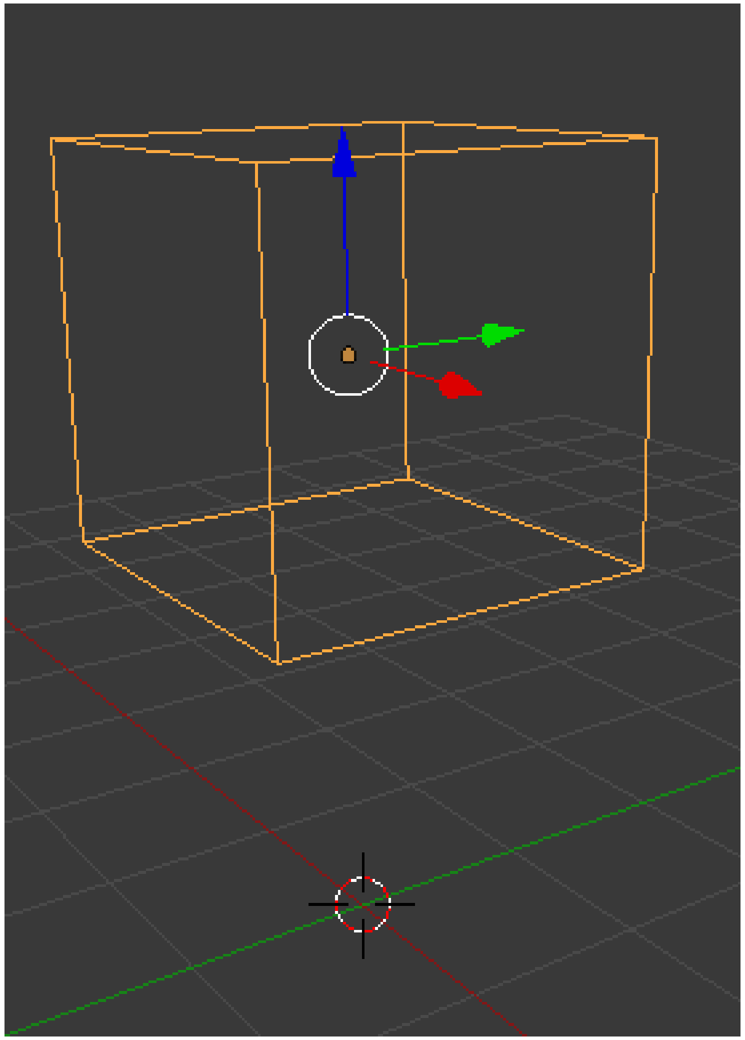
Model Position and Orientation

Models (Objects)

