Homework 5 (Answers): Sceneview

**Question 1** Define the classes you will use to represent a single subgraph in the scene file. These classes should include: how to store matrices, object attributes, the objects (cones, spheres, etc) themselves, and tree structure information. Remember that a subgraph can have a list of trans nodes, a trans node can have a trans node in it, and that there can be any number of matrices listed in a trans node. The next question will deal with named subgraphs and linking them into an existing subgraph. Write these out as you would a class header file, and include comments. Also include the methods you need on the classes (you do not need to write out the source code for the method, just the declaration).

This answer may vary between students, but should look something like:

```cpp
// Base class for tree
class Node {
    Array<Node *> aChildren; // children pointers, Array is a List type

    void Draw( const Matrix & ) const; // Draw me, then my children
    void AddNode( Node * );
}

class Trans : public Node {
    Matrix matrix; // Accumulated matrix
};

class Shape : public Node {
    Color diffuse, specular, ambient, reflect, transparent;
    int shine, ior;
    double textureU, textureV;
    char *texture;

    MyShape *objPtr;
};

class Subgraph : public Node {
}
```

**Question 2** For the following scenefile code, draw the class instances that will be created and how they will be linked together (this should be some form of a tree or graph structure). Your arrows must be labeled with the member variable names. Include where the tessellated objects are stored, and the appropriate pointers to them.
subgraph root [ 
  trans [ 
    rotate 0 1 0 30 
    scale 1 1 2 
    object cone [ 
      diffuse 1 0 0 
    ] 
  ] 
  trans [ 
    scale 2 2 2 
    object sphere [ 
      specular 0 1 0 
    ] 
  ] 
]

Note that in the diagram above, the trans nodes appear “linked” together only because we use a linked list that points to trans nodes to store them. You should not think that the trans nodes are passing their transformations to each other through this linked list. For further explanation, consider the following:

There should be some sort of class for the list structure, some sort of class for holding the object data (color, etc.), and some sort of class for holding the transformation information.

**Question 3** Describe the parsing algorithm you will use for the scenegraph in question 1.

Either a recursive or an iterative algorithm works here. If they do an iterative algorithm then the current matrix should be stored as a stack object.
Question 4 Add a named subgraph to your scene file in question 2. (Just show the new pieces and clearly indicate where they go in the original scene graph.)

```
subgraph Kanga
  trans  trans
  cylinder  cube

subgraph Roo
  trans  trans
  sphere  cube
```

Question 5 Describe the data structures and algorithm you will use to parse and store a named subgraph.

We suggest that you store subgraphs in some sort of lookup table or dictionary. You are permitted to use an STL map or whatever else you like. A valid approach to refer to subgraphs is to store their names and a pointer to the root nodes of each subgraph so that the subgraph can be traversed.

The same parsing routine can be used to create the root subgraph as is used for a regular tree.