

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

# Polybasite

## Conception générale

Référence : Polybasite - CCG - 5.0  
Fournisseur  
Date : 11 Février 2018  
Version/Édition : 5.0  
État : Préliminaire

Type de diffusion : Diffusion restreinte  
Autre référence :

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## FICHE DE SUIVI DES AUTORISATIONS ET DIFFUSIONS

### AUTORISATIONS PRESTATAIRE

	Fonction	Nom	Date
Auteur	Binôme	STAMEGNA Clément MICHON Guillaume	08 - 12 - 2017
Approuvé par	Enseignant	JEHANNO Yves	08 - 12 - 2017

### AUTORISATIONS CLIENT

	Fonction	Nom	Date
Approuvé par	Enseignant	MAVROMATIS Sébastien	08 - 12 - 2017

### DIFFUSION INTERNE

Nom	Fonction	Action	Date
STAMEGNA Clément	Concepteur	Rédaction	11 - 02 - 2018
MICHON Guillaume	Concepteur	Rédaction	11 - 02 - 2018
JEHANNO Yves	Enseignant	Vérification	11 - 02 - 2018
MAVROMATIS Sébastien	Enseignant	Vérification	11 - 02 - 2018

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## Historique des révisions

Date	Description et justification de la modification	Auteur	Pages / Chapitre	Edition / Révision
10 - 12 - 2017	Création	STAMEGNA Clément	Toutes	1.0
26 - 12 - 2017	Révision 1 : - Mise à jour de l'UML	STAMEGNA Clément	Toutes	2.0
12 - 01 - 2018	Révision 2 : - Ajout Aspect Dynamique	STAMEGNA Clément	Toutes	3.0
31 - 01 - 2018	Révision 3 : - Ajout Correspondance Classes - UseCases	STAMEGNA Clément	Toutes	4.0
11 - 02 - 2018	Révision 4 : - Ajout des dernières classes - Mise à jour de l'UML	STAMGNA Clément	Toutes	5.0

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## Table des matières

<b>FICHE DE SUIVI DES AUTORISATIONS ET DIFFUSIONS .....</b>	<b>2</b>
<b>Historique des révisions .....</b>	<b>3</b>
<b>Table des matières .....</b>	<b>4</b>
<b>1 Introduction.....</b>	<b>7</b>
1.1 Objet du document.....	7
1.2 Responsabilités .....	8
1.3 Evolution.....	8
1.4 Outils utilisés .....	8
<b>2 Terminologie .....</b>	<b>9</b>
2.1 Abréviations.....	9
2.2 Définitions des termes employés .....	9
<b>3 Description et analyse de l'environnement.....</b>	<b>10</b>
3.1 Description des ressources matérielles .....	10
3.2 Description des ressources logicielles .....	10
3.3 Organisation de l'espace de travail .....	12
<b>4 Structure statique .....</b>	<b>13</b>
4.1 Décomposition générale .....	13
4.2 Analyse statique.....	23
4.2.1 Classe « Game » .....	23
4.2.1.1 Définition .....	23
4.2.1.2 Attributs .....	23
4.2.1.3 Opérations .....	24
4.2.1.4 Fonctionnalité .....	26
4.2.2 Classe « Map » .....	27
4.2.2.1 Définition .....	27
4.2.2.2 Attributs .....	27
4.2.2.3 Opérations.....	27
4.2.2.4 Fonctionnalité .....	29
4.2.3 Classe « Entity ».....	30
4.2.3.1 Définition .....	30
4.2.3.2 Attributs .....	30
4.2.3.3 Opérations.....	30
4.2.3.4 Fonctionnalité .....	31
4.2.4 Classe « Miner » .....	32
4.2.4.1 Définition .....	32
4.2.4.2 Attributs .....	32
4.2.4.3 Opération .....	32
4.2.5 Classe « Basite ».....	35
4.2.5.1 Définition .....	35
4.2.5.2 Attributs .....	35
4.2.5.3 Opération .....	35
4.2.6 Classe « BlackHole ».....	37
4.2.6.1 Définition .....	37
4.2.6.2 Attributs .....	37
4.2.6.3 Opération .....	37
4.2.7 Classe « Bot ».....	39
4.2.7.1 Définition .....	39

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

4.2.7.2	Attributs .....	39
4.2.7.3	Opération .....	39
4.2.8	Classe « Socket » .....	41
4.2.8.1	Définition .....	41
4.2.8.2	Attributs .....	41
4.2.8.3	Opération .....	41
4.2.9	Classe « Serializable » .....	42
4.2.9.1	Définition .....	42
4.2.9.2	Attributs .....	42
4.2.9.3	Opération .....	42
4.2.10	Classe « Move » .....	43
4.2.10.1	Définition .....	43
4.2.10.2	Attributs .....	43
4.2.10.3	Opération .....	43
4.2.11	Classe « Color » .....	45
4.2.11.1	Définition .....	45
4.2.11.2	Attributs .....	45
4.2.11.3	Opération .....	45
4.2.12	Classe « Util » .....	46
4.2.12.1	Définition .....	46
4.2.12.2	Attributs .....	46
4.2.12.3	Opération .....	46
4.2.13	Classe « Score » .....	47
4.2.13.1	Définition .....	47
4.2.13.2	Attributs .....	47
4.2.13.3	Opération .....	47
4.2.14	Classe « ProgressBar » .....	48
4.2.14.1	Définition .....	48
4.2.14.2	Attributs .....	48
4.2.14.3	Opération .....	48
4.2.15	Classe « RichText » .....	50
4.2.15.1	Définition .....	50
4.2.15.2	Attributs .....	50
4.2.15.3	Opération .....	50
4.2.16	Classe « Line » .....	51
4.2.16.1	Définition .....	51
4.2.16.2	Attributs .....	51
4.2.16.3	Opération .....	51
4.2.17	Classe « Log » .....	52
4.2.17.1	Définition .....	52
4.2.17.2	Attributs .....	52
4.2.17.3	Opération .....	52
4.2.18	Classe « Loggable » .....	53
4.2.18.1	Définition .....	53
4.2.18.2	Opération .....	53
4.2.19	Classe « Error » .....	54
4.2.19.1	Définition .....	54
4.2.19.2	Opération .....	54
4.2.20	Classe « FatalError » .....	55
4.2.20.1	Définition .....	55
4.2.20.2	Opération .....	55
4.3	Correspondance Classes Fonctionnalité .....	56
5	Aspects dynamiques.....	58

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

5.1	Echanges entre éléments.....	58
5.2	Comportement du logiciel en erreur.....	59
5.3	Modes de fonctionnement.....	59
<b>6</b>	<b>Allocation des ressources .....</b>	<b>60</b>
6.1	Gestion de la mémoire .....	60

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

# 1 Introduction

## 1.1 Objet du document

La conception générale d'un logiciel est une démarche rationnelle, qui consiste à réduire la complexité initiale en la décomposant en constituants de niveau inférieur.

La conception permet de préciser *comment* vont être réalisées les spécifications.

La conception préliminaire permet d'élaborer une solution technique répondant aux spécifications. Elle précise l'architecture de cette solution (interfaces internes, constituants principaux, ...), ainsi que son comportement dynamique (logique d'enchaînements, diagrammes d'état, parallélisme, synchronisation, ...).

L'architecture du matériel cible est présentée dans la conception générale si cela n'a pas été fait dans un autre document.

Il faut présenter succinctement la structure du document :

- description et analyse des ressources matérielles et logicielles,
- diagramme des catégories ou des classes principales,
- structure statique (interfaces externes, interfaces internes, ...),
- structure dynamique (synchronisation entre tâches, séquençement des traitements, passages entre les différents états ou mode de traitement, ...),
- implémentation sur le matériel cible (allocation des ressources CPU mémoire bande passante, adresse des périphériques, couche basse de protocoles spécifiques, ...).

La conception générale peut être réalisée avec plus ou moins de détail en fonction de la taille du projet. Dans le cadre de notre projet, la conception générale est suffisante.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 1.2 Responsabilités

La rédaction de la conception générale est de la responsabilité du binôme de projet. Il juge de son état complet et décide de sa présentation en revue de conception.

La conception générale est souvent livrable, elle permet de décrire de façon complète le travail à réaliser.

## 1.3 Evolution

La conception générale fait partie de la référence de réalisation du système; toute modification de cette référence intervenant après le prononcé de revue de conception doit être traitée comme une demande d'évolution.

Quand dans la phase de développement des détails d'implémentation apparaissent, l'évolution du document de conception générale évolue en fonction des types de projets. La conception générale doit alors être mise à jour.

## 1.4 Outils utilisés

Les documents de base sont rédigés avec la suite bureautique Office sous MS-Windows ou la suite bureautique Libre Office sous une distribution Linux. D'autres outils peuvent être utilisés dans le cadre du projet mais les règles énoncées ci-après restent valable sur toute la durée du projet. Les schémas présentés respectent les conventions UML.



Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 2 Terminologie

### 2.1 Abréviations

UML                      Unified Modeling Language

IHM                      Interface Homme Machine

### 2.2 Définitions des termes employés

Attribut	un attribut est une information caractéristique mémorisée par un objet.
cas d'utilisation	cas d'utilisation du système, par extension il représente également la technique de modélisation mise en euvre dans UML (use case).
catégorie	une catégorie consiste en un regroupement logique de classes à forte cohérence interne et faible couplage externe, associée au concept UML de package. Ce concept permet une présentation plus synthétique du diagramme des classes d'un système réel.
Classe	une classe définit un ensemble d'objets similaires potentiels. Elle fournit le modèle de la structure et les possibilités de chaque objet.
Objet	un objet est une instance d'une classe, c'est une entité informatique unique possédant ses propres attributs et opérations
opération ou méthode	une opération est un traitement spécifique qu'un objet est en charge de fournir.
Tâche	une tâche représente un élément manipulé par le système et ordonnançable de manière individuelle. cela peut représenter un process d'un système Unix, une tâche d'un moniteur temps-réel, un thread d'une application.
Threads	codes exécutés de manière concurrente par le système d'exploitation mais partageant le même espace mémoire.
fichiers de log	Fichier de report d'erreur qui permet d'avoir une meilleure connaissance des conditions qui ont abouti à cette erreur.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

### 3 Description et analyse de l'environnement

#### 3.1 Description des ressources matérielles

Le jeu Polybasite nécessite des ressources matérielles très simples. Notamment un Ordinateur ayant un écran pour l'affichage ainsi que d'un clavier pour écrire des données.

