

Universidad de las Ciencias Informáticas

Facultad 5

Entornos Virtuales



Librería para la simulación de la lluvia
en un ambiente virtual en un simulador
de tiro.

Trabajo de Diploma para optar por el título de
Ingeniero en ciencias Informáticas

Autor(es): Gerda Carrillo Borroto.

Yanet Walters Romero.

Tutor: Lic.Liesner Acevedo Martínez.

Co-Tutor: Tte. Cor. Ing. José E. Villar Cociña

Ciudad de la Habana, Julio de 2007

Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.

Albert Einstein.

DECLARACIÓN DE AUTORÍA

Nosotros: Gerda Carrillo Borroto y Yanet Walters Romero declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yanet Walters Romero

Gerda Carrillo Borroto

Lic Liesner Acevedo Martínez

Firma del Autor

Firma del Autor

Firma del Tutor

Datos de Contacto

Tutor: Licenciado Liesner Acevedo Martínez.

E-mail: liener@gmail.com, frodo@uci.cu

Lic. Liesner Acevedo Martínez, categoría instructor. Asesor del departamento de técnicas de programación de la Universidad de las Ciencias Informáticas (UCI). Graduado de Licenciatura en Ciencias de la Computación en la Universidad Central “Marta Abreu de las Villas” (UCLV). Actualmente cursa un doctorado curricular en programación paralela y distribuida en conjunto con la CUJAE y la Universidad politécnica de valencia (UPV).

Agradecimientos

Queremos agradecer a nuestro tutor Liesner Acevedo por batallar con nosotras hasta el final y darnos aliento en todo momento. También a muchas personas que nos brindaron su ayuda para la realización de este trabajo, a Rigoberto Leander, Lázaro Abreu, Karel Pérez, Dagoberto Marrero, Frank Puig, a nuestros compañeros de aula que siempre se han preocupado por nuestros problemas y a todos los que hicieron posible este sueño. A todos ustedes le agradecemos y les aseguramos que forman ya parte de nuestros recuerdos. A nuestro Comandante Fidel y todos aquellos que viven por la UCI que hacen que este sueño sea realidad.

De Yanet

Quiero agradecer a mis amigos que siempre han estado conmigo en los momentos de agonía, en la tesis, a Cristina y a Brígida por sus ideas, a Rigo por formar parte de este trabajo, a Gerda por ser mi compañera de tesis y soportarme como amiga. A Suanny, Daili, René, Lenon y a las niñas del cuarto por apoyarme en todo.

De Gerda

Quiero darles las gracias a mi familia en especial a mi madre y a mi abuela que siempre estuvieron dándome ánimos en todos los momentos. A mi Titi por darme todo su cariño y soportarme en mis momentos de locuras. A Suanny por ser una amiga incondicional y aguantarme mis pesadeces. A Rigo que aunque no nos conocíamos nos dio su apoyo indispensable. Y por último quiero agradecer a Yanet por compartir este sueño conmigo y ser mi amiga.

Dedicatoria

De Yanet

A mi mamá por guiarme por el camino correcto.

A mis padres Jorge Luis y Samuel.

A mi hermana que quiero con la vida.

A los abuelos que viven y a los que hubiesen querido compartir este momento conmigo.

A mis tíos Román y José Ángel.

A todos mis familiares.

A todas las personas que me formaron en la EMCC de Camagüey.

De Gerda

A todos mis familiares pero en especial a mi mamá y a mi abuela que han sido la luz de mis ojos.

A mi hermano para que vea que todo es posible.

A mis seres queridos.

A mí que me la merezco después de cinco años de estudio.

Resumen

La Universidad de las Ciencias Informáticas en conjunto con las Fuerzas Armadas Revolucionarias (SIMPRO) desarrolla un simulador de tiro, el cual permitirá prácticas de tiro virtuales en diferentes vistas y contextos. Uno de los problemas que presenta el simulador es como representar la lluvia en cada una de estas vistas y como afecta el fenómeno de la visibilidad el blanco para los tiradores. Dependiendo de algunas características de la lluvia como la densidad, habría que determinar cual sería la nitidez para la visualización de los objetos. El objetivo principal de este trabajo es modelar una librería para simular la lluvia en un ambiente virtual en un simulador de tiro. Cumpliendo este objetivo, el simulador de tiro, tendrá una interfaz más completa y de mayor nivel de dificultad, convirtiéndose así, en una poderosa herramienta que permitirá mayor efectividad en el tiro, menos movimiento de personal y ahorros logísticos tanto en las instituciones docentes como en las unidades de combate de nuestro país.

Para representar la lluvia en un ambiente virtual se seleccionaron dos modelos matemáticos. Un modelo lineal que utiliza sistemas de partículas para representar la lluvia, un cubo tridimensional como escenario para lograr la visibilidad de la misma, donde cada partícula presenta una serie de atributos (color, transparencia, velocidad, etc.). También se utiliza un ángulo para los efectos de gravedad para hacer la lluvia más realista y otro para determinar el sentido de orientación en el plano, además un vector de traslación para los movimientos de las partículas.

El modelo del Doble Cono modela la lluvia con 4 texturas, las cuales cubren el cono centrado respecto a la cámara, enlosando cada textura para cubrir el cono. Se puede apreciar la utilización de técnicas como el **parallax** para mezclar dichas texturas y darle una sensación de profundidad. *Motion-blur* es otra técnica que proporciona el efecto de borrado difuso debido al movimiento rápido de los objetos en las escenas.

Palabras Claves

Lluvia, simulador de tiro, simulación, modelos matemáticos, librería, ambiente virtual.

Summary

The University of Informatics Sciences together with the Revolutionary Armed Forces (SIMPRO) develops a shooting simulator, which will allow virtual practices in different views and contexts. One of the problems that the simulator presents is how to represent rain in each one of these views and how the gunners' target is affected by the phenomenon of visibility. Depending on some characteristics of rain like density, it would be necessary to determine which it would be the objects' clearness for visualization. The main objective of this work is to design a bookstore to simulate rain in a virtual environment in a shooting simulator. Fulfilling this objective, the shooting simulator will have an interface much more complete and a higher level of difficulty, becoming a powerful tool that will allow a greater effectiveness when shooting, less logistic movement of personnel and logistic savings not only in the educational institutions but in the battle units of our country as well.

In order to represent rain in a virtual atmosphere two mathematical models were selected. A lineal model that uses systems of particles to represent rain, a three-dimensional cube as a scenario to obtain the visibility of it, where each particle presents a series of attributes (color, transparency, speed, etc.). An angle for the gravity effects is also used to make rain more realistic and another angle is used to determine the sense of direction in the plane. Besides, a translation vector for the particles movements is used too.

The model of the double cone, models rain through the displacement of 4 textures that cover it, which is centered regarding the camera that is located in its inner part. It is interesting to see how some techniques to give the sensation of depth and parallax are used in this model, besides the use of the diffuse deletion or motion blur, due to the fast movement of the objects in the scenes.

Key Words

Rain, shooting simulator, simulation, mathematical models, library, virtual environment.

| | |
|--|----|
| Índice | |
| Introducción | 5 |
| Capítulo 1: Fundamentación Teórica | 8 |
| Introducción | 8 |
| 1.1 Simulación | 8 |
| 1.2 Necesidad de la simulación..... | 10 |
| 1.3 Ventajas y Desventajas de la Simulación..... | 11 |
| 1.3.1 Ventajas | 11 |
| 1.3.2 Desventajas | 12 |
| 1.4 Simulador..... | 13 |
| 1.4.1 Simulador a nivel mundial..... | 13 |
| 1.4.2 Simuladores a nivel nacional..... | 13 |
| 1.4.3 Tipos de simuladores | 14 |
| 1.5 Simuladores SIMPRO | 16 |
| 1.5.1 Ventajas | 18 |
| 1.5.2 Desventajas | 18 |
| 1.6 Direc3d..... | 19 |
| 1.7 C++ | 20 |
| Conclusiones del capítulo | 21 |
| CAPÍTULO 2: Sistemas de partículas y comportamiento de la lluvia en ambientes virtuales. | 22 |
| Introducción | 22 |
| 2.1 Partícula..... | 22 |
| 2.2 Sistemas de partículas | 22 |
| 2.3 Lluvia | 24 |
| 2.4 Ambiente Virtual..... | 26 |
| 2.5 Librería..... | 28 |
| 2.5.1 Características de la librería..... | 29 |
| 2.5.2 Aportes de la librería al simulador de tiro..... | 29 |
| Conclusiones del capítulo | 30 |
| CAPÍTULO 3: Propuesta de los Modelos Matemáticos para la confección de un librería de lluvia. | 31 |
| 3.1 Modelo Lineal..... | 31 |

| | |
|---|----|
| 3.2 Modelo del Doble Cono | 34 |
| 3.3 Características de los modelos matemáticos..... | 40 |
| 3.4 Modelo más efectivo. | 42 |
| 3.5 Demo experimental | 42 |
| 3.4.1 Diagrama de clases del Demo..... | 43 |
| 3.4.2 Descripción de las clases | 44 |
| Conclusiones del capítulo | 47 |
| Conclusiones Generales | 48 |
| Recomendaciones | 49 |
| Referencia Bibliográfica | 50 |
| Bibliografía..... | 51 |
| Anexos..... | 53 |
| Glosario de Términos..... | 56 |

Índice de Figuras

| | |
|---|----|
| Fig. 1 Simulador de Carreras..... | 15 |
| Fig. 2 Simulador de Vuelo..... | 15 |
| Fig. 3 Simulador de Trenes..... | 16 |
| Fig. 4 Sims..... | 16 |
| Fig. 5 Simulador de Conducción..... | 17 |
| Fig. 6 Simulador de Tiro..... | 19 |
| Fig. 7 Ambiente Virtual..... | 27 |
| Fig. 8 Ambiente Virtual..... | 27 |
| Fig. 9 Cubo Tridimensional..... | 32 |
| Fig. 10 Eje de coordenadas..... | 32 |
| Fig. 11 Desplazamiento de la partícula..... | 34 |
| Fig. 12 Textura de la lluvia..... | 35 |
| Fig. 13 Doble Cono..... | 37 |
| Fig. 14 Representación de la lluvia sin mucha niebla..... | 46 |
| Fig. 15 Representación de la lluvia con niebla..... | 47 |

Índice de Ecuaciones

| | |
|--|----|
| Ec. 1 Vector traslación..... | 33 |
| Ec. 2 Transformación del vector | 33 |
| Ec. 3 Factor de desplazamiento | 36 |
| Ec. 4 Factor de elongación o elásticos | 36 |
| Ec. 5 Vector de trayectoria de la lluvia..... | 38 |
| Ec. 6 Matriz de Transformación de la textura | 38 |

Índice de Tablas

| | |
|---|----|
| Tabla 1 Descripción de la clase partícula | 44 |
| Tabla 2 Descripción de la clase posición | 44 |
| Tabla 3 Descripción de la clase lluvia..... | 45 |

Introducción

La Universidad de las Ciencias Informáticas (UCI) actualmente esta formada por varias facultades, cada una de ellas vinculada a un perfil de la especialidad. Este trabajo se desarrolla en la facultad 5, que se dedica fundamentalmente al desarrollo de aplicaciones de Realidad Virtual, en conjunto a la empresa (SIMPRO) de las Fuerzas Armadas Revolucionarias, específicamente en la creación de simuladores.

