Jeu d'aventure 2D

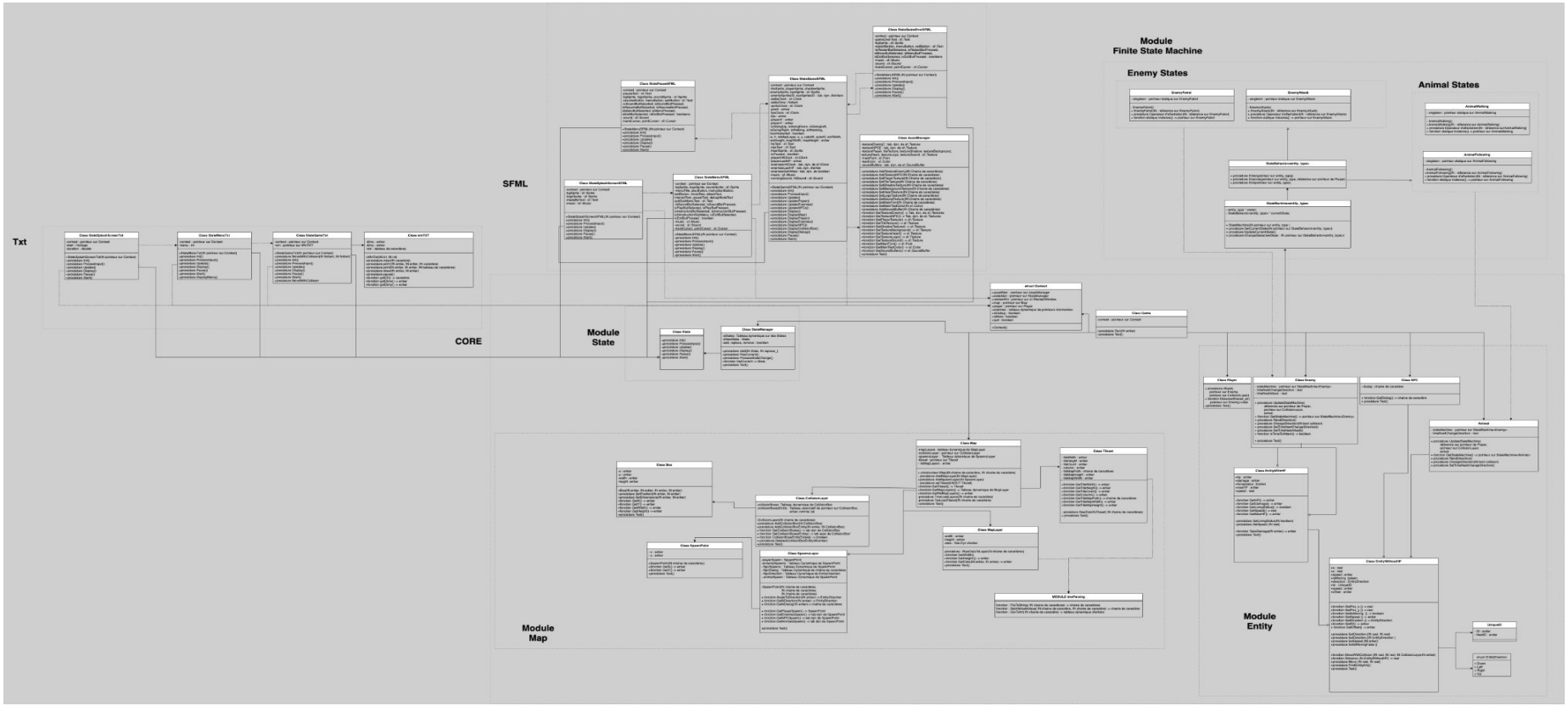
LEGEND OF NAUTIBUS

LIFAP4
PROJET SFL

- SIONI Farès
- BAGNOL Stanislas
- CHOUGAR Lyes

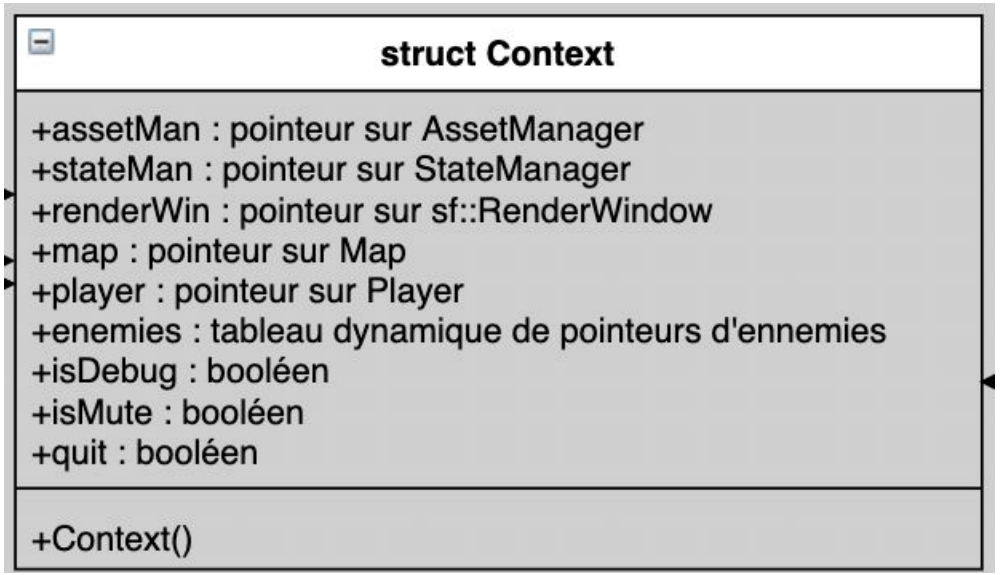
S4 - 2021
Université Claude Bernard Lyon 1

