

# **RAPPORT TP AOC METRONOME**

**Trinôme :**

**Anis salem Fadhloun**

**Chouaib Hentabli**

**Faical Irifi**

**Master 2 Génie Logiciel  
ISTIC - Université Rennes1  
Année Universitaire 2016-2017**

## Introduction

Le présent rapport a pour but de décrire le projet « Métronome » pour la matière « AOC ». Il s'agit de concevoir et de mettre en œuvre une application de type métronome dont l'interface graphique doit suivre la forme suivante :

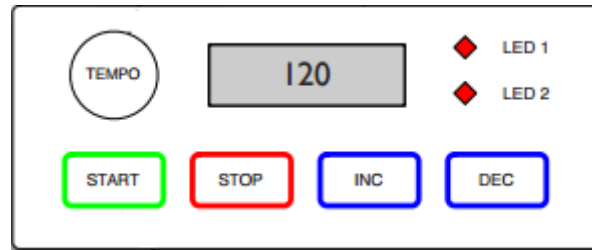


Figure 1 : interface graphique du métronome

Pour effectuer ce travail, nous nous sommes partis du fait qu'il y aurait trois versions du métronome :

- La première version a été développée avec une IHM écrite en JavaFx.
- La deuxième réutilise la même IHM - développée dans la version 1 - pour simuler une interface matérielle à l'aide du patron de conception adapter qui la rend passive en respectant les définitions des interfaces de base du sujet.
- La troisième version correspond à l'intégration d'une vraie IHM matérielle, cette étape n'est pas réalisable.

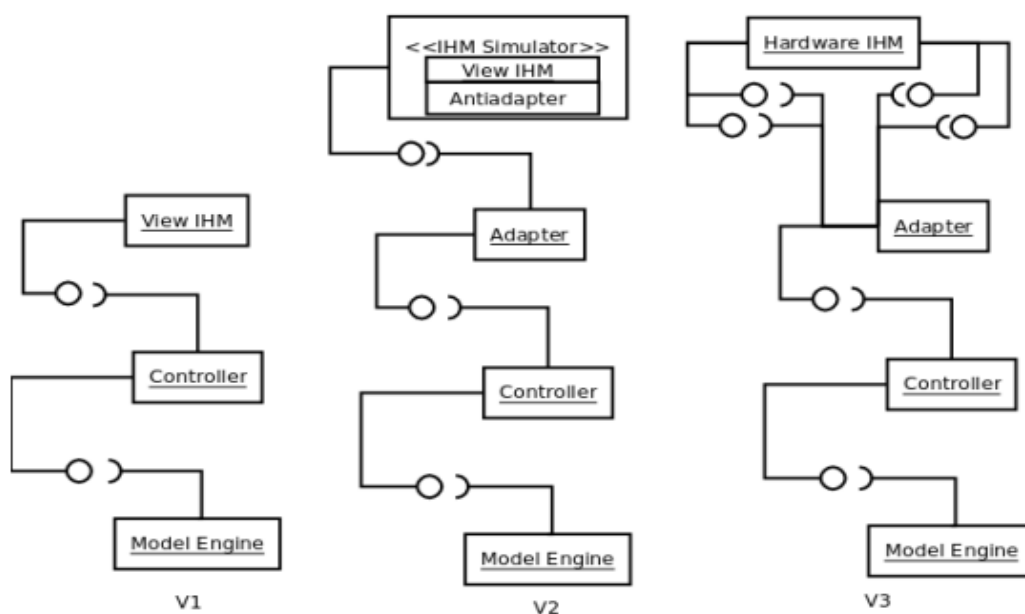


Figure 2 : Les trois versions du métronome.

Notre travail doit mettre en œuvre certains patterns de conception qu'on a vu durant les séances de cours et de TD.

Les parties suivantes vont détailler plus les patrons de conception utilisés ainsi expliquer les différentes phases du projet.

## **Réalisation :**

### **Environnement de travail :**

Le choix du langage de développement été imposé par nos professeurs et donc on a développé notre applications avec « Java FX » :

Java FX est une technologie créée par Sun Microsystems qui appartient désormais à Oracle. Java FX est devenu la bibliothèque de création d'interface graphique officielle du langage Java, pour toutes les sortes d'application (applications mobiles, applications sur poste de travail, applications Web), le développement de son prédécesseur Swing étant abandonné.

