

## Plataforma 3D Distribuida – Nueva Extensión Tecnológica (P3D NEXT)

Alejandro Acquesta<sup>1</sup>, Emiliano Lucero<sup>1</sup>, Gabriela Numma<sup>1</sup>, Graciela Defeo<sup>1</sup>, Pablo Ristori<sup>2</sup> y María Victoria Galán<sup>1</sup>

<sup>1</sup> Instituto de Investigaciones Científicas y Técnicas para la Defensa (CITEDEF)  
Div. Computación Gráfica, Lab. Simuladores 3D Distribuidos

<sup>2</sup> Instituto de Investigaciones Científicas y Técnicas para la Defensa (CITEDEF)  
Dpto. Láseres y Aplicaciones (DEILAP), Div. Sensado Remoto  
{aacquesta, elucero, gnumma, gdefeo, pristori,  
mvgalan}@citedef.gob.ar

**Resumen.** La Plataforma 3D Distribuida (P3D) fue implementada oportunamente por nuestro laboratorio, y es utilizada por nuestros simuladores de entrenamiento desde hace años. Sin embargo, utiliza un servicio propietario de Microsoft (DirectX), que ofrece servicios gráficos de bajo nivel, por lo que resulta costoso construir sobre ella capacidades gráficas más avanzadas.

El presente proyecto (P3D NEXT) presenta una nueva plataforma gráfica de simulación distribuida, construida sobre un motor de videojuegos 3D multiplataforma, libre y de código abierto de origen Argentino (Godot 4.x), que se integre a los servicios de sincronización y comunicaciones de P3D. Este cambio tecnológico permitirá ofrecer mayores prestaciones gráficas, mejorar la performance, incrementar la estabilidad, acceder a la actualización permanente que recibe el motor gracias a su comunidad de desarrollo, mantener soberanía tecnológica debido a su condición de código abierto, y conservar la capacidad de desarrollar simuladores de entrenamiento distribuidos. Asimismo, eliminar restricciones a su distribución dada su condición de software libre; reducir los tiempos de desarrollo y mantenimiento de las aplicaciones de simulación aprovechando la existencia abundante de documentación, comunidad de desarrolladores, foros de consulta y la posibilidad de modificación directa del propio motor.

**Palabras clave:** simulación, entrenamiento, sistemas distribuidos, computación gráfica, motores gráficos.

### 1 Introducción

La simulación es una herramienta cada vez más utilizada en el entrenamiento de procedimientos en ámbitos de las Fuerzas Armadas, de seguridad, combate del fuego,

aviación civil, etc., por múltiples razones. En primer lugar, resulta en una reducción significativa de costos en comparación con el entrenamiento en vivo. Además, permite un ambiente más seguro y controlado del proceso simulado, lo que reduce significativamente el riesgo de lesiones y accidentes.

Otra ventaja, es la posibilidad de entrenar en situaciones que podrían ser difíciles o imposibles de recrear en la realidad, como por ejemplo situaciones de combate en diferentes entornos o condiciones climáticas extremas. Al crear escenarios virtuales, es posible establecer parámetros como el clima, la hora del día y otros factores ambientales que no son controlables en situaciones reales.

También contribuye a aumentar el profesionalismo al permitir el entrenamiento de habilidades específicas de manera sistemática, al mismo tiempo se pueden realizar múltiples repeticiones de una misma situación para mejorar la comprensión y la respuesta del alumno. Por último, el uso de simulación, ofrece herramientas didácticas como ayudas visuales, rejillas, marcas virtuales y cambios de punto de vista, que podrían no ser factibles en un ejercicio en vivo, y que permite la implementación de diferentes técnicas de enseñanza y evaluación de los resultados del entrenamiento.

De todo esto se desprende que la incorporación de la computación gráfica en los simuladores de entrenamiento resulta esencial para la creación de ejercicios que representen ambientes y situaciones de manera realista. De este modo, se posibilita la formación sistemática de personal con costos reducidos y riesgos controlados, asegurando una alta calidad en la capacitación.

El empleo de la P3D significó un cambio sustancial para la generación de sistemas de simulación para entrenamiento, por lo que constituye una capacidad que P3D NEXT requiere mantener y potenciar.