Diagramme des classes



Structure Context Alias : l'état courant

- Elle est le "coeur" des données du jeu.
- Elle est initialisée au lancement et vit aussi longtemps que le programme.
- Elle contient toutes les variables essentielles.
  Ex. : L'état de la fenêtre, les textures (AssetManager), le joueur, etc.



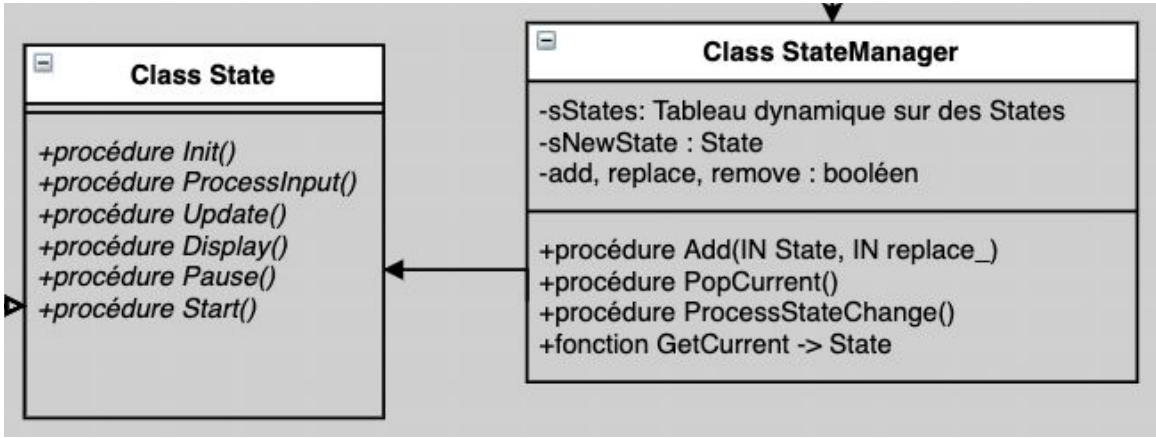
Le State Manager

- **Classe StateManager :**

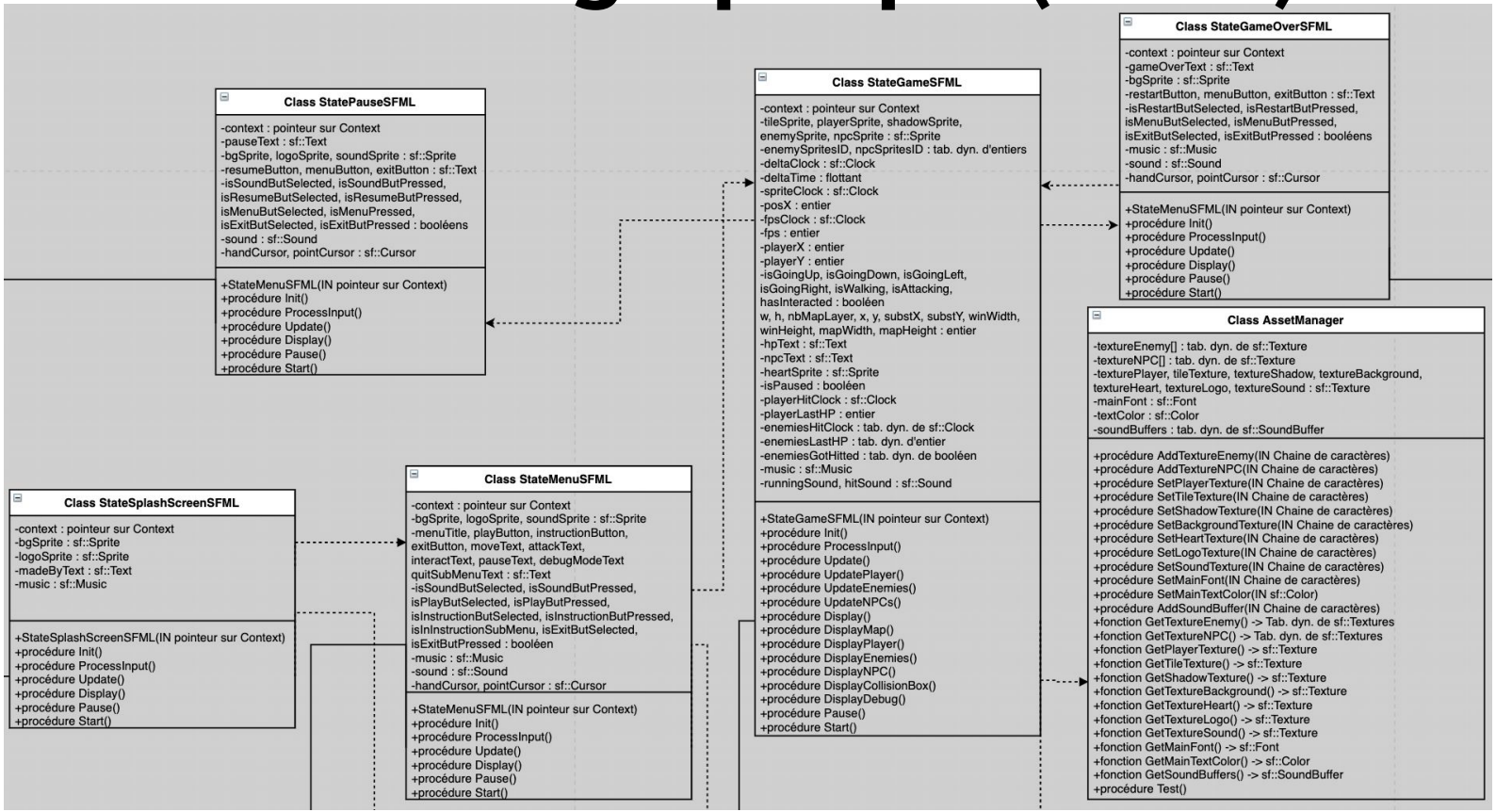
