

Rúbrica Proyecto Game Engine (v2) – CC5512

Criterio \ Puntaje	0	0.3	0.6	0.9	Bonus
Documentación	No se implementa funcionalidad	Se presenta un pequeño tutorial de uso como archivo markdown, en una pequeña wiki o en un informe pdf. Este tutorial debe estar presente en el repositorio del proyecto.	Se presenta documentación de la API en doxygen o similar. Proporcionando, por ejemplo una serie de archivos html que permitan una fácil navegación por el código.		<p>(+0.2 c/u, máximo 1.0 en total)</p> <p>Cada bonus implementado en los milestones 1 o 2 puede ser contabilizado nuevamente como bonus en esta entrega final, solo que esta vez otorgan 0.2 por cada 0.5 puntos previamente obtenidos.</p> <p>Del mismo modo, también es posible implementar bonus de los ya propuestos en las rúbricas anteriores que no hayan sido implementados. Estos también otorgarán 0.2 puntos por cada 0.5 de los indicados previamente. Este puntaje también se suma al mismo máximo de 1.0 como bonus de proyecto final.</p>
Game Engine API		Capa de software delgada sobre librerías de bajo nivel. Se permite aún el acceso a librerías de bajo nivel.	<p>Capa de software completamente separada de librerías y funcionalidades básicas. Esta API permite elaboradas funcionalidades.</p> <p>Sistema de inicialización/actualización y apagado del game engine. Que al mismo tiempo permita agregar, modificar y eliminar game objects.</p>	Engine API presenta funcionalidades claves que simplifican el desarrollo de videojuegos del tipo al que apunta. Ejemplos: generar personaje aleatorio, barajar mazo, construir y manejar grafo de desarrollo de tecnologías, controlar el clima, control sofisticado de cámara, etc.	
Game Object Model		<p>Mecanismo razonablemente bueno para manejar referencias a game objects.</p> <p>Modelo de game objects implementado permite añadir y eliminar game objects/componentes offline según se requiera.</p>	<p>Modelo de game objects implementado permite añadir y eliminar game objects/componentes en runtime según se requiera.</p> <p>La actualización de los distintos game objects y subsistemas es apropiada y consistente.</p>		
Manejo de Tiempo		Gobernar la tasa de refresco y/o blanking. No hay problemas al pausar o debuggear el juego. Marcador de fps disponible (título de ventana o texto)			
Dispositivos de Interacción Humana		<p>Se implementa uso de mouse+teclado ó joystick en un mínimo funcional suficiente para una aplicación tipo.</p> <p>Se implementa capa de abstracción de HID.</p> <p>Se manejan correctamente conexiones y desconexiones de dispositivos. Notificando al usuario del problema.</p>	<p>Se implementa al menos una de las siguientes funcionalidades:</p> <ul style="list-style-type: none"> - Player remapping - Smooth control - Chords - Sequences 		
Scripting		Se implementa un sistema de scripting nativo (mismo lenguaje del engine) utilizando funciones virtuales.	<p>Se implementa un sistema de scripting (sugerencia: Lua) que permita un amplio acceso a funcionalidades del engine e interconexiones con el mismo. En la práctica, implementan al menos 2 de las siguientes:</p> <ul style="list-style-type: none"> - Máquina de estado finita - Callbacks desde scripts a código nativo - Callbacks desde código nativo a scripts - Sistema de eventos simple - Creación/modificación de Game Objects. 		
Rendering Engine		<p>Implementa al menos uno de los siguientes sistemas:</p> <ul style="list-style-type: none"> - Sistema de manejo de sprites - Sistema de manejo de modelos 3D - Sistema de renderizado de texto 	<p>Al menos un sistema de optimización para rendering eficiente de una cantidad masiva de objetos en la escena. Ejemplos</p> <ul style="list-style-type: none"> - Clipping - Sistema de portales - Sistema de Antiportales - Grafo de escena para determinación de visibilidad. 	<p>Funcionalidades disponibles de alto nivel requeridas para el tipo de videojuego planificado. Ejemplos: dibujarCarta, dibujarPersonaje, dibujarMapa, etc.</p> <p>Performance apropiado para mantener al menos 30 fps en hardware del equipo docente.</p>	
Audio Engine			<p>Reproducción de audio 2D</p> <p>Manejo de múltiples fuentes sonoras estáticas</p>	<p>Reproducción de audio 3D</p> <p>Manejo de múltiples fuentes sonoras estáticas y dinámicas</p>	