Unos de los problemas que presentan estos simuladores y en especial el Simulador de Tiro, es ¿cómo representar la lluvia en diferentes vistas?, por ejemplo: dentro de un vehiculo en movimiento, al aire libre, etc. Otra problemática presente es ¿como afecta este fenómeno la visibilidad de los tiradores?, dependiendo fundamentalmente de la densidad de la lluvia se atenuaría la nitidez en la visualización de los objetos

La solución es la confección de una librería donde se puedan modelar la lluvia en un ambiente virtual en diferentes situaciones específicas, mediante el desarrollo dos modelos matemáticos, seleccionando el más efectivo, posibilitando prácticas de tiros virtuales en los diferentes contextos antes mencionados.

El objeto de estudio radica en la modelación de la lluvia en ambiente virtual

El campo de acción es el desarrollo de los modelos matemáticos que representen la lluvia en ambiente virtual en situaciones específicas.

El objetivo principal es modelar una librería para la simulación de la lluvia en un ambiente virtual en un simulador de tiro.

Mediante un estudio de modelos matemáticos para la simulación de la lluvia en ambientes virtuales, existentes para juegos, y otros simuladores, se podrá elegir un modelo adecuado para una primera versión del Simulador de Tiro, este modelo obtendrá su validación mediante su implementación en una librería y/o un demo que demuestre su funcionamiento es la **idea a defender** en este trabajo.

Los objetivos específicos de este trabajo están dados por:

- 1- Caracterizar los modelos matemáticos que represente la lluvia en ambientes virtuales.

2- Elegir los modelos más adecuados para el simulador de tiro teniendo en cuenta 1era y 2da versiones del mismo.

3- Confeccionar un demo para demostrar el funcionamiento de la librería de la lluvia con el modelo escogido para una versión inicial del simulador de tiro.

Las tareas propuestas para solucionar estos objetivos son:

1. Estudiar sobre los modelos matemáticos que representan la lluvia.
2. Seleccionar los modelos matemáticos más adecuados para la representación de la lluvia en el simulador de tiro.
3. Modelar la lluvia, utilizando el modelo seleccionado, en un ambiente virtual (Demo).

Para la realización y cumplimiento del objetivo y las tareas propuestas se utilizó el siguiente método teórico:

Analítico sintético: Son dos procesos inherentes al pensamiento, operaciones lógicas importantes; que nos permiten; como métodos teóricos, buscar la esencia de los fenómenos, los rasgos que lo caracterizan y los distinguen.

Su objetivo en una investigación es analizar las teorías, documentos, etc; permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.

El trabajo esta estructurado de la siguiente manera:

Capítulo 1: En este capítulo se aborda sobre los conceptos fundamentales que acercan al lector al mundo de la simulación. La importancia que representa viene dada por el conocimiento general que se adquiere acerca del tema antes mencionado para luego entender el desarrollo del trabajo.

Capítulo 2: En este capítulo se tocan aspectos más específicos que están estrechamente relacionados con la librería y los modelos matemáticos que se propondrán para la realización de la misma, donde se fundamenta la investigación relacionada con los sistemas de partículas, sus atributos y comportamiento de la partícula para la modelación de los modelos antes mencionados.

Capítulo 3: En el cuerpo de este capítulo se encuentra la propuesta de los modelos matemáticos que pueden conformar una librería inicialmente. En su desarrollo se expone una explicación detallada de cómo representan la lluvia en un ambiente virtual para el simulador de tiro, así como la caracterización de los modelo matemático que pueden conformar la librería de lluvia.

Capítulo 1: Fundamentación Teórica

Introducción

En este capítulo se tratarán aspectos importantes sobre el desarrollo de la Simulación, haciendo referencia a conceptos e investigaciones desarrolladas sobre el tema, además de realizar un estudio de las ventajas y desventajas que sitúa más al lector en el tema.

El país cuenta con un proyecto de investigación (SIMPRO), que se encarga de la simulación en la esfera militar, en el se desarrollan una serie de simuladores para perfeccionar las técnicas de combate en las Fuerzas Armadas. En la Universidad de la Ciencias Informáticas, existe una parte de este proyecto que se encarga del desarrollo de un simulador de tiro. Para lograr este propósito es necesario añadirle condiciones adversas del tiempo, como la lluvia, para darle mayor complejidad a los ejercicios prácticos, de aquí la necesidad de conocer al menos algunos aspectos fundamentales que expliquen que es la Simulación.

1.1 Simulación

La simulación es una técnica muy poderosa y ampliamente usada en las ciencias para analizar y estudiar sistemas complejos. En las investigaciones se formulan modelos que se resuelven en forma analítica. En casi todos estos modelos la meta es determinar soluciones óptimas. Cuando se intenta utilizar modelos analíticos para sistemas como éstos, en general necesitan de tantas hipótesis de simplificación que es probable que las soluciones no sean buenas, o bien, sean inadecuadas para su realización. En estos casos, con frecuencia la única opción de modelado y análisis de que dispone quien toma decisiones es la simulación. *Simular, es reproducir artificialmente un fenómeno o las relaciones entrada-salida de un sistema.* Esto ocurre siempre y cuando la operación de un sistema o la experimentación en él son imposibles, costosas, peligrosas o poco prácticas, como en el entrenamiento de personal de operación, pilotos de aviones, etc.

Si esta reproducción está basada en la ejecución de un programa en una computadora digital, entonces la simulación se llama digital y usualmente se conoce como simulación por computadora, aunque esto

incluye la simulación en las computadoras analógicas. A partir del advenimiento de las computadoras electrónicas, la simulación ha sido una de las herramientas más importantes y útiles para analizar el diseño y operación de complejos procesos o sistemas. Simular, según el Diccionario Universitario Webster, es “fingir, llegar a la esencia de algo, prescindiendo de la realidad”.

Se puede definir a la simulación como la técnica que imita el funcionamiento de un sistema del mundo real cuando evoluciona en el tiempo. En síntesis, cada modelo o representación de una cosa es una forma de simulación. La simulación es un tema muy amplio y mal definido que es muy importante para los responsables del diseño de sistemas, así como para los responsables de su operación.

A continuación se mencionan algunas de las definiciones formales de simulación más aceptadas:

"Simulación es una técnica numérica para conducir experimentos en una computadora digital. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos del mundo real a través de largos periodos de tiempo". (Naylor 1971)

"La simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con el mismo con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias, dentro de los límites impuestos por un cierto criterio o un conjunto de ellos, para el funcionamiento del sistema". (Shannon 1988)

Después de interiorizar estos dos conceptos, llegamos a la conclusión de que la simulación es la creación de modelos para representar una situación real utilizando técnicas numéricas para conducir experimentos en una computadora digital. Es necesario resolverlos mediante una programación que debe apoyar y comprender el funcionamiento de ciertos tipos de relaciones matemáticas y lógicas que conllevan a reproducir sensaciones que no están sucediendo.

Por lo que se entiende que el proceso de simulación incluye tanto la construcción del modelo como su uso analítico para estudiar un problema. Un modelo de simulación comúnmente toma la forma de un conjunto de hipótesis acerca del funcionamiento del sistema, expresado con relaciones matemáticas o lógicas entre los objetos de interés del sistema. En contraste con las soluciones matemáticas exactas disponibles en la mayoría de los modelos analíticos, el proceso de simulación incluye la ejecución del

modelo a través del tiempo, en general en una computadora, para generar nuestras representativas de las mediciones del desempeño o funcionamiento. En este aspecto, se puede considerar a la simulación como un experimento de muestreo acerca del sistema real, cuyos resultados son puntos de muestra. Por ejemplo, para obtener la mejor estimación del promedio de la medición del funcionamiento, calculamos el promedio de los resultados de muestra. Es claro que tanto más puntos de muestra generemos, mejor será nuestra estimación.

A través de la simulación se pueden estudiar los efectos de ciertos cambios informativos, de organización y ambientales, en la operación de un sistema, al hacer alteraciones en su modelo y observar los efectos de éstas en el comportamiento del problema. La observación detallada del sistema que se está simulando, conduce a un mejor entendimiento del mismo y proporciona sugerencias para mejorarlo. La simulación de sistemas complejos puede producir un valioso y profundo conocimiento acerca de cuales variables son más importantes que otras en el sistema y cómo ellas obran entre sí.

La simulación puede emplearse para experimentar con situaciones nuevas acerca de las cuales tenemos poca o ninguna información, con el objetivo de estar preparados para alguna eventualidad. Puede servir como una prueba de pre servicio para ensayar nuevas políticas y reglas de decisión en la operación de un sistema, antes de tomar el riesgo de experimentar con el sistema real. Son valiosas algunas veces, ya que proporcionan una forma conveniente de dividir un sistema complicado en subsistemas, cualesquiera de los cuales puede ser modelado por un analista o un equipo de expertos en esta área.

Permite estudiar los sistemas dinámicos, ya sea en tiempo real, tiempo comprimido o tiempo expandido. Cuando se presentan nuevos componentes de un sistema, puede emplearse para ayudar a descubrir los obstáculos y otros problemas que resultan de la operación del sistema.

1.2 Necesidad de la simulación

La humanidad ha tendido a realizar simulaciones desde la antigüedad, se simulaban armas para la preparación de los combatientes de futuras guerras, se simulaban batallas con maquetas, las escuelas de medicina utilizaban cadáveres para que los estudiantes aprendieran las técnicas de cirugía. Actualmente, existen muchos simuladores físicos como los de vuelo que se utilizan en las escuelas de piloto y de conducción. Está demostrado que el uso eficiente de simuladores genera, en determinados casos, un nivel de aprendizaje superior que si se compara con la mera explicación teórica. El principio racional para usar la

simulación en cualquier área del conocimiento es la búsqueda del hombre por adquirir conocimientos referentes a la predicción del futuro, o la explicación lógica de un fenómeno. Esta búsqueda es tan antigua como la historia de la humanidad.

Una corrida de un programa de simulación le dice al usuario cual será el comportamiento dinámico de su empresa o de la maquina que esta diseñando. Así podemos ver los pronósticos para la demanda y utilidad de nuestro producto, o ver cuando un mecanismo pueda fallar en las condiciones adversas del ambiente donde funcionará.

1.3 Ventajas y Desventajas de la Simulación.

1.3.1 Ventajas

La simulación se aplica a la educación y el entrenamiento. Aunque la técnica de simulación generalmente se ve como un método de último recurso, recientes avances en la metodología de simulación y la gran disponibilidad de software que actualmente existe en el mercado, han hecho que la técnica de simulación sea una de las herramientas más usadas en el análisis de sistemas. Se puede aplicar la simulación cuando no existe una formulación matemática del problema. Se aplica cuando los procedimientos matemáticos son muy complejos, cuando no existe un personal necesario para que resuelva el problema. También cuando se desea hacer experimentos por un cierto periodo de tiempo para observar el comportamiento del sistema y se requiera que el proceso sea en menos tiempo. Además de las razones antes mencionadas, se sugiere que un estudio de simulación es muy recomendable, pues presenta las siguientes ventajas:

- La técnica de simulación puede ser utilizada como un instrumento pedagógico para enseñar a estudiantes habilidades básicas en análisis estadísticos, análisis teórico, etc.
- Una observación detallada del sistema que se está simulando puede conducir a un mejor entendimiento del sistema y por consiguiente a sugerir estrategias que mejoren la operación y eficiencia del sistema.

