

INP Toulouse - ENSEEIHT

COUTURIER Arthur
COQUARD Pierre-Jean
DANTON LALOY Calliopé
DE ROCKER Tom
DI MARIA Enzo
EL BSITA Yassir
VILLEMONT-JEAN Cédric

TECHNOLOGIE OBJET

Projet Long

1ème Année

Semestre 6

05/04/2021- 29/05/2021

Rappel du sujet	3
Principales fonctionnalités	4
Architecture de l'application	5
Principaux choix de conception et de réalisation	6
Organisation de l'équipe & méthodes agiles	7

Rappel du sujet

SN StrikeBack 7 est un jeu d'artillerie dans lequel plusieurs équipes s'affrontent au tour par tour. Le but de chaque équipe est d'éliminer tous les joueurs des équipes adverses.

Principales fonctionnalités

Les menus (Réalisé à l'itération 2 : Statut Terminé)

Nous avons implanté plusieurs fonctionnalités afin de d'améliorer l'expérience utilisateur. Tout d'abord nous avons mis en place les menus.

On retrouve ainsi différent menu dans lesquels l'utilisateur peut naviguer avec les touches du clavier :

- Le menu principal : L'utilisateur peut lancer une partie ou alors quitter le jeu
- Le choix de la carte : Différente carte sont proposé à l'utilisateur
- Le choix des équipes : L'utilisateur choisi les équipes participante puis valide
- Le nombre de joueur par équipe : L'utilisateur choisi le nombre de joueur par équipe
- Le menu pause : Permet à l'utilisateur d'interrompre le jeu et lui permet de reprendre ou alors de quitter le jeu

```
[SPACE] Commencer  
[ESCAPE] Quitter  
[F] Plein ecran / Fenetre
```

```
[1] La salle de TP des MF2E  
[2] L'appartement d'un eleve de MF2E  
[ESCAPE] Retour  
[F] Plein ecran / Fenetre
```

```
[1] SN  
[2] 3EA  
[3] MF2E  
[4] SHS  
[SPACE] Valider  
[ESCAPE] Retour  
[F] Plein ecran / Fenetre
```

```
[1] 1 Joueur  
[2] 2 Joueurs  
[3] 3 Joueurs  
[4] 4 Joueurs  
[ESCAPE] Retour  
[F] Plein ecran / Fenetre
```

```
[SPACE] Reprendre  
[ESCAPE] Quitter  
[F] Plein ecran / Fenetre
```

Les déplacements des personnages (Réalisé à l'itération 2 : Statut Terminé)

L'utilisateur peut déplacer son personnage avec les touches de son clavier, et notamment ZQD. Ici, Z correspond à un saut, Q et D correspondent aux déplacements horizontaux.

La gestion des équipes (Réalisé à l'itération 2 : Statut Terminé)

Nous avons implanté un système d'équipe qui permet de jouer de 2 à 16 joueurs. Il y a 4 équipes: les SN, les 3EA, les MF2E et enfin les SHS. L'utilisateur a donc le choix entre ces 4 équipes.

Affichage des informations utiles à l'utilisateur (Réalisé à l'itération 3 : Statut Terminé)

À l'écran, l'utilisateur peut voir plusieurs informations. Tout d'abord, la vie du personnage courant, son équipe, ainsi que son temps restant. Mais aussi la vie des autres joueurs.



Changement de tour (Réalisé à l'itération 2 : Statut Terminé)

Une fonctionnalité importante est le changement de tour. En effet, le joueur courant possède un temps limité pour pouvoir se déplacer et attaquer. Ainsi, le joueur courant change si le temps est écoulé ou alors s'il décide d'arrêter son tour avec la touche E.

Changement d'arme (Réalisé à l'itération 3 : Statut commencé)

Cette fonctionnalité n'est pas encore fonctionnelle, néanmoins elle pourra permettre à l'utilisateur de changer d'arme en cours de partie.

Tirer sur un joueur (Réalisé à l'itération 3 : Statut Terminé)

L'utilisateur peut via la souris tirer sur un autre joueur et lui enlever de la vie. Cela fera passer son temps de jeu restant à deux secondes.



