Bomberman

0.1

Généré par Doxygen 1.8.17

Chapter 1

Bomberman

Il s'agit de notre projet d'INFO0402 dans lequel nous avons du développer le jeu du **Bomberman**.

Version 0.1

1.1 Auteurs

- CHEMIN Pierre
- HADID Hocine

1.2 Architecture du projet

En ce qui concerne l'architecture de notre projet, elle est assez simpliste.
A la racine vous disposerez des dossiers et fichiers suivants :

- Code [Dossier] => Le code du Bomberman
- Rapport [PDF] => Le rapport du projet
- Doc [PDF] => La documentation généré à l'aide de Doxygen
- Ce ReadMe

1.2.1 Le dossier "Code"

Dans ce dossier, nous retrouverons les dossiers et fichiers suivats :

- doc [Dossier] => Le dossier contient la documentation sous format html, latex et pdf
- src [Dossier] => Le dossier contenant tous les "packages"
- resources [Dossier] => Les ressources nécessaires, c'est-à-dire les maps au format .txt
- ProgBomberman.cpp => Le fichier contenant le main du jeu Bomberman et permettant de lancer une partie
- makefile => Le fichier permettant la création de l'exécutable intitulé "Bomberman"
- Doxyfile => Le fichier aidant à générer la documentation

1.2.2 L'organisation du dossier src

Le dossier "src" contenant tout le code du projet est subdivisé en 2 dossiers, le dossier "include" contenant tous les '.h' et le dossier "src" contenant tous les '.cpp'.
Ces dossiers sont également divisés en sous-dossier correspondant à nos "packages" expliqués dans le rapport.

1.3 Création d'un exécutable

Pour créer un exécutable du jeu Bomberman, il vous suffit d'utiliser le makefile disponible avec la commande suivante :

```
make
```

Une fois l'exécutable intitulé *Bomberman* créé il vous suffit de le lancer à l'aide de la commande suivante :

```
./Bomberman
```

Une fois tout cela effectué, il n'y a plus qu'ç jouer une partie !

Chapter 2

Index des espaces de nommage

2.1 Liste des espaces de nommage

Liste de tous les espaces de nommage documentés avec une brève description:

Chapter 3

Index hiérarchique

3.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

Chapter 4

Index des classes

4.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

Chapter 5

Index des fichiers

5.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

Chapter 6

Documentation des espaces de nommage

6.1 Référence de l'espace de nommage utilities

Énumérations

- enum EDirection {
  TOP, BOTTOM, LEFT, RIGHT,
  NONE }

  Enumeration de direction.
- enum EBombExplosionDirection { LINE, COLUMN, CENTER, NOEXPLOSION }

  Enumeration de direction d'explosion.

6.1.1 Description détaillée

Espace de nommage regroupant les éléments pouvant être utile de manière globale au projet

6.1.2 Documentation du type de l'énumération

6.1.2.1 EBombExplosionDirection

`enum utilities::EBombExplosionDirection`
Enumeration de direction d'explosion.
Enumeration pour clarifier l'orientation de l'explosion d'une bombe

Valeurs énumérées

LINE	Explosion en ligne par rapport au centre de l'explosion
COLUMN	Explosion en colonne par rapport au centre de l'explosion
CENTER	On se situe au centre de l'explosion
NOEXPLOSION	Il n'y a pas d'explosion

6.1.2.2 EDirection

`enum utilities::EDirection`
Enumeration de direction.
Enumeration pour clarifier la direction de déplacements

Valeurs énumérées

TOP	Direction vers le haut

Valeurs énumérées

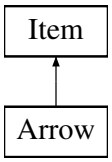
BOTTOM	Direction vers le bas
LEFT	Direction vers la gauche
RIGHT	Direction vers la droite
NONE	Aucune directon (pas de déplacement)

Chapter 7

Documentation des classes

7.1 Référence de la classe Arrow

La classe Arrow hérite de la classe Item et gère les flèches lancées par le Bowman.
`#include <Arrow.h>`
Graphe d'héritage de Arrow:



Fonctions membres publiques

- Arrow (int x=0, int y=0, int speed=1, int damage=1, utilities::EDirection direction=utilities::EDirection::NONE)

  Constructeur de la classe Arrow.
- void move ()

  Déplace la flèche.
- virtual void showTop () const override

  Affiche la partie haute de la case lors de l'affichage de la flèche.
- virtual void showMiddle () const override

  Affiche la partie du milieu de la case lors de l'affichage de la flèche.
- virtual void showBottom () const override

  Affiche la partie basse de la case lors de l'affichage de la flèche.
- virtual bool play (std::vector< std::vector< Tile * >> map, Bomberman *player, std::vector< Item * > *items) override

  Permet de jouer la flèche.

Membres hérités additionnels

7.1.1 Description détaillée

La classe Arrow hérite de la classe Item et gère les flèches lancées par le Bowman.

Voir également

Item

Permet de gérer les flèches présentes sur la Map

Auteur

Hocine HADID

7.1.2 Documentation des constructeurs et destructeur

7.1.2.1 Arrow()

```
Arrow::Arrow (
            int x = 0,
            int y = 0,
            int speed = 1,
            int damage = 1,
            utilities::EDirection direction = utilities::EDirection::NONE )
```

Constructeur de la classe Arrow.

Paramètres

x	La ligne où se situe la flèche
y	La colonne où se situe la flèche
speed	La vitesse de la flèche
damage	Les dégâts de la flèche
direction	La direction de la flèche

Voir également

utilities

7.1.3 Documentation des fonctions membres

7.1.3.1 move()

```
void Arrow::move ( )
```

Déplace la flèche.
Permets de déplacer la flèche en fonction de direction et de sa vitesse

Auteur

Hocine HADID

7.1.3.2 play()

```
bool Arrow::play (
            std::vector< std::vector< Tile * >> map,
            Bomberman * player,
            std::vector< Item * > * items )  [override], [virtual]
```

Permet de jouer la flèche.

Voir également

Bomberman

Item

Tile

Joue la flèche, vérifie si la flèche une fois déplacé est sur la même case que le Bomberman, si c'est le cas inflige des dégâts au Bomberman sinon vérifie si la flèche n'est pas sortie de la map ou si la case sur laquelle elle souhaite se déplacer est libre. Si la flèche serait encore dans la map et que la case est libre alors déplace la flèche.

Paramètres

map	La map sur laquelle la flèche se déplace
player	Le joueur présent sur la map
items	la liste des items présents sur la map

Renvoie

true si la flèche doit être supprimer sinon retourne false

Auteur

Hocine HADID

Implémente Item.

7.1.3.3 showBottom()

`void Arrow::showBottom ( ) const [override], [virtual]`
Affiche la partie basse de la case lors de l'affichage de la flèche.

Auteur

Hocine HADID

Implémente Item.

7.1.3.4 showMiddle()

`void Arrow::showMiddle ( ) const [override], [virtual]`
Affiche la partie du milieu de la case lors de l'affichage de la flèche.

Auteur

Hocine HADID

Implémente Item.

7.1.3.5 showTop()

`void Arrow::showTop ( ) const [override], [virtual]`
Affiche la partie haute de la case lors de l'affichage de la flèche.

Auteur

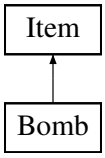
Hocine HADID

Implémente Item.
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Items/Arrow.h
- src/src/Items/Arrow.cpp

7.2 Référence de la classe Bomb

La classe Bomb hérite de la classe Item et permets de gérer les flèches.
`#include <Bomb.h>`
Graphe d'héritage de Bomb:



Fonctions membres publiques

- Bomb (int x=0, int y=0, int turnBeforeExplosion=3)

  Constructeur d'une Bombe.
- ~Bomb ()

  Destructeur de la bombe.
- double getScope () const

  accesseur de l'attribut scope
- int getTurnBeforeExplosion () const

  accesseur de l'attribut turnBeforeExplosion
- void decreaseTurnBeforeExplosion ()

  décrémente le nombre de tour avant l'explosion de la bombe
- void showTop () const override

  Affiche la partie haute de la case lors de l'affichage de la bombe.
- void showMiddle () const override

  Affiche la partie du milieu de la case lors de l'affichage de la bombe.
- void showBottom () const override

  Affiche la partie basse de la case lors de l'affichage de la bombe.
- virtual bool play (std::vector< std::vector< Tile * >> map, Bomberman *player, std::vector< Item * > *items) override

  Permet de jouer la bombe.

Attributs publics statiques

- static int power = 1
- static double scope = 1.5

Membres hérités additionnels

7.2.1 Description détaillée

La classe Bomb hérite de la classe Item et permets de gérer les flèches.

Voir également

Item

Permet de gérer les bombes posées par le Bomberman

Auteur

Hocine HADID

7.2.2 Documentation des constructeurs et destructeur

7.2.2.1 Bomb()

```
Bomb::Bomb (
            int x = 0,
            int y = 0,
            int turnBeforeExplosion = 3 )
```

Constructeur d'une Bombe.

Paramètres

x	La ligne sur laquelle se situe la bombe
y	La colonne sur laquelle se situe la bombe
turnBeforeExplosion	Le nombre de tour restant avant l'explosion de la bombe

7.2.2.2 ~Bomb()

`Bomb::~Bomb ( )`
Destructeur de la bombe.

7.2.3 Documentation des fonctions membres

7.2.3.1 decreaseTurnBeforeExplosion()

`void Bomb::decreaseTurnBeforeExplosion ( )`
décrémente le nombre de tour avant l'explosion de la bombe

Auteur

Hocine HADID

7.2.3.2 getScope()

`double Bomb::getScope ( ) const`
accesseur de l'attribut scope

Renvoie

double La porté de la bombe

7.2.3.3 getTurnBeforeExplosion()

`int Bomb::getTurnBeforeExplosion ( ) const`
accesseur de l'attribut turnBeforeExplosion

Renvoie

int Le nombre de tour avant l'explosion de la bombe

7.2.3.4 play()

```
bool Bomb::play (
            std::vector< std::vector< Tile * >> map,
            Bomberman * player,
            std::vector< Item * > * items )  [override], [virtual]
```
Permet de jouer la bombe.

