

Team 5 Racing: Uribe 2K

Grupo 5: Marcos Pianelli y Rodrigo Rearden

14/02/11

Introducción

Team 5 Racing: Uribe 2K (en adelante T5R) es un juego de carreras por tiempo implementado sobre la version 3 en estado Alpha de jMonkeyEngine para Java, en este proyecto se buscó integrar diferentes features que muestran el poder del engine, tales como luces y sombras, materiales, carga dinámica de assets e integración de, entre otros.

Características implementadas

- Se usó la version 3 de JMonkeyEngine, a pesar de encontrarse en estado Alpha, se la consideró ampliamente superior a la versión anterior, más ordenada, documentada y soportada en los foros de ayuda del sitio del motor.
- Se empleó un sistema de checkpoints señalizados con esferas para asegurarse que el jugador no tome “atajos” invalidos, cada checkpoint recompensa con tiempo extra.
- El puntaje final se calcula tomando el tiempo restante que le queda al jugador una vez terminada la carrera
- Para la interfaz de usuario se utilizó Nifty GUI, una biblioteca de GUI montada sobre OpenGL incluida como novedad dentro de jME 3, se encuentra en constante desarrollo.
- El chasis del auto es metalizado, gracias al material reflectivo incluido en el modelo del utilizado
- Posibilidad de cambiar de cámara (Apretar ‘Enter’ durante la carrera). Esta característica fue de fácil implementación, dado que desde el comienzo del proyecto se contó con una buena clase de posicionamiento de cámara.
- Posibilidad de elegir dos skyBoxes diferentes (Apretar ‘Sky’ en el menu)
- Generación de texturas para el terreno en forma procedural (asignar valor true en la propiedad regenerateTextures en el archivo Assets/Text/configuration.properties).
- Se implemento efectos de fade-in y fade-out entre el menú y el juego.
- Se pueden tomar screenshots presionando la tecla “P” la imagen se guarda en el archivo “capture.png” en la carpeta donde corre el ejecutable.
- Se implementaron árboles y arbustos utilizando billboards con un canal alpha.

Problemas encontrados

- No se logró configurar correctamente los materiales del terreno y las construcciones mediante el formato de archivo de materiales del engine (.j3md y .j3m), incluso los ejemplos de iluminación y bump mapping de la versión usada de jME 3 no funcionaban correctamente. En cambio el modelo del auto funciona correctamente ya que usa el formato Ogre XML, de todas formas no se consiguieron buenos exportadores a este formato para poder incluir los modelos de pista desarrollados.
- Los formatos de modelo soportados por jME 3 son extremadamente limitados, solo

OBJ, Ogre XML y j3o (el formato de modelos propio del engine), fue necesario utilizar Maya para exportar los modelos desarrollados en 3D Studio Max a OBJ.

- Además, los formatos OBJ y Ogre XML mencionados anteriormente solo son cargados por el engine en modo de debugging, el release final ignorará todo modelo que no esté en el formato j3o, afortunadamente el conversor viene integrado en la IDE de jME y funciona sin ningún tipo de problemas.
- El estado Alpha del engine hizo sentir la presencia de bugs y cambios en los contratos en el engine a través del tiempo de desarrollo, nos sucedió que luego de congelar la versión utilizada de jME 3 algunas características no disponibles comenzaron a funcionar o mejorarse en versiones posteriores. Este fue el caso de la integración de luces, materiales y de la GUI a través de NiftyGUI.

Formatos de Archivo

- Para los parámetros de configuración del juego se utilizó un archivo de propiedades de java, estos incluyen entre otros, características del auto, dirección de la luz solar y parámetros de las sombras de jME 3
- Los puntajes de la tabla de high scores se guardan en el archivo "Assets/Text/Scores.txt", cada línea está formada por el nombre del jugador (3 caracteres) y su puntaje (un número entero) separado por un espacio.
- La posición de los Checkpoints de la pista se guardan en el archivo "Assets/Text/Checkpoints.txt", cada línea es la posición de cada checkpoint, formada por tres valores flotantes (x, y, z) separados por espacios
- El valor de tiempo extra que otorga cada Checkpoint se guarda en el archivo "Assets/Text/TimeExtensions.txt", cada línea es un flotante que indica el tiempo extra que otorga pasar por cada checkpoint.
- La posición de los árboles formados por billboards se guarda en "Assets/Text/Trees.txt" con el mismo formato descripto para los Checkpoints
- La geometría de la pista se encuentra guardada en varios archivos OBJ en "Assets/Models/Track", los archivos de representación visual son:
 - Advertisements.obj
 - Crowd.obj
 - Grass.obj
 - GuardRails.obj
 - Seats.obj
 - Track.obj

mientras que el archivo "TrackCollisions.obj" contiene una malla de colisión que es usada para determinar las partes sólidas de la pista.

- Los materiales y texturas utilizados para la geometría descripta se encuentran en las carpetas "Assets/Materials" y "Assets/Textures" respectivamente
- Para la GUI se utilizaron archivos XML que son consumidos por NiftyGUI