Notes à propos de la conception

Vous trouverez deux fichiers PDF en plus de celui-ci : *Séquences.pdf* et *Classes.pdf*.

Le fichier contenant le diagramme de séquence explicite le fonctionnement de la boucle principale du jeu.
Nous nous sommes inspirés de la boucle qui fait partie de nombreux jeux vidéos, elle permettra notamment de gérer un temps en tours de boucle (frames).

Nous avons essayé de bien distinguer les différents packages de l'application.
La partie la plus compliquée est sans doute celle qui consiste à séparer l'affichage du reste.

Pour ce faire, nous avions 2 solutions possibles :
- faire une méthode afficher() pour chaque objet Affichable (donc une interface Affichable avec une fonction afficher que chaque objet implémente).
- Fabriquer un objet Afficheur représentant l'objet à afficher et permettant de l'afficher.

Nous avons choisi la deuxième solution car elle permet de séparer totalement la vue du reste du programme. De plus nous pouvons modifier à la volée l'affichage d'un objet (grâce à la création de nouveaux afficheurs).

Pour exécuter cette solution nous avons utilisé le design pattern Factory (FabriqueAfficheurs) qui permettra de créer les objets gérant l'affichage en fonction d'un objet Affichable.
Cette fois-ci Affichable joue simplement le rôle de marqueur (comme vu en cours) et il n'y a pas de méthode à implanter.
Nous n'avons pas mis de flèche d'implantation entre plateforme, personnage, attaque et affichable par soucis de clarté.

Puis, le programme ayant deux modes de jeu. Nous avons fait deux classes qui héritent de Jeu et qui définissent comment la façon dont une partie se termine et comment calculer le score.
Ceci permettra par la suite d'ajouter de nouveaux modes de jeu si nécessaire.

Enfin, le moteur physique a besoin de différents états pour les personnages (en l'air, au sol, attaqué, inactif …).
Nous avons pensé à utiliser le design pattern States mais nous avons peur de compliquer énormément le programme à cause des nombreuses données à traiter en fonction de chaque état.
Nous attendrons donc de voir s'il est réalisable avant de l'utiliser.