Couper le son de la musique (Réalisé à l'itération 3 : Statut Terminé)

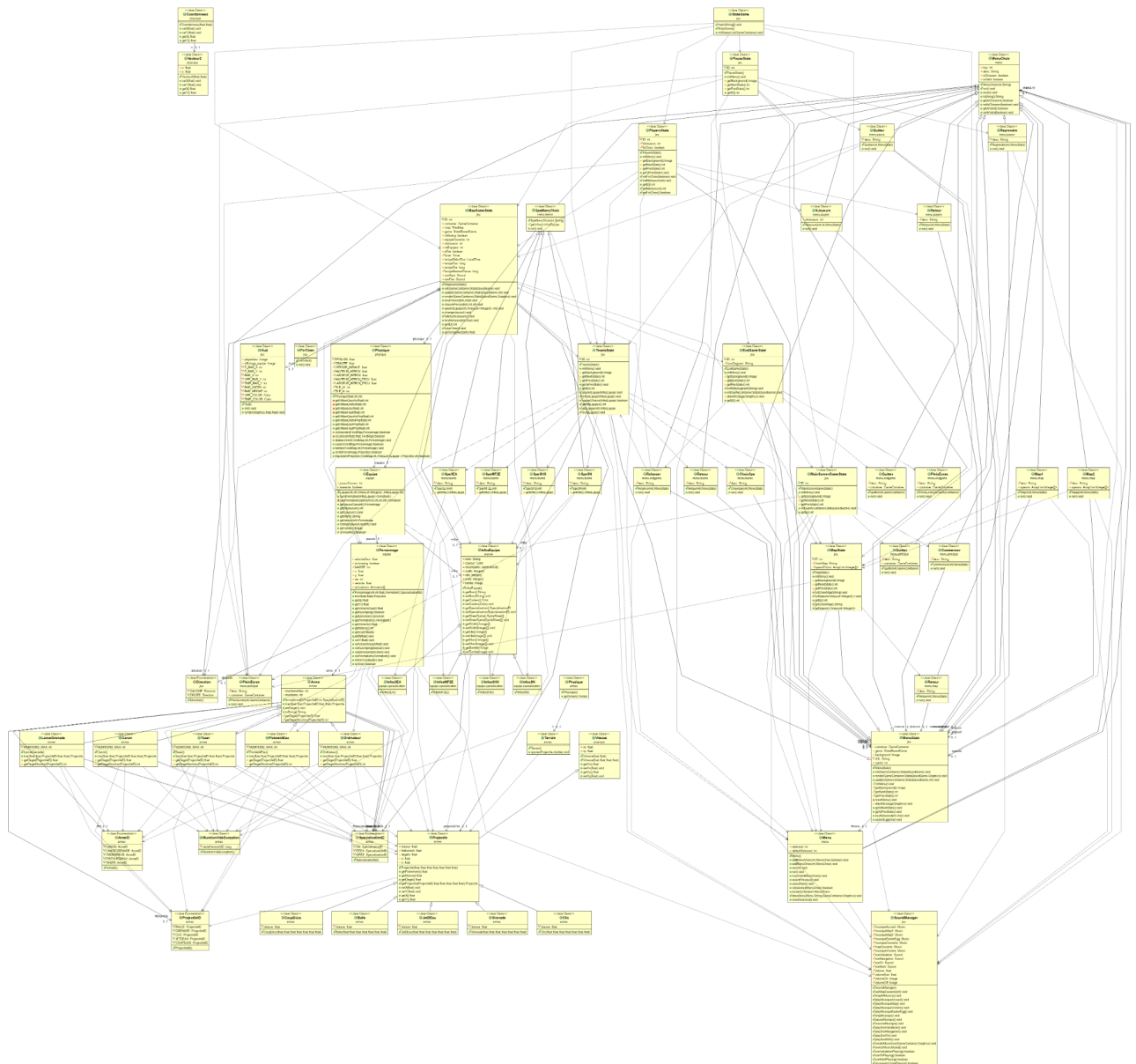
L'utilisateur peut à tout moment couper le son de la musique du jeu avec la touche M.