- A través de un estudio de simulación, se puede estudiar el efecto de cambios internos y externos del sistema, al hacer alteraciones en el modelo del sistema y observando los efectos de esas alteraciones en el comportamiento del sistema.

1.3.2 Desventajas

Un buen modelo requiere mucho tiempo y es bastante costoso. Puede reflejar con precisión una situación del mundo real, cuando en verdad no lo hace. La simulación es imprecisa, y no se puede medir el grado de error. Generalmente los resultados son numéricos y pueden ser imprecisos por algunas cifras decimales que con el tiempo se convierten en unas cifras muy significativas. La técnica de simulación presenta otras importantes desventajas:

Falla al producir resultados exactos: Se supone que un sistema está compuesto de uno o más elementos que están sujetos a un comportamiento al azar. Cuando una simulación se desarrolla con un modelo del sistema, los valores de cada variable son registrados y los promedios de estos valores son dados en una post-simulación. Pero el promedio en una muestra de observación, solo a veces provee un estimado de lo esperado, es decir, una simulación solo provee estimados, no resultados exactos.

Largo tiempo de conducción: Un estudio de simulación no puede ser conducido o llevado a cabo en solo un fin de semana. Se puede requerir de meses de esfuerzo para reunir información, construir, verificar y validar modelos, diseñar experimentos y evaluar e interpretar los resultados.

Costos para proveer capacidad de simulación: El establecimiento y mantenimiento de capacidad de simulación envuelve tener mejor personal, software, hardware, entrenamiento y otro tipo de costos.

Abuso de simulación: Hay muchas facetas para un balanceado y comprensivo estudio de la simulación. Ya que una persona debe tener conocimiento de una gran variedad de áreas antes de llegar a ser un practicante de la simulación. Este hecho es algunas veces ignorado, sin embargo como resultado, cada estudio puede ser desarrollado incorrectamente, podría estar incompleto, o podría caer en otro tipo de caminos, quizás resultado de una falla del esfuerzo de la simulación.

1.4 Simulador

Gracias al surgimiento de la poderosa técnica de simulación nacen los simuladores, podemos afirmar que *un simulador es un aparato que permite la simulación de un sistema, reproduciendo su comportamiento y sensaciones que en realidad no están sucediendo.*

Un simulador pretende reproducir tanto las sensaciones físicas (velocidad, aceleración, percepción del entorno) como el comportamiento de los equipos de la máquina que se pretende simular.

Un simulador puede ser un dispositivo o aparato que simula un fenómeno, el funcionamiento real de un aparato, dispositivo o las condiciones de entorno a la que está sometidos una máquina, posibilitan un aprendizaje significativo por descubrimiento y la investigación, según el simulador, puede realizarse en tiempo real o en tiempo acelerado.

1.4.1 Simulador a nivel mundial

La tecnología ha avanzado grandemente y con ella todo lo que se desenvuelve a su alrededor, dentro de ella tenemos a los simuladores, los cuales están a un nivel muy alto. En la gran mayoría de las ramas de la sociedad se utilizan simuladores, su puesta en práctica ahorra gran cantidad de tiempo y son capaces de realizar operaciones que en la vida real serían muy costosas tanto en lo económico como en calidad humana.

Han ganado una gran importancia en los sistemas de evaluación general y se usan para apoyar los procesos de decisión en muchas organizaciones. Los simuladores están en uso alrededor del mundo, los cuales han entrenado y calificado miles de personas. Hoy todas las personas de cualquier edad entienden su funcionamiento y manejo. Los simuladores que más se conocen en el mundo se pueden encontrar en los Video Juegos, ellos son los que revolucionan el mundo de la simulación por su gran demanda, son capaces de darle realismo a sus escenas.

1.4.2 Simuladores a nivel nacional

Nuestro país se esfuerza por mantener dentro del mundo de la tecnología de punta, ya que se cuenta con investigaciones de alto nivel en esta rama. Se puede afirmar que ya se encuentran simuladores cubanos en escuelas del país y en empresas nacionales e internacionales, por ejemplo en las Fuerzas Armadas Revolucionarias (FAR) existen múltiples simuladores de vuelo, de tanque, tiro entre otros. Ellos tienen gran importancia pues ahorran tiempo, recursos económicos y desastres o accidentes que pueden afectar la vida de los soldados y civiles. Hay que tener en cuenta que las prácticas militares pueden ser peligrosas.

Hay que destacar que con la ayuda de los simuladores nuestro país será capaz de extender nuevas directivas hacia el desarrollo y hacia la evolución de la informática en empresas, escuelas y otros centros laborales, capacitando a los habitantes cubanos a un futuro impostergable que se avecina a nuestras puertas.

1.4.3 Tipos de simuladores

Hoy en día hay varias categorías de simuladores que van desde simples sistemas hasta un fuerte entrenamiento, las cuales realizan numerosas actividades que pueden ser estudiadas e investigadas, proporcionando una serie de conocimientos acerca del tema del cual se realiza la actividad.

En la actualidad el desarrollo de la tecnología ha propiciado que la informática unida a otras ciencias, sea capaz de simular todo tipos de comportamientos en la vida del hombre. Ejemplos de simuladores

- **Simulador de carreras:** es el tipo de simulador más popular; se puede conducir un automóvil, motocicleta, camión, etc. Calculan por ejemplo el recorrido físico de la suspensión, el trabajo del motor y la fricción de los neumáticos con la pista.. Ejemplos: *rFactor*, *GTR*, *GT Legends*.



Fig. 1 Simulador de Carreras

- **Simulador de vuelo o de aviones:** permite dominar el mundo de la aviación y pilotar aviones, helicópteros etc. Es un sistema que intenta replicar, o simular, la experiencia de volar una aeronave de la forma más precisa y realista posible. Ejemplo: *Microsoft Flight Simulator*.



Fig. 2 Simulador de Vuelo

- **Simulador de trenes:** permite controlar un tren. Ejemplo: *Trainz*.



Fig. 3 Simulador de Trenes

- **Simulador de vida o de dinámica familiar:** permite controlar una persona y su vida. Ejemplo: *Los Sims*.



Fig. 4 Sims

1.5 Simuladores SIMPRO

SIMPRO es una empresa que se dedica a la investigación y desarrollo de simuladores. Sus trabajos han sido reconocidos en diferentes lugares, tanto nacionales como internacionales. En la actualidad se trabaja en conjunto con la Universidad de las Ciencias Informáticas (UCI) teniendo como principal logro el

desarrollo de un simulador de autos. El nuevo proyecto en el que se está trabajando es un simulador de tiro. Este simulador tendrá incluidas todas las condiciones ambientales necesarias como: lluvia, nubes, etc. Estas condiciones propiciarán que el simulador tenga un escenario más real, su propósito es lograr que el soldado tenga prácticas de tiros en un ambiente casi real.

SIMPRO trabaja en varias líneas de Simuladores:

- Simulador de conducción de autos ligeros
- Simulador de conducción de autos pesados (camiones)
- Simulador de conducción y operación de equipos de construcción (grúas, buldózer, palas, excavadoras...)
- Simulador de operación de equipos de perforación de pozos de petróleo
- Simulador de operación de equipos de carga (montacargas, grúas portuarias)
- Simulador de conducción de remolcadores marítimos.
- Otros simuladores similares a los anteriores.



Fig. 5 Simulador de Conducción

Un simulador de tiro es un sistema computarizado de entrenamiento militar que se utiliza para representar diferentes situaciones tácticas – militares, con el objetivo de mejorar el entrenamiento de las tropas realizando prácticas de tiro en ambientes virtuales. Su finalidad principal es ejercitar al soldado, cadete u oficial con diferentes armamentos en las condiciones reglamentadas para lograr alcanzar un alto grado de destreza.

Permite un mayor rendimiento en cada tirador elevando sus estadísticas, logra la efectividad en la puntería con el ejercicio diario, además de una visión muy similar de las condiciones reales que se pueden encontrar en el terreno.

El sistema para lograr las expectativas del tirador debe ser capaz de cumplir con varios principios como son el realismo, la progresión para que no sea muy notable el cambio al enfrentarse con la práctica real y la que permite diferentes condiciones cambiantes a la hora de efectuar el tiro. El simulador de tiro ofrece una serie de ventajas y desventajas para las tropas.

1.5.1 Ventajas

- Logra tener un seguimiento constante de las estadísticas en el tirador logrando un mayor perfeccionamiento en el tiro.
- No hay problemas de descalibración por transportación a un campo de tiro.
- No se desgasta el armamento.
- Siempre se sabe donde se proyecta la munición.
- El entrenamiento de las tropas es más rápido y efectivo.
- No se pone en peligro las vidas humanas.
- No ocurren accidentes ni fatales ni de otra índole.
- Imita el uso de las condiciones meteorológicas.
- Protegen el medio ambiente, la flora y la fauna.

1.5.2 Desventajas

- No es capaz de reemplazar el tiro con municiones de guerra por muy efectivo y sofisticado que sea el sistema.

- No reemplaza la realidad.
- El costo de los simuladores crece respecto a la calidad de los mismos.
- Las condiciones ambientales no son perceptibles por el tirador como en el tiro en el terreno.



Fig. 6 Simulador de Tiro

1.6 Direc3d

El Direct3d forma parte de la familia DirectX, es una API (Interfaz de Programación de Aplicaciones) patentada por Microsoft la cual está disponible en Windows de 32bit y 64bit para la programación de gráficos 3D. Es usada fundamentalmente en videojuegos aprovechando el hardware de aceleración gráfica disponible en tarjeta gráfica, consigue un mejor resultado en la presentación de los gráficos por pantalla. Está compuesto por dos APIs el modo retenido y el modo inmediato. El modo inmediato brinda soporte a todas las primitivas de procesamiento 3D que permiten las tarjetas gráficas (*luces, materiales, transformaciones, control de profundidad*). El modo retenido, construido sobre el anterior, presenta una abstracción de nivel superior ofreciendo funcionalidades preconstruidas de gráficos como jerarquías o animaciones.

1.7 C++

C++ se trata de un lenguaje de programación basado en el lenguaje C, estandarizado (ISO/IEC 14882:1998), ampliamente difundido, y con una biblioteca estándar C++ que lo ha convertido en un lenguaje universal, de propósito general, y ampliamente utilizado tanto en el ámbito profesional como en el educativo.

El **C++** es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C.

Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores más modernos

Las principales características del C++ son el soporte para programación orientada a objetos y el soporte de plantillas o programación genérica (*templates*). Se puede decir que C++ es un lenguaje que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos.

Además posee una serie de propiedades difíciles de encontrar en otros lenguajes de alto nivel:

- Posibilidad de redefinir los operadores (sobrecarga de operadores)
- Identificación de tipos en tiempo de ejecución (*RTTI*)

C++ está considerado por muchos como el lenguaje más potente, debido a que permite trabajar tanto a alto como a bajo nivel, sin embargo es a su vez uno de los que menos automatismos trae (obliga a hacerlo casi todo manualmente al igual que C) lo que "dificulta" mucho su aprendizaje.

El nombre **C++** fue propuesto por Rick Mascitti en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera de un laboratorio científico. Antes se había usado el nombre "C con clases". En C++, "C++" significa "incremento de C" y se refiere a que C++ es una extensión de C.

