

## CS1103

### Programación Orientada a Objetos II

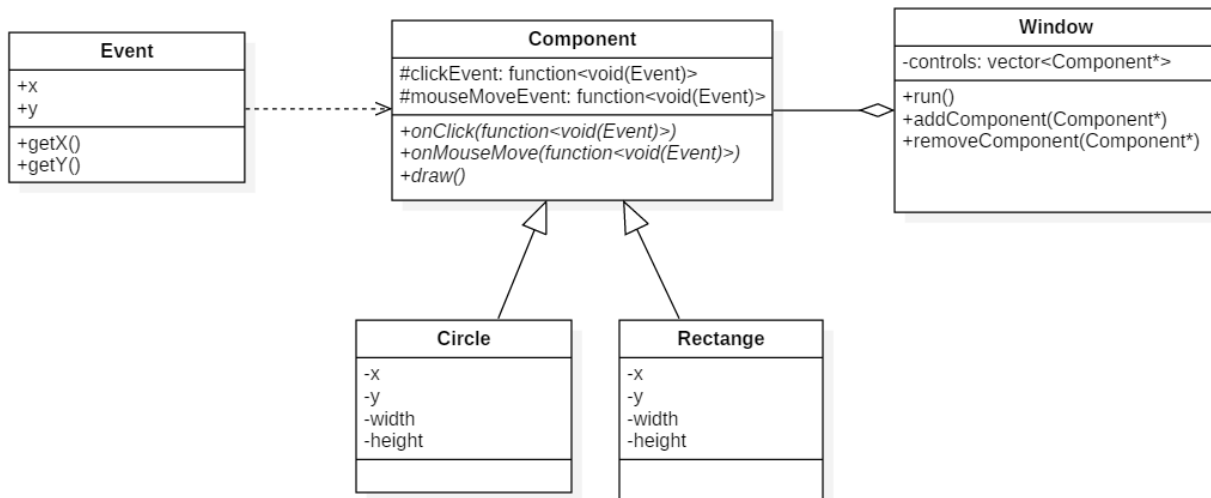
#### Observer Pattern

2019 - 1

Profesor: Rubén Rivas

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

La presente tarea es implementar el siguiente diagrama de clases:



Este diagrama representa una estructura muy simplificada de un UI Framework que utiliza el patrón Observador, donde **Window** representa el **Subject** y **Component** representa el **Observer**.

Para el caso de **Window** el método `run` representa tanto el `setStatus` como el `notify` en un solo método, por lo que, si se requiere dividir, tenga la libertad de realizarlo. Adicionalmente el método `addComponent` representa el `attach` y `removeComponent` el `detach`.

Para el caso de **Component**, debido a que puede recibir distintos tipos de mensajes, el método `update` se divide por el tipo de evento que se va a recibir (`OnClick`, `OnMouseMove`).

Inicialmente impleméntelo usando el output estándar (std::cout), pero investigue como podría hacer para impleméntalo en una librería grafica existente, se sugiere lo haga con SFML por su facilidad de uso.

Para aquellos que desean implementarlo en SFML les envié un enlace que podría ser de utilidad, con un modelo de proyecto CMAKE:

<https://github.com/rrivas-utec/game>

Barranco, 28 de junio del 2019