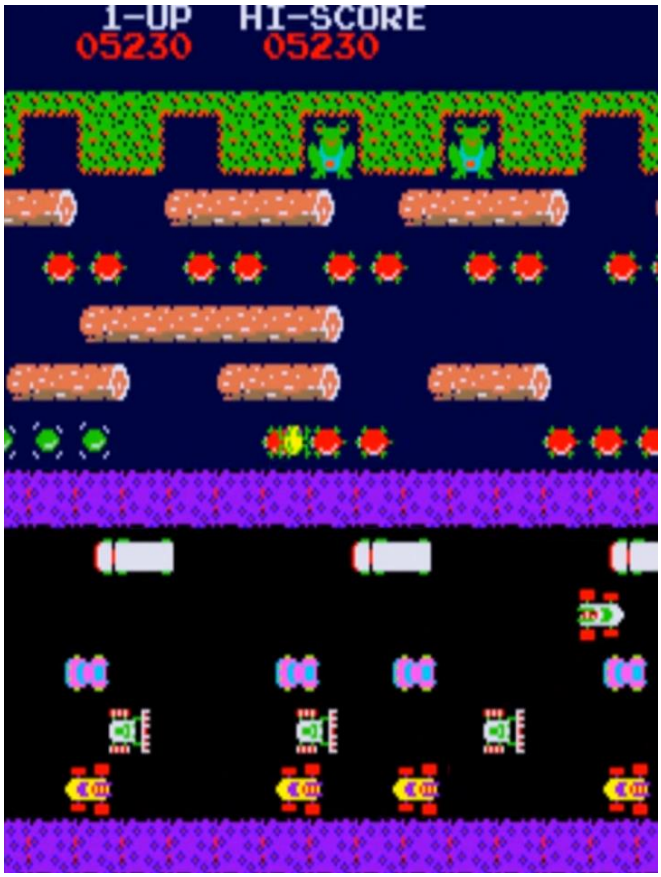
Tarea 2: Frogger en un Game Engine

Introducción

En esta tarea usted debe implementar el “mismo” juego simplificado Frogger que implementó en la tarea 1, solo que esta vez, debe implementarlo haciendo uso de un Game Engine y su lenguaje de scripting junto con el modelo de Game Objects disponible. También deberá añadir soporte tanto para teclado como para un joystick.



https://www.youtube.com/watch?v=l9fO-YuWPSk

Dispone de 3 opciones, debe elegir solo una de ellas: Godot (Zen), Unreal Engine (Madness) u Open 3D Engine (Berserker).

Su proyecto debe ser capaz de generar un ejecutable con el videojuego para el sistema operativo Windows. En el video se deberá apreciar este proceso.

Diving Deep

La invitación es a explorar un Game Engine con el que no se encuentre familiarizado.

1. Godot 4.0.2 - https://godotengine.org/ - Scripting: GDScript
    - Esta es la opción que requiere menos recursos computacionales, si su computador es antiguo o no posee tarjeta de video dedicada, lo recomendable sería tomar esta opción, en cualquier caso.
    - Este motor es simple y bueno para 2D, pero para 3D no posee muchas funcionalidades, por lo que en general no se utiliza para videojuegos AAA.

2. Unreal Engine 5.1.1 - https://www.unrealengine.com/en-US/ - Scripting: Blueprints
    - Unreal es famoso por ser altamente demandante en recursos computacionales, procesador, ram y espacio en disco. Utilice esta opción solo si dispone de un computador suficiente (revise usted los requerimientos mínimos).
    - Unreal es ampliamente utilizado en la industria AAA, y disponer de estos conocimientos es un plus importante en el CV.
    - En esta opción debe configurar su propio Pawn y asignarle control de entradas vía el sistema de EnhancedInput. Es decir, NO debe involucrarse con modelos articulados y animaciones (Por lo que no debe utilizar un ACharacter).
3. Open 3D Engine 23.05.0 - https://www.o3de.org/ - Scripting: Lua o Script Canvas
    - Este motor es bastante nuevo y famoso por la inestabilidad de las versiones anteriores, pero probablemente sea el futuro. Es open source y está a cargo de la Linux Fundation. Está pensado para videojuegos AAA y, al parecer, tiene un mejor diseño base que Unreal (Basta con revisar la clase Entity y compararla con Actor).
    - Opción recomendada por el profesor.
    - De igual forma que para la opción de Unreal, usted NO debe involucrarse con modelos articulados ni animaciones.

En esta tarea, el objetivo es la comprensión básica del funcionamiento de un Game Engine existente.

