Games and Simulation

2016/2017

Data: 20-APR-2017

Note: Don't forget the usual conventions: Points are represented by column vectors using homogeneous coordinates.

1. Consider a scene graph representing all the objects in a game and the cameras used in that game.

   A. Each Spatial *i* contains at least a Local Transformation (L*i*) and a link to is father Spatial in the graph. What is the purpose of this transformation? Give your answer making references to the Spatial itself and its parent.

   B. For performance reasons, Spatials usually also contain a cached World Transformation (W*i*). What does this transformation do to a point P? In which coordinate system is P given? What is the coordinate system after transformation?

   C. How can the transformation in World Transformation be computed for a certain Spatial using the transformations referred in A. (Local Transformations assigned to each node)?

   D. Camera nodes also have a Local Transformation L*i*, but it is usually specified in a different way than the other spatials. Also, cameras are direct descendants of the root element of the graph. What does this transformation do? Given an example of an appropriate method that we could use to specify the camera Local Transformation.

   E. Suppose you have a perspective camera that orbits around a given spatial at a certain distance. For that camera you have a controller that receives the given spatial and you are able to query its location in World Coordinates. What would you prefer to use for the specification of this camera, the lookAt() + perspective() functions or a system based in Phigs (VRP, VPN, VUP, PRP, window limits and front+back clip planes)? What parameters of the view specification would you change at each instant, to reach the intended orbital behaviour, and how?

   F. Suppose that a certain spatial of our scene is assigned a controller that checks whether the object has been hit by a projectile. The controller has access to the World Coordinates of the potential projectile and the hit detection should return true only if the projectile is inside the bounding volume of the spatial. This bounding volume starts by being an axis aligned box given by min/max values along each axis of the model. This box is affected by the transformations that are applied to the spatial exactly in the same way as its geometry. How would you program your hit detection routine?

2. Consider the implementation of a traditional 3D graphics rendering pipeline and the Phong illumination model formula below for multiple light sources:

$$\mathbf{I} = \mathbf{I}_a K_a \mathbf{O}_d + \sum_{l=1}^{L} f_{att,l} \mathbf{I}_{p,l} [k_d \mathbf{O}_d (\mathbf{N} \cdot \mathbf{L}) + K_s \mathbf{O}_s (\mathbf{N} \cdot \mathbf{H})^n]$$

   A. The view volume includes a front and a back clipping plane. Why are these planes important in the implementation of the pipeline operations? What other uses can they also have, not directly related to the implementation of the pipeline, but in the practical sense of a game programmer? Give some examples here.

   B. Specify two stages of the pipeline (and its corresponding coordinate systems) where illumination can take place. What parts of the given expression are sensitive to the coordinate systems that can be found in the 3D graphics pipeline? Justify your answer!

   C. Enumerate and describe the meaning of the entities involved in the ambient and diffuse terms only.