Voir également

Bomberman

Item

Tile

Joue la bombe, si le nombre de tour est supérieur à 0, appelle la méthode decreaseTurnBeforeExplosion.

Paramètres

map	La map sur laquelle la bombe est joué
player	Le joueur présent sur la map
items	La liste des items présents sur la map

Renvoie

true si la bombe doit être supprimé, false sinon

Auteur

Hocine HADID

Implémente Item.

7.2.3.5 showBottom()

`void Bomb::showBottom ( ) const [override], [virtual]`
Affiche la partie basse de la case lors de l'affichage de la bombe.

Auteur

Hocine HADID

Implémente Item.

7.2.3.6 showMiddle()

`void Bomb::showMiddle ( ) const [override], [virtual]`
Affiche la partie du milieu de la case lors de l'affichage de la bombe.

Auteur

Hocine HADID

Implémente Item.

7.2.3.7 showTop()

`void Bomb::showTop ( ) const [override], [virtual]`
Affiche la partie haute de la case lors de l'affichage de la bombe.

Auteur

Hocine HADID

Implémente Item.

7.2.4 Documentation des données membres

7.2.4.1 power

`int Bomb::power = 1 [static]`
La puissance de la bombe

7.2.4.2 scope

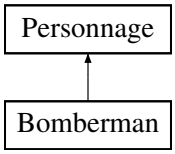
```
double Bomb::scope = 1.5  [static]
```

La portée de la bombe
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Items/Bomb.h
- src/src/Items/Bomb.cpp

7.3 Référence de la classe Bomberman

La classe Bomberman hérite de la classe Personnage et permets de gérer un Bomberman (le joueur)
`#include <Bomberman.h>`
Graphe d'héritage de Bomberman:



Fonctions membres publiques

- Bomberman (int x=0, int y=0, int life=3, int speed=1, int bombMax=5, int nbBomb=2)

  Constructeur d'un Bomberman.
- ~Bomberman ()

  Destructeur du Bomberman.
- int getBombMax () const

  Accesseur de l'attribut m_bombMax.
- int getNbBomb () const

  Accesseur de l'attribut m_nbBomb.
- void addBomb (int nbBomb)

  Permets d'ajouter des bombes au Bomberman.
- void addLife (int life)

  Permets d'ajouter de la vie au Bomberman.
- void addSpeed (int speed)

  Permets d'ajouter de la vitesse au Bomberman.
- void dropBomb ()

  Permets de retirer une bombe au Bomberman.
- virtual bool move (utilities::EDirection direction)

  Permets de déplacer le Bomberman dans une direction précise.
- bool move (utilities::EDirection direction, int speed)

  Permets de déplacer le Bomberman dans une direction et une vitesse précise.
- virtual void show () const override

  Permets d'afficher le Bomberman.

Membres hérités additionnels

7.3.1 Description détaillée

La classe Bomberman hérite de la classe Personnage et permets de gérer un Bomberman (le joueur)

Voir également

Personnage

Permets de gérer un Bomberman, il s'agit également du joueur d'une partie

Auteur

Pierre CHEMIN

7.3.2 Documentation des constructeurs et destructeur

7.3.2.1 Bomberman()

```
Bomberman::Bomberman (
            int x = 0,
            int y = 0,
            int life = 3,
            int speed = 1,
            int bombMax = 5,
            int nbBomb = 2 )
```

Constructeur d'un Bomberman.

Paramètres

x	La ligne sur laquelle se situe le Bomberman
y	La colonne sur laquelle se situe le Bomberman
life	Le nombre de point de vie du Bomberman
speed	La vitesse du Bomberman
bombMax	Le nombre de bombe maximal du Bomberman
nbBomb	Le nombre de bombe en possession du Bomberman

7.3.2.2 ~Bomberman()

```
Bomberman::~Bomberman ( )
```

Destructeur du Bomberman.

7.3.3 Documentation des fonctions membres

7.3.3.1 addBomb()

```
void Bomberman::addBomb (
            int nbBomb )
```

Permets d'ajouter des bombes au Bomberman.
Si le nombre de bombe est supérieur au nombre de bombe maximale alors mets le nombre de bombe au maximum

Paramètres

nbBomb	Le nombre de bombe a ajouté

Auteur

Pierre CHEMIN

7.3.3.2 addLife()

```
void Bomberman::addLife (
            int life )
```

Permets d'ajouter de la vie au Bomberman.

Paramètres

life	Le nombre de point de vie à ajouter au Bomberman

Auteur

Pierre CHEMIN

7.3.3.3 addSpeed()

```
void Bomberman::addSpeed (
            int speed )
```

Permets d'ajouter de la vitesse au Bomberman.

Paramètres

speed	La vitesse à ajouter au Bomberman

Auteur

Pierre CHEMIN

7.3.3.4 dropBomb()

```
void Bomberman::dropBomb ( )
```

Permets de retirer une bombe au Bomberman.

Auteur

Pierre CHEMIN

7.3.3.5 getBombMax()

```
int Bomberman::getBombMax ( ) const
```

Accesseur de l'attribut m_bombMax.

Renvoie

int Le nombre de bombe maximal du Bomberman

7.3.3.6 getNbBomb()

```
int Bomberman::getNbBomb ( ) const
```

Accesseur de l'attribut m_nbBomb.

```
@return int Le nombre de bombe maximal du Bomberman
```

7.3.3.7 move() [1/2]

```
bool Bomberman::move (
            utilities::EDirection direction )  [virtual]
```

Permets de déplacer le Bomberman dans une direction précise.

Voir également

Edirection

Paramètres

direction	La direction dans laquelle doit se déplacer le Bomberman

Renvoie

true si le déplacement a pu se faire, false sinon

Auteur

Pierre CHEMIN

Réimplémentée à partir de Personnage.

7.3.3.8 move() [2/2]

```
bool Bomberman::move (
            utilities::EDirection direction,
            int speed )
```

Permets de déplacer le Bomberman dans une direction et une vitesse précise.

Voir également

EDirection

Paramètres

direction	La direction dans laquelle doit se déplacer le Bomberman
speed	La vitesse avec laquelle le Bomberman doit se déplacer

Renvoie

true si le déplacement a pu se faire, false sinon

7.3.3.9 show()

```
void Bomberman::show ( ) const  [override], [virtual]
```

Permets d'afficher le Bomberman.

Affiche le Bomberman, le caractère "P"

Auteur

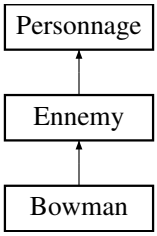
Pierre CHEMIN

Implémente Personnage.
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Persos/Bomberman.h
- src/src/Persos/Bomberman.cpp

7.4 Référence de la classe Bowman

La classe Bowman hérite de la classe Ennemy et permets de gérer les ennemies de type Bowman.
`#include <Bowman.h>`
Graphe d'héritage de Bowman:



Fonctions membres publiques

- Bowman (int x, int y, int life, int speed, int damage, int still)

  Constructeur d'un Bowman.
- ∼Bowman ()

  Destructeur d'un Bowman.
- virtual void show () const override

  Permets d'afficher le Bowman.
- virtual utilities::EDirection play (std::vector< std::vector< Tile ∗ >> map, Bomberman ∗player, std::vector< Item ∗ > ∗items) override

  Permets de faire jouer un Bowman.
- utilities::EDirection shootPlayer (Bomberman ∗player) const

  Permets de savoir si l'on peut lancer une flèche sur le joueur et dans quelle direction.

Membres hérités additionnels

7.4.1 Description détaillée

La classe Bowman hérite de la classe Ennemy et permets de gérer les ennemies de type Bowman.

Voir également

Ennemy

Permets de gérer un ennemie de type Bowman, il s'agit d'un ennemie pouvant lancer des flèches en direction du joueur

Auteur

Pierre CHEMIN

7.4.2 Documentation des constructeurs et destructeur

7.4.2.1 Bowman()

```
Bowman::Bowman (
            int x,
            int y,
            int life,
            int speed,
            int damage,
            int still )
```

Constructeur d'un Bowman.

Paramètres

x	La ligne sur laquelle se situe le Bowman
y	La colonne sur laquelle se situe le Bowman
life	Le nombre de point de vie du Bowman
speed	La vitesse du Bowman
damage	Les dagâts qu'inflige le Bowman
still	Le nombre de tour à patienter par le Bowman avant de jouer

7.4.2.2 ~Bowman()

```
Bowman::~Bowman ( )
```

Destructeur d'un Bowman.

7.4.3 Documentation des fonctions membres

7.4.3.1 play()

```
utilities::EDirection Bowman::play (
            std::vector< std::vector< Tile * >> map,
            Bomberman * player,
            std::vector< Item * > * items )  [override], [virtual]
```

Permets de faire jouer un Bowman.
Déplace un Bowman en fonction de la position du joueur sauf s'il peut tirer une flèche en direction du joueur et si c'est le cas le fait

Paramètres

map	La map de Tile sur laquelle le Bowman se déplace
player	Le joueur présent dans la partie
items	La liste des items de la partie

Renvoie

utilities::EDirection La direction dans laquelle se déplace le Bowman

Auteur

Pierre CHEMIN

Réimplémentée à partir de Ennemy.

7.4.3.2 shootPlayer()

```
utilities::EDirection Bowman::shootPlayer (
            Bomberman * player ) const
```

Permets de savoir si l'on peut lancer une flèche sur le joueur et dans quelle direction.

