

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

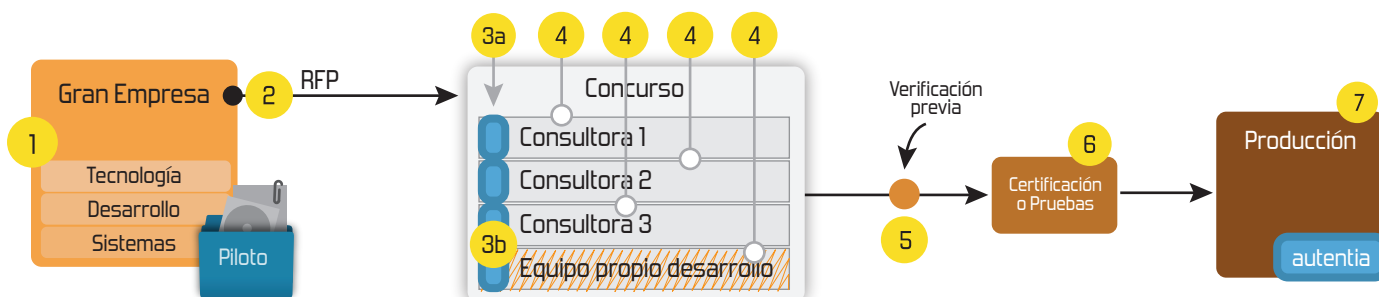
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)

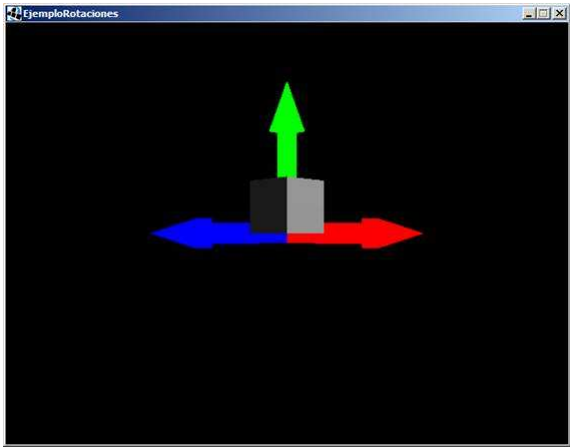
2009-05-21
Primeros pasos con Audacity: Un editor de
sonido libre y multiplataforma.



Vamos con las transformaciones de la escena actual. Me gustaría ver en diagonal lo que voy haciendo dentro de mis ejes: Entonces, tengo dos opciones: Rotar/mover la cámara o rotar lo que pinto. Ya os digo que a priori parece más sensato mover la cámara pero esto es tan solo un ejemplo.

Voy a empezar a mover lo que pinto. Usaremos este código fuente.

```
view plain print ?
01. package adictosaltrabajo;
02.
03. import java.util.logging.Logger;
04.
05. import com.jme.app.SimpleGame;
06.
07. import com.jme.math.Vector3f;
08.
09. import java.util.logging.Level;
10.
11. import com.jme.scene.shape.*;
12.
13. import com.jme.math.Quaternion;
14.
15. import com.jme.math.FastMath;
16.
17. /**
18.  *
19.  * Roberto Canales Basado en ejemplos de
20.  * @author Jack Lindamood
21.  */
22.
23. public class EjemploRotaciones extends
24. SimpleGame {
25.
26. private static final Logger logger =
27. Logger.getLogger(EjemploRotaciones.class.getName());
28.
29. private Quaternion q1 = new Quaternion();
30.
31. public static void main(String[] args){
32.
33. EjemploRotaciones app = new EjemploRotaciones();
34.
35. logger.log(Level.INFO, "La versión es: " + app.getVersion());
36.
37. app.setConfigShowMode(ConfigShowMode.AlwaysShow);
38.
39. app.start();
40.
41. }
42.
43.
44.
45.
46.
47.
48. protected void simpleInitGame() {
49.
50.
51. q1.fromAngles(0, -FastMath.QUARTER_PI, 0);
52.
53. rootNode.setLocalRotation(q1);
54.
55. // Pintar los ejes de coordenadas
56.
57. AxisRods ejes = new AxisRods("ejes", true, 5.0f);
58.
59. rootNode.attachChild(ejes);
60.
61.
62. // Pintar una caja
63.
64. Box b = new Box("MiCaja", new Vector3f(0, 0, 0), new Vector3f(2, 2, 2));
65.
66. b.setLocalTranslation(0, 0, 3);
67.
68. rootNode.attachChild(b);
69.
70. }
71.
72. }
73. }
```



Como veis, hemos rotado 45 grados en el eje y (en radianes) apoyándonos en el método setLocalRotation y la clase Quaternion que nos simplifica la gestión respecto al uso de matrices.

