

# **Projet Intelligence Artificielle**

## **Document d'architecture du logiciel**

# Sommaire

|     |                               |   |
|-----|-------------------------------|---|
| 1.  | Introduction                  | 3 |
| 2.  | Objectif du logiciel          | 3 |
| 2.1 | Contexte                      | 3 |
| 2.2 | Caractéristiques utilisateurs | 3 |
| 2.3 | Besoins fonctionnels          | 4 |
| 3.  | Structure et vue des couches  | 6 |

# Document d'architecture du logiciel

## 1. Introduction

Ce document a pour vocation de décrire l'architecture générale du programme afin de faciliter la mise en place par la suite. Nous verrons ici les différentes couches de notre logiciel ainsi que les différents cas d'utilisation à développer.

Ce document s'appuie sur le FRS, référencé : FRS\_ProjetIA\_Eche\_Riviere.doc

## 2. Objectif du logiciel

### 2.1 Contexte

Ce projet s'inscrit au sein d'un projet d'intelligence artificielle se déroulant au cours de l'été 2011. L'objet de celui-ci est de créer un moteur de simulation afin de répondre à une problématique sélectionnée par les différents groupes.

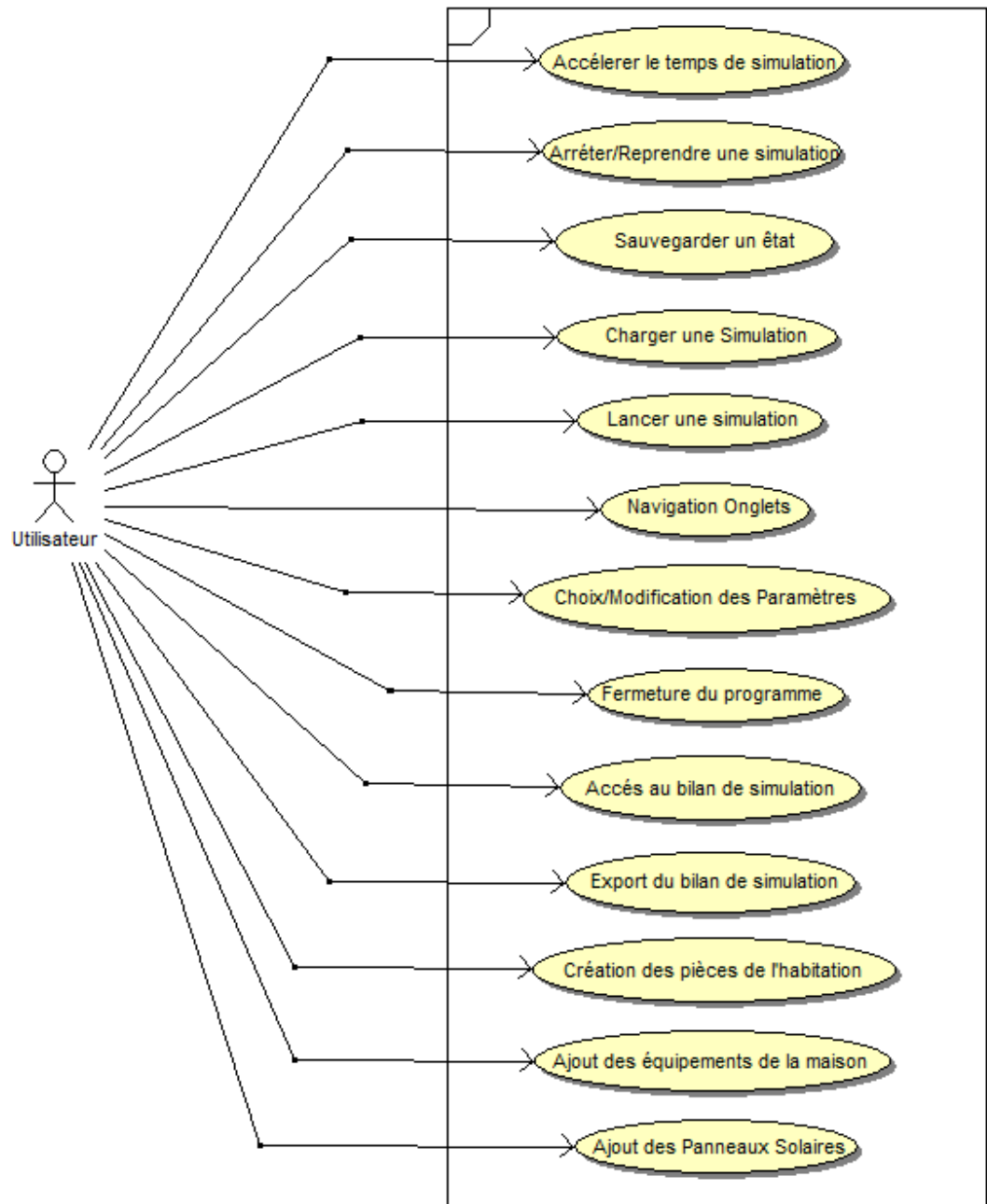
Dans notre cas, nous avons décidé de développer un logiciel de simulation de consommation énergétique d'une habitation ainsi que le gain que peut apporter des panneaux solaires sur celle-ci.

### 2.2 Caractéristiques utilisateurs

L'utilisateur est imaginé comme un commercial, vendeur de panneaux solaires qui se sert de ce logiciel afin de convaincre ses prospects d'installer des panneaux sur leur toit.

L'utilisateur n'a donc aucune compétence techniques et doit bénéficier d'une interface intuitive afin que sa démonstration se déroule au mieux. De plus, afin de convaincre de possibles futurs clients, il est nécessaire d'avoir une interface agréable qui soit vendeuse.

