

Unidad Didáctica 1 y 2: Ejercicio de Feedback

Técnicas de Programación

1 DISEÑO DE UN JUEGO DE COMBATE POR TURNOS: *MINI STREET FIGHTER*

1.1 CONTEXTO GENERAL

Estás colaborando en el diseño técnico de un videojuego de combate por turnos inspirado en *Street Fighter*, con un enfoque de arquitectura robusta, extensible y orientada a objetos. El equipo de diseño ha definido que los luchadores (personajes) podrán realizar diferentes tipos de ataques, defensas y habilidades especiales. Además, quieren que el juego sea fácilmente ampliable con nuevos personajes, modos de juego, y sistemas de notificación o repetición de combates.

Para garantizar que la arquitectura sea sostenible y escalable, se ha decidido aplicar **patrones de diseño de software**.

1.2 OBJETIVO DE LA PRÁCTICA

Diseñar e implementar un juego de lucha por turnos usando **seis patrones de diseño clásicos**:

1. **Singleton** – Control central del juego.
2. **Factory Method** – Creación de personajes según tipo.
3. **Strategy** – Variantes de ataques.
4. **Observer** – Seguimiento y notificación de eventos.
5. **Command** – Acciones encapsuladas como comandos reutilizables.
6. **State** – Estados del juego (inicio, turno de cada luchador, fin de partida).

1.3 REQUISITOS FUNCIONALES

- El juego debe tener dos personajes que combaten por turnos.
- En cada turno, un jugador puede elegir entre **atacar, defenderse o usar una habilidad especial**.
- Gana el personaje que reduce la vida del oponente a 0.
- Debe poder extenderse con facilidad (nuevos personajes, ataques, modos).
- Se debe mantener un registro del combate para mostrarlo o exportarlo.

2 TABLA HASH

Suponga que se dispone de las siguientes claves: 2, 6, 9, 8, 1 que se utilizan para insertar en una tabla hash. Suponga que la tabla es de tamaño 7 y que la función de hash es $fHash(k) = (k^2 - 1)$

1. a) Dibuje el estado de la tabla hash después de cada inserción en una tabla Hash cerrada, con resolución cuadrática de colisiones.
2. b) Dibuje el estado de la tabla hash si se realiza la inserción en una Tabla Hash abierta.
3. c) Dibuje el estado de la tabla hash después de cada inserción en una tabla Hash cerrada, con resolución cuadrática de colisiones, suponiendo que se hace un rehash de tamaño el doble de la tabla + 1 cuando se supera un factor de carga del 50%.
4. d) Escriba un programa en Java que introduzca los valores anteriores en una tabla hash de la clase HashMap creándola inicialmente de tamaño 7. El programa debe escribir el contenido de la tabla después de cada una de las inserciones realizadas. Comente el resultado.

3 GRAFOS

Dada la siguiente implementación de la clase Grafo

```
class Grafo {  
    public Grafo(int numNodos) {}  
    public boolean añadirArista(Arista arista){}  
    public boolean añadirNodo(Nodo nodo){}  
    public boolean eliminarArista(Arista arista){}  
    public boolean eliminarNodo(Nodo nodo){}  
    private int posicionNodo(Nodo nodo) {}  
    public Nodo[] listaNodos() {}  
    public Arista[] listaAristas() {}  
    private int numeroDeAristas() {}  
    private int numeroDeNodos() {}  
}  
public class Nodo {  
    private String nombre;  
  
    public Nodo(String nombre) {}  
    public String getNombre() {}  
    public boolean igual(Nodo v) {}  
}  
public class Arista {  
    private Nodo v1;  
    private Nodo v2;  
    private int peso;  
    public Arista(Nodo v1, Nodo v2, int peso) {}  
    public Nodo getV1() {}  
    public Nodo getV2() {}  
    public int getPeso() {}  
}
```

Donde es irrelevante la implementación del grafo como matriz de adyacencias o como lista de adyacencias, se pide lo siguiente: Para cierta aplicación se ha detectada la necesidad de realizar una operación de limpieza de un grafo dirigido. Esta operación consiste en eliminar todos los nodos de grado 0 y cambiar el signo de todas las aristas de peso negativo. Al finalizar la operación no deben quedar en el grafo ni aristas con peso negativo ni vértices de grado 0. El resultado debe ser un nuevo grafo. Escriba un método para esta operación de la siguiente forma:

```
public Grafo limpieza() {}
```

4 EVALUACIÓN

Pregunta	Nota
Patrones de diseño	50%
Tablas Hash	25%
Grafos	25%



Universidad
Alfonso X el Sabio

G R A C I A S