Paramètres

player	Le joueur qu'il faut viser

Renvoie

utilities::EDirection La direction dans laquelle se situe le joueur si on peut lui lancer une flèche

Auteur

Pierre CHEMIN

7.4.3.3 show()

```
void Bowman::show ( ) const  [override], [virtual]
```

Permets d'afficher le Bowman.
Affiche le Bowman, le caractère "B"

Auteur

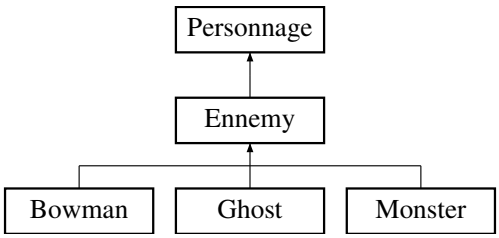
Pierre CHEMIN

Implémente Personnage.
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Persos/Bowman.h
- src/src/Persos/Bowman.cpp

7.5 Référence de la classe Ennemy

La classe abstraite Ennemy qui hérite de la classe Personnage.
`#include <Ennemy.h>`
Graphe d'héritage de Ennemy:



Fonctions membres publiques

- Ennemy (int x, int y, int life, int speed, int damage, int still)

  Constructeur d'un Ennemy.
- ~Ennemy ()

  Destructeur d'un Ennemy.
- virtual utilities::EDirection play (std::vector< std::vector< Tile * >> map, Bomberman *player, std::vector< Item * > *items)

  Permets de faire jouer un Ennemy.

Attributs protégés

- int m_damage
- int m_still

7.5.1 Description détaillée

La classe abstraite Ennemy qui hérite de la classe Personnage.

Voir également

Ennemy

Permets de définir un Ennemy

Auteur

Pierre CHEMIN

7.5.2 Documentation des constructeurs et destructeur

7.5.2.1 Ennemy()

```
Ennemy::Ennemy (
            int x,
            int y,
            int life,
            int speed,
            int damage,
            int still )
```

Constructeur d'un Ennemy.

Paramètres

x	La ligne sur laquelle se situe l'Ennemy
y	La colonne sur laquelle se situe l'Ennemy
life	Le nombre de point de vie de l'Ennemy
speed	La vitesse de l'Ennemy
damage	Les dégâts qu'inflie l'Ennemy
still	Le nombre de tour à patienter par le Bowman avant de jouer

7.5.2.2 ~Ennemy()

```
Ennemy::~Ennemy ( )
```

Destructeur d'un Ennemy.

7.5.3 Documentation des fonctions membres

7.5.3.1 play()

```
utilities::EDirection Ennemy::play (
            std::vector< std::vector< Tile * >> map,
            Bomberman * player,
            std::vector< Item * > * items )  [virtual]
```

Permets de faire jouer un Ennemy.
Déplace un Ennemy en fonction de la position du joueur et si la case est accessible

Paramètres

map	La map de Tile sur laquelle l'Ennemy se déplace
player	Le joueur présent dans la partie
items	La liste des items de la partie

Renvoie

utilities::EDirection La direction dans laquelle se déplace l'Ennemy

Réimplémentée dans Bowman, et Ghost.

7.5.4 Documentation des données membres

7.5.4.1 m_damage

`int Ennemy::m_damage [protected]`
Les dégâts infligés par un Ennemy

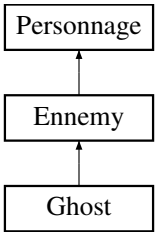
7.5.4.2 m_still

`int Ennemy::m_still [protected]`
Le nombre de tour durant lequel un Ennemy ne peut jouer
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Persos/Ennemy.h
- src/src/Persos/Ennemy.cpp

7.6 Référence de la classe Ghost

La classe Ghost hérite de la classe Ennemy et permets de gérer les ennemies de type Ghost.
`#include <Ghost.h>`
Graphe d'héritage de Ghost:



Fonctions membres publiques

- Ghost (int x, int y, int life, int speed, int damage, int still)

  Constructeur d'un Ghost.
- ~Ghost ()

  Destructeur d'un Ghost.
- virtual utilities::EDirection play (std::vector< std::vector< Tile * >> map, Bomberman *player, std::vector< Item * > *items) override

  Permets de faire jouer un Ghost.
- virtual void show () const override

  Permets d'afficher le Ghost.

Membres hérités additionnels

7.6.1 Description détaillée

La classe Ghost hérite de la classe Ennemy et permets de gérer les ennemies de type Ghost.

Voir également

Ennemy

Permets de gérer un ennemie de type Ghost, il s'agit d'un ennemie pouvant traverser toutes les cases même celle occupé

Auteur

Pierre CHEMIN

7.6.2 Documentation des constructeurs et destructeur

7.6.2.1 Ghost()

```
Ghost::Ghost (
            int x,
            int y,
            int life,
            int speed,
            int damage,
            int still )
```

Constructeur d'un Ghost.

Paramètres

x	La ligne sur laquelle se situe le Ghost
y	La colonne sur laquelle se situe le Ghost
life	Le nombre de point de vie de le Ghost
speed	La vitesse de le Ghost
damage	Les dégâts qu'inflie le Ghost
still	Le nombre de tour à patienter par le Ghost avant de jouer

7.6.2.2 ~Ghost()

```
Ghost::~Ghost ( )
```

Destructeur d'un Ghost.

7.6.3 Documentation des fonctions membres

7.6.3.1 play()

```
utilities::EDirection Ghost::play (
            std::vector< std::vector< Tile * >> map,
            Bomberman * player,
            std::vector< Item * > * items )  [override], [virtual]
```

Permets de faire jouer un Ghost.
Déplace un Ghost en fonction de la position du joueur

Paramètres

map	La map de Tile sur laquelle le Ghost se déplace
player	Le joueur présent dans la partie
items	La liste des items de la partie

Renvoie

utilities::EDirection La direction dans laquelle se déplace le Ghost

Auteur

Pierre CHEMIN

Réimplémentée à partir de Ennemy.

7.6.3.2 show()

```
void Ghost::show ( ) const  [override], [virtual]
```

Permets d'afficher le Ghost.
Affiche le Ghost, le caractère "G"

Auteur

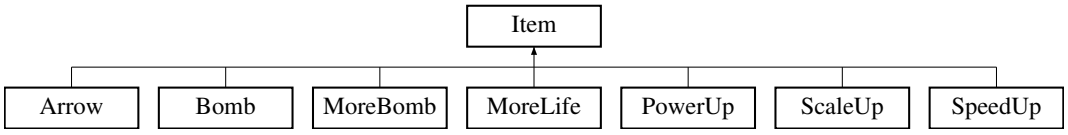
Pierre CHEMIN

Implémente Personnage.
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Persos/Ghost.h
- src/src/Persos/Ghost.cpp

7.7 Référence de la classe Item

La classe abstraite Item.
`#include <Item.h>`
Graphe d'héritage de Item:



Fonctions membres publiques

- Item (int x=0, int y=0)

  Constructeur de la classe Item.
- ~Item ()

  Destructeur de la classe Item.
- Position getPosition () const

  accesseur sur l'attribut position
- virtual void showTop () const =0

  Affiche la partie haute de la case lors de l'affichage de l'item.
- virtual void showMiddle () const =0

  Affiche la partie du milieu de la case lors de l'affichage de l'item.

- virtual void showBottom () const =0

  Affiche la partie basse de la case lors de l'affichage de l'item.
- virtual bool play (std::vector< std::vector< Tile * >> map, Bomberman *player, std::vector< Item * > *items)=0

  Permet de jouer un item.

Attributs protégés

- Position m_position

7.7.1 Description détaillée

La classe abstraite Item.

Voir également

Position

Permet de définir les différents items

Auteur

Hocine HADID

7.7.2 Documentation des constructeurs et destructeur

7.7.2.1 Item()

```
Item::Item (
            int x = 0,
            int y = 0 )
```

Constructeur de la classe Item.

Paramètres

x	La ligne sur laquelle se situe l'objet
y	La colonne sur laquelle se situe l'objet

7.7.2.2 ~Item()

```
Item::~Item ( )
```

Destructeur de la classe Item.

7.7.3 Documentation des fonctions membres

7.7.3.1 getPosition()

```
Position Item::getPosition ( ) const
```

accesseur sur l'attribut position

Renvoie

Position La position de l'item

7.7.3.2 play()

```
virtual bool Item::play (
            std::vector< std::vector< Tile * >> map,
            Bomberman * player,
            std::vector< Item * > * items )  [pure virtual]
```

Permet de jouer un item.

Voir également

Bomberman

Item

Tile

Paramètres

map	La map sur laquelle l'item se situe
player	Le joueur présent sur la map
items	La liste des items présents sur la map

Renvoie

true si l'item doit être supprimé, false sinon

Auteur

Hocine HADID

Implémenté dans Bomb, Arrow, MoreBomb, MoreLife, SpeedUp, PowerUp, et ScaleUp.

7.7.3.3 showBottom()

```
virtual void Item::showBottom ( ) const  [pure virtual]
```

Affiche la partie basse de la case lors de l'affichage de l'item.

Auteur

Hocine HADID

Implémenté dans Bomb, Arrow, MoreBomb, MoreLife, SpeedUp, PowerUp, et ScaleUp.

7.7.3.4 showMiddle()

```
virtual void Item::showMiddle ( ) const  [pure virtual]
```

Affiche la partie du milieu de la case lors de l'affichage de l'item.

Auteur

Hocine HADID

Implémenté dans Bomb, Arrow, MoreBomb, MoreLife, SpeedUp, PowerUp, et ScaleUp.

7.7.3.5 showTop()

```
virtual void Item::showTop ( ) const  [pure virtual]
```

Affiche la partie haute de la case lors de l'affichage de l'item.

Auteur

Hocine HADID

Implémenté dans Bomb, Arrow, MoreBomb, MoreLife, SpeedUp, PowerUp, et ScaleUp.

7.7.4 Documentation des données membres

7.7.4.1 m_position

`Position Item::m_position [protected]`
La position de l'item
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Items/Item.h
- src/src/Items/Item.cpp

7.8 Référence de la classe Map

La classe Map permets de gérer une map du jeu Bomberman.
`#include <Map.h>`

Fonctions membres publiques

- Map (int level=0)

  Constructeur d'une Map.
- ~Map ()

  Destructeur d'une Map.
- Tile getTarget () const

  Accesseur de l'attribut m_target.
- int getNbColumn () const

  Accesseur de l'attribut m_nbColumn.
- int getNbLine () const

  Accesseur de l'attribut m_nbLine.
- Bomberman getPlayer () const

  Accesseur de l'attribut m_player.
- std::vector< Ennemy * > getListEnnemy () const

  Accesseur de l'attribut m_listEnnemy.
- std::vector< Item * > getListItems () const

  Accesseur de l'attribut m_listItems.
- std::vector< Bomb * > getListBombs () const

  Accesseur de l'attribut m_listBombs.
- void playBomb ()

  Permets de rajouter une bombe sur la map à la position du joueur.
- void eraseBombExplosion ()

  Permets de remettre à 0 la map de l'explosion d'une bombe.
- void showMap () const

  Permet l'affichage de la map sur la console.
- void loadMap (int map)

  Permets de charger une map en fonction de son niveau.
- bool movePlayer (utilities::EDirection direction, int nbCase)

  Permets de déplacer le Bomberman dans une direction et un nombre de case précis.
- void playEnnemy (int ennemy)

  Permets de jouer un ennemie précis sur la map.
- void playItem (int item)

  Permets de jouer un item précis sur la map.
- void playBomb (int bomb)

  Permets de jouer une bombe présente sur la map.