## **2 Situación – Problema u Oportunidad**

El grupo de Computación Gráfica y Visualización ha utilizado la P3D para el desarrollo del simulador de observador adelantado (SIMOA) en sus tres versiones, la última del año 2016 y también del software para el simulador de combate NeoNahuel II en 2018 que fuera oportunamente instalado en el Colegio Militar de la Nación. El SIMOA se encuentra en uso en el Colegio Militar de la Nación, la Escuela de Artillería, la Escuela de Suboficiales “Sargento Cabral”, el Liceo “Gral. Paz” de Córdoba y en todas las Unidades de artillería del Ejército Argentino (EA).

Recientemente, con la llegada de nuevos pedidos de SIMOA para el EA y la Armada Argentina (ARA), registramos la necesidad de incorporar un mayor realismo en las escenas de los simuladores. Dado que la P3D utiliza un motor gráfico de bajo nivel de abstracción, hubiera sido necesario que el equipo de desarrollo implementara los servicios de mayor abstracción pagando un alto costo en tiempo y esfuerzo [2]; servicios éstos que los motores gráficos actuales ya ofrecen.

Esto planteó el desafío de evolucionar hacia una nueva plataforma para desarrollos 3D distribuidos sobre un motor gráfico de código abierto, que ofreciera servicios gráficos de alto nivel, y sacara provecho del hardware actual disponible.

### 3 Solución

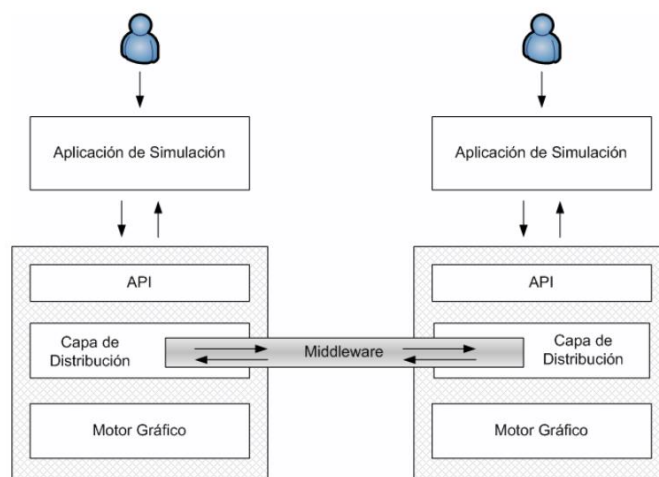
Luego de analizar diferentes posibilidades, identificamos como mejor alternativa de crecimiento a la migración de las capacidades de simulación distribuida a una plataforma de código abierto, multiplataforma y con mejores capacidades gráficas.

La opción seleccionada implica la modificación de un motor de videojuegos 3D de código abierto como Godot, con módulos que ofrezcan el middleware para simulación de P3D que se compilarán con el propio motor, lo cual permitirá construir simuladores distribuidos con escenarios de alta calidad, ofreciendo una alternativa eficaz y de bajo costo.

#### 3.1 Simulación Distribuida

La plataforma P3D actual implementa una arquitectura P2P que asegura la consistencia entre las réplicas del estado de la simulación presentes en cada nodo mediante un mecanismo de sincronización de eventos conservador [1]. La idea para P3D NEXT es mantener el mismo comportamiento del sistema distribuido que posee la plataforma P3D pero integrado con Godot.

Este comportamiento está basado en el que proponen Chandy y Misra [4], y Misra [5]; es un algoritmo que puede utilizarse para realizar la sincronización con un enfoque conservador introduciendo un mecanismo para simular la evolución de un sistema físico de manera distribuida. El sistema puede verse como un conjunto de procesos físicos relacionados que se comunican exclusivamente por mensajes y se encuentran distribuidos. Un proceso físico dado puede o no, recibir mensajes de otros, y a su vez, puede o no, enviar mensajes a sus pares.



**Fig 1.** Arquitectura de P3D NEXT

En una simulación, cada proceso físico es representado por un proceso lógico. Cada proceso lógico puede avanzar su simulación un paso en el tiempo si en ese instante ya ha recibido los mensajes de los procesos que se comunican con él. Para esto, el instante de tiempo en el que un mensaje es generado es codificado junto con este o, dicho de otra manera, los mensajes tienen una marca de tiempo. De este modo, para que la simulación no se bloquee y pueda avanzar en todos los procesos lógicos, en cada paso de tiempo, cada uno de ellos siempre envía a sus destinatarios un mensaje: si recibe un mensaje producido en el proceso físico, lo comunica; sino, envía un mensaje vacío, o nulo, que sólo tiene la marca de tiempo.

