Trabajo Práctico Final

Guillermo Campelo
Juan Ignacio Goñi
Juan Tenaillon
Santiago Vazquez

10 de julio de 2010

Resumen

El siguiente informe describimos como diseñamos e implementamos el **Rally Uribe 100K**. Mediante el uso de imágenes describimos cómo fueron diseñadas e implementadas las distintas partes que componen al total de la aplicación. También presentaremos los inconvenientes que surgieron a lo largo del desarrollo y cómo decidimos solucionarlos. Por último explicaremos cómo se utiliza el juego, cómo se modifican las distintas opciones de configuración del mismo y las conclusiones.

Índice

1. Introducción

Haciendo uso del Framework JMonkey, diseñamos e implementamos un juego de rally, llamado Rally Uribe 100K. A continuación, se describe cómo fue realizado el mismo, qué características posee y la forma en que se utiliza.

Para el diseño e implementación del juego, partimos de la base del ejemplo proporcionado por el Framework utilizado, y tomando esa base, realizamos el desarrollo de las nuevas características.

En la sección 2 explicaremos cómo es el juego, los distintos estados, cómo es el sistema de puntaje y los sonidos del mismo.

En la sección 3 explicaremos qué características adicionales desarrollamos para el juego, como las texturas procedurales, los screenshots, los arbustos billboard, los skybox intercambiables y demás.

En la sección 4 explicaremos cómo se utiliza el juego, explicando las teclas a utilizar por defecto, además explicaremos cómo utilizar las distintas opciones del menú.

En la sección 5 mostraremos las distintas opciones de configuración, así como los valores recomendados para la misma. También explicaremos cómo definir nuevas pistas, para que el usuario del mismo pueda crear sus propios recorridos y utilizarlos con el juego.

Por último, en la sección 6 presentamos las conclusiones que alcanzamos, además de posibles extensiones al mismo y la enseñanza obtenida del diseño y desarrollo del mismo.

2. El Juego

En esta sección explicaremos el juego como tal. El objetivo, el modo de juego, los distintos estados, cómo es el sistema de puntajes y los sonidos del mismo.

2.1. Descripción

El juego consiste en recorrer una pista de carreras, ambientada como si fuera de Rally. El ambiente en el cual se mueve el auto, está compuesto de un terreno, con una pista delimitada, con un conjunto de objetos que la componen.

El objetivo del mismo, es dar tres vueltas al recorrido propuesto. Además de completar la totalidad de las vueltas, el jugador debe transitar por la totalidad de los checkpoints, en el orden determinado por la pista. En caso de saltearse uno el juego no le permitirá transitar exitosamente por los siguientes, y el jugador deberá volver hacia atrás y transitar por el checkpoint adeudado.

Al inicio, el auto se encuentra estacionado en su *box*. Se decidió esto debido a que en el Rally, los autos no necesariamente realizan un recorrido cerrado, y no arrancan desde una grilla de partida comun y corriente como en la fórmula 1, sino que arrancan solos y desde un lugar predeterminado por la organización. Es por eso que nosotros elegimos que el auto arranque en ese lugar, también llamado *boxes*.

Al finalizar las tres vueltas, el jugador recibirá el tiempo de su recorrido, y mientras más bajo sea ese tiempo, más chances tendrá de ingresar al listado de los puntajes más altos.

Figura 1: Menú Principal.

Figura 2: Menú de Opciones.

2.2. Modo de Juego

Hay un solo modo de juego, y este consiste en intentar realizar el menor tiempo posible en la realización de las tres vueltas a la pista. Como comentamos con antelación, es necesario transitar en el orden establecido a través de los checkpoints para poder terminar con el recorrido.

2.3. Estados del Juego

Hay tres estados diferenciados en el juego. Antes de comenzar a jugar, durante el juego y luego de jugar.

Antes de comenzar a jugar, el jugador encuentra el menú principal de la aplicación, el cual le presenta todas las opciones disponibles. Una vez que el usuario selecciona la opción para comenzar a jugar, realizamos la transición al siguiente estado, el juego en sí.

