

Deuxième itération : Mario 2020

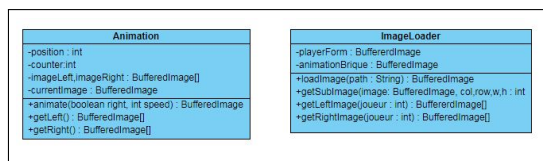
Houda.Ouhou (groupe A)

Avril 2020

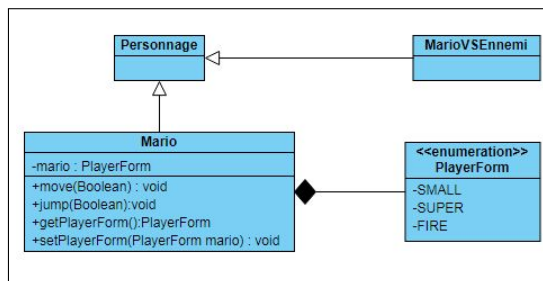
-Dans cette itération, j'ai décidé avec les membres de mon équipe d'ajouter ,et de développer le diagramme de classe sur lequel on s'est basés au début, afin d'offrir plus de fonctionnalités et d'améliorer le jeu.

-Comme prévu, j'ai continué à développer ma partie de travail qui se base sur :

-Affichage :



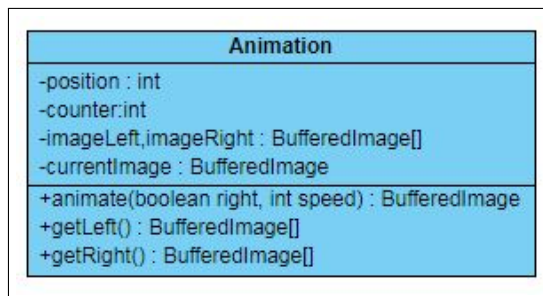
-Personnage :



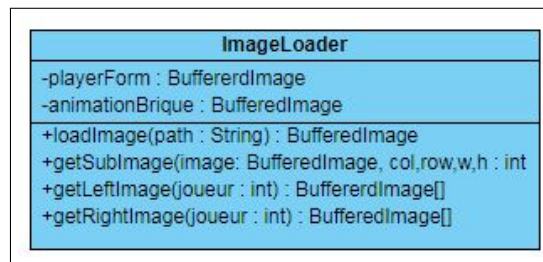
-La partie affichage: où je me suis intéressée de plus à L'animation et les sprites des personnages.

-J'ai défini, donc, deux autres classes :

*Classe **Animation** : qui utilise un tableau d'image *BufferedImage*[] regroupant l'ensemble de sprites qui illustrent le mouvement du Personnage et qui changent selon son mouvement, sa vitesse, et son gain .

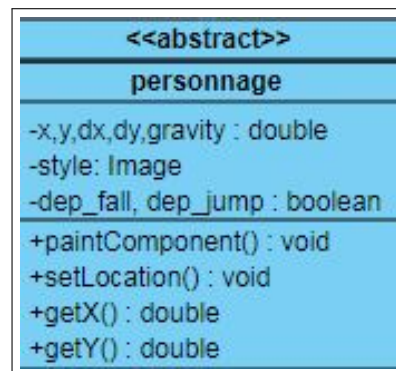


* Classe **ImageLoader** : qui génère l'ensemble d'images possible du Player selon les cas : Joueur(Mario) ou ennemi , ou bien la forme (SMALL,SUPER, ou bien FIRE)



La partie programmation de jeu: où je définis 4 différentes classes liées à la notion du Joueur :

* Classe *<< abstract >>* **Personnage**: que j'ai développé par rapport à la première itération , en ajoutant des nouveaux attributs et de méthodes .

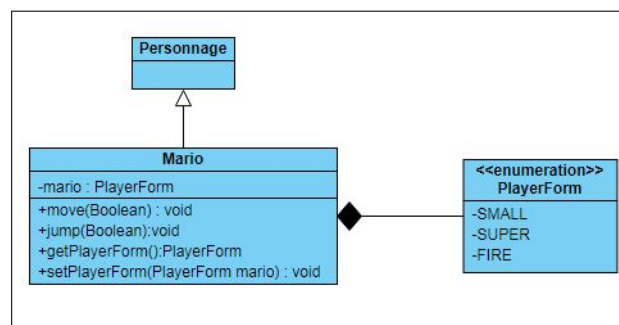


-Comme vous pouvez remarquer dans cette classe, de plus de la position du joueur (x, y) , j'ai ajouté des attributs dx, dy, gravité, et deux booléens qui gèrent tous les types de mouvement possibles du joueur.

-Il y a aussi les méthodes: paintComponent() qui dessine le sprite du personnage sur le JPanel, setLocation() qui permet de modifier sa position , ainsi que getX() et getY() qui renvoient ses coordonnées.

* Classe **Mario**: qui hérite de la Classe Personnage, et qui permet de définir le joueur de la partie via un attribut mario de type PlayerForm.

- En plus des opérations définies dans La super Classe Personnage, j'ajoute : getPlayerForm() qui renvoie la forme courante du joueur, setPlayerForm(PlayerForm mario) qui permet de modifier sa forme, move() et jump() qui permettent de modifier le mouvement de mario selon les cas.



* Classe *<< enumeration >>* **PlayerForm**: qui est une composante importante de la classe Mario :

Dans cette classe, je regroupe les formes générales du personnage mario : SMALL, SUPER, et FIRE (qui illustre le personnage mario vis-à-vis l'ennemi illustré dans la classe MarioVSEnnemi).



* Classe [MarioVSEnemi](#):

cette dernière classe hérite aussi de Personnage. Mais sa particularité est qu'elle définit un modèle différent de mario: Mario dans le cas de défense face à un ennemi quelconque.

