CS224 Project: Settlement

Released: Feb 1
Due: Feb 16, 11:59pm

Introduction

In this assignment you will use procedural modeling, lighting, and animation techniques to render a complex scene in an OpenGL Fragment Shader. Like the “Ball” assignment, you will do this in ShaderToy. The scene will be an outdoors-view of a human settlement, and include elements of nature.

This assignment contains aspects of physically-based and expressive rendering, interaction, animation, geometry, physics simulation, image editing. You will combine all these together into something that is not only technically impressive, but also aesthetically pleasing.

Example: Fishing Village [https://www.shadertoy.com/view/4dVGzD](https://www.shadertoy.com/view/4dVGzD)

Requirements

Shader

Make any kind of human settlement (a place where humans live), subject to the following constraints:

1. You may not make the same type of settlement as someone else in the class (including the TA, so that means you can’t make a fishing village). Post the type of settlement you intend to make to Piazza to stake your claim.
2. You may not make an urban city full of skyscrapers. The reason behind this is that city buildings are too big relative to everything else (cars, signs, etc), so it becomes hard to see anything other than the building skyline.

Beyond that, you have the freedom to make any settlement you like: it can be futuristic or ancient, inhabited or destroyed, painterly or physically rendered.

If you have trouble deciding what scene to make, you can choose one of these ideas below:

<table>
<thead>
<tr>
<th>Settlement</th>
<th>Nature</th>
<th>Reference Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukul Village with thatch roofs</td>
<td>Vegetation, dusty atmosphere, rocks</td>
<td><img src="image1.jpg" alt="Reference Image" /></td>
</tr>
<tr>
<td>Inuit Igloo village</td>
<td>Snow, Aurora Borealis, starry night sky, Mountains</td>
<td><img src="image2.jpg" alt="Reference Image" /></td>
</tr>
<tr>
<td>Campgrounds (Tents, campfires)</td>
<td>Forest, Crepuscular rays through the canopy</td>
<td><img src="image3.jpg" alt="Reference Image" /></td>
</tr>
<tr>
<td>Oregon Trail Pioneers (covered wagons)</td>
<td>Prairie grass, vast landscapes</td>
<td></td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-------------------------------</td>
<td></td>
</tr>
<tr>
<td>The Shire</td>
<td>Grass, trees, various colorful flora in the gardens</td>
<td></td>
</tr>
<tr>
<td>Santorini, Greece</td>
<td>Atmosphere, ocean, sea wind</td>
<td></td>
</tr>
<tr>
<td>Castle in the Sky (Laputa, Bespin, etc)</td>
<td>Clouds, Trees, Vines</td>
<td></td>
</tr>
</tbody>
</table>

*All the subsequent images in this handout also qualify as settlements that you may attempt.*
Documentation

In addition to your shader, you must provide a README file in PDF format (you can use MS Word, LaTeX, etc), describing how your shader was implemented. Someone with knowledge of GLSL, raymarching, and CS123 should be able to reproduce your shader completely from scratch, using only the rendered output as a reference and your documentation.

For this assignment, performance is more important than purity of code organization.

There are two reasons why documentation is required in this assignment:

1. It proves to us that you wrote the code, or at least have a complete understanding of how to write it
2. It helps us figure out how technically complex your code is. We give you more points if you invent a clever technique.

If your explanation in the documentation insufficiently explains what the function in the code is actually doing, you will receive no points for that feature. You may use drawings/powerpoint slides to aid your explanations.

Contents of the documentation:

● Any reference images you used or sketches you made
● Still image of your shader.
● Description of each global variable / macro
● For each method:
  ○ Function signature
  ○ Description of what each input and output is used for
  ○ If you didn’t invent this on your own, where you got the idea from.

You do not need to provide documentation for any methods used in IQ’s distance functions article (http://www.iquilezles.org/www/articles/distfunctions/distfunctions.htm). Ask a TA if you are unsure of whether your documentation is sufficiently understandable.

Collaboration

This assignment is hard, so we’re relaxing the collaboration policy slightly. However, violations of this policy will still be treated seriously. Read this section carefully.