Passer en mode Plein écran (Réalisé à l'itération 3 : Statut Terminé)

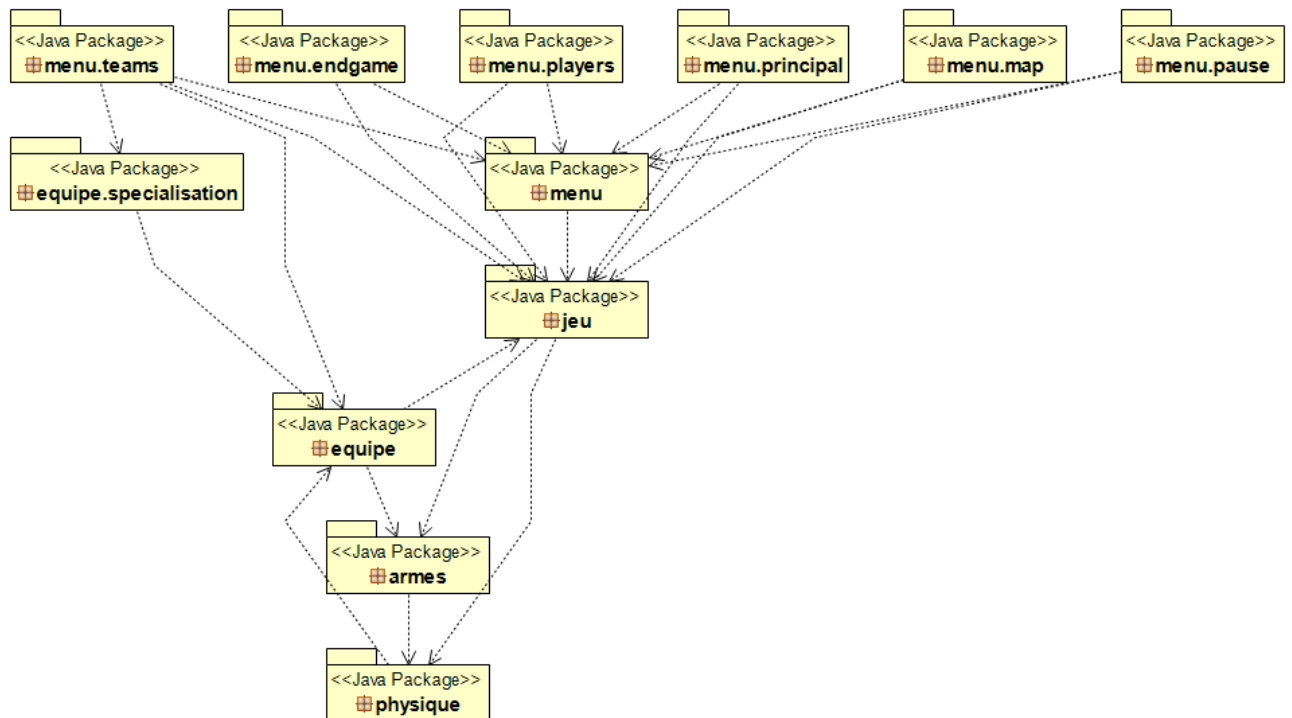
L'utilisateur peut à tout moment passer le jeu en mode plein écran ou repasser en mode fenêtré avec la touche F.

Architecture de l'application / Découpage de l'application en paquetages

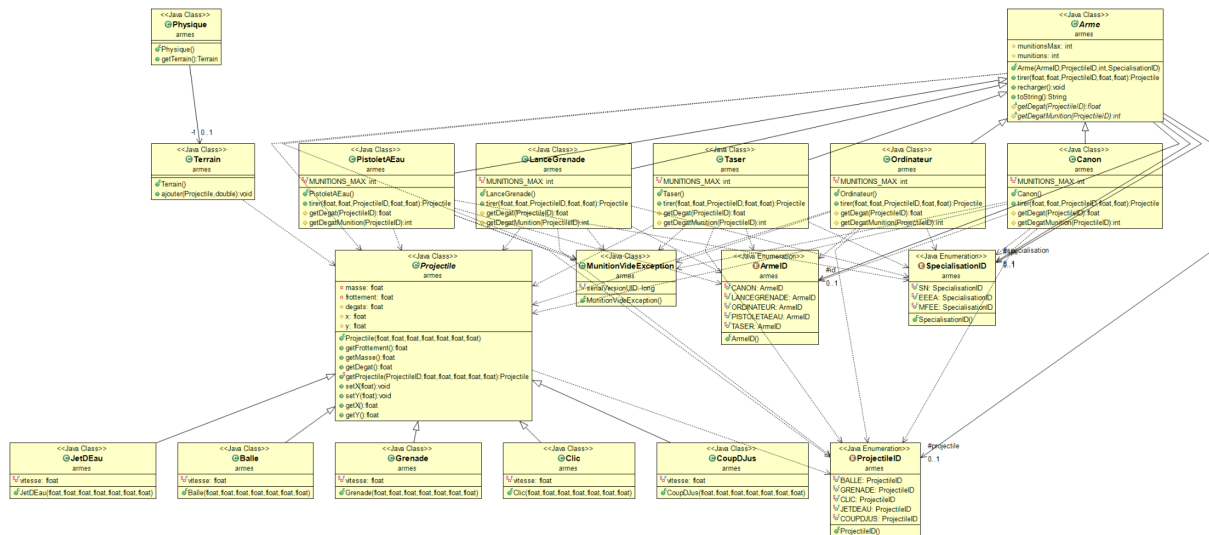
Voici le diagramme total de l'application :