Conclusiones del capítulo

Se puede comentar que en este capítulo se brindaron ideas necesarias para percibir todo lo relacionado con el mundo de la simulación, específicamente del simulador de tiro. En gran medida se demostró las ventajas y desventajas de los simuladores en la actualidad, las cuales ayudan a entender el objetivo de nuestra investigación.

CAPÍTULO 2: Sistemas de partículas y comportamiento de la lluvia en ambientes virtuales.

Introducción

En el desarrollo del capítulo se brinda la información necesaria para la comprensión del modelado de la lluvia usando sistemas de partículas, además el lector puede conocer algunas características de la lluvia, y el vínculo que tiene con un simulador de tiro, así como los aportes que esta librería proporcionaría al simulador y a su programador. También se abordara lo relacionado a la necesidad de simular un entorno real en diferentes situaciones de la vida diaria, así como los ambientes que se desarrollan dentro de los simuladores y la relación con nuestra institución.

2.1 Partícula

Una partícula es la parte más pequeña indivisible que puede encontrarse libre en la naturaleza. El término partícula engloba desde los constituyentes elementales de los átomos, es decir, electrones, protones y neutrones.

Es un cuerpo uniforme, que en la realidad no existe y que corresponde a la idealización matemática de un objeto cuyas dimensiones y orientación en el espacio son despreciables para la descripción particular del movimiento.

Las formas de partículas pueden ser esferas, elipsoides, recuadros pequeños u otras. La mayoría de ellas no pueden ser vistas. Las partículas pueden existir en cualquier forma, tamaño y pueden ser partículas sólidas o gotas líquidas.

2.2 Sistemas de partículas

El término de sistemas de partículas no está rigurosamente definido en la computación gráfica (Reeves 1983). Se ha usado para describir técnicas de modelado, técnicas de *renderización* y hasta algunos tipos de animación. La definición de sistemas de partículas depende de la aplicación donde se esté usando. No obstante si se han identificado características comunes en todos los sistemas de partículas:

- **Colección de Partículas:** Un sistema de partículas está formado por una o muchas partículas individuales. Cada una de esas partículas posee atributos que están directa o indirectamente relacionados con su comportamiento individual y en algunos casos como el cómo y cuándo ocurre su *rendering*. Usualmente las partículas son primitivas gráficas como puntos o líneas aunque normalmente no se limitan a esto.

- **Atributos definidos estocásticamente:** Otra característica común entre los sistemas de partículas es la introducción de elementos aleatorios (Reeves 1985). Estos elementos se pueden usar para controlar algunos atributos de las partículas como: posición velocidad y color, entre otros.

- **Atributos de las partículas.**

En (Reeves 1983) se define un objeto compuesto de partículas como *fuzzy*. También se define que cada partícula en un *sistema de partículas* debe tener los siguientes atributos.

- Posición Inicial
- Velocidad Inicial (velocidad y dirección)
- Tamaño inicial
- Color Inicial
- Transparencia Inicial
- Forma
- Tiempo de vida.

- **Ciclo de vida de una partícula.**

Cada partícula atraviesa distintas fases dentro de un sistema: generación, dinámica y muerte. A continuación se describen en más detalle:

- **Generación:** Las partículas en el sistema se generan de forma aleatoria dentro de determinada localización del objeto *fuzzy*. A este espacio se le denomina marco de generación del objeto *fuzzy*, dicho marco puede cambiar en el tiempo. Se le asigna a cada atributo de la partícula un valor inicial. Dicho valor puede ser fijo o puede estar determinado por un proceso estocástico.

- **Dinámica de la partícula:** Los atributos de cada una de las partículas pueden variar en el transcurso del tiempo. Por ejemplo, el color de una partícula en una explosión se hace más oscuro en la medida que la partícula se aleja del centro de origen. En general cada uno de los atributos de la partícula se puede representar por una ecuación paramétrica que dependa entre otras cosas del tiempo de vida de dicha partícula. Por ejemplo la posición de la partícula es una función que depende tanto del tiempo como de la velocidad.

- **Extinción o muerte:** Cada partícula tiene dos atributos muy importantes, la edad y el tiempo de vida. La edad es el tiempo en que la partícula ha estado viva, medido en *frames*, este valor siempre se inicializa en cero cuando se crea la partícula. El tiempo de vida es el máximo tiempo de existencia que tiene cada partícula. Cuando una partícula llega a que su edad alcanza a su tiempo de vida entonces la partícula se destruye. Existen algunos criterios adicionales que determinan cuando una partícula puede interrumpir su ciclo de vida aunque no haya llegado a su edad máxima:

- La partícula ha sobrepasado los límites del área de visión y no será *renderizada*, por lo tanto no hay razón para mantenerla con vida.

- La partícula ha goleado un objeto que provoca su destrucción, puede ser por ejemplo cuando la lluvia golpea el suelo u otro objeto.

- Alguno de los atributos alcanza un valor que provoca que la partícula no sea visible.

2.3 Lluvia

Lluvia, precipitación de gotas líquidas de agua. Las gotas de agua tienen en general diámetros superiores a 0,5 mm y pueden llegar a unos 3 mm. Las gotas grandes tienden a achatarse y a dividirse en gotas

menores por la caída rápida a través del aire. La precipitación de gotas menores, llamada llovizna, suele limitar fuertemente la visibilidad, pero no suele producir acumulaciones significativas de agua.

La lluvia se emplea en el simulador de tiro para cambiarle el entorno virtual al soldado, una vez que comienza a caer logra crearle dificultades a la hora de efectuar el tiro. El incremento de este fenómeno es capaz de distorsionar el blanco, medición de distancia, velocidad del blanco, y ubicación del punto de puntería logrando que el ejercicio sea más complejo y que el soldado se pueda acercar más a la realidad. Puede afectar operaciones tanto en un entorno abierto como cerrado, o sea, puede afectar tanto en un ejercicio al aire libre o dentro de cualquier pieza o vehículo de combate.

Su presencia, junto a otros fenómenos naturales completa el escenario de entrenamiento, permite elevar las potencialidades y la eficacia del simulador. El logro de la lluvia parte del hecho de una valoración adecuada de los fenómenos naturales que pueden presentarse en varias escenas del simulador donde puede elevarse el realismo, por la rigurosidad de su tratamiento. Una representación adecuada influye decisivamente en los resultados finales que se obtienen del simulador.

La lluvia es un fenómeno natural atmosférico extremadamente complejo. Consiste en numerosos efectos visuales que obran en conjunto. Crear una representación fiel de la lluvia en ambientes naturales complejos es un problema no trivial. El desafío proviene no solamente de la complejidad, de la diversidad visual de los componentes de la lluvia y de los objetos de la escena, sino también de la cantidad enorme de pequeños detalles que deben modelarse para obtener efectos visuales realistas y simulaciones físicas estimables. Sin embargo el *rendering* de la lluvia resalta enormemente escenas al aire libre. Filmar escenas de lluvia demanda un gran esfuerzo y costo debido a lo complicado de la estructura del escenario.

Esta tarea se hace más retadora cuando se trata de crear un *rendering* foto-realista de la lluvia en medios ricos, productivos a niveles interactivos. Algunos juegos recientes que incorporan lluvia usan enfoques simplistas, incluyendo *rendering stretched*, partículas combinadas para simular gotas de lluvia que caen o empleando texturas animadas combinadas como en (N.WANG and B. 2004) para el render de precipitaciones.

Los efectos de la lluvia se han examinado en el contexto de ciencias atmosféricas (T. and S 1975), (T. and S. 1975), así como en el campo de la visión de computadora (NAYAR, K et al. 2003),(K.GARG and S.

2004). Sin embargo, por el momento sólo algunos avances existen para crear el *rendering* de una lluvia realista que dinámicamente responda a un ambiente iluminado y al movimiento de la cámara.

2.4 Ambiente Virtual

El ambiente virtual es una simulación por computadora capaz de proporcionar información a nuestros sentidos: visión, sonido, tacto y gusto, con el propósito de que el usuario se sienta inmerso y atraído en el mundo que reacciona ante las acciones.

A diferencia de una película tridimensional donde la información se integra en una secuencia de imágenes definidas, en la que el usuario no puede intervenir. Un ambiente virtual es naturalmente tridimensional, dinámico y cambiante, según los movimientos y peticiones del usuario, quien puede explorar y experimentar de acuerdo a las situaciones generadas en el mundo virtual y la retroalimentación que este le proporciona.

Estos Ambientes Virtuales innovadores, como medio electrónico, permiten superar las barreras de espacio y tiempo, promueven la interacción de nuestros sentidos, creando un espacio con escenas muy parecidas a la realidad.

Las imágenes que son mostradas a continuación representan un ambiente virtual sencillo de un terreno de tiro.



Fig. 7 Ambiente Virtual

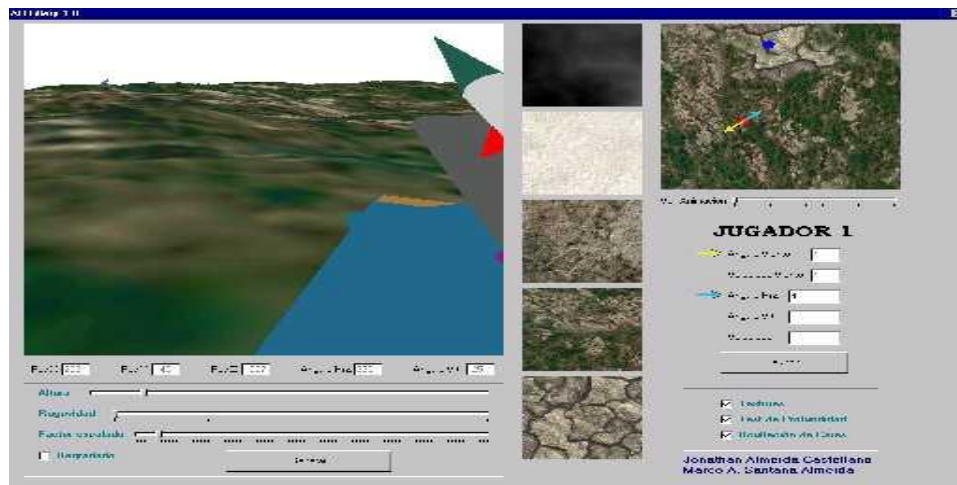


Fig. 8 Ambiente Virtual

2.5 Librería

En la informática una biblioteca es un conjunto de procedimientos y funciones agrupadas en un archivo, con la finalidad de ser aprovechadas y utilizadas por otros programas. Una librería se dice que son trozos de códigos que contienen una funcionalidad pre construida que puede ser utilizada por un ejecutable, los mismos suelen ser unidos por enlaces. En su interior presentan o están compuestas por variables y funciones las cuales pueden estar encapsuladas en forma de clases. De forma general el término *librería* se utiliza para hacer referencia a un a un conjunto de módulos de objetos donde todos van a estar agrupados en un solo fichero, el cual puede tener una serie de extensiones (.bpl, .lib, .dll). Estos ficheros te permiten tratar una serie de colecciones de módulos en una sola unidad, es muy conveniente a la hora de trabajar con grandes aplicaciones.

Existen dos tipos de librerías que a pesar de tener el mismo nombre son capaces de utilizar mecanismos distintos en su funcionalidad.

