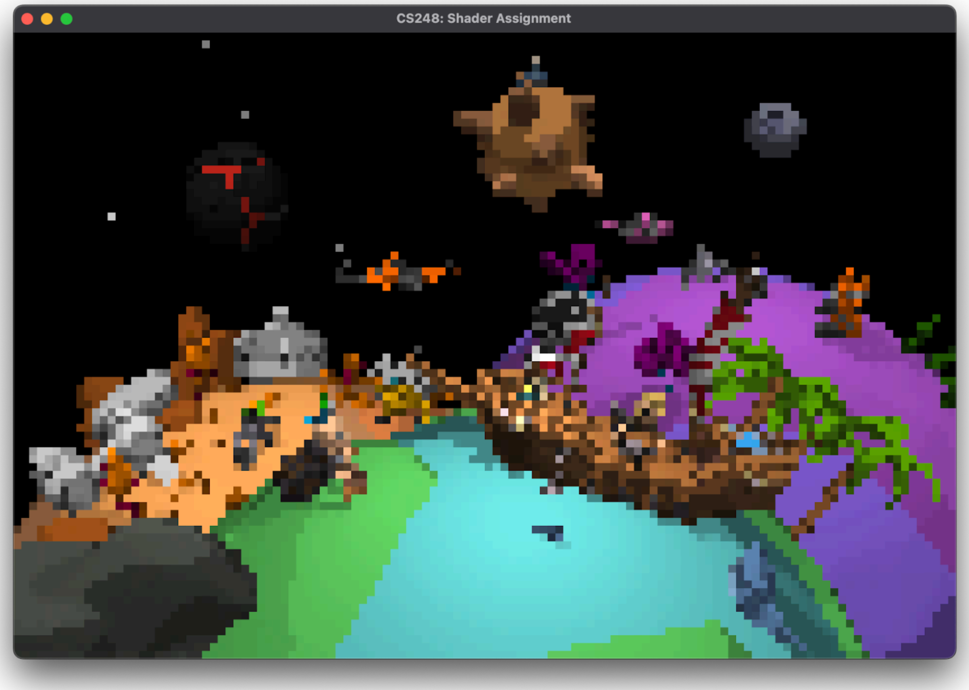


Figure 6. *Toon Shader + Pixelation Filter (16x)*


CS248: Shader Assignment