Métronome

Document de conception détaillée

Version 2.6

21 Décembre 2012

Gharsalli Mohamed Rafik GHAM02078801

Olivier Charrier CHAO25098007 Fahd Kacem KACF21038103

Historique des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| 17/11/2012 | 2.0 | Plan de travail | Gharsalli Rafik |
| 20/11/2012 | 2.0 | Première version du diagramme de classe | Fahd Kacem |
| 22/11/2012 | 2.1 | Version finale 1 du diagramme de classe | Olivier Charrier |
| 23/11/2012 | 2.2 | Diagramme de séquence | Gharsalli Rafik |
| 25/11/2012 | 2.3 | Implémentation de l’application | Olivier Charrier |
| 26/11/2012 | 2.3 | Implémentation de l’application | Gharsalli Rafik |
| 27/11/2012 | 2.4 | Implémentation de l’application | Fahd Kacem |
| 30/11/2012 | 2.6 | Implémentation des patrons | Olivier Charrier  Fahd Kacem  Gharsalli Rafik |
| 20/12/2012 | 2.7 | Finalisation du travail | Olivier Charrier  Fahd Kacem  Gharsalli Rafik |
| 23/12/2012 | 2.8 | Révision final | Olivier Charrier  Fahd Kacem  Gharsalli Rafik |

Table des matières

1. Introduction

1.1 But du document

1.2 Public visé

1.3 Définitions, acronymes et abréviations

1.4 Références

1.5 Structure du document

2. Description sommaire du système

3. Diagramme de classes et patron

3.1 Le choix de patron de conception Commande

3.2 Le choix de patron de conception Observer

3.3. Le choix de patron de conception Singleton

3.4. Le choix de patron de conception DAO

3.5. Le diagramme de classe

4. Diagrammes d’interaction

4.1 Diagramme de séquence bouton *Start* pressé

4.2 Diagramme de séquence bouton *Stop* pressé

4.3 Diagramme de séquence Régler Molette (tempo)

4.4 Diagramme de séquence bouton Increase pressé (mesure)

4.5 Diagramme de séquence bouton Decrease pressé (mesure)

5. Appendices

Conception détaillée

# Introduction

## But du document

Le présent document décrit de façon générale la conception du système tout en utilisant des patrons de conceptions cette fois-ci.

On y présente notamment la définition du système à développer, l'identification de ses fonctionnalités, le public visé, le diagramme de classes en représentant tous les patrons de conception qui entrent en jeu de réalisation ainsi que l'environnement ou le système évoluera. Ainsi, le document présente le diagramme de séquence et aussi il présente:

» Les exigences du système à développer.

» L'analyse des fonctionnalités.

## Public visé

Ce document s'adresse à Mr. Leshob Abderrahmane, professeur du cours INF5153, au démonstrateur du cours INF5153, à toute personne qui s'intéresse à ce type d'application et à toute personne intéressée à comprendre les patrons de conception appliqués sur des applications concrètes et relativement simples.

