Ingeniería del Software

Práctica 6

Patrones de Diseño

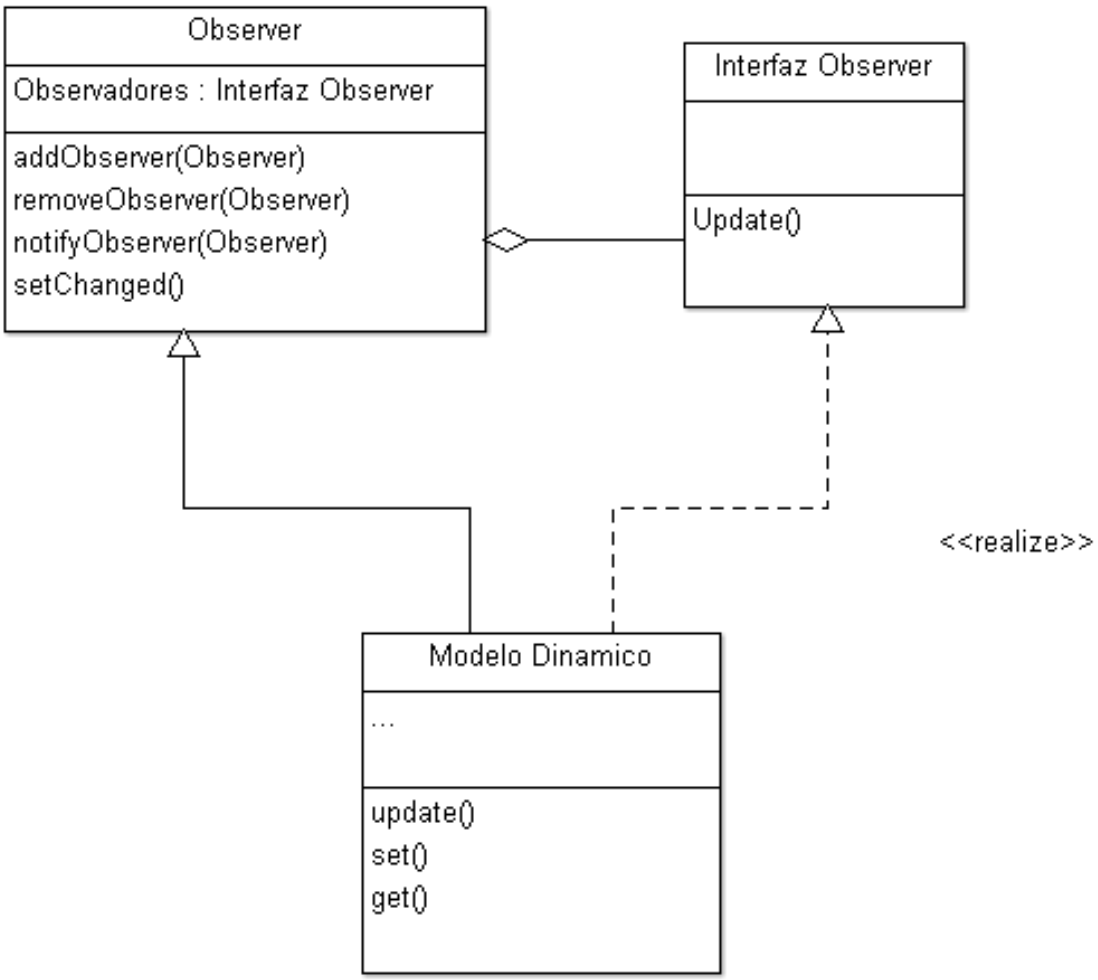
Jorge Castellano Castellano
Gabriel Fernández Díaz
David Morales Sáez

Patrones de Diseño

En la presente entrega presentaremos los diferentes patrones de diseño seleccionados para nuestro caso. Se analizarán y comentarán los patrones de tipo Observador, Singleton, Command y Fachada, así como la unión de Fachada y Observador en un diseño final combinado.

