

# Traitement d'images et Développement d'application interactive III

Projet de session présenté à  
Nicolas Bourré

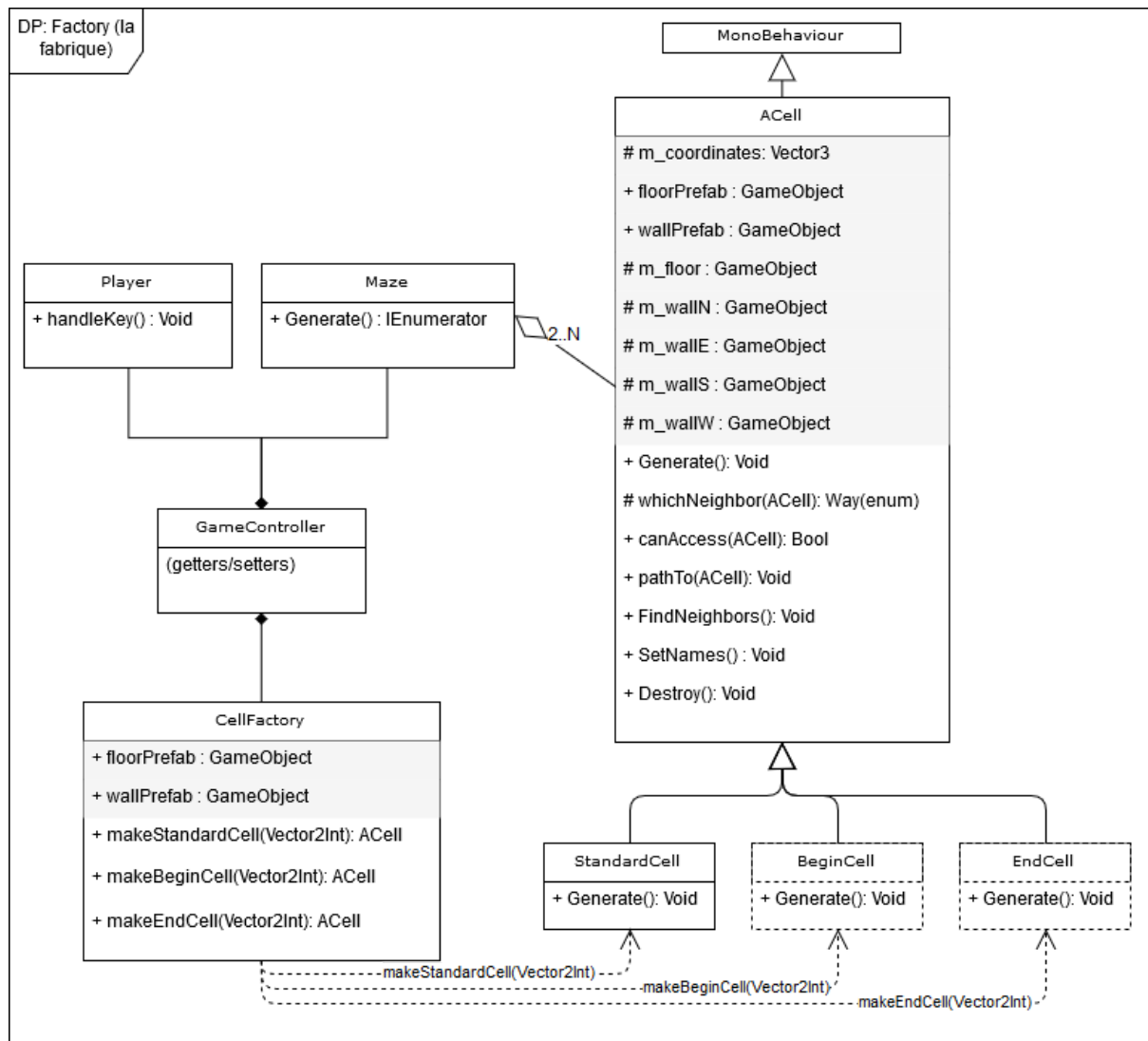
Par  
Maxime Damour

le 8 Décembre 2017

