

Bombberman

Generated by Doxygen 1.9.3

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

std	??
std::chrono_literals	??

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Arrow	??
Bombe	??
Item	??
MoreBomb	??
MoreLife	??
PowerUp	??
ScaleUp	??
SpeedUp	??
Joueur	??
Map	??
Mob	??
Bowman	??
Ghost	??
Monster	??
Objectif	??
System	??
Wall	??

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Arrow	Classe representant la Arrow	??
Bombe	Classe representant la bombe	??
Bowman	Classe representant le Bowman , hérite de Mob	??
Ghost	Classe representant le Ghost , hérite de Mob	??
Item	Classe representant l' Item	??
Joueur	Classe representant la Joueur	??
Map	Classe representant la Map	??
Mob	Classe representant le Mob	??
Monster	Classe representant un Monster , hérite de Mob	??
MoreBomb	Classe representant l'item MoreBomb , hérite de Item	??
MoreLife	Classe representant l'item MoreLife , hérite de Item	??
Objectif	Classe representant l' Objectif	??
PowerUp	Classe representant l'item PowerUp , hérite de Item	??
ScaleUp	Classe representant l'item ScaleUp , hérite de Item	??
SpeedUp	Classe representant l'item SpeedUp , hérite de Item	??
System	Classe representant le System	??
Wall	Classe representant le Wall	??

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

Arrow.h	Classe de la flèche	??
Bombe.h	Classe de la bombe	??
Bowman.h	Classe du Bowman	??
Ghost.h	Classe du ghost	??
Item.h	Classe de l' Item	??
Joueur.h	Classe du Joueur	??
main.cpp	Main du programme	??
Map.h	Classe de la map	??
Mob.h	Classe du Mob	??
Monster.h	Classe du Monster	??
MoreBomb.h	Classe de l'item MoreBomb	??
MoreLife.h	Classe de l'item MoreLife	??
Objectif.h	Classe de la bombe	??
PowerUp.h	Classe de l'item PowerUp	??
ScaleUp.h	Classe de l'item ScaleUp	??
SpeedUp.h	Classe de l'item SpeedUp	??
System.h	Classe du System	??
Wall.h	Classe de la bombe	??

Chapter 5

Namespace Documentation

5.1 std Namespace Reference

Namespaces

- namespace [chrono_literals](#)

5.1.1 Detailed Description

espace de nommage

5.2 std::chrono_literals Namespace Reference

5.2.1 Detailed Description

espace de nommage (pour utiliser les sleep)

Chapter 6

Class Documentation

6.1 Arrow Class Reference

classe representant la [Arrow](#)

```
#include <Arrow.h>
```

Public Member Functions

- [Arrow](#) (int degat=1, int position=-1, char direction=' ', bool touche=false)
constructeur par défaut et initialisation
- void [seDeplacer](#) (int déplacement)
seDeplacer
- int [getDegat](#) ()
getteur degat
- int [getPosition](#) ()
getteur position
- char [getDirection](#) ()
getteur direction
- bool [getTouche](#) ()
getteur touche
- void [setDegat](#) (int degat)
setteur degat
- void [setPosition](#) (int position)
setteur position
- void [setDirection](#) (char direction)
setteur direction
- void [setTouche](#) (bool touche)
setteur touche

6.1.1 Detailed Description

classe representant la [Arrow](#)

Author

SINET Theo

La classe gere l'objet [Arrow](#)

6.1.2 Constructor & Destructor Documentation

6.1.2.1 Arrow()

```
Arrow::Arrow (
    int degat = 1,
    int position = -1,
    char direction = ' ',
    bool touche = false )
```

constructeur par default et initialisation

Author

SINET Theo

Parameters

<i>degat</i>	: degat de la flèche
<i>position</i>	: position de la flèche
<i>direction</i>	: direction de la flèche
<i>touche</i>	: si la flèche touche ou non Constructeur par default et initialisation de la classe Arrow

6.1.3 Member Function Documentation

6.1.3.1 getDegat()

```
int Arrow::getDegat ( ) [inline]
```

getteur degat

Author

SINET Theo

Returns

degat récupère les degats de la flèche

6.1.3.2 `getDirection()`

```
char Arrow::getDirection ( ) [inline]
```

getteur direction

Author

SINET Theo

Returns

direction récupère la direction de la flèche

6.1.3.3 `getPosition()`

```
int Arrow::getPosition ( ) [inline]
```

getteur position

Author

SINET Theo

Returns

position récupère la position de la flèche

6.1.3.4 `getTouche()`

```
bool Arrow::getTouche ( ) [inline]
```

getteur touche

Author

SINET Theo

Returns

touche récupère true si la flèche touche, false sinon

6.1.3.5 `seDeplacer()`

```
void Arrow::seDeplacer (
    int deplacement )
```

seDeplacer

Author

SINET Theo

Parameters

<i>deplacement</i>	: le sens du déplacement de la flèche Permet à la flèche de se déplacer
--------------------	---

6.1.3.6 setDegat()

```
void Arrow::setDegat (  
    int degat ) [inline]
```

setteur degat

Author

SINET Theo

Parameters

<i>degat</i>	: degat change la degat de la flèche
--------------	--------------------------------------

6.1.3.7 setDirection()

```
void Arrow::setDirection (  
    char direction ) [inline]
```

setteur direction

Author

SINET Theo

Parameters

<i>position</i>	: direction change la direction de la flèche
-----------------	--

6.1.3.8 setPosition()

```
void Arrow::setPosition (  
    int position ) [inline]
```

setteur position

Author

SINET Theo

Parameters

<i>position</i>	: position change la position de la flèche
-----------------	--

6.1.3.9 setTouche()

```
void Arrow::setTouche (
    bool touche ) [inline]
```

setteur touche

Author

SINET Theo

Parameters

<i>touche</i>	: touche change le boolean touche
---------------	-----------------------------------

The documentation for this class was generated from the following files:

- [Arrow.h](#)
- [Arrow.cpp](#)

6.2 Bombe Class Reference

classe representant la bombe

```
#include <Bombe.h>
```

Public Member Functions

- [Bombe](#) (int puissance=1, int portee=2, int position=-1, int tourExplosion=0)
constructeur par default et initialisation
- [Bombe](#) (const [Bombe](#) &b1)
constructeur par copie
- void [toucherPersonnage](#) ()
toucher un personnage
- int [getPuissance](#) ()
getteur puissance
- int [getPortee](#) ()

- getteur portee*
- int [getPosition](#) ()
- getteur position*
- int [getTourExplosion](#) ()
- getteur tourExplosion*
- void [setPuissance](#) (int puissance)
- setteur puissance*
- void [setPortee](#) (int portee)
- setteur portee*
- void [setPosition](#) (int position)
- setteur position*
- void [setTourExplosion](#) (int tourExplosion)
- setteur explosion bombe*

6.2.1 Detailed Description

classe representant la bombe

Author

SINET Theo

La classe gere l'objet bombe

6.2.2 Constructor & Destructor Documentation

6.2.2.1 Bombe() [1/2]

```
Bombe::Bombe (
    int puissance = 1,
    int portee = 2,
    int position = -1,
    int tourExplosion = 0 )
```

constructeur par default et initialisation

Author

SINET Theo

Parameters

<i>puissance</i>	: puissance de la bombe
<i>portee</i>	: porter de la bombe
<i>position</i>	: position de la bombe
<i>tourExplosion</i>	: tour avant l'explosion de la bombe Constructeur par default et initialisation de la classe Bombe

6.2.2.2 Bombe() [2/2]

```
Bombe::Bombe (
    const Bombe & b1 )
```

constructeur par copie

Author

SINET Theo

Parameters

Bombe	: une bombe Constructeur par copie de la classe Bombe
-----------------------	---

6.2.3 Member Function Documentation

6.2.3.1 getPortee()

```
int Bombe::getPortee ( ) [inline]
```

getteur portee

Author

SINET Theo

Returns

portee de la bombe récupère la portee de la bombe

6.2.3.2 getPosition()

```
int Bombe::getPosition ( ) [inline]
```

getteur position

Author

SINET Theo

Returns

position de la bombe récupère la position de la bombe

6.2.3.3 getPuissance()

```
int Bombe::getPuissance ( ) [inline]
```

getteur puissance

Author

SINET Theo

Returns

puissance de la bombe récupère la puissance de la bombe

6.2.3.4 getTourExplosion()

```
int Bombe::getTourExplosion ( ) [inline]
```

getteur tourExplosion

Author

SINET Theo

Returns

tour avant l'explosion de la bombe récupère le nombre de tour avant l'explosion de la bombe

6.2.3.5 setPortee()

```
void Bombe::setPortee (
    int portee ) [inline]
```

setteur portee

Author

SINET Theo

Parameters

<i>portee</i>	: portee change la portee de la bombe
---------------	---------------------------------------

6.2.3.6 setPosition()

```
void Bombe::setPosition (
    int position ) [inline]
```

setteur position

Author

SINET Theo

Parameters

<i>position</i>	: position change la position de la bombe
-----------------	---

6.2.3.7 setPuissance()

```
void Bombe::setPuissance (
    int puissance ) [inline]
```

setteur puissance

Author

SINET Theo

Parameters

<i>puissance</i>	: puissance change la puissance de la bombe
------------------	---

6.2.3.8 setTourExplosion()

```
void Bombe::setTourExplosion (
    int tourExplosion ) [inline]
```

setteur explosion bombe

Author

SINET Theo

Parameters

<i>tourExplosion</i>	: tour avant l'explosion change le nombre de tour avant l'explosion de la bombe
----------------------	---

6.2.3.9 toucherPersonnage()

```
void Bombe::toucherPersonnage ( )
```

toucher un personnage

Author

SINET Theo

The documentation for this class was generated from the following files:

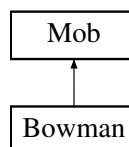
- [Bombe.h](#)
- [Bombe.cpp](#)

6.3 Bowman Class Reference

classe representant le [Bowman](#), hérite de [Mob](#)

```
#include <Bowman.h>
```