Os pongo también el ejemplo de cómo podríamos hacer para rotar la cámara alrededor del eje:

2009-05-11
[Introducción a TortoiseSVN](#)

2009-05-07
[Hacer 'scp' de varios ficheros sin solicitud de clave](#)

2009-05-02
[Plugin Hibernate3 para Maven](#)

2009-04-26
[AgileDraw: una técnica rápida de modelado](#)

2009-04-24
[Spring AOP: Cacheando aplicaciones usando anotaciones y aspectos con AspectJ](#)

2009-04-20
[Modelos de conocimiento con CmapTools](#)

2009-04-16
[Informes Crosstab con iReport](#)

2009-04-16
[Registro de un fichero de datos personales con el formulario NOTA](#)

2009-04-15
[Estadísticas de www.adictosaltrabajo.com Abril 2009](#)

2009-04-15
[Iniciación a OSWorkflow con Spring](#)

2009-04-14
[Tests de Selenium con librerías de componentes JSF: Apache Tomahawk.](#)

2009-04-13
[JTAPI. El API de Telefonía para Java](#)

2009-04-13
[Registro de Web Services con Apache Juddi. Configuración y ejemplo](#)

2009-04-13
[Cómo hacer UML con Eclipse y el plugin UML2](#)

2009-04-09
[Spring WS: Servicios Web a través del correo electrónico](#)

2009-04-02
[Creación de cursos con Moodle](#)

2009-03-31
[Integrar Liferay Portal 5.2.1 con Pentaho BI 2.0.0 sobre MySQL 5.1](#)

2009-03-31
[Spring WS: Construcción de Clientes de Servicios Web con Spring](#)

2009-03-30
[Administración de sitios Moodle](#)

2009-03-29
[Empaquetamiento de aplicaciones de escritorio \(standalone\) con Maven](#)

2009-03-27
[Primeros pasos con Moodle](#)

2009-03-26
[Introducción a JSF Java](#)

2009-03-25
[A1 Website Analyzer](#)

2009-03-24
[Cómo ver el correo de Gmail sin conexión a Internet](#)

2009-03-20
[JasperReports Maven Plugin](#)

Últimas ofertas de empleo

2009-06-29
[Atención a cliente - Call Center - BARCELONA.](#)

2009-06-25
[Atención a cliente - Call Center - BARCELONA.](#)

2009-06-20
[Comercial - Ventas - CASTELLON.](#)

2009-06-19
[Otras - Ingeniería \(minas, puentes y puertos\) - VALENCIA.](#)

2009-06-17
[Comercial - Ventas - ALICANTE.](#)