**Sujets:**  
Génération de labyrinthe  
Algorithme de recherche A\*  
Oscillations  
Collisions  
Unity3D

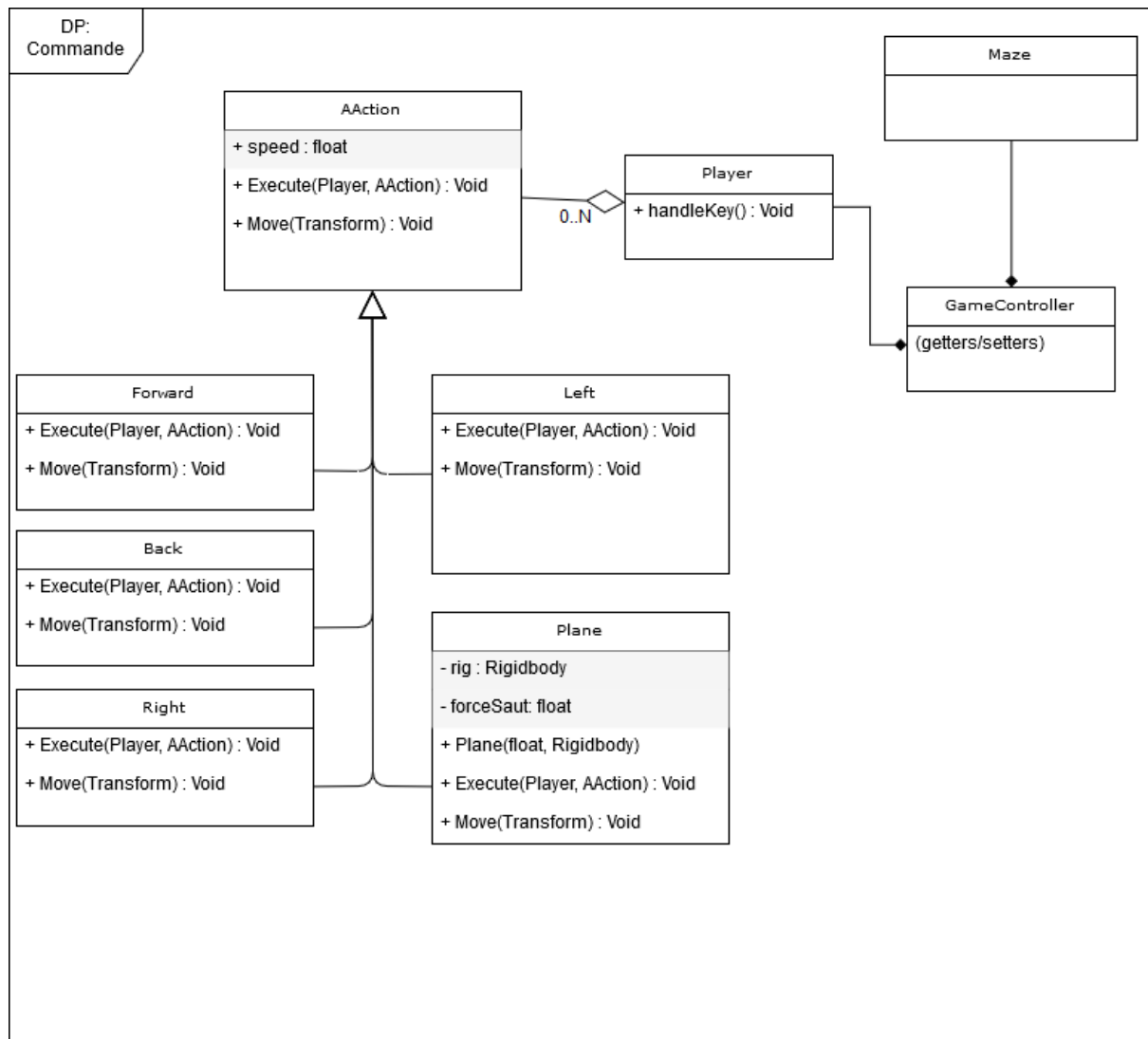
# Présentation des patrons de conception

