

## Assignment 6: Texture Mapping, Procedural Landscape Generation and Boids

Due Tuesday Nov 22, in groups of 1 or 2.

**Overview:** For this assignment, you will improve your boids program by adding lights, materials, texture mapping and landscape generation to create a more realistic scene.

First create a procedurally modeled island. The island landscape should be rendered as triangles in a height field; each vertex having a unique x,y, with z provided by a fractal function. Compute everything you need to render once, when you generate the island. During your rendering loop, you should only be responding to user interaction and rendering using pre-computed data.

To construct a fractal mountain, you start with a triangle or set of triangles. At each step, you split each existing triangle into four new triangles. This is shown in Figure 1 below with the new vertices marked with green dots. A simple splitting places the new vertices at the midpoints of the edges (as shown on the left of each pair in the figure), you will then perturb these new vertex locations vertically by some random amount (as shown on the right in each pair in the figure). At each step, reduce the scale of your random perturbation by  $1/2$ . This sequence gives the characteristic fractal self-similarity: at half the scale, you have half the noise amplitude. Sometimes when modeling natural processes (as we are here), this called  $1/f$  noise because the noise amplitude is proportional to  $1/f$  for noise frequency  $f$  (controlled in this case by the geometric splitting).

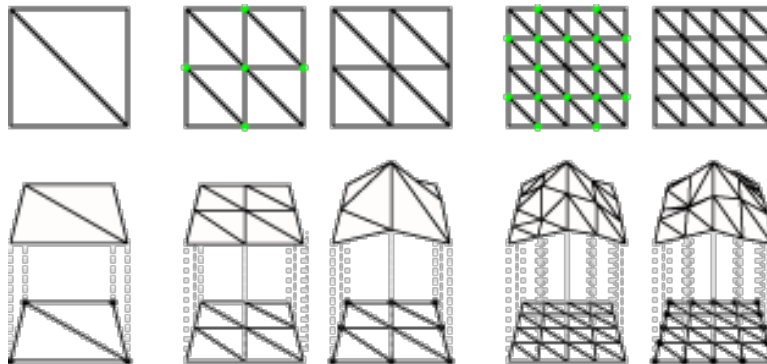


Figure 1: Procedural creation of islands

Intersect your generated landscape with a blue rectangle to simulate the ocean. The base triangles of your island should be below sea level. The shape of the shoreline will be determined by the contour where the sea polygon intersects the island. To make sure that the edges of your island don't pop up above the level of the sea; you should clamp all edge vertices to make sure they don't protrude above sea level. In Figure 1, the points marked with black dots can be allowed to move, but not above the level of the sea. Figure 2 shows an example of such an island.

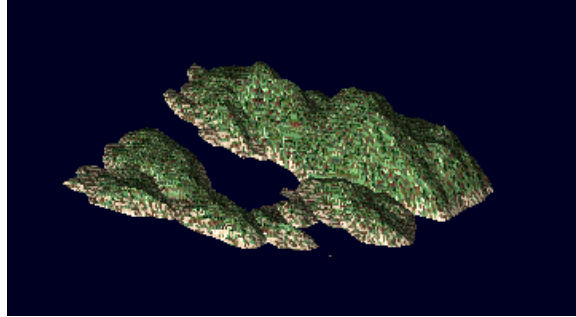


Figure 2: Procedurally generated island

Initially, your island should begin with 8 triangles (2nd and 3rd columns in Figure 1 above), which is the minimum necessary to make something resembling an island. You should choose a sufficient amount of subdivision (at least 7 levels deep) that gives a sufficiently good looking island from your flock's perspective. For a better looking scene, you might want to create multiple islands.

For lighting to work, you will need to compute surface normals at each vertex of your model. Compute these when you initially compute the landscape. Recall that the vertex normal is computed as the average of the normals for each face adjacent to the vertex. Be careful to make sure your normals point out from the island (not into it).