### **Conception :**

Dans cette partie, nous nous intéressons à la conception de notre solution. Pour cela, nous avons opté pour UML qui est un moyen d'exprimer des modèles objets.

### **Patron de conception « Command » :**

Le Pattern de conception « Command » nous permet de réaliser l'abstraction d'une action ce qui veut dire de découpler la création d'une action a son exécution.

On distingue différents parties qui agissent dans ce pattern. Le premier c'est Invocateur, c'est l'élément IHM qui déclenche l'exécution des commandes. Le deuxième élément est la commande concrète qui représente un type de commande particulier. Le Client qui fait l'instanciation de la ConcreteCommand, son rôle est de capturer les paramètres passés au constructeur de ConcreteCommand. Finalement, le Receptor qui est la cible du type commande.

Le principe de fonctionnement de ce pattern est : le Client capture dans le système les valeurs permettant de paramétrer une ConcreteCommand. Il l'instancie en lui passant ces paramètres l'Invoker déclenche une Command; la ConcreteCommand exécute une action impactant le Receptor en utilisant les paramètres que son constructeur a capturés.

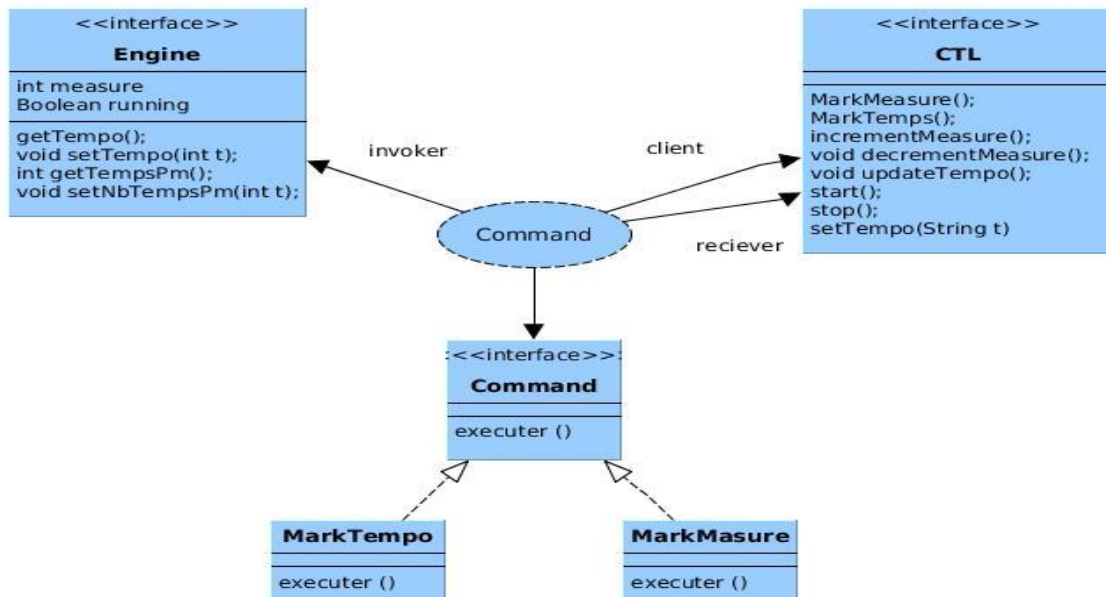


Figure 3 : Application du patron de conception command pour gérer les événements périodiques du moteur du métronome.

Dans ce qui suit, nous allons présenter un diagramme de séquence correspondant à l'initiation des événements :

