



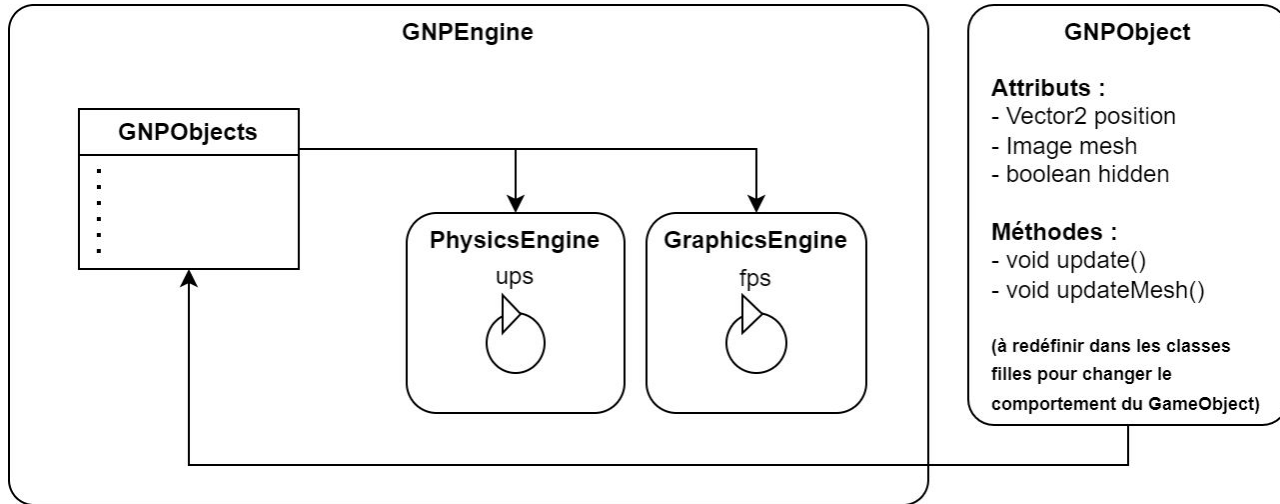
Projet JAVA GNPEngine

Création d'un moteur physique et graphique

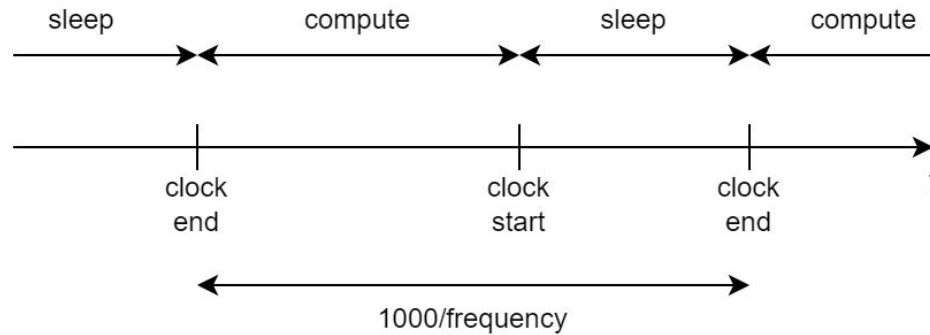
Restructuration complète

- Ajout d'une interface graphique utilisant Java Swing
- Multi-threading du moteur physique et du moteur de rendu
- Ajout d'une caméra pour pouvoir se déplacer dans la vue
- Standardisation du moteur et de son fonctionnement pour le rendre indépendant et réutilisable dans d'autres projets
- Applications de démonstration du moteur :
 - Maze : Génération et résolution de labyrinthe
 - Jeu de la vie : Simulation du Jeu de la vie de John Conway

Architecture du projet

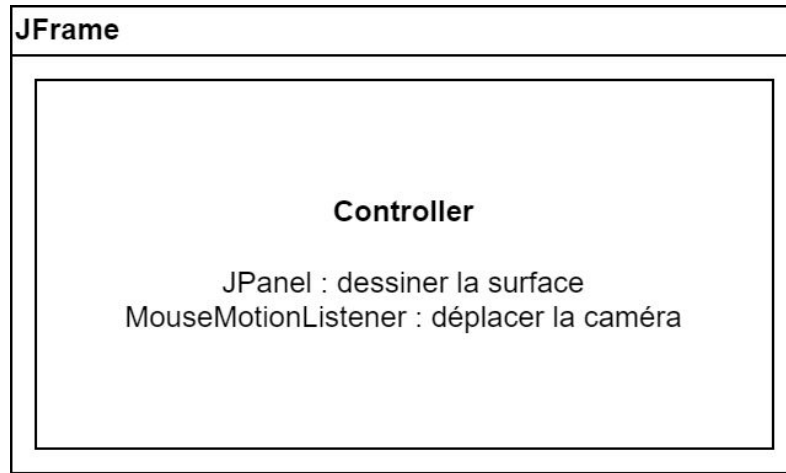


Fonctionnement de la Clock



$$\begin{aligned}\text{sleep} &= 1000/\text{frequency} - \text{compute} \\ &= 1000/\text{frequency} - (\text{end} - \text{start})\end{aligned}$$

Fonctionnement du GraphicsEngine



DISPLAY

- 1 - Nouvelle surface (BufferedImage)
- 2 - Pour tous les GNPObjets :
 - updateMesh
 - le dessiner sur la surface
- 3 - Controller.updateSurface()
- 4 - Controller.repaint()

Objectifs futurs

- GraphicsEngine :
 - Migration de l'interface graphique vers LWJGL (permettant ensuite la gestion d'un environnement 3D)
 - Ajout d'un système de caméras multiples
- PhysicsEngine : Ajout de nouveaux éléments au moteur (collisions de formes basiques, gravité, etc...)
- UI :
 - Ajout d'éléments d'interface utilisateur (boutons, champs de texte, etc...)
 - Meilleure gestion des inputs
- Networking : Version client/serveur pour faire travailler plusieurs moteurs en réseau, avec plusieurs mode de calcul (peer2peer, server side)