Anuncios Google

```

01. package adictosaltrabajo;
02.
03.
04. import java.util.logging.Logger;
05.
06. import com.jme.app.SimpleGame;
07.
08. import com.jme.math.Vector3f;
09.
10. import java.util.logging.Level;
11.
12. import com.jme.scene.shape.*;
13.
14. import com.jme.math.Quaternion;
15.
16. import com.jme.math.FastMath;
17.
18. import com.acarter.scenemonitor.*;
19.
20. /**
21.  *
22.  *
23.  * Roberto Canales Basado en ejemplos de
24.  * @author Jack Lindamood</p>
25.  *
26.  */
27.
28. */
29.
30. public class EjemploRotaciones extends SimpleGame {
31.
32.
33.
34.
35.
36.
37.
38.
39. // variables para hacer cosas una vez cada segund
40.
41. private long tini = System.currentTimeMillis();
42.
43. private long tfin = tini;
44.
45. // variables para el movimiento de la cámara
46.
47. private float angulo = 0;
48.
49. private Vector3f poscamara = new Vector3f(0, 0, 45);
50.
51. // contador para controlar la elevación
52.
53. private int contador = 0;
54.
55. public static void main(String[] args){
56.
57.     EjemploRotaciones app = new EjemploRotaciones();
58.
59.     logger.log(Level.INFO, "La versión es: " + app.getVersion());
60.
61.     app.setConfigShowMode(ConfigShowMode.AlwaysShow);
62.
63.     app.start();
64.
65. }
66.
67. protected void simpleInitGame() {
68.
69.     q1.fromAngles(0, -FastMath.QUARTER_PI, 0);
70.
71.     rootNode.setLocalRotation(q1);
72.
73. // Pintar Los ejes de coordenadas
74.
75. AxisRods ejes = new AxisRods("ejes", true, 5.0f);
76.
77.     rootNode.attachChild(ejes);
78.
79. // Pintar una caja</p>
80.
81. Box b = new Box("MiCaja", new Vector3f(0, 0, 0), new Vector3f(2, 2, 2));
82.
83.     b.setLocalTranslation(3, 0, 3);
84.
85.     rootNode.attachChild(b);
86.
87. //SceneMonitor.getMonitor().showViewer(true);</p>
88.
89. //SceneMonitor.getMonitor().registerNode(rootNode);</p>
90.
91. }
92.
93. // bucle de juego</p>
94.
95. protected void simpleUpdate() {</p>
96.
97.     if (tpf < 1){
98.
99.         angulo = angulo + (tpf * 1);</p>
100.
101.
102.         if (angulo > 360) {</p>
103.
104.             angulo = 0;</p>
105.
106.         }
107.
108.     }
109.
110. // establecemos la posición alrededor del eje y
111.
112.
113. poscamara.set((float)(45*Math.sin(angulo)),contador,(float)(-4 *Math.cos(angulo)));
114.
115.     cam.setLocation(poscamara); // cambiamos la posición de la cámara
116.
117.     cam.lookAt(Vector3f.ZERO,new Vector3f(0,1,0)); //que mire al punto 0 en vertical
118.
119. // vamos a ver la posición y orientación de la cámara
120.
121.     tfin = System.currentTimeMillis();
122.
123.     if(tfin - this.tini > 1000) //cada segundo
124.
125.     {
126.
127.         tini = tfin;// vamos a volcar cada segundo la posición de la cámara
128.
129.         Vector3f camarapos = this.cam.getLocation();
130.
131.         logger.log(Level.INFO, "La cámara est´ en : " + camarapos.toString());
132.
133.         Vector3f camaraor = this.cam.getDirection();
134.
135.         logger.log(Level.INFO, "La cámara apunta a : " + camaraor.toString());
136.
137.         contador++;
138.
139.     }
140.
141. }
142.
143. }

```

Hemos dicho que es mejor rotar la cámara porque sino, pierdes las referencias de las coordenadas absolutas.

Si volvemos a nuestras transformaciones de los objetos de la escena, habrá veces que querremos concatenar transformaciones o hacer cosas especiales como rotar todo desde un punto. Para ellos, simplemente tenemos que crear nodos intermedios o anidados para aplicar esos efectos. Podeis ver como se hace en este enlace y una mejor explicación:

http://www.jmonkeyengine.com/wiki/doku.php?id=rotate_about_a_point

Os propongo este código para entenderlo mejor:

```
package adictosaltrabajo;

/**
 * @author Base http://www.jmonkeyengine.com/wiki/doku.php?id=rotate_about_a_point
 */

import java.util.logging.Logger;

import com.jme.app.SimpleGame;
import com.jme.math.Vector3f;

import java.util.logging.Level;
import com.jme.scene.shape.*;
import com.jme.math.Quaternion;
import com.jme.scene.*;
import com.jme.bounding.*;

public class JMEjemplo1 extends SimpleGame{

    private static final Logger logger = Logger.getLogger(JMEjemplo1.class.getName());

    private Quaternion rotQuat1 = new Quaternion();
    private Quaternion rotQuat2 = new Quaternion();

    private float angle = 0;

    private Vector3f axis = new Vector3f(0, 1, 0);

    private Sphere s;

    private Sphere moon1, moon2;

    private Node pivotNode1, pivotNode2, pivotNode3;

    long tini = System.currentTimeMillis();

    long tfin = tini;

    /**
     * Entry point for the test,
     * @param args
     */
    public static void main(String[] args){

        JMEjemplo1 app = new JMEjemplo1();

        app.setConfigShowMode(ConfigShowMode.AlwaysShow);

        app.start();

    }

    protected void simpleUpdate() {

        if (tpf < 1) {

            angle = angle + (tpf * 1);

            if (angle > 360) {

                angle = 0;

            }

        }

        rotQuat1.fromAngleAxis(angle, axis);
        pivotNode1.setLocalRotation(rotQuat1);

        //Moon 2 orbits twice as fast
        rotQuat2.fromAngleAxis(angle * 2, axis);
        pivotNode2.setLocalRotation(rotQuat2);

        tfin = System.currentTimeMillis();

        if(tfin - this.tini > 1000)

            tini = tfin;

        Vector3f camarapos = this.cam.getLocation();
        logger.log(Level.INFO, "La camara está en :" + camarapos.toString());
        Vector3f camaraor = this.cam.getDirection();
        logger.log(Level.INFO, "La camara apunta a : " + camaraor.toString());

    }

}

protected void simpleInitGame() {

    display.setTitle("JME - Rotación alrededor de un punto");

    // Pintar Los ejes de coordenadas
    AxisRods ejes = new AxisRods("ejes", true, 100.0f);
    rootNode.attachChild(ejes);

    //Planet

    s = new Sphere("Planeta", 25, 25, 25);
    s.setModelBound(new BoundingSphere());
    s.updateModelBound();

    //Moons

    moon1 = new Sphere("Luna1", 25, 25, 10);
    moon1.setModelBound(new BoundingSphere());
    moon1.updateModelBound();

    moon1.setLocalTranslation(40, 0, 0);

    pivotNode1 = new Node("PivotNode 1");
    pivotNode1.attachChild(moon1);

    moon2 = new Sphere("Luna2", 25, 25, 7.5f);
    moon2.setModelBound(new BoundingSphere());
    moon2.updateModelBound();

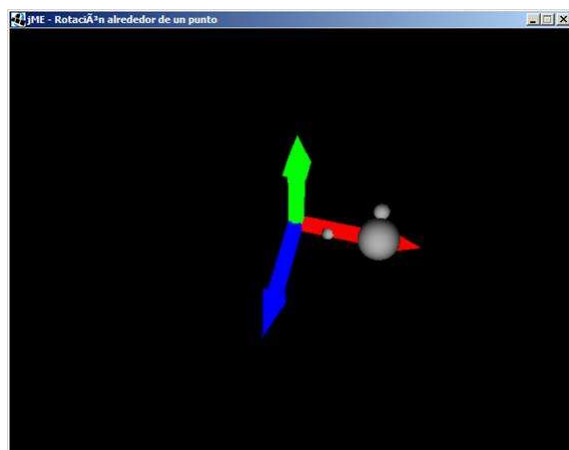
    moon2.setLocalTranslation(60, 0, 0);

    pivotNode2 = new Node("PivotNode 2");
    pivotNode2.attachChild(moon2);

    pivotNode3 = new Node("PivotNode 3");
    pivotNode3.setLocalTranslation(100, 0, 0);
    pivotNode3.attachChild(s);

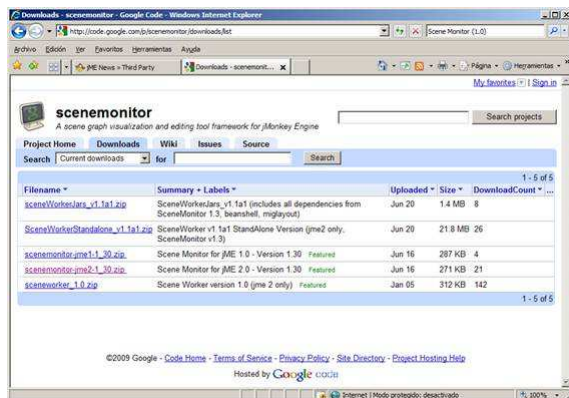
}
```

hemos creado un nodo hijo (el 3) y hemos hecho que los demás sean hijos a su vez. Como el primero le hemos dado (traslate), todos los demás parten de esas coordenadas.

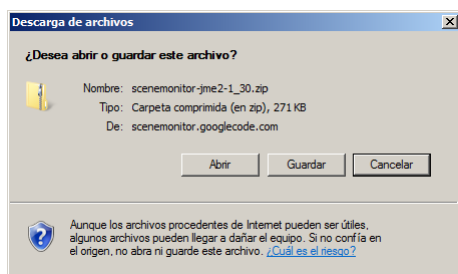


Bueno, está entendido como se realizan las transformaciones. Es un buen momento para añadir el código de la consola de la escena y ver qué está pasando. SceneMonitor es una herramienta que nos permite ver la estructura de nodos de nuestro programa actual.

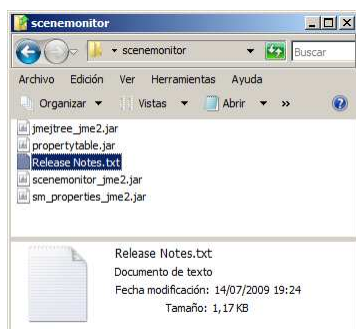
Vamos al link del producto/librería <http://code.google.com/p/scenemonitor/wiki/SceneMonitor>



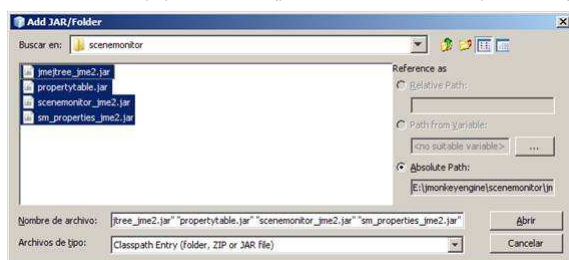
No es muy grande.