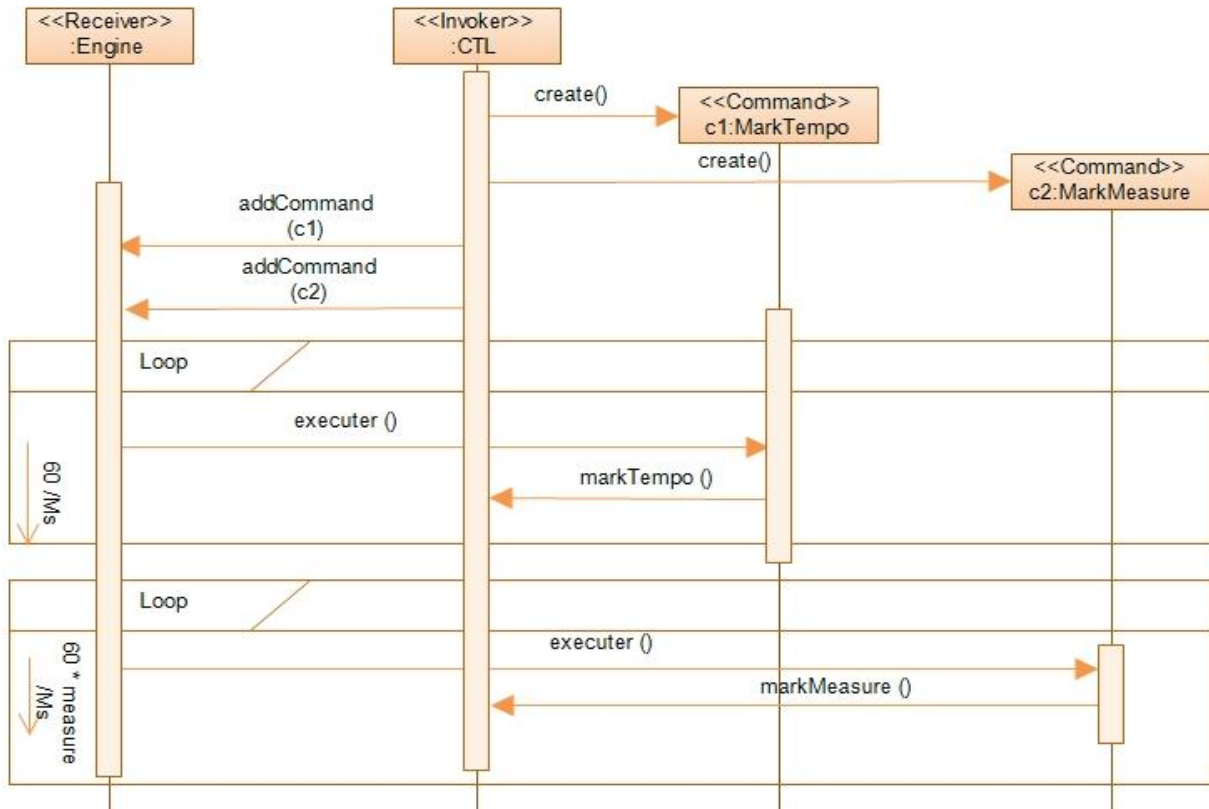


Figure 4 : Diagramme de séquences de l'initialisation des événements.

## Patron de conception « Observer » :

Le pattern Observer son principe est qu'on a un observé et plusieurs observateurs, a la modification de l'observé tous les observateurs seront notifié et avec ce pattern on a la possibilité d'ajouter ou de retirer dynamiquement des observateurs.