Simplificaciones

- Para las representaciones gráficas puede utilizar rectángulos, cajas 3D, modelos provistos por el engine o mallas estáticas simples importadas. Materiales pueden ser colores simples o texturas.
- Para los sonidos, deberá puede componer sonidos simples o conseguir sonidos libres de internet.
- Note que para esta tarea usted no necesita compilar el Game Engine elegido, puede instalarlo directamente.
- Los engines de las opciones han sido implementados en C++ y poseen mecanismos para trabajar en C++ desarrollando el mismo videojuego. En esta tarea no es requisito trabajar en C++, y se recomienda no hacerlo en este punto de la historia. Sí será requerimiento para la tarea 3.

Entregables

- Archivo adjunto: Pdf con la autoevaluación
    - Comente en una tabla como la mostrada en la sección “puntajes”, que ítems le faltaron y el puntaje que usted cree que tendría. Cada categoría tendrá 0, 0.5 o 1.0 según completitud.
    - Si no adjunta autoevaluación, no tendrá derecho a reclamo.
- Enlaces en la descripción de la entrega:
    - Link a repositorio GitHub **privado**, e indicación de commit o tag de entrega.
        - Su repositorio debe estar correctamente configurado para que se ignoren todos los archivos generados por el Game Engine, solo debe considerar los archivos “fuente”. Utilice git-lfs para los archivos binarios (como los uassets de unreal).
        - Deberá invitar como colaboradores a todo el equipo docente. Los usuarios GitHub del equipo docente se encuentran en u-cursos/Novedades.
    - Video (o link a video) presentando, el proceso de “abrir” el proyecto en el Engine, los scripts implementados, y como genera un executable Windows del videojuego.
        - Recuerde que u-cursos posee un límite al tamaño máximo de archivos

Importante

- Los Game Engines mencionados poseen un abanico enorme de funcionalidades. En esta tarea se reducen las funcionalidades. Por ejemplo, no se ponga a investigar sobre sistemas de animación, o de inteligencia artificial, o de visualizaciones con shaders sofisticados. Este es un ejercicio básico e introductorio de uso de un Game Engine.
- La investigación y familiarización con el Game Engine que elija es parte íntegra del trabajo para esta tarea. Esto demanda bastante tiempo y requiere dominio razonable de inglés. El equipo docente proveerá lineamientos básicos, pero es responsabilidad del estudiante el estudio necesario de la documentación oficial.
- Existen muchos video-tutoriales en youtube que son muy buenos, pero existen, también muchos, que son malos o muy malos. En general, usted no necesita que le enseñen a programar, sino simplemente a utilizar el engine. Sea cauteloso con los videos y elija los correctos, descarte los videos erróneos rápidamente.
    - Si encuentra buenas fuentes de información, favor compártalas en el foro del curso.
- El código de su tarea no debe ser compartido de ninguna forma durante el periodo de desarrollo de esta tarea. Por ejemplo, **no debe almacenar su código en un repositorio GitHub público**. Siempre puede debatir ideas y algoritmos, pero no puede compartir código.
- Tome decisiones razonables para obtener un prototipo funcional, el enunciado no posee letra chica. En cada categoría de puntaje se otorgará 0, 0.5 o 1.0, según criterio de completitud.

Flash Forward

- En la tarea 3 se pedirá implementar una extensión para un motor de videojuegos, idealmente deberá extender el mismo motor que utilice en esta tarea. Es decir,
    - En el caso de Godot, deberá implementar un módulo
    - En el caso de Unreal, deberá implementar un Plugin
    - En el caso de O3DE, deberá implementar un Gem

Puntajes

	Software	**Puntaje**
P1	Entradas de usuario a través de teclado y joystick, controlando a Frogger via el input manager del engine.	1.0
P2	Movimiento automático de autos y/o troncos, algunos hacia la derecha y otros hacia la izquierda ordenados en pistas.	1.0
P3	Condiciones de inicio y término de juego (ganar y perder, detectar colisiones)	1.0
P4	Organización del proyecto en varios archivos, separando distintos Game Objects, Components y Scripts.	1.0
P5	Gráficos y sonidos.	1.0

	Video	**Puntaje**
P6	Generación de videojuego ejecutable o “packaging” (ilustrado en el mismo video)	0.5
P7	Debe presentar los distintos GameObjects, Componentes y Scripts implementados, y como estos actúan sobre el gameplay. El equipo docente seguirá, paso a paso este video para revisar su tarea, y comprobar que todo funcione correctamente. Edite el video para que en ningún caso exceda los 5 minutos, pero mientras más breve, mejor.	0.5