Estos mensajes nulos son los que permiten la evolución de la simulación del sistema físico. Este algoritmo, entonces, permite la sincronización de eventos discretos distribuidos para la simulación de un sistema de manera asincrónica.

La situación a simular o el mundo virtual donde participan los usuarios es equivalente al sistema físico, donde cada participante representa un proceso físico. Como cada usuario desea percibir la totalidad del mundo virtual, las acciones de cada uno de ellos deben comunicarse a los demás. De este modo, al momento de avanzar la simulación en cada nodo deben encontrarse disponibles las acciones realizadas por todos los participantes hasta ese instante.

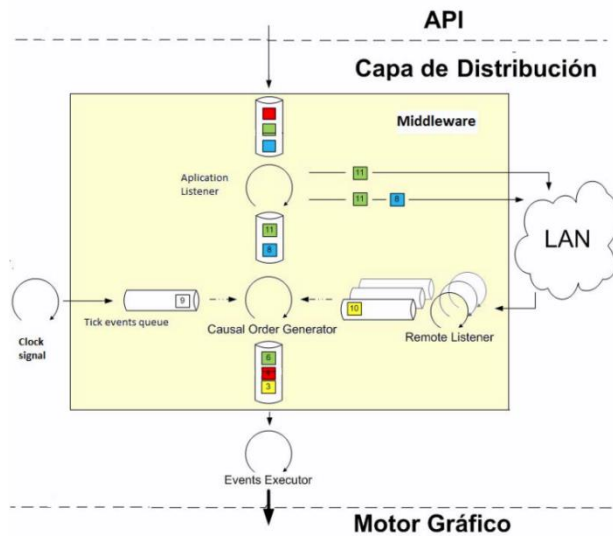


Fig 2. Arquitectura de P3D NEXT - detalle

La versión implementada por la P3D de este mecanismo se basa además en el trabajo de Rosenfeld[6], que utiliza el algoritmo de Chandy y Misra proponiendo tres procesos fundamentales: uno para detectar y difundir los eventos locales, otro para recibir eventos remotos y un último proceso que ordene todos los eventos en forma temporal a partir de la marca de tiempo de cada uno de ellos. Mediante este esquema logra garantizar las siguientes funcionalidades: recibir los eventos generados por la aplicación usuaria local, transmitir los eventos a los peers que pertenecen al contexto,

asignar la marca de tiempo a cada evento local, recibir los eventos de los demás peers y ordenarlos según su marca de tiempo. Además, se diferencia de propuestas anteriores ya que para optimizar el uso de la red de comunicaciones y tener un esquema más asincrónico, el evento nulo no es generado en cada avance discreto del tiempo. Este evento es generado después de una determinada cantidad de tiempo de inactividad, para acotar la latencia máxima.

Por último, en cada nodo hay un proceso que actúa como reloj local para simular el avance del tiempo. Este reloj es consultado por el middleware cuando éste asigna la marca de tiempo a cada evento local. Además, es este proceso quien genera la señal de reloj que es utilizada por el middleware para generar los eventos tick.

Cada nodo, debido a sus características de hardware y la carga con la que se encuentre trabajando, presentará desvíos en la evolución de su reloj respecto de los demás. Por este motivo es necesario que uno de estos relojes se tome como referencia (reloj global) para sincronizar, cada cierto tiempo, cada reloj local. En cada nodo, cuando corresponda, se consulta el estado del reloj global y como resultado de la comparación con el reloj local se determina si se el mismo se ha adelantado o atrasado. La corrección para volver al estado de sincronización con el tiempo global, se realizará de manera gradual. Para ello, el reloj local modifica la velocidad de avance de su contador interno. Por último, el reloj de referencia o reloj global se sincroniza con el reloj interno de la computadora donde se encuentra, con el mismo mecanismo mencionado anteriormente.

