**Title:** Instructions for Using the SceneManager Material Loading Artifact

**1. Overview**  
The artifact is the enhanced SceneManager class that dynamically loads object materials from a SQLite database instead of hardcoding them. This allows for easy modification or addition of materials without recompiling the project.

**2. Loading Materials**

* Materials are stored in materials.db (SQLite format).
* Each material has the following attributes: tag, diffuse\_r, diffuse\_g, diffuse\_b, specular\_r, specular\_g, specular\_b, shininess.
* The SceneManager automatically reads the database when instantiated and stores the materials in m\_objectMaterials.

**3. Adding or Modifying Materials**

1. Open materials.db using a SQLite editor (e.g., DB Browser for SQLite).
2. To add a new material:
   * Insert a new row in object\_materials with appropriate values.
3. To modify an existing material:
   * Edit the values of the row corresponding to the material tag.
4. Save the database.
5. Restart the application to load the updated materials.

**4. Applying Materials in Code**

* To apply a material to an object:

sceneManager.SetShaderMaterial("metal"); // Example using a tag from the database

* The object will render using the diffuse, specular, and shininess values from the database.

**5. Debugging**

* If a material does not appear correctly, check:
  + The material tag matches exactly between code and database.
  + The database path is correct (materials.db).
  + The database contains valid float values for colors and shininess.

**6. Notes**

* The system supports adding any number of materials dynamically.
* No code recompilation is required to add new materials — only a database update.
* Console output will display the materials as they are loaded for verification.