#### 3.2 Description des ressources logicielles

Le jeu nécessite des bibliothèques spécifiques pour s'exécuter :

- SFML

Bibliothèque pour l'interface graphique et le fenêtrage.

SFML est le cœur de notre projet, elle permettra de gérer les affichages ainsi que tous les autres outils et bibliothèques cités ci-dessous.

- Freetype

Bibliothèque pour les polices de caractères

- Jpeg
- X11

Système de fenêtrage

- Xrandr

Outil de gestion des paramètres d'affichage

- Xcb

Bibliothèque d'implémentation des protocoles de fenêtrage

- X11-xcb
- Xcb-image

- Opengl

Bibliothèque de calcul d'image 2D et 3D

- Flac

Codec de compression audio

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- Ogg

Type de fichiers compressés via la librairie Vorbis

- Vorbis

Codec de compression audio

- Vorbisenc
- Vorbisfile

- Openal

Librairie d'implémentation audio

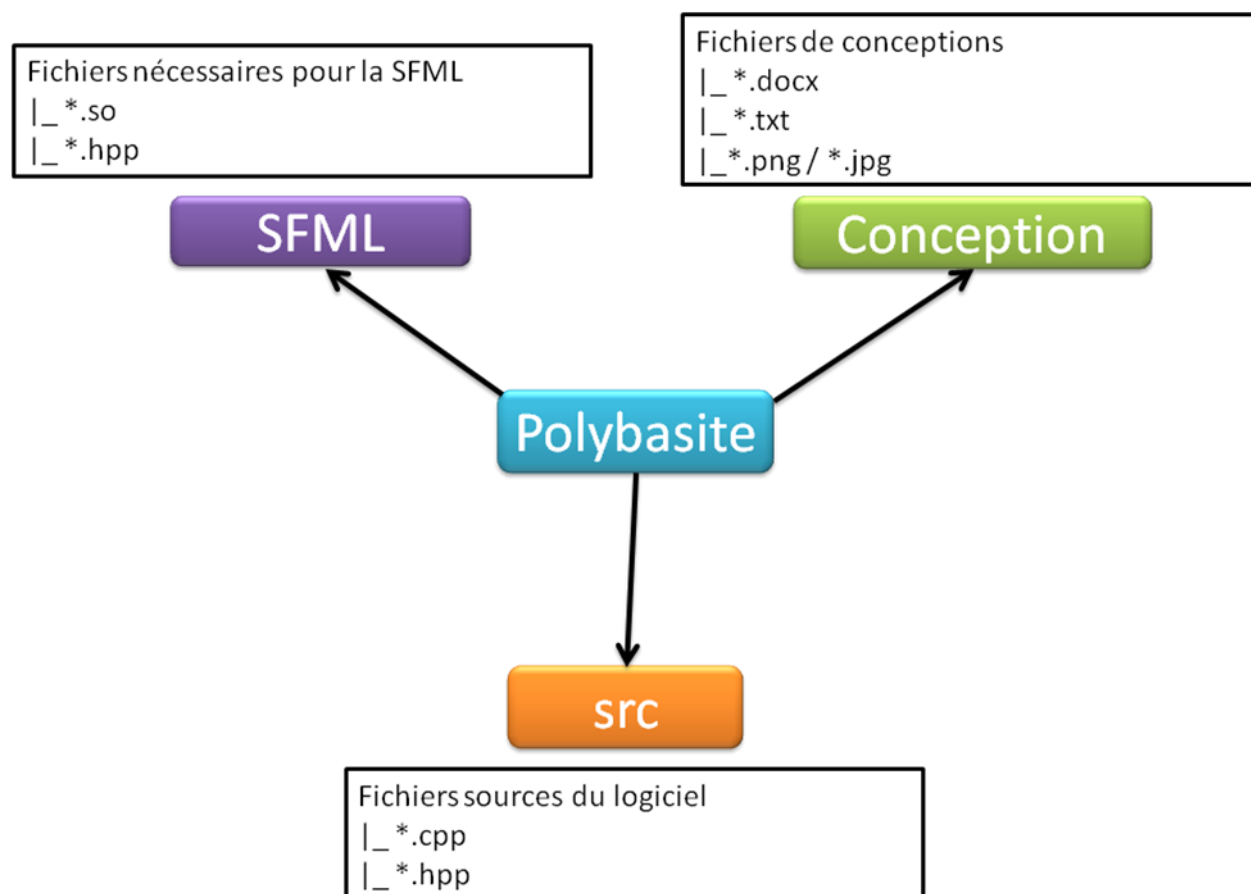
- Pthread

Librairie de gestion de threads POSIX

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

### 3.3 Organisation de l'espace de travail

Tous les fichiers du projet sont gérés par un dépôt Github. Le système de fichiers est organisé de cette façon :

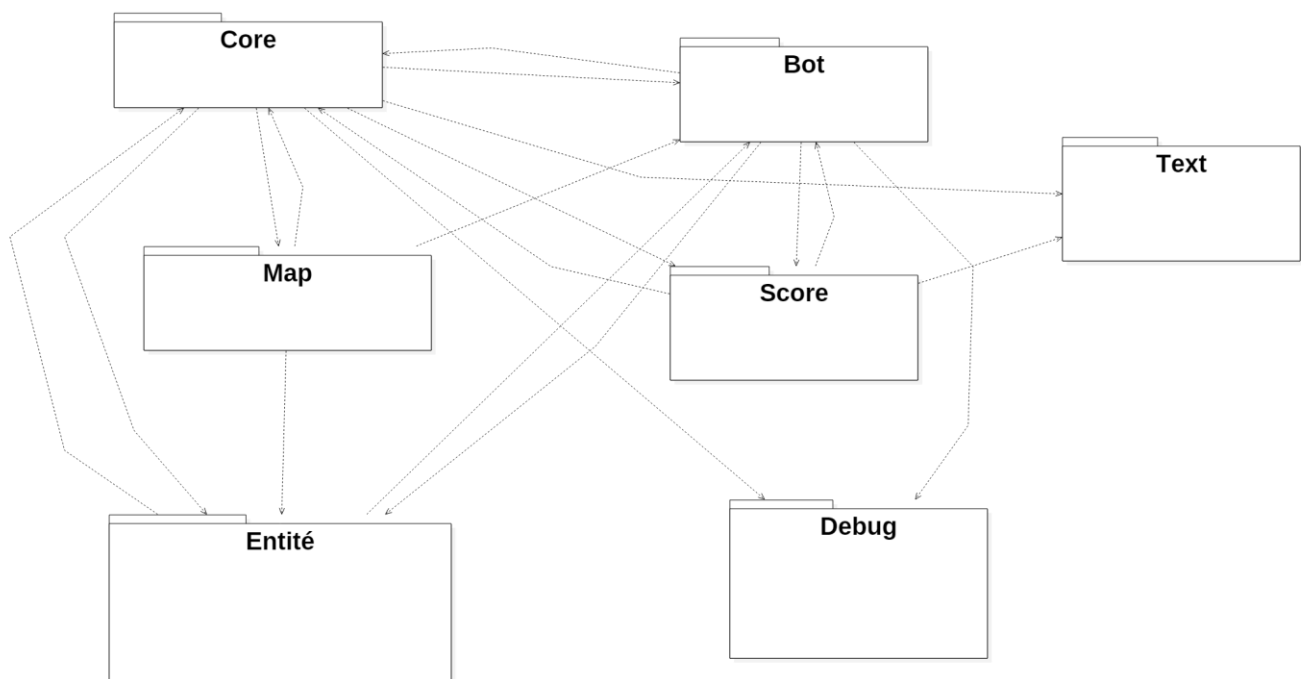




Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

Nous allons nous concentrer sur le Découpage en différents Package ainsi que leurs Contenus.

