Assignment1
Solar System in OpenGL

Francesco Andreussi

francesco.andreussi@uni–weimar.de

14 November 2019

Deadline

Wednesday, 27 November 2019 at 23:55.

Task

- Implement a Scene Graph. (35%)

- Extend the ApplicationSolar class in order to display a solar system with at least 8 planets revolving around a sun and one moon as in 2. (35%)

- Comment the code extensively. (10%)

- **Additional Task**: implement camera controls with mouse look using the mouse and the keyboard. (20%)

Tips & Suggestions

- You need a `SceneGraph` class, a `Node` superclass from which derive a `GeometryNode` and a `CameraNode` classes (see 1).

- The orbits can be assumed as being circular.

- A change of the mouse position results in the execution of the mouseCall- back function, containing the x and y movement as arguments.

- The relative distance between the planets is up to you.

- Do the first point for next week, I will be available to take a look to you work and help you.

```
        SceneGraph
-----------------------------
-String: name
-Node: root
-----------------------------
+getName(): String
-setName(String): void
+getRoot(): Node
-setRoot(Node): void
(+printGraph(): String)
-----------------------------
[singleton]
```

```
           Node
-----------------------------
-Node: parent
-List<Node>: children
-String: name
-String: path
-int: depth
-mat4: localTransform
-mat4: worldTransform
-----------------------------
+getParent(): Node
+setParent(Node): void
+getChildren(String): Node
+getChildrenList(): List<Node>
+getName(): String
+getPath(): String
+getDepth(): int
+getLocalTransform(): mat4
+setLocalTransform(mat4): void
+getWorldTransform(): mat4
+setWorldTransform(mat4): void
+addChildren(Node): void
+removeChildren(String): Node
```

```
        GeometryNode
-----------------------------
-model: geometry
-----------------------------
+getGeometry(): model
+setGeometry(model): void
```

```
         CameraNode
-----------------------------
-bool: isPerspective
-bool: isEnabled
-mat4: projectionMatrix
-----------------------------
+getPerspective(): bool
+getEnabled(): bool
+setEnabled(bool): void
+getProjectionMatrix(): mat4
+setProjectionMatrix(mat4):
```
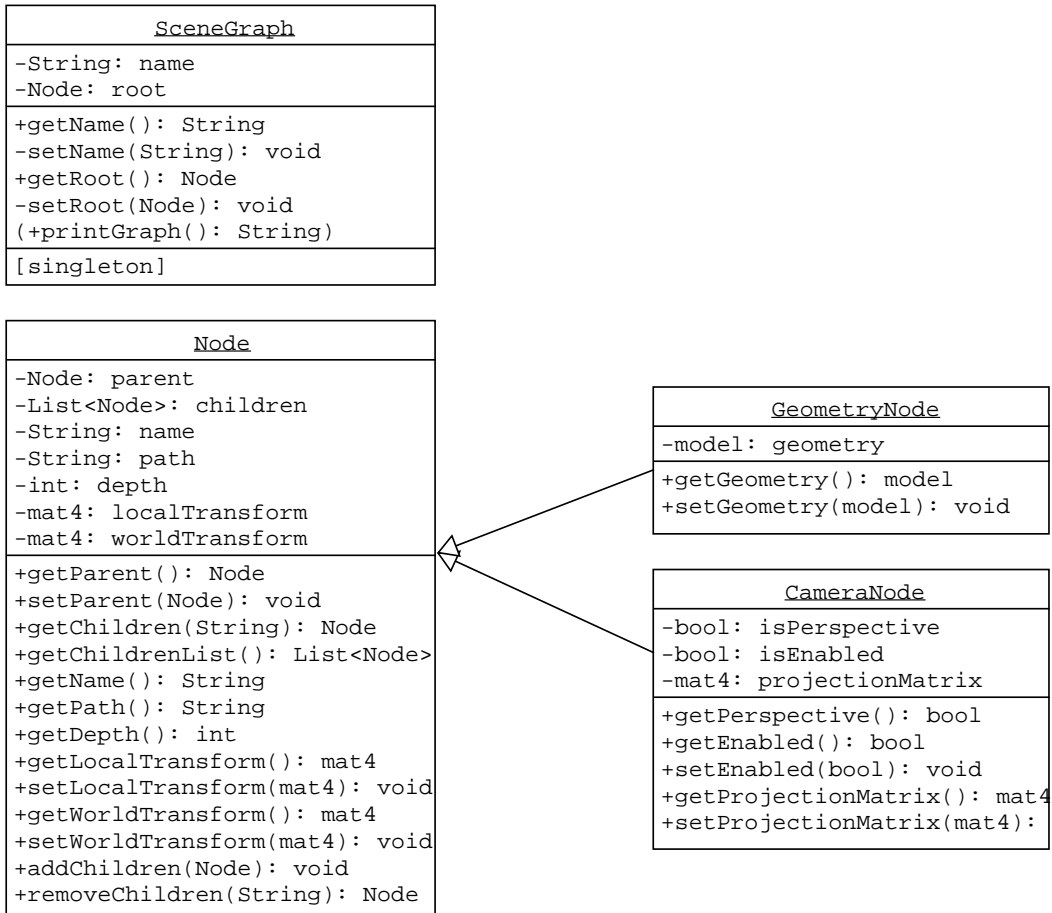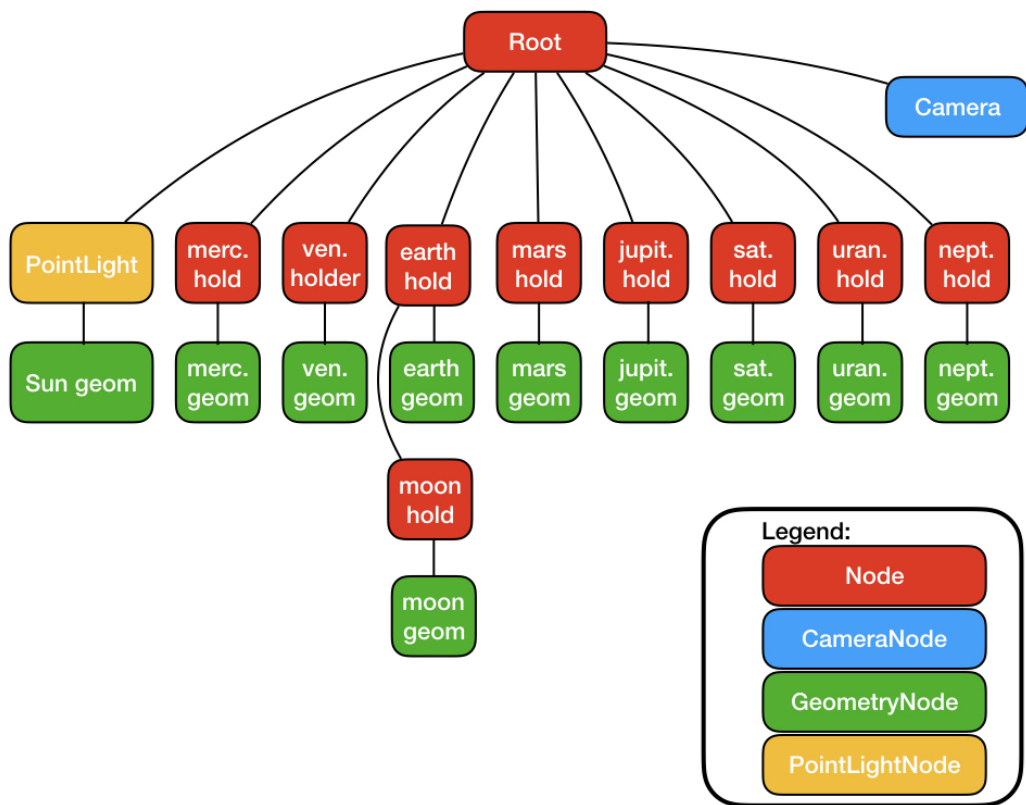
Figure 1: UML representation of the classes you have to implement

Figure 2: This is how your scenegraph should look like