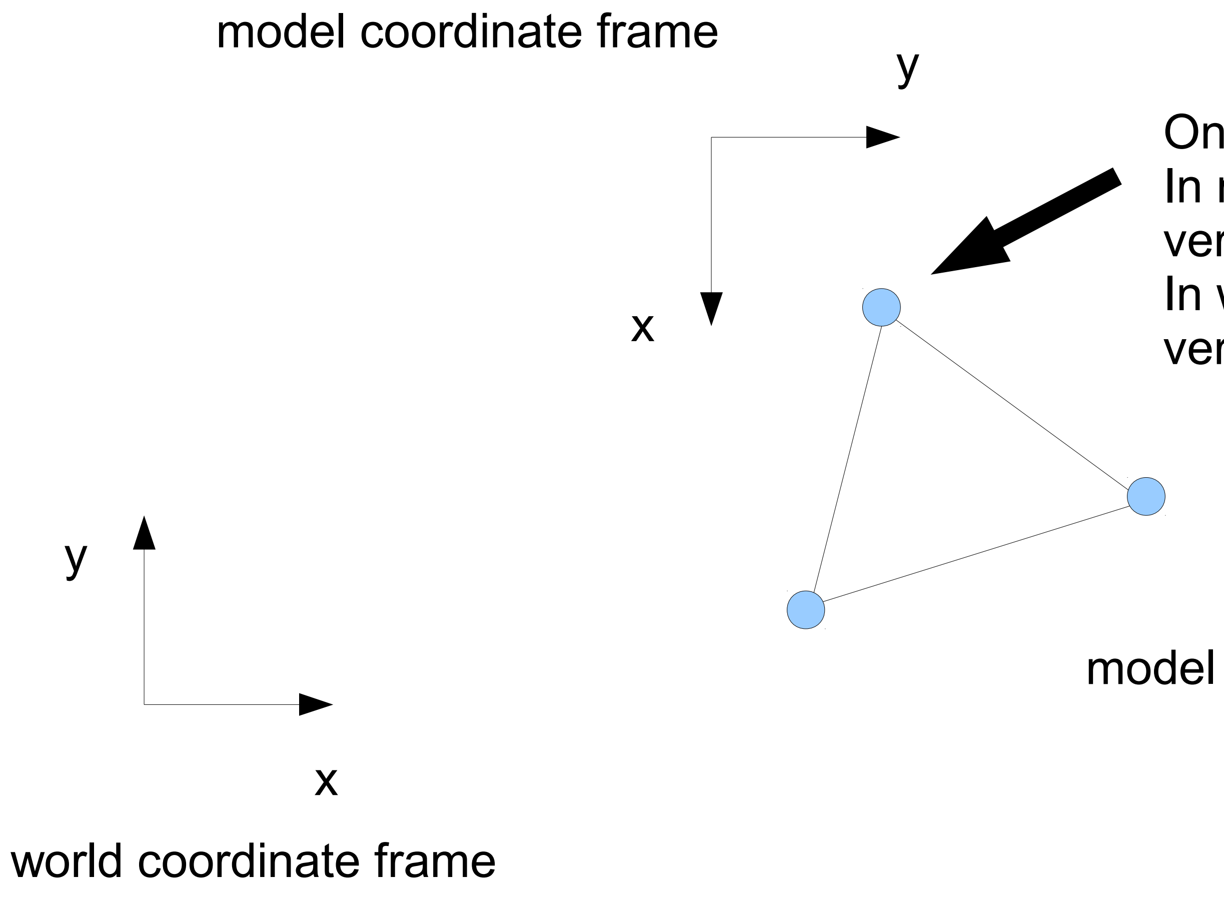
- Have a local coordinate frame
  - Origin
  - Axes (x, y, and z)
- When we talk about "the position" of a model, we mean the location of its origin
- The vertices of the model are specified with respect to this local coordinate frame



Issues

- The vertices of an model need to be rendered in the coordinates of the overall scene ("world coordinates"). Each time the scene is rendered, each vertex needs to be quickly converted into world coordinates.
- Objects may be located anywhere in the world
- Objects will have arbitrary orientations
- Specifying orientation

Trivial 2D Example



One vertex of the model.
In model coordinates, this vertex is at (1,1).
In world coordinates, this vertex is at (4, 2.5).

Specifying Orientation

- Euler angles (generally most intuitive, but look up "gimbal lock")
  - Three angles indication rotations to be performed around each coordinate axis in sequence (order matters!) Examples: yaw, pitch, roll for an aircraft or azimuth and elevation (roll always zero) for a gun.
- Rotation matrices (fastest)
  - 3x3 matrices. To apply simply multiply.
- Quaternion (best tradeoff?)
  - Four numbers capable of indicating the amount of rotation around an arbitrary axis. Applied by quaternion multiplication (special operator).

Homogeneous Transform Matrices

- 4x4 matrices
- Contain a 3x3 rotation matrix (r)
- Contain a translation vector (t)
- With proper choice of components, can convert coordinates of a vertex between any two frames
- To convert a vertex
  - Pad it to a 4 element vector by appending a one
  - Multiply by the homogeneous transform matrix

$$T = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Scene Graphs

- Occasionally more coordinate frames are in play than just those of the individual models and the world
- Example: We want to specify the location and orientation of a sailor model with respect to his ship, not directly with respect to the world.
- In the example above, the sailor's coordinate frame is the "child" and the ship's is the "parent"
- You can first multiply by the sailor transform to convert to ship coordinates and then by the ship transform to convert to world coordinates.

Data Flow in three.js

- If autoupdate is true
  - User specifies position and quaternion of an object
  - Alternatively, the user can specify Euler angles, which are then converted to a quaternion
  - "matrix" (HTM converting from child to parent coordinates) and then "matrixWorld" (HTM converting from child directly to world coordinates) are then constructed from the position and quaternion by three at render time

Data Flow in three.js

- If autoupdate is false
  - User directly manipulates "matrix"
  - At render time, three.js constructs "matrixWorld" using "matrix"