CMPM 163 Notes

Malcolm Riley

Winter 2019 — May 2, 2019

Quiz Review

- Quizzes are not worth 20% of the grade, but 10% - the syllabus will be updated accordingly
- Quiz Answers are below:

1. Textures are used for: Heightmaps, Normal Maps, Noise Textures, storage space for texel data, and also just as textures!
2. **Attribute Data** is data that is specific to each vertex or fragment (but not necessarily unique to that vertex or fragment), whereas **Uniform Data** is data that is explicitly non-unique to each vertex or fragment. These keywords are optional in Cg/HLSL
3. The term **Programmable Render Pipeline** refers to the fact that part of the functionality of the rendering process is mutable through a software program. The **Vertex Shader** is a program that handles transformations per vertex, and the **Fragment Shader** is a program that handles transformations per fragment (roughly meaning, per-pixel)
4. The **Box Blur** effect can be accomplished by examining the eight neighboring texel values and averaging the current fragment value with them.
5. (Smiley-Face drawing not covered in class)
6. Diffuse contribution: $\hat{L} \cdot \hat{N} = \sqrt{2} \approx 0.707$, where L is the vector from the vertex to the light source, N is the surface normal, and the hat implies that the vector has been normalized.
7. This question was a gimme, no wrong answers

Particle Systems

- Instead of rendering a mesh of an object, there will be a group of **Particle** meshes that are all rendered using the same shader.
- Offline systems typically render many times more particles than real-time applications - for the simple fact that they don't need to be rendered in real time. The computationally-expensive processing is performed ahead of time, then rendered.

- In Unity, it is possible to attach particle systems to each-other; particles that spawn other particles on death, or particles that have particle trails, etc.
- Generally, a particle is rendered as a **Billboard**: Two triangles forming a quad that has been textured. These **Billboard** meshes are instructed to face the camera; this is the distinguishing characteristic of a **Billboard** particle that is not necessarily true of other types. **O**ne of the primary optimizations that enable particles is the fact that the resources used to render, draw, and maintain them are preallocated. Individual particle instances are also typically pooled.
- Unity permits the attachment of shaders to particles in the same way as it attaches shaders to ordinary meshes: through materials! Furthermore, the particle and trail may use separate materials
- In Unity, particle systems may also have sub-emitters triggered by particle lifetime events