JAVASCRIPT examples
3 – script functions

Java game application:

```
import javax.script.ScriptEngine;
import javax.script.ScriptEngineFactory;
import javax.script.ScriptEngineManager;
import javax.script.ScriptException;
import javax.script.Invocable;

import java.io.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import net.java.games.input.Event;

import ray.rage.*;
import ray.rage.game.*;
import ray.rage.rendersystem.*;
import ray.rage.rendersystem.Renderable.*;
import ray.rage.scene.*;
import ray.rage.scene.Camera.Frustum.*;
import ray.rage.scene.controllers.*;
import ray.rml.*;
import ray.rage.rendersystem.gl4.GL4RenderSystem;
import ray.input.*;
import ray.input.action.*;

// This class demonstrates how to use JavaScript functions to apply
// changes to GameWorld nodes. It attaches a JavaScript function
// to a keyboard action;  hitting the SPACE key invokes the
// corresponding JavaScript function, which alters the light color.
```

```
public class MyGame extends BaseGame
{  . . .
   protected ScriptEngine jsEngine;
   private InputManager im;
   protected ColorAction colorAction;
   protected File scriptFile3;
```

```
protected void setupScene()
 {. . .
   // same as before, plus the following:

   // prepare script engine
   ScriptEngineManager factory = new ScriptEngineManager();
   java.util.List<ScriptEngineFactory> list = factory.getEngineFactories();
   jsEngine = factory.getEngineByName("js");

   // set up the script that associates the light color with the space bar
   scriptFile3 = new File("UpdateLightColor.js");
   this.runScript(scriptFile3);
   im = new GenericInputManager();
   String kbName = im.getKeyboardName();
   colorAction = new ColorAction(sm);
   im.associateAction(kbName,
            net.java.games.input.Component.Identifier.Key.SPACE,
            colorAction,
            InputManager.INPUT_ACTION_TYPE.ON_PRESS_ONLY);
   . . .
}
```

```
private void runScript()
 { try
   { FileReader fileReader = new FileReader(scriptFile);
     engine.eval(fileReader);
     fileReader.close();
   }
   catch (FileNotFoundException e1)
   { System.out.println(scriptFile + " not found " + e1); }
   catch (IOException e2)
   { System.out.println("IO problem with " + scriptFile + e2); }
   catch (ScriptException e3)
   { System.out.println("Script Exception in " + scriptFile + e3); }
   catch (NullPointerException e4)
   { System.out.println ("Null ptr exception reading " + scriptFile + e4); }
 }
```

```
// an Action for invoking a script function

private class ColorAction extends AbstractInputAction
{  private SceneManager sm;

   private ColorAction(SceneManager s)  {  sm = s; }  // constructor

   public void performAction(float time, Event e)
   {  //cast the engine so it supports invoking functions
      Invocable invocableEngine = (Invocable) jsEngine ;

      //get the light to be updated
      Light lgt = sm.getLight("testLamp1");

      // invoke the script function
      try
      {  invocableEngine.invokeFunction("updateAmbientColor", lgt); }
      catch (ScriptException e1)
      {  System.out.println("ScriptException in " + scriptFile3 + e1); }
      catch (NoSuchMethodException e2)
      {  System.out.println("No such method in " + scriptFile3 + e2); }
      catch (NullPointerException e3)
      { System.out.println ("Null ptr exception reading " + scriptFile3 + e3); }
}  }
```

"UpdateLightColor.js" (javascript file)

```
var JavaPackages = new JavaImporter(
   Packages.ray.rage.scene.Light,
   Packages.java.awt.Color
   );

with (JavaPackages)
{
   function updateAmbientColor(thisLight)
   {  thisLight.setAmbient(java.awt.Color.blue);
   }
}
```