Inheritance diagram for Bowman:



Public Member Functions

- [Bowman](#) (int nbFleche=1, [Arrow](#) *arrow=new [Arrow](#)())
constructeur par défaut et initialisation
- bool [isBowman](#) ()
est un [Bowman](#)
- bool [peutTirer](#) (int positionJoueur, std::string *tab, int nbLigne, int nbColonne)
verifie si le [Bowman](#) peut tirer
- char [tirer](#) (int positionJoueur, std::string *tab, int nbLigne, int nbColonne)
tirer une flèche
- int [getNbFleche](#) ()
getteur nbFleche
- [Arrow](#) * [getArrow](#) ()
getteur arrow
- void [setNbFleche](#) (int nbFleche)
setteur nbFleche

Additional Inherited Members

6.3.1 Detailed Description

classe representant le [Bowman](#), hérite de [Mob](#)

Author

SINET Theo La classe gere l'objet [Bowman](#)

6.3.2 Constructor & Destructor Documentation

6.3.2.1 Bowman()

```
Bowman::Bowman (
    int nbFleche = 1,
    Arrow * arrow = new Arrow() )
```

constructeur par défaut et initialisation

Author

SINET Theo Constructeur par défaut et initialisation du [Bowman](#)

6.3.3 Member Function Documentation

6.3.3.1 getArrow()

```
Arrow * Bowman::getArrow ( ) [inline], [virtual]
```

getteur arrow

Author

SINET Theo

Returns

la flèche du [Bowman](#) récupère la flèche du [Bowman](#)

Reimplemented from [Mob](#).

6.3.3.2 getNbFleche()

```
int Bowman::getNbFleche ( ) [inline], [virtual]
```

getteur nbFleche

Author

SINET Theo

Returns

le nombre de flèche récupère le nombre de flèche du [Bowman](#)

Reimplemented from [Mob](#).

6.3.3.3 isBowman()

```
bool Bowman::isBowman ( ) [inline], [virtual]
```

est un [Bowman](#)

Author

SINET Theo

Returns

true Methode qui retourne true lorsque elle est appelée

Reimplemented from [Mob](#).

6.3.3.4 peutTirer()

```
bool Bowman::peutTirer (
    int positionJoueur,
    std::string * tab,
    int nbLigne,
    int nbColonne ) [virtual]
```

verifie si le [Bowman](#) peut tirer

Author

SINET Theo

Parameters

<i>positionJoueur</i>	: position du joueur
<i>tab</i>	: la map
<i>nbLigne</i>	: nombre de ligne de la map
<i>nbColonne</i>	: nombre de colone de la map

Returns

true si le [Bowman](#) peut tirer, fasle sinon Methode qui verifie si le joueur peut tirer

Reimplemented from [Mob](#).

6.3.3.5 setNbFleche()

```
void Bowman::setNbFleche (
    int nbFleche ) [inline], [virtual]
```

setteur nbFleche

Author

SINET Theo

Parameters

<i>nbFleche</i>	: le nombre de flèche du Bowman change le nombre de flèche du Bowman
-----------------	--

Reimplemented from [Mob](#).

6.3.3.6 tirer()

```
char Bowman::tirer (
    int positionJoueur,
    std::string * tab,
    int nbLigne,
    int nbColonne ) [virtual]
```

tirer une flèche

Author

SINET Theo

Parameters

<i>positionJoueur</i>	: position du joueur
<i>tab</i>	: la map
<i>nbLigne</i>	: nombre de ligne de la map
<i>nbColonne</i>	: nombre de colone de la map

Returns

la direction du tir Methode qui permet au [Bowman](#) de tirer une flèche

Reimplemented from [Mob](#).

The documentation for this class was generated from the following files:

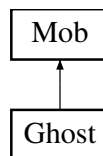
- [Bowman.h](#)
- [Bowman.cpp](#)

6.4 Ghost Class Reference

classe representant le [Ghost](#), hérite de [Mob](#)

```
#include <Ghost.h>
```

Inheritance diagram for Ghost:

**Public Member Functions**

- [Ghost](#) ()
constructeur par défaut
- bool [isBowman](#) ()
est un [Bowman](#)

Additional Inherited Members

6.4.1 Detailed Description

classe representant le [Ghost](#), hérite de [Mob](#)

Author

SINET Theo La classe gere l'objet [Ghost](#)

6.4.2 Constructor & Destructor Documentation

6.4.2.1 Ghost()

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

constructeur par défaut

Author

SINET Theo Constructeur par défaut du [Ghost](#)

6.4.3 Member Function Documentation

6.4.3.1 isBowman()

```
bool Ghost::isBowman ( ) [inline], [virtual]
```

est un [Bowman](#)

Author

SINET Theo

Returns

false Methode qui retourne false lorsque elle est appelée

Reimplemented from [Mob](#).

The documentation for this class was generated from the following files:

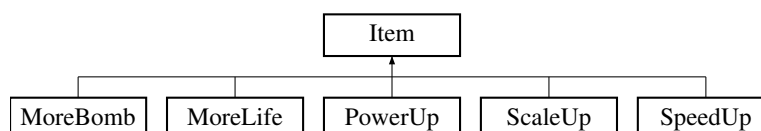
- [Ghost.h](#)
- Ghost.cpp

6.5 Item Class Reference

classe representant l'[Item](#)

```
#include <Item.h>
```

Inheritance diagram for Item:



Public Member Functions

- [Item](#) (int position=-1)
constructeur par default et initialisation
- int [getPosition](#) ()
getteur position
- void [setPosition](#) (int position)
setteur position
- virtual void [ajouterVie](#) ([Joueur](#) *joueur)
ajouterVie
- virtual void [ajouterBombe](#) ([Joueur](#) *joueur)
ajouterBombe
- virtual void [augmenterVitesse](#) ([Joueur](#) *joueur)
augmenterVitesse
- virtual void [augmenterPuissance](#) ([Joueur](#) *joueur)
augmenterPuissance
- virtual void [augmenterPortee](#) ([Joueur](#) *joueur)
augmenterPortee

6.5.1 Detailed Description

classe representant l'[Item](#)

Author

SINET Theo La classe gere l'objet [Item](#)

6.5.2 Constructor & Destructor Documentation

6.5.2.1 Item()

```
Item::Item (  
    int position = -1 )
```

constructeur par default et initialisation

Author

SINET Theo

Parameters

<i>position</i>	: position de l' Item Constructeur par default et initialisation de la classe Item
-----------------	--

6.5.3 Member Function Documentation

6.5.3.1 ajouterBombe()

```
virtual void Item::ajouterBombe (
    Joueur * joueur ) [inline], [virtual]
```

ajouterBombe

Author

SINET Theo

Parameters

<i>joueur</i>	: le joueur augmente le nombre de bombe du joueur
---------------	---

Reimplemented in [MoreBomb](#).

6.5.3.2 ajouterVie()

```
virtual void Item::ajouterVie (
    Joueur * joueur ) [inline], [virtual]
```

ajouterVie

Author

SINET Theo

Parameters

<i>joueur</i>	: le joueur ajoute de la vie au joueur
---------------	--

Reimplemented in [MoreLife](#).

6.5.3.3 augmenterPortee()

```
virtual void Item::augmenterPortee (
    Joueur * joueur ) [inline], [virtual]
```

augmenterPortee

Author

SINET Theo

Parameters

<i>joueur</i>	: le joueur augmente la porter de la bombe du joueur
---------------	--

Reimplemented in [ScaleUp](#).**6.5.3.4 augmenterPuissance()**

```
virtual void Item::augmenterPuissance (
    Joueur * joueur ) [inline], [virtual]
```

augmenterPuissance

Author

SINET Theo

Parameters

<i>joueur</i>	: le joueur augmente la puissance de la bombe du joueur
---------------	---

Reimplemented in [PowerUp](#).**6.5.3.5 augmenterVitesse()**

```
virtual void Item::augmenterVitesse (
    Joueur * joueur ) [inline], [virtual]
```

augmenterVitesse

Author

SINET Theo

Parameters

<i>joueur</i>	: le joueur augmente la vitesse du joueur
---------------	---

Reimplemented in [SpeedUp](#).

6.5.3.6 getPosition()

```
int Item::getPosition ( ) [inline]
```

getteur position

Author

SINET Theo

Returns

position de l'item récupère la position de l'item

6.5.3.7 setPosition()

```
void Item::setPosition (
    int position ) [inline]
```

setteur position

Author

SINET Theo

Parameters

<i>position</i>	: la position de l'item met a jour la position de la item
-----------------	---

The documentation for this class was generated from the following files:

- [Item.h](#)
- [Item.cpp](#)

6.6 Joueur Class Reference

classe representant la [Joueur](#)

```
#include <Joueur.h>
```

Public Member Functions

- [Joueur](#) (int hp=3, int speed=1, int position=50, int nbBombMax=1, int nbBombRestante=1, [Bombe](#) *bombe=new [Bombe](#)())

- constructeur par défaut et initialisation*
- [Joueur](#) (const [Joueur](#) &j1)
- constructeur par copie*
- [~Joueur](#) ()
- Destructeur.*
- void [seDeplacer](#) (char direction, int nbColonne)
- se deplacer*
- void [poserBombe](#) ()
- poser bombe*
- int [getHP](#) ()
- getteur hp*
- int [getSpeed](#) ()
- getteur speed*
- int [getPosition](#) ()
- getteur position*
- int [getNbBombMax](#) ()
- getteur nbBombMax*
- int [getNbBombRestante](#) ()
- getteur nbBombRestante*
- [Bombe](#) * [getBombe](#) ()
- getteur bombe*
- void [setHP](#) (int hp)
- setteur hp*
- void [setSpeed](#) (int speed)
- setteur speed*
- void [setPosition](#) (int position)
- setteur position*
- void [setNbBombMax](#) (int nbBombMax)
- setteur nbBombMax*
- void [setNbBombRestante](#) (int nbBombRestante)
- setteur nbBombRestante*
- void [setBombe](#) ([Bombe](#) *b1)
- setteur bombe*

6.6.1 Detailed Description

classe representant la [Joueur](#)

Author

SINET Theo

La classe gere l'objet [Joueur](#)

6.6.2 Constructor & Destructor Documentation

6.6.2.1 Joueur() [1/2]

```
Joueur::Joueur (
    int hp = 3,
    int speed = 1,
    int position = 50,
    int nbBombMax = 1,
    int nbBombRestante = 1,
    Bombe * bombe = new Bombe () )
```

constructeur par défaut et initialisation

Author

SINET Theo

Parameters

<i>hp</i>	: hp du joueur
<i>speed</i>	: speed du joueur
<i>position</i>	: position du joueur
<i>nbBombMax</i>	: nombre de bombe max à poser
<i>nbBombRestante</i>	: nombre de bombe restante à poser
<i>bombe</i>	: bombe du joueur Constructeur par défaut et initialisation du joueur

6.6.2.2 Joueur() [2/2]

```
Joueur::Joueur (
    const Joueur & j1 )
```

constructeur par copie

Author

SINET Theo

Parameters

<i>Joueur</i>	: joueur Constructeur par copie du joueur
---------------	---

6.6.2.3 ~Joueur()

```
Joueur::~~Joueur ( ) [inline]
```

Destructeur.