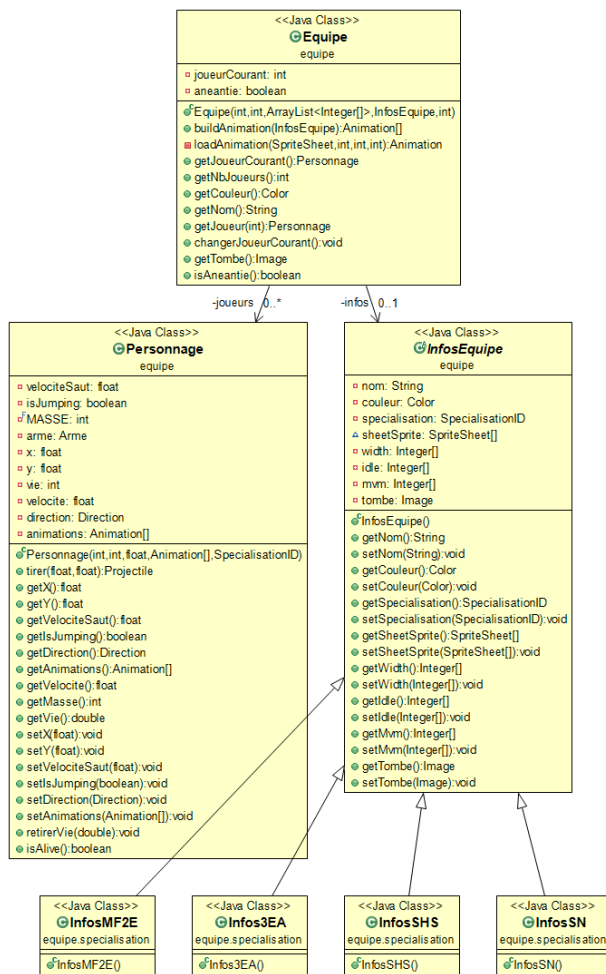
Ce diagramme étant plutôt illisible, il est plus opportun de faire un diagramme montrant les liens entre les différents packages puis des sous diagrammes de chaque package :



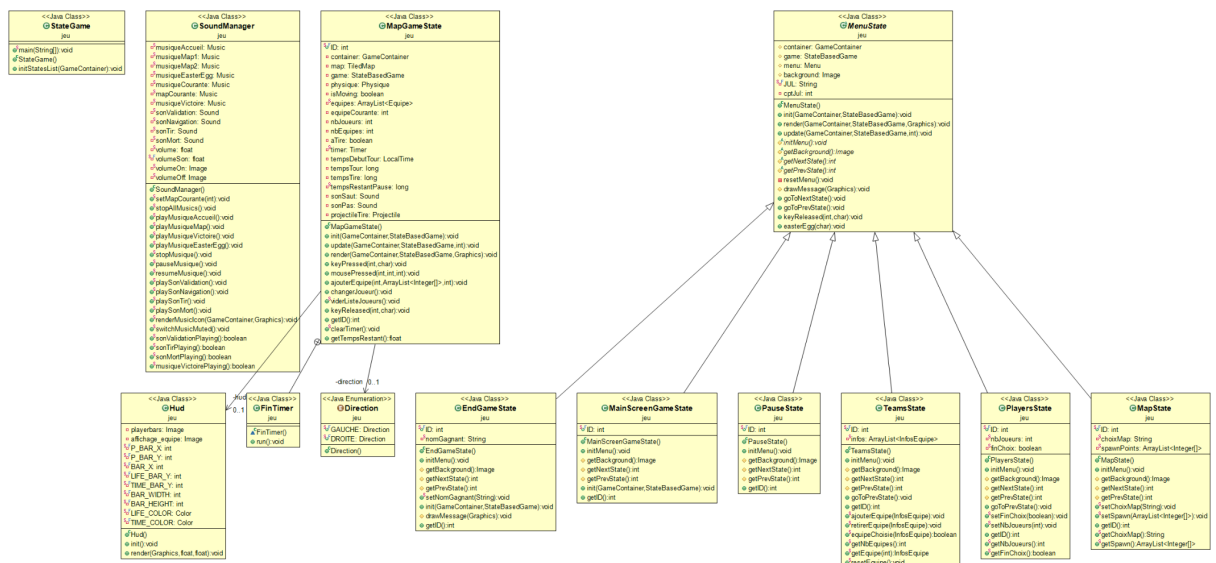
- armes : gestion des armes et projectiles