En este estado, es cuando el jugador maneja el vehículo a través del camino propuesto. El vehículo en esta ocasión, puede colisionar contra los distintos objetos que componen la escena, como los árboles, los límites del terreno y las pirámides presentes en el recorrido. Una vez finalizado el recorrido, realizamos la transición al tercer y último estado del juego.

En este tercer y último estado, le presentamos al usuario el menú nuevamente, donde podrá ver los puntajes máximos hasta el momento o comenzar nuevamente el juego para tratar de mejorar su tiempo.

2.4. Puntajes

El juego tiene la opción de ver quienes fueron los jugadores que realizaron los diez mejores tiempo en el recorrido a la pista.

Una vez finalizado el recorrido, le presentamos al jugador su tiempo total de recorrido, y si el tiempo es lo suficientemente bueno el mismo ingresa en la lista de los mejores tiempos, donde el identificador es la fecha en que fue realizado el record.

En el menú inicial del juego, el jugador puede elegir la opción de *highscores*, para ver cuales son los mejores tiempos a la pista y cuando lo realizó.

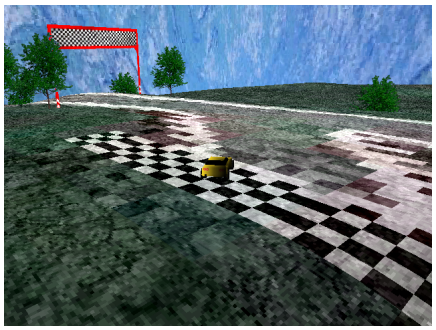

Figura 3: Grilla de Partida del Rally Uribe 100K. Bajo nivel de detalle.

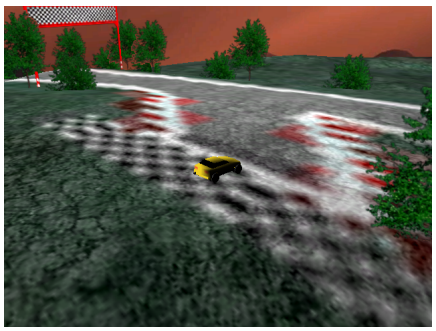

Figura 4: Grilla de Partida del Rally Uribe 100K. Alto nivel de detalle.

2.5. Sonidos

El juego, como todos los de su tipo, tiene diferentes sonidos que ayudan a ambientar el mismo.

Comenzando por los sonidos de ambiente, donde decidimos que se iban a utilizar dos pistas para tener cierto cambio en la música de ambiente pero sin entrar en la exageración de utilizar muchas.

También fueron utilizados dos sonidos para ser utilizados por el vehículo para denotar situaciones en las que se está acelerando y situaciones en las que no.

Por último, utilizamos un sonido para los momentos en que el vehículo colisiona contra algun objeto presente en el terreno.

Todos los sonidos propios del vehículo y su entorno fueron obtenidos de Internet, de páginas de uso libre y gratuito. Los temas utilizados para la música ambiental, corresponden a la autoría de Santiago Vazquez, integrante del equipo.

2.6. Iluminación

El terreno en el cual se desarrolla el juego, cuenta con tres tipos distintos de iluminación: una luz ambiente, una luz direccional y cinco luces puntuales que iluminan la totalidad del recorrido.

La iluminación ambiente, está configurada por defecto para tener un nivel bajo de iluminación. Elegimos esto para que el resto de las luces tomen un papel más importante en la iluminación del escenario.

Los valores de rojo, verde, azul y canal alfa que determinan la intensidad y color de la luz, son fácilmente configurables en el archivo *global.properties*, bajo el elemento $TRACK1.AMBIENT.LIGHT$.

La luz direccional la configuramos por defecto con un color rojo intenso y está emplazada encima del primer checkpoint del recorrido. La razón de esta decisión radica en que teníamos que buscar un lugar donde la luz fue visible para demostrar la existencia de la misma, es por eso que cada vez que el auto transita por debajo de la grilla de partida, se puede ver un color rojizo en el techo del mismo.

