

Programación de videojuegos 2

Sistema de Desafíos 2024

TECNICATURA EN DISEÑO Y PROGRAMACIÓN DE VIDEOJUEGOS • FICH-UNL

Profesor: Gastón Caminiti

Bienvenido/a al escrito donde te presentamos algunas recomendaciones para sacarle el máximo beneficio a la experiencia de cursado.



Un proyecto para practicarlo todo

Programación de Videojuegos II es la materia número diez (10) del plan de estudio. Es decir, estas a cinco (5) materias de finalizar la tecnicatura ¡Felicidades!

Pensando en lo anterior, creamos la propuesta práctica (que llamamos **desafíos**) con el objetivo de que desarrolles durante la cursada un videojuego 2D para sumar a tu portafolio.

A continuación, encontrarás una serie de recomendaciones para empezar a trabajar en tu proyecto, sumado a la ficha de cada **desafío**, la cual especifica las tareas mínimas a realizar por cada prototipo entregable.



Tip 1: No sobredimensionar

Al contar con un poco más de dos (2) meses de cursado, es prioritario planificar un videojuego 2D a producir en dicho plazo. Si bien, emplear motores de videojuegos, y particularmente Unity, permite acotar los tiempos de desarrollo, también precisas espacio para estudiar el motor y nuevas técnicas de programación.

Te recomendamos iniciar el diseño del proyecto personal teniendo en cuenta el siguiente criterio:

***Crear un videojuego 2D, para un solo jugador, del género
plataforma, casual o aventura.***

Tip 2: Evaluación = Playtest

El sistema de *desafíos* se creó sabiendo que, en desarrollo de videojuegos, el [Playtest](#) es una etapa recurrente donde se evalúa el [gameplay](#) producido por el equipo de trabajo. Por eso los criterios de evaluación de los *desafíos* (ver Mapa de Cursado) son indicadores de calidad del producto, no exclusivamente de la programación.

Claro que aprender a codificar mecánicas y sistemas de juego complejos es el objetivo central de nuestra materia, pero buscamos alentarte a pensar el código fuente como una pieza más del desarrollo de nuevas experiencias de entretenimiento, el cual debe integrarse con otras características del proyecto, como ocurre en el escenario profesional.

Tip 3: Reconocer los antipatrones

Un [antipatrón](#) es una forma de trabajar a la que consideramos positiva, pero el resultado inmediato o futuro es contrario. Existen diferentes Antipatrones en desarrollo de videojuegos, pero durante la cursada buscaremos evitar los siguientes:

- [Optimización prematura](#): Los esfuerzos de trabajo deben centrarse en la programación del gameplay. La optimización es una tarea en respuesta al resultado de una [prueba de rendimiento](#) o cubierta por determinados patrones (como veremos en el *nivel* 7).
- [Código espagueti](#): Se busca escribir un código legible. Incluso el motor Unity tiene su propia [guía de estilo para C#](#)

- [Reinventar la rueda](#): En Programación de Videojuegos II puedes emplear recursos de terceros para crear el videojuego: imágenes, sonidos, música, partículas y otros recursos de la [tienda de Unity](#), siempre y cuando su incorporación no infrinja derechos de autor.

Tip 4: Explorar

A medida que avanzamos en las unidades, particularmente del **nivel** 4 en adelante, puede que ciertos ejemplos no se relacionen con el gameplay que estás creando. Los Tipos de Dato Abstracto (TDA), Eventos y Patrones de diseño son contenidos que pueden estudiarse de forma general, pero su implementación varía según el proyecto.

Aquí te recomendamos que explores diferentes formas de incorporarlos, porque estos elementos y recursos pueden determinar el [rendimiento](#), [escalabilidad](#) y [mantenimiento](#) de un videojuego comercial.

Tip 5: Publicar

Te presentamos este trayecto como una oportunidad para dar a conocer tu trabajo. Construir un portafolio es una excelente idea para visualizar tu progreso en el proyecto y en los aprendizajes (y es una herramienta de innegable potencial para la inserción laboral). También podés sumarte a publicaciones periódicas en redes sociales como [#screenshotsaturday](#).

Por último, encontrarás en el aula virtual contenido sobre [control de versiones](#) con Unity. *Esta herramienta* te simplificará la forma de desarrollar y distribuir los desafíos. Además, es una buena idea empezar a usar herramientas como [Git](#) por su adopción en el desarrollo profesional de videojuegos, siendo prácticamente un requisito para conseguir empleo.

Ficha del Desafío 0

Nombre
Puesta en marcha
Objetivo: Preparar nuestro entorno de desarrollo
Tareas: <ul style="list-style-type: none"> • Instalar Unity HUD y la última versión LTS del editor de Unity. • Instalar GitHub Desktop • Crear en GitHub el repositorio donde trabajaremos durante el cursado.

Ficha del Desafío 1

Nombre	1° Fecha de Entrega	2° Fecha de Entrega
Personaje y Movimiento	23/08/24	31/08/24
Objetivo: Crear movimiento 2D interactivo y pre programado		
Tareas: <ul style="list-style-type: none"> • Programar movimiento interactivo del jugador. • Codificar movimiento de un enemigo o elemento del nivel. • Gestionar colisiones y triggers con objetos estáticos y dinámicos. 		
Entregables: <ul style="list-style-type: none"> • 1° Prototipo del videojuego 2D exportado para Windows o publicado con WebGL. • Enlace al repositorio del Proyecto de Unity correspondiente al 1° prototipo. 		

