**Design Pattern: Snake**

**2 Interfaces**

1. *PublisherInterface* met 2 abstracte methodes: “Register en Unregister.” Deze 2 methodes moeten beiden een Object meekrijgen die de *ObserverInterface* implementeert.
2. *ObserverInterface* die een abstracte methode Update heeft. Elke class die deze interface implementeert, zal deze methode dus ook moeten aanmaken.

**Publisher Class**

De Publisher class implementeert de PublisherInterface en moet dus die 2 methodes (Register en Unregister) ook in zijn class aanmaken. De Publisher class implementeert niet enkel de PublisherInterface, het erft ook over van de TimerTask class.

In de Publisher class kan men dan over de Run methode van de TimerTask class schrijven.

In de Publisher class wordt er een Arraylist aangemaakt van Objecten die de ObserverInterface implementeren. Met de class Register kan je er één toevoegen, met de methode Unregister verwijder je er een. De Publisher zal in de Run methode (overgeschreven van de TimerTask) door deze Arraylist lopen met een For-lus. Per object zal hij dan de Update methode van dit object uitvoeren.

In de HelloXlet class kan er dan een Publisher worden aangemaakt en een Timer om aan te geven hoe snel de scene moet updaten.

**Example**

* Ik wil een slang hebben in mijn spel Snake. In ons project wordt er een class aangemaakt genaamd: SnakeChain. Deze snakeChain zal moeten updaten per timertik, we moeten het dus op een manier kunnen laten updaten in een Run methode van een Timertask.
* Ik implementeer de ObserverInterface bij de SnakeChain class en geef het dus een Update methode. In deze Update methode schrijf ik wat er moet gebeuren per Timertik met deze slang.
* In de HelloXlet class maak ik een Publisher aan en een Timer. Ik geef mijn SnakeChain mee aan de Publisher aan de hand van de Register methode van de Publisher.
* Nu zal de Publisher in zijn ArrayList een Object hebben dat de ObserverInterface Implementeert (de SnakeChain) en dus een Update methode heeft. De Publisher zal deze Update methode uitvoeren per Timertik.