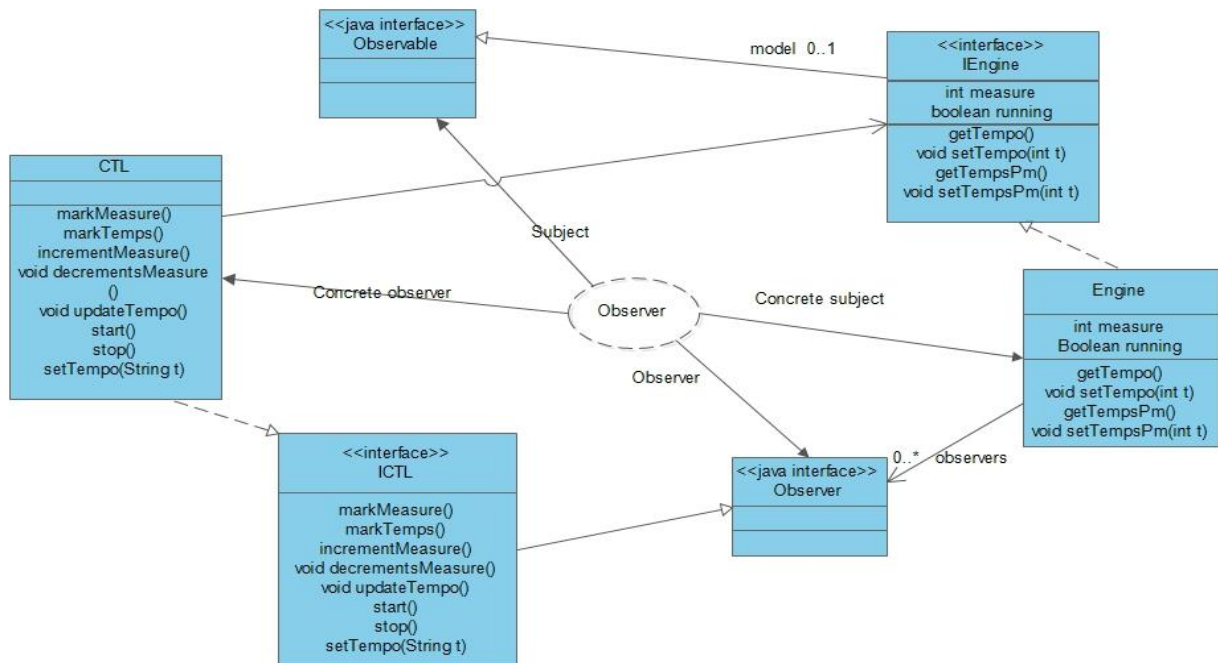


Figure 5 : Patron de conception observer entre le moteur et le contrôleur

Le contrôleur observe le moteur et met à jour si nécessaire l'IHM, De ce fait, on a utilisé un observer multi méthodes. Le but de ces méthodes consiste à la mise à jour du tempo, la mise à jour de la longueur d'une mesure et le démarrage ou l'arrêt du métronome.

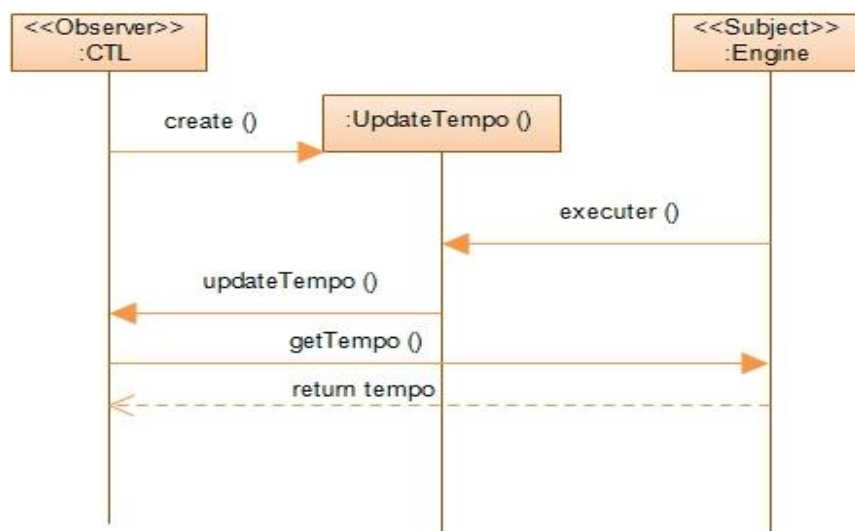


Figure 6: Diagramme de séquences des interactions entre le moteur et le contrôleur

## Patron de conception « Adapter » :

Le patron de conception Adapter sert à faire le lien entre les interfaces de base du sujet et le moteur du métronome. Chaque composant de l'interface matérielle est adopté pour fournir la même interface que les composants de la V1 (Adapter).

Dans un deuxième temps, le même patron de conception permet aux composants de la V1 d'implémenter interfaces matérielles de base pour devenir passive et constituer un simulateur pour la V2 (Anti-Adapter).