Permet de gérer les passage d'un état à un autre.
Exemple : Passage de SplashScreen au Menu.

- **Classe State :**

Classe représentant un état du programme..
State est une classe virtuelle pure : elle doit être héritée et toutes les fonctions qui en héritent sont implémentées.
Exemple de State : SplashScreen, Menu, Game.



Le mode graphique (SFML)



- **StateSplashScreenSFML :**

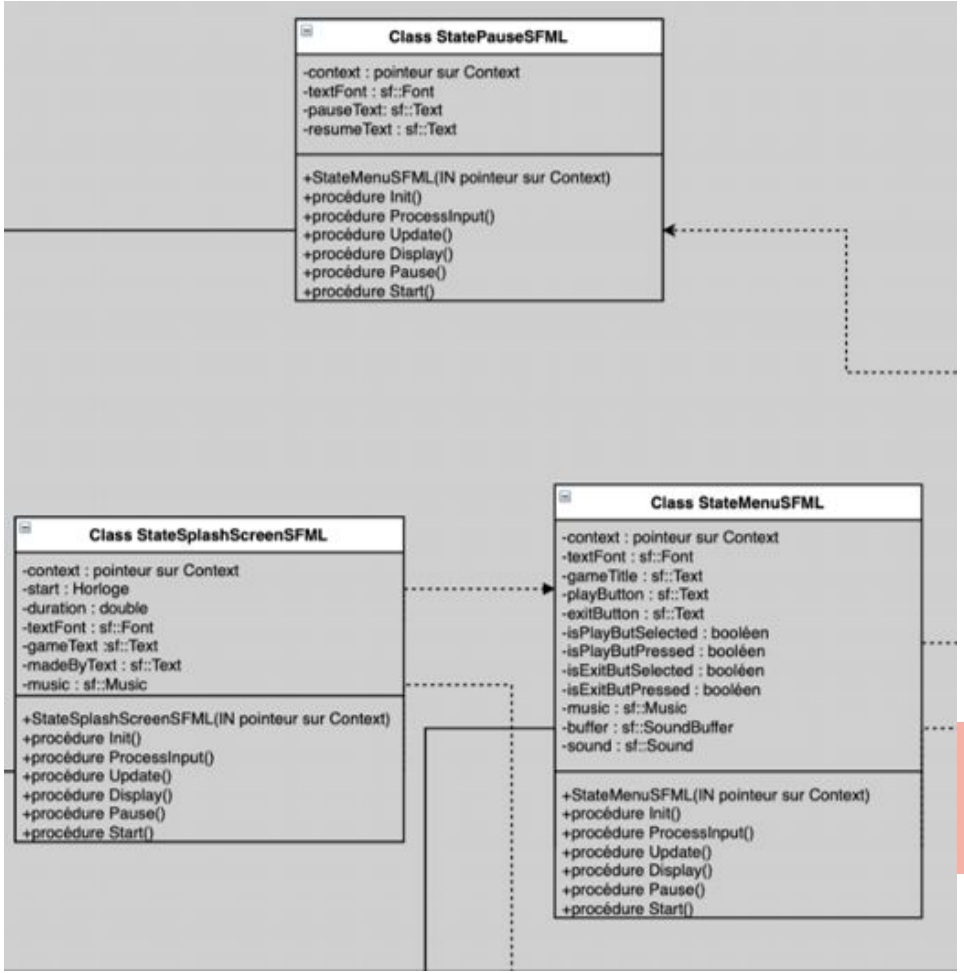
Classe représentant l'état SplashScreen du mode graphique.