7.8.1 Description détaillée

La classe Map permets de gérer une map du jeu Bomberman.

Voir également

Bomberman

Tile

Ennemy

Item

Bomb

EBombExplosionDirection

Auteur

Pierre CHEMIN & Hocine HADID

7.8.2 Documentation des constructeurs et destructeur

7.8.2.1 Map()

```
Map::Map (
            int level = 0 )
```

Constructeur d'une Map.

Paramètres

level	Le niveau de la Map à charger

7.8.2.2 ~Map()

```
Map::~Map ( )
```

Destructeur d'une Map.

7.8.3 Documentation des fonctions membres

7.8.3.1 eraseBombExplosion()

```
void Map::eraseBombExplosion ( )
```

Permets de remettre à 0 la map de l'explosion d'une bombe.

Auteur

Hocine HADID

7.8.3.2 getListBombs()

```
std::vector< Bomb * > Map::getListBombs ( ) const
```

Accesseur de l'attribut m_listBombs.

Renvoie

std::vector<Bomb*> La liste des bombes présentes sur la Map

7.8.3.3 getListEnnemy()

`std::vector< Ennemy * > Map::getListEnnemy ( ) const`
Accesseur de l'attribut m_listEnnemy.

Renvoie

std::vector<Ennemy*> La liste des ennemies de la Map

7.8.3.4 getListItems()

`std::vector< Item * > Map::getListItems ( ) const`
Accesseur de l'attribut m_listItems.

Renvoie

std::vector<Item*> La liste des items de la Map

7.8.3.5 getNbColumn()

`int Map::getNbColumn ( ) const`
Accesseur de l'attribut m_nbColumn.

Renvoie

int Le nombre de colonne de la Map

7.8.3.6 getNbLine()

`int Map::getNbLine ( ) const`
Accesseur de l'attribut m_nbLine.

Renvoie

int Le nombre de ligne de la Map

7.8.3.7 getPlayer()

`Bomberman Map::getPlayer ( ) const`
Accesseur de l'attribut m_player.

Renvoie

Bomberman Le Bomberman (joueur) de la Map

7.8.3.8 getTarget()

`Tile Map::getTarget ( ) const`
Accesseur de l'attribut m_target.

Renvoie

Tile L'objectif de la map

7.8.3.9 loadMap()

```
void Map::loadMap (
            int map )
```

Permets de charger une map en fonction de son niveau.

Paramètres

map	Le niveau de la map à charger

Auteur

Pierre CHEMIN

7.8.3.10 movePlayer()

```
bool Map::movePlayer (
            utilities::EDirection direction,
            int nbCase )
```

Permets de déplacer le Bomberman dans une direction et un nombre de case précis.

Exceptions

MoveException	lève une exception si le déplacement ne peut être effectué pour quelconques raisons

Paramètres

direction	La direction dans laquelle doit se déplacer le Bomberman
nbCase	Le nombre de case dont doit se déplacer le Bomberman

Renvoie

true si le déplacement a bien été effectué, false sinon

Auteur

Pierre CHEMIN

7.8.3.11 playBomb() [1/2]

```
void Map::playBomb ( )
```

Permets de rajouter une bombe sur la map à la position du joueur.

Auteur

Pierre CHEMIN

7.8.3.12 playBomb() [2/2]

```
void Map::playBomb (
            int bomb )
```

Permets de jouer une bombe présente sur la map.

Paramètres

bomb	La bombe à jouer sur la map

Auteur

Hocine HADID

7.8.3.13 playEnnemy()

```
void Map::playEnnemy (
            int ennemy )
```

Permets de jouer un ennemie précis sur la map.

Paramètres

ennemy	L'ennemie à faire joueur sur la map

Auteur

Hocine HADID

7.8.3.14 playItem()

```
void Map::playItem (
            int item )
```

Permets de jouer un item précis sur la map.

Paramètres

item	L'item à jouer sur la map

Auteur

Hocine HADID

7.8.3.15 showMap()

```
void Map::showMap ( ) const
```

Permet l'affichage de la map sur la console.

Auteur

Pierre CHEMIN

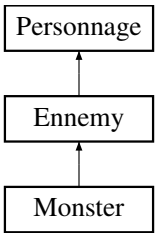
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Map/Map.h
- src/src/Map/Map.cpp

7.9 Référence de la classe Monster

La classe Monster hérite de la classe Ennemy et permets de gérer les ennemies de type Monster.

`#include <Monster.h>`
Graphe d'héritage de Monster:



Fonctions membres publiques

- Monster (int x, int y, int life, int speed, int damage, int still)

  Constructeur d'un Monster.
- ~Monster ()

  Destructeur d'un Monster.
- virtual void show () const override

  Permets d'afficher le Monster.

Membres hérités additionnels

7.9.1 Description détaillée

La classe Monster hérite de la classe Ennemy et permets de gérer les ennemies de type Monster.

Voir également

Ennemy

Permets de gérer un ennemie de type Monster, il s'agit d'un ennemie basique

Auteur

Pierre CHEMIN

7.9.2 Documentation des constructeurs et destructeur

7.9.2.1 Monster()

```
Monster::Monster (
            int x,
            int y,
            int life,
            int speed,
            int damage,
            int still )
```

Constructeur d'un Monster.

Paramètres

x	La ligne sur laquelle se situe le Monster
y	La colonne sur laquelle se situe le Monster
life	Le nombre de point de vie de le Monster
speed	La vitesse de le Monster
damage	Les dégâts qu'inflie le Monster
still	Le nombre de tour à patienter par le Monster avant de jouer

7.9.2.2 ~Monster()

`Monster::~Monster ( )`
Destructeur d'un Monster.

7.9.3 Documentation des fonctions membres

7.9.3.1 show()

`void Monster::show ( ) const [override], [virtual]`
Permets d'afficher le Monster.
Affiche le Monster, le caractère "M"

Auteur

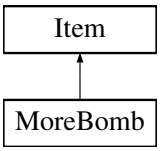
Pierre CHEMIN

Implémente Personnage.
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Persos/Monster.h
- src/src/Persos/Monster.cpp

7.10 Référence de la classe MoreBomb

La classe MoreBomb hérite de la classe Item et permets de gérer les items de type MoreBomb.
`#include <MoreBomb.h>`
Graphe d'héritage de MoreBomb:



Fonctions membres publiques

- MoreBomb (int x=0, int y=0, int nbBomb=1)

  Constructeur d'un item MoreBomb.
- ~MoreBomb ()

  Destructeur de la MoreBomb.
- void increaseNbBomb (Bomberman *b)

  Rajoute le nombre de Bombe au Bomberman.
- void showTop () const override

  Affiche la partie haute de la case lors de l'affichage de la MoreBomb.
- void showMiddle () const override

  Affiche la partie du milieu de la case lors de l'affichage de la MoreBomb.
- void showBottom () const override

  Affiche la partie basse de la case lors de l'affichage de la MoreBomb.
- virtual bool play (std::vector< std::vector< Tile * >> map, Bomberman *player, std::vector< Item * > *items) override

  Permet de jouer la MoreBomb.

Membres hérités additionnels

7.10.1 Description détaillée

La classe MoreBomb hérite de la classe Item et permets de gérer les items de type MoreBomb.

Voir également

Item

Permets de gérer les items de type MoreBomb qui permettent d'ajouter des bombes au Bomberman

Auteur

Hocine HADID

7.10.2 Documentation des constructeurs et destructeur

7.10.2.1 MoreBomb()

```
MoreBomb::MoreBomb (
            int x = 0,
            int y = 0,
            int nbBomb = 1 )
```

Constructeur d'un item MoreBomb.

Paramètres

x	La ligne sur laquelle se situe la MoreBomb
y	La colonne sur laquelle se situe la MoreBomb
nbBomb	Le nombre de bombe que rajoutera l'item

7.10.2.2 ~MoreBomb()

```
MoreBomb::~MoreBomb ( )
```

Destructeur de la MoreBomb.

7.10.3 Documentation des fonctions membres

7.10.3.1 increaseNbBomb()

```
void MoreBomb::increaseNbBomb (
            Bomberman * b )
```

Rajoute le nombre de Bombe au Bomberman.

Voir également

Bomberman

Paramètres

b	Le bomberman auquelle il faut rajouter le nombre de bombe

Auteur

Hocine HADID

7.10.3.2 play()

```
bool MoreBomb::play (
            std::vector< std::vector< Tile * >> map,
            Bomberman * player,
            std::vector< Item * > * items )  [override], [virtual]
```

Permet de jouer la MoreBomb.

Voir également

Bomberman

Item

Tile

Joue la MoreBomb, si la MoreBomb est sur la même case que le joueur alors rajoute le nombre de bomb indiqué au joueur.

Paramètres

map	La map sur laquelle la MoreBomb est joué
player	Le joueur présent sur la map
items	La liste des items présents sur la map

Renvoie

true si la MoreBomb doit être supprimé, false sinon

Auteur

Hocine HADID

Implémente Item.

7.10.3.3 showBottom()

```
void MoreBomb::showBottom ( ) const  [override], [virtual]
```

Affiche la partie basse de la case lors de l'affichage de la MoreBomb.

Auteur

Hocine HADID

Implémente Item.