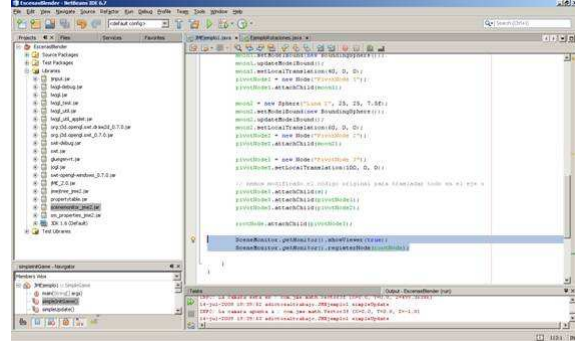
Descomprimos los ficheros en un directorio.



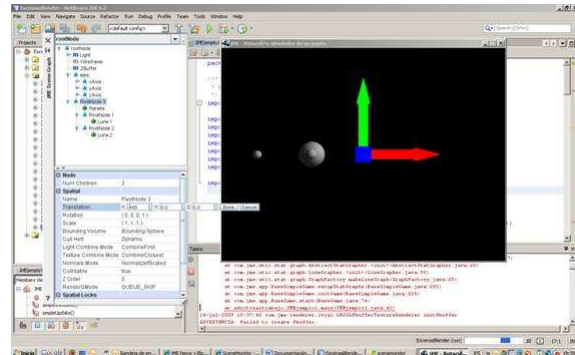
Las añadimos a nuestro proyecto en NetBeans (pulsando el botón derecho en la carpeta de librerías)



En nuestro programa solo hay que añadir un par de líneas (eso sí, tampoco se olvide el `import com.acarter.scenemonitor.*;`)



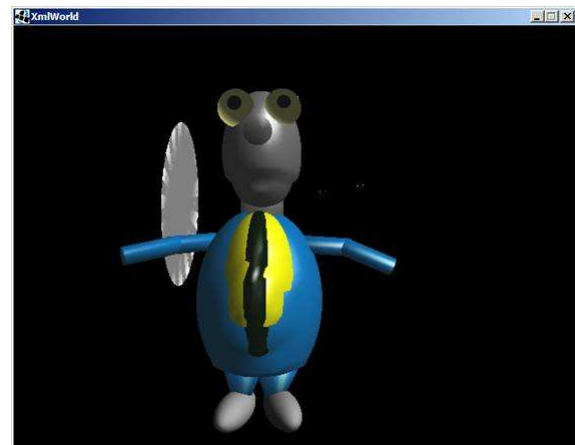
Y lo curioso es no solo que vemos la estructura de la escena sino que "la podemos modificar" y cambiar la posición de los elementos.



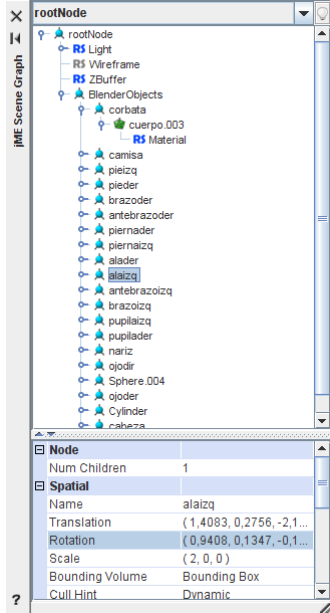
Bueno, vamos a probar lo mismo con la mosca importada de blender a ver si se corresponden los nodos con los mismos nombres que les di dentro de la herramienta (recordad que todo esto está descrito en los tutoriales anteriores sobre [blender](#)).



