

# Encargo N°4: Debug Mini Supermercado

## **MODALIDAD:**

Trabajo individual con nota en clases.

## **CONTENIDO A EVALUAR:**

Contenido de Programación II.

## **PONDERACIÓN:**

25% de la Nota de Presentación.

## **DESCRIPCIÓN GENERAL:**

Durante la clase, deberán arreglar el proyecto de Unity entregado. Dicho proyecto contiene errores y fallas comunes que ya deberían haber enfrentado en los trabajos del semestre.

El ejercicio está pensado para que se arregle y se reconstruya lo que ya se tiene (Es más fácil y rápido que construir todo desde 0)

Además, se presentará una rúbrica que corresponde al estado final en el que debería quedar el prototipo, es decir, una build con todo arreglado y funcional.

El resultado final es el mismo del Encargo N°3. Una experiencia en un supermercado en dónde se puedan agarrar y soltar objetos, ver su información, dejarlos, sacarlos de un inventario, efectuar la compra con una boleta final y realizar algunas configuraciones sonoras. La diferencia es que esta vez deberán arreglar un proyecto ya hecho.

*(Solo está permitido realizar una cantidad de preguntas correspondientes a las acumuladas durante el semestre)*

## **PROHIBICIONES:**

1. Hablar entre compañeros, copiarse.
2. Usar teléfono, internet, códigos antiguos, pendrives, apuntes y/o similares.

### **RÚBRICA:**

<b>10%</b>	Se cuenta con un dinero inicial visible en la UI	Se presenta un sistema de movimiento y rotación del jugador y la cámara funcional	-
<b>30%</b>	El jugador cuenta con un sistema de agarrar y soltar objetos funcional	Solo cuando se apunta a un objeto, se despliega la información de su nombre y precio	El cuenta con efectos de sonidos paga agarrar objetos, soltar objetos, compra correcta, compra fallida, música de fondo y sus respectivas configuración de audio funcional
<b>60%</b>	Al momento de pagar, se generará una boleta con el listado de productos comprados con su nombre y precio, el valor total de la compra, el dinero con el que se pagó y el vuelto correspondiente. Adicionalmente, la compra deberá fallar si no tenemos dinero suficiente		

### **ORDEN RECOMENDADO:**

1. Clonar proyecto con GitHub Desktop mediante URL:  
[https://github.com/Timigatari/Debug\\_Programacion\\_II\\_Roto](https://github.com/Timigatari/Debug_Programacion_II_Roto)
2. Fetch en GitHub Desktop para asegurarse de tener la última versión
3. Revisar carpeta "05\_Archivos" contiene Build Original y PDF con este documento
4. Leer la consola e ir arreglando error por error hasta no tener ninguno
5. Sistemas (Movimiento / Raycast / Inventario / Boleta)
6. UI / SFX
7. Buildear
8. Subir a Canvas

### **ENTREGA:**

- Fecha: Hoy (Martes 12 de Noviembre) hasta el final de la clase.
- Formato: Archivo en formato **.Zip**, de nombre “Encargo 4 - Nombre y Apellido”, que debe contener lo siguiente:
  - Carpeta con **build funcional**, de nombre “Build”
- Subir a: CANVAS → Programación II → Tareas → Encargos → Encargo N°4

### **REGLAS:**

1. Los estudiantes pueden consultar al docente siempre y cuando **tengan preguntas disponibles** en base a las **actividades de reforzamiento** enviadas durante el semestre.
2. Se penalizará con **Nota 1.0** a quién se le sorprenda hablando, buscando en internet o cualquier actividad prohibida.
3. **No se reciben entregas atrasadas.** En caso de que el estudiante falte, se re programará un nuevo trabajo para la próxima semana.
4. **Es responsabilidad del estudiante** asegurarse del **funcionamiento e integridad** del encargo. Si el encargo llega corrupto, será calificado con nota 1.0.
5. **NO se revisarán encargos** en otro formato que no sean los exigidos. Leer en “ENTREGA”
6. Se asume que TODOS los estudiantes leyeron el documento y rúbrica enviada el 12/11/2024