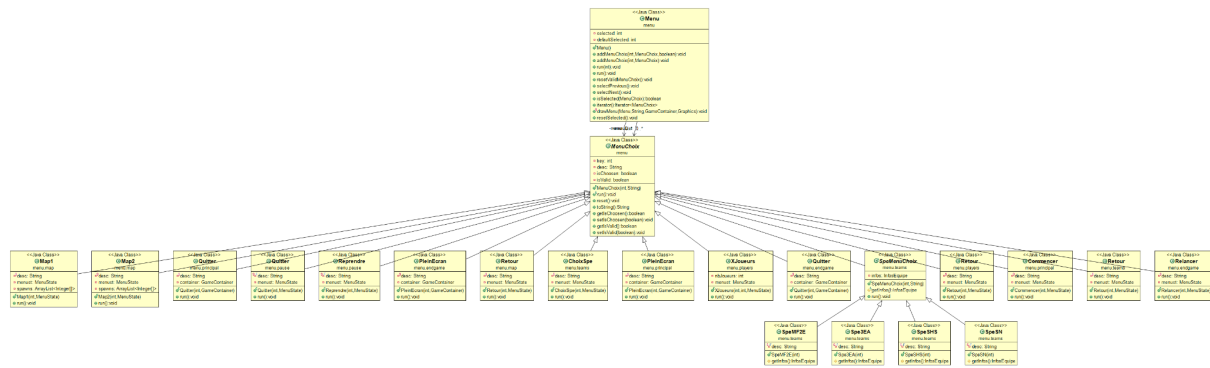
- equipe : gestion des équipes et des personnages qui la compose
 - spécialisation : contient les différents modèles équipes créées.



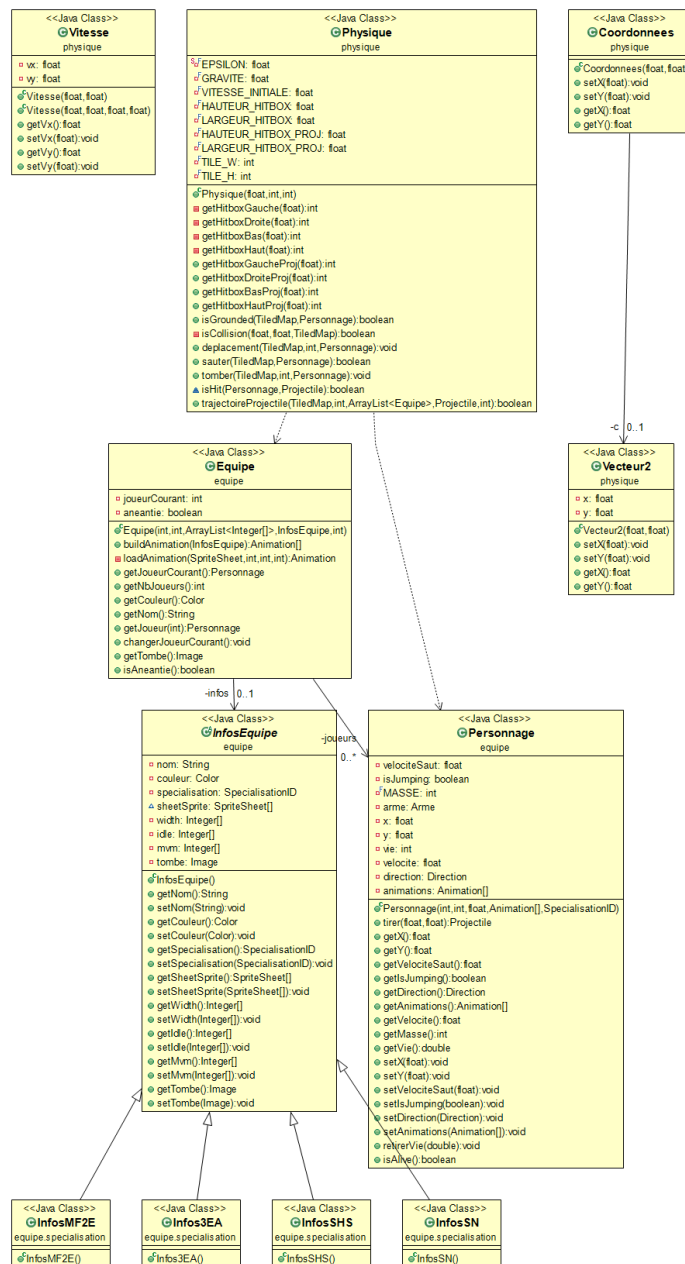
- jeu : gestion de l'instance d'un jeu et de ses différents états. Contient le main