- **Diagrammes de Package :**



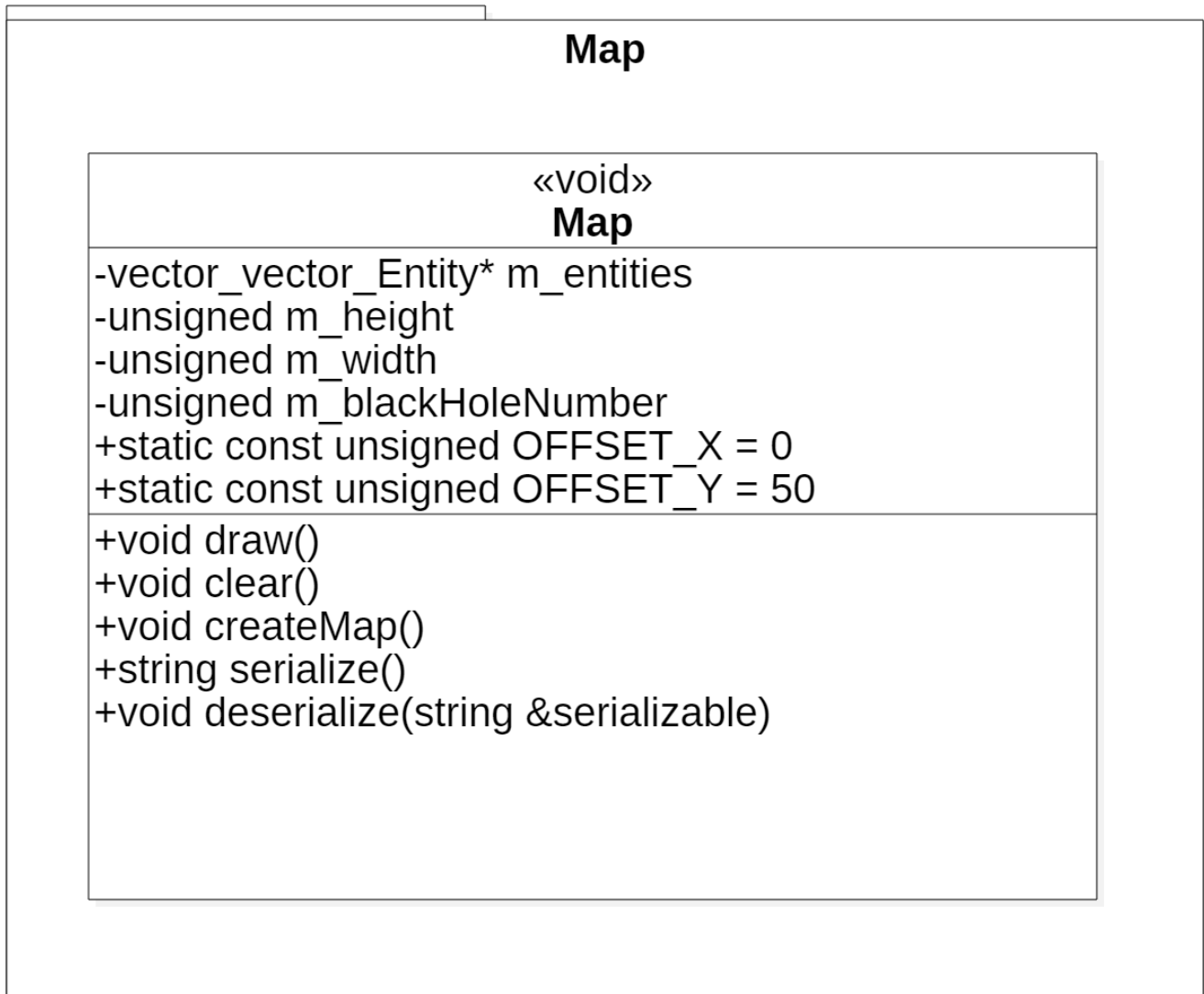
Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- **Package Core :**

<div>Core</div> <div>Game</div> <pre>{enum GameState="STATE_UNINITIALISED, STATE_PLAY, STATE_REPLAY, STATE_PAUSE, STATE_FINISH, STATE_QUIT,"} +static const unsigned SCREEN_WIDTH = 1024 +static const unsigned SCREEN_HEIGHT = 650 +static const unsigned MAX_TURN = 1500 +static const unsigned SPEED_STEP = 50 +static char** save_argv +static int save_argc +static GameState m_state +static sf_RenderWindow m_main_window +static Map m_map +static vector_Score* m_scores +static vector_Bot* m_bots +static unsigned m_nb_turn +static unsigned m_turn_speed +static void start(int argc, char* argv[]) +static void restart() +static void quit() -static void turn() -static void loop() -static void displayBotNames() -static bool hasWinner() -static void displayWinner() -static void displayScore() -static void displayTurn()</pre>
---

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

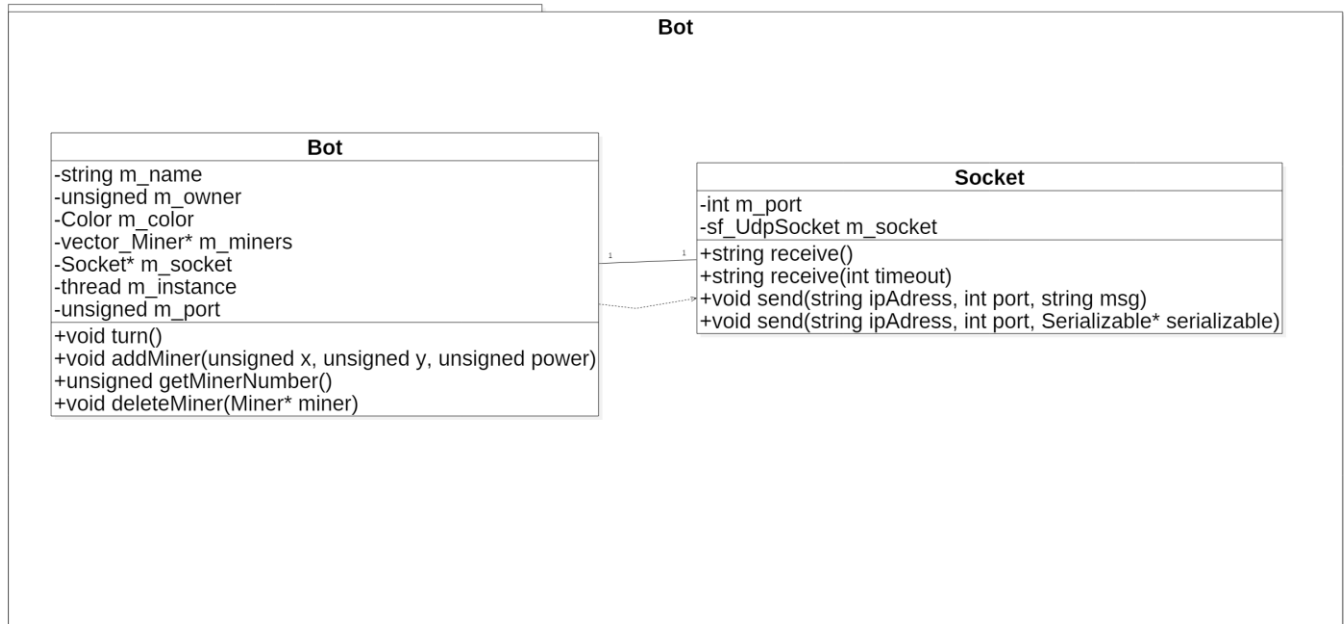
- **Package Map :**





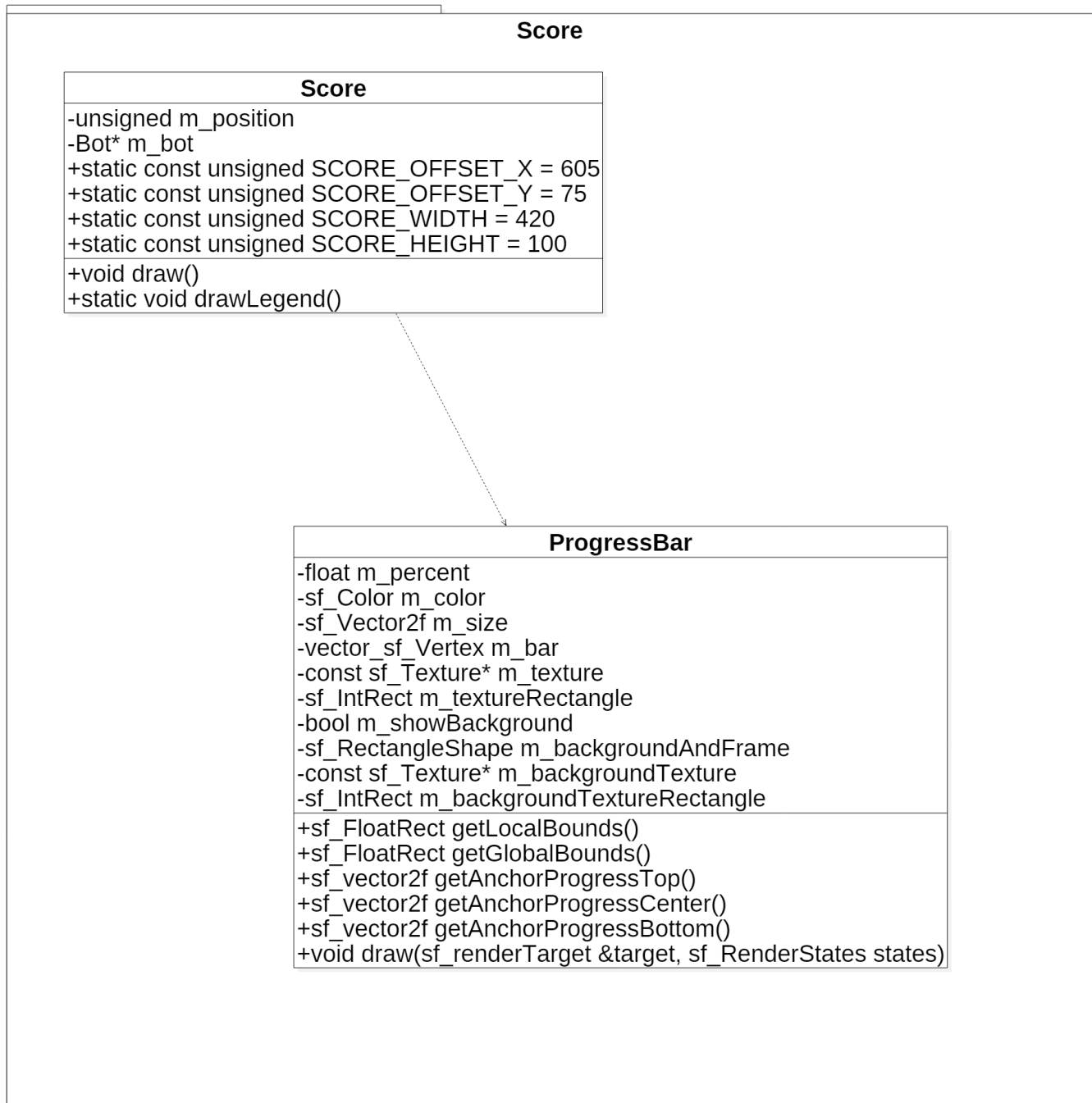
Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	<b>Projet</b> Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	--	---