Esta luz es también configurable en cuanto a los colores rojo, verde, azul y el canal alfa. Para configurar esto, es necesario modificar en el archivo de configu-

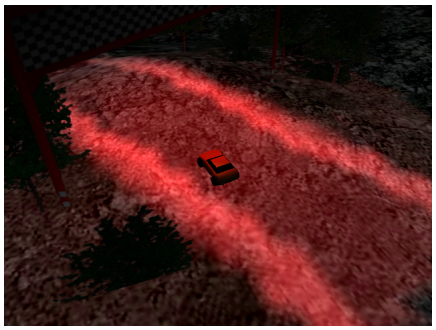

Figura 5: Juego con luz ambiental tenue y luz direccional roja.

Figura 6: Juego con todas las luces habilitadas.

ración, las propiedades que se encuentran bajo el elemento $TRACK1.LIGHT.SPOTLIGHT$.

Por último, las luces puntuales. Las cinco luces puntuales están en distintos lugares a lo largo del recorrido. A diferencia de los otros dos tipos de iluminacón, estos tienen una posición determinada por el usuario y están configuradas para dar una luz blanca.

Tanto la posición de la luz como el color, son configurables. Al igual que el resto de las luces, para cambiar la posición y color de estas, hay que modificar las propiedades de cada luz en el archivo de configuración. Cada luz puntual está contemplada bajo la propiedad $TRACK1.POINTLIGHTn$ donde n es el número de la luz.

En las figuras 5 y 6 apreciamos el juego con las luces puntuales deshabilitadas y con las luces puntuales habilitadas respectivamente. Cabe destacar que en la figura 5 observamos una luz ambiental muy tenue junto con la luz direccional en color rojo apuntando hacia abajo sobre el primer checkpoint.

3. Características Extra

Dentro de las características opcionales a implementar, nos inclinamos por las siguientes: texturas procedurales, capturas de pantallas, árboles con billboards, diferentes skyboxes, cámaras estáticas, velocimetro y mapa de posicionamiento.

3.1. Texturas Procedurales

Para la realización de las texturas procedurales, el desarrollo fue basado en el diseñado para la extensión del *Trabajo Práctico Número 2 - Ray Tracer*. Se implementaron dos texturas procedurales, piedra y marmol.

Donde en la figura 7 y 8 se puede observar las dos texturas procedurales realizadas, marmol y piedra respectivamente.

Para las mismas, utilizamos Perlin Noise. Para más detalle, remitirse al informe del *Trabajo Práctico Número 2 - Extensión*.

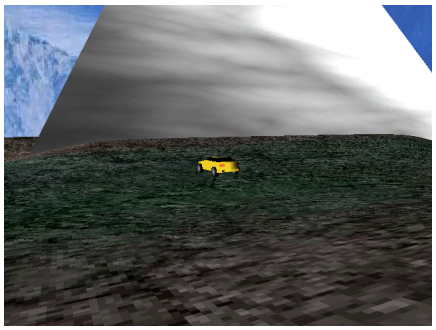

Figura 7: Textura Procedural: Marmol.

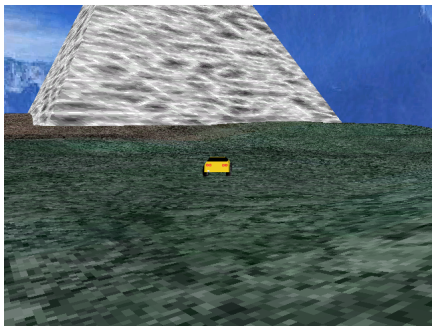

Figura 8: Textura Procedural: Piedra

3.2. Capturas de Pantalla

Otra de las características que desarrollamos, fue la posibilidad de obtener capturas de pantallas en cualquier momento del juego. Para realizar dicha acción, hay que presionar la tecla 0 (cero) en cualquier momento del mismo. Las imágenes producidas, son almacenadas en la carpeta *screenshots* presente en la raíz del proyecto, y el nombre con el que se guardan es la fecha en que fueron realizadas.

Esta característica la utilizamos mucho a la hora de realizar este informe.

3.3. Billboards

Otra característica que implementamos, fue el hecho de contar con árboles y arbustos realizados con *billboards* que nos proveyó el Framework de JME.