Author

SINET Theo Destructeur de la classe [Joueur](#)

6.6.3 Member Function Documentation

6.6.3.1 getBombe()

```
Bombe * Joueur::getBombe ( ) [inline]
```

getteur bombe

Author

SINET Theo

Returns

la bombe du joueur Permet de récupérer la bombe du joueur

6.6.3.2 getHP()

```
int Joueur::getHP ( ) [inline]
```

getteur hp

Author

SINET Theo

Returns

hp du joueur Permet de récupérer les hp du joueur

6.6.3.3 getNbBombMax()

```
int Joueur::getNbBombMax ( ) [inline]
```

getteur nbBombMax

Author

SINET Theo

Returns

le nombre de bombe max Permet de récupérer le nombre max de bombe que le joueur peut poser

6.6.3.4 getNbBombRestante()

```
int Joueur::getNbBombRestante ( ) [inline]
```

getteur nbBombRestante

Author

SINET Theo

Returns

le nombre de bombe restante Permet de récupérer le nombre de bombe restante que le joueur peut poser

6.6.3.5 getPosition()

```
int Joueur::getPosition ( ) [inline]
```

getteur position

Author

SINET Theo

Returns

position du joueur Permet de récupérer la position du joueur

6.6.3.6 getSpeed()

```
int Joueur::getSpeed ( ) [inline]
```

getteur speed

Author

SINET Theo

Returns

speed du joueur Permet de récupérer la speed du joueur

6.6.3.7 poserBombe()

```
void Joueur::poserBombe ( )
```

poser bombe

Author

SINET Theo Permet au joueur de poser une bombe

6.6.3.8 seDeplacer()

```
void Joueur::seDeplacer (
    char direction,
    int nbColonne )
```

se deplacer

Author

DARVILLE Killian, SINET Theo permet au joueur de se déplacer

6.6.3.9 setBombe()

```
void Joueur::setBombe (
    Bombe * b1 ) [inline]
```

setteur bombe

Author

SINET Theo

Parameters

Bombe	: une bombe actualise la bombe du joueur
-----------------------	--

6.6.3.10 setHP()

```
void Joueur::setHP (
    int hp ) [inline]
```

setteur hp

Author

SINET Theo

Parameters

<i>hp</i>	: les hp du joueur actualise les hp du joueur
-----------	---

6.6.3.11 setNbBombMax()

```
void Joueur::setNbBombMax (
    int nbBombMax ) [inline]
```

setteur nbBombMax

Author

SINET Theo

Parameters

<i>nbBombMax</i>	: le nombre de bombe max du joueur actualise le nombre de bombe max du joueur
------------------	---

6.6.3.12 setNbBombRestante()

```
void Joueur::setNbBombRestante (
    int nbBombRestante ) [inline]
```

setteur nbBombRestante

Author

SINET Theo

Parameters

<i>nbBombRestante</i>	: le nombre de bombe restant du joueur actualise le nombre de bombe restant du joueur
-----------------------	---

6.6.3.13 setPosition()

```
void Joueur::setPosition (
    int position ) [inline]
```

setteur position

Author

SINET Theo

Parameters

<i>position</i>	: la position du joueur actualise la position du joueur
-----------------	---

6.6.3.14 setSpeed()

```
void Joueur::setSpeed (
    int speed ) [inline]
```

setteur speed

Author

SINET Theo

Parameters

<i>speed</i>	: la speed du joueur actualise la speed du joueur
--------------	---

The documentation for this class was generated from the following files:

- [Joueur.h](#)
- [Joueur.cpp](#)

6.7 Map Class Reference

classe representant la [Map](#)

```
#include <Map.h>
```

Public Member Functions

- [Map](#) (int nbLigne=0, int nbColonne=0, int nbCaractere=0, std::string *tab=new std::string(), std::string nom←Fichier="maptest.txt", bool emoji=false, [Wall](#) **wall=new [Wall](#) *(), int nbWallMap=0, [Objectif](#) *objectif=new [Objectif](#)())
constructeur par default et initialisation
- void [affichageMap](#) (int nbHP, int puissanceBombe, int nbBombMax, int speed, int portee)
affiche la map
- void [actualiserMap](#) ()

- actualise la map*
- void [lectureMap](#) ()
lecture de la map
- void [infoMap](#) ()
information sur la map
- void [afficherPersoObjet](#) (int pos, char lettre)
afficher bombe et fleche
- void [gestionDeplacementMap](#) (int pos, int x)
gère les déplacements des entités
- void [explosionBombe](#) (int pos, int porteeBombe)
fait exploser une bombe
- void [retirerExplosionBombe](#) (int positionBombe)
retire l'explosion de la bombe
- bool [verifJoueurBloque](#) (int position)
verifie si le joueur est bloquer
- void [deplacementFleche](#) (int pos, char direction)
deplacement de la flèche
- int [compteMurs](#) ()
compte le nombre de murs dans la map
- void [alloueMurs](#) ()
alloue le nombre de murs qu'il faut
- void [positionObjectif](#) ()
trouver l'objectif
- int [getNbLigne](#) ()
getteur nbLigne
- int [getNbColonne](#) ()
getteur nbColonne
- int [getNbCaractere](#) ()
getteur nbColonne
- std::string * [getTab](#) ()
getteur tab
- std::string [getNomFichier](#) ()
getteur nomFichier
- bool [getEmoji](#) ()
getteur emoji
- void [setNbLigne](#) (int nbLigne)
setteur nbLigne
- void [setNbColonne](#) (int nbColonne)
setteur nbColonne
- void [setNbCaractere](#) (int nbCaractere)
setteur nbCaractere
- void [setTab](#) (std::string *tab)
setteur tab
- void [setNomFichier](#) (std::string nomFichier)
setteur nomFichier
- void [setEmoji](#) (bool emoji)
setteur emoji
- int * [getPorteePossibleBombe](#) ()
getteur porteePossibleBombe
- [Wall](#) ** [getWall](#) ()
getteur wall
- int [getNbWallMap](#) ()
getteur nombre de murs de la [Map](#)
- [Objectif](#) * [getObjectif](#) ()

6.7.1 Detailed Description

classe representant la [Map](#)

Author

DARVILLE Killian, SINET Theo

La classe gere l'objet [Map](#)

6.7.2 Constructor & Destructor Documentation

6.7.2.1 Map()

```
Map::Map (
    int nbLigne = 0,
    int nbColonne = 0,
    int nbCaractere = 0,
    std::string * tab = new std::string(),
    std::string nomFichier = "maptest.txt",
    bool emoji = false,
    Wall ** wall = new Wall*(),
    int nbWallMap = 0,
    Objectif * objectif = new Objectif() )
```

constructeur par default et initialisation

Author

SINET Theo

Parameters

<i>nbLigne</i>	: nombre de ligne de la map
<i>nbColonne</i>	: nombre de colonne de la map
<i>nbCaractere</i>	: nombre de caractere de la map
<i>tab</i>	: tableau de string contenant la map
<i>nomFichier</i>	: nom du fichier .txt de la map
<i>emoji</i>	: choix du joueur (version emoji ou version simple)
<i>wall</i>	: tous les murs de la map
<i>nbWallMap</i>	: nombre de murs Constructeur par default et initialisation de la classe Map

6.7.3 Member Function Documentation

6.7.3.1 actualiserMap()

```
void Map::actualiserMap ( )
```

actualise la map

Author

DARVILLE Killian Actualise la map dans le fichier .txt

6.7.3.2 affichageMap()

```
void Map::affichageMap (
    int nbHP,
    int puissanceBombe,
    int nbBombMax,
    int speed,
    int portee )
```

affiche la map

Author

DARVILLE Killian Affiche la map dans le terminal

6.7.3.3 afficherPersoObjet()

```
void Map::afficherPersoObjet (
    int pos,
    char lettre )
```

afficher bombe et fleche

Author

SINET Theo

Parameters

<i>pos</i>	: position de l'objet
<i>lettre</i>	: lettre de l'objet Affiche la bombe quand elle est poser et la flèche quand elle est tirer

6.7.3.4 alloueMurs()

```
void Map::alloueMurs ( )
```

alloue le nombre de murs qu'il faut

Author

SINET Theo alloue le nombre de murs qu'il faut

6.7.3.5 compteMurs()

```
int Map::compteMurs ( )
```

compte le nombre de murs dans la map

Author

SINET Theo

Returns

le nombre de murs compte le nombre de murs dans la map

6.7.3.6 deplacementFleche()

```
void Map::deplacementFleche (
    int pos,
    char direction )
```

deplacement de la flèche

Author

DARVILLE Killian

Parameters

<i>pos</i>	: position de la flèche
<i>direction</i>	: direction de la flèche Déplace la flèche sur la map