- **StateMenuSFML :**

Classe représentant l'état menu du mode graphique.

- **StatePauseSFML :**

Classe représentant l'état pause du mode graphique.



- **StateGameSFML :**

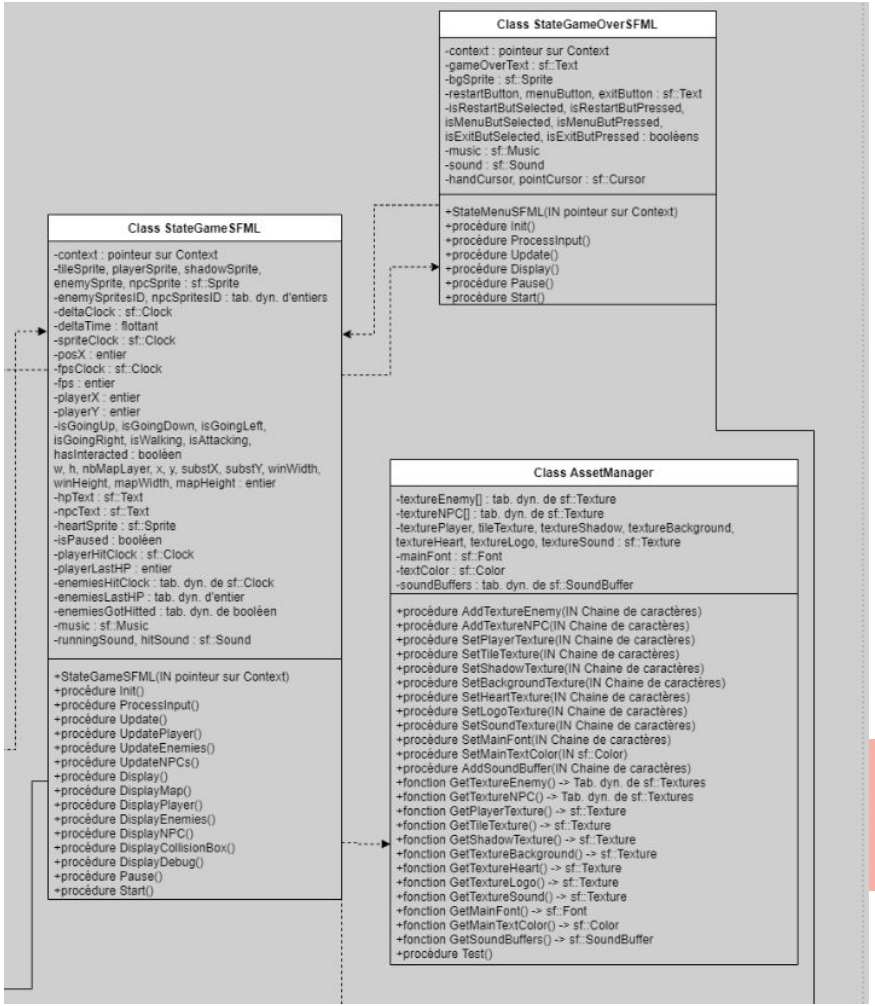
Classe représentant l'état « jeu » du mode graphique.

- **StateGameOverSFML :**

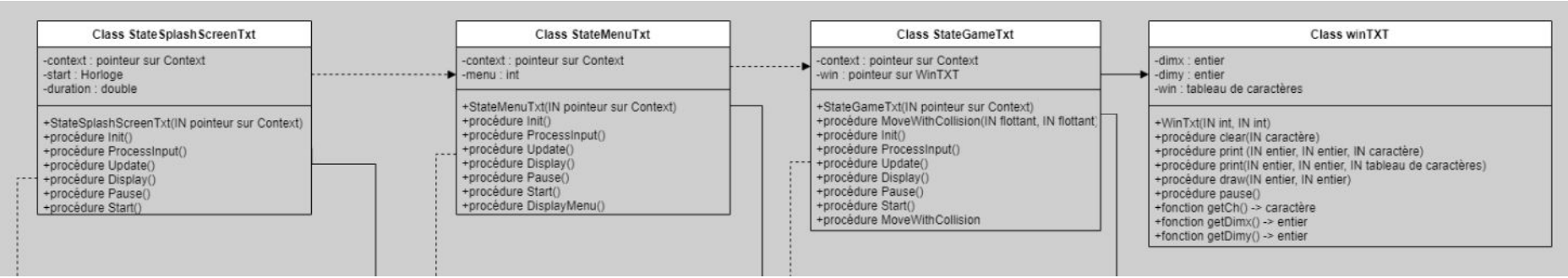
Classe représentant l'état “game over” du mode graphique.