- menu : gestion des instances de menu et des divers choix
 - map : gestion du menu de choix de la carte
 - pause : gestion du menu de pause au cours du jeu
 - players : gestion du menu du choix du nombre de joueurs
 - principal : gestion du menu d'accueil
 - teams : gestion du menu du choix des équipes



- physique : physique du jeu et des diverse corps qui le composent



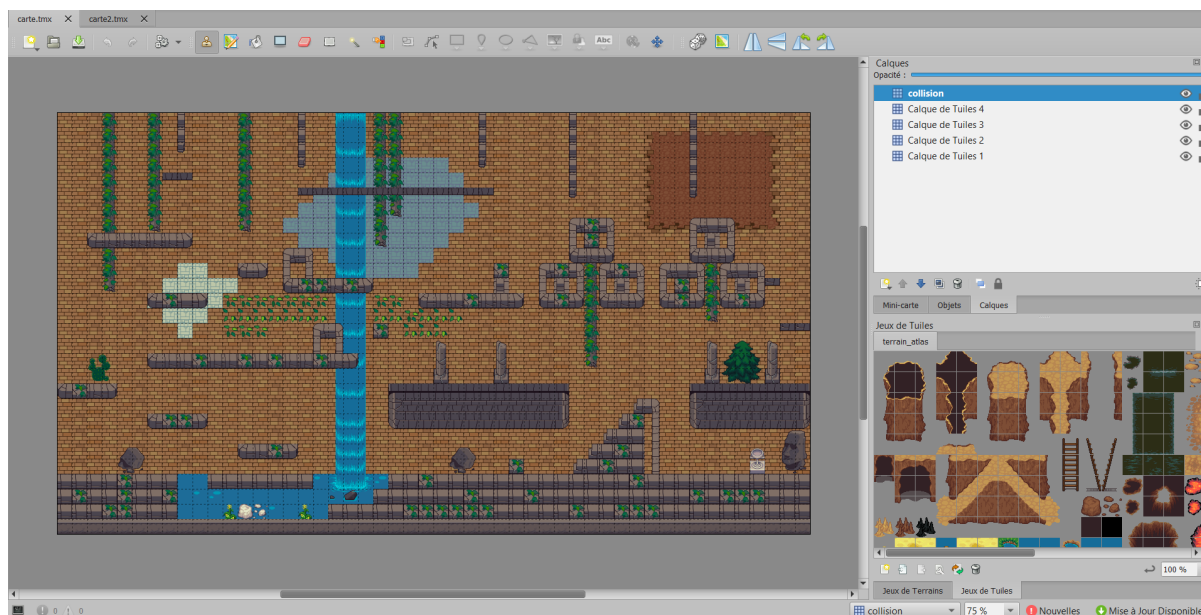
Principaux choix de conception et de réalisation

Afin de créer SN Strikes Back, nous avons décidé d'utiliser une librairie bien utile, la librairie Slick2D.

