

Práctica 01

Modelado y Programación

Objetivo: *El objetivo de esta práctica es implementar adecuadamente los patrones Strategy y Observer en la resolución del problema propuesto. Su solución debe estar integrada en un solo diagrama de clases así como una sola implementación.*

1. Menciona los principios de diseño esenciales del patrón Strategy y Observer. Menciona una desventaja de cada patrón.
2. Se te ha encargado implementar la siguiente simulación:

Korby es un personaje popular de la empresa Nientiendo. La característica principal de este personaje es la de replicar habilidades de los enemigos que consume.

MeganMan es un personaje popular creado por la empresa Copcam. Se trata de un robot que es capaz de replicar las habilidades de aquellos robots a los que derrota.

Dittuu es un chinpokomon capaz de transformarse en cualquier otro chinpokomon que vea, con la capacidad de copiar sus características.

Por azares del destino, la empresa de videojuegos “De la Rosa” ha decidido integrar a todos estos personajes en un juego de peleas y les ha pedido implementar una simulación para ver si será una idea rentable.

El objetivo final es el de tener un combate, en una plataforma, donde los jugadores puedan elegir a alguno de los personajes mencionados y ponerlos a pelear. Durante el combate, aparecerán objetos especiales, uno asociado a cada personaje, lo que permitirá que obtenga sus poderes y cambie su forma de pelear. La habilidad que se obtiene se genera al azar hasta el momento en que el objeto se crea y es consumido.

Los personajes tienen formas de atacar y defenderse. Estas acciones cambian con respecto al poder que consumieron durante el combate. Desde el inicio deberían tener una forma estándar de actuar, sin embargo se espera que este comportamiento cambie durante las peleas.

Para este prototipo se les dio la libertad de elegir 3 “poderes” asociados a cada personaje. Sean creativos con las elecciones. De esta forma, cada personaje tendrá 3 opciones de poderes a asumir, para tener un total de 9 opciones. Para tratar de hacer el juego más balanceado, cada personaje individual tendrá una opción limitada de transformaciones correspondientes a sus propios juegos. Es decir que Korby sólo podrá transformarse en alguno de los enemigos que aparecen en su franquicia, lo mismo para MeganMan, y para Dittuu (Por ejemplo, Korby no debería poder obtener poderes de MeganMan).

Además de estos requerimientos, la empresa “De la Rosa” les pidió agregar una opción para transmitir los combates a un público. Se desea que al iniciar cada combate, un grupo

grande de espectadores puedan unirse a una sesión para apoyar a los peleadores. El evento debería decirle a cada espectador lo que está pasando.

Por supuesto, sería muy costoso elaborar un prototipo bastante detallado, por lo que la empresa “De la Rosa” está pidiendo algo muy concreto.

Se necesita una simulación donde un usuario no tenga que interactuar más allá de iniciar el programa. Cada combate tendrá a los 3 personajes peleando al mismo tiempo. Estos no tendrán controles y su comportamiento estará definido por ustedes. En otras palabras se necesitan casos de prueba generados por ustedes, donde se vean las características del sistema. Dado que no se necesita un prototipo con ambiente gráfico, es suficiente con manejar los eventos a través de texto. Los espectadores sólo necesitan leer lo que está pasando, pero es importante que sea sumamente detallado (quién golpea a quién, quién gana, quién pierde, cuál habilidad asume x personaje, etc.) Todo esto será guardado en una bitácora, en un archivo txt. Por cada espectador debe existir una bitácora donde se reconozca un id para cada espectador (podría ser un número de visitante al azar “espectador123” o un nombre de usuario específico). Cada espectador puede elegir un personaje preferido a apoyar. Para este prototipo, se deben crear 4 entidades correspondientes a espectadores simulados. No se necesita el uso de hilos de ejecución ni cómputo en paralelo. Sólo se necesita ver cómo se generan las bitácoras a partir de los datos del espectador, y del combate que se “transmite” dicha información. Aunque estas bitácoras dirán exactamente lo mismo, son importantes para revisar la capacidad del sistema, y que sean “personalizadas” a cada espectador. La diferencia estará en el id del espectador y un mensaje especial que indicará si el personaje que decidió apoyar ganó o perdió.

En teoría ustedes arreglarán las peleas para que haya algún ganador al final del combate. La forma en que esto ocurrirá depende de ustedes, pero se sugiere manejar puntos de vida en cada personaje y que éstos vayan disminuyendo con cada ataque. Si un personaje se defiende, perderá menos puntos de vida. Los ataques y las defensas se ven afectados por los poderes que asume cada personaje, así que tengan eso en cuenta. Como mínimo deben tener 3 casos de prueba, y cada uno debe reflejar un combate diferente. En ellos se debe ver que cada personaje asuma una habilidad, todos peleen entre ellos, y al final se anuncie un ganador. Al ejecutar el programa, se elige al azar solamente uno de estos casos de prueba, se ejecuta la pelea, se generan las bitácoras, y se termina la ejecución.

Realiza los diagramas de clase necesarios y agrégalos como imagen en la carpeta de entrega.

Sigan los lineamientos para la entrega de prácticas. El documento estará en el classroom.

Evaluación

20% Implementación de cada patrón (en total 40%).

10% Parte teórica.

10% Funcionamiento correcto.

40% Diagramas (se califica que concuerden con la implementación y sean correctos).

NOTA: Si el código no está comentado en formato Javadoc, como se indica en los lineamientos, se descontarán 10 puntos de 100.