### 3.2 Servicios gráficos avanzados

P3D NEXT incorporará entonces los siguientes servicios gráficos que la plataforma actual no ofrece, o lo hace de forma limitada o parcial:

**Niveles de detalle (LOD) para los objetos 3D.** Se podrá gestionar de forma dinámica los diferentes niveles de detalle con que se re-presentan los objetos en el simulador. Esto implica una modificación en la cantidad de polígonos utilizados para adecuar el esfuerzo computacional a la distancia a la que se encuentra la cámara, de las diferentes instancias de un mismo objeto. Esto permite visualizar un mayor detalle en los objetos cercanos y tener mayor eficiencia para representar objetos lejanos, manteniendo el realismo ante un desplazamiento de la cámara.



Fig 3. LOD en objetos 3D

**Niveles de detalle dinámicos para el terreno.** A diferencia de los LOD para objetos, el nivel de detalle para objetos extensos como el terreno requieren que la cantidad

6

de polígonos utilizados para representar regiones próximas a la cámara sean mayores a las requeridas para describir las regiones alejadas, dentro de un mismo mallado. P3D NEXT contará con un mecanismo de decimado dinámico gradual para los diferentes polígonos que conforman el terreno. Es decir, que reducirá escalonadamente la cantidad de polígonos a medida que la distancia al observador sea mayor. Esto mantendrá el máximo detalle en el mallado próximo y ahorrará recursos.

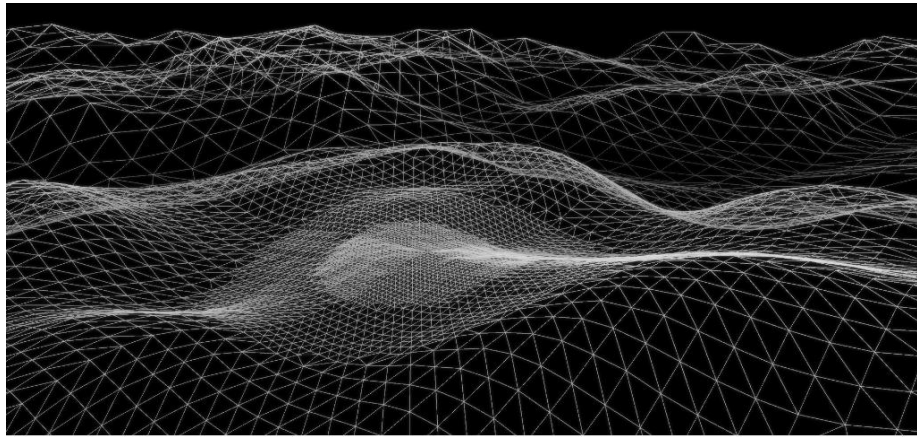


Fig 4. LOD en terrenos

**Efectos volumétricos.** Los efectos volumétricos se refieren a la representación de fenómenos atmosféricos que ocupen un volumen en el espacio tridimensional, por ejemplo humo, nubes, niebla y efectos avanzados de iluminación, como los llamados God rays o rayos solares, utilizados para simular los rayos de luz que se filtran a través de un ambiente con partículas en suspensión, produciendo un aspecto luminoso y etéreo.



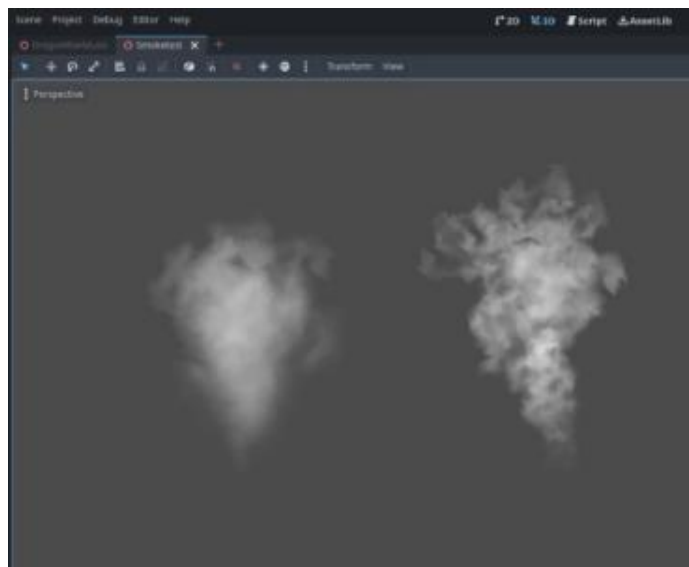
Fig 5. Efectos de iluminación volumétrica



**Fig 6.** Escena con niebla

Los God rays pueden utilizarse para mejorar la inmersión en ambientes exteriores, como paisajes naturales o urbanos con elementos de polvo y humedad en suspensión. El efecto puede ajustarse para simular diferentes condiciones de atmosféricas, lo que permite una mayor personalización del ambiente de simulación.