## Définitions, acronymes et abréviations

|  |  |
| --- | --- |
| **Terme** | **Définition** |
| Exigences | Ensemble d'actions que doivent faire le logiciel. |
| Utilisateur | Individu qui utilise le mini-éditeur de texte |
| UQÀM | Université du Québec à Montréal |
| UML | Unified Modeling Language |
| IHM | Interface homme-machine |
| MVC | Le Modèle-Vue-Contrôleur (en abrégé MVC) est une [architecture](http://fr.wikipedia.org/wiki/Architecture_logicielle) basée sur une méthode de [conception](http://fr.wikipedia.org/wiki/Conception_(logiciel)) qui organise l'[interface homme-machine](http://fr.wikipedia.org/wiki/Interface_homme-machine) (IHM) d'une application [logicielle](http://fr.wikipedia.org/wiki/Logiciel). Ce [paradigme](http://fr.wikipedia.org/wiki/Paradigme_(programmation)) divise l'IHM en un [modèle](http://fr.wikipedia.org/wiki/Mod%C3%A8le_de_donn%C3%A9es) (modèle de données), une [vue](http://fr.wikipedia.org/wiki/Vue_(informatique)) (présentation, interface utilisateur) et un contrôleur (logique de contrôle, gestion des événements, synchronisation), chacun ayant un rôle précis dans l'interface (wikipedia.org). |
| Patron de conception | Un patron de conception est issu de l'expérience des concepteurs de logiciels[2](http://fr.wikipedia.org/wiki/Patron_de_conception#cite_note-2). Il décrit sous forme de diagrammes un arrangement récurrent de rôles et d'actions joués par des modules d'un logiciel, et le nom du patron sert de vocabulaire commun entre le concepteur et le programmeur[3](http://fr.wikipedia.org/wiki/Patron_de_conception#cite_note-3). D'une manière analogue à un patron de couture, le patron de conception décrit les grandes lignes d'une solution, qui peuvent ensuite être modifiées et adaptées en fonction des besoins[4](http://fr.wikipedia.org/wiki/Patron_de_conception#cite_note-4).  Les patrons de conception décrivent des procédés de conception généraux et permettent en conséquence de [capitaliser](http://fr.wikipedia.org/wiki/Capitalisation_des_connaissances) l'expérience appliquée à la [conception de logiciel](http://fr.wikipedia.org/wiki/Conception_de_logiciel). Ils ont une influence sur l'[architecture logicielle](http://fr.wikipedia.org/wiki/Architecture_logicielle) d'un [système informatique](http://fr.wikipedia.org/wiki/Syst%C3%A8me_informatique).  (Wikipédia) |
| IHM | Les '**interactions homme-machine** (IHM), définissent, les moyens et [outils](http://fr.wikipedia.org/wiki/Outil) mis en œuvre, afin qu'un humain puisse contrôler et communiquer avec une [machine](http://fr.wikipedia.org/wiki/Machine). Les [ingénieurs](http://fr.wikipedia.org/wiki/Ing%C3%A9nieur) en ce domaine étudient la façon dont les humains interagissent avec les ordinateurs ou entre eux à l'aide d'ordinateurs, ainsi que la façon de concevoir des systèmes qui soient [ergonomiques](http://fr.wikipedia.org/wiki/Ergonomie), efficaces, faciles à utiliser ou plus généralement adaptés à leur [contexte](http://fr.wikipedia.org/wiki/Contexte) d'utilisation.  (Wikipedia) |

## Références

* Le site web du professeur Leshob Abderrahmane: <http://www.moodle.uqam.ca/> - Note du cours INF5153
* Document de présentation de l'application
* <http://granddictionnaire.com/>, Le Grand Dictionnaire de la langue française, page consultée le 2011-01-27, [en ligne], *Institut*, Mar 8, 2002, 4 p.
* LAROUSSE, *Le Petit Larousse illustré*, 2004, 1823 p.
* Software Engineering Standards Committee of the IEEE Computer Society, *IEEE Recommended Practice for Software Requirements Specifications*, Std 830, The Institute of Electrical and Electronics Engineers, Inc., 1998, 37 p.
* Software Engineering Standards Committee of the IEEE Computer Society, *IEEE Standard Glossary of Software Engineering Terminology*, Std 610.12, The Institute of Electrical and Electronics Engineers, Inc., 1990, 84 p.
* [LAR05] LARMAN, CRAIG – UML 2 et les designs patterns – Troisième édition, Pearson Éducation, 2005

## Structure du document

Le document commence avec une description sommaire du système suivie par les diagrammes de classes avec les différents patrons de conception et leurs rôles en justifiant leur utilisation dans la conception logiciel du Métronome.

Ensuite, on présente les diagrammes de séquences en tenant compte des patrons de conception proposés dans la conception architecturale et aussi une description courte de chaque diagramme de séquence.

# Description sommaire du système

Cette application nous permet de réaliser un métronome qui dispose plusieurs fonctionnalités de base :

Démarrer, éteindre, régler molette.

Le Métronome comporte les concepts et les actions suivants :

* démarrer : l'utilisateur presse sur le bouton Start.
* Arreter : l’utlisateur presse sur le bouton Stop.
* Regler Molette : l’utilisateur choisi la mesure de tempo en modifiant la position de la molette.
* Decrease mesure : l’utilisateur diminue la mesure à l’aide du boutton Dec.
* Increase mesure : l’utilisateur augmente la mesure à l’aide du boutton Dec.

# Diagramme de classes et patrons

## Le choix de patron de Conception Commande :

Nous avons utilisé le patron de Conception Commande  pour gérer toutes les commandes utilisées dans notre métronome (Start, Stop, FlasherLed 1, FlasherLed 2, Émettre Bip Sonore, Scruter Start, Scruter Stop, Scruter molette). Ce patron nous a permis de séparer complètement le code initiateur des actions (qui se trouve dans notre cas dans détecteur ou Cerveau(horloge), du code de l'action elle-même qui est fait dans Écran ou Cerveau(horloge).