Pues efectivamente vemos que nos conserva los nombres de los nodos y dentro est'n nuestros elementos. Cambio atributos de las alas y vemos como le afecta (importante porque nos da pistas de cómo pintar los objetos para no encontramos luego problemas ya en nuestro programa):



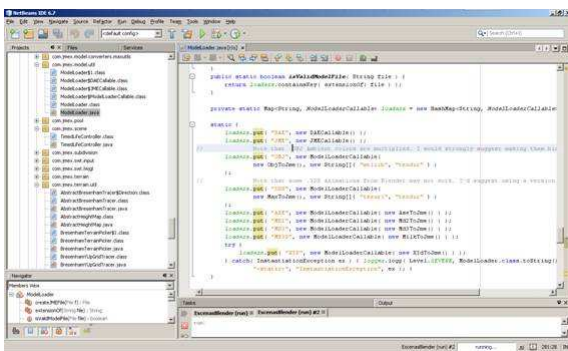
Este es el aspecto de la consola. Jugad con ella cambiando atributos, curioso.



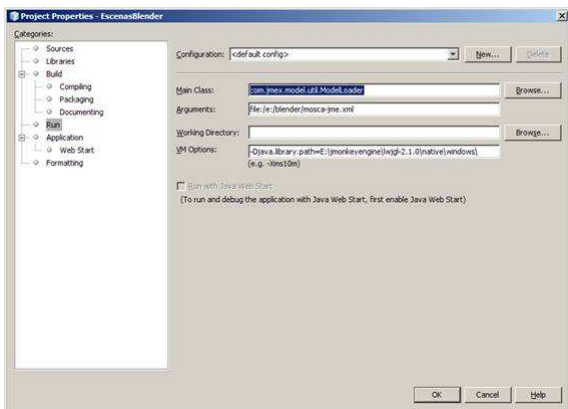
Si queréis comprobar que es lo que pasa al importar cualquier otro formato de fichero os invito a que reviséis este ejemplo que viene con JMonkeyEngine. Es un programa gráfico que te permite navegar por los directorios y cargar cualquier fichero de los formatos soportados: `ModelLoader.java`

<http://code.google.com/p/jmonkeyengine/source/browse/trunk/src/com/jmex/model/util/ModelLoader.java>

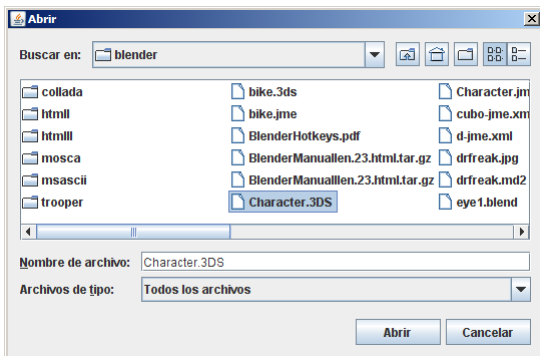
Directamente en el código vemos los formatos.



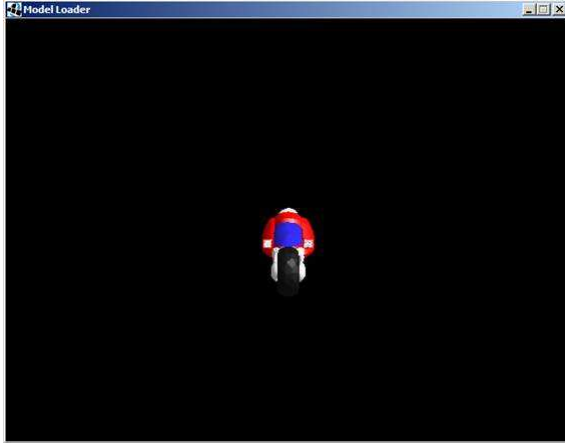
Para arrancarlo solo tenéis que modificar las características del proyecto en NetBeans y elegir la clase principal de arranque.



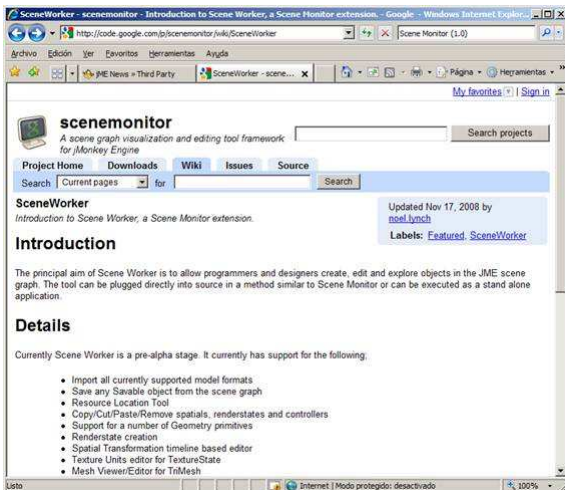
Elegimos un fichero 3DS.



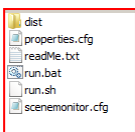
Y vemos cómo queda. Si el fichero contiene una animación esta se activará.