6.7.3.7 explosionBombe()

```
void Map::explosionBombe (
    int pos,
    int porteeBombe )
```

fait exploser une bombe

Author

DARVILLE Killian

Parameters

<i>pos</i>	: position de l'entité
<i>porteePossibleBombe</i>	: porter de la bombe selon les éléments de la map
<i>porteeBombe</i>	: porter max de la bombe Permet de faire exploser une bombe

6.7.3.8 gestionDeplacementMap()

```
void Map::gestionDeplacementMap (
    int pos,
    int x )
```

gère les déplacements des entités

Author

DARVILLE Killian

Parameters

<i>pos</i>	: position de l'entité
<i>x</i>	: futur position de l'entité Permet aux entités de se déplacer

6.7.3.9 getEmoji()

```
bool Map::getEmoji ( ) [inline]
```

getteur emoji

Author

SINET Theo

Returns

true si le joueur joue avec les emoji, false sinon Permet de savoir si le joueur joue avec les emoji ou non

6.7.3.10 getNbCaractere()

```
int Map::getNbCaractere ( ) [inline]
```

getteur nbColonne

Author

SINET Theo

Returns

le nombre de caractere de la map retourne le nombre de caractere de la map

6.7.3.11 getNbColonne()

```
int Map::getNbColonne ( ) [inline]
```

getteur nbColonne

Author

SINET Theo

Returns

le nombre de colonne de la map retourne le nombre de colonne de la map

6.7.3.12 getNbLigne()

```
int Map::getNbLigne ( ) [inline]
```

getteur nbLigne

Author

SINET Theo

Returns

le nombre de ligne de la map retourne le nombre de ligne de la map

6.7.3.13 getNbWallMap()

```
int Map::getNbWallMap ( ) [inline]
```

getteur nombre de murs de la [Map](#)

Author

SINET Theo

Returns

le nombre de murs de la [Map](#) récupère nombre de murs de la [Map](#)

6.7.3.14 getNomFichier()

```
std::string Map::getNomFichier ( ) [inline]
```

getteur nomFichier

Author

SINET Theo

Returns

le nom du fichier .txt retourne le nom du fichier .txt

6.7.3.15 getPorteePossibleBombe()

```
int * Map::getPorteePossibleBombe ( ) [inline]
```

getteur porteePossibleBombe

Author

SINET Theo

Returns

le tableau de porter de la bombe récupère le tableau de porter de la bombe

6.7.3.16 getTab()

```
std::string * Map::getTab ( ) [inline]
```

getteur tab

Author

SINET Theo

Returns

la map retourne la map

6.7.3.17 getWall()

```
Wall ** Map::getWall ( ) [inline]
```

getteur wall

Author

SINET Theo

Returns

le tableau de murs récupère le tableau de murs

6.7.3.18 infoMap()

```
void Map::infoMap ( )
```

information sur la map

Author

DARVILLE Killian Récupère le nombre de ligne, nombre de colonne et nombre de caractère du fichier .txt

6.7.3.19 lectureMap()

```
void Map::lectureMap ( )
```

lecture de la map

Author

DARVILLE Killian Lis la map dans dans le fichier .txt

6.7.3.20 positionObjectif()

```
void Map::positionObjectif ( )
```

trouver l'objectif

Author

SINET Theo trouve l'objectif

6.7.3.21 retirerExplosionBombe()

```
void Map::retirerExplosionBombe (
    int positionBombe )
```

retire l'explosion de la bombe

Author

DARVILLE Killian

Parameters

<i>pos</i>	: position de l'entité
<i>porteePossibleBombe</i>	: porter de la bombe selon les éléments de la map Permet de retirer l'explosion de la bombe

6.7.3.22 setEmoji()

```
void Map::setEmoji (
    bool emoji ) [inline]
```

setteur emoji

Author

SINET Theo

Parameters

<i>emoji</i>	: emoji change l'emoji
--------------	------------------------

6.7.3.23 setNbCaractere()

```
void Map::setNbCaractere (
    int nbCaractere ) [inline]
```

setteur nbCaractere

Author

SINET Theo

Parameters

<i>nbLigne</i>	: nombre de caractere change le nombre de nbCaractere
----------------	---

6.7.3.24 setNbColonne()

```
void Map::setNbColonne (
    int nbColonne ) [inline]
```

setteur nbColonne

Author

SINET Theo

Parameters

<i>nbLigne</i>	: nombre de colonne change le nombre de colonne
----------------	---

6.7.3.25 setNbLigne()

```
void Map::setNbLigne (
    int nbLigne ) [inline]
```


setteur nbLigne

Author

SINET Theo

Parameters

<i>nbLigne</i>	: nombre de ligne change le nombre de ligne
----------------	---

6.7.3.26 setNomFichier()

```
void Map::setNomFichier (
    std::string nomFichier ) [inline]
```

setteur nomFichier

Author

SINET Theo

Parameters

<i>nomFichier</i>	: le nom du fichier change le nom du fichier
-------------------	--

6.7.3.27 setTab()

```
void Map::setTab (
    std::string * tab ) [inline]
```

setteur tab

Author

SINET Theo

Parameters

<i>tab</i>	: le tableau change la tableau (la map)
------------	---

6.7.3.28 verifJoueurBloque()

```
bool Map::verifJoueurBloque (
    int position )
```

verifie si le joueur est bloquer

Author

DARVILLE Killian

Parameters

<i>position</i>	: position du joueur
-----------------	----------------------

Returns

false si le joueur est bloquer, true sinon vérifie si le joueur est bloquer

The documentation for this class was generated from the following files:

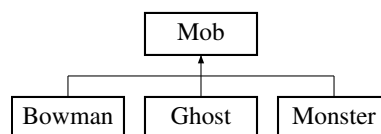
- [Map.h](#)
- [Map.cpp](#)

6.8 Mob Class Reference

classe representant le [Mob](#)

```
#include <Mob.h>
```

Inheritance diagram for Mob:



Public Member Functions

- [Mob](#) (int [hp](#)=1, int [speed](#)=1, int [damage](#)=1, int [position](#)=42, int [tourAttente](#)=0, bool [mort](#)=false)
constructeur par defaut et initialisation
- [Mob](#) (const [Mob](#) &m1)
constructeur par copie
- void [seDeplacer](#) (int positionJoueur, int nbLigne, int nbColonne)
se deplacer
- int [attaquer](#) (int hpJoueur)
attaquer

- virtual bool `isBowman` ()
est un [Bowman](#)
- virtual bool `peutTirer` (int positionJoueur, std::string *tab, int nbLigne, int nbColonne)
est un [Bowman](#)
- virtual char `tirer` (int positionJoueur, std::string *tab, int nbLigne, int nbColonne)
tirer une flèche
- virtual int `getNbFleche` ()
getteur nbFleche
- virtual [Arrow](#) * `getArrow` ()
getteur arrow
- virtual void `setNbFleche` (int nbFleche)
setteur nbFleche
- int `getHP` ()
getteur hp
- int `getSpeed` ()
getteur speed
- int `getDamage` ()
getteur damage
- int `getPosition` ()
getteur position
- int `getTourAttente` ()
getteur tourAttente
- bool `getMort` ()
getteur mort
- void `setHP` (int hp)
setteur hp
- void `setSpeed` (int speed)
setteur speed
- void `setDamage` (int damage)
setteur damage
- void `setPosition` (int position)
setteur position
- void `setTourAttente` (int tourAttente)
setteur tourAttente
- void `setMort` (bool mort)
setteur mort

Protected Member Functions

- int `getLigne` (int positionPerso, int nbLigne, int nbColonne)
récupère la ligne du [Mob](#)
- int `getColumnne` (int positionPerso, int nbColonne)
récupère la colonne du [Mob](#)

Protected Attributes

- int `hp`
- int `speed`
- int `damage`
- int `position`
- int `tourAttente`
- bool `mort`

6.8.1 Detailed Description

classe representant le [Mob](#)

Author

SINET Theo

La classe gere l'objet [Mob](#)

6.8.2 Constructor & Destructor Documentation

6.8.2.1 Mob() [1/2]

```
Mob::Mob (
    int hp = 1,
    int speed = 1,
    int damage = 1,
    int position = 42,
    int tourAttente = 0,
    bool mort = false )
```

constructeur par default et initialisation

Author

SINET Theo

Parameters

<i>hp</i>	: hp du Mob
<i>speed</i>	: speed du Mob
<i>damage</i>	: damage du Mob
<i>position</i>	: position du Mob
<i>tourAttente</i>	: tour avant que le Mob puisse jouer Constructeur par default et initialisation de la classe Mob

6.8.2.2 Mob() [2/2]

```
Mob::Mob (
    const Mob & m1 )
```

constructeur par copie

Author

SINET Theo

Parameters

Mob	: un Mob Constructeur par copie de la classe Mob
---------------------	--

6.8.3 Member Function Documentation

6.8.3.1 attaquer()

```
int Mob::attaquer (
    int hpJoueur )
```

attaquer

Author

SINET Theo

Returns

les hp du joueur après avoir pris des dégats Permet au [Mob](#) d'attaquer et de faire des dégats

6.8.3.2 getArrow()

```
virtual Arrow * Mob::getArrow ( ) [inline], [virtual]
```

getteur arrow

Author

SINET Theo

Returns

la flèche du [Bowman](#) récupère la flèche du [Bowman](#)

Reimplemented in [Bowman](#).

