Terceros Encuentros de Programadores Java

Video-juegos en red

Presentación

- David Fernández Mota
  - Ingeniero informático [UJI]
- José Antonio Gil Altaba
  - Ingeniero informático [UJI]
  - Ingeniero técnico en diseño Industrial [UJI]

Historia

- William Higinbot.
  - diseñó un sistema de juego simulando al tenis donde una línea horizontal y otra pequeña céntrica vertical a modo de red simulaba una pista de tenis.

Historia

Smash Court Tennis - Pro Tournament (Wii) Top Spin 4 (nintendo)

Motor de juegos

- Requisitos:
  - Multiplataforma
  - Proyecto activo
  - Audio 3D
  - Visión estereoscópica
  - Escenario realista

Motor de juegos

- Ogre4j es un *port* del proyecto Ogre 3D.
- El *port* no está en un estado estable .
- Las últimas versiones del proyecto Ogre 3D no se corresponden con las últimas versiones de los *ports.*
- Previsible falta de mantenimiento del proyecto.

Motor de juegos

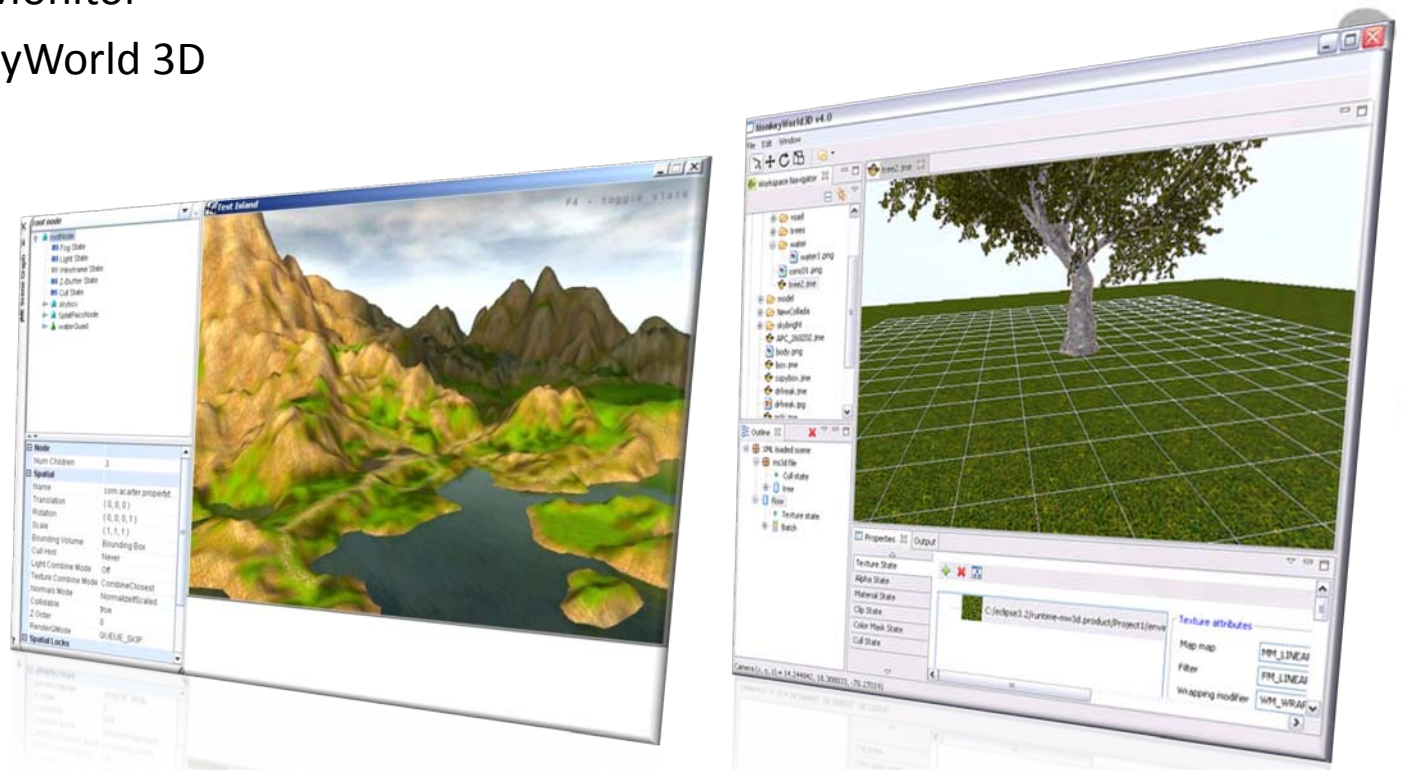
	Java3D	jME	Observaciones
Grado de mantenimiento	Medio	Alto	Sun decide cambiar Java 3D por jMonkeyEngine para su proyecto de generación de mundos virtuales Wonderland
Audio 3D	OpenAL y librería propia	OpenAL	
Visión estereoscópica	Sí	Sí	
Generación de terrenos	No	Sí	jMonkeyEngine ofrece utilidades para generar terrenos con muy buenos resultados.
Generación de sombras	No	Sí	
Multitextura	Sí	Sí	jMonkeyEngine tiene muchas más opciones para personalizar la multitextura
Colisiones	Sí	Sí	
Billboarding	Sí	Sí	
Sistemas de partículas			
Efecto niebla	Sí	Sí	
Simulación de agua	No	Sí	
Motor de físicas	Sí, pero no integrado	Sí, pero no integrado	Ambos dependen del motor de físicas OdeJava https://odejava.dev.java.net/
Escrito completamente en va o porta a Java	Escrito completamente en Java	Escrito completamente en Java	

Motor de juegos

- jMonkeyEngine ofrece…
  - Mayores alternativas para crear escenarios más realistas
  - Sun Microsystems migró el proyecto de Wonderland a jMonkeyEngine.