## • Package Bot :



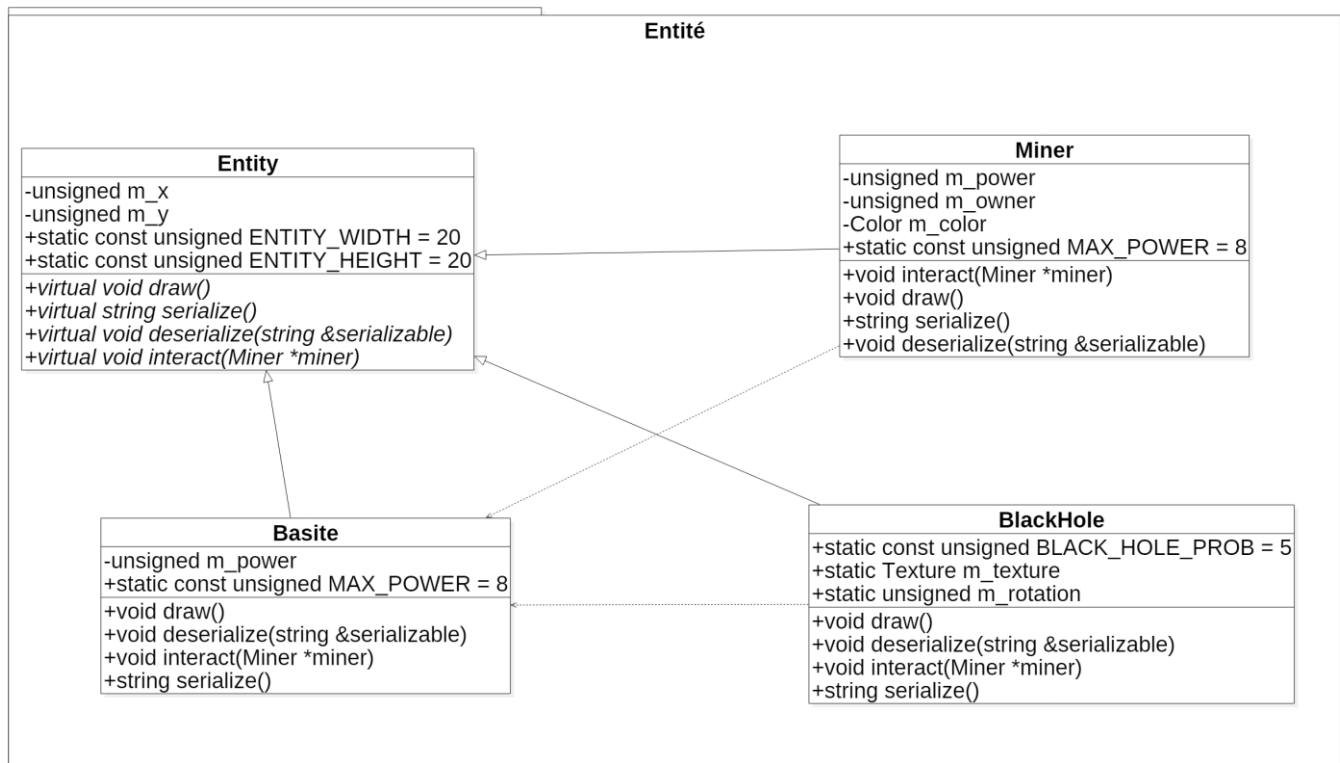
Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- **Package Score :**



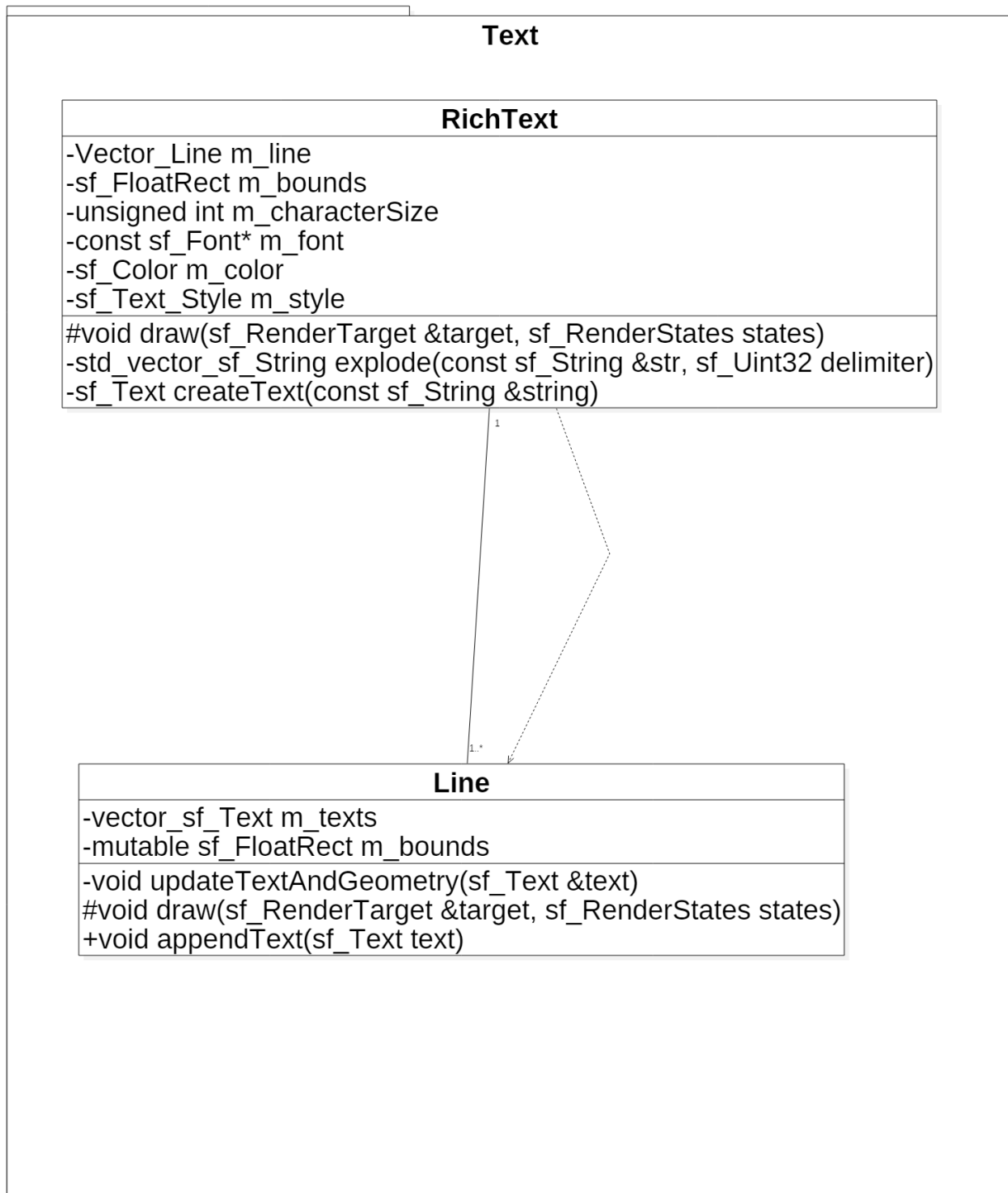
Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## • Package Entité :



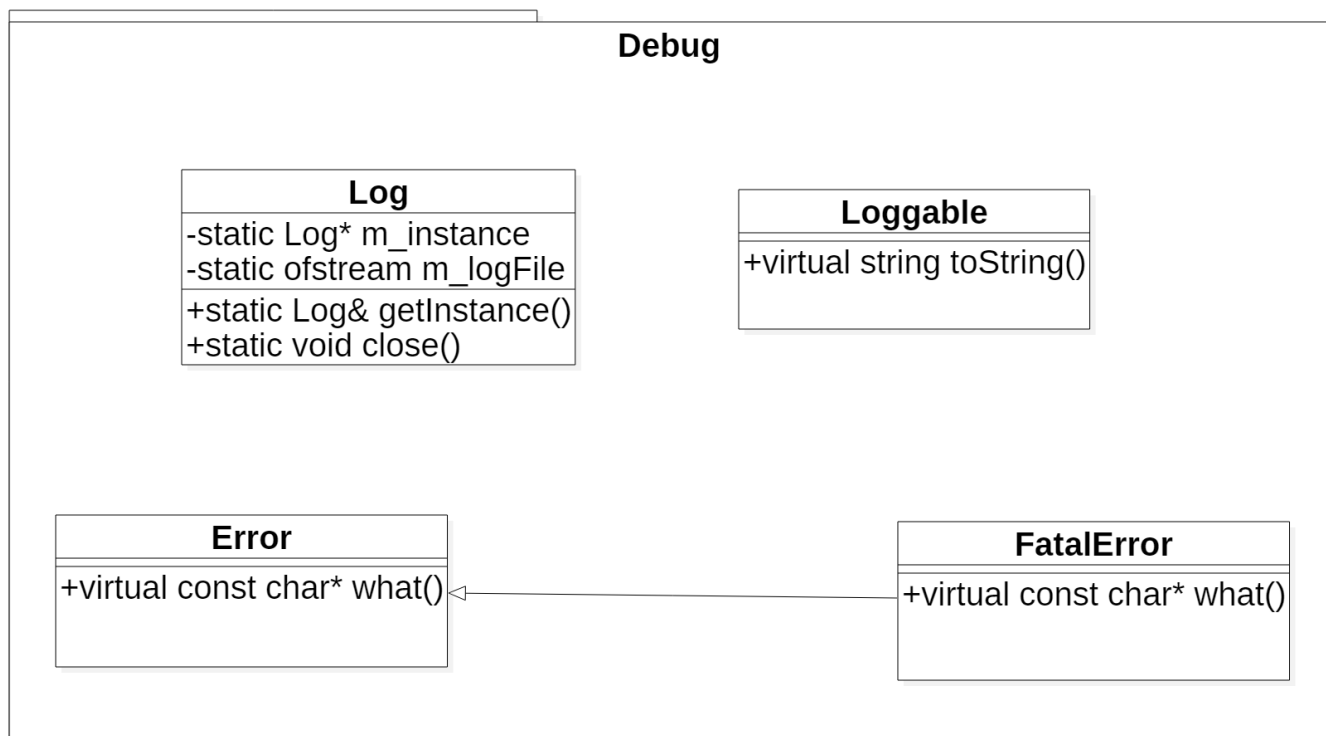
Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- **Package Text :**