- **Estáticas** (de enlace estático) o librerías objetos.
- **Dinámicas** (de enlace dinámico)

Las librerías estáticas son una colección de ficheros objetos (compilados), son capaces de arreglar las referencias a rutinas en el programa para que apunten a su localización en la biblioteca en el momento de compilación. Los ficheros son agrupados en un solo fichero de extensión **.lib**, **.a**, junto con uno o varios ficheros de cabecera (generalmente **.h**). Con las librerías estáticas que contengan las funciones que se usen más frecuentemente no será necesario escribir el código cada vez.

Las librerías dinámicas, DLL, también conocidas como de enlace dinámico. Son las utilizadas por el Sistema Operativo Windows, el mismo contiene una serie de librerías del tipo DLL, donde se encapsulan el conjunto de objetos que controlan todo el funcionamiento del sistema. La mayoría de los ficheros de disco asociados con el Sistema son de este tipo, y se ha llegado a afirmar que escribir una DLL es escribir una extensión del propio Windows. (Petzold).

2.5.1 Características de la librería.

La librería que modela la lluvia debe satisfacer las necesidades del escenario donde se va a desarrollar el soldado, es importante que en su cuerpo se encuentren implementados métodos que puedan modelar las características reales de la lluvia para darle un mayor toque de realismo a las escenas. Para lo anterior dicho se debe tener en cuenta algunos aspectos como el tamaño de las gotas, la posición de la cámara y proporción adecuada de la caída de la lluvia. Estos suelen ser los elementos más importantes que deben estar presentes en cualquier situación, aunque se puede mejorar en dependencia del nivel de dificultad que se desee.

Debe ser capaz de visualizar la lluvia en diferentes situaciones, tratar de dar una solución lo más óptimas posible. Una mayor eficiencia de sus métodos traería como consecuencia un mejor resultado, un final exitoso.

2.5.2 Aportes de la librería al simulador de tiro.

Partiendo del concepto de librería y conociendo la situación problemática que existe con los simuladores de tiro en la actualidad en Cuba, se puede afirmar que la elaboración de una librería para la simulación de la lluvia en un ambiente virtual en un simulador de tiro, constituye un gran contribución a el trabajo que se ha estado desarrollando en los simuladores. Su realización es de gran importancia ya que acarrea los siguientes aportes:

- Permite que se pueda integrar la lluvia a cualquier simulador independientemente de su motor gráfico.
- Brinda la posibilidad de mejorar el modelamiento de la lluvia, sin afectar otro modulo de implementación del simulador.
- Ofrece al programador un trabajo más fácil y cómodo.

Conclusiones del capítulo

Al finalizar este capítulo los conocimientos adquiridos nos permitirán un mejor desenvolvimiento a la hora de entender cada uno de los modelos matemáticos que se desarrollan en el tercer capítulo. De los principales temas desarrollados queremos hacer énfasis en sistemas de partículas y lluvia, los cuales tienen una estrecha relación con el modelado de la lluvia. Se logra comprender algunos temas generales concernientes a un ambiente virtual y una librería dentro del marco investigativo.

CAPÍTULO 3: Propuesta de los Modelos Matemáticos para la confección de un librería de lluvia.

Introducción

Simular efectos medioambientales dentro de un entorno virtual de un simulador de tiro es de gran importancia, ya que proporciona un mayor grado de realismo para todo el personal que practique en él. En el desarrollo del capítulo se proponen dos modelos matemáticos, en los cuales interfieren una serie de factores para representar la lluvia con diferentes técnicas. Se realiza una caracterización de cada uno de los modelos, para la elección de uno de los modelos que pueda ser utilizado en una primera versión del simulador de tiro. El Primer modelo es el más simple y aunque se menciona en la literatura no existe una referencia donde se describa matemáticamente, uno de los aportes de este trabajo es que trata de formalizarlo. El segundo modelo propuesto en (N.WANG and W. B. (2004)), es un poco más complejo y no utiliza sistemas de partículas para tratar de ganar en eficiencia y calidad de la representación.

3.1 Modelo Lineal

Para representar la lluvia como un sistema de partículas mediante un modelo matemático lineal, es necesario tener conocimiento del comportamiento del fenómeno y de las características de la partícula que va a representar una gota de lluvia.

Las partículas se generan en el sistema mediante métodos aleatorios controlados por un proceso, se determina el número de partículas que entran en el sistema durante cada intervalo de tiempo. La cantidad de partículas esta dado por una variable que se inicializa al comenzar la simulación es decir al comenzar la lluvia, el diseñador escoge un valor (podrá cambiar el numero inicial de partículas). Por cada partícula el sistema determina una posición inicial, velocidad, duración de la partícula, ángulo de caída, tiempo de vida, tamaño, transparencia y color inicial.

La posición inicial de la partícula esta dada por un plano que se define por la dirección de la partícula, existe un ángulo de caída para la lluvia que refleja los efectos de gravedad para hacerla más realista y otro para el sentido de orientación desde el plano donde yace el origen de la partícula. Las partículas se mueven dentro de una escena (espacio tridimensional) y cambian sus atributos, color, transparencia y

tamaño, se extingue su tiempo de vida y dan paso a las que nacen. Estas partículas se mueven utilizando una función lineal, que unido a la velocidad inicial describe la trayectoria de la partícula dentro del espacio.

Se toma un cubo tridimensional como escenario para lograr visualizar la caída de la lluvia, ver Fig. 9. Se toma como referencia los ejes de coordenadas X, Y, Z, para las posibles transformaciones que puede sufrir cada partícula (posición, color).

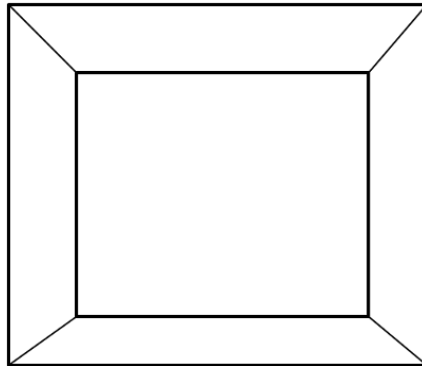


Fig. 9 Cubo Tridimensional

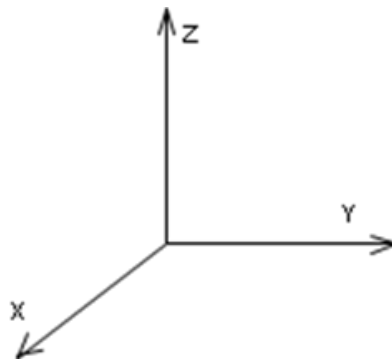


Fig. 10 Eje de coordenadas

Para posicionar las partículas se toma en cuenta la dirección (X, Y, Z) en que se desea que caiga la lluvia y para su desplazamiento y dirección se necesita un vector de traslación:

$$V = aI + bJ + cK$$

Ec. 1 Vector traslación

El proceso de desplazamiento de la partícula dentro del cubo tomando como referencia la posición inicial, es de una forma sencilla, las nuevas coordenadas del punto $P(x, y, z)$ al desplazarse a $P'(x', y', z')$ se calcula utilizando la transformación:

$$Tv : \begin{cases} x' = x + a \\ y' = y + b \\ z' = z + c \end{cases}$$

Ec. 2 Transformación del vector

Donde a, b y c tomarán el valor de I, J, K respectivamente. La Fig.10 demuestra como quedaría el desplazamiento de la partícula dentro del plano.

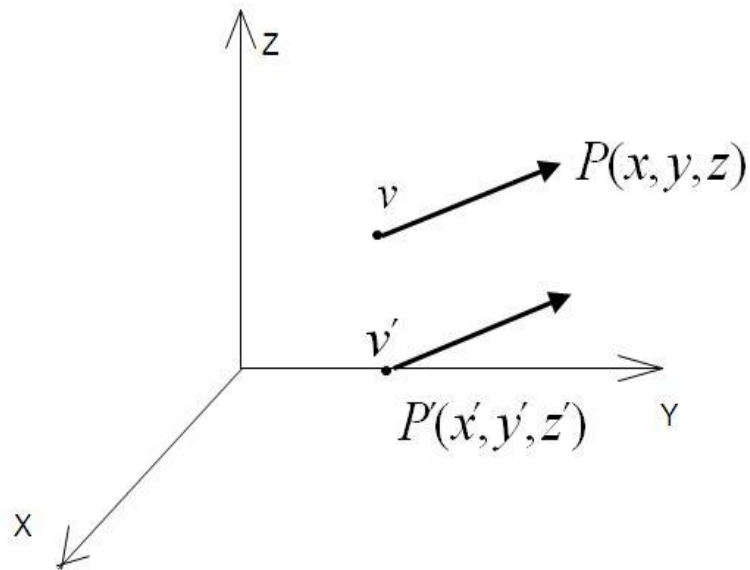


Fig. 11 Desplazamiento de la partícula

3.2 Modelo del Doble Cono

Este modelo se propone en (N.WANG and W. B. (2004)) para la simulación de la lluvia y la nieve en juegos de conducción, aunque es generalizable a los simuladores en general, en este caso a el simulador de tiro en cuestión.

Cuando se crea un juego de conducción, o un simulador de tiro, se puede agregar realismo a la escena con una mirada realista de la lluvia. Muchos juegos han modelado la lluvia usando sistema de partículas (Reeves 1983), que produce un movimiento realista simulando cada una de las gotas, por ejemplo cada gota de lluvia tiene su propia trayectoria, lo que a veces resulta algo costoso, pues en ocasiones a la hora de simular una lluvia muy densa hay que simular muchas más partículas.

La idea básica es usar una textura para simular la lluvia, para lograr una baja sobrecarga en el rendimiento. La textura consiste en un grupo de gotas creadas por un artista o un diseñador, ver la Fig.12.



Fig. 12 Textura de la lluvia

Se necesita aplicar una traslación a la textura de modo que la precipitación aparezca cayendo en un cierto plazo. Se comienza con el acercamiento obvio de traslación de textura con una velocidad proporcional a la velocidad deseada de la precipitación, según el reloj del juego.

Un simple acercamiento produce dos efectos visuales. Primero, cuando la traslación es mucho más grande que el tamaño de una gota individual, el ojo no conecta la gota de lluvia en un *frame* con la misma traslación de la gota de lluvia en el otro *frame*, segundo con la falta de *motion blur* y el mayor espacio de separación, se ve como si las gotas de lluvia estuvieran desconectadas, hacen un revoloteo aleatorio rápido después del paso de la cámara, cuando lo preferible es que una gota de lluvia haga un movimiento continuo. Si la traslación coordinada de la textura es incluso más grande, de modo que sea casi la longitud de la textura entera, desde un *frame* hasta el siguiente, la lluvia puede parecer que se mueve al revés, como el efecto típico de la rueda del carro en viejas películas occidentales. Para ajustarlo se traslada la textura por tramo fijo de la anchura de una gota en cada *frame*. Esto asegura que el ojo conectará el movimiento de cada gota de agua a través de los *frames*.