Observador

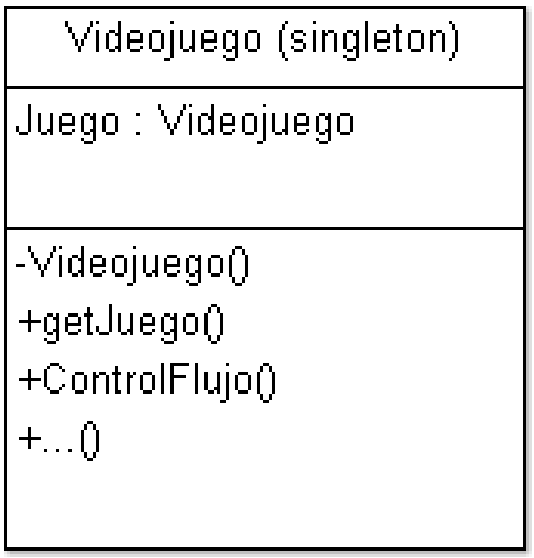
Este patrón se aplicará a Modelo Dinámico, siendo los Observadores y los Observados modelos dinámicos. Su objetivo es organizar y optimizar los ciclos de actualización del juego. Por ejemplo, el personaje del jugador al entrar en una nueva zona se subscribirá como observador del resto de los Modelos Dinámicos de la misma y viceversa. Así se optimizará la interacción entre estos modelos y se facilitará la tarea de programación.



Observador

Singleton

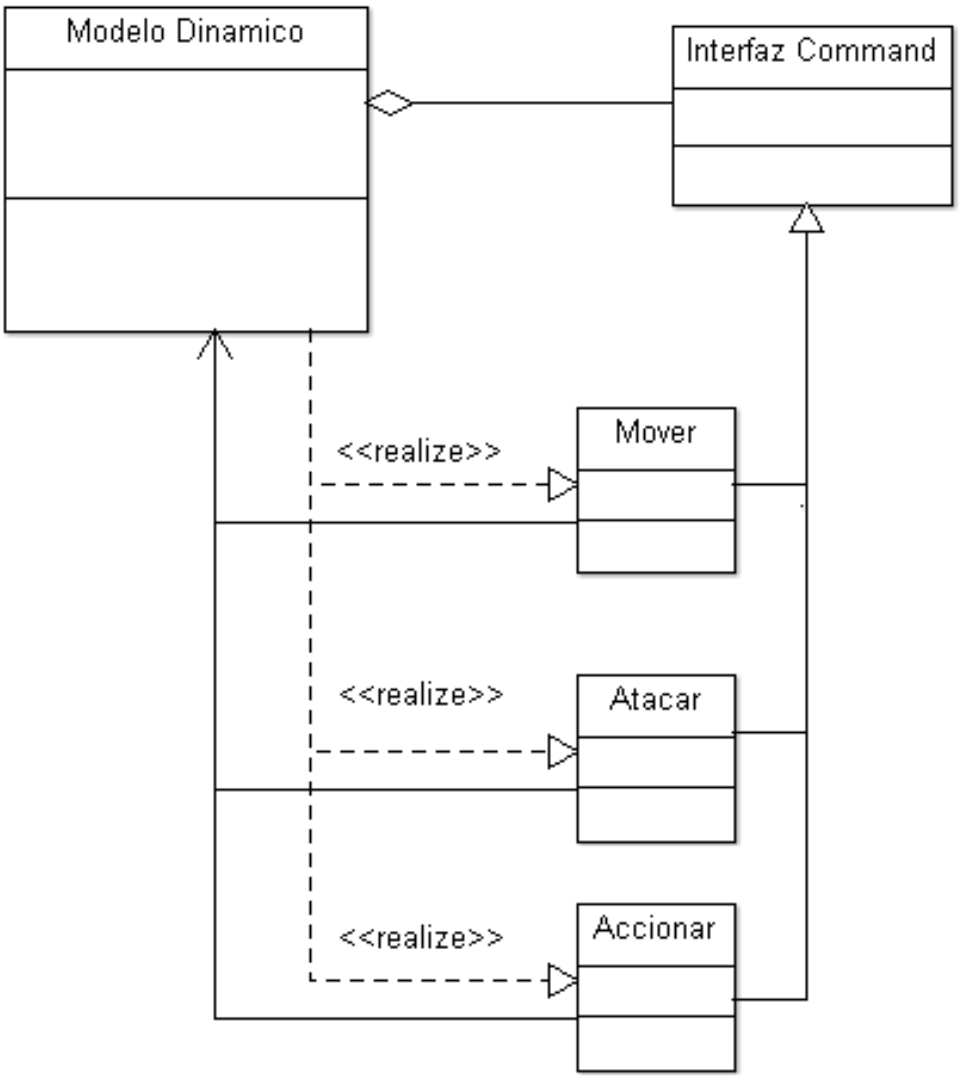
En nuestro diseño hay cuatro entidades sensibles de ser convertidas usando el patrón Singleton: Escenario, Fase, Interfaz y Videojuego. Esto se debe a que, en todo momento de la ejecución, habrá y sólo deberá haber una instancia de estas clases, y su función de control y centro de información.