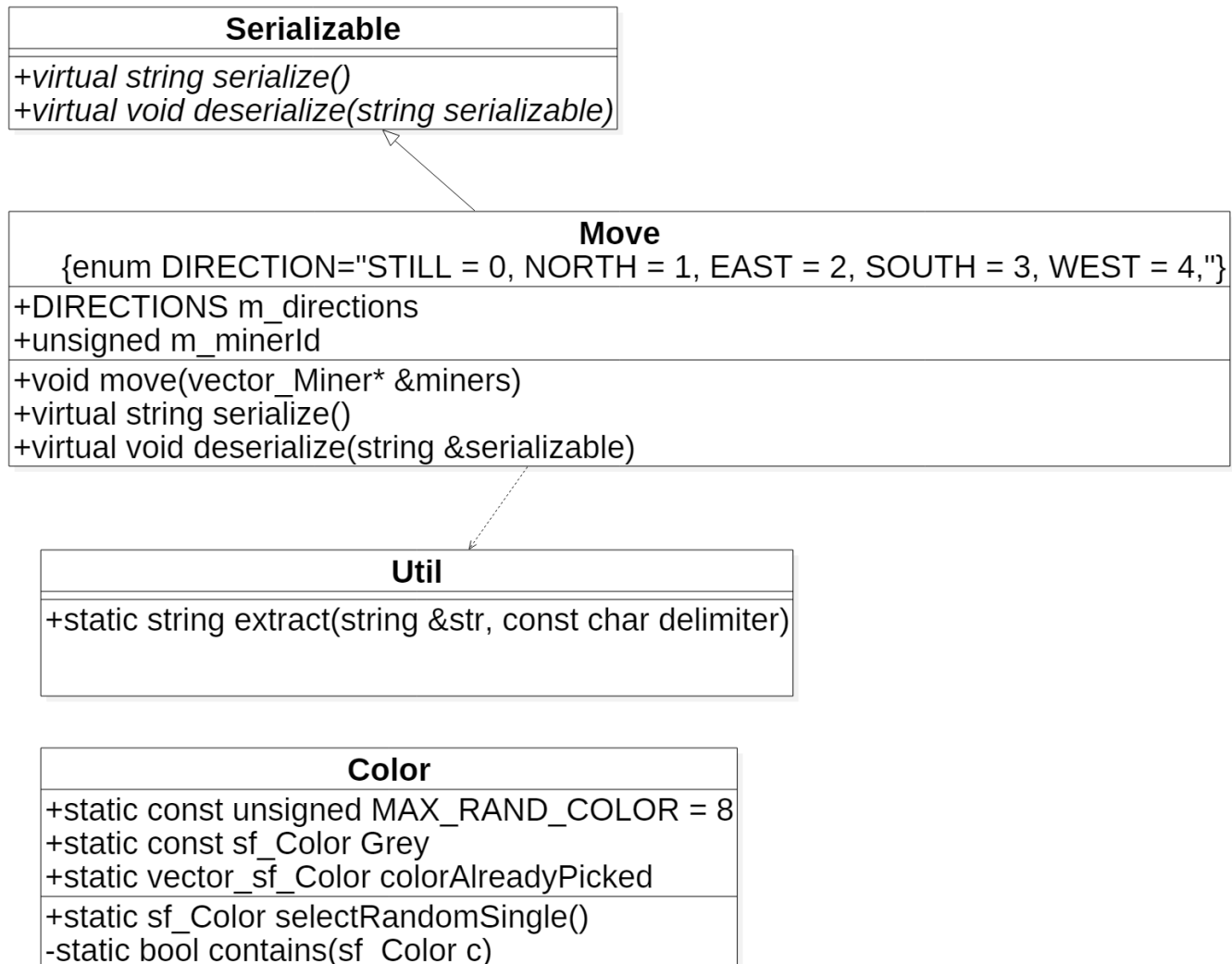
Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- **Package Debug :**



Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

• **Autres Classes :**



Nous allons voir comment toutes ces Classes fonctionnent ainsi que leur utilités dans les fonctionnalités décrites dans les Spécificité (ref : PRJ\_Polybasite).

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 4.2 Analyse statique

### 4.2.1 Classe « Game »

#### 4.2.1.1 Définition

La classe « Game » est la classe principale du jeu. Elle représente le jeu.

#### 4.2.1.2 Attributs

- Char\*\* : save\_argv : Arguments utilisé lors du lancement du jeu
- Int : save\_argc : Nombre d'arguments pour le jeu.
- GameState : m\_state : Etat du jeu

Structure « GameState » comprenant 6 états possible pour le jeu.

- STATE\_UNINITIALISED :

Premier état du jeu, il correspond au jeu non initialisé.

- STATE\_PLAY :

Etat principale du jeu, c'est dans cet état que les IA jouent et vont agir.

- STATE\_REPLAY :

Etat de relance du jeu cet état permet de relancer le jeu.

- STATE\_PAUSE :

Etat de pause du jeu.

- STATE\_FINISH :

Etat de fin du jeu lorsqu'une partie est terminée.

- STATE\_QUIT :

Etat de fin de jeu lors de cet état on se prépare à quitter le jeu.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- SF::RenderWindow : m\_main\_window : Fenêtre principale
- Map : m\_map : Carte du jeu

Grille de jeu de 30x30 cases

- Vector <Score\*> : m\_scores : Collection comprenant les scores
- Vector <Bot\*> : m\_bots : Collection comprenant les IA
- Unsigned : m\_nb\_turn : Nombre de tour.
- Unsigned : m\_turn\_speed : Vitesse de passage des tours.

Voici les constantes générales du jeu :

- Unsigned : SCREEN\_WIDTH = 1024 : Longueur horizontale de la fenêtre.
- Unsigned : SCREEN\_HEIGHT = 650 : Longueur vertical de la fenêtre.
- Unsigned : MAX\_TURN = 1500 : Nombre de tour Maximum.
- Unsigned : SPEED\_STEP = 50 : Vitesse pour le changement de tour.

#### 4.2.1.3 Opérations

- Start (int argc, char\* argv[])

La Méthode « Start » va initialiser le jeu, elle va créer la fenêtre de jeu et initialiser les variables et les affichages puis lancer la boucle de jeu via la méthode « loop () »

Les arguments :

Int argc : nombre d'arguments du programme au lancement.

Char\* argv[] : tableau des arguments fournis au programme au lancement.

- Restart ()

La Méthode « Restart » permet de relancer le jeu au départ et de refaire une partie.



Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- Quit ()

La Méthode « Quit » va permettre de libérer la mémoire pour bien fermer le jeu lors de sa fin. Elle ferme la fenêtre du jeu.

- Turn ()

Il s'agit de la Boucle d'actions du jeu c'est dans cette méthode que l'on va faire toutes les actions obtenues des IA.

Pseudo Code :

```

Turn()
|   For (bot : Game.m_bots)
|       |   bot.Turn() // exécutes les actions de chaque IA
|       |
|       |   IF (bot.getMinerNumber() = 0) // si l'IA n'a plus de Miners elle a perdu
|       |       |   bot.erase() // on supprime l'IA
|       |
|       Game.m_nb_turn + 1
|

```

- Loop ()

Il s'agit de la Boucle de jeu principale. Elle va actualiser les données du jeu ainsi que la Map.

Pseudo Code :

```

Loop()
|   while (Game.main_window.isOpen())
|       |   if (evenement == Close)
|       |       |   Game.quit()
|       |
|       Game.main_window.clear() // Clear screen
|       Game.map.draw() // Draw map
|       Game.main_window.display() // Display screen
|

```

- DisplayBotNames ()

Fonction d'affichage du nom des bots.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- Boolean : hasWinner ()

Méthode qui renvoi :

- « True » si la partie à une IA qui à gagné ou bien le jeu à atteint le nombre de tour Maximum (ref : PRJ\_Polybasite : UC 04)
- « False » si aucune IA n'a remporté la partie.

- DisplayWinner ()

Affiche le Gagnant de la Partie.

- DisplayScore ()

Affiche les Scores.

- DisplayTurn ()

Affiche le Numéro du Tour actuel.

#### 4.2.1.4 Fonctionnalité

Puisque la classe représente le jeu, elle va englober les autres classes nécessaires pour répondre aux spécificités.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 4.2.2 Classe « Map »

### 4.2.2.1 Définition

Cette classe représente la zone de jeu, elle permet de gérer la grille où les Entités se trouveront.

### 4.2.2.2 Attributs

Voici les différents attributs que possède la classe « Map » ainsi que leurs utilités :

- Vector <vector <Entity \*>> : m\_entities : Collection d'entité représentée sur la grille de jeu.
- Unsigned : m\_height : Hauteur de la zone de jeu
- Unsigned : m\_width : Largeur de la zone de jeu
- Unsigned : m\_blackHoleNumber : Nombre de « Trou Noir » que possède la Map.

Les constantes que possède la classe :

- Unsigned : OFFSET\_X = 0 : Coordonnée X du point de départ pour dessiner la Map
- Unsigned : OFFSET\_Y = 50 : Coordonnée Y du point de départ pour dessiner la Map

### 4.2.2.3 Opérations

Pour la l'intégration graphique via la sfml il est nécessaire d'implémenter certaines méthodes :

- Draw ()

Fonction de dessin, cette opération permet de dessiner la zone de jeu, la « map » sur la fenêtre graphique. Cette opération appelle la fonction de dessin des entités qui compose la « map ».