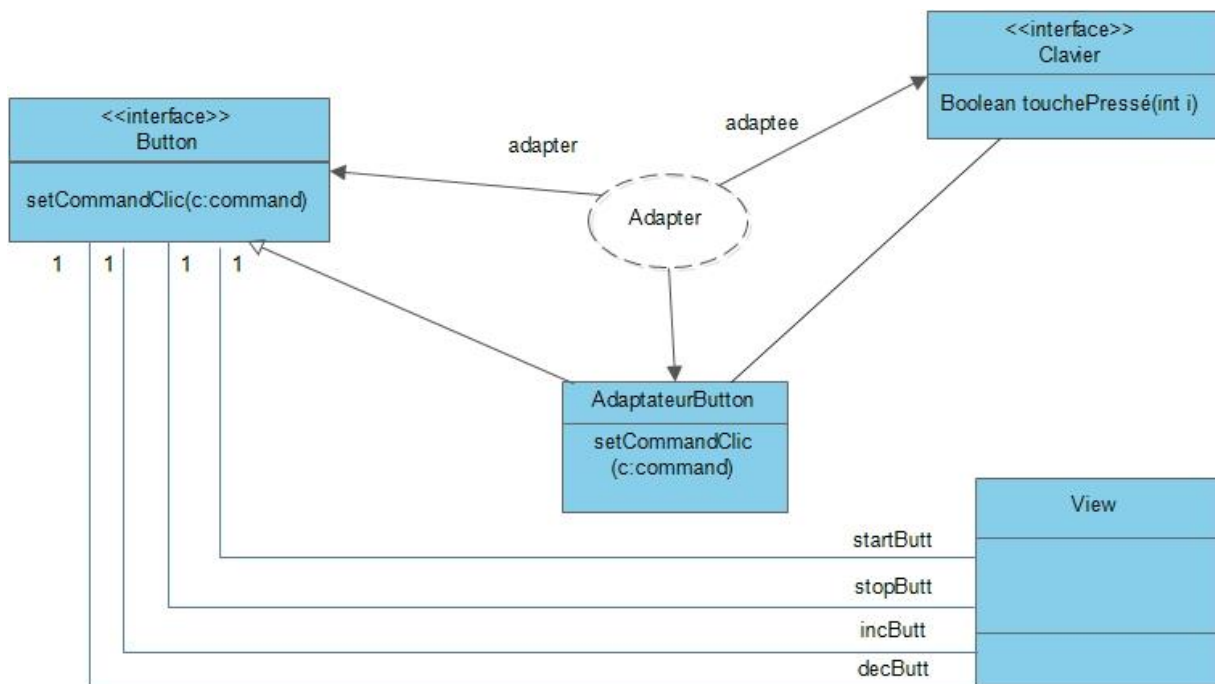


Figure 7 : Applications du patron de conception Adapter pour l'utilisation des interfaces

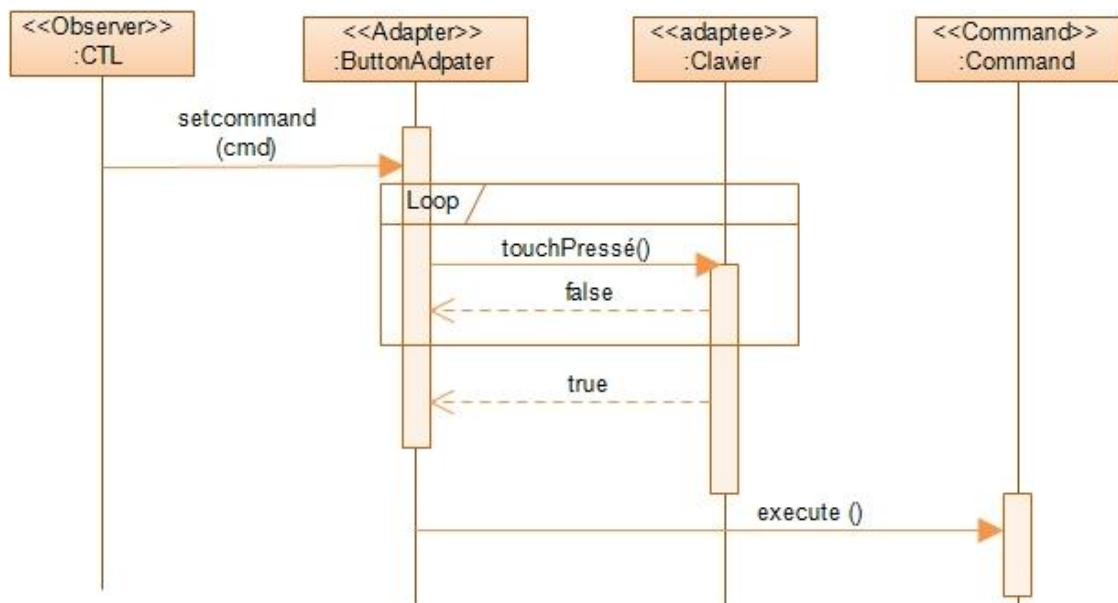


Figure 8 : Diagramme de séquences de la mise en place d'un système de polling pour la mise en œuvre de Button employant l'IHM

### **Exécution de l'application**

Pour exécuter la version1, il faut lancer la classe main.Main. Et pour la version2 il faut lancer la classe sample.Main.

### **Conclusion**

Ce projet nous a permis de nous familiariser avec les patterns de conception et d'appliquer certains. Il représente pour nous une bonne expérience pour étudier et appliquer les cours théorique qu'on a eues en classe durant le module AOC.