6.8.3.3 getColumnne()

```
int Mob::getColumnne (
    int positionPerso,
    int nbColonne ) [protected]
```

récupère la colonne du [Mob](#)

Author

SINET Theo

Parameters

<i>positionPerso</i>	: position du Mob
<i>nbColonne</i>	: nombre de colonne de la map

Returns

le colonne du [Mob](#) récupère la colonne du [Mob](#)

6.8.3.4 getDamage()

```
int Mob::getDamage ( ) [inline]
```

getteur damage

Author

SINET Theo

Returns

damage du [Mob](#) Permet de récupérer les damage du [Mob](#)

6.8.3.5 getHP()

```
int Mob::getHP ( ) [inline]
```

getteur hp

Author

SINET Theo

Returns

hp du [Mob](#) Permet de récupérer les hp du [Mob](#)

6.8.3.6 getLigne()

```
int Mob::getLigne (
    int positionPerso,
    int nbLigne,
    int nbColonne ) [protected]
```

récupère la ligne du [Mob](#)

Author

SINET Theo

Parameters

<i>positionPerso</i>	: position du Mob
<i>nbLigne</i>	: nombre de ligne de la map
<i>nbColonne</i>	: nombre de colonne de la map

Returns

le ligne du [Mob](#) récupère la ligne du [Mob](#)

6.8.3.7 getMort()

```
bool Mob::getMort ( ) [inline]
```

getteur mort

Author

SINET Theo

Returns

si le joueur est mort Permet de récupérer si le joueur est mort

6.8.3.8 getNbFleche()

```
virtual int Mob::getNbFleche ( ) [inline], [virtual]
```

getteur nbFleche

Author

SINET Theo

Returns

le nombre de flèche récupère le nombre de flèche du [Bowman](#)

Reimplemented in [Bowman](#).

6.8.3.9 getPosition()

```
int Mob::getPosition ( ) [inline]
```

getteur position

Author

SINET Theo

Returns

position du [Mob](#) Permet de récupérer la position du [Mob](#)

6.8.3.10 getSpeed()

```
int Mob::getSpeed ( ) [inline]
```

getteur speed

Author

SINET Theo

Returns

speed du [Mob](#) Permet de récupérer la speed du [Mob](#)

6.8.3.11 getTourAttente()

```
int Mob::getTourAttente ( ) [inline]
```

getteur tourAttente

Author

SINET Theo

Returns

nombre de tour en attente Permet de récupérer le nombre de tour durant lequel le [Mob](#) est en attente

6.8.3.12 isBowman()

```
virtual bool Mob::isBowman ( ) [inline], [virtual]
```

est un [Bowman](#)

Author

SINET Theo

Returns

false Methode qui retourne false lorsque elle est appelée

Reimplemented in [Bowman](#), [Ghost](#), and [Monster](#).

6.8.3.13 peutTirer()

```
virtual bool Mob::peutTirer (
    int positionJoueur,
    std::string * tab,
    int nbLigne,
    int nbColonne ) [inline], [virtual]
```

est un [Bowman](#)

Parameters

<i>positionJoueur</i>	: position du joueur
<i>tab</i>	: tableau de jeu
<i>nbLigne</i>	: nombre de ligne de la map
<i>nbColonne</i>	: nombre de colonne de la map

Returns

false Methode qui retourne false lorsque elle est appelée

Reimplemented in [Bowman](#).

6.8.3.14 seDeplacer()

```
void Mob::seDeplacer (
    int positionJoueur,
    int nbLigne,
    int nbColonne )
```

se deplacer

Author

DARVILLE Killian, SINET Theo permet au [Mob](#) de se déplacer

6.8.3.15 setDamage()

```
void Mob::setDamage (
    int damage ) [inline]
```

setteur damage

Author

SINET Theo

Parameters

<i>speed</i>	: les damage du Mob actualise les damage du Mob
--------------	---

6.8.3.16 setHP()

```
void Mob::setHP (
    int hp ) [inline]
```

setteur hp

Author

SINET Theo

Parameters

<i>hp</i>	: les hp du Mob actualise les hp du Mob
-----------	---

6.8.3.17 setMort()

```
void Mob::setMort (
    bool mort ) [inline]
```

setteur mort

Author

SINET Theo

Parameters

<i>mort</i>	: si le Mob est mort ou non actualise le boolean mort
-------------	---

6.8.3.18 setNbFleche()

```
virtual void Mob::setNbFleche (
    int nbFleche ) [inline], [virtual]
```

setteur nbFleche

Author

SINET Theo

Parameters

<i>nbFleche</i>	: le nombre de flèche du Bowman change le nombre de flèche du Bowman
-----------------	--

Reimplemented in [Bowman](#).**6.8.3.19 setPosition()**

```
void Mob::setPosition (
    int position ) [inline]
```

setteur position

Author

SINET Theo

Parameters

<i>position</i>	: la position du Mob actualise la position du Mob
-----------------	---

6.8.3.20 setSpeed()

```
void Mob::setSpeed (
    int speed ) [inline]
```

setteur speed

Author

SINET Theo

Parameters

<i>speed</i>	: la speed du Mob actualise la speed du Mob
--------------	---

6.8.3.21 setTourAttente()

```
void Mob::setTourAttente (  
    int tourAttente ) [inline]
```

setteur tourAttente

Author

SINET Theo

Parameters

<i>tourAttente</i>	: le nombre de tour en attente du Mob actualise le nombre de tour en attente du Mob
--------------------	---

6.8.3.22 tirer()

```
virtual char Mob::tirer (  
    int positionJoueur,  
    std::string * tab,  
    int nbLigne,  
    int nbColonne ) [inline], [virtual]
```

tirer une flèche

Author

SINET Theo

Parameters

<i>positionJoueur</i>	: position du joueur
<i>tab</i>	: la map
<i>nbLigne</i>	: nombre de ligne de la map
<i>nbColonne</i>	: nombre de colone de la map

Returns

la direction du tir Methode qui permet au [Bowman](#) de tirer une flèche

Reimplemented in [Bowman](#).

6.8.4 Member Data Documentation

6.8.4.1 damage

```
int Mob::damage [protected]
```

un entier

6.8.4.2 hp

```
int Mob::hp [protected]
```

un entier

6.8.4.3 mort

```
bool Mob::mort [protected]
```

un booleen

6.8.4.4 position

```
int Mob::position [protected]
```

un entier

6.8.4.5 speed

```
int Mob::speed [protected]
```

un entier

6.8.4.6 tourAttente

```
int Mob::tourAttente [protected]
```

un entier

The documentation for this class was generated from the following files:

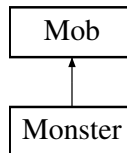
- [Mob.h](#)
- [Mob.cpp](#)

6.9 Monster Class Reference

classe representant un [Monster](#), hérite de [Mob](#)

```
#include <Monster.h>
```

Inheritance diagram for Monster:



Public Member Functions

- [Monster](#) ()
constructeur par défaut
- bool [isBowman](#) ()
est un [Bowman](#)

Additional Inherited Members

6.9.1 Detailed Description

classe representant un [Monster](#), hérite de [Mob](#)

Author

SINET Theo La classe gere l'objet [Monster](#)

6.9.2 Constructor & Destructor Documentation

6.9.2.1 Monster()

```
Monster::Monster ( )
```

constructeur par défaut

Author

SINET Theo Constructeur par défaut de la classe [Monster](#)

6.9.3 Member Function Documentation

6.9.3.1 isBowman()

```
bool Monster::isBowman ( ) [inline], [virtual]
```

est un [Bowman](#)

Author

SINET Theo

Returns

false Methode qui retourne false lorsque elle est appelée

Reimplemented from [Mob](#).

The documentation for this class was generated from the following files:

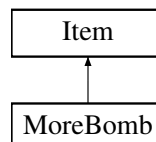
- [Monster.h](#)
- [Monster.cpp](#)

6.10 MoreBomb Class Reference

classe representant l'item [MoreBomb](#), hérite de [Item](#)

```
#include <MoreBomb.h>
```

Inheritance diagram for MoreBomb:



Public Member Functions

- [MoreBomb](#) ()
constructeur par défaut et initialisation
- void [ajouterBombe](#) ([Joueur](#) *joueur)
ajouterBombe

6.10.1 Detailed Description

classe representant l'item [MoreBomb](#), hérite de [Item](#)

Author

SINET Theo

La classe gere l'objet [MoreBomb](#)

6.10.2 Constructor & Destructor Documentation

6.10.2.1 MoreBomb()

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

constructeur par défaut et initialisation

Author

SINET Theo Constructeur par défaut et initialisation de l'item [MoreBomb](#)

6.10.3 Member Function Documentation

6.10.3.1 ajouterBombe()

```
void MoreBomb::ajouterBombe (
    Joueur * joueur ) [virtual]
```

ajouterBombe

Author

SINET Theo

Parameters

<i>joueur</i>	: un joueur permet d'ajouter une bombe supplémentaire au joueur
---------------	---

Reimplemented from [Item](#).

The documentation for this class was generated from the following files:

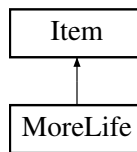
- [MoreBomb.h](#)
- [MoreBomb.cpp](#)

6.11 MoreLife Class Reference

classe representant l'item [MoreLife](#), hérite de [Item](#)

```
#include <MoreLife.h>
```


Inheritance diagram for MoreLife:



Public Member Functions

- [MoreLife](#) ()
constructeur par défaut et initialisation
- void [ajouterVie](#) ([Joueur](#) *joueur)
ajouterVie

6.11.1 Detailed Description

classe representant l'item [MoreLife](#), hérite de [Item](#)

Author

SINET Theo

La classe gere l'objet [MoreLife](#)

6.11.2 Constructor & Destructor Documentation

6.11.2.1 MoreLife()

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

constructeur par défaut et initialisation

Author

SINET Theo Constructeur par défaut et initialisation de l'item [MoreLife](#)

6.11.3 Member Function Documentation

6.11.3.1 ajouterVie()

```
void MoreLife::ajouterVie (  
    Joueur * joueur ) [virtual]
```

ajouterVie

Author

SINET Theo

Parameters

<code>joueur</code>	: un joueur permet d'ajouter de la vie au joueur
---------------------	--

Reimplemented from [Item](#).

