

## Rapport itération 2

-Programmation du code et ajout de la javadoc dans les classes :

-Actor, modification des types pour les attributs position et scale (int->float), ajout de l'attribut components qui contient tout les ActorComponent que l'acteur possède et de l'attribut direction qui donne l'orientation de l'acteur

.

-Pawn, ajout du test de collision pendant le déplacement avec les autres acteurs du monde.

-Controller, modification du getActorLocation() qui retourne la position du pion possédé si possible ou l'origine du monde sinon.

-PlayerController, prise en compte de la direction de déplacement.

-ActorComponent, création de la classe, elle représente des comportements réutilisable par différent acteur.

-SpriteAnimationComponent, création de la classe, elle représente l'acteur le possédant en dessinant une série d'image afin d'obtenir un animation.

-RectangleCollisionComponent, création de la classe, elle représente le rectangle de collision de l'acteur le possédant, il est possible de l'afficher en changeant l'attribut canDraw à true.

-SatCollisionDectection, création de la classe, elle teste si 2 polygones sont en collision en utilisant le théorème de séparation des hyperplans (cela va changer car ce théorème ne fonctionne que pour des polygones convexe).

-Map, création de la classe, elle représente le monde dans lequel le joueur évolue, ne fonctionne que avec les cartes créée par tiled exportée en json. Elle affiche les couches une par une et lorsque que le numéro de la couche est égal au numéro de la couche des acteurs ont affiche les acteurs. Il reste à rajouter les polygones pour obtenir les collisions. Optimisation à faire afin d'éviter d'utiliser des ressources inutilement.