

Projet de Session  
Développement d'application interactives III

Remis à  
Nicolas Bourré

Réalisé par  
Martin Ricard

Remis le  
08 Décembre 2017

## Table des matières

Description du projet.....	3
Diagramme de Classes.....	3
Patron de Conception.....	3
La Fabrique .....	3
L'état.....	3
La méthode.....	3
ANNEXES.....	4
Diagramme de Classe .....	5
La Fabrique .....	5
L'état.....	6
Patron de méthode .....	6

## Description du projet

Ce projet est un simulateur de saut à l'élastique. Un objet est attaché à un élastique et se dirige vers le bas de l'écran. En chemin, il croise un filet. Si l'objet touche le bas de l'écran, qui simule le sol, celui meurt. Les sujets utilisés sont : l'oscillation représenté par les différents élastiques, un système de particules qui est représenté par le sang de la victime si elle décède, les forces qui est représenté à plusieurs endroits et finalement les collisions que l'on peut voir lorsque l'objet touche le filet ou encore le bas de l'écran.

## Diagramme de Classes

Voir section Annexes.

## Patron de Conception

### La Fabrique

Dans ce projet, j'ai inclus le patron de la fabrique car j'ai deux classes qui se ressemblent, mais qui ne sont pas les mêmes. J'aurais pu utiliser la même classe en ajoutant des variables, mais pour la maintenabilité cela devenait un enfer, donc avec le patron de la fabrique, cela empêchait cela de se produire. Le diagramme de ce patron se retrouve dans la section Annexes.

### L'état

Afin d'assurer une méthode de développement optimale, l'implantation de l'état s'avérait un choix judicieux. En effet, en utilisant ce patron, cela permettait de donner un aperçu du projet à l'auditoire avant qu'il démarre. De plus, cela permet d'initialiser les objets sans utiliser des ressources d'affichage inutiles. Le diagramme de ce patron se retrouve dans la section Annexes.