**Generación de partículas.** Se refiere a la creación y animación de pequeñas entidades visuales, que permiten representar fenómenos naturales como humo, fuego, nieve, lluvia, granizo, entre otros. Estas partículas son creadas y animadas en un entorno virtual, y pueden interactuar con otros objetos y elementos de la escena, que responden también a parámetros como la dirección y velocidad del viento o la gravedad, entre otros.



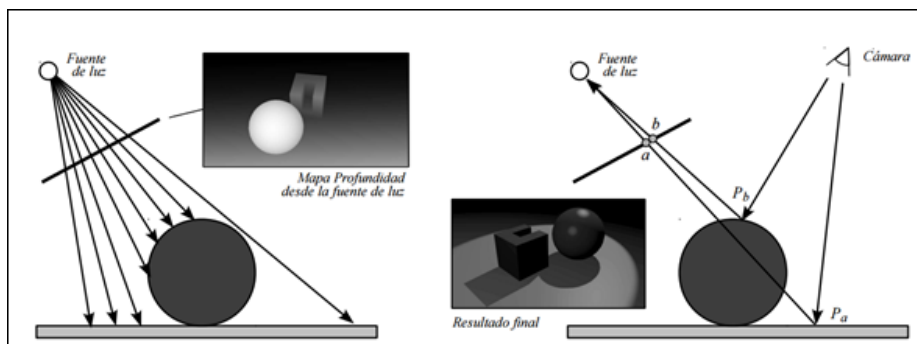
**Fig. 7.** Humo generado con partículas

La generación de partículas puede ser útil en la simulación realista de escenarios naturales en diversos aspectos. Por ejemplo, la creación de partículas de humo y fuego en un simulador de incendios forestales puede ayudar a los bomberos a entrenar ante situaciones de emergencia y mejorar su capacidad de respuesta. La generación de partículas de lluvia o nieve en una escena puede, no sólo agregar un mayor nivel de detalle, sino que constituir un condicionante para la observación del tiro de la artillería al aire libre.



**Fig 8.** Fuego generado con partículas

**Cálculo de sombras.** Se refiere al proceso de simular la ocultación parcial o total de la luz en una escena, lo que produce sombras que se proyectan sobre objetos y superficies. Esto se logra mediante el uso de algoritmos que determinan la trayectoria de la luz y su interacción con los objetos en la escena. El proceso se vuelve más complejo cuando se consideran múltiples fuentes de luz y objetos que proyectan sombras sobre otros objetos en la escena.



**Fig 9.** Calculo de sombras



La generación de sombras es esencial para lograr una simulación realista de escenarios naturales, ya que ayuda a dar sensación de profundidad y realidad a la escena. Además, las sombras también pueden ser utilizadas para indicar la hora del día o la ubicación geográfica de la escena, lo que agrega un nivel adicional de detalle y realismo. P3D NEXT incorpora el mapeo de sombras en tiempo real que es nativo en Godot 4.x.

**Skybox realistas.** El concepto de skybox reúne diferentes técnicas utilizadas en computación gráfica para simular un fondo visual realista en escenas 3D. Consiste en crear una imagen 2D o una textura que rodea todo el entorno 3D del escenario, como si se tratara de una caja que lo envuelve. Esta imagen representa el cielo, la atmósfera, montañas, árboles o cualquier otro elemento que se desee simular en la parte externa del escenario. El objetivo principal del skybox es mejorar el realismo visual y la inmersión del usuario en el entorno virtual, creando la ilusión de que el escenario es más amplio y realista de lo que realmente es. Al moverse por el escenario, la perspectiva del cielo y la atmósfera cambian en función de la orientación en la que mira el observador. Existen diversas técnicas complementarias para dotar de un mayor realismo visual a la representación de la luz atmosférica, como la ya mencionada técnica de rayos solares. Otra técnica es el uso de niebla volumétrica, que puede ser utilizada para simular la presencia de vapor de agua en suspensión en la atmósfera, de manera de brindar una noción de gran profundidad en el escenario. Además, se pueden utilizar técnicas de iluminación global, como el trazado de rayos (ray tracing), para generar sombras y reflejos realistas en el entorno virtual.



