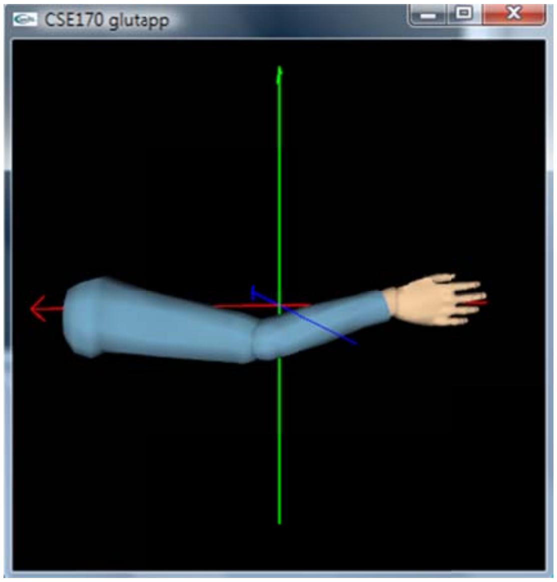
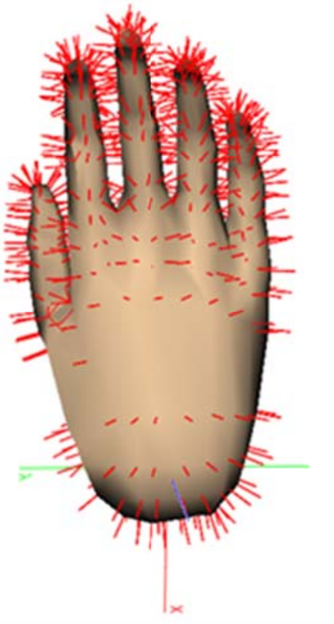
Programing Assignment #3

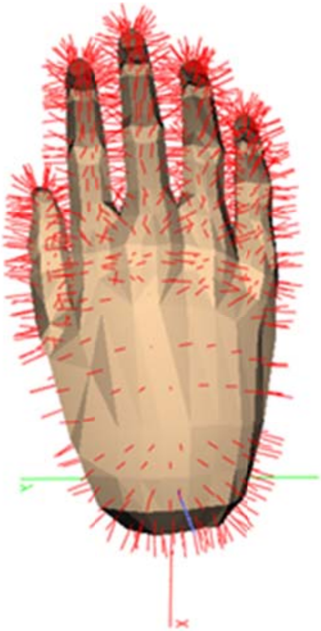
In this assignment you will load models to compose an articulated arm. The arm consists of 3 parts: the hand, the lower arm, and the upper arm. Each of these parts has a corresponding model file (.m) representing the geometry of the part. The model files are provided to you in the package **arm.7z** available in CatCourses.



The Three Models of the Arm Correctly Positioned



Hand Model Smooth Shading Normals



Hand Model Flat Shading Normals

The structure of the .m files is very simple. It contains a list of vertices, faces, materials and normal vectors. You are not required to understand the .m file, you will just load it using the GsModel class and display it.

Note: make sure you are able to display the arm in “smooth shading”; see GsModel methods set_mode(), flat(), and smooth(), among others, to explore different shading results. You can see the normal vectors (in the image above) that are loaded from the .m file. The right-most image shows normals computed for flat shading. While we will be covering shading in class, this assignment is still just about transformations.

For this assignment you have to accomplish the following:

Requirement 1 (30%) – Correct Display and Positioning of the Arm Models

Create a scene graph where each arm part will appear correctly as illustrated in the left-most figure above. Therefore you will need to use three instances of SnModel, one for each arm part to be loaded. Each SnModel contains a GsModel, which you can access and use to load the .m files. GsModel is a class that maintains a triangle-based model representation based on triangle lists. GsModel can be used independently of the scene graph. Use transformation matrices to place the parts in the correct locations such that they look like a real human arm, as you can see in the left-most picture above.

Requirement 2 (30%) – Rotations and Controls

After the models are correctly positioned to form the entire arm, you will then apply rotations at the joints. Compute and apply transformations such that you achieve correct rotations at each arm articulation:

'q' and 'a' - will rotate the shoulder articulation up and down
'w' and 's' - will rotate the elbow articulation up and down
'e' and 'd' - will rotate the wrist articulation up and down

Note that your rotations have to lead to a correct and meaningful control of the arm; for example, when the shoulder is rotated, the entire arm will correctly move together with the shoulder. To achieve this you will need to combine rotations and translations in the correct order, and correctly accumulate them with respect to each model.

Requirement 3 (30%) – Animation

The final requirement is to generate a simple animation of your arm. You will achieve this by interpolating rotation values of your joint parameterizations (previous requirement) in an animation loop that you will write. Any animation counts, it just cannot be a trivial animation:

- Have the 3 joints to move together.
- Make the animation to have at least three “key frames”: one key frame means one “pose”, which in our case is represented with at least the 3 angles of the previous requirement. So, do an animation at least from the current pose, then interpolate the angles towards a 2nd pose, and then towards a 3rd pose. For example you can achieve the arm moving up and down like shaking hands with another arm.
- Any animation with at least “3 keyframes” will be fine as long as it looks reasonable (do not just put random values in the rotations!)
- You should trigger your animation with a simple menu button “animate”

The previous PAs should have covered enough information for you to do this assignment without any major difficulties. Just start from the **sigapp.7z** project, build the scene graph that you will need, and then write your animation loop.

Requirement 4 (10%): Overall quality.

Everything counts here and, once again, there is no need to do anything complex, just make sure your project looks good and you will get full points.

Submission:

Please present and submit your PA as usual according to parules.txt. (Do not forget to upload your project before the deadline!)