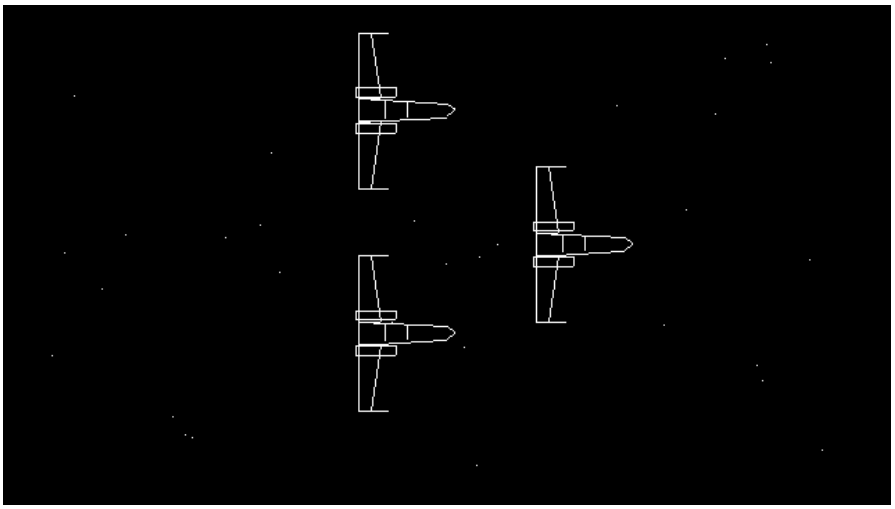
Project 6 - Hierarchical Modeling

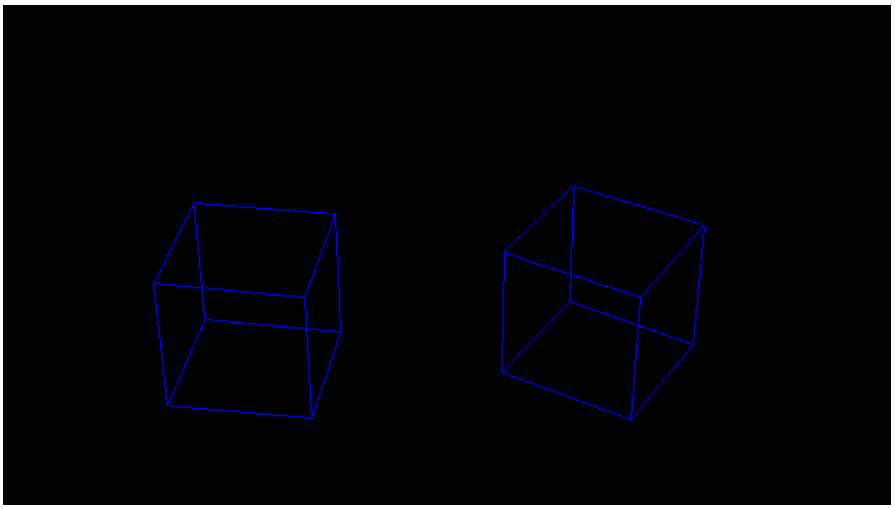
Created by Gregory Attra, last modified just a moment ago