**Fig 10.** Skybox realistas

La nueva plataforma utilizará diferentes técnicas para crear cielos más realistas, mediante la combinación de modelado de la dispersión atmosférica de la luz, nubes volumétricas, texturas de alta resolución y la simulación del movimiento del sol y la luna.

**Motor de física de objetos rígidos.** Esta herramienta permite simular el comportamiento físico de objetos tridimensionales en tiempo real. Es fundamental para generar simulaciones realistas, ya que describe el movimiento y la interacción de objetos con una alta fidelidad a las leyes físicas del mundo real. El motor de física de objetos rígidos se encarga de calcular la posición, velocidad, aceleración, rotación y colisiones de los objetos en un escenario virtual. Para ello, toma en cuenta factores como la masa, la fricción, la gravedad, la inercia, entre otros.

En la literatura especializada, el término más frecuente para referirse a esta herramienta es "physics engine".

Los motores de física de objetos rígidos se utilizan en una amplia variedad de aplicaciones, desde videojuegos hasta simuladores. Por ejemplo, en un simulador de vuelo militar, se utiliza un motor de física para representar el movimiento de la aeronave, el comportamiento del viento, la resistencia aerodinámica, etc. De esta forma, se puede entrenar a los pilotos en una amplia variedad de situaciones, sin tener que exponerlos a los riesgos de vuelo.

Otro ejemplo de su utilización se da en los sistemas de entrenamiento para operaciones de rescate y seguridad, donde se pueden simular escenarios de emergencia y entrenar a los equipos de rescate en la manipulación de objetos y herramientas, en condiciones realistas y seguras.

La nueva plataforma permitirá incorporar a los nuevos desarrollos el motor de física nativo, por estar desarrollada sobre Godot 4.x. Esto permitirá una mayor riqueza en la interacción entre los elementos de las escenas.

**Interfaces de usuario 2D propias del motor.** Se refieren a los controles en pantalla y elementos gráficos que permiten al usuario interactuar con el simulador, como botones, menús, indicadores, etc. P3D NEXT, utilizará diferentes técnicas para crear interfaces de usuario más intuitivas y eficientes, aprovechando las capacidades propias del motor.

Al implementar P3D NEXT sobre Godot 4.x, es posible crear interfaces de usuario personalizadas que se integran directamente con las aplicaciones, lo que hace que el proceso de desarrollo sea más fluido y eficiente.

Entre las posibilidades que se ofrecerá se destacan:

- Soporte para varias resoluciones: cuenta con la posibilidad de crear interfaces que se adapten automáticamente a diferentes resoluciones, lo que hace que los desarrollos sean accesibles para una variedad más amplia de hardware.
- Controles de UI predefinidos: como botones, etiquetas y cajas de texto, que se pueden personalizar según las necesidades del proyecto.
- Personalización de estilos: Los estilos de los controles se pueden personalizar fácilmente en Godot 4.x, permitiendo que la interfaz de usuario se adapte a la estética de la aplicación.
- Soporte para animaciones: Es posible crear animaciones personalizadas para los elementos de la interfaz de usuario en Godot 4.x, lo que permite agregar un mayor nivel de interacción y dinamismo.

- Soporte para múltiples idiomas: Godot 4.x ofrece soporte para múltiples idiomas, lo que permite que los desarrolladores puedan crear aplicaciones con inter-faces de usuario localizadas.

**Editor avanzado de escenarios y aplicaciones.** Contar con un editor integrado para escenarios y aplicaciones permite a los desarrolladores crear nuevos simuladores de manera más eficiente y rápida. P3D NEXT, permitirá acceder a numerosas herramientas para crear y editar nodos, escenas y aplicaciones multiplataforma, de manera más precisa y detallada, permitiendo una alta personalización en el diseño de los simuladores.

Al basarse en Godot 4.x, se dispondrá de forma nativa de un completo editor que permite crear y diseñar tanto escenas 3D, como también las aplicaciones de simulación que las utilizan. Ofrece una amplia variedad de herramientas para el diseño, modelado, materiales, iluminación y animación de objetos en 3D.

