

Programación Orientada a Objetos II

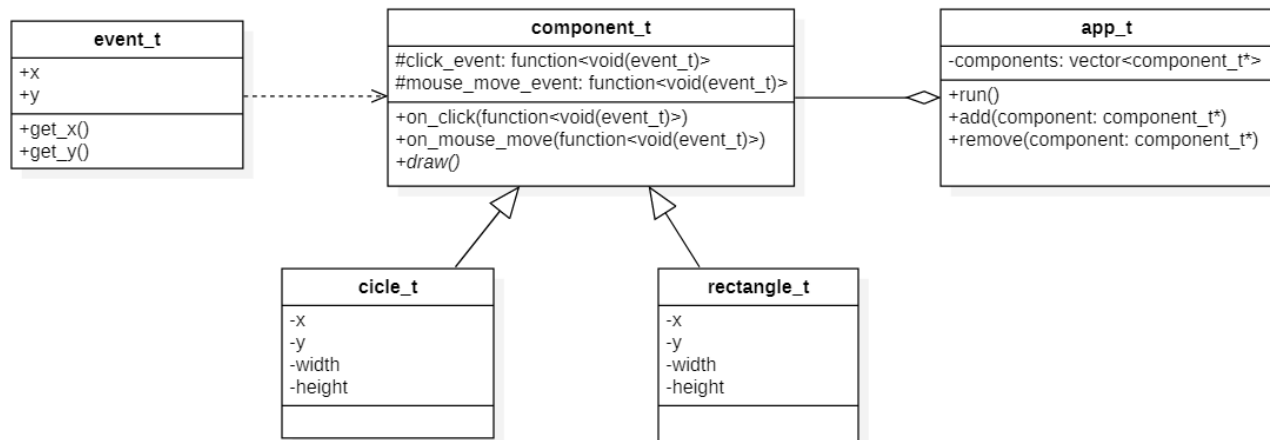
Observer Pattern

2020 - 1

Profesor: Rubén Rivas

El patrón de diseño Observador (Observer Pattern) es una herramienta usada en la programación reactiva para propagar información. Uno de los usos comunes de este patrón es en el manejo de eventos en entornos de Interfaz Gráfica de Usuario (GUI).

La presente tarea es implementar el siguiente diagrama de clases:



Este diagrama representa una estructura simplificada de un Framework de interfaz gráfica de usuario (GUI), donde la clase **app_t** asume el rol de la clase **Observable** y la clase **component_t** asume el rol de la clase **Observer**.

Usualmente la clase **app_t** contiene la pantalla principal y un método que inicia su ejecución, en este ejemplo la clase **run** que gestiona además las acciones que se realizan el usuario al interactuar con el programa, así como en el caso de la clase **Observable** la clase **app_t** cuenta con el método **notify**, que cuenta con filtro que permite el llamado de diferentes métodos similares al método **update** de la clase **Observer**. De otro lado el método **add** cumple el role del método **attach** y el método **remove** el role de **detach**.

La clase **component_t**, cuenta con atributos que son del tipo `std::function<void(event_t)>` y que cumple el role del método **update** especializados por el tipo de evento que controlan (**Click**, **MouseMove**, etc.).

Debido a que los atributos `std::function` no definen realmente el `callable` (método, función o lambda) que va a ser llamado, la clase `component_t` implementa para cada atributo un método que permita asignarle un `callable`, esto le da flexibilidad a `component_t` para que a través de una clase derivada el callable real sea implementado.

En el diagrama se observa también las clases `circle_t` o `rectangle_t` derivadas de `component_t`, donde opcionalmente deben implementarse los atributos `std::function`. También se observa la clase `event_t` que permite transmitir desde `app_t` hacia `component_t` la información asociada al evento relativo al movimiento del mouse.

Se solicita implementar el diagrama en un entorno consola simulando los eventos con comandos `iostream` y posteriormente investigue como podría hacer para implementarlo en una librería grafica existente, se sugiere lo haga con SFML, SDL u otro.

Barranco, 13 de junio del 2020