- **AssetManager :**

Classe représentant le gestionnaire des données de l'application (Textures, sons, polices...).



Le mode texte



- **Classe StateSplashScreenTxt :**

Classe représentant l'état SplashScreen du mode texte.

- **Classe StateMenuTxt :**

Classe représentant l'état menu du mode texte.

- **Classe StateGameTxt :**

Classe représentant l'état jeu du mode texte.

- **Classe winTxt :**

Classe représentant la “fenêtre” de jeu dans le terminal. (Récupérée du PacMan version terminal)

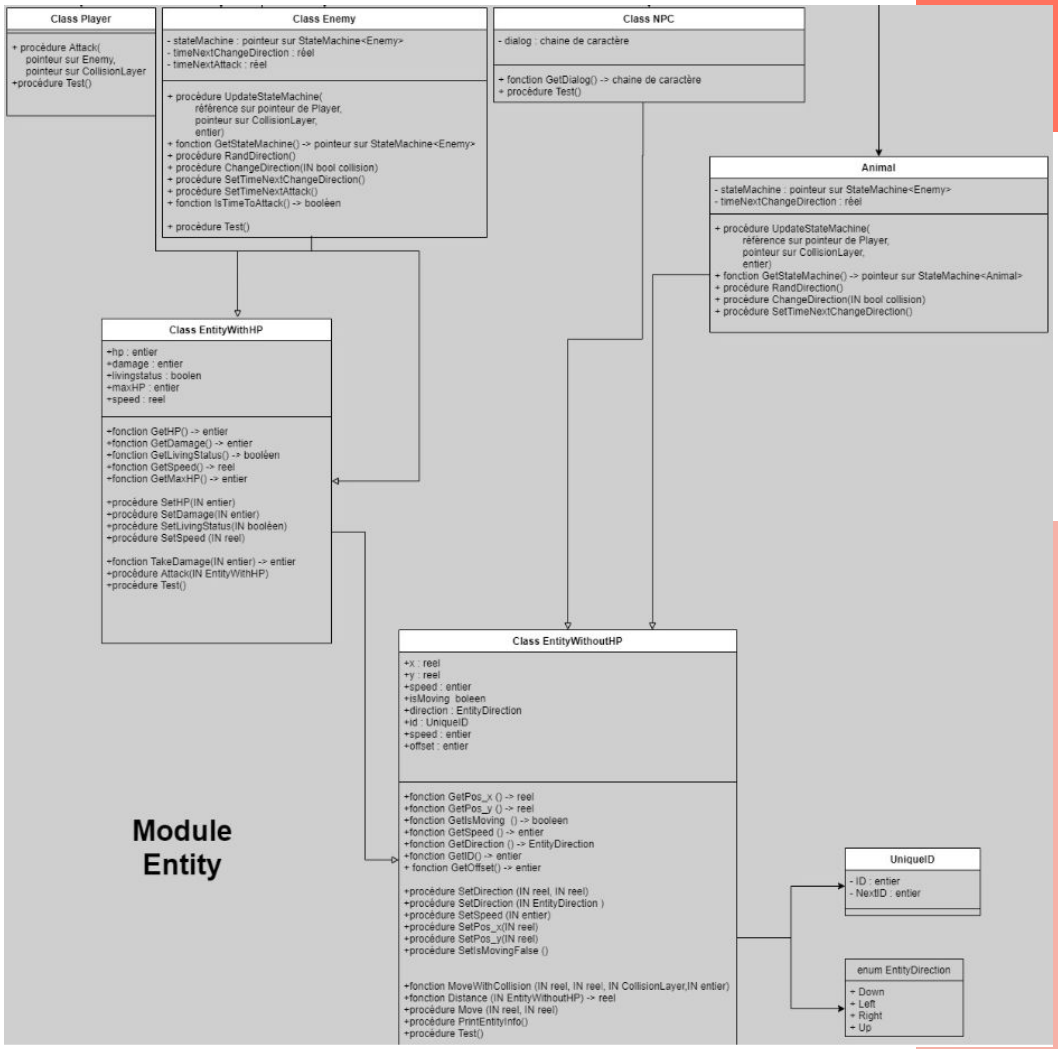
Module Entity

Toutes les classes du module Entity héritent de la classe mère EntityWithoutHP.