Ficha del Desafío 2

Nombre	1° Fecha de Entrega	2° Fecha de Entrega
Animaciones y Efectos	03/09/24	11/09/24
Objetivo: Incorporar efectos visuales y de sonido al primer prototipo.		
Tareas: <ul style="list-style-type: none"> • Animar el personaje principal. • Codificar respuestas de audio para las interacciones del jugador. • Emplear sistema de partículas para ambientar el escenario o programar una mecánica. 		
Entregables: <ul style="list-style-type: none"> • 2° Prototipo del videojuego 2D exportado para Windows o publicado con WebGL. • Enlace al repositorio del Proyecto de Unity correspondiente al 2° prototipo. 		

Ficha del Desafío 3

Nombre	1° Fecha de Entrega	2° Fecha de Entrega
Nivel 1	16/09/24	24/09/24
Objetivo: Crear el primer nivel del videojuego 2D.		
Tareas: <ul style="list-style-type: none"> • Programar condición de victoria y de derrota del nivel. • Crear el mapa con Tilemap. • Animar elementos del nivel con Timeline. 		
Entregables: <ul style="list-style-type: none"> • 3° Prototipo del videojuego 2D exportado para Windows o publicado con WebGL. • Enlace al repositorio del Proyecto de Unity correspondiente al 3° prototipo. 		

Ficha del Desafío 4

Nombre	1° Fecha de Entrega	2° Fecha de Entrega
Generadores	28/09/24	06/10/24
Objetivo: Incorporar generación aleatoria en el nivel 1.		
Tareas: <ul style="list-style-type: none"> • Codificar la invocación aleatoria de objetos en el escenario. • Programar la ejecución de situaciones de juego transcurrido cierto tiempo. • Garantizar que los objetos a generar y el momento sea configurable desde el editor. 		
Entregables: <ul style="list-style-type: none"> • 4° Prototipo del videojuego 2D exportado para Windows o publicado con WebGL. • Enlace al repositorio del Proyecto de Unity correspondiente al 4° prototipo. 		

Ficha del Desafío 5

Nombre	1° Fecha de Entrega	2° Fecha de Entrega
Progresión	09/10/24	17/10/24
Objetivo: Desarrollar un sistema de progresión con tipos de datos abstractos		
Tareas: <ul style="list-style-type: none"> • Codificar el sistema de progresión. • Emplear Scriptable Object para extender el sistema de progresión. • Garantizar que los componentes del sistema sean configurables desde el editor. 		
Entregables: <ul style="list-style-type: none"> • 5° Prototipo del videojuego 2D exportado para Windows o publicado con WebGL. • Enlace al repositorio del Proyecto de Unity correspondiente al 5° prototipo. 		

Ficha del Desafío 6

Nombre	1° Fecha de Entrega	2° Fecha de Entrega
Eventos	21/10/24	29/10/24
Objetivo: Incorporar interfaces al videojuego 2D		
Tareas: <ul style="list-style-type: none"> • Crear la interfaz de usuario HUD. • Crear la interfaz de usuario Menú Principal. • Codificar el funcionamiento y actualización de la interfaz de usuario con eventos. 		
Entregables: <ul style="list-style-type: none"> • 6° Prototipo del videojuego 2D exportado para Windows o publicado con WebGL. • Enlace al repositorio del Proyecto de Unity correspondiente al 6° prototipo. 		

Ficha del Desafío 7

Nombre	1° Fecha de Entrega	2° Fecha de Entrega
Optimización	30/10/24	06/11/24
Objetivo: Aplicar técnicas de optimización al videojuego 2D		
Tareas: <ul style="list-style-type: none"> • Codificar el <i>Object Pool</i> para un caso de generación aleatoria. • Refactorizar el comportamiento de un GameObject con el principio SOLID. 		
Entregables: <ul style="list-style-type: none"> • 7° Prototipo del videojuego 2D exportado para Windows o publicado con WebGL. • Enlace al repositorio del Proyecto de Unity correspondiente al 7° prototipo. 		

Ficha del Desafío Final

Nombre	1° Fecha de Entrega	2° Fecha de Entrega
Producto Viable Mínimo	08/11/24	13/11/2024
Objetivo: Publicar un prototipo con el gameplay base del videojuego 2D		
Tareas: <ul style="list-style-type: none"> • Programar el <i>singleton</i> para conservar información entre niveles. • Codificar la persistencia de datos de progresión. • Garantizar el funcionamiento integral del prototipo (Ajustando o integrando mecánicas). • Publicar el videojuego 2D en Itch.io o Unity Play. 		
Entregables: <ul style="list-style-type: none"> • 8° Prototipo del videojuego 2D exportado para Windows o publicado con WebGL. • Enlace al repositorio del Proyecto de Unity correspondiente al 8° prototipo. • Enlace de página de la publicación del proyecto. 		

Progresión en el Sistema de Desafíos

La siguiente tabla señala los estados que podés adquirir en Programación de Videojuego II mientras vas progresando con la entrega de los prototipos:

Requisito	Estado
Aprobar Desafíos del 1 al 5	<i>Regular</i>
Aprobar Desafíos del 1 al 7	<i>Promoción con nota de 7(siete)</i>
Aprobar todos los Desafíos propuestos	<i>Promoción con una nota de hasta 10(diez)</i>

■ ■ ■