Polybasite

## Rapport d’Avancement



## STAMEGNA Clément

## MICHON Guillaume

## Sommaire

1. Niveau Conception *page.3 - 4*
2. Niveau Programmation *page.5*

# I. Niveau Conception

## Cahier des charges :

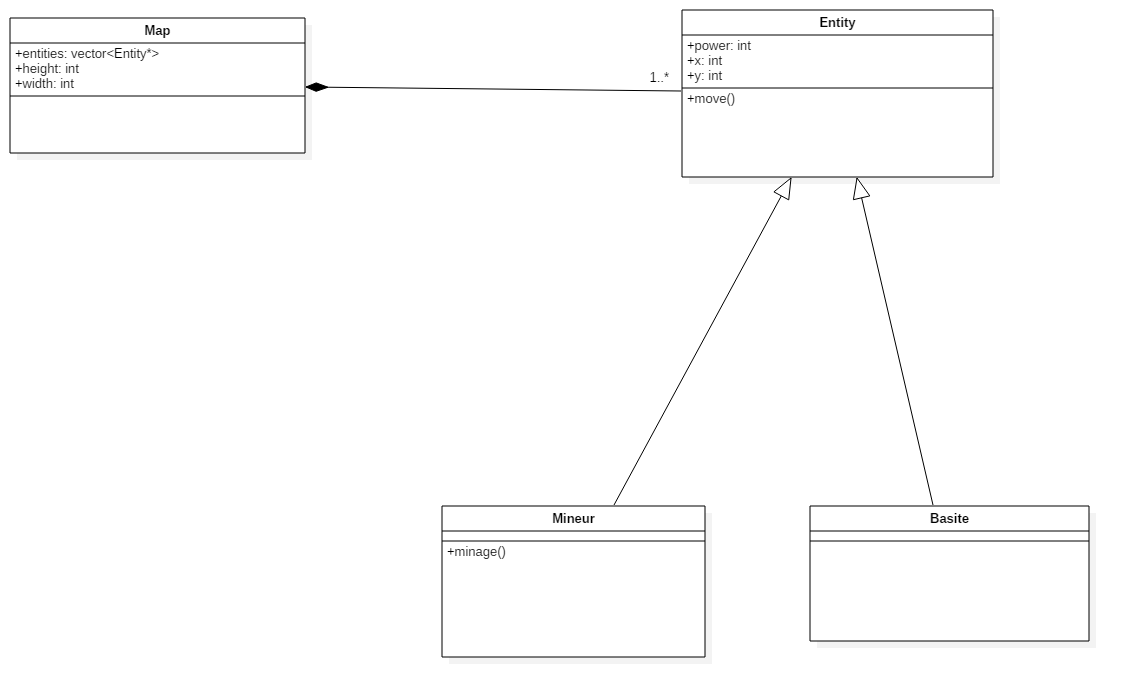
Etat : Terminé (Modifications possible)

Le cahier des charges version 1 a été envoyé le 04 / 12 / 2017, grâces aux remarques constructive avec l’enseignant il a été amélioré puis renvoyer en version 2 le 08/12/2017.

## Spécification :

Etat : En cours de rédaction.

Les spécifications sont en cours de rédactions, un nouveau modèle de domaine pour notre application a été pensé, voici le nouveau diagramme de classe.



Les idées principales retenues sont :

* Suppression complète de la classe Case pour cause de peu d’utilité.
* Les entités connaissent leur position en X et Y donc seulement un Collection d’entité est nécessaire pour les stocker et non une matrice.
* Nous avons renommé la classe « Minerai » en « Basite » pour correspondre au nom du projet, elle garde le même but .

## Cahier de Conception :

Etat : Première Version terminé en attente de validation

Notre cahier de conception spécifie la conception de base de notre projet, il peut évoluer tout au long du projet.

# II. Niveau Programmation

Pour la programmation nous avons mis en place des règles de codage pour une meilleure lisibilité et une intercompréhension du code de chacun.

Voici les quelque points que nous avons commencé à programmer :

## Classes de base :

Nous avons programmé quelques classes de notre diagramme de classe :

* La classe « Map »

Nous avons respecté les attributs du diagramme de classe, nous avons implémenté notre classe par rapport à la SFML de telle façon à pouvoir gérer l’affichage via une méthode « draw »

* La classe « Entity »

Idem que pour la classe « Map » nous avons implémenté notre classe en traitant l’affichage via la méthode « draw »

* La classe « Basite »

Idem que pour la classe « Map » nous avons implémenté notre classe en traitant l’affichage via la méthode « draw »

Concrètement nous avons implémenté des classes efficaces en permettant l’affichage pour obtenir rapidement un visuel pour voir et valider/affiné la conception et quelque spécificité.