Nótese que la traslación es constante a pesar de la velocidad y del *framerate*. Dado que se quiere que la precipitación parezca caer en una velocidad constante a pesar del *framerate*, se deben sacar cuentas para la variación del tiempo del *frame*, de otra manera, caerá más rápido con *framerate* más alto y más lento con *framerate* inferior. Con este fin, se desplaza la textura cada *frame* por la siguiente ecuación:

$$D_{precip} = s_{streak} + \frac{t_{delta}}{t_{const}}$$

Ec. 3 Factor de desplazamiento

Ahora, para simular *motion blur*, se usa un factor elástico. Este factor es proporcional a la velocidad del movimiento, tal que las líneas más largas surgen cuando la cámara se mueve rápidamente. Este factor se calcula como se muestra en la ecuación de elongación.

$$E = \frac{s_{streak}}{s_f * (c_{pixel} * s_f * Vec_{precip} + s_{streak})}$$

Ec. 4 Factor de elongación o elásticos

Rendering lluvia con paraxall

Además de proveer el sentido de las partículas en movimiento y del jugador, muchas veces se desea proveer también un sentido de profundidad y *parallax*. Las gotas en la distancia deben verse más pequeñas y su caída debe ser más lenta que las que están más cerca. Para simular esta sensación se aplica una técnica de desplazamiento de textura para múltiples texturas. Esta técnica hace el trabajo con las 4 texturas que utiliza este modelo, pues se reduce la escala de cada textura progresivamente para producir una lluvia más pequeña, se utiliza funciones para mezclar las 4 texturas juntas.

El tamaño de la reducción de la escala depende del tipo e intensidad de la lluvia lo cual es controlado por los artistas o diseñadores.

Aunque esto produce el efecto de *parallax*, la información de la Z es incorrecta, pues las texturas son diseñadas para mirar lejos de la cámara, pero son realmente *rendered* en un plano a corta distancia. Cuando las caídas son lo suficientemente pequeñas, éste no es un efecto notable.

Simulando el movimiento de la cámara con un cono

Cuando la cámara viaja adelante, la lluvia debería inclinarse para que parezca que está lloviendo detrás de la cámara. De forma similar, cuando la cámara se mueve de lado, las líneas de la lluvia esperan que halle componentes oblicuos para su movimiento. Para simular este efecto se traza un mapa de las cuatros texturas de la lluvia sobre el doble cono como muestra la Fig. 13. Estas texturas son diseñadas para cubrir, de manera que las líneas que son visibles, no se noten cuando están repetidas en la superficie del cono o cuando se están desplazando a través del cono.

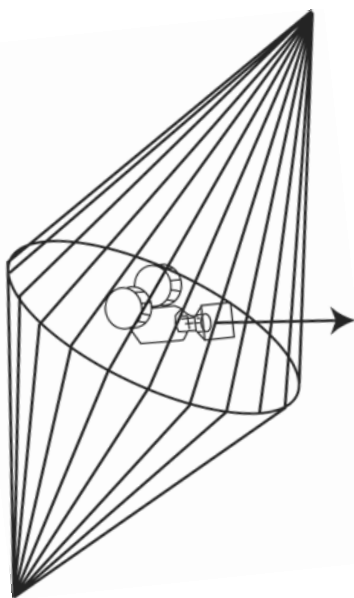


Fig. 13 Doble Cono

Para reducir las singularidades del cono, se aumenta la transparencia del cono en esos puntos, desvaneciendo gradualmente las imágenes hacia el círculo donde los conos se unen. Incluso se puede seleccionar una altura apropiada del cono para ajustar el ángulo en el cual al lluvia parecerá caer, pues si el cono es muy pequeño, cuando la cámara es estacionaria, la lluvia caerá en un ángulo inclinado, poco realista, en vez de a plomo debido a los bordes pronunciados del cono, por otra parte, el cono tampoco debe ser demasiado alto. Esto asegura que cuando la cámara se mueve y el cono es rotado, la lluvia caerá del centro de la pantalla hacia los bordes como se desea.

Como la cámara se mueve, se gira el doble cono para que la precipitación parezca caer hacia la cámara a partir de la dirección del movimiento. Esta operación se realiza determinando la matriz de transformación del cono para mirar a lo largo del vector de trayectoria de la lluvia descrito en la Ec.5, donde Vel_{camera} es la velocidad de la cámara, C_f es una amortiguación constante para limitar lo ladeado del cono, $Vel_{gravity}$ es la velocidad de la precipitación debido a la gravedad, y t_{delta} es el tiempo transcurrido desde el último *frame*.

$$Vec_{precip} = (c_f * Vel_{camera} + Vel_{gravity}) * t_{delta}$$

Ec. 5 Vector de trayectoria de la lluvia

Unidos en una sola matriz

D_{precip} y E están combinados en la matriz de transformación de textura mostrada en la Ec. 6 donde D_{precip} es representado en espacio de píxel y es convertido en espacio real por E , el cual también baja la velocidad del desplazamiento de las texturas sucesivas.

$$\begin{bmatrix} S_f & 0 & 0 & 0 \\ 0 & \frac{S_{streak}}{S_f * (C_{pixel} * S_f * Vec_{precip} + S_{streak})} & 0 & 0 \\ 0 & S_{streak} * \frac{t_{delta}}{t_{const}} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ec. 6 Matriz de Transformación de la textura

También se pueden simular diferentes intensidades de lluvia, variando la escala y la traslación en la matriz. Por ejemplo, la lluvia fuerte es modelada con gotas mayores (menos escala) y movimiento rápido

(mayor traslación). Una gran ventaja para diseñar un sistema con tantas variables de cambio es que puede ser controlado por los diseñadores o artistas.

Añadiendo control del artista

El sistema está diseñado para que los artistas tengan más control sobre el *fine-grained* del resultado final. Primero los artistas deben crear la textura, así controlan la distribución de las gotas de lluvia, deben usar un lugar diferente de texturas, para la lluvia ligera uno opuesto al de lluvia fuerte, así pueden ajustar la densidad de caída para cada intensidad. También pueden agregar una niebla en la textura, que oscurece objetos en la escena esta niebla puede ser más pesada en algunas partes de las texturas para agregar variedad.

Igualmente se le concede al artista el control en las ecuaciones del aguacero, permitiendo ajustar la humedad y los factores de escala en varios puntos.

El artista puede ajustar a S_f para cambiar el tamaño de la gota y la velocidad del desplazamiento. Incluso ajustar a S_{streak} , para la longitud de las líneas (por ejemplo, para darle a las lluvias ligeras líneas más alargadas que a la lluvia fuerte) y a C_f para los límites de la inclinación del cono.

Para cada intensidad de la lluvia, los artistas experimentan con los ajustes del parámetro hasta que encuentren la combinación correcta para la longitud de la veta, el tamaño de la gota, velocidad, y humedad del cono. Inmediatamente la retroalimentación visual y el control artístico sobre los parámetros serán la clave para tener una gran calidad en el resultado final.

3.3 Características de los modelos matemáticos

Los modelos matemáticos antes mencionados presentan una serie de ventajas y desventajas fundamentales con respecto al funcionamiento y representación de la lluvia en un ambiente virtual. Cada ventaja reflejada del modelo lineal se convierte en diferencia del modelo del doble cono y viceversa, sin dejar de mencionar que presentan algunas en común. Sus técnicas y métodos difieren en su totalidad, al caracterizarlos, tratamos de resaltar las principales cualidades de cada uno para su uso en el simulador de tiro, ya que ambos están propuestos para conformar la librería de lluvia para el simulador de tiro.

Modelo Lineal

- Se toma como escenario un cubo como primitiva geométrica para representar la caída de la lluvia.
- Los sistemas de partículas representan los objetos como una nube de partículas primitivas que definen su volumen.
- Un sistema de partículas no se comporta como una entidad estática debido a que las partículas pueden cambiar su forma y logran moverse con el paso del tiempo, a su vez las partículas nuevas nacen y las viejas mueren.
- Los objetos formados por sistemas de partículas no son determinísticos, ya que su forma y apariencia no están definidas completamente. Por tanto son usados procesos estocásticos para poder cambiar la forma y apariencia de los objetos.
- Con un sistema de partícula se puede procesar en el mismo tiempo de cómputo, más primitivas básicas y producir una imagen más completa. En una lluvia fuerte hay que simular más partículas.
- Por la simplicidad que presenta la partícula se le puede aplicar difusión de movimiento (*Montion blur*).
- Los sistemas de partículas modelan objetos vivos, que son aquellos que cambian con el pasar del tiempo.

- El modelo resultante es capaz de representar movimiento, cambios de forma y dinámicas que son imposibles de representar con las representaciones clásicas basadas en superficies.
- La caída de la lluvia representada por el modelo lineal transforma el entorno virtual del simulador.

Modelo del Doble Cono.

- Se toma como escenario un doble cono como primitiva geométrica con una cámara en su interior para representar la caída de la lluvia.
- Expande las texturas para simular *Motion blur*.
- El sistema satisface más a los juegos o simuladores con *framerate* relativamente constante, pues cuando el *framerate* del juego tartamudea, el desplazamiento de la textura tartamudea también y puede provocar que aparezca el efecto de la lluvia al revés.
- Tiene un impacto insignificante en el *framerate* total del juego o simulador, el costo del funcionamiento es igual para la lluvia ligera como para la lluvia fuerte a diferencia del sistema de partículas.
- Utiliza la API Direct3d para facilitar el manejo con las entidades gráficas elementales (texturas), donde se pueden efectuar de forma transparente transformaciones geométricas a dichas entidades.
- El modelo del doble cono necesita una tarjeta gráfica que permita la programación directa del procesador gráfico (GPU), esta técnica permite escalar el hardware usando tarjetas gráficas en paralelo, logrando incrementar la velocidad del proceso al ofrecer varias unidades computacionales que se especializan en la representación de píxeles en la pantalla que son usadas por este modelo matemático.

3.4 Modelo más efectivo.

Después de ampliar los conocimientos acerca de los modelos planteados, se dispone a escoger un modelo óptimo. El Modelo Lineal representado mediante un demo, no con todas las funcionalidades sino con una gran parte de ellas es el escogido, debido a que a la hora de realizar un simulador, ya sea por primera vez o para resolver una situación determinada. Es necesario contar con un modelo simple que represente la lluvia de forma rápida y eficiente, de manera que logre los objetivos trazados.

En SIMPRO es necesario este modelo lineal, pues para hacer una versión primera del simulador, la lluvia que ofrece el modelo resuelve el problema planteado, además que no necesita de la utilización de hardware de última generación para su mejor funcionamiento. Se puede ejecutar con un Sistema Operativo sin muchos requerimientos no funcionales como memoria RAM, tarjetas gráficas y de video, etc.