- Diseño de escenas: Permite el diseño y la organización de las escenas en una jerarquía de nodos. Cada nodo puede ser un objeto, una luz, una cámara, un emisor de partículas, etc. Los nodos se pueden conectar entre sí y configurar para crear escenas complejas.
- Modelado: Permite la creación de modelos 3D a partir de primitivas básicas como cubos, esferas y cilindros. También se pueden importar modelos 3D generados por otros programas y editarlos dentro de la plataforma.
- Materiales: Usa un sistema de materiales PBR (Physically Based Rendering) que permite crear una biblioteca de múltiples materiales con parámetros y texturas diferentes, y aplicarlos a distintos objetos a requerimiento del diseño
- Iluminación: Ofrece una variedad de herramientas de iluminación para la creación de escenas con luz natural o artificial. Se pueden agregar fuentes de luz, sombras y efectos de luz volumétrica.
- Animación: Permite la creación de animaciones de objetos y personajes, con la posibilidad de utilizar animación por esqueleto, animación de objetos o mezcla de animaciones.

En cuanto a la generación de aplicaciones de simulación, el editor de Godot también ofrece la posibilidad de integrar scripts en el proyecto para programar el comportamiento de los objetos en la escena. Esto permite la simulación de eventos y situaciones en tiempo real.

#### 4 Innovación e Inédito

El diseño de P3D NEXT permite ofrecer servicios de sincronización y comunicaciones para simulación distribuida junto con servicios gráficos y sonoros de alto nivel de abstracción. No hay registros de áreas del Estado Nacional que cuenten con un motor gráfico avanzado propio con capacidad para simulación distribuida. El único antece-

dente pertenece también a nuestro equipo de trabajo con la implementación de P3D.[1][3].

Al construir sobre un motor gráfico de videojuegos libre, de código abierto y multiplataforma, se alcanzan nuevas capacidades, por ejemplo integrar en los sistemas de simulación componentes de hardware que utilizan Android (para virtualizar un binocular o telémetro); o bien desarrollar para sistemas operativos libres como GNU/Linux.

Adicionalmente, puesto que varios de los servicios gráficos de alto nivel requeridos ya se encuentran disponibles en el nuevo motor, nos permite proyectar una capa de mayor abstracción especializada para escenas al aire libre. Esta capa estará integrada por nuevos nodos (Godot 4.x) que representan elementos de la realidad desde el aspecto visual y sonoro, y que modelan dinámicamente su comportamiento frente a cambios en parámetros simulados en la escena. Por ejemplo, la geolocalización, la fecha y la hora del escenario, tienen un efecto en la simulación del movimiento del sol y la luna; la dirección y velocidad del viento modifica cómo y cuánto se mueve la vegetación; las condiciones climáticas, la fecha y hora también afectan texturas estacionales para el terreno y los objetos.

Estos cambios podrán controlarse desde el editor durante el desarrollo, o bien desde las aplicaciones en tiempo de ejecución, generando una evolución dinámica y más natural de las escenas.

## 5 Beneficiarios

El proyecto tiene múltiples beneficiarios, tanto directos como indirectos. Entre los directos podemos mencionar: Equipos que desde el Estado desarrollan simuladores 3D distribuidos, simuladores de tiro con armas portátiles, simuladores de conducción (terrestres, navales y aéreos), simuladores de duelo, simuladores de combate contra incendios, simuladores inmersivos de navegación terrestre, etc. El hecho que P3D NEXT sea un proyecto basado en P3D y en Godot 4.x, y desarrollado por un equipo de investigadores del Ministerio de Defensa, cuenta con las ventajas de: ofrecer simulación distribuida integrada a un motor gráfico avanzado, libre, de código abierto, multiplataforma, con amplia comunidad de desarrolladores y abundante documentación en español; permite lograr una gran independencia tecnológica, y contar con profesionales especializados, que puedan modificar y contribuir a la mejora del motor, adecuarlo a las necesidades específicas de futuros proyectos, otorgar licencias sobre los productos construidos sin límites externos y a muy bajo costo para las dependencias del Estado.

Estas áreas pueden considerarse entonces beneficiarios indirectos, que verán reducidos los tiempos de desarrollo de las herramientas específicas necesarias para la formación de recursos humanos estratégicos, y tendrán una mayor libertad para contar con licencias de uso a bajo costo para estas aplicaciones de simulación.