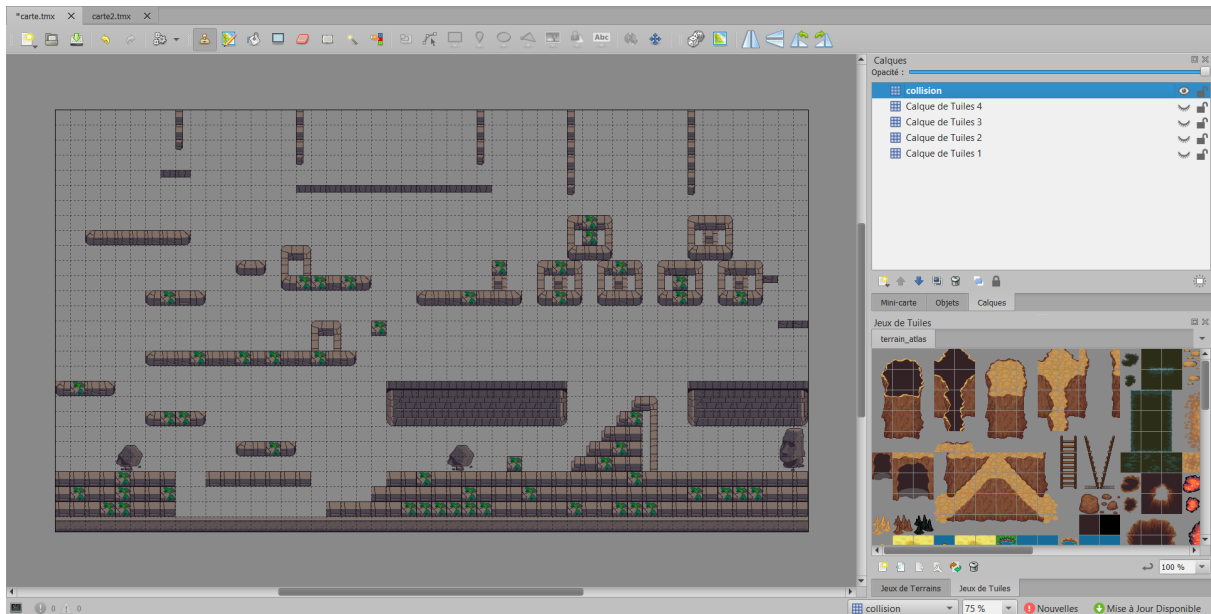
Slick2D est un ensemble d'outils et d'utilitaires faciles à utiliser, open-source, basé sur OpenGL pour faciliter le développement de jeux Java en 2D. Il est utilisable sur MAC, Windows et Linux. Son utilisation a permis de faciliter la gestion des musiques, des bruitages, des graphismes, et des menus.

En effet, les musiques et bruitages sont gérés dans notre application par une classe nommée SoundManager, et puisque Slick2D fait la différence entre un objet *Sound* et un objet *Music*, il est aisé de pouvoir mixer les deux en parallèle grâce au panel de fonction qu'il offre. Toutefois, seul le format .ogg est reconnu.

Les graphismes se gère en deux temps. Dans un premier temps, il s'agit de télécharger un pack de textures libre de droit depuis <https://opengameart.org>, puis de l'utiliser sur l'application Tiled. Une carte est constituée de plusieurs calques, une de collision et les autres qui sont purement cosmétiques.



> *Une carte complète*



> *La carte avec collisions uniquement*

La carte est par la suite chargée dans le code Java de la façon suivante :

```
this.map = new TiledMap("exemple.map");
```

et les collisions sont gérées de la façon suivante :

```
int coucheCollision = map.getLayerIndex("collision");
```

Le calque "collision" est maintenant stocké dans une variable, et on peut de là empêcher les personnages de se confondre avec la texture.