Criterio \ Puntaje	0	0.3	0.6	0.9	Bonus
Animation Engine (*)	No se implementa funcionalidad	Funcionalidades de LERP y SLERP para animaciones simples. Ó Sistema de animación por sprites. Ó Sistema de seguimiento de splines.	Sistema de animación con esqueletos.	Sistema de animación funciona con skinning.	(+0.2) Lectura y utilización de animaciones desde archivos en algún formato de la industria. Ejemplos: glTF (recomendado), fbx, collada. (+0.4) Técnica de blending de animaciones. Ejemplo: transición entre caminar y correr. (+0.2) Sistema de animación por vértices. Ejemplo: morphing.
Physics/Collisions Engine (*)		Detección de colisiones entre volúmenes alineados con los ejes (AABBox) y/o basados en distancias.	Uso de sistema sofisticado para el manejo y detección de colisiones. EJ: vía funciones callbacks cuando se ejecute la colisión. Ó Mecánica simple basada en física que involucre física lineal y rotacional. (+0.1) Sistema de física es 3D.	Se permiten abundantes interacciones basadas en física entre los distintos game objects con esta característica habilitada. (+0.2) Sistema de física es 3D.	(+0.2) Manejo de restricciones para distintos objetos de mundo físico. Ej: Point to point, stiff springs, planar, wheel, pulley. (al menos 2) (+0.4) Sistema de ragdolls (+0.1) Implementa Ray casting (+0.1) Implementa Shape casting (+0.1) Implementa collision filtering
AI Engine – Path Planning (*)			A* Search sobre una malla de navegación proporcionada por el usuario del engine. Ej: malla regular de quads, irregular de triángulo, etc. Usted puede especificar el tipo de malla que soporte su motor. El usuario debe entregarla en un formato estándar definido por usted, ejemplos: OFF, OBJ.	Estrategia de Probabilistic Roadmap Planner para que game objects se muevan por el mapa. ó Estrategia de Probabilistic Roadmap Planner para que game objects articulen sus distintos grados de libertad para moverse en un ambiente complejo. (Ej: brazo robótico)	(+0.2) Sistema de Maquinas de estado (+0.2) Sistema de Decision Tree (+0.4) Generador de navmesh automático (Ej: Recast)
AI Engine – Behavior Trees (*)			Estrategia de Behavior Trees y un ejemplo simple de aplicación	Estrategia de Behavior Trees y un ejemplo adicional de aplicación (i.e. al menos 2 en total) de mayor complejidad que considere módulos de otros Behavior Trees	

Demo

Criterio \ Factor de éxito	0	0.7	0.8	1.0	1.1
Demo prototipo de videojuego	No hay demo.	No se utiliza uno o más de los módulos requeridos obligatoriamente. Ó No se implementa ninguno de los módulos optativos (marcados con *).	Se utiliza efectivamente toda la API provista por el motor de juegos implementado.	Se implementan dos mecánicas básica del tipo de juego comprometido para el engine.	Prototipo es funcional según el tipo de juego comprometido y permite jugar tranquilamente por al menos 4 minutos.

Importante:

- Si no hay un correcto uso de control de versiones implica nota 1 en el proyecto.
- El proyecto game engine debe corresponder al inscrito previamente. Nota 1 si esto no se cumple.
- Se deben mantener las implementaciones ya comprometidas en los milestone 1 y 2. Si por distintas razones se produjo un cambio relevante o se debió eliminar una funcionalidad. Adjunte en el Readme.md de su repositorio un párrafo justificando esta decisión.
- 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.
- El puntaje se otorga cuando se implemente una API/framework que permita una conexión fácil y directa con el resto del motor. El usuario de su engine NO DEBE tener conocimiento de las librerías de bajo nivel que pueda estar utilizando. Ejemplo: OGRE, OpenAL. Esto es, si proporciona una buena capa de abstracción a librerías de terceros, recibirá el mismo puntaje completo que si implementa los algoritmos de bajo nivel por su cuenta.
- En el Readme.md de su repositorio, incluya una sección clara de como poder compilar su entregable de milestone, detallando plataforma y estrictamente todo lo requerido para que funcione en ella.
- Adicionalmente, debe adjuntar a su entrega de u-cursos, un video donde se ilustre el proceso desde la descarga de los archivos del repositorio, compilación y ejecución de demo funcional. Este item es especialmente importante si utiliza un sistema operativo distinto de Windows o Linux.

Sobre la nota:

- En caso de que alguna categoría no alcance un puntaje mínimo de 0.3, no se hace efectivo ningún puntaje de la columna bonus.
- La nota final se calcula sumando los puntajes obtenidos, multiplicando este resultado por el factor de éxito del demo, y luego sumando el punto base.
- Si supera el umbral de nota 7 para esta entrega, el puntaje restante, con un máximo de 0.5 puntos total, se utilizará como bonificación extra en los milestones 1 o 2.
- (*) Usted puede elegir un sistema entre: Animation Engine, Physics/Collisions Engine, AI Engine (Path planning) y AI Engine (Behavior Tree). Se recomienda elegir el más apropiado para el tipo de videojuegos al que apunta su engine. Si implementa varios de los sistemas electivos, simplemente se suma el puntaje al total.

Demo funcional:

- El puntaje de cada ítem de la rúbrica será evaluado observando el código fuente y analizando un ÚNICO demo funcional de videojuego donden todos los criterios.
- Video de descarga/compilación/ejecución de su engine+demo funcional. 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.

Entrega por u-cursos:

- Autoevaluación en pdf.
- Comentario de entrega: commit de su repositorio remoto git que será evaluado.