Motor de juegos

- jMonkeyEngine ofrece…
  - SceneMonitor
  - MonkeyWorld 3D

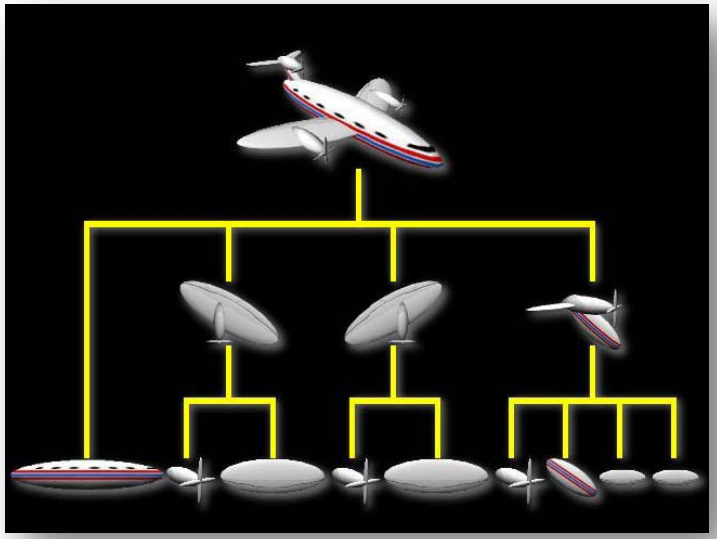


jMonkeyEngine

- API gráfica
- Satisfacer las carencias del resto de motores gráficos escritos en Java
- Permite unirse a cualquier sistema de renderizado
  - LWJGL
  - JOGL

jMonkeyEngine

- Grafos de escena:
  - Organizar los datos en una estructura de árbol.
  - Evita procesar las ramas que no aportan información en la visualización final.
  - Todos los nodos son genéricos.



jMonkeyEngine

- Arquitectura:

java.lang.Object
- com.jme.app.AbstractGame
  - com.jme.app.BaseGame
    - com.jme.app.BaseSimpleGame
      - com.jme.app.SimpleGame

java.lang.Object

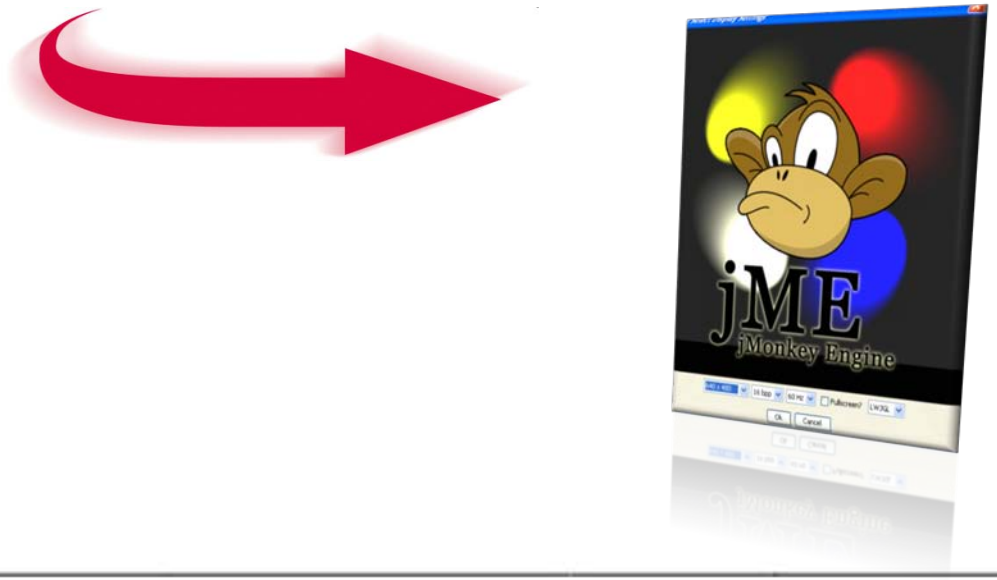
- Propiedades:

- Métodos:
  - `clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

jMonkeyEngine

com.jme.app.AbstractGame

- Propiedades:
  - **display**, finished, **settings**
- Métodos:
  - assertDisplayCreated, finish, getAttributes, getVersion, **setConfigshowMode**



com.jme.app.BaseGame

- Propiedades:
  - `throwableHandler`
- Métodos:
  - `getNewSettings, getThrowableHandler, setThrowableHandler,`
    `start`

com.jme.app.BaseSimpleGame

- Propiedades:
  - alphaBits, **cam**, depthBits, graphNode, input, lightState, pause, **rootNode**, samples, showBounds, showDepth, showGraphs, showNormals, statNode, stencilBits, timer, **tpf**, wireState
- Métodos:
  - cameraParallel, cameraPerspective, **cleanup**, initGame, initSystem, quit, reinit, setupStatGraphs, setupStats, **simpleInitGame**, **simpleRender**, **simpleUpdate**, updateInput

com.jme.app.SimpleGame

- Propiedades:
- Métodos:
  - `doDebug, render, update`

jMonkeyEngine

jMonkeyEngine

jMonkeyEngine

jMonkeyEngine

jMonkeyEngine

www.jmonkeyengine.com
www.youtube.com/user/jMonkeyEngine

¡¡Muchas gracias!!