Allowed:

● Sharing and discussing high-level pseudocode & math of how techniques are implemented on the Google Group.
● Posting rendered images and asking your classmates for suggestions - “Does this hut look weird to you?”. Note: only share the image, not the actual code.
● You may optionally work on this project in pairs. If you do, we expect pair-programmed shaders to be about twice as pretty and twice as clever as a shader made by one person. Each partner must write their own README on their own. Furthermore, you must list who you worked with and how much each partner contributed to the project.
• Copy-pasting (verbatim) any of the functions from http://www.iquilezles.org/www/articles/distfunctions/distfunctions.htm
• Studying the TA binary, existing shaders on ShaderToy, and online articles to learn techniques, then re-implementing those algorithms on your own. Make sure you understand why the technique works before implementing it. You still need to provide documentation for these.
• You are welcome to make your shader publicly viewable on ShaderToy 3 days after the assignment due date - you should be proud of this work!

Not Allowed:
• Copy-pasting other people’s code (other than IQ’s distance functions) into your shader.
• Making any part of your shader source code public on ShaderToy before the assignment is due. This makes it too easy for others to violate the collaboration policy.

Grading

This assignment is out of 100 point. Points are awarded additively - you will receive credit for each feature that you implement and document. It is possible to receive more than 100 points on this assignment. We prefer that you do a few things really well and make an aesthetically pleasing shader than to try to score as many points as possible. All point values “X pts” should read as “up to X pts”.

For an C grade (70/100), implement the minimum requirements + at least 10pts extra credit.
For a B grade (80/100), implement the minimum requirements + at least 20pts extra credit.
For an A / A+ grade(90/100 +) implement the minimum requirements + at least 30pts extra credit.

Minimum Requirements
• 10 pts - Visually interesting composition, atmosphere, & lighting. We won’t be grading you on technical art stuff (i.e. rule of thirds, color palette) but your scene should be nice to look at.
• 10 pts - Settlement resembles reference images or real life.
• 10 pts - The non-settlement components (nature) are believable in context of the rest of the scene.
• 10 pts - Efficiency of implementation
• 10 pts - Multiple levels of detail (mix of large objects, medium-sized objects, small objects).
• 10 pts - Lighting and shading.

Extra Credit Category

• 10 pts for each physical phenomena.
  ○ Examples: fire, water, wind, smoke, rain, fog, snow, caustics, tidal forces.
  ○ Stylized depictions of physics are fine, as long as they are convincing and are consistent with the style used in the rest of your assignment.
Relatively simple effects like “soft shadowing” do not count as “physical phenomena” for this assignment.

- 15 pts - Mimic a famous rendering style. A big part of implementing this is identifying the “limited palette” used by the artist. Monochrome is OK!

- 10 pts - Add animated wildlife/humans (flock of birds, grazing animals).
● 10 pts - Tell a simple story using your shader. It doesn't have to be complex story, and you can accomplish this without using any words or people. We should be able to guess what happened in the scene, just by looking at the shader.

“this village was burned down by the ravages of war”

“What do pigeons, airplanes, houses have in common?”

● Interactivity

  ○ 10 pts - Day/night variation (sun rising and setting, stars appearing).
  ○ 10 pts - Navigating the scene (first person perspective flying through / exploring the scene)
  ○ 10 pts - Build a little mouseover-control panel at the bottom of the shader, and use that to influence shader behavior.

If you do something cool not listed here, mention it in your documentation and we’ll take it into consideration for additional extra credit. Ask a TA if you want to run an idea by us.

Penalties:

● Your grade will be deducted 10pts if it fails to run on ShaderToy on an average Sunlab machine (which has a GeForce GTX 460 graphics card). GPU behavior may vary across Windows/Mac/Linux. Make sure to test your shader works on one of these machines. Some machines in the dept. have different graphics cards (row 6 of the Sunlab has GTX 970 cards).

● Points may be deducted for aliasing, flickering, and other ugly visual artifacts.

Handing In

Your handin folder should contain the following:

  ● README.pdf containing writeup and private ShaderToy URL.
  ● Source code of shader in a text file.

Hand in by running the following command in the terminal.

`cs224_handin settlement`

Hints & Resources:
• The first project and first 2 labs will provide an intro to shader programming for those not familiar with GLSL. However, you will need to spend the time learning some techniques on your own. Start as early as possible on this assignment.

• Help each other out! Discussion on Piazza about how shaders work is encouraged. As per the “community service” clause of the course syllabus, we award extra credit if you explain how a complicated procedural technique works on Piazza:
  ○ Example: how a binary search algorithm can be used to increase accuracy during height field tracing (with an illustrated diagram).

• A useful macro to have around is \#define clamp01(a) clamp(a,0.0,1.0). Underflow and overflow behaviors vary depending on graphics cards, so always do stuff like clamp01(dot(n,dir)). Test your shader on the dept. machines well in advance of the due date.

Good luck on this assignment, and start early!