

Simulador 3D para Manejo de Vagones del Transporte Colectivo Metro

Batalla González Edgar Said, Gabriel Sepúlveda Cervantes y Portilla Flores Edgar

Centro de Innovación y Desarrollo Tecnológico en Cómputo

I.P.N. México D.F.

said_batalla@hotmail.com, gsepulvedac@ipn.mx

Resumen

Este artículo presenta la simulación con características físicas y lógicas de una maquina combinada con elementos de un entorno virtual para entrenamiento de conductores. El desarrollo permite la obtención de una herramienta capaz de apoyar a un individuo en el desarrollo de sus habilidades y conocimientos respecto a la maquinaria, permitiendo entender el manejo real del elemento simulado.

La aplicación final del desarrollo proporcionará una herramienta que auxilie al personal del Sistema de Transporte Colectivo Metro en su capacitación en el manejo de los vagones; con el manejo de un sistema de realidad virtual

Palabras clave: Realidad Virtual, Simulación, Entrenamiento Virtual.

1. Introducción

Gracias a los beneficios que nos proporcionan los avances en la tecnología se puede reducir significativamente el tiempo de capacitación, en actividades que pueden ser de riesgo y puedan costar pérdidas humanas y materiales, llevando el entrenamiento del personal a un nivel diferente de percepción, donde por medio de una herramienta desarrollada por computadora, podemos reproducir una ruta que logre aparentar ser la ruta real original por la que podría conducir, de modo que le sea posible interactuar libremente en un mundo virtual y lograr así acceder a la información necesaria las veces que se quiera sin problema alguno.



Figura 1. Simulación del Metro de Bilbao.

Las mejoras en ambientes 3D no solo en el diseño de los modelos sino en las incorporaciones de factores físicos e implementaciones de algoritmos de Inteligencia Artificial (IA), permiten actualmente generar un verdadero evento aleatorio el cual puede llegar a representarse de manera exacta lo que ocurriría en una situación real.

Existen diversos simuladores con funciones semejantes a las de Sistema de Transporte Colectivo-Metro (STCM) entre las cuales se pueden mencionar: El simulador de conducción y averías del metro de Bilbao [1]. La cual es una herramienta de adiestramiento práctico que permite instruir en la identificación y resolución de averías, mejorar los hábitos de conducción y ayudará a solventar con mayor precisión y agilidad todas las incidencias posibles que puedan alterar la normal circulación de los trenes.

De igual forma Code3D (TM) de Ops Sim Studios [2]. Es un conjunto de herramientas que está disponible para los servicios de emergencia donde es posible crear escenarios propios 3D. En Code3D los usuarios pueden crear su propia formación, el medio ambiente; añadir los peligros y las fotos que se pueden cambiar con el tiempo. Los usuarios pueden utilizar la función de reproducción de caminar por el medio ambiente y el escenario en 3D en tiempo real.



Figura 2 Simulador de Emergencias Code3D.

Bajo el motor gráfico de UDK el desarrollo de una herramienta de capacitación como lo es el simulador de manejo propone una mejora en la calidad del manejo en gráficos, superior a los mencionados, combinado con la capacidad que tiene el entorno para poder manejar la instrumentación de un dispositivo de control permiten el desarrollo de una aplicación de entrenamiento dedicada desde su creación para las necesidades del STCM.

2. Modelado 3D y Desarrollo

Para crear el simulador con características realistas se requiere de un sistema que pueda simular todos los elementos con los que cuenta la ruta que recorre el metro, así como la cabina de manejo y proporcionar al conductor todos los elementos necesarios para percibir un entorno semejante a aquel donde se desarrollará sobre la operación del vagón real y lograr una capacitación adecuada.