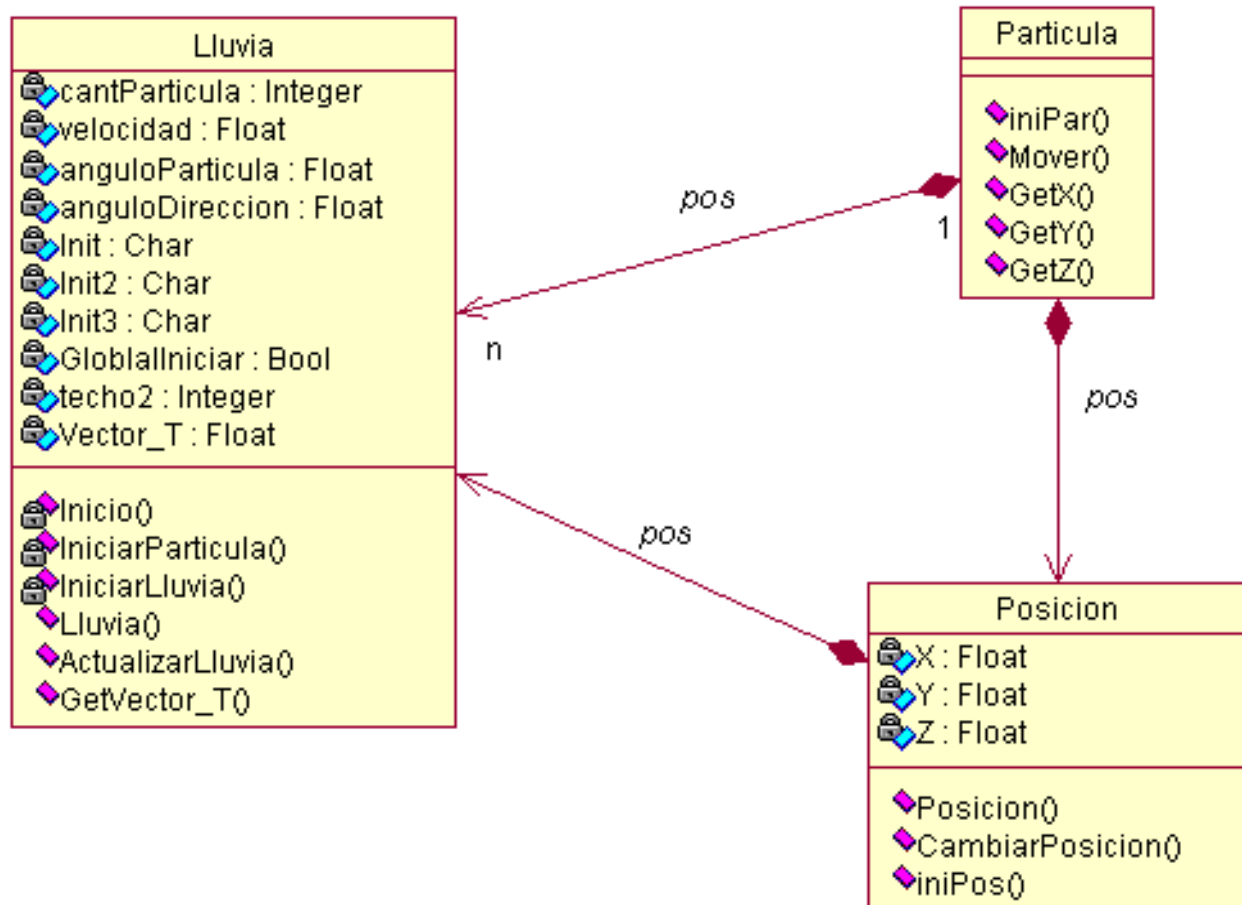
Este modelo ofrece una calidad que aunque no es extrema se acomoda a las necesidades del simulador para mejorar los escenarios donde se desarrollan maniobras prácticas, ofrece la sensación deseada y obliga al soldado, con su resultado final, a incluir la lluvia como un factor determinista para la realización del ejercicio práctico, pues acrecienta dificultad. Sin embargo el modelo del Doble Cono logra una visión mucho más realista de la lluvia y se puede utilizar para mejorar la lluvia en otra versión más completa del simulador.

3.5 Demo experimental

Se realizó una prueba que demuestra la funcionalidad del modelo lineal usando sistema de partículas, a modo experimental se hizo un Demo que trató de utilizar las principales funcionalidades del modelo, utilizando un ambiente sencillo para lograr una representación más realista de la caída de la lluvia.

A continuación se expone de una forma sintetizada las clases fundamentales, que se usaron en el Demo para su representación, ver anexo 1, se utiliza un diagrama de clases que describe como se relacionan para lograr representar la lluvia.

3.4.1 Diagrama de clases del Demo



3.4.2 Descripción de las clases

Tabla 1

| | |
|----------------------------|---|
| Nombre: Partícula | |
| Para cada responsabilidad: | |
| Nombre: | iniPar () |
| Descripción: | Función que inicializa la posición de la partícula. |
| Nombre: | Mover() |
| Descripción: | Función que actualiza la posición de la partícula. |
| Nombre: | GetX () |
| Descripción: | Función auxiliar de la posición que me devuelve X. |
| Nombre: | GetY () |
| Descripción: | Función auxiliar de la posición que me devuelve Y. |
| Nombre: | GetZ () |
| Descripción: | Función auxiliar de la posición que me devuelve Z. |

Tabla 2

| | |
|----------------------------|---|
| Nombre: Posición | |
| Atributo | Tipo |
| X | Float |
| Y | Float |
| Z | Float |
| Para cada responsabilidad: | |
| Nombre: | CambiarPosicion(float X, float Y, float Z) |
| Descripción: | Función que cambia las coordenadas de la posición |
| Nombre: | iniPos(float X, float Y, float Z) |
| Descripción: | Función que inicializa las coordenadas de la posición |
| Nombre: | Posicion(float X=0, float Y=0, float Z=0) |
| Descripción: | Constructor de la clase |

Tabla 3

| | |
|----------------------------|---|
| Nombre: Lluvia | |
| Atributo | Tipo |
| cantParticulas | Integer |
| velocidad | Float |
| anguloParticula | Float |
| anguloDireccion | Float |
| Init | Char |
| Init2 | Char |
| Init3 | Char |
| GlobalIniciar | Bool |
| techo2 | Integer |
| Vector_T | Float |
| Para cada responsabilidad: | |
| Nombre: | Inicio() |
| Descripción: | Inicializa el vector de traslación. |
| Nombre: | IniciarParticula(Particula * par) |
| Descripción: | Inicializa la posición de una partícula |
| Nombre: | IniciarLluvia() |
| Descripción: | Inicializa la lluvia |
| Nombre: | Lluvia(int cantParticulas, float velocidad, float anguloDireccion, float anguloParticula, Posicion * pos) |
| Descripción: | Constructor de la clase lluvia. |
| Nombre: | ActualizarLluvia() |
| Descripción: | Función encargada de mover las partículas. |
| Nombre | GetVector_T(int pos) |
| Descripción. | Devuelve la posición del vector |

Después de una breve descripción de las clases utilizadas, se puede ver algunas imágenes que muestran como queda representada la lluvia en el demo al cual hacemos alusión.

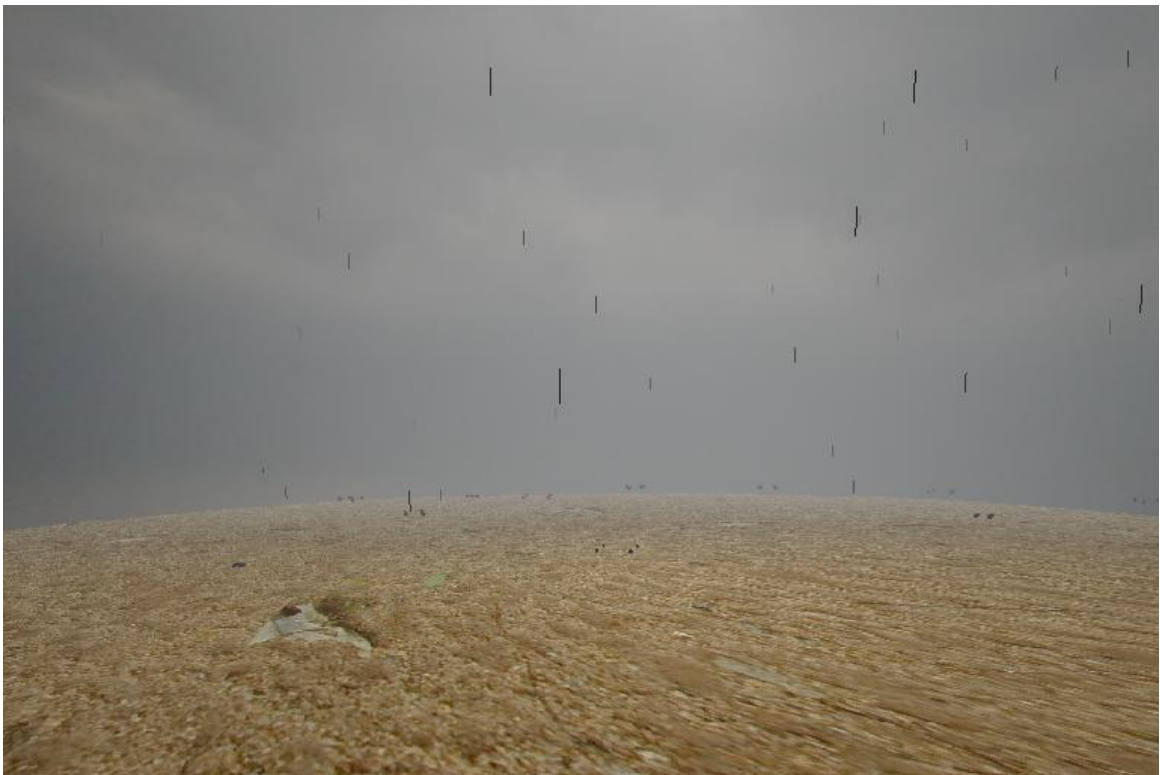


Fig. 14 Representación de la lluvia sin mucha niebla.

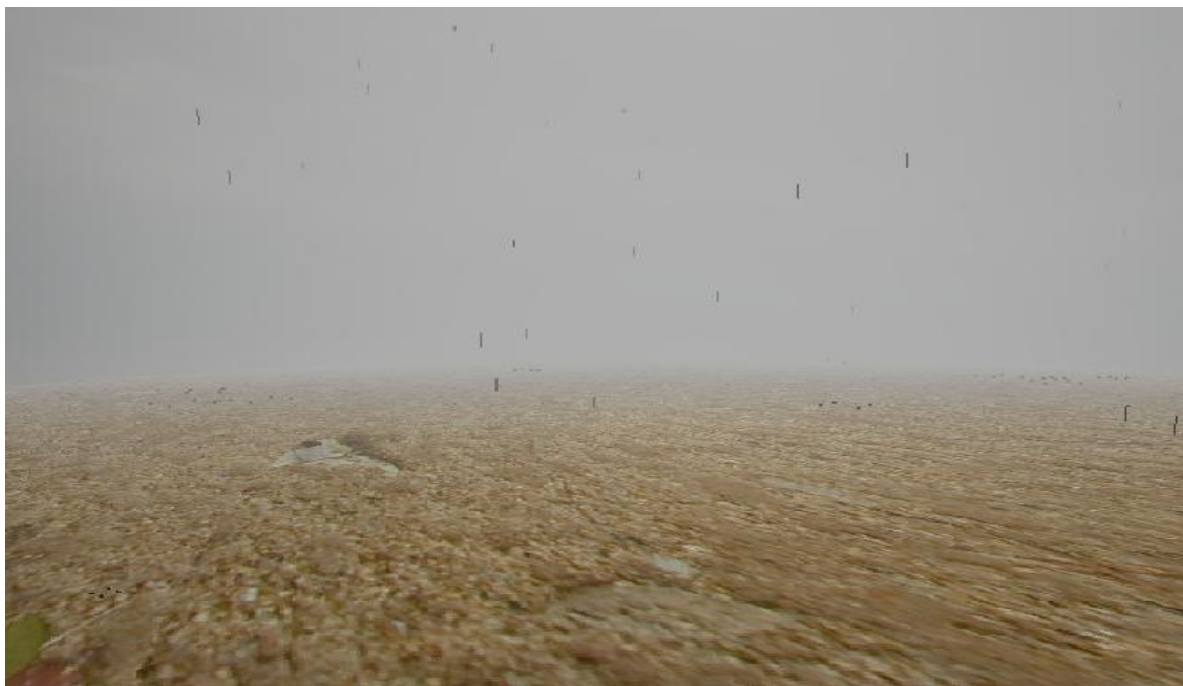


Fig. 15 Representación de la lluvia con niebla.