The documentation for this class was generated from the following files:

- [MoreLife.h](#)
- [MoreLife.cpp](#)

6.12 Objectif Class Reference

classe representant l'[Objectif](#)

```
#include <Objectif.h>
```

Public Member Functions

- [Objectif](#) (int hp=2, int position=-1, bool sortieValide=false)
constructeur par default et initialisation
- int [getHP](#) ()
getteur hp
- int [getPosition](#) ()
getteur position
- bool [getSortieValide](#) ()
getteur sortieValide
- void [setHP](#) (int hp)
setteur hp
- void [setPosition](#) (int position)
setteur position
- void [setSortieValide](#) (bool sortieValide)
setteur sortieValide

6.12.1 Detailed Description

classe representant l'[Objectif](#)

Author

SINET Theo

La classe gere l'objet [Objectif](#)

6.12.2 Constructor & Destructor Documentation

6.12.2.1 Objectif()

```
Objectif::Objectif (
    int hp = 2,
    int position = -1,
    bool sortieValide = false )
```

constructeur par défaut et initialisation

Author

SINET Theo

Parameters

<i>hp</i>	: hp de l' Objectif
<i>position</i>	: porter de l' Objectif
<i>sortieValide</i>	: si la sortie est valide ou non Constructeur par défaut et initialisation de la classe Objectif

6.12.3 Member Function Documentation

6.12.3.1 getHP()

```
int Objectif::getHP ( ) [inline]
```

getteur hp

Author

SINET Theo

Returns

hp de l'[Objectif](#) récupère les hp de l'[Objectif](#)

6.12.3.2 getPosition()

```
int Objectif::getPosition ( ) [inline]
```

getteur position

Author

SINET Theo

Returns

position de l'[Objectif](#) récupère la position de l'[Objectif](#)

6.12.3.3 `getSortieValide()`

```
bool Objectif::getSortieValide ( ) [inline]
```

getteur sortieValide

Author

SINET Theo

Returns

true si la sortie est valide, false sinon récupère si la sortie de l'[Objectif](#) est valide

6.12.3.4 `setHP()`

```
void Objectif::setHP (
    int hp ) [inline]
```

setteur hp

Author

SINET Theo

Parameters

<i>hp</i>	: hp change les hp de l' Objectif
-----------	---

6.12.3.5 `setPosition()`

```
void Objectif::setPosition (
    int position ) [inline]
```

setteur position

Author

SINET Theo

Parameters

<i>position</i>	: position change la position de l' Objectif
-----------------	--

6.12.3.6 setSortieValide()

```
void Objectif::setSortieValide (
    bool sortieValide ) [inline]
```

setteur sortieValide

Author

SINET Theo

Parameters

<i>sortieValide</i>	: sortieValide change si la sortie de l' Objectif est valide ou non
---------------------	---

The documentation for this class was generated from the following files:

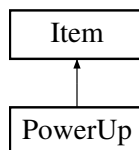
- [Objectif.h](#)
- [Objectif.cpp](#)

6.13 PowerUp Class Reference

classe representant l'item [PowerUp](#), hérite de [Item](#)

```
#include <PowerUp.h>
```

Inheritance diagram for PowerUp:



Public Member Functions

- [PowerUp](#) ()
constructeur par défaut et initialisation
- void [augmenterPuissance](#) ([Joueur](#) *joueur)
augmenterPuissance

6.13.1 Detailed Description

classe representant l'item [PowerUp](#), hérite de [Item](#)

Author

SINET Theo

La classe gere l'objet [PowerUp](#)

6.13.2 Constructor & Destructor Documentation

6.13.2.1 PowerUp()

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

constructeur par défaut et initialisation

Author

SINET Theo Constructeur par défaut et initialisation de l'item [PowerUp](#)

6.13.3 Member Function Documentation

6.13.3.1 augmenterPuissance()

```
void PowerUp::augmenterPuissance (
    Joueur * joueur ) [virtual]
```

augmenterPuissance

Author

SINET Theo

Parameters

<i>joueur</i>	: un joueur permet d'augmenter la puissance des bombes du joueur
---------------	--

Reimplemented from [Item](#).

The documentation for this class was generated from the following files:

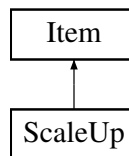
- [PowerUp.h](#)
- [PowerUp.cpp](#)

6.14 ScaleUp Class Reference

classe representant l'item [ScaleUp](#), hérite de [Item](#)

```
#include <ScaleUp.h>
```

Inheritance diagram for ScaleUp:



Public Member Functions

- [ScaleUp](#) ()
constructeur par défaut et initialisation
- void [augmenterPortee](#) ([Joueur](#) *joueur)
augmenterPortee

6.14.1 Detailed Description

classe representant l'item [ScaleUp](#), hérite de [Item](#)

Author

SINET Theo

La classe gere l'objet [ScaleUp](#)

6.14.2 Constructor & Destructor Documentation

6.14.2.1 ScaleUp()

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

constructeur par défaut et initialisation

Author

SINET Theo Constructeur par défaut et initialisation de l'item [ScaleUp](#)

6.14.3 Member Function Documentation

6.14.3.1 augmenterPortee()

```
void ScaleUp::augmenterPortee (
    Joueur * joueur ) [virtual]
```

augmenterPortee

Author

SINET Theo

Parameters

<i>joueur</i>	: un joueur permet d'augmenter la porter de la bombe du joueur
---------------	--

Reimplemented from [Item](#).

The documentation for this class was generated from the following files:

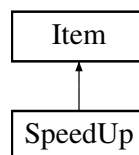
- [ScaleUp.h](#)
- [ScaleUp.cpp](#)

6.15 SpeedUp Class Reference

classe representant l'item [SpeedUp](#), hérite de [Item](#)

```
#include <SpeedUp.h>
```

Inheritance diagram for SpeedUp:



Public Member Functions

- [SpeedUp](#) ()
constructeur par défaut et initialisation
- void [augmenterVitesse](#) ([Joueur](#) *joueur)
augmenterVitesse

6.15.1 Detailed Description

classe representant l'item [SpeedUp](#), hérite de [Item](#)

Author

SINET Theo

La classe gere l'objet [SpeedUp](#)

6.15.2 Constructor & Destructor Documentation

6.15.2.1 SpeedUp()

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

constructeur par défaut et initialisation

Author

SINET Theo Constructeur par défaut et initialisation de l'item [SpeedUp](#)

6.15.3 Member Function Documentation

6.15.3.1 augmenterVitesse()

```
void SpeedUp::augmenterVitesse (
    Joueur * joueur ) [virtual]
```

augmenterVitesse

Author

SINET Theo

Parameters

<i>joueur</i>	: un joueur permet d'augmenter le nombre de déplacement du joueur
---------------	---

Reimplemented from [Item](#).

The documentation for this class was generated from the following files:

- [SpeedUp.h](#)
- [SpeedUp.cpp](#)

6.16 System Class Reference

classe representant le [System](#)

```
#include <System.h>
```

Public Member Functions

- [System](#) ([Map](#) *map=new [Map](#)(), [Joueur](#) *joueur=new [Joueur](#)(), [Mob](#) **mob=new [Mob](#) *(), [Item](#) **item=new [Item](#) *(), int nbMobMap=0, [Wall](#) **wall=new [Wall](#) *())
constructeur par défaut et initialisation
- void [afficherCarte](#) (int nbHP, int puissanceBombe, int nbBombMax, int speed, int portee)
afficherCarte
- bool [demanderAction](#) ()
demanderAction
- void [deplacerMonstres](#) (int numMob)
deplacerMonstres
- void [exploserBombe](#) ([Bombe](#) *bombe)
exploserBombe
- void [enleverExplosionBombe](#) (int position)
enleverExplosionBombe
- void [resetBomb](#) ([Bombe](#) *bombe)
resetBomb
- void [actualiserCarte](#) ()
actualiser la map
- bool [verifFuturPosition](#) (int position, char deplacement, int futurPosition, std::string lettreJoueur)
verifie si la futur position de l'entité est bonne
- int [nombreMobMap](#) ()
nombreMobMap
- void [alloueMob](#) ()
alloueMob
- int [nombreItemMap](#) ()
nombreItemMap
- void [alloueItem](#) ()
alloueItem
- void [alloueBomb](#) ()
alloueBomb
- void [lirePositionJoueur](#) ()
lirePositionJoueur
- void [jouer](#) (std::string nomFichier)
jouer
- [Map](#) * [getMap](#) ()
getteur getMap
- [Mob](#) ** [getMob](#) ()
getteur getMob
- [Joueur](#) * [getJoueur](#) ()
getteur joueur
- int [getNbMobMap](#) ()
getteur nbMobMap
- void [setNbMobMap](#) (int nbMobMap)
setteur nbMobMap

6.16.1 Detailed Description

classe representant le [System](#)

Author

SINET Theo

La classe gere l'objet [System](#)

6.16.2 Constructor & Destructor Documentation

6.16.2.1 System()

```
System::System (
    Map * map = new Map(),
    Joueur * joueur = new Joueur(),
    Mob ** mob = new Mob*(),
    Item ** item = new Item*(),
    int nbMobMap = 0,
    Wall ** wall = new Wall*() )
```

constructeur par défaut et initialisation

Author

SINET Theo

Parameters

<i>map</i>	: map
<i>joueur</i>	: joueur
<i>mob</i>	: mob
<i>item</i>	: item
<i>nbMobMap</i>	: nombre de mob
<i>wall</i>	: wall Constructeur par défaut et initialisation du joueur

6.16.3 Member Function Documentation

6.16.3.1 actualiserCarte()

```
void System::actualiserCarte ( )
```

actualiser la map

Author

DARVILLE Killian appelle la fonction pour actualiser le fichier map.txt

6.16.3.2 afficherCarte()

```
void System::afficherCarte (
    int  nbHP,
    int  puissanceBombe,
    int  nbBombMax,
    int  speed,
    int  portee )
```

afficherCarte

Author

DARVILLE Killian, SINET Theo permet d'appeler la fonction d'affichage de la map

6.16.3.3 alloueBomb()

```
void System::alloueBomb ( )
```

alloueBomb

Author

SINET Theo créer le tableau de bombe

6.16.3.4 alloueItem()

```
void System::alloueItem ( )
```

alloueItem

Author

SINET Theo créer le tableau d'item

6.16.3.5 alloueMob()

```
void System::alloueMob ( )
```

alloueMob

Author

SINET Theo créer le tableau de [Mob](#)