2.1 Objetivo.

Desarrollar un simulador que permita al personal del Sistema de Transporte Colectivo Metro capacitar al personal en el manejo de Actualmente el Sistema de Transporte Colectivo Metro cuenta con dos simuladores los cuales proporcionan: la simulación del monitoreo de las redes de transporte y flujo de vagones, y un simulador del manejo de la cabina de control para los vagones del metro. Sin embargo el simulador actual de manejo con el que cuenta permite la interacción únicamente con un video que es desplegado y sincronizado con una cabina sin permitir interacción fuera de la permitida en el video, lo cual es significativo para el entrenamiento.

2.2 Metodología.

Aunado a esto, en el ámbito profesional actual, el modelo orientado a objetos es el más utilizado en la industria del software en la actualidad, gracias a sus grandes bondades y flexibilidad, ofreciendo un diseño de software de calidad y acorde a los resultados esperados al aplicarlo de manera correcta.

Los procesos que se asumirán de acuerdo al ciclo de vida del software serán:

- El Proceso de Adquisición en sus fases de inicio, preparación de la solicitud de propuestas, aceptación y finalización.
- Proceso de Desarrollo. Todas las actividades a desempeñarse sobre el proyecto se acoplan a este proceso por lo que es apta su implementación en cierto grado.
- Proceso de Operación. Implementación del proceso, Pruebas de operación y Operación del sistema son fases necesarias en el desarrollo de este proceso ya que proporcionaremos un servicio para la operación de un entorno virtual a un usuario.
- Proceso de instrumentación de un dispositivo para el manejo del simulador.
- Proceso de Documentación. El registro de cada actividad debe de estar documentado y registrado.
- Proceso de Aseguramiento de la calidad. Aseguramiento del producto y aseguramiento del proceso, deberá cumplir con lo especificado en el producto así como lo estipulado en el calendario de actividades.

2.3 Lenguaje de Programación.

Unreal Development Kit (UDK) cuenta con su propio lenguaje de programación orientado en scripting, UnrealScripting (UScript), basado en el lenguaje de programación C++ Además cuenta con la capacidad de extender las funcionalidades originales mediante el uso de librerías de enlace dinámico (DLL) las cuales extienden las capacidades del software para comunicarse con dispositivos externos, los cuales servirán como interfaces entre el usuario y el sistema.

2.4 Diseño y Modelado.

Para el modelado de los elementos encontrados en la estación se tuvieron que realizar una sesión previamente autorizada de fotos. Gracias a estas fotos se lograron obtener las bases para cada modelo del simulador. Los modelos de los elementos presentados en el simulador se generaron en Blender 3D [4].



Figura 3 Andén del Metro, dirección Pantitlán.

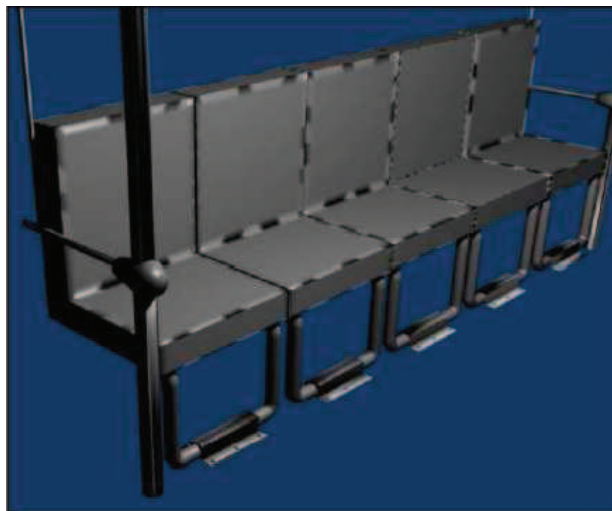


Figura 4 Renderizado de un modelo hecho en Blender 3D.

UDK proporciono los elementos necesarios para la generación de puntos de luz, colisiones.

