Hierarchical Modelling

- Modelling
  - Modelling and Rendering
  - Transformations

- Moving this model?
  - Change the transformations over time.

WHY?!?
Modelling

- Modelling a two-link arm
  - Rigid links
  - Hinges:
    - Upper arm link B has two joints p and q (shoulder and elbow)
    - Lower arm link A has one joint, r
    - Attach points p on B, q on B, and r on A.
  - Parameters to control:
    - Shoulder position T
    - Shoulder angle $\theta$ (A and B together rotate about p)
    - Elbow angle $\phi$ (A rotates about r, and stays attached to B at q)

Modelling

- Modelling a two-link arm
  - Start with A and B in their original positions
  - Apply only to A:
    - Translate by -r
    - Rotate by $\phi$ about the origin.
    - Translate by q, bringing r and q together.
  - We can now consider q as the origin of the lower arm link, and regard A as being in this coordinate system.

Hierarchical Modelling

- Store the modelling sequence in a hierarchy
  - Leave the geometry:
  - Internal nodes have transformations,
  - Transformations apply to everything under them – start at the bottom and work your way up.

Hierarchical Modelling

- Another view:
  - The shoulder coordinate transformation moves everything below it (i.e., the shoulder)
  - B
  - A and its transformation.
  - The elbow coordinate transformation moves A with respect to the shoulder coordinate transformation.

Modelling

- Modelling a two-link arm
  - Now the transformations apply to both A and B:
    - Translate by -p
    - Translate by T to place the two link arm at the proper position.

Modelling

- Note:
  - $\theta$, $\phi$, and T are parameters – we change these to animate the model
  - p, q, and r are structural constraints. If we change them – model falls apart.
Hierarchical Modelling

- Articulated Figures
  - Each node represents the geometry, rotation parameters and structural transformations.
  - Root can be anywhere – here it is at the hip.
  - A realistic human is much more complex
  - Difficult to control so many DoF's (later problem)
  - A Directed Acyclic Graph
  - Not necessarily a tree, as geometry can be transformed instances of each other.

Hierarchical Modelling

- Articulated Figures
  - Character Rigging and skinning

We can model a lot of things this way

Hierarchical Modelling

- We can model a lot of things this way

Hierarchical Modelling

- Doing this in OpenGL 2.x and earlier
  - Use the Matrix Stack
    - Current matrix is automatically product of everything already on the stack
    - This is the matrix on top of the stack
  - Recursive algorithm
    - Load identity matrix
    - For each internal node
      - Push new matrix into stack
      - Concatenate transformations onto current matrix.
      - Recursively descend tree
      - Pop matrix off stack
    - For each leaf node
      - Draw the geometry using the current transformation
Hierarchical Modelling

- Doing this in OpenGL
- Using VAO, VBO and shaders

http://www.gamedev.net/reference/articles/article1267.asp