CS-AD-216: Foundations of Computer Graphics

Assignment 7, Due: November 10

Instructions:

- Assignments can be submitted in groups of at most three. The purpose of groups is to learn from each other, not to divide work. Each member should participate in solving the problems and have a complete understanding of the solutions submitted.
- Submit your assignments as a zip file (one per group).

Problem 1 (10 points)**.**
In Problem 2 of assignment 6, you wrote the code for modeling a portion of a surface $z = f(x, y)$. The goal of this assignment is to make the surface time varying and add lighting to it. Modify the earlier code as follows:

- Replace the function $f(x, y)$ by a function $f(x, y, t)$ which also depends on the current time t.
- Write a funtion `Grid(n)` that returns an $n \times n$ grid as an object `Obj` with the `positions` and `triangles` set appropriately so that we can pass it to the function `objInit` defined in the file `Object.js` in the `Common` folder. The z coordinates of all the vertices in the `positions` array can be set to 0. Use the `draw` function attached by `ObjInit` to draw the object.
- In the vertex shader set the z coordinate of the vertex $(x, y, 0)$ to $f(x, y, t)$, where f is some function and t is the current time.
- Write appropriate code in the shaders so that Phong Lighting and Shading are used. In order to do this, you will need to compute the normal vector to the surface at the point (x, y, z). It can be shown that the normal is given by a unit vector along the direction $(-\frac{\partial f}{\partial x}, -\frac{\partial f}{\partial y}, 1)$ where the derivates are computed at the point (x, y, z). Instead of computing partial derivatives, we can use the following approximations:

$$\frac{\partial f}{\partial x} \approx \frac{f(x+\delta, y, t) - f(x, y, t)}{\delta}$$

$$\frac{\partial f}{\partial y} \approx \frac{f(x, y+\delta, t) - f(x, y, t)}{\delta}$$

  where δ is a small value. You can set δ to 0.01 in your code.
- The material settings for the entire object should be the same.
- Your code should allow moving the object with a trackball. It should also allow moving around the camera using the controls used in Problem 1 of Assignment 6.

Problem 2 (10 points)**.**
Write appropriate code in the function `computeNormals()` in the file `Object.js`. The function should construct an array `Obj.normals` so that `Obj.normals[i]` is the normal computed at the vertex with position `Obj.positions[i]`. The normal at a vertex should be computed by taking a weighted average of the normal vectors of the triangles adjacent to the vertex. The weight of the normal coming from a particular triangle is equal to the area of that triangle. The triangles in the `Obj.triangles` array are encoded so that the vertices are in counterclockwise order as seen from the visible side of the triangle. Test your code by loading models using the code in the directory "`Load Model`".