Algorithmes :

```

Draw :
|   Boucle [pour chaque entité dans la Map]
|   |   entité.Draw()
|   |
|   Fin Boucle

```

- Clear ()

Méthode qui va effacer toutes les entités de la map pour la remettre à zéro.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- CreateMap ()

Méthode de création de la Map qui va Remplir la Map avec les différentes Entité (Trou Noir et Basite).

Pseudo Code :

```

CreateMap :
|   For ( i = 0 ; i < m_width ; i++)
|       |   For (j = 0 ; j < m_height ; j++)
|           |   if ( random() % 100 <= BLACK_HOLE_PROB) // < 5%
|               |       m_blackHoleNumber + 1
|               |       m_entities[i][j] = new BlackHole(i, j)
|               |   else
|                   |       m_entities[i][j] = new Basite(i, j, random()% MAX_POWER) // = 8
|
|

```

- String : serialize ()

Méthode Importante qui va Mettre la Map sous forme d'une chaîne de caractères permettant ainsi de la transmettre plus aisément aux IA ou au jeu.

Pseudo Code :

```

Serialize :
|   String resultat
|   For ( i = 0 ; i < m_width ; i++)
|       |   For (j = 0 ; j < m_height ; j++)
|           |       resultat.concatene(m_entities[i][j].serialize() + « ; »)
|
|   return resultat

```

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- Deserialize (string &serializable)

Méthode inverse de serialize elle prend la chaîne de caractères et la retransforme en information utilisable par les IA et le jeu.

Pseudo Code :

```

Deserialize :
|   String buff
|   For ( i = 0 ; i < m_width ; i++)
|       |   For (j = 0 ; j < m_height ; j++)
|           |   buff = serializable.substring(serializable.find(« : »))
// On prend la sous chaîne correspondant à l'entité et on l'analyse
|   |   |   if (buff = « Entity »)
|           |   |   m_entities[i][j] = new Entity()
|           |   |   else if (buff = « BlackHole »)
|           |   |   |   m_entities[i][j] = new BlackHole()
|           |   |   else if (buff = « Basite »)
|           |   |   |   m_entities[i][j] = new Basite()
|           |   |   else if (buff = « Miner »)
|           |   |   |   m_entities[i][j] = new Miner()

```

#### 4.2.2.4 Fonctionnalité

La classe Map permet mettre en place la fonctionnalité de mise en place des entités ainsi avec la sfml cela permet de placer les entités graphiquement sur l'interface (ref : PRJ\_Polybasite 3.2.5 UC : 05). Cela permet aussi d'avoir une base pour le mouvement des mineurs (ref : PRJ\_Polybasite 3.2.3 UC : 03).

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

### 4.2.3 Classe « Entity »

#### 4.2.3.1 Définition

Cette classe représente les entités du jeu, celles qui feront et subiront les différentes actions, on compte trois types d'entité les « Mineurs », les minerais les « Basites » ainsi que les « Trou Noir ».

#### 4.2.3.2 Attributs

La liste des différents attributs que possèdent les « Entity » :

- Integer : m\_x : coordonnées X de l'entité.
- Integer : m\_y : coordonnées Y de l'entité.

Les constantes que possède la classe :

- Integer : ENTITY\_WIDTH = 20 : longueur de l'entité
- Integer : ENTITY\_HEIGHT = 20 : hauteur de l'entité

#### 4.2.3.3 Opérations

La liste des opérations des entités :

- Draw ()

Opération qui permet le dessin sur l'interface graphique. Elle crée une forme qui correspond à l'entité et la dessine sur la fenêtre graphique.

- String : serialize ()

Méthode Importante qui va Mettre l'Entité sous forme d'une chaîne de caractères permettant ainsi de la transmettre plus aisément aux IA ou au jeu.

Pseudo Code :

```
Serialize :
|   String resultat
|   resultat.concatene (« Entity : » + m_x + « ; » + m_y)
|   return resultat
```

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- Deserialize (string &serializable)

Méthode inverse de serialize elle prend la chaîne de caractères et la retransforme en information utilisable par les IA et le jeu.

Pseudo Code :

```

Deserialize :
|   String buff
|   buff = serializable.substring(serializable.find(« : »))
// On prend la sous chaîne correspondant à l'entité et on l'analyse
|
|   buff = Util.Extract ( serializable, « ; ») // retire les caractères de contrôle
|   m_x = toNumber(buff) // on prend la coordonnée x de l'entité
|
|   buff = Util.Extract ( serializable, « ; ») // retire les caractères de contrôle
|   m_y = toNumber(buff) // on prend la coordonnée y de l'entité

```

- Interact (Miner \*miner)

Méthode qui définit l'interaction entre une entité et un miner. Cette Méthode est à redéfinir pour chaque type d'entité (Miner, BlackHole, Basite) et doit faire les actions appropriées (ref : PRJ\_Polybasite UC : 03)

#### 4.2.3.4 Fonctionnalité

La classe « entité » utilise la méthode « Draw » avec la SFML pour pouvoir afficher les entités telle que définies par les spécificités (ref : PRJ\_Polybasite 3.2.5 UC : 05).

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

#### 4.2.4 Classe « Miner »

##### 4.2.4.1 Définition

La Classe « Miner » représente les Mineurs elle hérite de « Entity » les Mineurs sont les Entités qui peuvent se mouvoir d'après les actions données par les IA.

##### 4.2.4.2 Attributs

La Classe hérite des Attributs de la classe « Entity »

- Unsigned : m\_power : puissance de l'entité.
- Unsigned : m\_owner : numéro du camp du miner.
- Color : m\_color. : couleur du Mineur.

Les constantes que possède la classe :

- Unsigned : MAX\_POWER = 8 : puissance maximum d'un Miner.

##### 4.2.4.3 Opération



Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- Interact (Miner \*miner)

Méthode qui définit l'interaction entre un mineur et un autre mineur (ref : PRJ\_Polybasite UC : 03)

Pseudo Code :

```

Interact :
|   Bot chef = Game.m_bots[getOwner()] // on prend le bot qui possède ce mineur
|   Bot target = Game.m_bots[miner.getOwner()] // et le bot qui possède l'autre mineur
|   if (getOwner() != miner.getOwner()) // si les mineurs sont ennemis
|       |   if (getPower() == miner.getPower())
|           |   Game.m_map.kill(this) // le mineur meurt
|           |   chef.deleteMiner(this)
|           |   Game.m_map.kill(miner) // le mineur ennemi meurt
|           |   target.deleteMiner(miner)
|
|       |   else if (getPower() > miner.getPower())
|           |   m_power = 0
|           |   chef.addMiner(miner)
|           |   target.deleteMiner(miner)
|
|       |   else if (getPower() < miner.getPower())
|           |   miner.m_power = 0
|           |   target.addMiner(this)
|           |   chef.deleteMiner(this)
|
|   else // si les mineurs sont alliés
|       |   Int remaining = (MAX_POWER - 1) - miner.getPower()
|       |   if (remaining > 0)
|           |   |   if (m_power >= remaining)
|               |   |   miner.setPower(miner.getPower() + remaining)
|               |   |   m_power = remaining
|           |   |   else
|               |   |   miner.setPower(miner.getPower() + m_power)
|               |   |   m_power = 0

```

// TODO

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- Draw ()

Opération qui permet le dessin sur l'interface graphique. Elle crée une forme qui correspond à l'entité et la dessine sur la fenêtre graphique.

Les « Mineur » seront des carrés et utiliseront la couleur de l'IA qui les contrôle.

- String : serialize ()

Fonction Similaire a Serialize de la classe Entité sauf qu'elle agit pour tous les attributs du miner.

- Deserialize (string &serializable)

Fonction Similaire a Deserialize de la classe Entité sauf qu'elle agit pour tous les attributs du miner.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 4.2.5 Classe « Basite »

### 4.2.5.1 Définition

La Classe « Basite » représente les Minerais elle hérite de « Entity » les Minerais sont des Entités Inerte. Qui seront attaqué par les Mineurs.

### 4.2.5.2 Attributs

La Classe hérite des Attributs de la classe « Entity »

- Integer : m\_power : puissance de l'entité.

Les constantes que possède la classe :

- Unsigned : MAX\_POWER = 8 : Puissance maximum d'un Basite.

### 4.2.5.3 Opération

- Draw ()

Opération qui permet le dessin sur l'interface graphique. Elle crée une forme qui correspond à l'entité et la dessine sur la fenêtre graphique.

Pour le cas des « Basite » on dessinera des carrés blancs

Pseudo Code :

```

Draw :
|   int positionX = ENTITY_WIDTH * x
|   int positionY = ENTITY_HEIGHT * y
|
|   Rectangle rectangle = new Rectangle(ENTITY_WIDTH, ENTITY_HEIGHT)
|   rectangle.move(positionX, positionY)
|   rectangle.fill(Color.GREY)
|
|   int power = MAX_POWER + (MAX_POWER - power)
|
|   Game.main_window.draw(rectangle)

```

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- Deserialize (string &serializable)

Fonction Similaire a Deserialize de la classe Entité sauf qu'elle agit pour tous les attributs de la Basite.

- Interact (Miner \*miner)

Méthode qui définit l'interaction entre une Basite et un miner (ref : PRJ\_Polybasite UC : 03)

```

Interact :
|   Bot target = Game.m_bots[miner.getOwner()] // le bot qui possède l'autre mineur
|   if (miner.getPower() >= m_power)
|       |   target.addMiner(m_x, m_y, miner.getPower() - m_power) // le miner gagne
|       |   miner.setPower(miner.getPower() - m_power)
|       |
|       else
|           |   target.deleteMiner(miner) // le miner perd, il meurt
|           |   Game.m_map.setEntity(new Basite(miner.get_X(), miner.get_Y(), 0))
|           |

```

- Serialize ()

Fonction Similaire a Serialize de la classe Entité sauf qu'elle agit pour tous les attributs de la Basite.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 4.2.6 Classe « BlackHole »

### 4.2.6.1 Définition

La Classe « BlackHole » représente les Trou Noir elle hérite de « Entity » les Trou Noir sont des entités semblable au Minerai sauf qu'ils possèdent une puissance infinie et donc gagne systématiquement face au mineurs.