## Le choix de patron de conception Observer :

Le patron de Conception Observer nous permet d’envoyer un signal ou avertir nos modules qui jouent le rôle d’observateurs. En cas de notification, les observateurs effectuent alors l’action adéquate en fonction des informations qui parviennent depuis les modules qu’ils observent appelés les Observables.

Le patron de Conception Observer a été utilisé entre la classe Détecteur et la classe Cerveau(horloge). Le Détecteur est l’objet observé et il notifie, lors d’un changement de la molette ou d’un des 4 boutons, l’observateur qui est le Cerveau(horloge). Cela permet d’éliminer un fort couplage entre ces deux classes.

## Le choix de patron de conception Singleton :

Le patron de conception Singleton a pour but de limiter l’instanciation d’une classe à un seul objet et la rendre accessible à toutes les classes de l’application. Il est utilisé lorsque l'on a besoin d'exactement un objet pour coordonner des opérations dans un système. Ce patron nous permet de nous assurer qu’il n’y aura, par exemple, qu’un et un seul un écran et qu’il n’y aura qu’un seul Cerveau(horloge) par exemple.

Nous nous sommes aperçu qu’il n’était pas utile d’implémenter ce patron dans toutes les classes. Nous avons implémenté ce patron dans la classe écran car nous ne voulions qu’un écran à tout moment et que ce patron facilitait l’appel de l’instance. Nous avons implémenté ce patron dans Cerveau(horloge) car l’instance est utilisée souvent (la plupart des commandes y sont reliées) et que lorsque nous implémentions la mesure et par une erreur dans le code, nous créions en fait une nouvelle instance de Cerveaux(horloge) (à ce moment, nous avions deux horloge qui faisait flasher les leds à des moments différents). Bien entendu, il est possible de faire tous marcher sans le patron singleton mais il facilite le codage car on est sûr de reférer au seul et unique Cerveau(horloge).

## Le choix de patron de DAO :

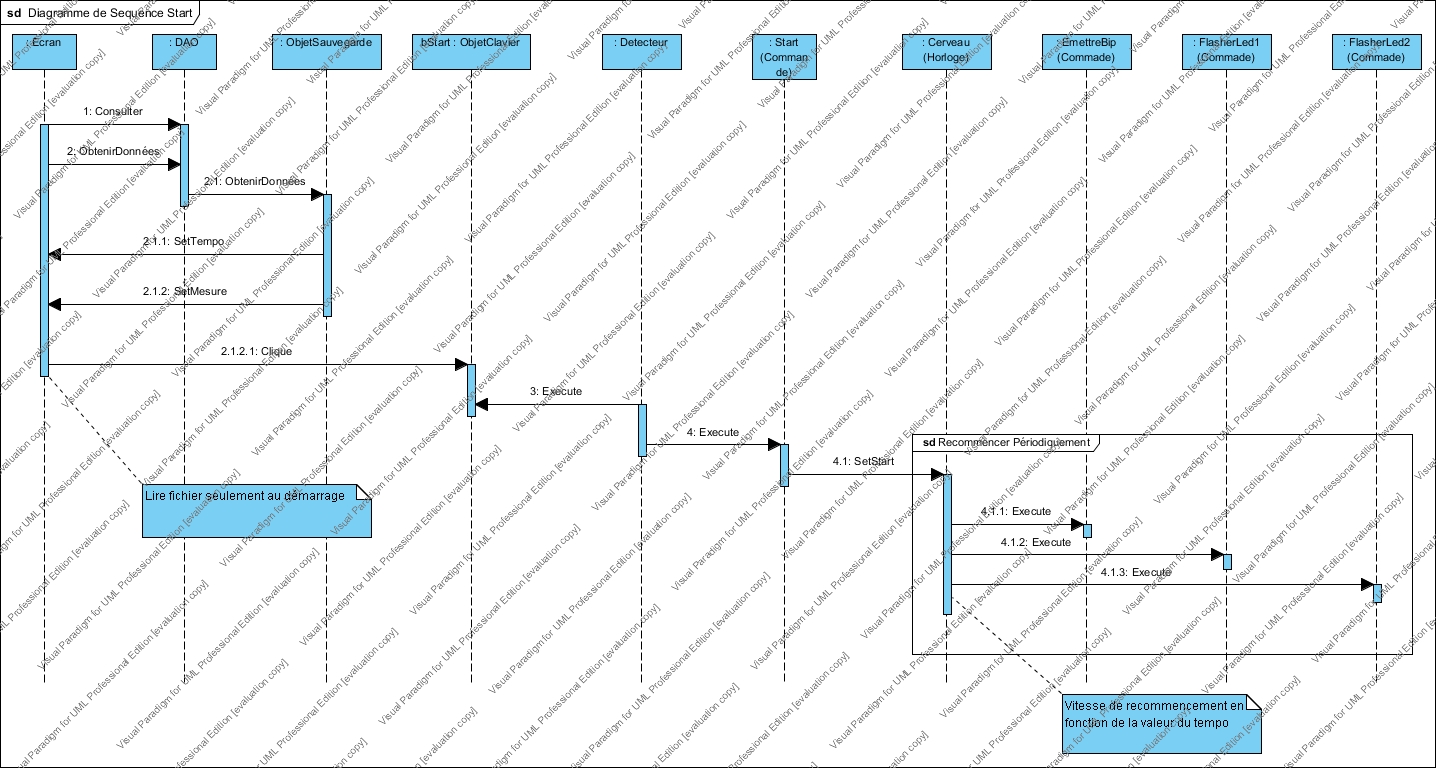
Le patron DAO fournira des opérations de manipulation de données persistantes liée aux objets du métronome (tempo et mesure). Nous utilisons le patron DAO pour regrouper les accès aux données stockés dans le fichier "Sauvegarde.txt" dans des classe à part. Ceci nous permet de ne pas écrire ce code dans les classes métiers. Ce patron nous permettra de changer la manière de sauvegarder les 2 informations du métronome.