Une distinction est faite entre les entités :

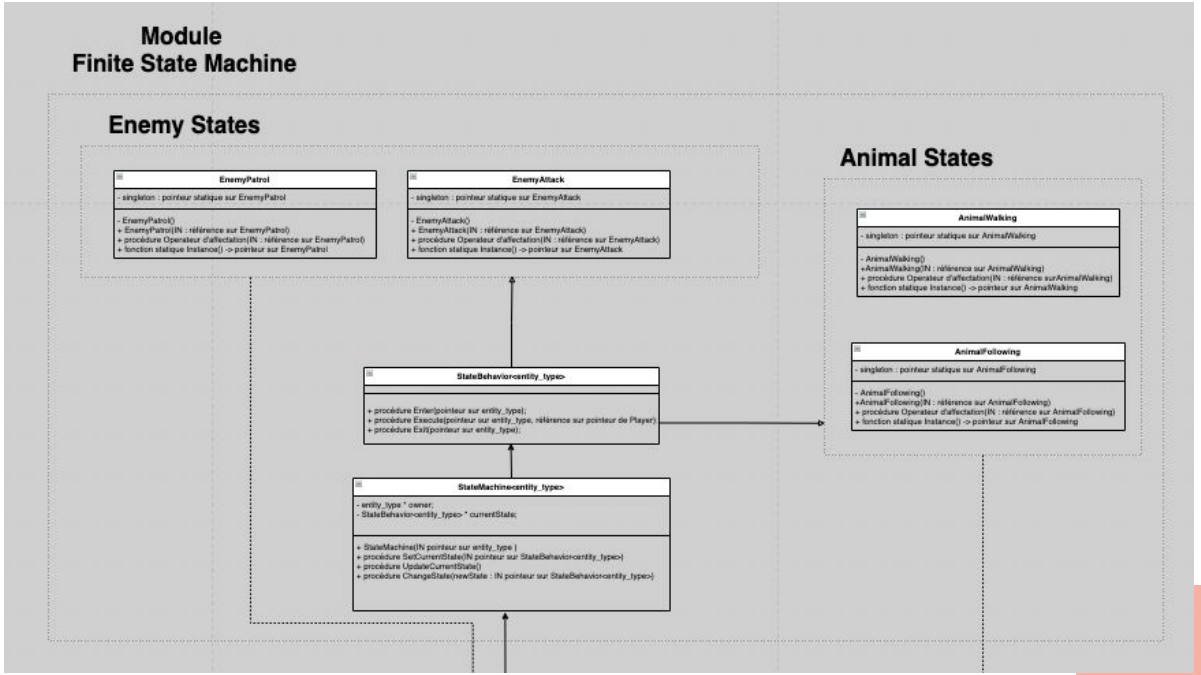
- Les classes Player et Enemy (entités mortelles) sont des classes filles de EntityWithHP.
- Les classes NPC et Animal (entités immortelles) sont des classes héritées de EntityWithoutHP.

La classe EntityWithHP hérite d'EntityWithoutHP.