### Program Requirements:

1. Display the island in a variety of lighting conditions. Have your island go from dawn-day-dusk-night-dawn-day-... while the program is running. Time should cycle through a 24 hour day in approximately 2 minutes and repeat.
2. Implement a first-person view from one of the boids in addition to the usual trailing and side view. It is imperative that you maintain nice stable viewing parameters so that the fly-bys are not dizzying. If your boids have tendencies to cause wild camera swings (i.e. overshooting, lagging, scattering, etc), you must either tune to fix the flocking, or compute your camera parameters differently.
3. Default view can be centered over the islands.
4. Do not render boids goal any more.
5. Add steerable headlights to one of your boids (the one that the camera is on) that illuminate the scene while it's dark. At night, the island should be lit by a low level of ambient light with a higher blue component.
6. You may also add various stationary lights (street lights, house lights, etc.) as appropriate.
7. You can see that this program assumes that you have boids that are nicely steerable. If you can not get your boid movements to behave reasonably, it is okay to program some alternative way to fly-by your island world. Please discuss with me first.

8. There should be a way to zoom in and out on the island. Be careful not to allow zooming through the island! Pressing 'n' should return to the normal view. Automatically increase the triangulation as you zoom in for more realism.
9. Texture the ocean, island, and objects on it to make the scene look realistic. Texture the boids as well.
10. Add at least one object that is environment-mapped.
11. You only need to populate a small section of the island with objects; if you choose to do this, then your boids should always start in the same location. If you wish to populate the entire island with objects (logs, rocks, trees, houses, etc.), you may place them randomly.
12. When any of '1' to '3' is pressed, you should transport the flock to a nice location on a pre-chosen island (i.e., pre-set a random seed to create the island and a starting location for the flock that you think looks particularly nice) and start a pre-programmed flight sequence and view selection that shows off your flock, island, objects, lighting etc.
13. Create and submit a short video capture to demo your world. The video length should not exceed 5 minutes, but can be much shorter if you wish. Try to show off all features of your scene as best you can, along with any extra credit features if you successfully implemented them. At the minimum, you should record the pre-programmed flight sequences you programmed under '1'-'3'.
14. Include good documentation in your README on key/mouse commands needed to run your program.

**Strategy** As usual, the assignment is open-ended and there are many ways you can program this scene. Those details are up to you. Suggestions:

1. Plan your development with lots of intermediate milestones with visual results.
2. If you do not enjoy designing/constructing objects from scratch, google for suitable meshes to place in your scene. You already learned how to read an OFF file. Although not all meshes come in OFF format, converting to it is not difficult and MeshLab can handle conversions between most popular formats.
3. Having trouble getting the normals right? Turn off lighting and use the normals to colorize your landscape.
4. If you need to do these a lot, you can add new key handlers to change what you render.
5. Know how you're going to find the position, height, and orientation of the views before you start programming.
6. If you are not sure where the camera is, draw it.
7. Single-step debugging. I am not making you do it any more, but you really should.
8. If lighting is wrong, try shading with only ambient/diffuse/specular to debug.

### Extra credits:

1. Enable the creation of more complex islands by randomizing the starting center point of the subdivision (i.e., using non-even triangles) and/or creating multiple initial peaks. You may also find it useful to modify the noise amplitude denominator (e.g., try  $1/3$  instead of  $1/2$ , or use different amplitudes in different locations to vary the roughness of the terrain in different locations).
2. Color/texture the island based on the elevation of each vertex. For example, create snow-capped peaks and sandy beaches on top of a lush green island. Randomize the colors a bit between adjacent polygons (e.g. different shades of green and brown) or add textures for more realism. Make the terrain dependent on the direction of the normal (e.g., beaches in flat areas, make sharp rocky cliffs, etc.)..
3. Add scenery to the island. This could include fractal trees (generated procedurally!), fields, a flock of birds in the background, etc. You should either prevent flying through the scenery, or ensure that something cute happens when you do (e.g., boids plow down the trees or something). Maybe your boids can launch red spheres that eliminate the scenery. Maybe they can launch green spheres that randomly generate new scenery. Be creative!!
4. Add scenery to your world. Draw a sun and clouds for the day, or moon and stars during the night, random floating islands with hanging waterfalls (a la Avatar), giant mirror balls that are environment-mapped, etc
5. Experiment with more advanced textures, such as bump mapping, normal mapping, displacement mapping, etc.
6. Experiment with more advanced shading models beyond the basic Phong.
7. Add environment effects to your world, fog, waves, etc.