## C:\Users\User\AppData\Local\Temp\Diagramme de Classes-1.jpgLe diagramme de classes :

# Diagrammes d’interaction

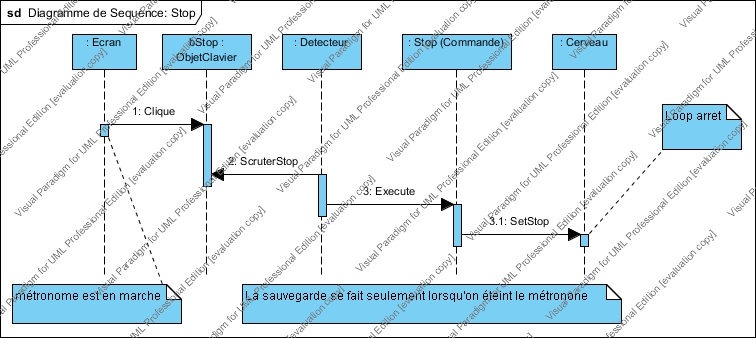
**4.1 Diagramme de séquence bouton *Start* pressé**

L'utilisateur presse le bouton Start :



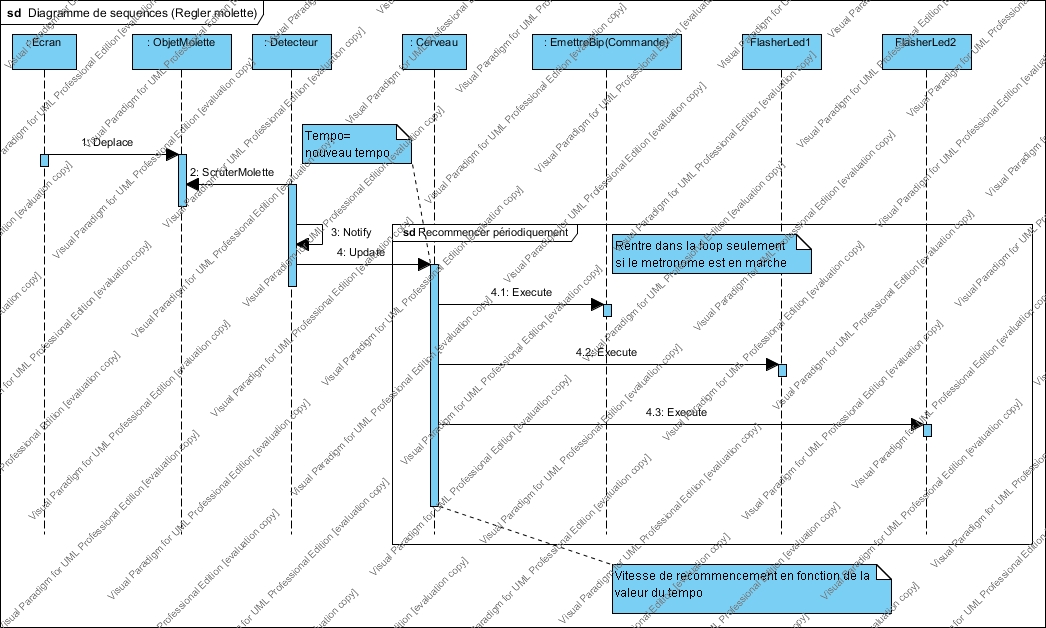
**4.2 Diagramme de séquence bouton *Stop* pressé**