7.10.3.4 showMiddle()

```
void MoreBomb::showMiddle ( ) const  [override], [virtual]
```

Affiche la partie du milieu de la case lors de l'affichage de la MoreBomb.

Auteur

Hocine HADID

Implémente Item.

7.10.3.5 showTop()

`void MoreBomb::showTop ( ) const [override], [virtual]`
Affiche la partie haute de la case lors de l'affichage de la MoreBomb.

Auteur

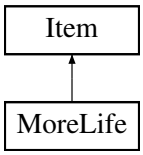
Hocine HADID

Implémente Item.
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Items/MoreBomb.h
- src/src/Items/MoreBomb.cpp

7.11 Référence de la classe MoreLife

La classe MoreLife hérite de la classe Item et permets de gérer les items de type MoreLife.
`#include <MoreLife.h>`
Graphe d'héritage de MoreLife:



Fonctions membres publiques

- MoreLife (int x=0, int y=0, int lifePoint=1)

  Constructeur d'un item MoreLife.
- ~MoreLife ()

  Destructeur de la MoreLife.
- void increaseLife (Bomberman *b)

  Rajoute le nombre de point de vie au Bomberman.
- void showTop () const override

  Affiche la partie haute de la case lors de l'affichage de la MoreLife.
- void showMiddle () const override

  Affiche la partie du milieu de la case lors de l'affichage de la MoreLife.
- void showBottom () const override

  Affiche la partie basse de la case lors de l'affichage de la MoreLife.
- virtual bool play (std::vector< std::vector< Tile * >> map, Bomberman *player, std::vector< Item * > *items) override

  Permet de jouer la MoreLife.

Membres hérités additionnels

7.11.1 Description détaillée

La classe MoreLife hérite de la classe Item et permets de gérer les items de type MoreLife.

Voir également

Item

Permets de gérer les items de type MoreLife qui permettent d'ajouter de la vie au Bomberman

Auteur

Hocine HADID

7.11.2 Documentation des constructeurs et destructeur

7.11.2.1 MoreLife()

```
MoreLife::MoreLife (
            int x = 0,
            int y = 0,
            int lifePoint = 1 )
```

Constructeur d'un item MoreLife.

Paramètres

x	La ligne sur laquelle se situe la MoreLife
y	La colonne sur laquelle se situe la MoreLife
lifePoint	Le nombre de point de vie que rajoutera l'item

7.11.2.2 ~MoreLife()

```
MoreLife::~MoreLife ( )
```

Destructeur de la MoreLife.

7.11.3 Documentation des fonctions membres

7.11.3.1 increaseLife()

```
void MoreLife::increaseLife (
            Bomberman * b )
```

Rajoute le nombre de point de vie au Bomberman.

Voir également

Bomberman

Paramètres

b	Le bomberman auquelle il faut rajouter le nombre de point de vie

Auteur

Hocine HADID

7.11.3.2 play()

```
bool MoreLife::play (
            std::vector< std::vector< Tile * >> map,
            Bomberman * player,
            std::vector< Item * > * items )  [override], [virtual]
```

Permet de jouer la MoreLife.

Voir également

Bomberman

Item

Tile

Joue la MoreLife, si la MoreLife est sur la même case que le joueur alors rajoute le nombre de point de vie indiqué au joueur.

Paramètres

map	La map sur laquelle la MoreLife est joué
player	Le joueur présent sur la map
items	La liste des items présents sur la map

Renvoie

true si la MoreLife doit être supprimé, false sinon

Auteur

Hocine HADID

Implémente Item.

7.11.3.3 showBottom()

`void MoreLife::showBottom ( ) const [override], [virtual]`
Affiche la partie basse de la case lors de l'affichage de la MoreLife.

Auteur

Hocine HADID

Implémente Item.

7.11.3.4 showMiddle()

`void MoreLife::showMiddle ( ) const [override], [virtual]`
Affiche la partie du milieu de la case lors de l'affichage de la MoreLife.

Auteur

Hocine HADID

Implémente Item.

7.11.3.5 showTop()

`void MoreLife::showTop ( ) const [override], [virtual]`
Affiche la partie haute de la case lors de l'affichage de la MoreLife.

Auteur

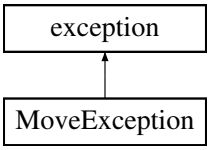
Hocine HADID

Implémente Item.
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Items/MoreLife.h
- src/src/Items/MoreLife.cpp

7.12 Référence de la classe MoveException

La classe MoveException hérite de la classe std::exception.
`#include <MoveException.h>`
Graphe d'héritage de MoveException:



Fonctions membres publiques

- MoveException (std::string message)

  Constructeur de la MoveException.
- virtual const char * what () const throw ()

  Permets de savoir pourquoi l'exception est levé

7.12.1 Description détaillée

La classe MoveException hérite de la classe std::exception.
Permets de gérer les exceptions de déplacement

Auteur

Pierre CHEMIN

7.12.2 Documentation des constructeurs et destructeur

7.12.2.1 MoveException()

```
MoveException::MoveException (
            std::string message )  [inline]
```

Constructeur de la MoveException.

Paramètres

message	Le message à afficher par l'exception

7.12.3 Documentation des fonctions membres

7.12.3.1 what()

```
virtual const char* MoveException::what ( ) const throw ( )   [inline], [virtual]
```

Permets de savoir pourquoi l'exception est levé

Renvoie

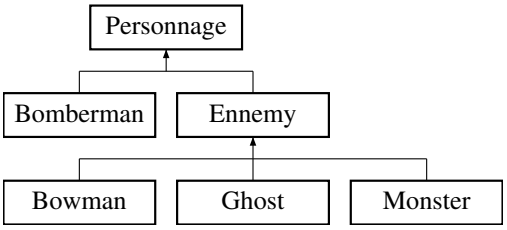
char* Le message dec l'exception à afficher à l'utilisateur

La documentation de cette classe a été générée à partir du fichier suivant :

- src/include/Map/MoveException.h

7.13 Référence de la classe Personnage

La classe abstraite personnage.
`#include <Personnage.h>`
Graphe d'héritage de Personnage:



Fonctions membres publiques

- Personnage (int x=0, int y=0, int life=3, int speed=1)

  Constructeur d'un Personnage.
- ~Personnage ()

  Destructeur d'un Personnage.
- Position getPosition () const

  Accesseur de l'attribut m_position.
- int getLife () const

  Accesseur de l'attribut m_life.
- int getSpeed () const

  Accesseur de l'attribut m_speed.
- void receiveDamage (int damage)

  Lorsque le personnage reçoit des dégâts.
- virtual bool move (utilities::EDirection direction)

  Permets de déplacer un personnage.
- virtual void show () const =0

  Permets d'afficher le personnage.

Attributs protégés

- Position m_position
- int m_life
- int m_speed

7.13.1 Description détaillée

La classe abstraite personnage.

Voir également

Position

Permets de définir un personnage

Auteur

Pierre CHEMIN

7.13.2 Documentation des constructeurs et destructeur

7.13.2.1 Personnage()

```
Personnage::Personnage (
            int x = 0,
            int y = 0,
            int life = 3,
            int speed = 1 )
```

Constructeur d'un Personnage.

Paramètres

x	La ligne sur laquelle se situe le Personnage
y	La colonne sur laquelle se situe le Personnage
life	Le nombre de point de vie du Personnage
speed	La vitesse du Personnage

7.13.2.2 ~Personnage()

```
Personnage::~Personnage ( )
```

Destructeur d'un Personnage.

7.13.3 Documentation des fonctions membres

7.13.3.1 getLife()

```
int Personnage::getLife ( ) const
```

Accesseur de l'attribut m_life.

Renvoie

int Le nombre de point de vie restant du Personnage

7.13.3.2 getPosition()

```
Position Personnage::getPosition ( ) const
```

Accesseur de l'attribut m_position.

Renvoie

Position La position du Personnage

7.13.3.3 getSpeed()

```
int Personnage::getSpeed ( ) const
```

Accesseur de l'attribut m_speed.

Renvoie

int La vitesse du Personnage

7.13.3.4 move()

```
virtual bool Personnage::move (
            utilities::EDirection direction )  [virtual]
```

Permets de déplacer un personnage.

Voir également

EDirection

Paramètres

direction	La direction dans laquelle doit se déplacer le personnage

Renvoie

true si le déplacement a pu se faire, false sinon

Auteur

Pierre CHEMIN

Réimplémentée dans Bomberman.

7.13.3.5 receiveDamage()

```
void Personnage::receiveDamage (
            int damage )
```

Lorsque le personnage reçoit des dégâts.
Permets d'infliger des dégâts au personnage

Paramètres

damage	Le nombre de dégât infligé

Auteur

Pierre CHEMIN

7.13.3.6 show()

```
virtual void Personnage::show ( ) const  [pure virtual]
```

Permets d'afficher le personnage.

Auteur

Pierre CHEMIN

Implémenté dans Bomberman, Bowman, Ghost, et Monster.

7.13.4 Documentation des données membres

7.13.4.1 m_life

```
int Personnage::m_life  [protected]
```

La vie du personnage

7.13.4.2 m_position

```
Position Personnage::m_position  [protected]
```
La position du personnage

7.13.4.3 m_speed

```
int Personnage::m_speed  [protected]
```
La vitesse du personnage
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Persos/Personnage.h
- src/src/Persos/Personnage.cpp

7.14 Référence de la classe Position

La classe Position permets de définir une position.
`#include <Position.h>`

Fonctions membres publiques

- Position (int x=0, int y=0)

  Constructeur d'un Position.
- int getX () const

  Accesseur de l'attribut m_x.
- int getY () const

  Accesseur de l'attribut m_y.
- void setX (int x)

  Accesseur de l'attribut m_x.
- void setY (int y)

  Accesseur de l'attribut m_y.
- bool operator== (const Position &other)

  Surcharge de l'opérateur de comparaison, il permet de vérifier si deux Positions sont égales ou non.
- bool operator!= (const Position &other)

  Surcharge de l'opérateur de différence, il permet de vérifier si deux Positions sont différentes ou non.

