

Nombre: Mauricio Bennett Alcantar Domínguez		Matrícula: 2938988
Nombre del curso: Programación y prototipo de videojuegos	Nombre de la profesora: Claudia Lorenzo Muradas	
Tema 6: Game programming	Actividad 6: Análisis de patrones de diseño para videojuegos	
Fecha: 6 de marzo de 2022		

Referencias

Gabriel de la Cruz Luna. (2017). Introducción a la Programación de Videojuegos y Gráficos . 2022, de Silo.Tips Sitio web: <https://silo.tips/download/introduccion-a-la-programacion-de-videojuegos-y-graficos-2>

kaolong. (2011). Patrones de diseño I. 2022, de slideshare Sitio web: <https://es.slideshare.net/kaolong/patrones-de-diseo-i>

Wikipedia. (2021). Patrón de diseño. 2022, de Wikipedia Sitio web: https://es.wikipedia.org/wiki/Patrón_de_diseño

Patrones

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

- Un patrón de diseño resulta ser una solución a un problema de diseño.

Existen multitud de criterios para clasificarlos (patrones de arquitectura, de diseño, dialectos, interacciones...)

Pueden definirse tres categorías fundamentales:

1. Patrones de creación
2. Patrones estructurales
3. Patrones de comportamiento

1. Patrones de creación

Proporcionan una solución relacionada con la construcción de clases, objetos y otras estructuras de datos.

2. Patrones estructurales

Tratan sobre la forma de organizar las jerarquías de clases, las relaciones y las diferentes composiciones entre objetos para obtener un buen diseño bajo un determinado contexto.

3. Patrones de comportamiento

Las soluciones de diseño que proporcionan los patrones de comportamiento están orientadas al envío de mensajes entre objetos y cómo organizar ejecuciones de diferentes métodos para conseguir realizar algún tipo de tarea de forma más conveniente.

Patrones de creación:

Singleton (instancia única): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.

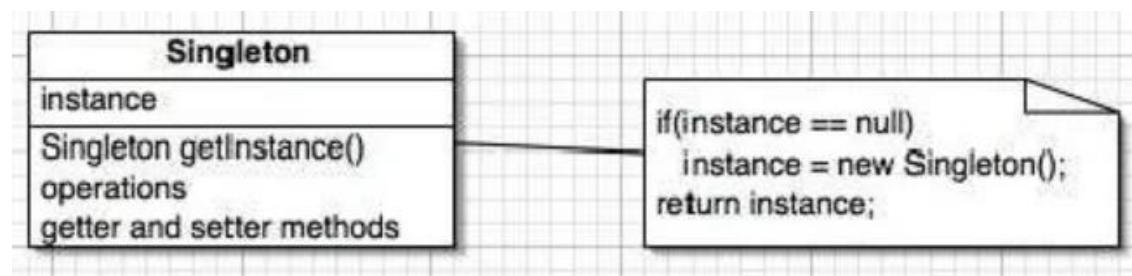
Ventajas y desventajas

- La clase Singleton es la única clase que puede crear una instancia de sí misma. No se puede obtener ninguna instancia sin utilizar el método estático proporcionado.
- No necesita pasar la referencia a todos los objetos que acceden al Singleton.
- Sin embargo, este patrón puede presentar problemas de acceso multithread, dependiendo de cómo se haya realizado la implementación.

Dónde se emplea

Se usa únicamente cuando se quiera usar una instancia de una clase, pero debe estar disponible globalmente.

Estructura básica



Builder (constructor virtual): abstrae el proceso de creación de un objeto complejo, centralizando dicho proceso en un único punto.

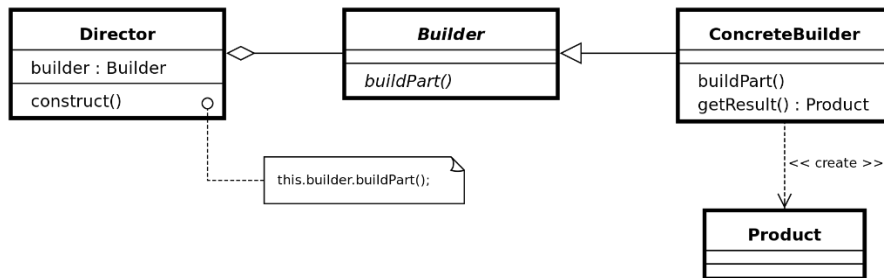
Ventajas y desventajas

- Reduce el acoplamiento.
- Permite variar la representación interna de estructuras complejas, respetando la interfaz común de la clase Builder.

Dónde se emplea

Para construir un objeto complejo este patrón nos será de utilidad, ya que nos permite crear un objeto por partes.

Estructura básica



Patrones estructurales:

Facade (Fachada): Provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema.

Ventajas y desventajas

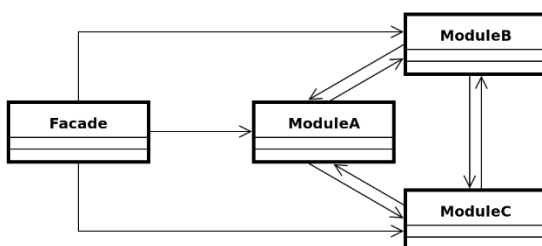
La principal ventaja del patrón fachada consiste en que, para modificar las clases de los subsistemas, sólo hay que realizar cambios en la interfaz/fachada, y los clientes pueden permanecer ajenos a ello. Además, y como se mencionó anteriormente, los clientes no necesitan conocer las clases que hay tras dicha interfaz.

