

Proyecto Game Engine - Hito 1

Introducción

El propósito de este hito es concretar el proyecto a realizar y a la vez, generar un ejecutable ordenado que nos sirva de base para desarrollar el game engine y demos que se irán concretando en los hitos posteriores. Este primer acercamiento tiene barreras de entrada como familiarización con CMake, C++, Git, GLFW/OpenGL, modelos 2D/3D, u otros sistemas alternativos que usted opte por utilizar. Posterior a esta entrega puede cambiar el motor de rendering.

La evaluación de este hito se realizará utilizando la rúbrica de la página siguiente. Deberá adjuntar en su entrega una autoevaluación que le permita al equipo docente comprender rápidamente el trabajo que usted ha realizado.

Importante:

- La nota final se calcula sumando los puntajes obtenidos a un punto base.
- El puntaje de cada ítem de la rúbrica será evaluado observando el código fuente y analizando un demo funcional.
- Para acceder a una columna de puntaje, se debe cumplir con todo lo pedido en dicha celda, y en todas las celdas anteriores que otorgan menor puntaje.
- En caso de que alguna categoría no alcance un puntaje mínimo de 0.5, no se hace efectivo ningún puntaje de la columna bonus.
- Si supera el umbral de nota 7, el puntaje restante se guarda como puntaje extra para el hito 2.
- Si no hay un correcto uso de control de versiones, implica nota 1 en el hito.

Video:

- Debe adjuntar a su entrega, un video donde se ilustre el proceso desde la descarga de los archivos del repositorio, compilación y ejecución de demo funcional.
- Este ítem es especialmente importante si utiliza un sistema operativo distinto de Windows o Linux, y un lenguaje de programación distinto de C++.
- Edite el video para que en ningún caso tarde más de 5 minutos, pero mientras más breve, mejor.
- Adjunte un link a Youtube o suba el video a u-cursos en su entrega (u-cursos tiene restricción de tamaño de archivo).

Autoevaluación:

- Con el fin de facilitar la evaluación, y a la vez dejar en claro lo que usted ha implementado y lo que no, se solicita que complete la rúbrica anterior, indicando en cada celda lo que implementó para obtener dicho puntaje, y el archivo/función/clase donde habría que buscarlo. En material docente encontrará un archivo .odt con la rúbrica para completar su autoevaluación.

Entregables:

- u-cursos:
 - Autoevaluación en pdf.
 - Ficha de inscripción de proyecto en pdf.
 - Comentario de entrega:
 - commit de su repositorio remoto git que será evaluado.
 - Link de Youtube a video de descarga/compilación/ejecución, puede ser privado.
 - Alternativamente al link de Youtube, puede adjuntar el video a su entrega si es que u-cursos lo permite dada su restricción para archivos muy grandes.
- Repositorio Git:
 - Archivo Readme.md
 - Licencia open source (comunicar vía e-mail al profesor de cátedra si es que desea que su código no sea open source).
 - Código fuente del game engine.
 - Código fuente del demo.

Presentación

- Debe asistir a la sesión de presentaciones en la fecha y hora indicada en la última versión de la planificación del curso.
- Si por razones de fuerza mayor no puede asistir, comuníquelo vía e-mail al profesor lo antes posible. Cuando sea la fecha, deberá enviar su presentación vía link a video privado de Youtube.
- La presentación se evalúa simplemente de manera Logrado/No logrado, lo importante es que explique y presente muy brevemente sus avances desde el hito anterior.

Rúbrica Hito 1 – CC5512

Criterio \ Puntaje	0	0.5	1.0	1.5	Bonus
Ficha de inscripción de proyecto	No se entrega o no se implementa	Ficha de inscripción de proyecto enviada vía Tareas/u-cursos y publicada adicionalmente en Material de Alumnos/u-cursos. Se especifica claramente tipo de juegos al que apunta el game engine a implementar con al menos 2 videos de ejemplo.	Se concreta correctamente el tipo de cámara y mecanismo de control del videojuego.	Se especifica lenguaje y plataforma	
Repositorio Git		Repositorio Git remoto en Github (u otro) creado. Se incluye licencia (idealmente de código abierto).	Archivo Readme.md del repositorio incluye al menos: - Breve descripción del proyecto. - Instrucciones claras y breves de compilación y ejecución.	Desarrollo se realiza ordenadamente en una rama específica (ejemplo “dev”), y la entrega para el hito actual se concreta en una rama o tag de nombre “hito 1”.	
Organización de archivos		Todo el código se encuentra en un único archivo que se compila y funciona.	El código se distribuye organizadamente en distintas carpetas y archivos en una estructura mínima pero coherente.	Código genera una biblioteca estática o dinámica donde se encuentren todas las funcionalidades, y un código “main” que utiliza estas funciones ejerciéndolas en un ejecutable. (El propósito aquí es que nuestro engine sea una lib y que se puedan generar distintos proyectos de juego enlazándose a ella sin modificarla).	
Demo funcional		Demo solo imprime texto en la consola	Demo abre una ventana y dibuja al menos un par de objetos estáticos en ella (2D o 3D) (+0.2 si es 3D).	Los objetos de la escena se mueven siguiendo alguna trayectoria o movimiento repetitivo.	(+0.5) La visual presentada se parece a un videojuego del tipo que se quiere. Esto requiere el uso de al menos unas 5 figuras y/o texturas con sentidos diferentes.