7.14.1 Description détaillée

La classe Position permets de définir une position.
Permets de définir une position sur un axe X et Y

Auteur

Hocine HADID

7.14.2 Documentation des constructeurs et destructeur

7.14.2.1 Position()

```
Position::Position (
            int x = 0,
            int y = 0 )
```
Constructeur d'un Position.

Paramètres

x	L'axe X (ou la ligne) de la Position
y	L'axe Y (ou la colonne) de la Position

7.14.3 Documentation des fonctions membres

7.14.3.1 getX()

```
int Position::getX ( ) const
```

Accesseur de l'attribut m_x.

Renvoie

int L'axe X (ou la ligne) de la Position

7.14.3.2 getY()

```
int Position::getY ( ) const
```

Accesseur de l'attribut m_y.

Renvoie

int L'axe Y (ou la colonne) de la Position

7.14.3.3 operator"!=()

```
bool Position::operator!= (
            const Position & other )
```

Surcharge de l'opérateur de différence, il permet de vérifier si deux Positions sont différentes ou non.

Paramètres

other	La position qu'il faut comparé avec la position courante

Renvoie

true si les positions comparées sont différentes, false sinon

Auteur

Hocine HADID

7.14.3.4 operator==()

```
bool Position::operator== (
            const Position & other )
```

Surcharge de l'opérateur de comparaison, il permet de vérifier si deux Positions sont égales ou non.

Paramètres

other	La position qu'il faut comparé avec la position courante

Renvoie

true si les positions comparées sont égales, false sinon

Auteur

Hocine HADID

7.14.3.5 setX()

```
void Position::setX (
            int x )
```

Accesseur de l'attribut m_x.

Paramètres

x	La nouvelle valeur de m_x

7.14.3.6 setY()

```
void Position::setY (
            int y )
```

Accesseur de l'attribut m_y.

Paramètres

y	La nouvelle valeur de m_y

La documentation de cette classe a été générée à partir des fichiers suivants :

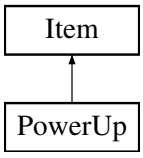
- src/include/Map/Position.h
- src/src/Map/Position.cpp

7.15 Référence de la classe PowerUp

La classe PowerUp hérite de la classe Item et permets de gérer les items de type PowerUp.

`#include <PowerUp.h>`

Graphe d'héritage de PowerUp:



Fonctions membres publiques

- PowerUp (int x=0, int y=0, int powerPoint=1)

  Constructeur d'un item PowerUp.
- ~PowerUp ()

  Destructeur de la PowerUp.

- void increasePower ()

  Rajoute la puissance précisé aux bombes.
- void showTop () const override

  Affiche la partie haute de la case lors de l'affichage de la PowerUp.
- void showMiddle () const override

  Affiche la partie du milieu de la case lors de l'affichage de la PowerUp.
- void showBottom () const override

  Affiche la partie basse de la case lors de l'affichage de la PowerUp.
- virtual bool play (std::vector< std::vector< Tile * >> map, Bomberman *player, std::vector< Item * > *items) override

  Permet de jouer la PowerUp.

Membres hérités additionnels

7.15.1 Description détaillée

La classe PowerUp hérite de la classe Item et permets de gérer les items de type PowerUp.

Voir également

Item

Permets de gérer les items de type MoreLife qui permettent d'augmenter la puissance des bombes

Auteur

Hocine HADID

7.15.2 Documentation des constructeurs et destructeur

7.15.2.1 PowerUp()

```
PowerUp::PowerUp (
            int x = 0,
            int y = 0,
            int powerPoint = 1 )
```

Constructeur d'un item PowerUp.

Paramètres

x	La ligne sur laquelle se situe la PowerUp
y	La colonne sur laquelle se situe la PowerUp
powerPoint	La puissance que rajoutera l'item aux bombes

7.15.2.2 ~PowerUp()

```
PowerUp::~PowerUp ( )
```

Destructeur de la PowerUp.

7.15.3 Documentation des fonctions membres

7.15.3.1 increasePower()

```
void PowerUp::increasePower ( )
```
Rajoute la puissance précisé aux bombes.

Auteur

Hocine HADID

7.15.3.2 play()

```
bool PowerUp::play (
            std::vector< std::vector< Tile * >> map,
            Bomberman * player,
            std::vector< Item * > * items )  [override], [virtual]
```
Permet de jouer la PowerUp.

Voir également

Bomberman

Item

Tile

Joue la PowerUp, si la PowerUp est sur la même case que le joueur alors rajoute la puissance précisé aux bombes

Paramètres

map	La map sur laquelle la PowerUp est joué
player	Le joueur présent sur la map
items	La liste des items présents sur la map

Renvoie

true si la PowerUp doit être supprimé, false sinon

Auteur

Hocine HADID

Implémente Item.

7.15.3.3 showBottom()

```
void PowerUp::showBottom ( ) const  [override], [virtual]
```
Affiche la partie basse de la case lors de l'affichage de la PowerUp.

Auteur

Hocine HADID

Implémente Item.

7.15.3.4 showMiddle()

```
void PowerUp::showMiddle ( ) const  [override], [virtual]
```
Affiche la partie du milieu de la case lors de l'affichage de la PowerUp.

Auteur

Hocine HADID

Implémente Item.

7.15.3.5 showTop()

`void PowerUp::showTop ( ) const [override], [virtual]`
Affiche la partie haute de la case lors de l'affichage de la PowerUp.

Auteur

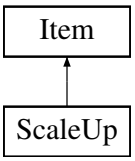
Hocine HADID

Implémente Item.
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Items/PowerUp.h
- src/src/Items/PowerUp.cpp

7.16 Référence de la classe ScaleUp

La classe ScaleUp hérite de la classe Item et permets de gérer les items de type ScaleUp.
`#include <ScaleUp.h>`
Graphe d'héritage de ScaleUp:



Fonctions membres publiques

- ScaleUp (int x=0, int y=0, double scope=0.5)

  Constructeur d'un item ScaleUp.
- ~ScaleUp ()

  Destructeur de la ScaleUp.
- void increaseScope ()

  Rajoute la porté précisé aux bombes.
- void showTop () const override

  Affiche la partie haute de la case lors de l'affichage de la ScaleUp.
- void showMiddle () const override

  Affiche la partie du milieu de la case lors de l'affichage de la ScaleUp.
- void showBottom () const override

  Affiche la partie basse de la case lors de l'affichage de la ScaleUp.
- virtual bool play (std::vector< std::vector< Tile * >> map, Bomberman *player, std::vector< Item * > *items) override

  Permet de jouer la ScaleUp.

Membres hérités additionnels

7.16.1 Description détaillée

La classe ScaleUp hérite de la classe Item et permets de gérer les items de type ScaleUp.

Voir également

Item

Permets de gérer les items de type ScaleUp qui permettent d'augmenter la portée des bombes

Auteur

Hocine HADID

7.16.2 Documentation des constructeurs et destructeur

7.16.2.1 ScaleUp()

```
ScaleUp::ScaleUp (
            int x = 0,
            int y = 0,
            double scope = 0.5 )
```

Constructeur d'un item ScaleUp.

Paramètres

x	La ligne sur laquelle se situe la ScaleUp
y	La colonne sur laquelle se situe la ScaleUp
scope	La porté qui sera rajouté à la porté des bombes

7.16.2.2 ~ScaleUp()

```
ScaleUp::~ScaleUp ( )
```

Destructeur de la ScaleUp.

7.16.3 Documentation des fonctions membres

7.16.3.1 increaseScope()

```
void ScaleUp::increaseScope ( )
```

Rajoute la porté précisé aux bombes.

Auteur

Hocine HADID

7.16.3.2 play()

```
bool ScaleUp::play (
            std::vector< std::vector< Tile * >> map,
            Bomberman * player,
            std::vector< Item * > * items )  [override], [virtual]
```

Permet de jouer la ScaleUp.

Voir également

Bomberman

Item

Tile

Joue la ScaleUp, si la ScaleUp est sur la même case que le joueur alors rajoute la porté précisé aux bombes

Paramètres

map	La map sur laquelle la ScaleUp est joué
player	Le joueur présent sur la map
items	La liste des items présents sur la map

Renvoie

true si la ScaleUp doit être supprimé, false sinon

Auteur

Hocine HADID

Implémente Item.

7.16.3.3 showBottom()

`void ScaleUp::showBottom ( ) const [override], [virtual]`
Affiche la partie basse de la case lors de l'affichage de la ScaleUp.

Auteur

Hocine HADID

Implémente Item.

7.16.3.4 showMiddle()

`void ScaleUp::showMiddle ( ) const [override], [virtual]`
Affiche la partie du milieu de la case lors de l'affichage de la ScaleUp.

Auteur

Hocine HADID

Implémente Item.

7.16.3.5 showTop()

`void ScaleUp::showTop ( ) const [override], [virtual]`
Affiche la partie haute de la case lors de l'affichage de la ScaleUp.

Auteur

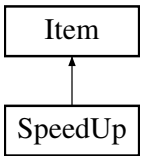
Hocine HADID

Implémente Item.
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Items/ScaleUp.h
- src/src/Items/ScaleUp.cpp

7.17 Référence de la classe SpeedUp

La classe SpeedUp hérite de la classe Item et permets de gérer les items de type SpeedUp.
`#include <SpeedUp.h>`
Graphe d'héritage de SpeedUp:



Fonctions membres publiques

- SpeedUp (int x=0, int y=0, int speed=1)

  Constructeur de la SpeedUp.
- ~SpeedUp ()

  Destructeur de la SpeedUp.
- void increaseSpeed (Bomberman *b)

  Rajoute la vitesse au Bomberman.
- void showTop () const override

  Affiche la partie haute de la case lors de l'affichage de la SpeedUp.
- void showMiddle () const override

  Affiche la partie du milieu de la case lors de l'affichage de la SpeedUp.
- void showBottom () const override

  Affiche la partie basse de la case lors de l'affichage de la Speedup.
- virtual bool play (std::vector< std::vector< Tile * >> map, Bomberman *player, std::vector< Item * > *items) override

  Permet de jouer la Speedup.

Membres hérités additionnels

7.17.1 Description détaillée

La classe SpeedUp hérite de la classe Item et permets de gérer les items de type SpeedUp.