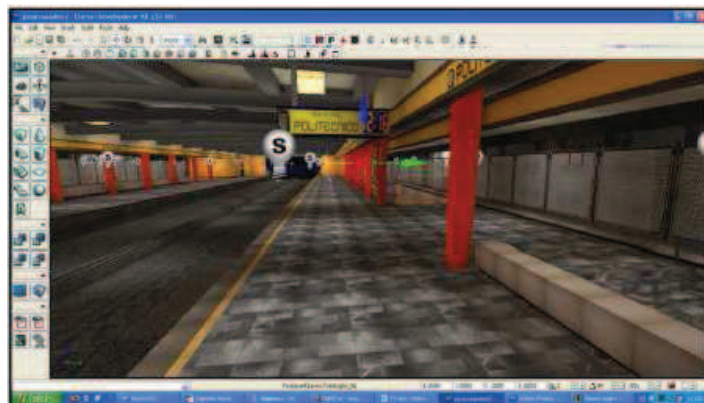


Figura 5 Escena de los andenes de la estación Politécnico desarrollados en UDK.

2.5 Generación de eventos.

Para manejar la simulación del control de un vagón es necesario manejar un evento, el cual es una acción definida sobre una herramienta propia de UDK. Este evento se controla a partir de un trigger (Figura 6), que dispara la acción cuando esta sea tocada en un área definida por su caja de colisiones sobre el espacio.

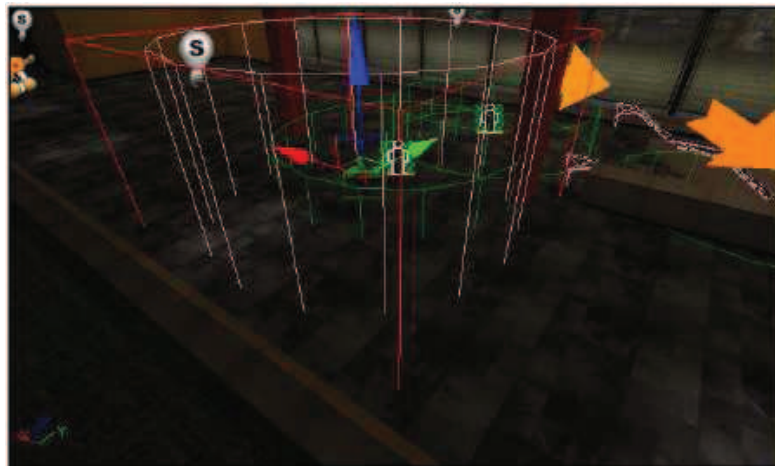


Figura 6 Caja de colisiones de eventos.

Esta acción es definida sobre un editor especial denominado Kismet (Figura 7). Gracias a este editor se pudo crear la AI de los integrantes de la simulación, como usuarios, vendedores o demás personajes controlados por el sistema y no por el usuario.

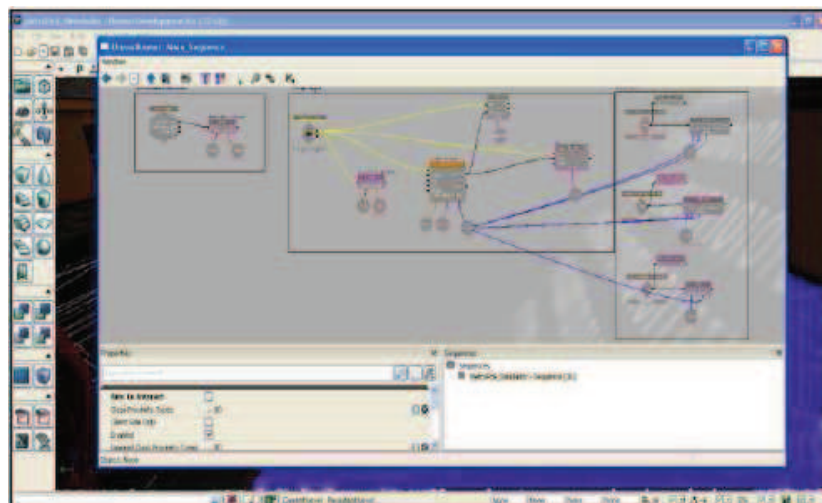


Figura 7 Área de trabajo UnrealKismet.