### 4.2.6.2 Attributs

La Classe hérite des Attributs de la classe « Entity »

- Texture : m\_texture : texture du « Trou Noir »
- Unsigned : m\_rotation : variable pour l'animation du « Trou Noir »

Les constantes que possède la classe :

- Integer : BLACK\_HOLE\_PROB = 5 : probabilité de création d'un trou noir.

### 4.2.6.3 Opération

La Classe hérite de la Méthode d' « Entity »

- Draw ()

Opération qui permet le dessin sur l'interface graphique. Elle crée une forme qui correspond à l'entité et la dessine sur la fenêtre graphique.

Pour les « Trou Noir » il s'agira de carré de couleur noir.

- Deserialize (string &serializable)

Fonction Similaire a Deserialize de la classe Entité sauf qu'elle agit pour tous les attributs du BlackHole.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- Interact (Miner \*miner)

Méthode qui définit l'interaction entre une Basite et un mineur (ref : PRJ\_Polybasite UC : 03)

Pseudo Code :

```
Interact :
|   Bot target = Game.m_bots[miner.getOwner()] // le bot qui possède l'autre mineur
|
|   target.deleteMiner(miner)
|
|   Gmae.m_map.detEntity(new Basite(miner.getX(), miner.getY(), 0))
```

- String : serialize ()

Fonction Similaire a Serialize de la classe Entité sauf qu'elle agit pour tous les attributs du BlackHole.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 4.2.7 Classe « Bot »

### 4.2.7.1 Définition

La Classe « Bot » représente les IA elle permet de faire l'interface entre le jeu et les IA.  
Une IA est représentée par un objet Bot. Elle contient donc naturellement les informations et moyen de connections aux IA.

### 4.2.7.2 Attributs

- String : m\_name : le nom de l'IA
- Integer : m\_owner : le Camp de l'IA
- Color : m\_color : la couleur du camp de l'IA
- Vector<Miner\*> : m\_miners : Collection de mineur que possède l'IA
- Socket\* m\_socket : socket de connexion à l'IA
- Thread m\_instance : instance de l'IA
- Integer m\_port : port de connexion à l'IA

### 4.2.7.3 Opération

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- Turn ()

Il s'agit de la Méthode qui permettra de communiquer avec les IA et de les faire jouer. (ref : PRJ\_Polybasite UC : 04)

Pseudo Code :

```

Turn :
|   For (miner : m_miners) // on augmente la puissance des mineurs présents
|   |   if (miner.getPower() < MAX_POWER)
|   |   |   miner.setPower(miner.getPower() + 1)
|
|   m_socket.send(Game.m_map) // on envoi la Map au IA
|   String moveString = m_socket.receive()
|   moves[]
|
|   Boucle (moveString.size() > 0) // on decode les Mouvements reçu
|   |   Move m
|   |   move.deserialize(moveString)
|   |   moves.add(move)
|
|   For (move : moves) // on exécute les mouvements
|   |   move.move(m_miners)
|   |

```

- AddMiner (unsigned x, unsigned y, unsigned power)

Méthode qui ajoute un mineur en position \$x,\$y avec une puissance de \$power au camp de l'IA

- Unsigned : getMinerNumber ()

Méthode qui retourne le nombre total de mineur que possède l'IA.

- deleteMiner (Miner \*miner)

Méthode qui supprime un mineur \$miner du camp de l'IA



Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

#### 4.2.8 Classe « Socket »

##### 4.2.8.1 Définition

La Classe Socket Permet de définir les socket qui vont permettre la connexion du jeu au IA.

##### 4.2.8.2 Attributs

- Integer : m\_port : Port de connexion de la Socket.
- SF::UdpSocket : m\_socket : Socket de la SfmL qui va établir le contact entre le jeu et l'IA.

##### 4.2.8.3 Opération

- String : Receive ()

Méthode qui retourne les données reçu par la socket.

- String : Receive (int timeout)

Même méthode mais permet de spécifier un temps limite \$timeout pour la réception de données.

- Send (string ipAdress, int port, string msg)

Méthode qui permet d'envoyer des données de type chaîne de caractères par la socket

- Send (string ipAdress, int port, Serializable\* serializable)

Même méthode mais envoi des données de type serializable c'est-à-dire qui peuvent être formaté en String.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 4.2.9 Classe « Serializable »

### 4.2.9.1 Définition

Classe permettant de formaté des données en chaines de caractères pour les envoyer et les recevoir plus facilement.

### 4.2.9.2 Attributs

Aucuns attributs pour cette classe.

### 4.2.9.3 Opération

- Virtual String : serialize ()

Méthode Importante qui va Mettre l'Entité sous forme d'une chaine de caractères permettant ainsi de la transmettre plus aisément aux IA ou au jeu.

- Virtual : deserialize (string serializable)

Méthode inverse de serialize elle prend la chaine de caractères et la retransforme en information utilisable par les IA et le jeu.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

#### 4.2.10 Classe « Move »

##### 4.2.10.1 Définition

Classe représentant un Mouvements d'entité dans notre jeu.

##### 4.2.10.2 Attributs

- DIRECTIONS : m\_directions :

Structure « DIRECTIONS » comprenant 5 directions possible pour le mouvement.

- STILL :

Aucun Mouvement

- NORTH :

Mouvement vers le haut

- EAST :

Mouvement vers la droite

- SOUTH :

Mouvement vers le bas

- WEST :

Mouvement vers la gauche

- Unsigned : m\_minerId : numéro représentant le miner.

##### 4.2.10.3 Opération

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- Move (vector<Miner \*> &miners)

Méthode permettant d'appliquer un Mouvement a un minier.

Pseudo Code :

```

Move :
|   miner = miners[m_minerId]
|   int offset_x = miner.getX()
|   int offset_y = miner.getY()
|
|   Switch (m_directions)
|       case NORTH :
|           |   offset_y = (offset_y - 1 +
Game.m_map.getHeight())%Game.m_map.getHeight()
|
|           case EAST :
|               |   offset_x = (offset_x + 1 +
Game.m_map.getWidth())%Game.m_map.getWidth()
|
|               case SOUTH :
|                   |   offset_y = (offset_y + 1 +
Game.m_map.getHeight())%Game.m_map.getHeight()
|                   case EAST :
|                       |   offset_x = (offset_x + 1 +
Game.m_map.getWidth())%Game.m_map.getWidth()
|
|   if (m_directions != STILL)
|       |   entity = Game.m_map[offset_x][offset_y]
|       |   entity.interact(miners[m_minerId])

```

- String : serialize ()
- Deserialize (string &serializable)

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 4.2.11 Classe « Color »

### 4.2.11.1 Définition

Classe permettant de faire des opérations sur les Couleurs

### 4.2.11.2 Attributs

- Sf::Color : Grey : couleurs grise
- Vector <Sf::Color> : colorAlreadyPicked : couleurs déjà utilisé par les IA

Les constantes que possède la classe :

- Unsigned : MAX\_RANDOM\_COLOR = 8 :

### 4.2.11.3 Opération

- Sf::Color : SelectRandomSingle ()

Méthode qui choisi aléatoirement une couleur à donner a un IA

- Bool : Contains (sf::Color c)

Méthode qui renvoi true si la couleur a déjà été sélectionnée sinon false.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 4.2.12 Classe « Util »

### 4.2.12.1 Définition

Classe contenant des méthodes utilitaires.

### 4.2.12.2 Attributs

### 4.2.12.3 Opération

- String : extract (string &str, char delimiter)

Méthode qui permet d'extraire de la chaîne de caractère une sous chaîne séparée par le \$delimiter.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

#### 4.2.13 Classe « Score »

##### 4.2.13.1 Définition

Classe permettant la gestion des scores ainsi que leurs affichages.

##### 4.2.13.2 Attributs

- Unsigned : m\_position : Position du score
- Bot \* : m\_bot : Bot à qui le score appartient.

Les constantes que possède la classe :

- Unsigned : SCORE\_OFFSET\_X = 605 : Coordonnées X de la position de départ des scores
- Unsigned : SCORE\_OFFSET\_Y = 75 : Coordonnées Y de la position de départ des scores
- Unsigned : SCORE\_WIDTH = 420 : Longueur de la zone d'affichage d'un score.
- Unsigned : SCORE\_HEIGHT = 100 : Hauteur de la zone d'affichage d'un score.

##### 4.2.13.3 Opération

- Draw ()

Méthode qui va dessiner le score sous forme de barres de progression.

- DrawLegend ()

Méthode qui va dessiner la légende utilisé pour les scores.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

#### 4.2.14 Classe « ProgressBar »

##### 4.2.14.1 Définition

Classe représentant les barres de progression utilisée par les scores.

##### 4.2.14.2 Attributs

- Float : m\_percent : Pourcentage de la barre
- Sf::Color : m\_color : Couleur de la barre
- Sf::Vector2f : m\_size : taille de la barre.
- Sf::Texture \* : m\_texture : texture utilisé par la barre
- Sf::IntRect : m\_textureRectangle : Rectangle qui va contenir la Texture.
- Bool : m\_showBackground : Booléen pour activer le background ou pas.
- SF::Texture \* : m\_backgroundTexture : texture d'arrière plan
- SF::RectangleShape : m\_backgroundAndFrame : Rectangle qui va contenir la barre.
- SF::IntRect m\_backgroundTextureRectangle : rectangle qui va contenir les texture d'arrière plan.