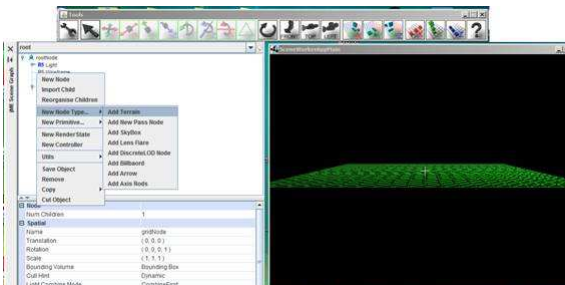
No se si os habeis fijado en que hay otro programa llamado SceneWorker que viene a hacer una mezcla de las cosas anteriores.



Vamos a la sección de descargas y nos bajamos el fichero de SceneWorker StandAlone

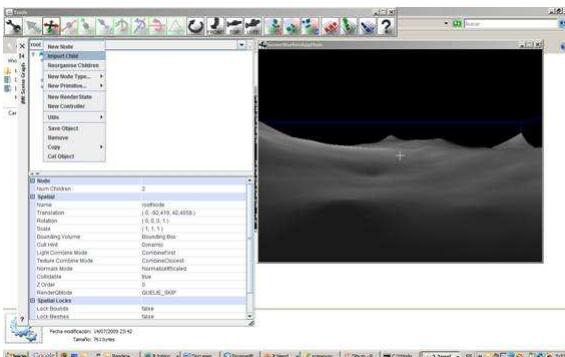


Los descomprimos en un directorio y ejecutamos el bat.

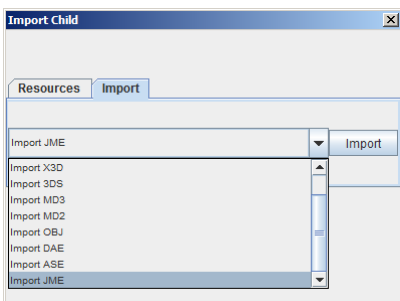


Podemos insertar en la escena cualquier elemento como terrenos, figuras básicas, el cielo, etc.

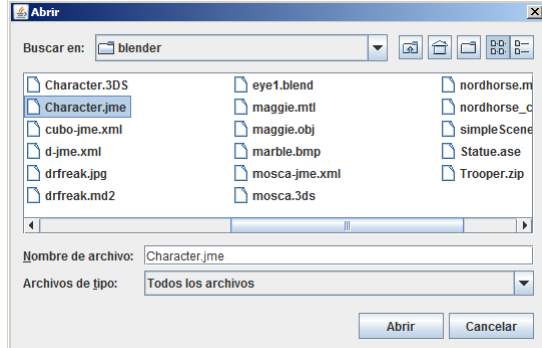
Está francamente bien porque te da incluso una idea más directa de los atributos que tendrías que tocar en tus programas para hacer lo mismo.



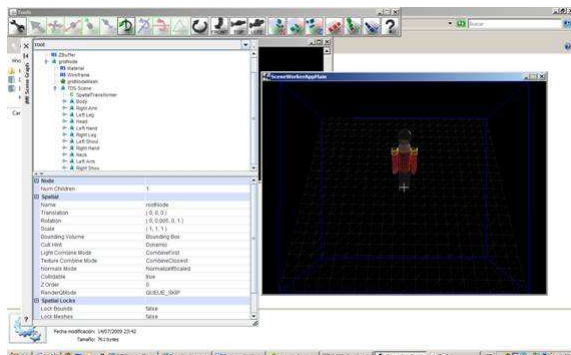
Vamos a importar en la escena un fichero de otro programa. Los ficheros JME son los de JmonkeyEngine. Si usasteis la clase ModelLoader descrita más arriba comprobareis que cada vez que cargais un fichero en otro formato, en el mismo directorio, te genera un fichero JME. Tambien este programa guarda todos los trabajos con extensión JME (se la tienes que poner tu).



Vamos a cargar el muñequito que anda usado en otros ejemplos del site.



Y no lo carga en nuestra escena (moviéndose)



Bueno, lo vamos a dejar por aquí. Espero que hayáis vistos las posibilidades de manipulado de la escena: Transformar nodos, mover la cámara, ver la escena en memoria, importar ficheros externos, generar tus ficheros JME (aunque es transparente puedes recorrer el código y ver como lo hace... que podría estar chupao para captar instantáneas) y generar escenas de un modo externo.

Cuanto más investigo más me gusta esto.

