

Trabajo Practico Final de

Programación Orientada a

Objetos

- Materia: Programación Orientada a Objetos
- Nombre del alumno: Lucas Salvarrey Ezequiel
- Nombre del profesor: Gamaliel Natanael Quiroz
- Carrera: Analista de Sistemas

Tabla de contenido

| | |
|-------------------------------------|----------|
| Introducción | 2 |
| Clases | 3 |
| La clase Main | 3 |
| La clase Equipo | 3 |
| La clase GestorEquipos | 4 |
| La clase Jugadores | 4 |
| La clase Capitulo | 5 |
| UML | 6 |
| Conclusión | 7 |
| Sección extra | 7 |
| ¿Se puede mejorar? | 7 |

Introducción:

El videojuego escogido durante el desarrollo de los dos parciales y para el examen final es:

Left 4 Dead 2

Este proyecto implementa una simulación básica del videojuego Left 4 Dead 2 en Java, utilizando los temas vistos en la materia de la Programación Orientada a Objetos. Se desarrollaron cinco clases principales para implementar el comportamiento de los jugadores, equipos, capítulos de campaña y la lógica general del juego.

¿En qué consiste este juego? Consiste en un enfrentamiento entre dos equipos: supervivientes (4 jugadores) y zombies (4 jugadores), un total de 8 jugadores. Los jugadores de ambos equipos deben acumular puntos y que un equipo debe llegar a un puntaje superior al del otro para ganar la partida. La partida está dividida en cualquier cantidad de capítulos máximo de 5, durante está, los jugadores supervivientes deberán llegar a un lugar seguro para finalizar el capítulo y sobrevivir a los jugadores zombies, y estos últimos deberán hacer daño a los supervivientes para acumular puntos de cada jugador del equipo zombie. En cada capítulo hay dos enfrentamientos, es decir, si los jugadores supervivientes empiezan primero el capítulo, cuando finalice, los jugadores del equipo de supervivientes pasaran a ser del equipo de zombies y los jugadores el equipo de zombies que estaban

anteriormente pasaran a ser del equipo de supervivientes y así sucesivamente hasta que termine la partida.

A continuación, se mostrará la función que cumple cada clase del proyecto:

Clases:

La clase Main

Lo que hace es controlar el menú principal del juego y toda la interacción con el usuario. La clase Main funciona como interfaz de usuario, permite al jugador registrar nombres, verificar los equipos e iniciar la campaña. Maneja los menús, las validaciones y coordina el uso de las demás clases. En resumen, es el panel y controlador principal del juego.

La clase Equipo

Lo que hace es coordinar las campañas de enfrentamientos entre los equipos. Se encarga de distribuir los enfrentamientos y controlar el diseño general de capítulos. La clase Equipo se encarga de la lógica general de la campaña, a partir de los jugadores registrados, organiza los enfrentamientos capítulo por capítulo, ejecuta las rondas, calcula los puntos (de forma aleatoria) y muestra el equipo ganador al final (en base a los puntos acumulados de los dos equipos durante el enfrentamiento). En resumen, esta clase implementa la lógica principal de la campaña, gestionando los enfrentamientos capítulo a capítulo.

La clase GestorEquipos

Lo que hace es administrar todos los jugadores que han sido registrados y verifica condiciones, por ejemplo, evitar nombres duplicados. La clase GestorEquipos funciona como una base de datos simple de jugadores. Se encarga de agregar jugadores, evitar duplicados y mostrar la información cuando se necesite. En resumen, enlista los jugadores ingresados y evita duplicados.

La clase Jugadores

Lo que hace es representar a cada jugador del juego, ya sea un zombie o un superviviente, y guarda su nombre, salud y puntos acumulados. La clase Jugadores encapsula los datos individuales de cada participante del juego. Cada jugador tiene salud, puntos y un nombre. Esta clase permite aplicar daño y sumar puntos según el resultado de cada capítulo. En resumen, obtiene y guarda la información de cada jugador dentro del juego.