## La fabrique:



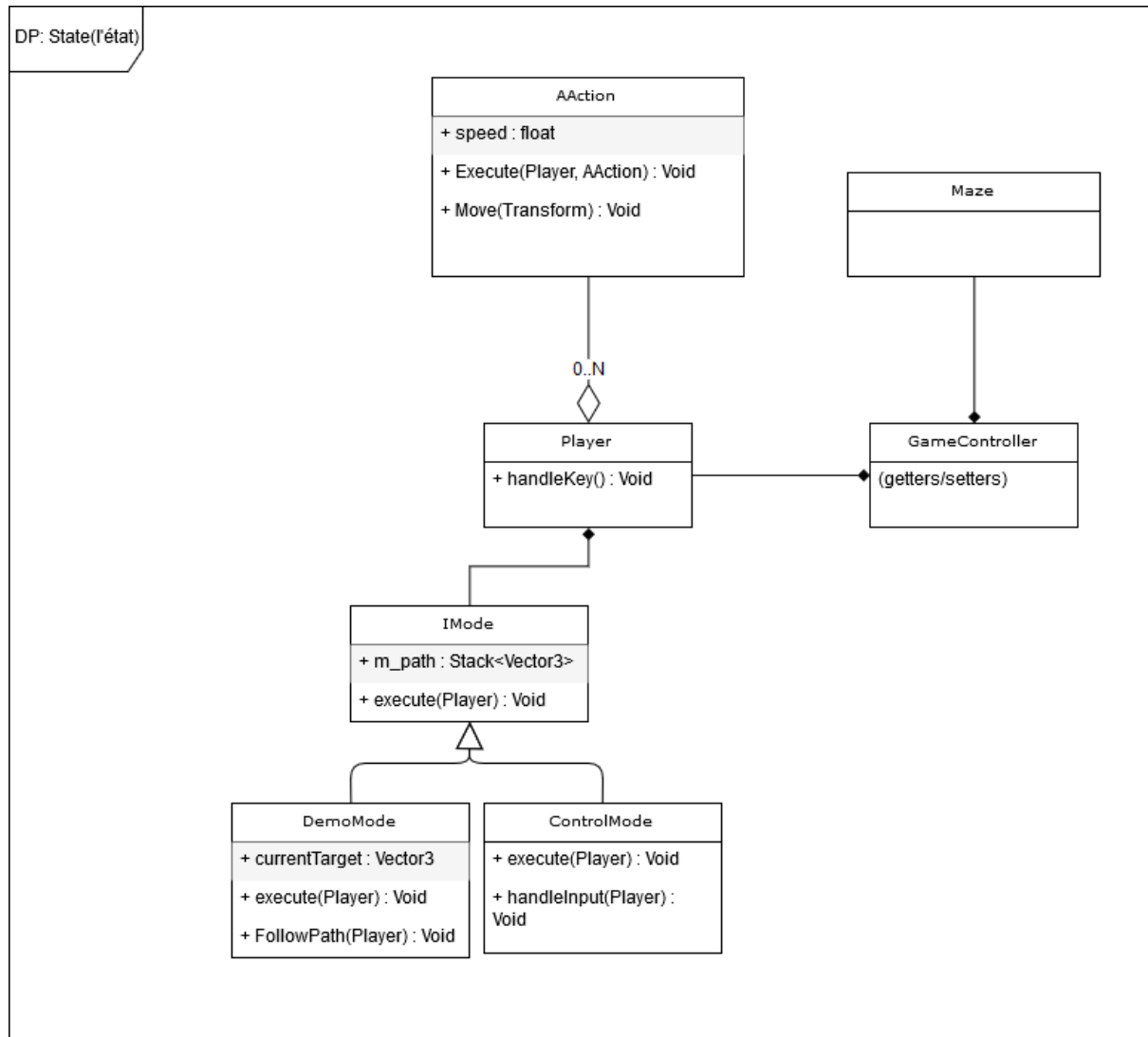
Les pièces les plus importantes du maze sont sûrement les cellules de celui-ci. Pour rendre la réutilisation du code ainsi que la compréhension, le patron de conception de la fabrique a été utilisé pour créer les cellules. Cette fabrique se trouve dans le GameController, qui gère le jeu, et est donc accessible par le Maze qui l'utilise pour créer toutes les cellules. Les classes en pointillés sont des exemples de classe qui peuvent être implémentées. Basiquement, il n'y a que la fonction `Generate()` qui dépend entièrement de l'enfant de la classe abstraite **ACell**.

## La commande:



Le patron de conception de la commande est tout à fait approprié pour le type d'application présent. En effet, l'utilisateur doit être capable de bouger dans l'espace, mais le code en arrière doit rester lisible et facile à maintenir. Ici, il serait facile de rajouter, par exemple, une commande pour que le joueur se baisse, ou encore une autre pour qu'il courre. À chaque fois que l'usager appuie sur une touche, si celle-ci fait partie des commandes possible, l'action liée à cette touche va pouvoir s'exécuter.

## L'état:



Le patron de conception de l'état permet ici, notamment, de passer ou en en mode démo. À chaque frame, l'objet player va chercher à exécuter son état présent. Quand l'utilisateur actionne une commande (dans la classe Player) et que son état est "ControlMode", la sous classe de l'état va exécuter l'action liée à la commande, et le joueur pourra ainsi bouger. Dans le mode auto/démo, un chemin sera déterminé lors de l'appuie pour changer d'état, et le joueur suivra ce chemin jusqu'à la fin.

## Références:

- <https://docs.unity3d.com/ScriptReference/index.html>
  - Une source immanquable dans la mesure où toute l'application a été développée sur cette plateforme.
- <https://answers.unity.com/questions/390075/move-an-object-in-a-sine-wave-oscillating-around-a.html>
  - L'idée de faire une oscillation qui se déplace en fonction m'est venue d'ici, mais l'inspiration du code était sur les documents de cours
- <http://natureofcode.com/book/>
  - Une sources très importantes concernant les lois de la physique appliquées au jeu vidéo, même si l'on ne développe pas sur processing, ce livre est très intéressant.
- <http://catlikecoding.com/unity/tutorials/>
  - Des tutoriaux très intéressant quand on débute sur la plateforme Unity.
- Les notes de cours
  - Notamment pour le calcul du chemin avec la méthode A\*