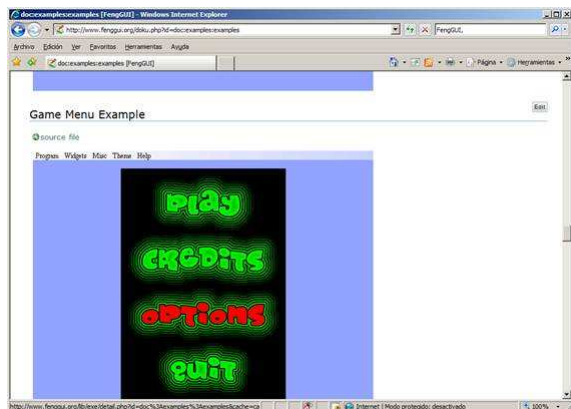
Voy a compartir con vosotros algunos enlaces de interés sobre cosas que deberíamos investigar. Lo dejamos ya para otro momento.

Un juego de tanques con fuentes incluidos: <http://sourceforge.net/projects/tankbattles/files/tankbattles/tankBattles-pre-alpha-R03.zip>



Y una librería para hacer los créditos y formularios de nuestros juegos

<http://www.fenggui.org/doku.php?id=doc:examples:examples>



También veréis que el camino de los juegos y la edición de escenarios van muy ligada por lo que deberíamos trabajar en esta línea paralelamente.

Primero visitar el Web de www.blender.es donde encontrareis cosas curiosas como plantillas con las teclas rápidas, enlaces a cientos de texturas y demás cosas.

Aceleradores de teclas de Blender <http://www.blender.es/documents/BlenderHotkeys.pdf>

Desde luego hay que dar gracias a mucha gente, hay que ver como se lo curran.

¿Qué te ha parecido el tutorial? Déjanos saber tu opinión y ivota!

Muy malo Malo Regular Bueno Muy bueno



Anímate y coméntanos lo que pienses sobre este tutorial

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Nombre: E-Mail:

Comentario:

Enviar comentario

[Texto Legal y condiciones de uso](#)

- Puedes inscribirte en nuestro servicio de notificaciones [haciendo clic aquí](#).

■

Puedes firmar en nuestro libro de visitas [haciendo clic aquí](#).

■

Puedes asociarte al grupo AdictosAlTrabajo en XING [haciendo clic aquí](#).
- Añadir a favoritos Technorati.

 Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com



Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	Valoración	Votos	Pdf
Detalles del juego de la moto en JMonkeyEngine.	Detalles del juego de la moto creado en JMonkeyEngine, donde se definen movimientos y chaserCameras	2009-07-15	42	Muy bueno	8	
JMonkeyEngine, Creación de nuestro primer juego.	Intentaremos enseñaros a crear vuestro primer juego, partiremos de un ejemplo hecho de JMonkeyEngine, que trata sobre el manejo de una moto por un escenario.	2009-07-14	143	Bueno	9	
Ajax tests con Selenium: prototype.js e ICEfaces.	En este tutorial se va habla de cómo escribir tests funcionales con Selenium IDE sobre aplicaciones que realizan recargas controladas de la interfaz de usuario con Ajax.	2009-07-13	133	Muy bueno	1	
AOP con AspectJ y Maven	Programacion orientada a aspectos con AspectJ y Maven	2009-07-08	273	Bueno	1	
Iniciarse en el manejo de JME, Creación de un Cloth.	Primeros pasos con Jmonkey engine, crearemos una pequeña aplicación animada basada en un Cloth con colisiones	2009-07-07	359	Muy bueno	6	
Primeros pasos con Blender: Pintando nuestra mascota en 3D	Inicio del manejo de blender, os mostraremos paso a paso como se dibuja en blender	2009-07-06	327	Bueno	6	
Juegos 3D en Java: Blender y JMonkeyEngine	En este tutorial se abordara una pequeña iniciación a los juegos 3D en java usando la herramienta Blender y como motor gráfico JMonkeyEngine	2009-07-02	868	Bueno	15	
Creación de cursos con Moodle	En este tercer tutorial de la serie veremos en detalle la creación de un curso	2009-04-02	2966	Muy bueno	11	
Cómo ver el correo de Gmail sin conexión a Internet	En este tutorial vamos a ver como podemos configurar el navegador Firefox 3 para poder acceder a todo nuestro histórico de correos sin necesitar de un conexión a Internet.	2009-03-24	2566	Bueno	12	
Redimensionar Imagenes en Windows Vista	Instalación y uso de ImageResizer, para poder redimensionar en Windows Vista nuestras imágenes	2009-02-26	776	Muy bueno	2	

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.