En las imágenes presentadas con antelación y más precisamente en la figura 9, puede observarse cómo este tipo de árboles son utilizados a lo largo de todo el recorrido; donde también cabe mencionar que el vehículo puede colisionar contra este tipo de texturas.

Algo interesante sobre este tipo de texturas es que nos brindan la posibilidad de verlas correctamente desde cualquier ángulo, además permitiendo que se vean con claridad las otras texturas detras de ellas.

La diferencia entre los árboles y los arbustos, es que estos últimos no son colisionables. Es decir, uno puede atravesar un arbusto pero uno no puede atravesar un árbol ya que se chocaría con el tronco del mismo.

En el escenario presentado por defecto, hay árboles y arbustos alternados.

3.4. Uso y Selección de distintos *Skyboxes*

Uno de los opcionales seleccionados para el desarrollo del juego, fue darle al usuario la posibilidad de elegir distintos skyboxes. Se ofrecen tres, uno para el día, otro para la tarde y el último para la noche.

Cuando el jugador selecciona la opción de $NewGame$, tiene la posibilidad de seleccionar el momento del día en que quiere correr la carrera. Esto se verá reflejado en la carrera en si, una vez que comience la misma. En todas las imágenes presentes en el informe, se puede apreciar el skybox que pertenece al momento *día* mientras que en la figura 6 se puede apreciar un cielo rojizo, simulando el atardecer.

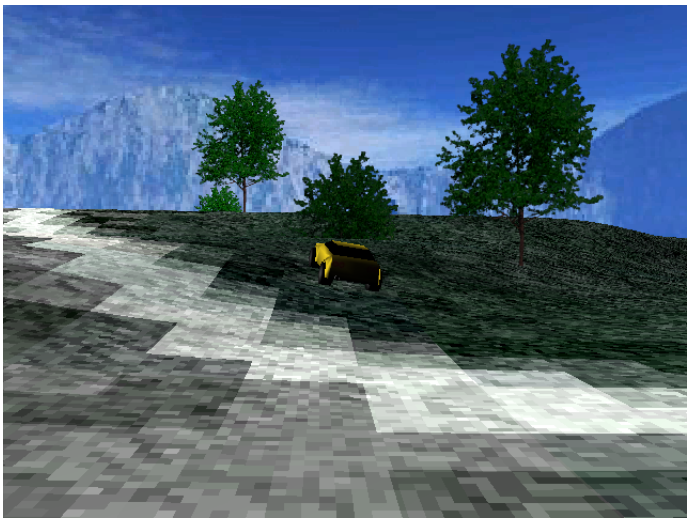

Figura 9: Árboles y arbustos billboard.

3.5. Cámaras Estáticas

A lo largo de la carrera, el jugador puede presionar la tecla C y esto va a cambiar la cámara desde donde se ve la carrera. En cada checkpoint hay una cámara estática que está apuntando al mismo.

En la figura 10 observamos una cámara estática en el segundo checkpoint.

3.6. Mapa de Posicionamiento

Otra característica que decidimos implementar, fue el mapa de posicionamiento en pantalla. Durante el juego, está presente un mapa en la parte inferior izquierda que nos indica la posición en la que estamos.

En este mapa se pueden diferenciar tres tipos de puntos bien diferenciados. Con amarillo, el auto del jugador, con color rojo, la grilla de partida y con color naranja, los distintos checkpoints presentes a través de la pista.

En la figura 11 observamos, abajo a la izquierda, el mapa de posicionamiento en la pista.

3.7. Velocímetro

El último opcional que decidimos implementar, fue el velocimetro con una aguja. Para realizar esto, utilizamos una imagen de un velocimetro sin la aguja, y un cuadrilatero por separado. Una vez que tuvimos esto, lo posicionamos en su lugar y luego le aplicamos a la aguja, una rotación dependiendo de la velocidad actual de vehículo.

Dependiendo de la velocidad lineal del vehículo, hacemos una traducción de ese valor lineal a un ángulo entre $-112{,}5$ y $112{,}5$ grados, que nos da la rotación de la aguja para un velocimetro que tolera velocidades entre 0 y 160 kilometros por hora.