Las acciones se definen dentro del Kismet como una secuencia animada de movimiento que interactuara con el recorrido sujeta a cambios en el mismo recorrido a partir de las propiedades físicas que se otorguen a la cabina para generar un recorrido más realista, por ejemplo la simulación presentará mayor inercia si se consideran los vagones llenos de usuarios a diferencia de un vagón vacío.

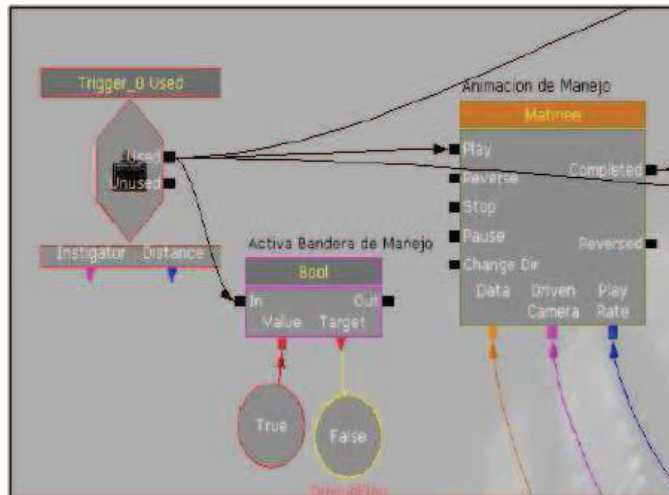


Figura 8 Gatillo lanzador del evento.

El recorrido entre la estación Politécnico e Instituto del Petróleo de la Línea 5 del STCM, se desarrolló con base en las fotos y videos tomadas durante una serie de recorridos reales permitidos por las autoridades del STCM.



Figura9 Tramo del recorrido de las estaciones Politécnico a Instituto del Petróleo.



Figura 10 Tramo de la vía terminada.

3. Instrumentación

Para lograr la interacción entre el usuario y el simulador se requiere de un sistema de adquisición de datos (DAQ) por lo que se requirió de un dispositivo capaz de ser programado y reutilizado dentro del lenguaje de programación de UDK. Para lograr lo antes mencionado se consideró la tarjeta NI6009 de la compañía National Instruments [7], ver Figura 11.



Figura 11 Tarjeta de adquisición de datos NI6009.

La tarjeta de adquisición de datos NI6009 se programó utilizando el conjunto de Interfaces Programación de Aplicaciones (APIs) proporcionadas por la compañía National Instruments junto con los controladores de la tarjeta de adquisición de datos. El panel de control que se proporcionó al usuario cuenta con tres instrumentos de los más de 30 posibles dentro de una cabina real. Los instrumentos son la palanca de velocidades, la palanca de “hombre muerto” y el botón de reversa, esto se muestra en la Figura 12.



Figura 12 Panel de control.

La tarjeta de adquisición de datos se conectó con el panel de control utilizando dos entradas con convertidor analógico digital para la palanca de velocidades y la de “hombre muerto” y una entrada digital para el botón de reversa, ver Figura 13.

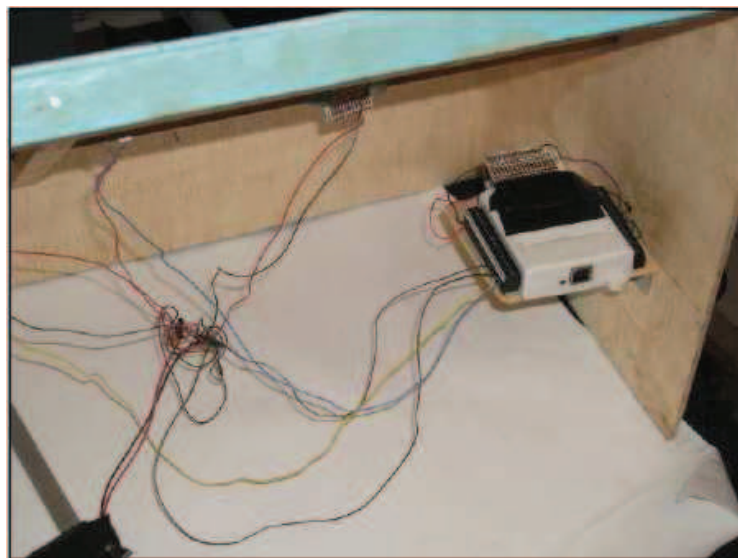


Figura 13 Simulación.

