Unit: INB381 Modelling and Animation Techniques

Problem Solving Task Part B
Due: Tuesday, 25th October 2015, 11:50pm
Group of 2 people (1 or 3 people teams need the unit coordinator's approval).

This assignment is worth 30% of your marks for the unit.

I Believe I Can Fly

For full marks in this submission you need to complete all five of the tasks below AND following good programming practice e.g. well-commented code, meaningful function and variable names etc. In addition to the functionality for each task, you need to submit a Statement of Completion that documents: the tasks you complete, how to use your software and typical screenshots that provide examples of your executing code. Also provide a short video demonstrating all the functions of you have implemented.

Your code should be able to be executed at least in two modern browsers (Safari/Firefox/Chrome/IE/Edge).

The theme of the following tasks is flying. What we want is a thingy that can fly around e.g. tinker bell, phoenix, flying dragon, flying horse, flying pig, flying cow, etc.

Task 1 for 50%

1. Complete all tutorial and practial questions of Week8-13. Organise the questions and answers in the order of weeks. Provide well-commented code for all coding questions. Also provide one or two typical screen-shots for the output of all coding questions.
1. Create some graphical objects for your flying thingy using Blender- at least a body, two wings, and a few other components such as head and limbs. Save the objects to a file using the Wavefront obj format (File/Export/Wavefront). From export options uncheck all and select only 'Write Normals', 'Triangulate Faces', and 'Objects as OBJ objects'.

   Note: you are NOT allowed to export your object using other formats e.g "`Collada (default) .dae`".
2. Write WebGL and JavaScript code to read the objects from file, storing information about vertices, vertex indices etc.
3. Write WebGL and JavaScript code that continually flaps the thingy's wings. To do this, you need to think of the flying object as a tree. The root of the tree corresponds to the entire object or the body of the object and its children to subcomponents of the flying thingy.
4. Wing flapping must be achieved by traversing the tree, updating information and then rendering by again traversing the tree to draw the component objects.

Task 2 for 65%

1. Complete Task 1

2. Add WebGL and JavaScript code to steer your flying thingy around the screen using the keyboard.
   - key 'a' for left, 'd' for right
   - key 's' for forwards
3. Your flying thingy should move at a constant speed in the direction of its motion (no flying backward) and never escape the world view.

Task 3 for 75%

1. Complete Tasks 1 and 2
2. Add material to the objects making up the flying thingy and a single point light to the scene so that shading of the wings can clearly be seen as they move. Lighting must be achieved by implementing the Phong reflection model in your own code, not using third-party libraries such as three.js.

3. Add functionality so that when the 'z' key is hit, keyboard control stops working and the flying thingy falls straight down.
4. Add a ground to your scene (flat is fine).
5. When crash-landed, keyboard control should be resumed.

Task 4 for 85%

1. Complete Tasks 1, 2 and 3
2. Add functionality so that when the 'x' key is hit, the keyboard control stops working and the flying thingy spirals towards the ground, always oriented in the direction of its motion.
3. The spiral path needs to have a wide radius at the top which reduces as the flying thingy continues to fly down until it reaches the centre of the bottom of the viewport.
4. The flying thingy should accelerate all the way down to your ground, then decelerate quickly over a short straight-line distance parallel to the ground plane before it comes to a complete stop.
5. When landed, keyboard control should be resumed.

Task 5 for 100%

1. Describe an additional feature and argue for it's inclusion in your Statement of Completion.
2. Implement your feature.

What and How to Submit

Your submission is to be uploaded to INB381 central server: fastapps04.qut.edu.au. The manual/instruction for accessing this server is listed in Week4 teaching materials on the BlackBoard site of this unit.

The files to include in the submission are:
1. A statement of completeness including an indication of how to run your program, program design (what your shaders do etc) and typical screenshots in Word or PDF format,
2. A copy of your source code,
3. A copy of any input files that are required by your program.
4. A short video demonstrating all the functionalities of your program.
5. A brief Statement of Contribution from the team members

Suggestions

- Don’t forget to triangulate your model (eg. Control t in edit mode in blender)
- Initially, just use the default orthographic projection, so keep your x, y and z values for your model in the range [-1, 1]. Then swap to a perspective projection and rescale the objects if necessary.