En la figura 11 observamos, abajo a la derecha, el velocímetro del auto.

Figura 10: Cámara estática en el segundo checkpoint

Figura 11: El Juego.

4. Modo de Uso

En esta sección, comentaremos la forma de jugar al Rally Uribe 100K. Cabe destacar que estas instrucciones están basadas en la configuración default del juego, ya que cabe la posibilidad de modificar las teclas a utilizar.

4.1. Cómo se Juega?

Para jugar al juego, solo es necesario conocer las teclas para moverse a traves de la pista. Lo único necesario es saber con qué teclas avanzar, retroceder, girar a la izquierda y a la derecha. Luego, solo es necesario atravesar los diferentes checkpoints hasta llegar a la meta.

Por defecto, las teclas para mover el vehiculo a traves del terreno son las flechas hacia arriba para acelerar, hacia abajo para desacelerar y ir hacia atras, hacia la izquiera para doblar en esa dirección y hacia la derecha para doblar a la derecha.

Durante el juego, es posible pausar el mismo mediante la tecla P, ir al menú preseionando la tecla ESC y sacar *screenshots* con la tecla 0.

Además, es posible cambiar la cámara con la que se ve el vehículo a otras cámaras estáticas mediante la tecla C.

4.2. Los Menúes

Para movernos a través de los menúes, es necesario utilizar las flechas hacia arriba, abajo, izquierda, derecha y el enter. Con las flechas de arriba y abajo, marcaremos la opción deseada, mientras que con la tecla de enter, seleccionaremos la misma.

En las opciones del menú que contengan distintas posibilidades de configuración, ya sea el momento del día en cual vamos a correr, estado del sonido y demás opciones, es necesario utilizar las flechas hacia la derecha e izquierda para cambiar la opción.

5. Configuración

El juego, como mencionamos con antelación, tiene ciertas características que son configurables. Desde las teclas, las texturas de los skyboxes, las características de las pistas y las pistas en sí son configurables.

Todas estas características configurables, pueden ser modificadas desde el archivo *global.properties*.

5.1. Configuración del Juego

Desde el archivo previamente mencionado, es posible configurar todas las teclas del juego además de los nombres y texturas de los skyboxes, el estado de la música y los efectos como las pistas a escucharse ante cada evento y/o pistas de la música ambiental. Además, tanto para la música como para los efectos, es posible configurar en qué nivel quiere uno que se encuentren, del 0 al 100, siendo 100 el volumen más alto de la música o efecto.

5.2. Teclas

Desde el archivo de configuración es posible modificar las teclas que van a ser utilizadas para jugar al juego. Las teclas vienen configuradas por defecto, pero modificando los valores en el archivo *global.properties* es posible modificarlas.

```
FORWARD=C8
BACKWARD=D0
STEERLEFT=CB
STEERRIGHT=CD
HORN=23
AXIS=2F
SCREENSHOT=32
PAUSE=19
ARROWUP=C8
ARROWDOWN=D0
ARROWLEFT=CB
ARROWRIGHT=CD
SELECT=1C
BACK=0E
```

Los valores que tiene cada opción, son un número hexadecimal que está mapeado a una tecla. Se eligió esta configuración por que es la que maneja la clase de Java que nos brinda esta funcionalidad.

Tanto las teclas para mover el vehículo en el juego como las teclas para moverse en el menú, están configuradas de la misma forma. Esto es debido a que utilizan las flechas del teclado. También sería posible si el usuario quisiera, cambiar las teclas con las que maneja el vehículo o las teclas con las que se mueve en el menú.

5.3. Configuración de Nuevas Pistas

Desde el archivo de configuración, es posible determinar la posición de los distintos objetos que componen a la escena.

Los árboles, checkpoints, pirámides con texturas procedurales, como el resto de los objetos presentes, pueden ser configurados fácilmente desde el archivo de configuración.

Para configurar los árboles presentes en la escena, es necesario indicar qué cantidad de árboles habrá, y acto seguido, indicar para cada uno de esos árboles, que posición van a ocupar en el terreno.