Finalmente las variables capturadas por el sistema de adquisición de datos se capturan mediante comunicación USB, son almacenadas mediante un programa desarrollado con las APIs propias de la tarjeta y finalmente son comunicadas a la aplicación del simulador mediante funciones programadas en una librería de enlace dinámico (DLL) creada especialmente para el proyecto, ver Figura 14.

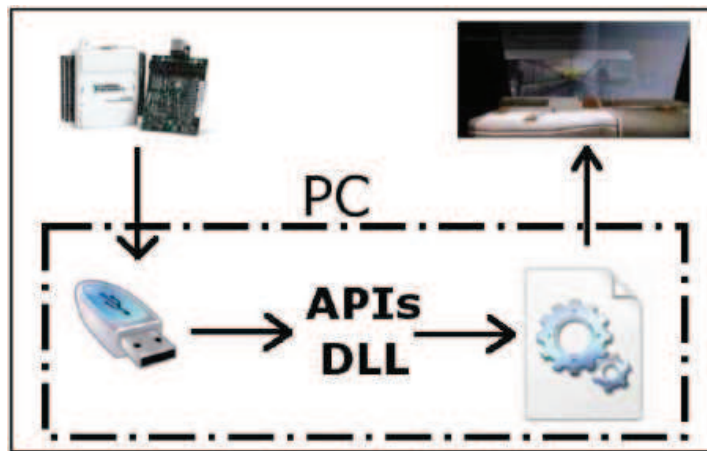


Figura 13 Conexiones para la adquisición y procesamiento de señales.



Figura 14 Simulación de la cabina.

4. Resultados.

Se obtuvo un simulador que proporciona una interfaz sencilla la cual permite a los empleados del STCM el manejo y visualización de los elementos con los que se interactúa durante el recorrido entre las estaciones presentadas en el simulador. El manejo de la simulación mediante un dispositivo que proporciona los elementos de frenado, avance a determinada velocidad hacia el frente y reversa.

5. Conclusiones y trabajo futuro.

El manejo de los eventos, renderizado, manipulación de objetos y comunicación entre el motor gráfico y la tarjeta de adquisición de datos proporcionan los elementos suficientes para generar un simulador de manejo que cumpla con los estándares de calidad necesarios. Los inconvenientes que trae consigo el manejo de un motor gráfico de esta calidad refiriéndose al consumo de recursos de memoria y video, son costeables frente al beneficio que acarrea.

Como planteamiento a futuro se espera modelar la continuación del recorrido sobre la Línea 5 del STCM así como la inclusión de situaciones de riesgo, como podrían ser incendios, sismos y/o suicidios sobre las líneas de transporte.

6. Agradecimientos.

Agradecemos al Sistema de Transporte Colectivo-Metro todas las facilidades proporcionadas para la creación del prototipo de manejo presentado en este artículo.

Referencias.

- [1] Sitio oficial metro de Bilbao. <http://www.metrobilbao.com/accesible/cas/metro/> simulador.html.
- [2] Jason Gregory "Game Engine Architecture". Editorial, Sales, and Customer Service Office, 2009.
- [3] Panda3D Free 3D Game Engine
<http://www.panda3d.org/showss.php?page=0&shot=ssg-code3d/code3D05>.
- [4] Blender Foundation. <http://www.blender.org/>.
- [5] Sitio oficial de GIMP. <http://www.gimp.org>.
- [6] Sitio Oficial de Unreal Development kit.
<http://developer.nvidia.com/object/udk.html>. [7] <http://www.ni.com/>