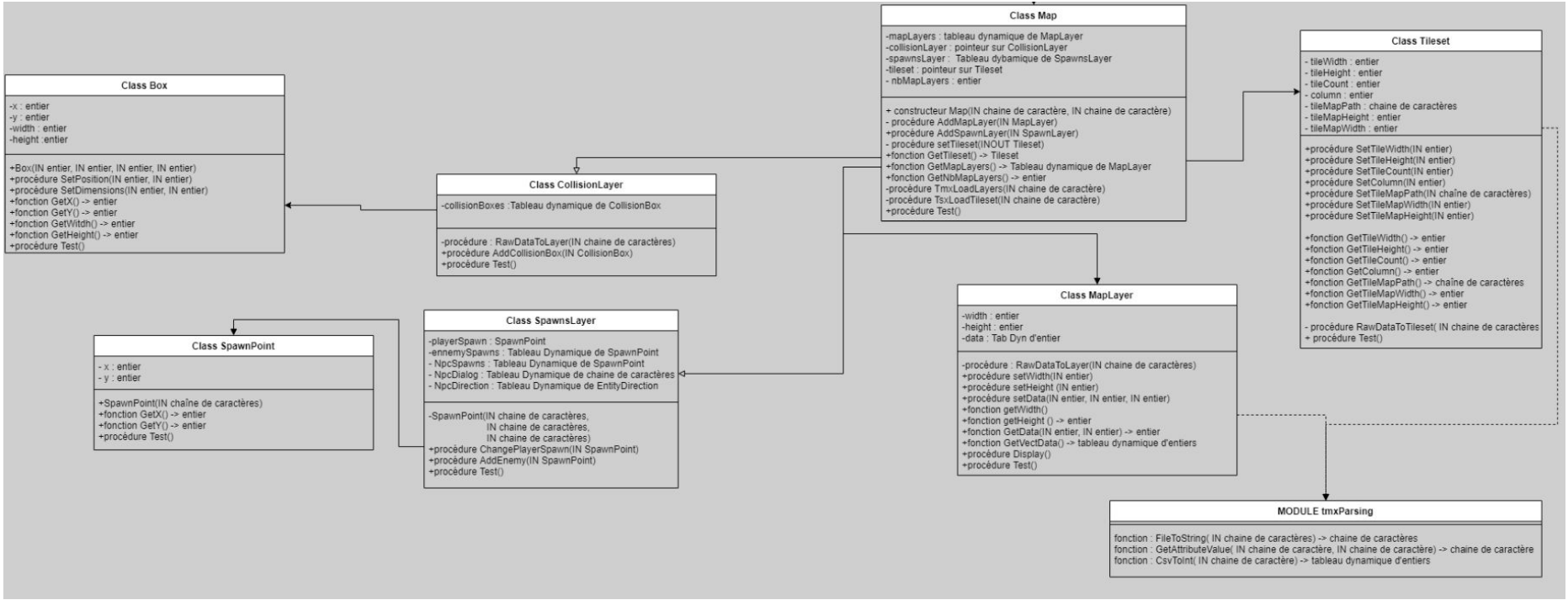


Module Finite State Machine

Un Finite State Machine permet de définir des états prédéfini pour les entités. C'est une méthode plutôt simple pour créer une intelligence artificielle.



tmxParsing et Module Map



tmxParsing

<layer id="2" name="Trees" width="100" height="50">
 <data encoding="csv">
0,15,16,0,0,15,16,0,15,16,0,0,0,0,0,0,0,11,12,0,0,0,0,0,11,12,0,0,0,0,11,12,0,0,0,0,0,0,15,16,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,15,16,0,0,15,16,0,15,16,0,0,0,15,16,0,0,0,0,0,15,16,0,0,
0,0,15,16,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,23,24,15,16,23,24,0,23,24,0,0,0,11,12,0,0,19,20,0,11,12,0,0,19,20,0,11,12,0,19,20,0,0,11,12,0,0,23,24,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,23,24,15,16,23,24,0,23,24,0,15,16,23,24,0,1
5,16,0,0,23,24,0,0,0,0,23,24,0,0,0,0,44,0,0,0,0,0,0,0,44,0,
0,0,0,23,24,0,0,0,0,15,16,0,0,19,20,0,0,0,11,12,19,20,0,0,0,0,0,0,19,20,0,0,0,15,16,19,20,0,15,16,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,23,24,0,0,15,16,0,0,23,24,0,0,0,23,24,0,0,0
,15,16,0,15,16,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,15,16,0,0,15,16,0,23,24,15,16,0,0,0,11,12,19,20,0,0,0,11,12,0,0,0,0,15,16,0,23,24,15,16,0,23,24,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,15,16,0,0,0,0,23,24,0,0,0,0,0,15,16,0,0,15,16,
0,23,24,0,23,24,0,15,16,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

- **Classe Box :**

Représente une boîte dans un monde 2D.
Avec comme propriétés basiques une position, une dimension et un ID.

- **Classe SpawnPoint :**

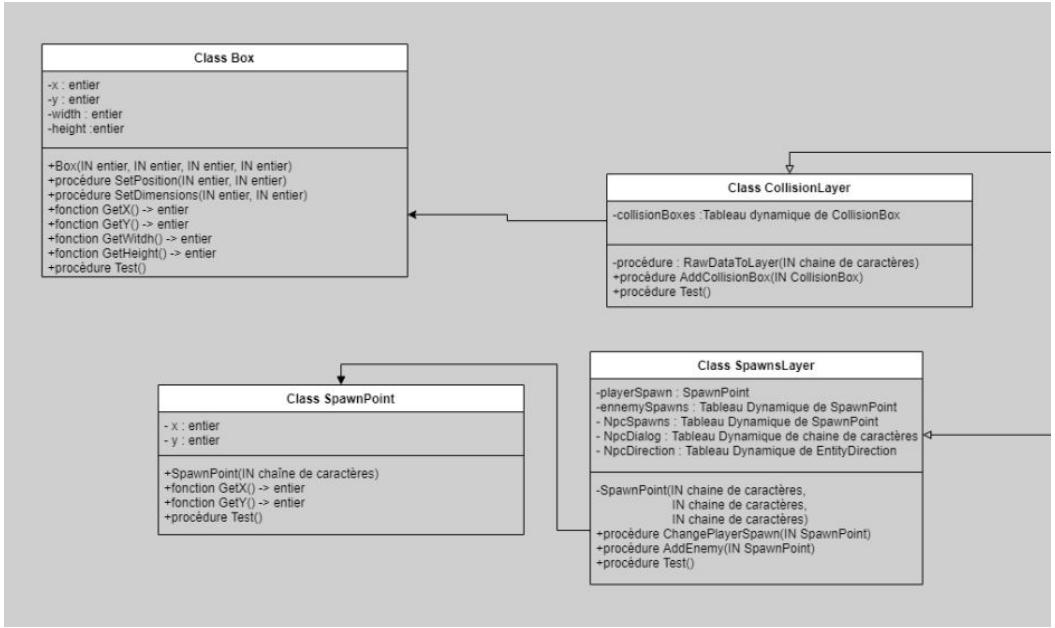
Représente un point d'apparition dans un monde 2D.
Avec des coordonnées sur un plan 2D :
x et y.