Voir également

Item

Permets de gérer les items de type SpeedUp qui permettent d'augmenter la vitesse du Bomberman

Auteur

Hocine HADID

7.17.2 Documentation des constructeurs et destructeur

7.17.2.1 SpeedUp()

```
SpeedUp::SpeedUp (
            int x = 0,
            int y = 0,
            int speed = 1 )
```

Constructeur de la SpeedUp.

Paramètres

x	La ligne sur laquelle se situe la ScaleUp
y	La colonne sur laquelle se situe la ScaleUp
speed	La vitesse qui sera rajouté au joueur

7.17.2.2 ~SpeedUp()

```
SpeedUp::~SpeedUp ( )
```

Destructeur de la SpeedUp.

7.17.3 Documentation des fonctions membres

7.17.3.1 increaseSpeed()

```
void SpeedUp::increaseSpeed (
            Bomberman * b )
```

Rajoute la vitesse au Bomberman.

Voir également

Bomberman

Paramètres

b	Le bomberman auquelle il faut rajouter la vitesse

Auteur

Hocine HADID

7.17.3.2 play()

```
bool SpeedUp::play (
            std::vector< std::vector< Tile * >> map,
            Bomberman * player,
            std::vector< Item * > * items )  [override], [virtual]
```

Permet de jouer la Speedup.

Voir également

Bomberman

Item

Tile

Joue la Speedup, si la Speedup est sur la même case que le joueur alors rajoute la vitesse précisé au joueur

Paramètres

map	La map sur laquelle la Speedup est joué
player	Le joueur présent sur la map
items	La liste des items présents sur la map

Renvoie

true si la Speedup doit être supprimé, false sinon

Auteur

Hocine HADID

Implémente Item.

7.17.3.3 showBottom()

`void SpeedUp::showBottom ( ) const [override], [virtual]`
Affiche la partie basse de la case lors de l'affichage de la Speedup.

Auteur

Hocine HADID

Implémente Item.

7.17.3.4 showMiddle()

`void SpeedUp::showMiddle ( ) const [override], [virtual]`
Affiche la partie du milieu de la case lors de l'affichage de la SpeedUp.

Auteur

Hocine HADID

Implémente Item.

7.17.3.5 showTop()

`void SpeedUp::showTop ( ) const [override], [virtual]`
Affiche la partie haute de la case lors de l'affichage de la SpeedUp.

Auteur

Hocine HADID

Implémente Item.
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Items/SpeedUp.h
- src/src/Items/SpeedUp.cpp

7.18 Référence de la classe SystemGame

Classe permettant de gérer une partie.
`#include <SystemGame.h>`

Fonctions membres publiques

- SystemGame (int level=1)
  Constructeur de la classe SystemGame.
- void showMap ()
  Affiche la map et les informations du joueur.
- void playTurn ()
  Joue un tour de jeu.
- bool getEndGame ()
  accesseur de l'attribut endGame

7.18.1 Description détaillée

Classe permettant de gérer une partie.
C'est depuis cette classe que l'on gère une partie de Bomberman, en gérant le tour du joueur, des ennemies, des items et des bombes.

7.18.2 Documentation des constructeurs et destructeur

7.18.2.1 SystemGame()

```
SystemGame::SystemGame (
            int level = 1 )
```

Constructeur de la classe SystemGame.

Paramètres

level	La map à charger

7.18.3 Documentation des fonctions membres

7.18.3.1 getEndGame()

```
bool SystemGame::getEndGame ( )
```

accesseur de l'attribut endGame

Renvoie

vrai si la partie est terminé, faux sinon

7.18.3.2 playTurn()

```
void SystemGame::playTurn ( )
```

Joue un tour de jeu.
Permet de jouer un tour en faisant jouer le joueur puis les items, les ennemies et enfin les bombes pour au final afficher la Map. A la fin du tour vérifie si le joueur a gagné ou perdu.

Auteur

Hocine HADID

7.18.3.3 showMap()

```
void SystemGame::showMap ( )
```

Affiche la map et les informations du joueur.

Voir également

Map

Permet d'afficher la Map avec les informations du joueur : les points de vie et le nombre de bombes restantes. Fais appel à la méthode showMap de la classe Map

Auteur

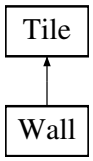
Hocine HADID

La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/engine/SystemGame.h
- src/src/engine/SystemGame.cpp

7.19 Référence de la classe Tile

La classe Tile permets de définir une case de la map.
`#include <Tile.h>`
Graphe d'héritage de Tile:



Fonctions membres publiques

- Tile (int x=0, int y=0, bool beCrossed=true)

  Constructeur d'une Tile.
- ∼Tile ()

  Destructeur d'une Tile.
- Position getPosition () const

  Accesseur de l'attribut m_position.
- void setBeCrossed (bool beCrossed)

  Accesseur de l'attribut m_beCrossed.
- bool getBeCrossed () const

  Accesseur de l'attribut m_beCrossed.
- virtual void show () const

  Permets d'afficher une Tile.

7.19.1 Description détaillée

La classe Tile permets de définir une case de la map.

Voir également

Position

Auteur

Hocine HADID

7.19.2 Documentation des constructeurs et destructeur

7.19.2.1 Tile()

```
Tile::Tile (
            int x = 0,
            int y = 0,
            bool beCrossed = true )
```

Constructeur d'une Tile.

Paramètres

x	La ligne sur laquelle se situe la Tile
y	La colonne sur laquelle se situe la Tile
beCrossed	Si c'est possible de traverser la Tile

7.19.2.2 ~Tile()

```
Tile::~Tile ( )
```
Destructeur d'une Tile.

7.19.3 Documentation des fonctions membres

7.19.3.1 getBeCrossed()

```
bool Tile::getBeCrossed ( ) const
```
Accesseur de l'attribut m_beCrossed.

Renvoie

bool La possibilité de traverser la Tile ou non

7.19.3.2 getPosition()

```
Position Tile::getPosition ( ) const
```
Accesseur de l'attribut m_position.

Renvoie

Position La position de la Tile

7.19.3.3 setBeCrossed()

```
void Tile::setBeCrossed (
            bool beCrossed )
```
Accesseur de l'attribut m_beCrossed.

Paramètres

beCrossed	si c'est possible de traverser la Tile ou non

7.19.3.4 show()

```
void Tile::show ( ) const  [virtual]
```
Permets d'afficher une Tile.
Affiche la Tile, un espace

Auteur

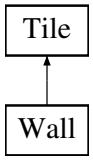
Hocine HADID

Réimplémentée dans Wall.
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Map/Tile.h
- src/src/Map/Tile.cpp

7.20 Référence de la classe Wall

La classe Wall hérite dec la classe Tile et permets de gérer les murs présents sur la map.
`#include <Wall.h>`
Graphe d'héritage de Wall:



Fonctions membres publiques

- Wall (int x=0, int y=0, bool beCrossed=false, bool destructible=true, int nbNecessaryBomb=2)

  Constructeur d'un Wall.
- ~Wall ()

  Destructeur d'un Wall.
- bool getDestructible () const

  Accesseur de l'attribut m_destructible.
- int getNbNecessaryBomb () const

  Accesseur de l'attribut m_nbNecessaryBomb.
- virtual void show () const override

  Permets d'afficher une Wall.
- void weaken ()

  Permets d'affaiblir un Wall.

7.20.1 Description détaillée

La classe Wall hérite dec la classe Tile et permets de gérer les murs présents sur la map.

Voir également

Tile

Auteur

Hocine HADID

7.20.2 Documentation des constructeurs et destructeur

7.20.2.1 Wall()

```
Wall::Wall (
            int x = 0,
            int y = 0,
            bool beCrossed = false,
            bool destructible = true,
            int nbNecessaryBomb = 2 )
```

Constructeur d'un Wall.

Paramètres

x	La ligne sur laquelle se situe le Wall
y	La colonne sur laquelle se situe le Wall
beCrossed	Si c'est possible de traverser le Wall
destructible	Si le Wall est destructible ou non
nbNecessaryBomb	Le nombre de bombe nécessaire pour détruire le Wall

7.20.2.2 ~Wall()

```
Wall::~Wall ( )
```

Destructeur d'un Wall.

7.20.3 Documentation des fonctions membres

7.20.3.1 getDestructible()

```
bool Wall::getDestructible ( ) const
```

Accesseur de l'attribut m_destructible.

Renvoie

true si le Wall est destructible, false sinon

7.20.3.2 getNbNecessaryBomb()

```
int Wall::getNbNecessaryBomb ( ) const
```

Accesseur de l'attribut m_nbNecessaryBomb.

Renvoie

int Le nombre de bombe nécessaire pour détruire le Wall

7.20.3.3 show()

```
void Wall::show ( ) const  [override], [virtual]
```

Permets d'afficher une Wall.
Affiche le Wall, un caractère "W" pour les murs, "w" pour les murs ne nécessitant qu'une seule bombe pour être détruit et "I" pour les murs indestructibles

Auteur

Hocine HADID

Réimplémentée à partir de Tile.

7.20.3.4 weaken()

```
void Wall::weaken ( )
```

Permets d'affaiblir un Wall.

Auteur

Hocine HADID

La documentation de cette classe a été générée à partir des fichiers suivants :

- src/include/Map/Wall.h
- src/src/Map/Wall.cpp

Chapter 8

Documentation des fichiers

8.1 Référence du fichier ProgBomberman.cpp

Fichier contenant la classe principal permettant de gérer le jeu du Bomberman.

```
#include <iostream>
#include <limits>
#include <fstream>
#include "src/include/engine/SystemGame.h"
```

Fonctions

- void showHome ()

  Permets d'afficher le message d'accueil.
- void showMenu ()

  Permets d'afficher le menu du jeu.
- void playGame (int level)

  Permets de lancer une parie.
- int **main** (void)

8.1.1 Description détaillée

Fichier contenant la classe principal permettant de gérer le jeu du Bomberman.

Auteur

Pierre CHEMIN & Hocine HADID

Version

0.1

Date

2022-04-08

Copyright

8.1.2 Documentation des fonctions

8.1.2.1 playGame()

```
void playGame (
            int level )
```

Permets de lancer une parie.