Como desventaja, si se considera el caso de que varios clientes necesiten acceder a subconjuntos diferentes de la funcionalidad que provee el sistema, podrían acabar usando sólo una pequeña parte de la fachada, por lo que sería conveniente utilizar varias fachadas más específicas en lugar de una única global.

Dónde se emplea

Cuando se necesite proporcionar una interfaz simple para un sistema complejo, o cuando se quiera estructurar varios subsistemas en capas, ya que las fachadas serían el punto de entrada a cada nivel. Asimismo, se puede usar cuando se desee reducir el acoplamiento entre los clientes y los subsistemas, haciéndolo más independiente, portable y reutilizable.

Estructura básica



Bridge (Punto): Desacopla una abstracción de su implementación.

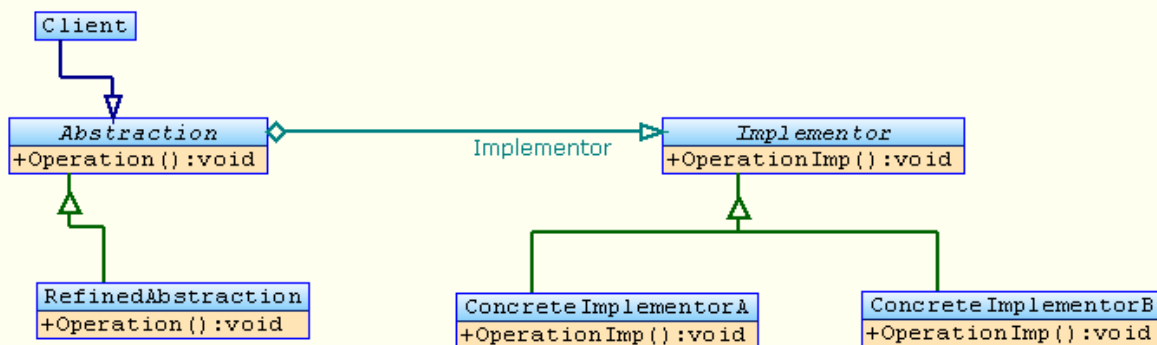
Ventajas y desventajas

- Desacopla interfaz e implementación.
- Mejora la extensibilidad
- Esconde los detalles de la implementación a los clientes.

Dónde se emplea

Se usa este patrón cuando se desea evitar un enlace permanente entre la abstracción y su implementación. De igual manera cuando las abstracciones como sus implementaciones deben ser extensibles por medio de subclases. En este caso, el patrón Bridge permite combinar abstracciones e implementaciones diferentes y extenderlas independientemente.

Estructura básica



Patrones de comportamiento:

Memento (Recuerdo): Permite volver a estados anteriores del sistema.

Ventajas y desventajas

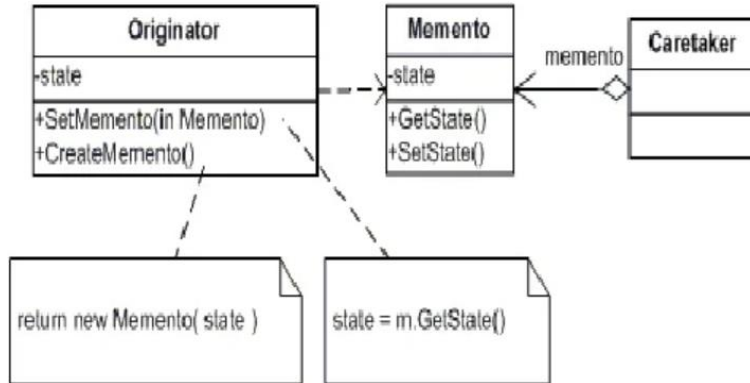
Deja alguna información en un objeto para que sea accesible por otro objeto utilizando control de acceso por defecto.

- Es conveniente de usar en transacciones a Bases de Datos
- En este patrón se da la responsabilidad de almacenar los distintos estados a la parte solicitante (el cliente).

Dónde se emplea

Se usa este patrón cuando se quiere poder restaurar el sistema desde estados pasados y, por otra parte, es usado cuando se desea facilitar el hacer y deshacer de determinadas operaciones, para lo que habrá que guardar los estados anteriores de los objetos sobre los que se opere (o bien recordar los cambios de forma incremental).

Estructura básica



Conclusión

Ya sea que estemos desarrollando y se nos atraviere un obstáculo en el camino, los patrones de diseño están para ayudarnos resolviéndolos con mayor eficacia, a su vez estos nos servirán en un futuro, ya que son reutilizables. Eso sí, tiene su grado de complejidad usarlos, más que nada es aprender e investigar a profundidad sobre cada uno para conocer su función.

Una vez que investigué sobre estos patrones puedo decir que es algo totalmente nuevo para mí, me tomó tiempo entender este tema. Encontré muchos patrones y en cierta parte me pareció algo pesado. Solamente poniéndolo en práctica podré entenderlos con mayor facilidad.