6.16.3.6 demanderAction()

```
bool System::demanderAction ( )
```

demanderAction

Author

DARVILLE Killian, SINET Theo

Returns

true si le joueur veux quitter la partie Demande au joueur dans quel direction il veux aller

6.16.3.7 deplacerMonstres()

```
void System::deplacerMonstres (
    int numMob )
```

deplacerMonstres

Author

SINET Theo

Parameters

<i>numMob</i>	: numéro du Mob qui va se dplacer Déplace les monstres
---------------	--

6.16.3.8 enleverExplosionBombe()

```
void System::enleverExplosionBombe (
    int position )
```

enleverExplosionBombe

Author

DARVILLE Killian

Parameters

<i>position</i>	: position de la bombe retire l'explose de la bombe
-----------------	---

6.16.3.9 exploserBombe()

```
void System::exploserBombe (
    Bombe * bombe )
```

exploserBombe

Author

SINET Theo

Parameters

<i>bombe</i>	: une bombe explose la bombe si le détonateur est à 0
--------------	---

6.16.3.10 getJoueur()

```
Joueur * System::getJoueur ( ) [inline]
```

getteur joueur

Author

SINET Theo

Returns

le joueur retourne le joueur

6.16.3.11 getMap()

```
Map * System::getMap ( ) [inline]
```

getteur getMap

Author

SINET Theo

Returns

la map retourne la map

6.16.3.12 getMob()

```
Mob ** System::getMob ( ) [inline]
```

getteur getMob

Author

SINET Theo

Returns

le tableau de [Mob](#) retourne le tableau de [Mob](#)

6.16.3.13 getNbMobMap()

```
int System::getNbMobMap ( ) [inline]
```

getteur nbMobMap

Author

SINET Theo

Returns

le nombre de [Mob](#) sur la map retourne le nombre de [Mob](#) sur la map

6.16.3.14 jouer()

```
void System::jouer (
    std::string nomFichier )
```

jouer

Author

SINET Theo

Parameters

<i>nomFichier</i>	le nom du fichier de la map Lance le jeu
-------------------	--

6.16.3.15 lirePositionJoueur()

```
void System::lirePositionJoueur ( )
```

lirePositionJoueur

Author

SINET Theo Lis la position du joueur sur la map

6.16.3.16 nombreItemMap()

```
int System::nombreItemMap ( )
```

nombreItemMap

Author

SINET Theo

Returns

le nombre d'item sur la map calcule le nombre d'item sur la map

6.16.3.17 nombreMobMap()

```
int System::nombreMobMap ( )
```

nombreMobMap

Author

SINET Theo

Returns

le nombre de mob calcule le nombre de [Mob](#) sur la map

6.16.3.18 resetBomb()

```
void System::resetBomb (
    Bombe * bombe )
```

resetBomb

Author

SINET Theo

Parameters

<i>bombe</i>	: bombe redonne une bombe au joueur
--------------	-------------------------------------

6.16.3.19 setNbMobMap()

```
void System::setNbMobMap (
    int nbMobMap ) [inline]
```

setteur nbMobMap

Author

SINET Theo

Parameters

<i>nbMobMap</i>	le nombre de Mob sur la map change le nombre de Mob sur la map
-----------------	--

6.16.3.20 verifFuturPosition()

```
bool System::verifFuturPosition (
    int position,
    char deplacement,
    int futurPosition,
    std::string lettreJoueur )
```

verifie si la futur position de l'entité est bonne

Author

DARVILLE Killian

Parameters

<i>position</i>	: position de l'entité
<i>deplacement</i>	: deplacement de l'entité
<i>futurPosition</i>	: futurPosition de l'entité
<i>lettreJoueur</i>	: lettre de l'entité vérifie si l'entité peut se déplacer

The documentation for this class was generated from the following files:

- [System.h](#)
- [System.cpp](#)

6.17 Wall Class Reference

classe representant le [Wall](#)

```
#include <Wall.h>
```

Public Member Functions

- [Wall](#) (int hp=2, int position=-1)
constructeur par default et initialisation
- int [getHP](#) ()
getteur hp
- int [getPosition](#) ()
getteur hp
- void [setHP](#) (int hp)
setteur hp
- void [setPosition](#) (int position)
setteur position

6.17.1 Detailed Description

classe representant le [Wall](#)

Author

SINET Theo

La classe gere l'objet [Wall](#)

6.17.2 Constructor & Destructor Documentation

6.17.2.1 Wall()

```
Wall::Wall (
    int hp = 2,
    int position = -1 )
```

constructeur par default et initialisation

Author

SINET Theo

Parameters

<i>hp</i>	: hp du Wall
<i>position</i>	: position du Wall Constructeur par default et initialisation de la classe Wall

6.17.3 Member Function Documentation

6.17.3.1 getHP()

```
int Wall::getHP ( ) [inline]
```

getteur hp

Author

SINET Theo

Returns

les hp du [Wall](#) récupère les hp du [Wall](#)

6.17.3.2 getPosition()

```
int Wall::getPosition ( ) [inline]
```

getteur hp

Author

SINET Theo

Returns

la position du [Wall](#) récupère la position du [Wall](#)

6.17.3.3 setHP()

```
void Wall::setHP (
    int hp ) [inline]
```

setteur hp

Author

SINET Theo

Parameters

<i>hp</i>	: hp change les hp du wall
-----------	----------------------------

6.17.3.4 setPosition()

```
void Wall::setPosition (  
    int position ) [inline]
```

setteur position

Author

SINET Theo

Parameters

<i>position</i>	: position change la position du wall
-----------------	---------------------------------------

The documentation for this class was generated from the following files:

- [Wall.h](#)
- [Wall.cpp](#)

Chapter 7

File Documentation

7.1 Arrow.h File Reference

Classe de la flèche.

Classes

- class [Arrow](#)
classe representant la [Arrow](#)

7.1.1 Detailed Description

Classe de la flèche.

Author

SINET Theo

Version

0.1

Date

09 avril 2022

7.2 Arrow.h

[Go to the documentation of this file.](#)

```
1 #ifndef __ARROW_H__
2 #define __ARROW_H__
19 class Arrow
20 {
21     private:
22         int degat;
23         int position;
24         bool touche;
25         char direction;
26     public:
36         Arrow(int degat=1, int position=-1, char direction=' ', bool touche=false);
43         void seDeplacer(int déplacement);
44
51         int getDegat() {return this->degat;}
58         int getPosition() {return this->position;}
65         char getDirection() {return this->direction;}
72         bool getTouche() {return this->touche;}
73
80         void setDegat(int degat) {this->degat = degat;}
87         void setPosition(int position) {this->position = position;}
94         void setDirection(char direction) {this->direction = direction;}
101         void setTouche(bool touche) {this->touche = touche;}
102 };
103
104
105 #endif
```

7.3 Bombe.h File Reference

Classe de la bombe.

Classes

- class [Bombe](#)
classe representant la bombe

Namespaces

- namespace [std](#)

7.3.1 Detailed Description

Classe de la bombe.

Author

SINET Theo

Version

0.1

Date

09 avril 2022

7.4 Bombe.h

[Go to the documentation of this file.](#)

```

1 #ifndef __BOMBE_H__
2 #define __BOMBE_H__
11 // #include <string>
12 // #include <iostream>
13 // #include "Mob.h"
14 // #include "Joueur.h"
15
19 using namespace std;
20
28 class Bombe
29 {
30     private:
31         int puissance;
32         int portee;
33         int position;
34         int tourExplosion;
37     public:
47         Bombe(int puissance=1, int portee=2, int position=-1, int tourExplosion=0);
54         Bombe(const Bombe& b1);
55
61         void toucherPersonnage();
62
69         int getPuissance(){return puissance;}
76         int getPortee(){return portee;}
83         int getPosition(){return position;}
90         int getTourExplosion(){return tourExplosion;}
91
98         void setPuissance(int puissance){this->puissance = puissance;}
105        void setPortee(int portee){this->portee = portee;}
112        void setPosition(int position){this->position = position;}
119        void setTourExplosion(int tourExplosion){this->tourExplosion = tourExplosion;}
120 };
121
122
123 #endif

```

7.5 Bowman.h File Reference

Classe du [Bowman](#).

```

#include "Mob.h"
#include "Arrow.h"

```

Classes

- class [Bowman](#)
classe representant le [Bowman](#), hérite de [Mob](#)

7.5.1 Detailed Description

Classe du [Bowman](#).

Author

SINET Theo

Version

0.1

Date

09 avril 2022

7.6 Bowman.h

[Go to the documentation of this file.](#)

```
1 #ifndef __BOWMAN_H__
2 #define __BOWMAN_H__
3
11 #include "Mob.h"
12 #include "Arrow.h"
13
20 class Bowman : public Mob
21 {
22     private:
23         int nbFleche;
24         Arrow* arrow;
25         bool verifObstacle(std::string* tab, int i);
26
27     public:
28         Bowman(int nbFleche=1, Arrow* arrow = new Arrow());
29         bool isBowman(){return true;}
30         bool peutTirer(int positionJoueur, std::string* tab, int nbLigne, int nbColonne);
31         char tirer(int positionJoueur, std::string* tab, int nbLigne, int nbColonne);
32
33         int getNbFleche(){return nbFleche;}
34         Arrow* getArrow(){return arrow;}
35
36         void setNbFleche(int nbFleche){this->nbFleche = nbFleche;}
37 };
38
39 #endif
```

7.7 Ghost.h File Reference

Classe du ghost.

```
#include "Mob.h"
```

Classes

- class [Ghost](#)

classe representant le [Ghost](#), hérite de [Mob](#)

7.7.1 Detailed Description

Classe du ghost.