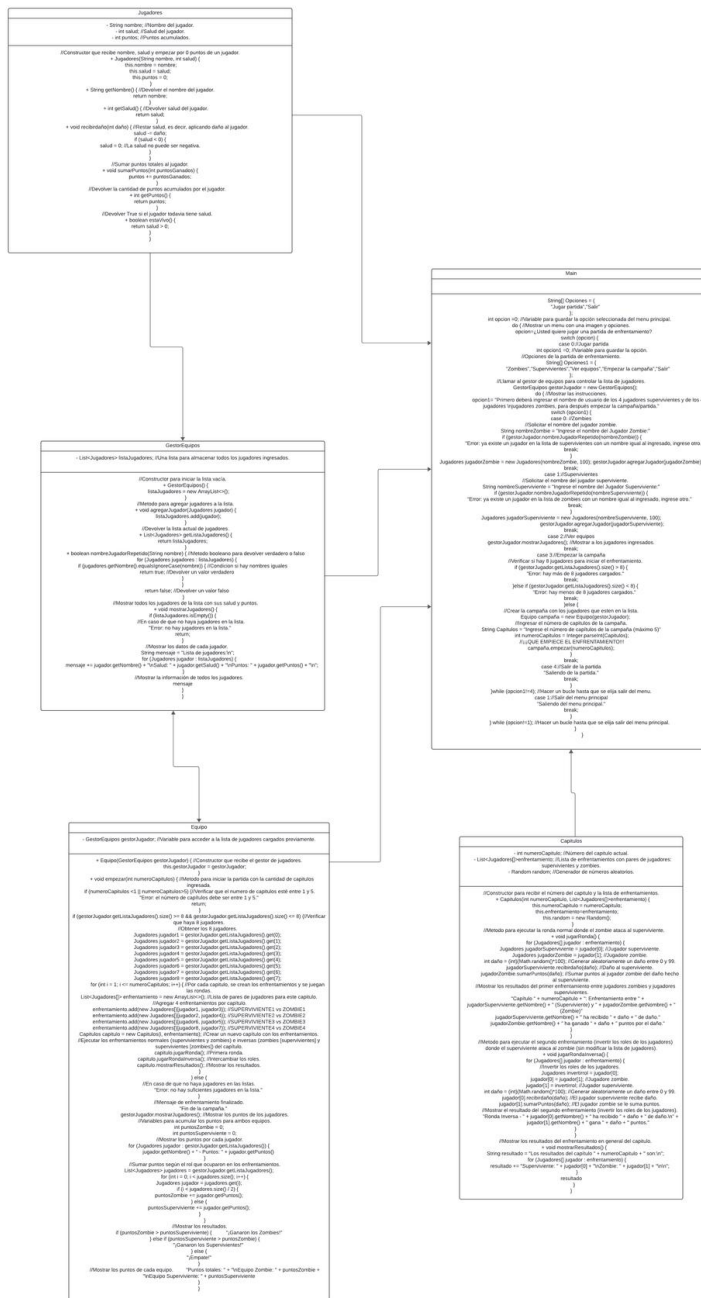
La clase Capítulos

Lo que hace es simular cada capítulo del juego donde ocurren los enfrentamientos entre jugadores zombies y supervivientes. La clase Capítulos representa una etapa de la campaña donde se enfrentan los dos equipos. Cada capítulo contiene pares de jugadores: un superviviente y un zombie.

En cada ronda se simula un combate con un daño aleatorio. Hay dos rondas: una normal y una inversa. Al final, se muestran los resultados del capítulo, y los puntos ganados se acumulan para decidir qué equipo ganó cuando finalice de la campaña.

En resumen, controla la lógica de las dos rondas de enfrentamientos entre ambos equipos, es el encargado de proporcionar los puntos acumulados de los equipos y decidir qué equipo gana la partida en base a dichos puntos.

UML:



Conclusión:

Gracias a la separación en clases y responsabilidades, el sistema simula correctamente el comportamiento de una campaña de enfrentamientos por turnos entre zombies y supervivientes. La aplicación de los temas vistos en la materia de Programación Orientada a Objetos permitió organizar el código de manera eficiente, útil y fácil de mantener.

Sección extra:

¿Se puede mejorar?

Si, se puede mejorar. Por ejemplo:

Mostrar un mensaje que diga que objetos utilizo cada superviviente durante una ronda, por ejemplo, que armas utilizo (ya sea blancas y las de fuego), cuantos botiquines y píldoras utilizo cada superviviente para curarse/subir su salud.

Agregar habilidades para cada zombie, como distintas formas de ataques.

En caso de que uno o más jugadores supervivientes hayan muerto, mostrar quien o quienes son.

Más condiciones como, por ejemplo, si un superviviente usa un desfibrilador para revivir a un compañero del equipo, se le resta 25 puntos al equipo de supervivientes, y si usa más de un desfibrilador, se le resta el doble de puntos, o sea, 50 puntos y así sucesivamente.

Siguiendo con los dos ejemplos anteriores, también agregar la condición de que, si uno o más jugadores hayan muerto, se le resta 25 o más puntos al equipo de supervivientes.