L'usager presse le bouton Stop :



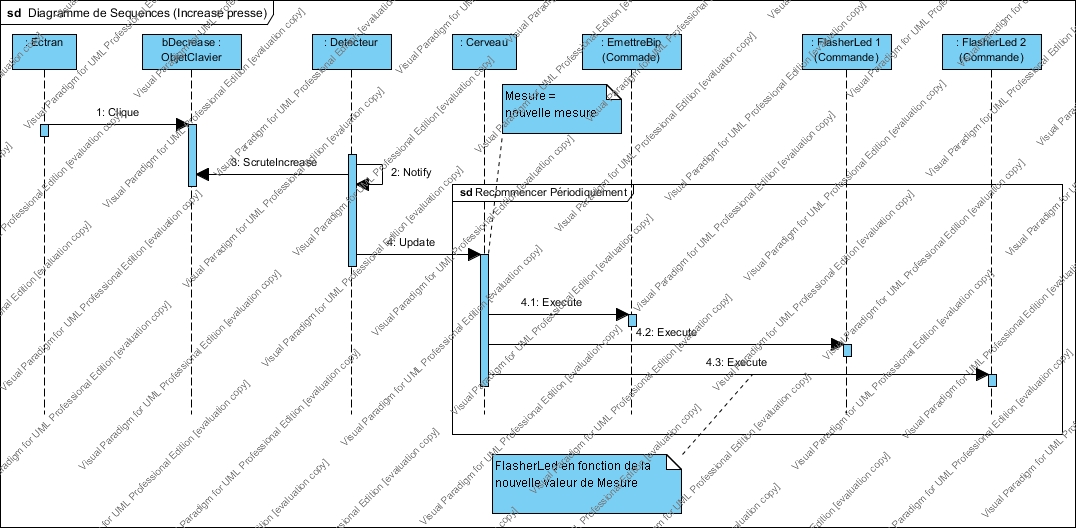
**4.3 Diagramme de séquence Régler Molette (tempo)**

L’usager choisit la nouvelle mesure en réglant la molette



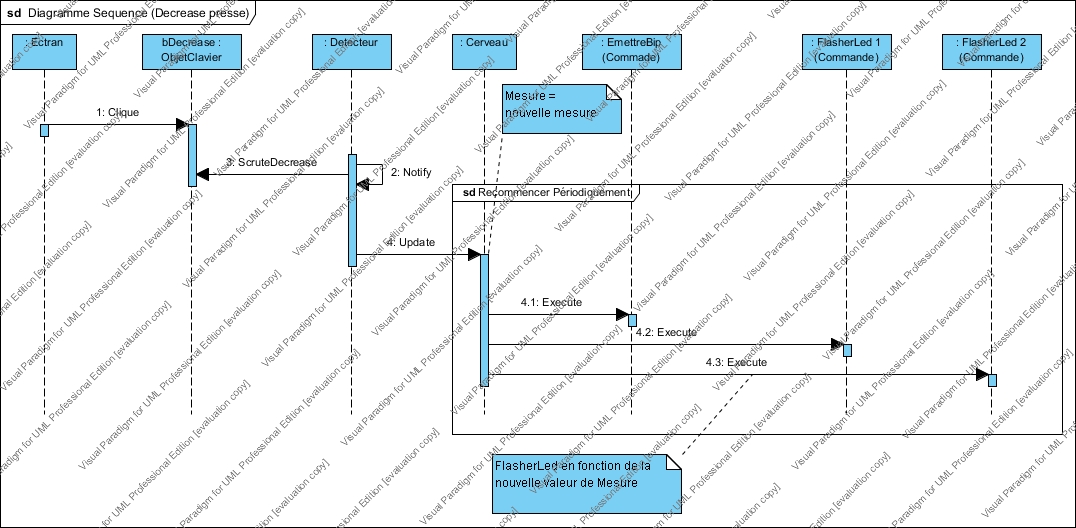
**4.4 Diagramme de séquence bouton Increase pressé (mesure)**

L’usager presse le bouton Inc.



**4.5 Diagramme de séquence bouton Decrease pressé (mesure)**

L’usager presse le bouton Dec.



# Appendices

N/A