Paramètres

level	Le niveau de la Map choisit

8.1.2.2 showHome()

```
void showHome ( )
```

Permets d'afficher le message d'accueil.

8.1.2.3 showMenu()

```
void showMenu ( )
```

Permets d'afficher le menu du jeu.

8.2 Référence du fichier src/include/engine/SystemGame.h

Fichier contenant la classe SystemGame.

```
#include "../Persos/Bomberman.h"
#include "../Map/Map.h"
```

Classes

- class SystemGame

  Classe permettant de gérer une partie.

8.2.1 Description détaillée

Fichier contenant la classe SystemGame.

Auteur

Hocine HADID

Version

0.1

Date

2022-04-06

Copyright

8.3 Référence du fichier src/include/engine/utilities.h

Fichier contenant des éléments pouvant être utile de manière globale au projet.

Espaces de nommage

- utilities

Énumérations

- enum utilities::EDirection {
  utilities::TOP, utilities::BOTTOM, utilities::LEFT, utilities::RIGHT,
  utilities::NONE }

  Enumeration de direction.
- enum utilities::EBombExplosionDirection { utilities::LINE, utilities::COLUMN, utilities::CENTER, utilities::NOEXPLOSION }

  Enumeration de direction d'explosion.

8.3.1 Description détaillée

Fichier contenant des éléments pouvant être utile de manière globale au projet.

Auteur

Pierre CHEMIN

Version

0.1

Date

2022-04-06

Copyright

8.4 Référence du fichier src/include/Items/Arrow.h

Fichier contenant la classe Arrow.

```
#include "Item.h"
#include "../engine/utilities.h"
```

Classes

- class Arrow

  La classe Arrow hérite de la classe Item et gère les flèches lancées par le Bowman.

8.4.1 Description détaillée

Fichier contenant la classe Arrow.

Auteur

Hocine HADID

Version

0.1

Date

2022-04-06

Copyright

8.5 Référence du fichier src/include/Items/Bomb.h

Fichier contenant la classe Bomb.

```
#include "Item.h"
```

Classes

- class Bomb

  La classe Bomb hérite de la classe Item et permets de gérer les flèches.

8.5.1 Description détaillée

Fichier contenant la classe Bomb.

Auteur

Hocine HADID

Version

0.1

Date

2022-04-06

Copyright

Copyright (c) 2022

8.6 Référence du fichier src/include/Items/Item.h

Fichier contenant la classe Item.

```
#include "../Map/Position.h"
#include "../Map/Tile.h"
#include "../Persos/Bomberman.h"
#include <vector>
```

Classes

- class Item

  La classe abstraite Item.

8.6.1 Description détaillée

Fichier contenant la classe Item.

Auteur

Hocine HADID

Version

0.1

Date

2022-04-06

Copyright

Copyright (c) 2022

8.7 Référence du fichier src/include/Items/MoreBomb.h

Fichier contenant la classe MoreBomb.

```
#include "Item.h"
#include "../Persos/Bomberman.h"
```

Classes

- class MoreBomb

  La classe MoreBomb hérite de la classe Item et permets de gérer les items de type MoreBomb.

8.7.1 Description détaillée

Fichier contenant la classe MoreBomb.

Auteur

Hocine HADID

Version

0.1

Date

2022-04-06

Copyright

8.8 Référence du fichier src/include/Items/MoreLife.h

Fichier contenant la classe MoreLife.

```
#include "Item.h"
#include "../Persos/Bomberman.h"
```

Classes

- class MoreLife

  La classe MoreLife hérite de la classe Item et permets de gérer les items de type MoreLife.

8.8.1 Description détaillée

Fichier contenant la classe MoreLife.

Auteur

Hocine HADID

Version

0.1

Date

2022-04-06

Copyright

8.9 Référence du fichier src/include/Items/PowerUp.h

Fichier contenant la classe PowerUp.

```
#include "Item.h"
#include "../Persos/Bomberman.h"
```

Classes

- class PowerUp

  La classe PowerUp hérite de la classe Item et permets de gérer les items de type PowerUp.

8.9.1 Description détaillée

Fichier contenant la classe PowerUp.

Auteur

Hocine HADID

Version

0.1

Date

2022-04-06

Copyright

8.10 Référence du fichier src/include/Items/ScaleUp.h

Fichier contenant la classe ScaleUp.

```
#include "Item.h"
```

Classes

- class ScaleUp

  La classe ScaleUp hérite de la classe Item et permets de gérer les items de type ScaleUp.

8.10.1 Description détaillée

Fichier contenant la classe ScaleUp.

Auteur

Hocine HADID

Version

0.1

Date

2022-04-06

Copyright

8.11 Référence du fichier src/include/Items/SpeedUp.h

Fichier contenant la classe SpeedUp.

```
#include "Item.h"
#include "../Persos/Bomberman.h"
```

Classes

- class SpeedUp

  La classe SpeedUp hérite de la classe Item et permets de gérer les items de type SpeedUp.

8.11.1 Description détaillée

Fichier contenant la classe SpeedUp.

Auteur

Hocine HADID

Version

0.1

Date

2022-04-06

Copyright

8.12 Référence du fichier src/include/Map/Map.h

Fichier contenant la classe Map.

```
#include <vector>
#include "Tile.h"
#include "Position.h"
#include "Wall.h"
#include "../Persos/Ennemy.h"
#include "../Persos/Bomberman.h"
#include "../Items/Item.h"
#include "../Items/Bomb.h"
#include "../engine/utilities.h"
```

Classes

- class Map

  La classe Map permets de gérer une map du jeu Bomberman.

8.12.1 Description détaillée

Fichier contenant la classe Map.

Auteur

Pierre CHEMIN & Hocine HADID

Version

0.1

Date

2022-03-12

Copyright

8.13 Référence du fichier src/include/Map/MoveException.h

Fichier contenant la classe MoveException.

```
#include <exception>
#include <string>
```

Classes

- class MoveException

  La classe MoveException hérite de la classe std::exception.

8.13.1 Description détaillée

Fichier contenant la classe MoveException.

Auteur

Pierre CHEMIN

Version

0.1

Date

2022-04-08

Copyright

8.14 Référence du fichier src/include/Map/Position.h

Fichier contenant la classe Position.

Classes

- class Position

  La classe Position permets de définir une position.

8.14.1 Description détaillée

Fichier contenant la classe Position.

Auteur

Hocine HADID

Version

0.1

Date

2022-04-07

Copyright

8.15 Référence du fichier src/include/Map/Tile.h

Fichier contenant la classe Tile.

```
#include "Position.h"
```

Classes

- class Tile

  La classe Tile permets de définir une case de la map.

8.15.1 Description détaillée

Fichier contenant la classe Tile.

Auteur

Hocine HADID

Version

0.1

Date

2022-04-07

Copyright

8.16 Référence du fichier src/include/Map/Wall.h

Fichier contenant la classe Wall.

```
#include "Tile.h"
```

Classes

- class Wall

  La classe Wall hérite dec la classe Tile et permets de gérer les murs présents sur la map.

8.16.1 Description détaillée

Fichier contenant la classe Wall.

Auteur

Hocine HADID

Version

0.1

Date

2022-04-07

Copyright

Copyright (c) 2022

8.17 Référence du fichier src/include/Persos/Bomberman.h

Fichier contenant la classe Bomberman.

```
#include "Personnage.h"
#include "../Map/Position.h"
```

Classes

- class Bomberman

  La classe Bomberman hérite de la classe Personnage et permets de gérer un Bomberman (le joueur)

8.17.1 Description détaillée

Fichier contenant la classe Bomberman.

Auteur

Pierre CHEMIN

Version

0.1

Date

2022-04-07

Copyright

Copyright (c) 2022

8.18 Référence du fichier src/include/Persos/Bowman.h

Fichier contenant la classe Bowman.

```
#include "Ennemy.h"
#include "../engine/utilities.h"
#include "../Items/Arrow.h"
```

Classes

- class Bowman

  La classe Bowman hérite de la classe Ennemy et permets de gérer les ennemies de type Bowman.

8.18.1 Description détaillée

Fichier contenant la classe Bowman.

Auteur

Pierre CHEMIN

Version

0.1

Date

2022-04-07

Copyright

8.19 Référence du fichier src/include/Persos/Ennemy.h

Fichier contenant la classe Ennemy.

```
#include "Personnage.h"
#include "Bomberman.h"
#include "../Map/Position.h"
#include "../Map/Tile.h"
#include "../Items/Item.h"
#include "../engine/utilities.h"
#include <vector>
```

Classes

- class Ennemy

  La classe abstraite Ennemy qui hérite de la classe Personnage.

8.19.1 Description détaillée

Fichier contenant la classe Ennemy.

Auteur

Pierre CHEMIN

Version

0.1

Date

2022-04-07

Copyright

8.20 Référence du fichier src/include/Persos/Ghost.h

Fichier contenant la classe Ghost.

```
#include "Ennemy.h"
#include "Bomberman.h"
```

Classes

- class Ghost

  La classe Ghost hérite de la classe Ennemy et permets de gérer les ennemies de type Ghost.

8.20.1 Description détaillée

Fichier contenant la classe Ghost.

Auteur

Pierre CHEMIN

Version

0.1

Date

2022-04-07

Copyright

8.21 Référence du fichier src/include/Persos/Monster.h

Fichier contenant la classe Monster.

```
#include "Ennemy.h"
```

Classes

- class Monster

  La classe Monster hérite de la classe Ennemy et permets de gérer les ennemies de type Monster.

8.21.1 Description détaillée

Fichier contenant la classe Monster.

Auteur

Pierre CHEMIN

Version

0.1

Date

2022-04-07

Copyright

8.22 Référence du fichier src/include/Persos/Personnage.h

Fichier contenant la classe Personnage.

```
#include "../Map/Position.h"
#include "../engine/utilities.h"
```

Classes

- class Personnage

  La classe abstraite personnage.

8.22.1 Description détaillée

Fichier contenant la classe Personnage.

Auteur

Pierre CHEMIN

Version

0.1

Date

2022-04-07

Copyright