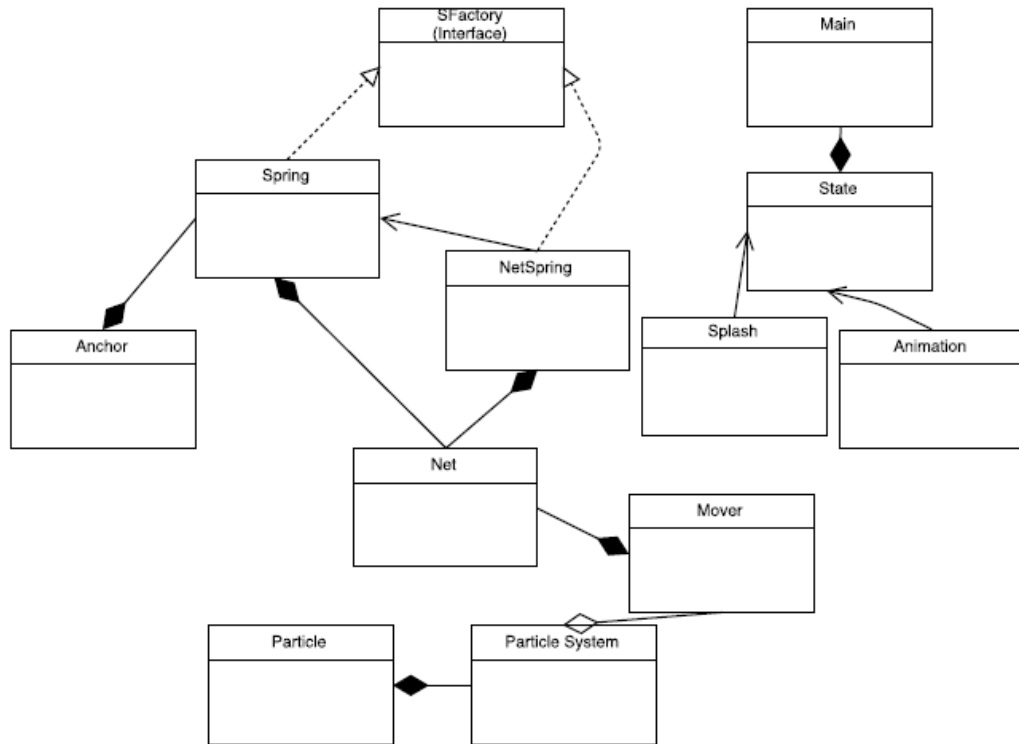
### La méthode

Le patron de la méthode a été utilisé dans ce projet afin de différencier deux classes qui servent à faire des choses semblables. Ce sont les mêmes classes créées par la fabrique. Une de ces deux classes possède des attributs différents de l'autre et en utilisant ce patron, il devenait plus facile de maintenir le code de ses classes et de comprendre le fonctionnement de ces dernières.

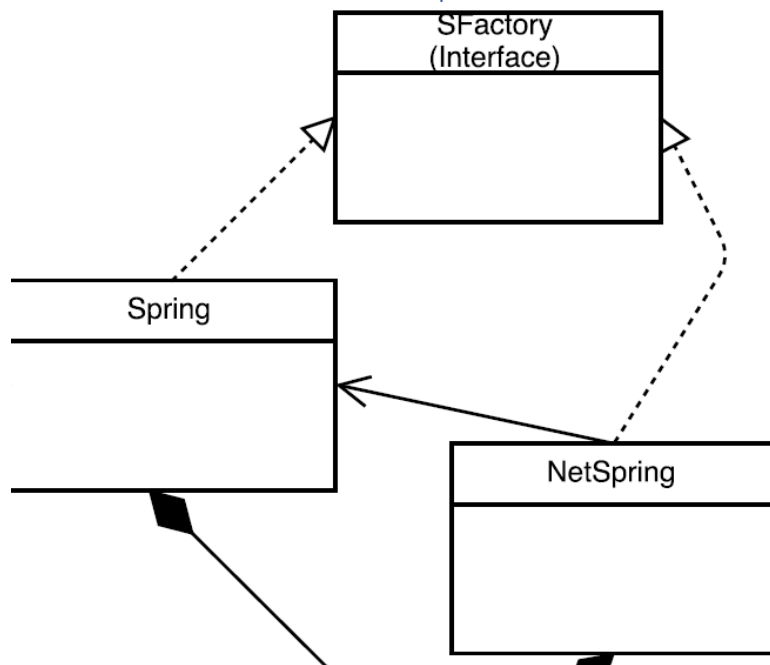
Le diagramme de ce patron se retrouve dans la section Annexes.

# ANNEXES

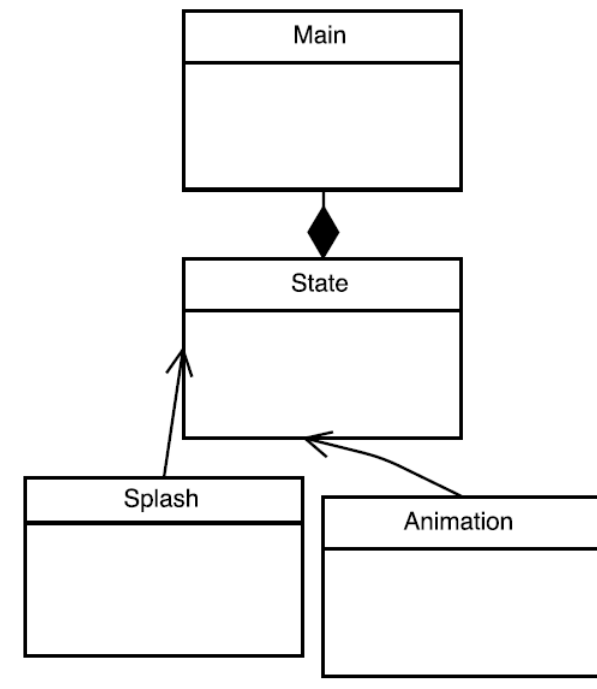
## Diagramme de Classe



## La Fabrique



### L'état



### Patron de méthode