Un ejemplo del archivo de configuración para la determinación de la posición de los árboles es:

```
TRACK1.TREE.COUNT=425
TRACK1.TREE1.POS=672, -846
TRACK1.TREE2.POS=664, -859
TRACK1.TREE3.POS=642, -889
...
TRACK1.TREE425.POS=1642, -1889
```

Donde se puede observar cómo en la primera línea, se le indica la cantidad de árboles presentes en la escena y luego, para cada uno de los árboles, una posición dada por dos puntos.

Para el caso de los checkpoints, es similar a los árboles pero con una pequeña variación, dada por la necesidad de determinar la rotación del plano que contiene al checkpoint.

Un ejemplo del archivo de configuración para la determinación de la posición de los checkpoints es:

```
TRACK1.CHECKPOINT.COUNT=8
TRACK1.CHECKPOINT1.POS=515,-806
TRACK1.CHECKPOINT1.ROT=6.68925
TRACK1.CHECKPOINT2.POS=-4434,-3150
TRACK1.CHECKPOINT2.ROT=6.32625
TRACK1.CHECKPOINT3.POS=694,-4591
TRACK1.CHECKPOINT3.ROT=4.748
TRACK1.CHECKPOINT4.POS=3880,-3081
TRACK1.CHECKPOINT4.ROT=3.604
TRACK1.CHECKPOINT5.POS=4094,-138
TRACK1.CHECKPOINT5.ROT=3.126
TRACK1.CHECKPOINT6.POS=2069,1334
TRACK1.CHECKPOINT6.ROT=7.718
TRACK1.CHECKPOINT7.POS=-674,4184
TRACK1.CHECKPOINT7.ROT=2.126
TRACK1.CHECKPOINT8.POS=-4363,3862
TRACK1.CHECKPOINT8.ROT=6.451
```

Donde aca podemos observar como están presentes los 8 checkpoints utilizados en la pista presentada, donde cada uno tiene su posición y su rotación en radiantes.

Otra posibilidad sería la de configurar las pirámides con texturas procedurales. De forma similar que los ejemplos presentados con antelación, un ejemplo de una entrada en el archivo de configuración sería:

```
TRACK1.PIRAMID.COUNT=2
TRACK1.PIRAMID1.POS=2858,2142
TRACK1.PIRAMID1.TYPE=Marble
TRACK1.PIRAMID2.POS=1850,-150
TRACK1.PIRAMID2.TYPE=Stone
```

Donde además de indicar la cantidad de pirámides que van a haber en el terreno, es necesario indicarle que textura utilizaran. Previamente en el informe comentamos la existencia de dos tipos de texturas procedurales, piedra y marmol, tal como se pueden apreciar en el archivo de configuración de la pista.

Para realizar la pista, el Framework de JME nos provee de una función para realizar terrenos. Uno le indica la altura máxima y mínima y el terreno es generado por el Framework. Para modificar la textura del terreno, utilizamos una imagen obtenida de Google Maps y modificada para que sea una pista de carreras. Hay dos archivos utilizados para la pista, *autodromo2.png* y *autodromo2low.png*. El primero es la pista con alta nivel de detalle y el segundo es la pista con un nivel de detalle más bajo.

6. Conclusión

A lo largo del desarrollo del juego, pudimos adquirir varios conceptos y enseñanzas del proceso que conlleva crear y desarrollar uno.

Para comenzar, adquirimos conocimientos sobre las distintas partes que componen a un juego, las distintas fases de los mismos y además, logramos adquirir cierta experiencia utilizando un framework para el desarrollo de videojuegos en Java.

Cabe destacar que debido a que decidimos utilizar JME2 para el desarrollo del juego, y que ante cada problema o situación en la que requerimos asistencia, logramos sortear el inconveniente gracias a la extensa documentación sobre el Framework que está disponible en los distintos foros y en la misma documentación propia de los desarrolladores del mismo.

Por último, también nos parece valioso remarcar que a lo largo del desarrollo nos encontramos con varias disyuntivas las que nos llevaron a trabajar, discutir y decidir en equipo cual era la mejor solución para un problema dado. De esta forma, pudimos desarrollar nuestra capacidad colaborativa y de trabajo en equipo.