

Backgrounds

- The AccessibleComponent interface which is supported by any object that is rendered on the screen provides a setBackground() method which permits the background colour to be set in Java 2D
- In Java 3D there is a Background class which defines a leaf node in the scene graph and has a region of influence
 - The region of influence can be specified as a Bounds object or as a BoundingLeaf

Setting up a Background

`Background()`

Constructs a default Background node

`Background(Color3f color)`

Constructs a Background node with the specified colour

`Background(ImageComponent2D image)`

Constructs a Background node with the specified image

`Background(BranchGroup branch)`

Constructs a Background node with the specified geometry

`bg.setApplicationBounds (Bounds bounds);`

`bg.setApplicationBoundingLeaf`

`(BoundingLeaf boundLeaf);`

Associate a region of influence with a background bg

Textured Backgrounds

- Java 3D allows backgrounds to have geometry associated with them
 - Geometry objects can be textured of course
- We can envisage the virtual universe of a scene graph as being circumscribed by a sphere
 - Or a hemisphere if there is a floor to the scene
- A spherical background can have a texture applied to its inner surface
 - A background sphere is drawn at infinity irrespective of the radius used to create it

Example - Spherical Background

```
BranchGroup bgRoot = new BranchGroup();
Background bg = new Background();
bg.setApplicationBounds(bounds);
BranchGroup bgGeomBranch = new BranchGroup();
Sphere bgSphere = new Sphere(1.0f,
                             Sphere.GENERATE_NORMALS_INWARD |
                             Sphere.GENERATE_TEXTURE_COORDS, 20);
bgGeomBranch.addChild(bgSphere);
bg.setGeometry(bgGeomBranch);
bgRoot.addChild(bg);
```

Example - Attaching a Texture

```
NewTextureLoader tex =  
    new NewTextureLoader("image.jpg");  
TextureAttributes texAttr =  
    new TextureAttributes();  
Transform3D textureTrans = new Transform3D();  
textureTrans.setScale(20);  
texAttr.setTextureTransform(textureTrans);  
Appearance bgApp = bgSphere.getAppearance();  
bgApp.setTextureAttributes(texAttr);  
if (tex != null)  
    bgApp.setTexture(tex.getTexture());
```