## 6 Viabilidad Técnica, Financiera y Política Organizacional

Desde el punto de vista técnico, este proyecto de desarrollo está llevándose a cabo en el laboratorio de Simuladores 3D Distribuidos, un equipo que desde el año 2000 diseña y desarrolla simuladores de entrenamiento para las FFAA. Dentro de la experiencia de los profesionales del área, desde 1997 se ha trabajado en desarrollos de visualizadores con todos los niveles de profundidad: desde aplicaciones en D.O.S con motor de representación de bajo nivel completamente propio (usado en el SIMOA I), pasando por el diseño y la implementación de dos generaciones de herramientas gráficas para el desarrollo de simuladores: Plataforma de generación de Simuladores PGS (usado en SIMOA II, demo de Simulador del IA-63 PAMPA) y su sucesora, la Plataforma de Desarrollos 3D Distribuidos P3D (usada en NEONAHUEL, SITARAN, SIMTIATIC, SIMECOM, SIMOA III, SIMOA III EA 2022 y SIMOA III ARA, entre otros).

P3D NEXT es un proyecto de desarrollo de capacidades que se realiza en el Instituto de Investigaciones Científicas y Técnicas para la Defensa (CITEDEF), con personal propio, perteneciente al Régimen Científico de las Fuerzas Armadas (RPIDFA). Por lo tanto es viable desde el punto de vista financiero.

El proyecto se lleva a cabo en paralelo con los desarrollos de SIMOA III EA 2022 y SIMOA III ARA, proyectos que hoy se apoyan en P3D y ponen de manifiesto los requerimientos que una plataforma sucesora debería proveer. El proyecto P3D NEXT tiene dos etapas: la primera busca reemplazar a P3D en todos nuestros simuladores, y la segunda incorporará funcionalidades avanzadas en el apartado gráfico y sonoro.

Puede mencionarse también que desarrollar aplicaciones que procesan y utilizan información geográfica y satelital, tal como requiere la elaboración de escenas georeferenciadas que representan campos de instrucción de tiro; están identificadas como actividades de I+D estratégicas en la convocatoria vigente del Programa de Investigación y Desarrollo para la Defensa (PIDDEF), por lo que su prosecución es política del Ministerio de Defensa de la Nación.

## 7 Conclusiones

P3D NEXT representa un salto en funcionalidad, performance, sostenibilidad y soberanía tecnológica para nuestros desarrollos, manteniendo las capacidades específicas del tipo de simulación que realizamos. Persigue requerimientos concretos de nuestros usuarios y satisface con creces las necesidades de realismo para entrenamiento del personal.

Será una plataforma eficiente y escalable, que permitirá la construcción de simuladores más completos y detallados. Además, al ser de código abierto, tendremos mayor flexibilidad para adaptar su uso a diferentes contextos, satisfacer requerimientos específicos de realismo, al tiempo que nos permitirá mantenerla actualizada y vigente.

## Referencias

1. Guaycochea, L., Luiso, J. and Del Rio Garcia, F.: Middleware P2P para la Sincronización de Eventos Discretos en una Simulación Distribuida de Sistemas que evolucionan en el tiempo. XVIII Congreso Argentino de Ciencias de la Computación, CACIC 2012. Bahía Blanca, Argentina, (2012)
2. Guaycochea, L., Abbate, H.: Error-Bounded Terrain Rendering Approach based on Geometry Clipmaps. IX Workshop Computación Gráfica, Imágenes y Visualización (WCGIV), XVII Congreso Argentino de Ciencias de la Computación (CACIC), 2011
3. Luiso, J., Guaycochea, L., Abbate, H.: Simuladores de entrenamiento distribuidos: Plataforma de desarrollo para ocultar los aspectos de la distribución. 1er Workshop Argentino sobre Videojuegos, WAVI 2010, 2010
4. Chandy KM, Misra J: Distributed Simulation: A Case Study in Design and Verification of Distributed Programs. IEEE Transactions on Software Engineering, Vol. SE-5, No. 5, September 1979.
5. Misra J.: Distributed Discrete Event Simulation. ACM Computing Surveys (CSUR), 1986.
6. Rosenfeld, M.: Un framework para comunicación Peer-to-peer en juegos multiusuarios. Tesis de Grado. Instituto de Informática, Universidad de La Plata, 2009.