Singleton

Command

Se aplicará este patrón para generalizar las interacciones de las clases de personajes pertenecientes a una Zona y esta misma. Así se regulará y guardará las acciones llevadas a cabo por los personajes.

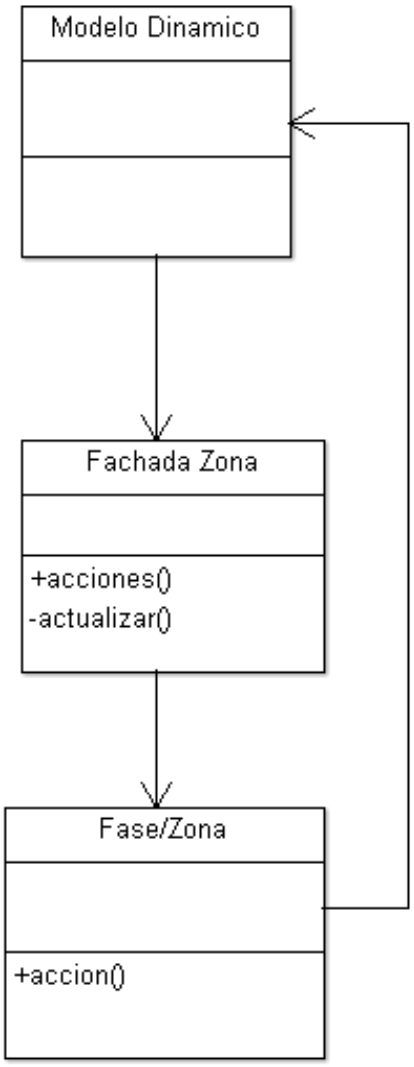


Command

Fachada

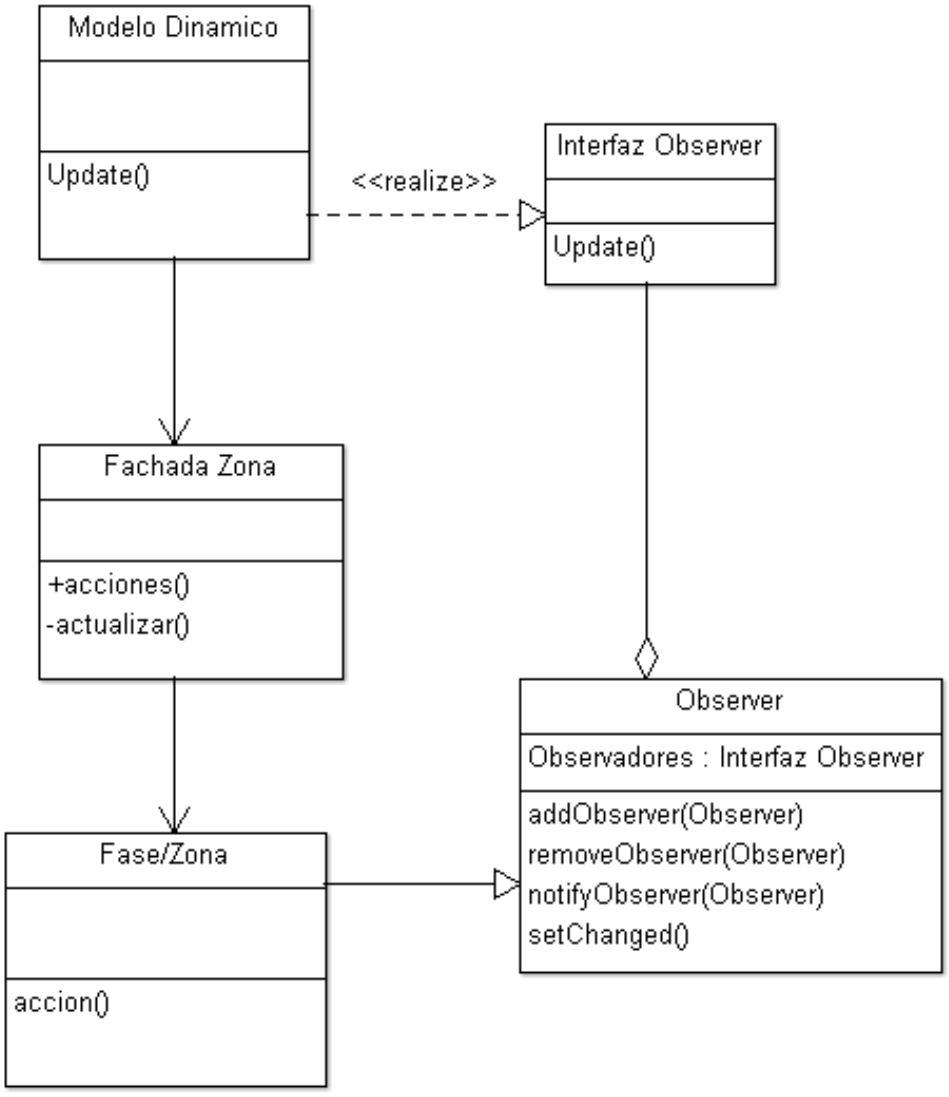
Con el diseño original, la estructura del juego, cada modelo dinámico (Jugadores, no jugadores, etc.), ejecutaba sus propias acciones y era responsable de controlar los efectos de estas. Por ejemplo, cuando el jugador interaccionase con un objeto, el objeto "personaje jugador" tendría que realizar la acción sobre el objeto.

Gracias al patrón Fachada, se puede cambiar este esquema para que todas las acciones se realicen de forma ordenada y sistemática. Para ello habría una fachada entre los "modelos dinámicos", que sería usada para solicitar las acciones, y "Fase/Zona", que sería la encargada de modificar a todos los objetos afectados por estas acciones. Volviendo al ejemplo anterior, ahora el "personaje jugador" solo tendría que llamar a "interacción" en la Fachada, que se lo comunicaría a "fase/zona", y este a su vez haría las modificaciones pertinentes en el sistema.



Fachada con Observador

De los patrones anteriormente mencionados, podemos combinar el de Fachada y el Observador, ya que actúan sobre el mismo conjunto de clases y sus funciones son complementarias. De este modo, el proceso de actualización del sistema de las consecuencias de las acciones recibidas por la fachada que realiza "Fase/Zona", se completaría usando el patrón observador. Así pues "Fase/Zona" pasaría a ser el observado, y el resto de los modelos dinámicos serían los observadores.



Fachada con Observador