- **Classe SpawnLayer :**

Représente tous les SpawnPoint du jeu : des ennemis, du joueur et des NPC.

- **Classe CollisionLayer :**

Représente une couche de CollisionBox d'une partie particulière de la Map
Par exemple : des structures, des ennemis, etc..



- **Classe MapLayer :**

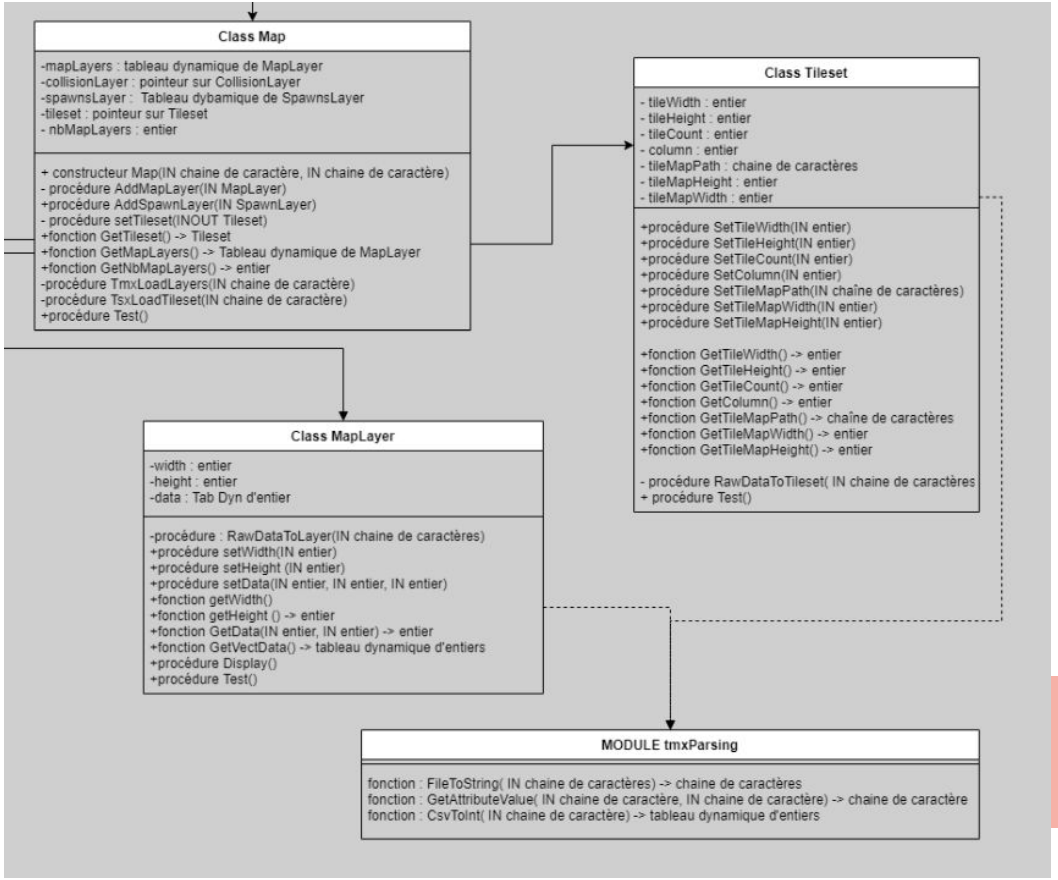
Classe représentant une couche de tuiles

- **Classe Tileset :**

Classe représentant un tileset (ensemble de tuiles stockées sur une image).
Possède toutes les informations sur le tileset :
taille des tuiles, nombre de tuiles par ligne, chemin vers l'image.

- **Classe Map :**

Classe stockant la map.
Possède toutes les informations sur la map comme les différents
layers ou encore le tileset.



Aller plus loin

Système de sauvegarde

Possibilité d'enregistrer/charger la structure Context

Gestion des armes et armures

Le joueur peut porter des armes et armures différentes, visuellement ou dans les caractéristiques

Histoire

Le joueur a un/des objectifs précis, des quêtes à accomplir, etc.

Réglages plus poussés

Possibilité de modifier l'assignation des touches, le volume du son, etc.

Système de compétences

Après avoir gagné de l'expérience, le joueur peut augmenter ses dégâts, sa vitesse, etc.

Bâtiments et PNJ interactifs

Possibilité de rentrer dans des bâtiments, interagir avec les PNJ : nous donner des objets/soins/quêtes

Ce que cela nous a appris

Le travail en équipe	La gestion de projet	Les outils de développement
Communication, affectation/division des tâches	Respect des délais, rigueur, Trello, Gantt, cahier des charges	Gitlab, Doxygen, Valgrind, SFML

L’autonomie

Le problème ne peut être réglé que par nous-même

MERCI
A VOUS!

Avez-vous des questions ?

LEGEND OF
NAUTIBUS