

Polybasite

Rapport d'Avancement

STAMEGNA Clément

MICHON Guillaume



Sommaire

I. Niveau Conception

page.3 - 4

II. Niveau Programmation

page.5

I. Niveau Conception

Rappel du github du groupe de projet :

<https://github.com/PolyDevTeam/Polybasite>

Cahier des charges :

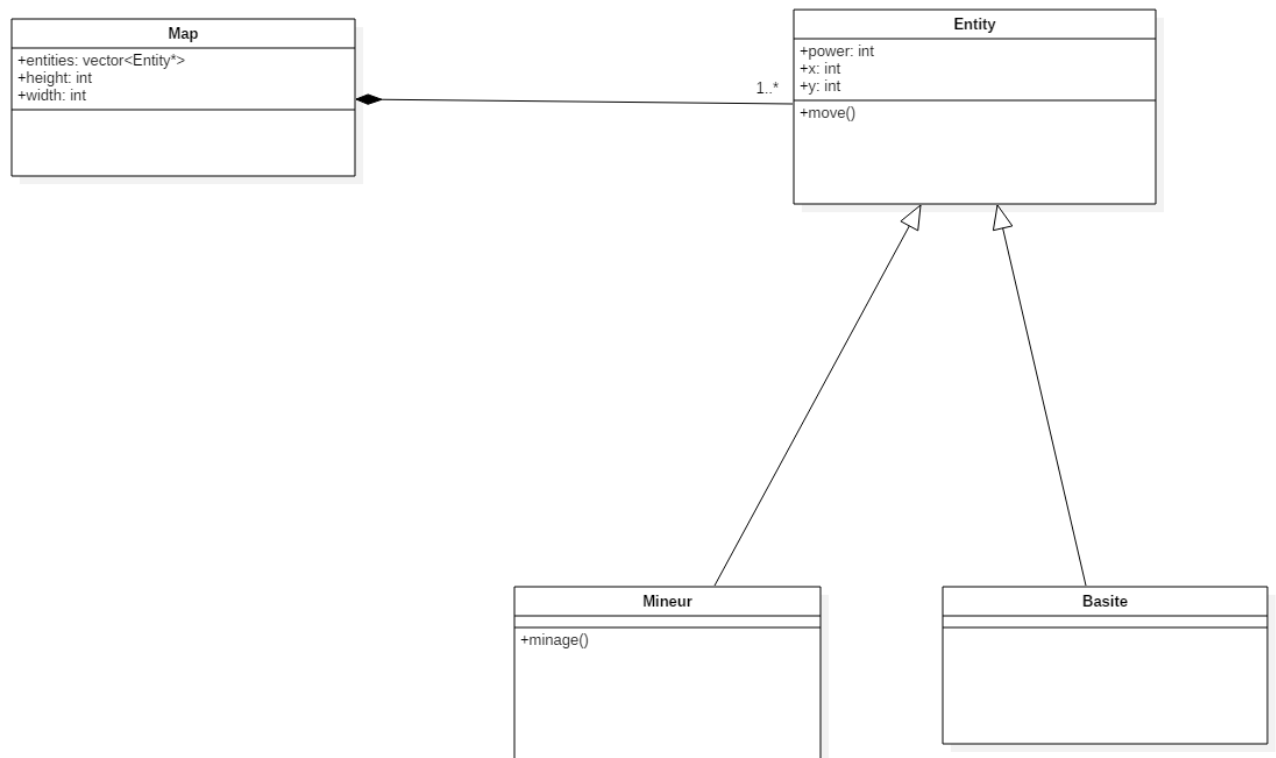
Etat : Terminé (Modifications possibles)

Le cahier des charges version 1 a été envoyé le 04 / 12 / 2017, grâce aux remarques constructives 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 classes.



Les idées principales retenues sont :

- Suppression complète de la classe Case pour cause de peu d'utilité.
- Les entités connaissent leurs positions en X et Y donc seulement une collection d'entité est nécessaire pour les stocker et non une matrice.
- Nous avons renommé la classe « Minerais » en « Basite » pour correspondre au nom du projet, elle garde le même but .

Cahier de Conception :

Etat : Première Version terminée en attente de validation

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

De nouveaux Diagrammes d'états ont été créés pour la compréhension des différentes méthodes programmées.

Un schéma pour représenter notre espace de travail a été implémenté.

Notre conception du projet se base sur deux Packages importants « Map » et « Entité », dont voici un diagramme.



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.

Nous avons aussi créé un Makefile pour faciliter et accélérer le développement et la compilation.

Voici les quelques points que nous avons commencé à programmer :

Classes de base :

Nous avons programmé quelques classes de notre diagramme de classes :

- La classe « Map »

Nous avons respecté les attributs du diagramme de classes, 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.