Homework 02: Flight Simulator

DIRECTIONS

For this homework assignment we will be making a basic flight simulator over a finite, random, terrain. We will be flying something more like a quadcopter than an airplane as the camera won't fly forward on its own. This project will make use of meshes, perspective projections, lighting, and collision detection.

For this project you are given a minimal HTML document and an empty JS file. You must write the JS from scratch, copying the necessary parts from other examples as necessary. However, any unnecessary parts included should not be included and will cause deductions. You are also given 2 additional JS files that include code for various algorithms (including some functions done in class). Feel free to import this code (not copy-paste it but include the script).

The terrain is to be generated using the diamond-square algorithm which is a recursive algorithm. Read the documentation of the provided function to figure out how to use it. The terrain should be drawn with lighting so the contours and shapes of the terrain can be easily seen (however do not worry about shadows). Exact colors and shading are up to you, but it should look somewhat like a realistic landscape. The colors must depend on the range of heights in the terrain and/or the slope at the current location. The entire terrain should be drawn with a single call to `drawElements`.

The flyer should start in the middle of the random terrain slightly above the surface. Note that the random terrain is generated in such a way that the middle of it may be negative or positive thus you can't just start at a height of 0, you have to find the height of the middle of the terrain and place the flyer slightly above that. The controls for the flyer are as follows:

- Up/Down arrow: moves forward/backward based on the direction it is facing
- Left/Right arrow: rotates to the left/right (yaw)
- W/S: rotates down/up (pitch)
- A/D: rotates clockwise/counter-clockwise (roll)

The flyer is not allowed to pass through the terrain but nothing bad happens when it "hits" the terrain. Make sure that it also does not get stuck inside a piece of the terrain either. To assist you with collision detection you are given a function to detect if a line segment and a triangle intersect in 3D. Your collision detection should be efficient (i.e. do not examine every triangle to see if you will intersect it but instead just the ones that could possibly be intersected). If the flyer does not leave the defined area then the underside of the terrain can never be seen, however, since the terrain is finite in area it may end up leaving the region where terrain is defined and fly under the terrain.

The view from the flyer is to use a perspective such that no part of the terrain is ever clipped if the flyer is above the defined region and not absurdly far off the ground. The canvas should be full-screen and maintain the proper aspect ratio regardless of the dimensions of the window. Try to pick values that make it look "real" without too much fish-eye effect.

GRADING

For full credit, be sure to follow the directions above. There are 100 pts and they are broken down as follows:

- 10 pts for code formatting (although really bad code can go negative here)
- 15 pts for rendering the random terrain mesh
- 15 pts for providing appropriate colors and lighting to the terrain
- 15 pts for using an appropriate perspective view of the world
- 20 pts for appropriate movement of the flyer
- 25 pts for not allowing the flyer to pass through the terrain

GUIDANCE

Recommended order of working on this project is as follows so that you don't get overwhelmed or intimidated by the assignment:

- Mesh:
  - Draw the terrain mesh using techniques from class
  - Adjust colors so that it looks more like terrain and works with the ranges of heights produced
  - Add in somewhat realistic lighting
- Flyer:
  - Switch to using a perspective view that works appropriately
  - Remove the mouse handlers (if you have them) and add the keyboard controls of the flyer
  - Start the flyer in the middle and slightly above the terrain making sure it moves correctly
  - Add in collision detection so the flyer cannot pass through the terrain
- Check your code and make sure it follows all of the directions