

Primer obligatorio de CGA

21 de agosto de 2025

Introducción

El obligatorio consiste en desarrollar una aplicación completa de visualización de entornos 3D (*rendering*) en tiempo real utilizando OpenGL, con la que se deberá obtener el mayor realismo visual posible, optimizando el uso de los recursos disponibles.

Para lograr esto, se deberán seleccionar e implementar distintas técnicas de visualización y aceleración. Si bien dicha selección se realizará en función de lo que se quiera visualizar, para la aprobación del obligatorio, será necesario construir un conjunto mínimo de ellas.

El “mundo” visualizado deberá cumplir con los siguientes requerimientos:

- Estar poblado por un número arbitrario de objetos, configurable por el usuario.
- Tener una dimensión mucho mayor que la región visible.
- El observador no podrá atravesar el terreno ni los objetos de gran tamaño. En caso de que el mundo incluya agua, se permitirá atravesarla.

El mundo

Los estudiantes podrán presentar su propia propuesta del mundo a visualizar:

- Playa: realizar una visualización interactiva de una escena de playa que contenga agua, arena, cielo, personas, animales, sombrillas, olas, viento, etc.
- Realizar una simulación en tiempo real que logre transmitir la *sensación visual* de estar en la Facultad de Ingeniería. Para ello se podrá disponer de un modelo 3D del exterior, generado anteriormente por un módulo-taller (que puede ser descargado de <https://sketchfab.com/models/5161780b47a44b749bd2afdf596feb4df#download>).

También hay otro modelo provisto por la Facultad:

http://www.fing.edu.uy/inco/cursos/cga/Obligatorios/2022/Facultad_3D.obj

- Pueblo/Ciudad: una escena interactiva que contenga edificios, calles, personas, automóviles, plazas, semáforos, etc.
- Bosque: una visualización interactiva que contenga árboles, lagos, ríos, animales, cabañas, humo.
- también se aceptarán propuestas alternativas, siempre que sean discutidas y aprobadas previamente por los docentes.

En cualquiera de los mundos que implementen, debe haber personas (o animales, o drones), deambulando y realizando acciones simples predefinidas. La elección de cuáles efectos implementar queda a su voluntad e interés, pero el resultado final deberá contener *por lo menos* un elemento animado.

El usuario podrá caminar y volar por la simulación. Se podrá, además, interactuar de alguna forma con alguno de los objetos presentes. Por ejemplo, una opción podría ser controlar a alguno de los personajes.

Técnicas

El motor desarrollado debe incluir varias de las técnicas vistas en clase, aunque no tiene por qué (y se sugiere no hacerlo) restringirse a ellas.

Asimismo, se exige la utilización de *shaders*, junto con cualquier otra técnica que aproveche el poder de cómputo de las tarjetas gráficas. Podrán programar sus propios shaders, aunque se recomienda también utilizar shaders desarrollados por terceros.

Se debe poner especial atención en lograr una buena velocidad de rendering.

También es bueno tener en cuenta la importancia de la sensación de profundidad, por ejemplo:

- Los objetos más lejanos se ven menos nítidos (aunque no se requiere un efecto de profundidad de campo) y menos saturados (con una paleta más azulada).
- Otra clave de profundidad se da a nivel de la iluminación. Esto se puede dar por sombras provocadas por agrupamiento más denso de objetos, pasaje de nubes, desniveles geográficos, etc.
- La niebla es un recurso muy sencillo para acentuar el efecto de lejanía.

Entre las posibles técnicas a implementar, destacamos:

- técnicas de aceleración de *rendering*:
 - estructuras de datos espaciales: octrees, loose octrees, BSP culling: *occlusion*, portales, *shaft*, etc.
- nivel de detalle
 - Alpha LOD
 - LOD discreto
- tests de intersección
 - detección de colisiones
 - respuestas a las colisiones
- manejo de texturas
 - animación
 - *billboarding*
 - *alpha blending*
 - *environment mapping*
- manejo de técnicas poligonales
 - temas avanzados de texturas
 - iluminación por píxel, reflexión especular
 - *bump Mapping*, *emboss mapping*, *gloss mapping*
 - texturas proyectadas,
 - sombras
- sistemas de partículas

Otros temas posibles a desarrollar son:

- nurbs (p.ej. como en Quake III)
- horizontes de oclusión
- cinemática inversa
- luces dinámicas
- modelos físicos
 - *ragdoll physics*
- animación por cuadros clave (*keyframes*), posiblemente usando el formato md3
- animación en trayectorias con *splines*
- HDR: *high dynamic range rendering*
- *motion blur*
- iluminación dinámica
- iluminación estática
- etc.

Si bien para aprobar es necesario implementar algunas de estas técnicas, la decisión de cuáles funcionalidades se implementen deberá ser tomada en función del problema a resolver, es decir, de las características gráficas de la aplicación a desarrollar.

Sin embargo, se deberá implementar una técnica de nivel de detalle.

Proceso

El proceso deberá seguir los siguientes pasos:

1. Presentar el concepto de la aplicación a desarrollar. Se pueden mostrar imágenes de referencia del resultado visual deseado, tanto del mundo y sus escenas como de los efectos a implementar. Pueden ser ideas propias o basarse en ideas extraídas de demos o juegos, o vídeos ya existentes, en cuyo caso se deberá mencionar el origen.
2. Reuniones para contestar dudas, mostrar avances, etc.
3. Entrega final.

Reutilización de software, modelos, ideas, etc.

Se deberá documentar **todas** las fuentes de información extra-curso. Además, **se deberá compartir con los otros grupos** (en los foros del EVA) toda fuente o recurso encontrado que se considere de utilidad.

Dado que la finalidad del obligatorio no es la de aprender a realizar buenos modelos 3D, es posible y recomendable que utilicen modelos construidos por terceros.

Evaluación

Se considerarán especialmente las siguientes características del software entregado:

- resultado general (calidad, rendimiento, etc.).
- arquitectura de la aplicación.
- técnicas y algoritmos empleados.

- resultados globales obtenidos (calidad perceptual lograda, como grado de realismo, coherencia visual y fluidez de simulación).
- cantidad y complejidad de las técnicas implementadas.
- prolijidad y compleción de la documentación.

Documentación

Se deberá documentar el sistema entregado, incluyendo el problema propuesto y su solución, justificando esta última.

Se deberá documentar la arquitectura del motor y de la aplicación, así como posibles mejoras.

También se deberá incluir un análisis cuantitativo básico de rendimiento en el que se detallen los tiempos insumidos, cuellos de botella, ideas de cómo mejorar, etc.

Se pretende una documentación concisa que incluya los siguientes ítems.

- Problema a resolver (con imágenes de ejemplo de lo que se pretende lograr).
- Solución creada (arquitectura básica).
- Técnicas implementadas.
- Análisis de resultado (comparación con las imágenes de ejemplo y análisis de performance).
- propuestas concretas de mejora, como optimización de algoritmos, incorporación de nuevas técnicas de rendering o ampliación del repertorio de animaciones.

Utilicen esta lista como un *checklist* verificando que todo esté presente en la documentación.

Plazos de la entrega de ideas, consultas y entrega final

Al ser esta una evaluación eliminatoria, contarán con tres tipos de instancias.

Fechas:

- Reuniones de consulta: jueves 28/8, 4/9, 11/9
- Estado de avance del Obligatorio: Jueves 18/9.
- ...
- Pre-entrega jueves 27/11
- Entrega final: jueves 4/12 con una presentación preparada y video.