Enfin les menus peuvent se gérer grâce à la fonction *addState* :

```
addState(new Menu());
```

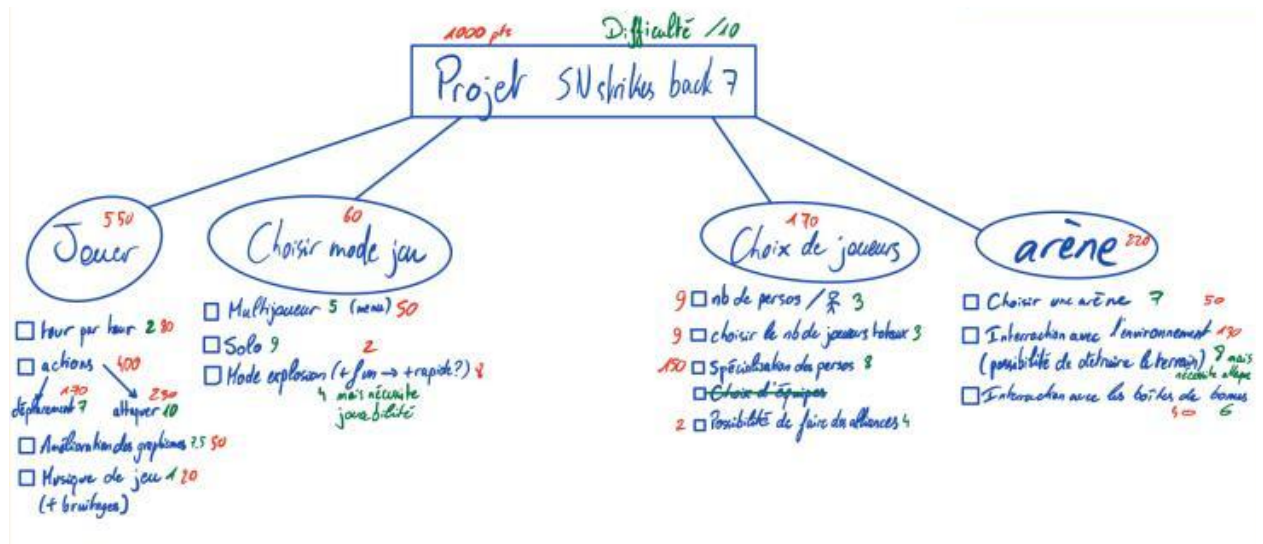
Chaque menu dispose d'un ID de type *public static int*, et il suffit d'utiliser *enterState* :

```
game.enterState(Menu.ID);
```

afin de naviguer d'un menu à l'autre.

Organisation de l'équipe & méthodes agiles

- début de l'itération : détermination des objectifs et répartition
- au fur et à mesure : points de l'avancement/éventuelles questions de choix de conception ou d'implémentation



Nous avons divisé le projet en 4 fonctionnalités majeures: jouer au jeu, choisir le mode de jeu, choisir les joueurs et choisir l'arène, et ces fonctionnalités sont elles-mêmes divisées en sous fonctionnalités. Nous avons réparti 1000 points sur toutes ces sous fonctionnalités pour estimer leur importance, et nous avons donné une note de 1 à 10 estimant la difficulté pour implémenter la sous fonction.

L'objectif d'une gestion de projet agile est de livrer tôt de la valeur métier, nous avons donc choisi de prioriser les fonctionnalités à forte valeur ajoutée par rapport à la difficulté d'implémentation. La première itération du projet s'est donc concentrée sur l'objectif d'avoir un personnage pouvant se déplacer sur une carte. Nous nous sommes répartis les tâches par groupe avec 2 ou 3 pour implémenter les fonctionnalités souhaitées. Comme à la fin de chaque, nous avons discuté de l'avancement du projet et défini de nouveaux objectifs. La seconde itération s'est concentrée sur la création du menu et le début de l'implémentation des armes, qui est la fonctionnalité la plus exigeante du projet, mais aussi une des plus importantes. À la fin de la seconde itération, le tir n'était toujours pas fonctionnel donc cette fonctionnalité a été priorisée pour la 3ème itération. Durant cette itération nous avons également ajouté de nombreuses fonctionnalités plus petites avec une nouvelle interface de jeu, des menus améliorés, ou encore de la musique et des bruits dans le jeu.

À ce stade de développement, toutes les fonctionnalités n'ont pas été développées. Cependant cette méthode de travail nous a permis de développer en priorité les fonctionnalités les plus importantes, et aucune des de celles restantes n'est forcément nécessaire pour pouvoir jouer au jeu. De plus en se concentrant sur certaines fonctionnalités seulement, nous avons pu nous rendre compte à plusieurs reprises que certaines fonctions sont inutiles avant de commencer à travailler dessus, ce qui nous a fait gagner en temps et en productivité.