Se puede añadir que el ambiente virtual juega un papel fundamental en el resultado final como demuestran las imágenes, pues complementa la actividad de que el tirador sienta la sensación de encontrarse en el lugar que muestra el escenario donde se realizará la práctica de tiro.

Conclusiones del capítulo

En este capítulo se logra hacer una propuesta de dos modelos matemáticos que pueden conformar la librería de lluvia del simulador de tiro. Después de una caracterización se fundamenta porque uno de ellos es el más efectivo para una primera versión del simulador. Está incluida también una actividad experimental para la demostración de cómo sería su representación a la hora de utilizarlo en el simulador, así como una sencilla descripción de clases que hicieron capaz la modelación de la lluvia.

Conclusiones Generales

Gracias a una labor realizada y expuesta en los capítulos anteriores es necesario concluir algunos aspectos fundamentales sobre el tema: Librería para la simulación de la lluvia en un ambiente virtual en un simulador de tiro. Se logró hacer dos propuestas de modelos matemáticos para la modelación de la librería, siguiendo la idea de una mejor representación grafica de la lluvia. Donde se explica de manera exquisita las características de cada uno para la futura utilización de los mismos, brindando una forma cómoda y fácil a la hora del modelado de la lluvia.

Para el progreso de la propuesta se realizó una investigación previa acerca del tema de simulación, simulador, etc., con el objetivo de enmarcar al lector en el tema, donde se brindaron ideas necesarias para percibir todo lo relacionado con el mundo de la simulación, específicamente del simulador de tiro, así como el estudio de los modelos matemáticos que sirven para la modelación de la lluvia.

Las propuestas se desarrollaron siguiendo las necesidades del simulador de tiro, tratando de dar la mejor solución para lograr la escena más realista posible. Se trató de utilizar las ecuaciones matemáticas más sencillas, pero a su vez más óptimas, modificando algunos aspectos que se podrían volver engorrosos a la hora de programar. También en el trabajo se hizo un trabajo minucioso con las propuestas matemáticas, ofreciendo un mejor conocimiento de los resultados que se pueden alcanzar con los mismos, dando paso a la selección del modelo que más oportuno a las necesidades actuales del simulador, teniendo en cuenta sus características.

Por todo lo planteado anteriormente se concluye que los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente, incluyendo una serie de recomendaciones que deben tenerse en cuenta para el trabajo futuro.

Recomendaciones

Después de haber concluido este trabajo investigativo, se añadirá algunas referencias importantes para un futuro trabajo:

1. Confeccionar la librería para incorporar el fenómeno de la lluvia en los escenarios del simulador de tiro.
2. Implementar en tiempo futuro el modelo del doble cono para brindar un mayor realismo a la librería.
3. Seguir investigando en otros modelos matemáticos que puedan modelar la lluvia.
4. Profundizar y seguir estudiando el modelo lineal para realizar posibles cambios, sobre todo el tema de la reutilización de las partículas.

Referencia Bibliográfica

K.GARG and N. S. (2004). "etection and removal of rain from videos."

N.WANG and W. B. (2004). "Rendering falling rain and snow."

NAYAR, A., S. K, et al. (2003). "Shedding light on the weather."

Naylor, T. H. (1971). Técnicas de simulación en computadoras. . Mexico, Limusa: 390.

Petzold, C. "Programming Windows" S. E. 1.990.

Reeves, W. T. (1983). "Particle Systems - A Technique for Modeling a Class of Fuzzy Objects." Computer Graphics, Retrieved no. 3 vol. 17.

Reeves, W. T. (1985). "Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems." Computer Graphics, vol. 19(no. 3): pp 313-322.

Shannon, R. (1988). Systems simulation: the art and science. . Mexico, Trillas.

T., W. and C. R. S. (1975). "Use of rainfall-induced optical scintillations to measure path-averaged rain parameters."

T., W. and C. R. S. (1975). "Use of rainfall-induced optical scintillations to measure path-averaged rain parameters."

Bibliografía

"Curso C++." from http://www.zator.com/Cpp/E1_4_4b1.htm.

GARG, K. and N. S. (2006). "Photorealistic rendering of rain streaks"

K.GARG and N. S. (2004). "Detection and removal of rain from videos."

KANEDA, K., I. S., et al. (1999). "Animation of water droplets moving down a surface."

N.WANG and W. B. (2004). "Rendering falling rain and snow."

NAYAR, A., S. K, et al. (2003). "Shedding light on the weather."

Naylor, T. H. (1971). Técnicas de simulación en computadoras. . Mexico, Limusa: 390.

Petzold, C. "Programming Windows" S. E. 1.990.

Reeves, W. T. (1983). "Particle Systems - A Technique for Modeling a Class of Fuzzy Objects." Computer Graphics, Retrieved no. 3 vol. 17.

Reeves, W. T. (1985). "Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems." Computer Graphics, **vol. 19**(no. 3): pp 313-322.

Shannon, R. (1988). Systems simulation: the art and science. . Mexico, Trillas.

T., W. and C. R. S. (1975). "Use of rainfall-induced optical scintillations to measure path-averaged rain parameters."

T., W. and C. R. S. (1975). "Use of rainfall-induced optical scintillations to measure path-averaged rain parameters."

WANG, H., M. P. J., et al. (2005). "Water drops on surfaces."

Zhang., M. L. a. Q. (2003). "Rendering falling snow using an inverse fourier transform." Computer Graphics Proceedings, Annual Conference Series.

Fierro, J. (1999). "Algunas Propiedades del Agua."

. "Encarta".

Nolasco, N., A. Z. Benitez, et al. "Implementacion de Efectos Medioambientales mediante GPUs."

Anexos

Anexo 1

```
//Nota: Esta va a ser la clase encargada de
//      simular la lluvia moviendo cada partícula.
//      tiene en cuenta el angulo de trayectoria de la partícula respecto al plano XZ,
//      el angulo de direccion con respecto al plano YZ y la velocidad de la misma.
//
//-----
-----

#ifndef __LLUVIA_H__
#define __LLUVIA_H__

#include "Utils.h"
#include "particula.h"

class Lluvia
{
private:
    Particula ** particulas; //lista de particulas
    int cantParticulas; //cantidad de particulas
    float velocidad; //velocidad de la particula
    float anguloParticula; //angulo de caida
    float anguloDireccion; //angulo de salida
    Posicion * pos; //arreglo de posiciones del escenario
    float * Vector_T; //vector de traslacion
    unsigned char * Init; //vector aux para inicializar la pos de la
particula
    unsigned char * Init2; //vector aux para el calculo del Vector de
traslacion
    unsigned char * Init3; //vector aux para inicializar la pos de la
particula
    bool GlobalIniciar; //vector aux para inicializar las particulas
    int techo2; //valor aux para inicializar las particulas

    void Inicio( ); // inicializa el vector de traslacion
    void IniciarParticula( Particula * par ); //inicializa la posicion de
una particula
    void IniciarLluvia( ); //inicializa la lluvia

public:
    /* Constructor de la Clase Lluvia */
    Lluvia( int cantParticulas, float velocidad, float anguloDireccion,
float anguloParticula, Posicion * pos );
```

```

        ~Lluvia( );
        void ActualizarLluvia( );//funcion encargada de mover las particulas.
        int GetcantPar(){return cantParticulas;}//funcion aux que devuelve la
cantidad de particulas utilizadas
        Particula ** Getparticulas(){return this->particulas;}//funcion aux
que devuelve las particulas en su pos actualizada.
        float GetVector_T( int pos ){return Vector_T[pos];}
};

//-----
#endif

//Nota: Esta va a ser la clase encargada de
//      de manipular la partícula.
//
//
//-----
#ifndef __PARTICULA_H__
#define __PARTICULA_H__

#include "posicion.h"

class Particula
{
    private:
        Posicion * pos;

    public:
        /* Constructor de la Clase Particula */
        Particula( Posicion * pos );
        ~Particula( );
        void iniPar( float X, float Y, float Z );//funcion que inicializa la
pos de la particula.
        void Mover( float X, float Y, float Z );//funcion que actualiza la
pos de la particula.
        float GetX(){return pos->GetX();};//funcion aux de la pos
        float GetY(){return pos->GetY();};//funcion aux de la pos
        float GetZ(){return pos->GetZ();};//funcion aux de la pos
};

//-----
#endif

```

```

//Nota: Esta va a ser la clase encargada de
//      tener la posicion de una particula
//      en cada instante de tiempo.
//
//-----
-----

#ifndef __POS_H__
#define __POS_H__

#include "math.h"
#include "stdlib.h"

class Posicion
{
    private:
        float X;
        float Y;
        float Z;

    public:
        /* Constructor de la Clase Particula */
        Posicion( float X=0, float Y=0, float Z=0 );
        ~Posicion(){};
        void CambiarPosicion( float X, float Y, float Z );//funcion que
cambia las coordenadas de la pos
        void iniPos( float X, float Y, float Z );//funcion que inicializa las
coordenadas de la pos
        float GetX( ){ return this->X; }//funcion aux
        float GetY( ){ return this->Y; }//funcion aux
        float GetZ( ){ return this->Z; }//funcion aux
};

//-----
-----
#endif

```

Glosario de Términos

1. **Rendering:** Proceso de obtención de imágenes por computadora.
2. **Fuzzy:** Directamente el término en inglés, su traducción es difuso.
3. **Frames:** Cantidad de imágenes por unidad real de tiempo. En informática un frame o marco es la división de la pantalla del navegador en dos o más cuadros, que muestran el contenido de dos o más sitios simultáneamente.
4. **Rendering Streched:** Directamente el término en inglés, su traducción es representación ampliada.
5. **Fine- grained:** Directamente el término en inglés, su traducción es granulado fino.
6. **Hardware vertex Shader:** Es una función de proceso gráfico, es decir, añade efectos especiales a los objetos de una escena 3D.
7. **Motion Blur:** Es una técnica que se aplica para dar la sensación de movimiento, es como un borrón provocado por el movimiento. Efecto difuso causado por el movimiento rápido de un objeto en escena.
8. **Parallax:** Técnica que se encarga de la sensación de profundidad.
9. **Frametime:** Es una imagen independiente, una sucesión de *frame* compone una animación
10. **Framerate:** Indica el número de fotogramas por minutos. Es el número de imágenes por segundo o frecuencia de refrescamiento.
11. **Procesos Estocásticos:** Se denominan estocásticos a aquel sistema que funciona, sobre todo por el azar. Las leyes causa-efecto no implican como actúa el sistema de manera determinista, sino una función de probabilidades.

12. **Determinísticos:** Es aquel fenómeno determinístico en que se obtienen siempre el mismo resultado bajo las mismas condiciones iniciales.
13. **Rendereada:** Abuso de la españolización, viene del rendering.
14. **DLL:** Dynamic Link Library.