## 2.3 Besoins fonctionnels

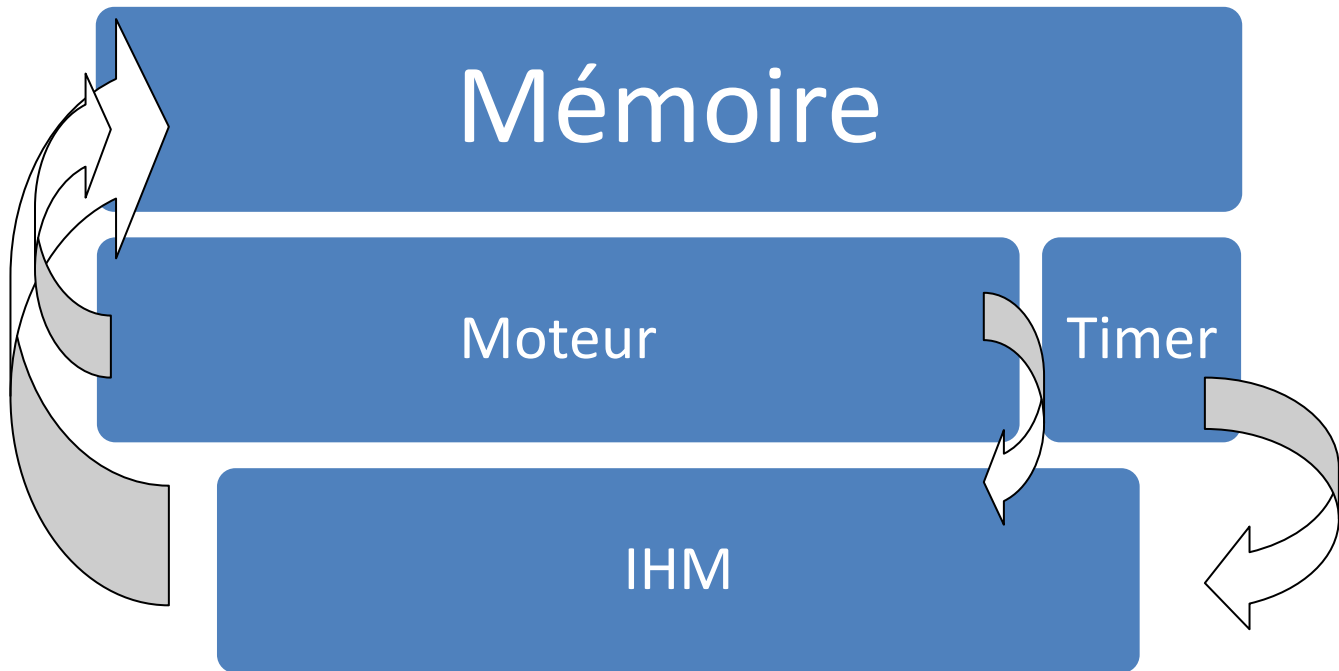


| Identifiant | Titre                                     |
|-------------|---|
| UC001       | Création d'une habitation                 |
| UC002       | Equipement d'une habitation               |
| UC003       | Ajout de panneaux solaires                |
| UC004       | Lancement d'une simulation                |
| UC005       | Modification des paramètres de simulation |
| UC006       | Arrêter/Reprendre une simulation          |
| UC007       | Modifier le temps de simulation           |
| UC008       | Sauvegarder une simulation                |
| UC009       | Chargement une simulation                 |
| UC010       | Navigation par onglet dans l'IHM          |
| UC011       | Accéder au rapport de simulation          |
| UC012       | Export d'un rapport de simulation         |
| UC013       | Fermeture du programme                    |

*Cette liste constitue un seul rappel des cas d'utilisation, leur description et déroulement est explicité dans le FRS.*

### 3. Structure et vue des couches

Le programme se décomposera des couches suivantes :



Ce découpage par couche a plusieurs vocations. La toute première est de préserver à tout moment l'intégrité de la mémoire, partie la plus critique de notre programme. En isolant cette couche on obtient ainsi davantage de sécurité.

La deuxième étape est de séparer la gestion du temps du moteur. Le moteur va générer une file d'événement correspondant à la totalité de notre simulation que le timer va afficher suivant le rythme choisit. Nous aurons donc une file d'événement totale correspondant à l'intégralité de notre simulation mais avec une vitesse d'affichage différente.

Dernièrement, parlons effectivement de cet affichage. Celui-ci est totalement séparé des autres couches afin de pouvoir modifier l'interface graphique sans avoir à modifier le code sur laquelle elle s'appuie. Nous pouvons alors imaginer avoir un logiciel multi plateforme et compatible à différents types d'écrans, où nous aurions juste à porter et modifier l'IHM.

Par ailleurs, ce découpage par couches nous permet de correctement fractionner le travail ainsi que de facilement réaliser test unitaires et tests d'intégration.

La mémoire est constituée d'un ensemble de fichier binaire ou CSV. Les fichiers CSV servant à compléter manuellement et dans un tableur nos différentes bases de données (normales météorologiques, équipements des habitations, consommations énergétiques). Les fichiers binaires serviront pour les fichiers de sauvegarde.

L'IHM s'appuiera sur la librairie GTK+ avec le plugin drawing afin de pouvoir tracer les habitations.

Le timer se réalisera à partir de la commande processeur sleep en C.

Le moteur de simulation fonctionne à partir de données réelles s'appuyant sur des sources (articles,...).

La création d'événement fonctionne comme indiqué dans le document : PROJET IA P20 Simulation.pdf