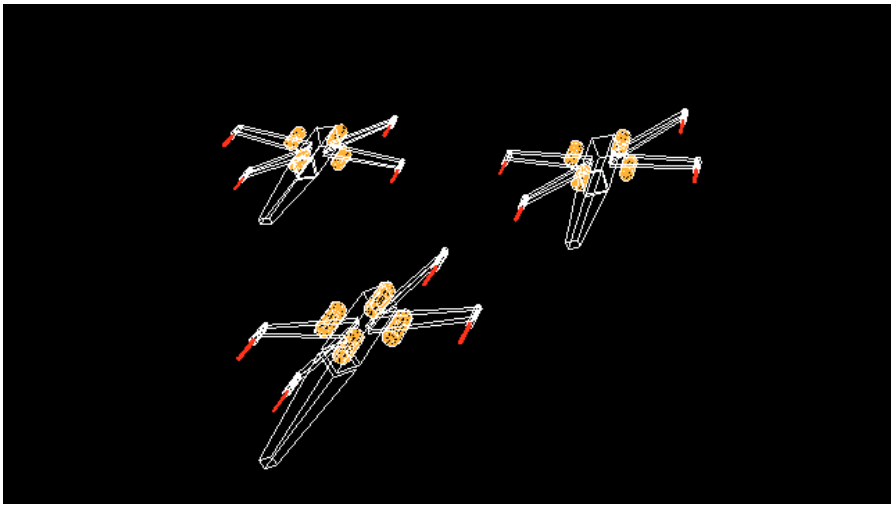
Introduction

Hierarchical modeling is a way to programmatically build complex scenes using a graph of objects and transformation matrices defining their relative poses. If you consider an object as being composed of smaller sub-objects (i.e. a chair is composed of multiple cubes which have been scaled in various ways), you can then programmatically define the objects as a graph, where each node in the graph specifies either the graphics primitive to draw, a transformation matrix to move the object to a certain location in the scene, or the root of another object tree. At the bottom of every object tree are the graphics primitives which make up the whole object. Once you have built a scene graph, the next step is to draw the scene by traversing this graph. By keeping track of our global and local position when traversing a graph, we can define object models whose points are defined in local space, and still be able draw these objects in global space by simply transforming the local space into the global space using the local and global transformation matrix respectively.

Required Images

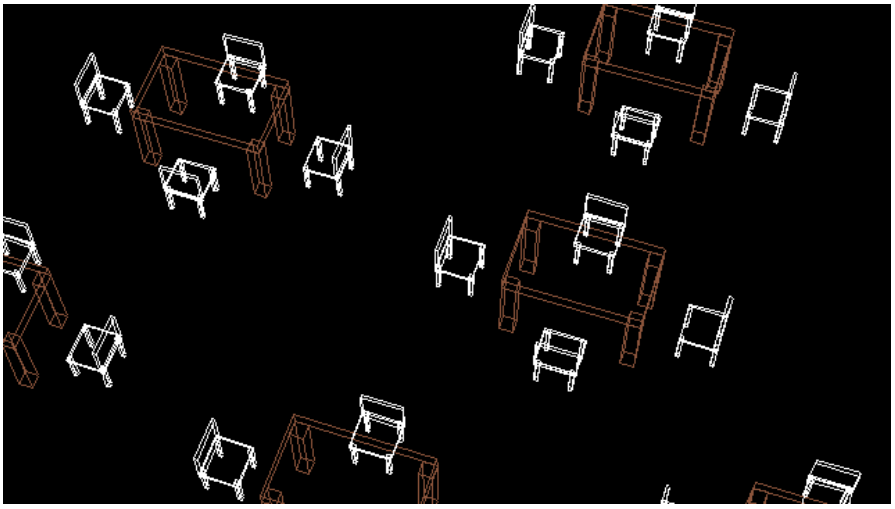






The first image is a demonstration of the 2D implementation of a scene graph (`test6a`). The second is a demonstration of the `module_cube` function which generates a unit cube at local point (0, 0, 0). The third image is a demonstration of the 3D implementation of a scene graph (`test6b`).

Custom Images



For my custom image I created a restaurant scene. The scene graph is as follows:

```
restaurant:
    - dining_set:
        - table:
            - scale & translate
            - cube
            - scale & translate
            - cube
            ...
        - chair:
            - scale & translate
            - cube
            - scale & translate
            - cube
            ...
```

```
X 6
```

Each `DiningSet` is composed of a `Table` and four `Chair` objects. Using translations and rotations, a `DiningSet` module will draw a table at local point (0, 0, 0) and then place four chairs around the table, rotated to face inward.

At the scene level, simple translations across world space in between each `DiningSet` draw results in the multiple dining sets placed at various locations around the scene.

Reflection

Having spent a lot of my free time tinkering with game engines like Unity or 3D rendering engines like Blender, I caught on fairly quickly to the concepts behind hierarchical modeling and scene graph traversal. It has been very rewarding to learn what goes on under-the-hood in that sense. I also think the API we are implementing has been very well defined upfront. Whenever I've found myself thinking: I need a function right now which does X, 9/10 times that function already exists and was defined by the API specs. This has been implementation much smoother and allowed me to focus on the hard parts and less on the tedious parts.

Acknowledgements

- Stack Overflow
- Prof. Maxwell's Lecture Notes
- Graphics Engine Specs for Project 6

No labels