

|   |                          |             |
|---|--------------------------|-------------|
| <b><u>UTN – FR Mar del Plata – TSP</u></b><br><b>Laboratorio III - Primer Parcial Lab 3</b><br>Abril 2023 | <b>Nombre y Apellido</b> | <b>Nota</b> |
|---|--------------------------|-------------|

### IMPORTANTE:


- Crear un proyecto con su Nombre y Apellido (**solamente**).
- Añadir comentarios a su código, modularice y sea prolijo en la codificación, no sumará puntos pero podría restar.
- Al finalizar, comprímalo en un archivo **solamente** con su apellido para luego subirlo al campus
- Desarrolle un main() e invoque únicamente a las funciones que compilen correctamente. La resolución del mismo es obligatoria y sumará puntos extras en caso de necesitarlos. **No se pide ingreso por teclado de ningún tipo.**

Luego del éxito de The Last of Us Parte 1 y Parte 2 junto con sus remasterizaciones (el porteo a PC no cuenta) y la serie de HBO con Pedro Pascal (como algo puede salir mal con Pedrito, bueno si, Wonder Woman pero el problema no fue de él exclusivamente), Naughty Dog nos encarga la titánica tarea de modelar y programar la Parte 3 de esta historia de monstruos y zombies.

El juego propiamente dicho sera en 2D por lo tanto de todas las entidades del juego nos interesara conocer la posición X e Y del mapa. Hay dos grandes grupos de entidades. Los personajes y los objetos.

Vamos con los primeros. Los personajes tienen un nombre, una skin, un nivel de vida y armas (es un String, no se compliquen).

Hay dos tipos de personaje que comparten características, los que el usuario manipula (vamos a decirles Player) y los zombies. El Player va a tener un comportamiento adicional que será curarse y recargar arma (no se compliquen y devuelvan un mensaje “estoy recargando el arma”). Hay dos tipos de Player jugables o sea concretos. Joel y Ellie. Joel tendrá un atributo que indica cuán mala onda es y que tan testarudo es. Ellie en cambio poseerá un grado de humor y una cantidad de chistes malos.

Los zombies tendrán el comportamiento de morder. Al fin y al cabo son zombies, pueden morder no calcular derivadas o hacer sistemas como ustedes. Ahora, todos los personajes pueden atacar, defenderse, morir, y cargar energía. Pero desde NG nos dijeron que ese comportamiento puede adjuntarse a otros elementos que no sean personajes. (guiño guiño, acá se activa el sentido  o me levanto y me voy al buffet por un café y algunas medialunas, al profe le gustan las dulces).

Los zombies pueden ser Corredores (Runners) / Acechadores (Stalkers) / Chasqueadores (Clickers) / Hinchados o Gordinflones (Bloaters). Cuando estamos corriendo por nuestras vidas en un apocalipsis todos los zombies son iguales (decir esto es racista?) pero en el juego si son diferentes. A cada tipo de zombie vamos a inventarle un atributo y un método de acuerdo a su nombre. Los Gordinflones solamente tendrán la capacidad de camuflarse. NG nos avisó que en un DLC habrá otros enemigos que no serán zombies que tendrán la misma habilidad.

El otro grupo de entidades que tenemos en el juego son los objetos que disponen un ancho y un alto. Los objetos pueden ser un muro (tendrá una resistencia) y las plataformas (conocemos su material y el soporte que dispone)

Si, son muchas clases. Si, son muchas interfaces. Si, hay muchas CA también. Por lo que el diseño de clases tendrá una ponderación muy alta. En cuanto a lo funcional vamos a necesitar un arreglo/colección de todos los personajes y otro de objetos. Métodos para agregar y listar. Cada clase debe tener sobrecargado el toString y los constructores correspondientes. En el main mostramos toda la info. No es un juego funcional ni mucho menos ni tampoco hay relación entre las clases, por ejemplo no hace falta que Joel mate a un Clicker. No hay funcionalidad. Eso quedará para el recuperatorio (?)

#MuchoTexto #MejorMeCambioDeComisiónPeroEnLaOtraEstaTambienBenoffi

**MAY THE —  
FORCE  
— BE WITH  
YOU**