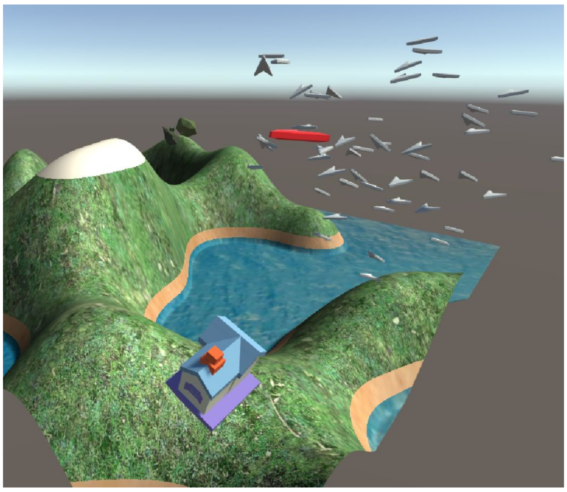
CSC 305 Assignment 3 Requirements

Due Sunday, Apr 7, 2019, 11:55 PM



In assignment 3, we will continue to work on the terrain project created from assignment 2, and add an animated boid flock that flies above the terrain.

Again, the assignment is divided into Basic Requirements and Advanced Requirements. Correctly completing all features the Basic Requirements part will award you a grade up to 70% of the assignment. To get 100%, you must selectively implement a few features in the Advanced Requirements part, whose solutions are not completely covered in lectures and requires original research. The Advanced Requirements are open-ended, and you are welcomed to propose and work on your own preferred features. If you want to work on an idea not mentioned in the Advanced Requirements list, you should come to discuss your own idea with the TA or the instructor; and point values will be assigned to your proposed feature after we discuss it.

After completing the Basic Requirements in assignment 3, you may opt to continue work on unfinished Advanced Requirements in assignment 2 and claim their points, if you prefer to study raster rendering to animation.

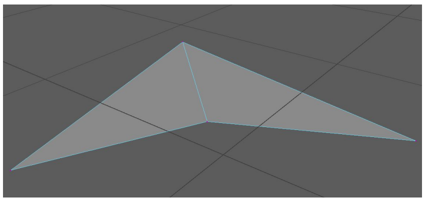
Basic Requirements (70%):

- Moving 1 model (red arrow in the demo) following a smooth cubic spline trajectory with a few hand-inputted control points above your Perlin Noise terrain. The animation should be loopable, that is, the end point of the trajectory overlaps with its starting point, and the movement of the model loops stably when you run the program. (Hint: you can draw the spline using Unity's LineRenderer to check if your spline looks correct. Hermit or Catmull-Rom cubic splines are recommended). **(20%)**
- Changing the interpolation to add geodesic distance interpolation such that the model is always moving at a constant world-space speed on the spline. **(20%)**
- Create a few (~50) models to form a boid flock (white arrows in the demo). Lead the crowd using the model moving on the spline. Tune the boids' parameters so the flock behaves realistically and stable over long simulation time. You will be asked for your Boid parameter tuning experiences during grading. **(30%)**

Advanced Requirements (up to 30%):

1. Write rotational transforms for all the moving models such that they always point their tips towards the moving direction (see demo) **(5%)**.

2. Add boid collision avoidance with the Perlin Noise terrain, such that boids don't crash into the terrain surface and they lift themselves up whenever they're too close to the terrain. Demonstrate this feature. (Hint: you can directly use the Perlin Noise height field for this calculation instead of calculating collision with the terrain mesh) **(10%)**
3. Add Hailing effect: cast a number of randomized rays from the sky towards the terrain. If the ray intersects with a boid, push the model to the direction of the ray (add a momentum following the ray's direction) to simulate a hail hitting a boid. You should also draw the hails with small cubes or spheres. **(10%)**
4. Create wing flapping animation for the flying boids. The simplest winged boid is only two triangles that share one edge (see pic). With an animated angle between the two triangles over time, an impression of flapping wings can be created. You may program this animation by hand or explore Unity's animation system to encode this for you (Hint: You need to put two more reversed triangles on the same position, otherwise the boid will disappear when observed from the other side, due to back-face culling). **(10%)**



5. Create an animated windsock using the generalized cylinder method we covered in class. Warp the generalized cylinder around a spline and animate the spline according to a randomized wind direction and strength. **(10%)**