Proyecto 2 - Patrones de Diseño

Yuen Law

Semestre I, 2022

Objetivos

1. Aplicar conceptos de ingeniería de software.
2. Aplicar diferentes patrones de diseño en el desarrollo de una aplicación.

Descripción

El objetivo de este proyecto es desarrollar un juego sencillo utilizando diferentes patrones de diseño. En específico, se deben usar 3 patrones: MVC, Factory y Observer. En el juego, se controla un personaje por medio del teclado para moverlo, atacar enemigos y rescatar aliados.

Juego

Se trata de un juego "rogue-like" en donde se controla un personaje con movimiento matricial (arriba, abajo, izquierda, derecha) para combatir enemigos generados de manera procedural de manera "infinita" hasta que el jugador pierda.

El juego es por turnos, primero se mueve el personaje y después los enemigos, por lo que no es necesario usar hilos. Cada turno del o la jugadora, termina inmediatamente con un movimiento del personaje (no incluye los ataques).

Personaje principal

Como se mencionó, puede moverse en una matriz por medio del teclado en 4 direcciones. Se puede usar ya sea las teclas direccionales o WASD (su elección). Cada presión de la tecla correspondiente mueve al personaje una casilla en dicha dirección. Adicionalmente, el personaje debe poseer un ataque, el cual se activa con alguna otra tecla, por ejemplo Espacio o Enter. Cuando se presiona la tecla, el personaje ataca la casilla directamente en frente. Otro atributo del personaje es la vida o HP; si la vida del personaje llega a 0, el juego termina.

Enemigos

Cada cierta cantidad de turnos (determinada por diseño propio) se generan nuevos enemigos en el mapa. Los enemigos siempre se mueven hacia el jugador, también de manera matricial (arriba, abajo, izquierda o derecha). Los enemigos no atacan, pero si llegan a estar en la misma casilla que el jugador, le restarán vida pero también desaparecerán. Es decir, cada enemigo puede restarle vida al personaje principal solo 1 vez. Todos los enemigos existentes se mueven una casilla después de que el jugador se ha movido.

Aliados

Los aliados también aparecen cada cierta cantidad de turnos, pero debe haber un máximo de aliados totales. Es decir, si se llega a ese máximo, no se crearán aliados nuevos aunque haya pasado la cantidad de turnos determinada. Los aliados no se mueven y son invisibles; solo serán visibles cuando el personaje esté cerca (definido también por diseño propio). Para rescatar a los aliados, el personaje debe moverse hacia la misma casilla que estos. Una vez que son rescatados, desaparecen y también recargan un poco de vida al personaje.

Mapa

El mapa es un tablero con un tamaño también definido por diseño propio, pero debe ser lo suficientemente grande para poder jugar sin mucha dificultad.

Uso de Patrones

Para el desarrollo del juego es obligatorio el uso de los siguientes patrones en las siguientes situaciones:

1. Model-View-Controller: Se debe usar este patrón para separar la lógica, la juego de la interfaz gráfica y los datos (el modelo del mapa o tablero).
2. Factory: este patrón se debe usar para crear enemigos y aliados. La clase controladora debe solicitar al Factory una instancia de enemigo o aliado; no las puede crear directamente.
3. Observer: se usará este patrón para notificarle a los enemigos y aliados cuando exista un cambio en el estado (posición) del personaje, para que cada uno de estos pueda reaccionar debidamente.

Requerimientos

- Se debe programar usando el paradigma orientado a objetos.
- Se debe aplicar todos los conocimientos de diseño y POO adquiridos, así como buenas prácticas de programación (Clean Code).
- Debe preparar y presentar un avance, en el avance es necesario presentar un diagrama de clases que muestre el uso de los tres patrones.
- Debe implementar una interfaz gráfica
- Para el control de desarrollo se debe usar la pizarra de proyecto en Github y el sistema kanban para registrar el progreso de las actividades realizadas.

Entregables

1. Documentación PDF, usando una plantilla de Overleaf con una descripción general de la solución implementada y el diagrama de clases y una breve explicación de cada clase y cómo se comunican entre ellas, por ejemplo por medio de diagramas de actividad o de secuencia. Además deben incluir una sección donde detallen cómo se aplicó los patrones de diseño, así como ventajas y desventajas de su uso que observaron durante el desarrollo del proyecto. Por último debe incluir una sección de referencias donde detalle los recursos externos utilizados para el desarrollo del proyecto, por ejemplo, tutoriales, ejemplos de código, videos, etc.
2. Código en un repositorio de Git. El último commit realizado debe ser antes de la fecha y hora límite de entrega.

Aspectos administrativos

1. Fecha de entrega: Viernes 27 de mayo
2. El pdf debe ser enviado por medio de Teams
3. El proyecto debe ser desarrollado en Java.

Evaluación

Rubro	Valor
Documento PDF	(15)
- Descripción	2.5
- Diagrama de clases	5
- Patrones	5
- Referencias	2.5
Código	(65)
- GUI	5
- Mapa	5
- Enemigos y Aliados	10
- Personaje Principal	10
- Uso correcto de MVC	15
- Uso correcto de Factory	10
- Uso correcto de Observer	10
Avance	(10)
Workflow	(10)
- Uso de git	5
- uso del board de github	5
Total	100