##### 4.2.14.3 Opération

- Sf::FloatRect : getLocalBounds () :

Méthode pour obtenir le rectangle de dessin avec des coordonnées Local

- Sf::FloatRect : getGlobalBounds () :

Méthode pour obtenir le rectangle de dessin avec des coordonnées Global

- Sf::Vector2f : getAnchorProgressTop () :

Sert à obtenir les dimensions de la barre remplie à 100%

- Sf::Vector2f : getAnchorProgressCenter () :

Sert à obtenir les dimensions de la barre remplie à 50%

- Sf::Vector2f : getAnchorProgressBottom () :

Sert à obtenir les dimensions de la barre remplie à 0%



Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

- Draw (Sf::renderTarget &target, Sf::RenderStates states)

Permet de dessiner la barre de Progression.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 4.2.15 Classe « RichText »

### 4.2.15.1 Définition

Classe permettant de créer des textes améliorés.

### 4.2.15.2 Attributs

- Vector<Line> : m\_line : Lignes de textes
- Sf::FloatRect : m\_bounds : Rectangle qui va contenir le texte
- Unsigned int : m\_characterSize : Taille des caractères
- Sf::Font \* : m\_font : Police de caractère utilisée.
- Sf::Color : m\_color : Couleur du texte
- Sf::Text::Style : m\_style : Style du texte (gras souligné ...)

### 4.2.15.3 Opération

- Draw (sf::RenderTarget &target, sf::RenderStates states) :

Méthode pour dessiner le texte avec un état \$states a l'endroit indiqué par \$target.

- Vector<sf::String> : explode (Sf::String &str, Sf::Uint32 delimiter)

Méthode qui va couper une chaine de caractères \$str par rapport au délimiteur \$delimiter.

- Sf::Text : createText(Sf::String &string)

Méthode qui va créer une Text affichable par rapport à la String \$string.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 4.2.16 Classe « Line »

### 4.2.16.1 Définition

Classe représentant une ligne de texte.

### 4.2.16.2 Attributs

- Vector<Sf::Text> : m\_texts : le Texte à afficher
- Sf::FloatRect : m\_bounds : le rectangle qui va contenir la ligne.

### 4.2.16.3 Opération

- Draw (sf::RenderTarget &target, sf::RenderStates states) :

Méthode pour dessiner le texte avec un état \$states a l'endroit indiqué par \$target.

- UpdateTextAndGeometry (Sf::Text &text)

Permet la mise à jour du texte dans la ligne.

- AppendText (Sf::Text text)

Permet de concaténer du texte.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 4.2.17 Classe « Log »

### 4.2.17.1 Définition

Classe permettant d'écrire dans un fichier de Log les différents commentaires et erreurs possibles. Utilisé pour le débogage.

### 4.2.17.2 Attributs

- Log \* : m\_instance : instance d'objet Log.
- Ofstream : m\_logFile : flux de texte qui permet de remplir le fichier de log.

### 4.2.17.3 Opération

- Log & : getInstance () :

Méthode de création d'un objet Log permet d'obtenir une et une seule instance de l'objet dans tout le programme et ouvre le fichier de log.

- Close () :

Ferme le fichier de log et supprime l'instance d'objet Log.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 4.2.18 Classe « Loggable »

### 4.2.18.1 Définition

Permet de Convertir les informations du jeu en chaines de caractères pour être ajouté au fichier de log.

### 4.2.18.2 Opération

- String toString ()

Méthode qui va convertir les informations des objets du jeu en String.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 4.2.19 Classe « Error »

### 4.2.19.1 Définition

Classe représentant les Erreurs

### 4.2.19.2 Opération

- Char \* What () :

Méthode qui décrit l'erreur et indique sa position.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 4.2.20 Classe « FatalError »

### 4.2.20.1 Définition

Classe représentant les erreurs fatale.

### 4.2.20.2 Opération

- Char \* What () :

Méthode qui décrit l'erreur et indique sa position.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

### 4.3 Correspondance Classes Fonctionnalité

Fonction Classe	UC : 01 Lancer le Jeu	UC : 05.1 & UC : 05.2 Affichage Grille de jeu & Entités	UC : 02 & UC : 05.4 Calcul & Affichage du Nombre de Tours	UC : 04.2 & UC 05.3 Calcul & Affichage Scores	UC : 03 Mouvement des Mineurs	UC : 04 IA
Game	X	X	X	X	X	X
Map		X			X	
Entity		X				
Miner		X			X	
Basite		X				
BlackHole		X				
Bot				X	X	X
Socket						X
Score				X		
ProgressBar				X		
RichText			X	X		
Line			X	X		
Serializable					X	X
Move					X	X
Color		X		X	X	X



Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

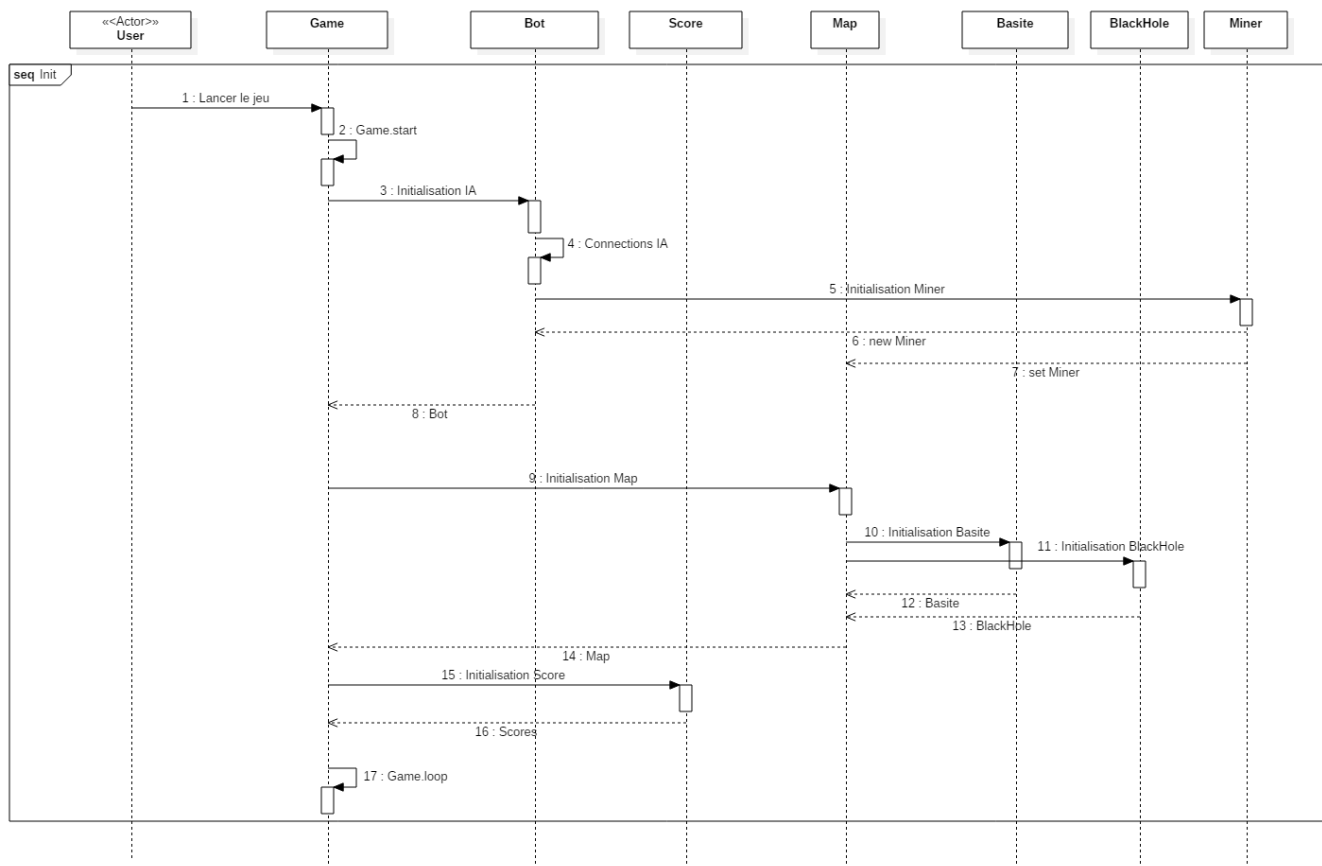
Util					X	X
Log						
Loggable						
Error						
FatalError						

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 5 Aspects dynamiques

### 5.1 Echanges entre éléments

Voici l'étude dynamique des classes lors de l'initialisation du jeu au lancement de celui-ci.



Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	Projet Guide pour l'élaboration d'une conception générale	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	---	---

## 5.2 Comportement du logiciel en erreur

Lors d'une erreur, notre logiciel va effectuer ces actions :

- avertir l'utilisateur d'une erreur
- Libérer les différentes Allocations effectuées
- créer des fichiers de logs
- se fermer

Pour revenir à un mode d'utilisation normale il suffira de relancer le jeu.

## 5.3 Modes de fonctionnement

Le logiciel possède deux modes d'utilisation :

- nominal

Le fonctionnement simple du jeu

- débogage

Un mode permettant l'affichage de plus d'information utile seulement pour le débogage l'utilisateur ni a pas accès.

Ref : Polybasite - CCG - 5.0 Emetteur : STAMEGNA Clément MICHON Guillaume Client : MAVROMATIS Sébastien Projet : Polybasite	<b>Projet</b> <b>Guide pour l'élaboration d'une conception</b> <b>générale</b>	Date: 11 Février 2018 Version : 5.0 Service : Polytech Marseille Etat : Préliminaire
---	--	---

## 6 Allocation des ressources

### 6.1 Gestion de la mémoire

Polybasite est codée en c++ il n'y a donc pas de ramasse-miettes ou autre dispositif de gestion de la mémoire. La gestion de la mémoire est donc à faire manuellement. Via les allocations et libérations de mémoire propre au langage.

Les Allocations notables dans notre projet sont.

Au niveau des allocations :

- Le fenêtrage
- Les zones de jeu
- Les entités