Author

SINET Theo

Version

0.1

Date

09 avril 2022

7.8 Ghost.h

[Go to the documentation of this file.](#)

```
1 #ifndef __GHOST_H__
2 #define __GHOST_H__
3
11 #include "Mob.h"
12
19 class Ghost : public Mob
20 {
21     public:
27     Ghost(/*int hp, int speed, int damage, int position, int tourAttente*/);
34     bool isBowman(){return false;}
35
36 };
37
38
39 #endif
```

7.9 Item.h File Reference

Classe de l'[Item](#).

```
#include "Joueur.h"
#include "Bombe.h"
```

Classes

- class [Item](#)
classe representant l'[Item](#)

7.9.1 Detailed Description

Classe de l'[Item](#).

Author

SINET Theo

Version

0.1

Date

09 avril 2022

7.10 Item.h

[Go to the documentation of this file.](#)

```
1 #ifndef __ITEM_H__
2 #define __ITEM_H__
3
11 #include "Joueur.h"
12 #include "Bombe.h"
13
20 class Item
21 {
22     private:
23         int position;
24     public:
32         Item(int position=-1);
33
40         int getPosition(){return this->position;}
47         void setPosition(int position){this->position = position;}
48
55         virtual void ajouterVie(Joueur* joueur){}
62         virtual void ajouterBombe(Joueur* joueur){}
69         virtual void augmenterVitesse(Joueur* joueur){}
76         virtual void augmenterPuissance(Joueur* joueur){}
83         virtual void augmenterPortee(Joueur* joueur){}
84 };
85
86 #endif
```

7.11 Joueur.h File Reference

Classe du [Joueur](#).

```
#include "Bombe.h"
```

Classes

- class [Joueur](#)
classe representant la [Joueur](#)

7.11.1 Detailed Description

Classe du [Joueur](#).

Author

DARVILLE Killian, SINET Theo

Version

0.1

Date

09 avril 2022

7.12 Joueur.h

[Go to the documentation of this file.](#)

```

1  #ifndef __JOUEUR_H__
2  #define __JOUEUR_H__
3
11 #include "Bombe.h"
12
20 class Joueur
21 {
22     private:
23         int hp;
24         int speed;
25         int position;
26         int nbBombMax;
27         int nbBombRestante;
28         Bombe *bombe;
29
30     public:
42         Joueur(int hp=3, int speed=1, int position=50, int nbBombMax=1, int nbBombRestante=1, Bombe*
bombe = new Bombe());
49         Joueur(const Joueur& j1);
50
56         ~Joueur() { /*delete bombe;*/ }
57
63         void seDeplacer(char direction, int nbColonne);
69         void poserBombe();
70
77         int getHP() {return hp;}
84         int getSpeed() {return speed;}
91         int getPosition() {return position;}
98         int getNbBombMax() {return nbBombMax;}
105        int getNbBombRestante() {return nbBombRestante;}
112        Bombe* getBombe() {return bombe;}
113
120        void setHP(int hp) {this->hp = hp;}
127        void setSpeed(int speed) {this->speed = speed;}
134        void setPosition(int position) {this->position = position;}
141        void setNbBombMax(int nbBombMax) {this->nbBombMax = nbBombMax;}
148        void setNbBombRestante(int nbBombRestante) {this->nbBombRestante = nbBombRestante;}
155        void setBombe(Bombe* b1) {this->bombe = b1;}
156 };
157
158
159 #endif

```

7.13 main.cpp File Reference

main du programme

```

#include <iostream>
#include <fstream>
#include <chrono>
#include <thread>
#include "System.h"
#include "Joueur.h"
#include "Bombe.h"
#include "Mob.h"
#include "Map.h"

```

Namespaces

- namespace `std`
- namespace `std::chrono_literals`

Functions

- int `menuGeneral` ()
menuGeneral
- void `affichageRegles` ()
affichageRegles
- void `affichageLegende` ()
affichageRegles
- int `main` ()
main

7.13.1 Detailed Description

main du programme

Author

DARVILLE Killian, SINET Theo

Version

0.1

Date

09 avril 2022

7.13.2 Function Documentation

7.13.2.1 `affichageLegende()`

```
void affichageLegende ( )
```

`affichageRegles`

Author

DARVILLE Killian, SINET Theo Affiche les règles

7.13.2.2 `affichageRegles()`

```
void affichageRegles ( )
```

`affichageRegles`

Author

SINET Theo Ouvre le fichier regles.txt et les affiches

7.13.2.3 `main()`

```
int main ( )
```

`main`

Author

DARVILLE Killian, SINET Theo

Returns

0 pour quitter le programme lance le programme

7.13.2.4 `menuGeneral()`

```
int menuGeneral ( )
```

`menuGeneral`

Author

SINET Theo

Returns

le choix du joueur Demande au joueur ce qu'il veut faire lorsque il lance le programme

7.14 Map.h File Reference

Classe de la map.

```
#include <fstream>
#include "Wall.h"
#include "Objectif.h"
```

Classes

- class [Map](#)
classe representant la [Map](#)

7.14.1 Detailed Description

Classe de la map.

Author

DARVILLE Killian, SINET Theo

Version

0.1

Date

09 avril 2022

7.15 Map.h

[Go to the documentation of this file.](#)

```

1 #ifndef __MAP_H__
2 #define __MAP_H__
3
11 #include <fstream>
12 #include "Wall.h"
13 #include "Objectif.h"
14
22 class Map
23 {
24     private:
25         int nbLigne;
26         int nbColonne;
27         int nbCaractere;
28         std::string *tab;
29         std::string nomFichier;
30         bool emoji;
31         std::fstream monFlux;
32         int porteePossibleBombe[4];
33         Wall** wall;
34         int nbWallMap;
35         Objectif* objectif;
36         int recupPorteeMaxBombe(int pos, char direction, int porterDeLaBombeGetteur);
37         int getLigne(int positionPerso, int nbLigne, int nbColonne);
38         int getColonne(int positionPerso, int nbColonne);
39         bool flecheBloquer(int pos, char deplacement);
40         void retirerFleche(int pos);
41
42     public:
43         Map(int nbLigne=0, int nbColonne=0, int nbCaractere=0, std::string* tab=new std::string(),
44             std::string nomFichier="maptest.txt", bool emoji=false, Wall** wall = new Wall*(), int nbWallMap=0,
45             Objectif* objectif = new Objectif());
46
104         void affichageMap(int nbHP, int puissanceBombe, int nbBombMax, int speed, int portee);
110         void actualiserMap();
116         void lectureMap();
122         void infoMap();
130         void afficherPersoObjet(int pos, char lettre);
138         void gestionDeplacementMap(int pos, int x); // pos : la position du personnage ; x : la valeur
139         liée au déplacement du personnage
147         void explosionBombe(int pos, int porteeBombe);
155         void retirerExplosionBombe(int positionBombe);
163         bool verifJoueurBloque(int position);

```

```

171         void deplacementFleche(int pos, char direction);
172         int compteMurs();
173         void alloueMurs();
174         void positionObjectif();
175
176         int getNbLigne() {return nbLigne;}
177         int getNbColonne() {return nbColonne;}
178         int getNbCaractere() {return nbCaractere;}
179         std::string* getTab() {return tab;}
180         std::string getNomFichier() {return nomFichier;}
181         bool getEmoji() {return emoji;}
182
183         void setNbLigne(int nbLigne) {this->nbLigne=nbLigne;}
184         void setNbColonne(int nbColonne) {this->nbColonne=nbColonne;}
185         void setNbCaractere(int nbCaractere) {this->nbCaractere=nbCaractere;}
186         void setTab(std::string* tab) {this->tab=tab;}
187         void setNomFichier(std::string nomFichier) {this->nomFichier=nomFichier;}
188         void setEmoji(bool emoji) {this->emoji = emoji;}
189         int* getPorteePossibleBombe() {return porteePossibleBombe;}
190         Wall** getWall() {return wall;}
191         int getNbWallMap() {return nbWallMap;}
192         Objectif* getObjectif() {return objectif;}
193     };
194 #endif

```

7.16 Mob.h File Reference

Classe du [Mob](#).

```

#include <iostream>
#include <string>
#include "Arrow.h"

```

Classes

- class [Mob](#)
classe representant le [Mob](#)

7.16.1 Detailed Description

Classe du [Mob](#).

Author

SINET Theo

Version

0.1

Date

09 avril 2022

7.17 Mob.h

[Go to the documentation of this file.](#)

```

1  #ifndef __MOB_H__
2  #define __MOB_H__
3
11 #include <iostream>
12 #include <string>
13 #include "Arrow.h"
14
22 class Mob
23 {
24     protected:
25         int hp;
26         int speed;
27         int damage;
28         int position;
29         int tourAttente;
30         bool mort;
41         int getLigne(int positionPerso, int nbLigne, int nbColonne);
50         int getColonne(int positionPerso, int nbColonne);
51
52     public:
63         Mob(int hp=1, int speed=1, int damage=1, int position=42, int tourAttente=0, bool mort=false);
70         Mob(const Mob& ml);
71
77         void seDeplacer(int positionJoueur, int nbLigne, int nbColonne);
84         int attaquer(int hpJoueur);
91         virtual bool isBowman() {return false;}
101        virtual bool peutTirer(int positionJoueur, std::string* tab, int nbLigne, int nbColonne){return
false;}
112        virtual char tirer(int positionJoueur, std::string* tab, int nbLigne, int nbColonne){return '
';}
119        virtual int getNbFleche() {return 0;}
126        virtual Arrow* getArrow() {return NULL;}
133        virtual void setNbFleche(int nbFleche){}
134
141        int getHP() {return hp;}
148        int getSpeed() {return speed;}
155        int getDamage() {return damage;}
162        int getPosition() {return position;}
169        int getTourAttente() {return tourAttente;}
176        bool getMort() {return mort;}
177
184        void setHP(int hp){this->hp = hp;}
191        void setSpeed(int speed){this->speed = speed;}
198        void setDamage(int damage){this->damage = damage;}
205        void setPosition(int position){this->position = position;}
212        void setTourAttente(int tourAttente){this->tourAttente = tourAttente;}
219        void setMort(bool mort){this->mort=mort;}
220 };
221
222 #endif

```

7.18 Monster.h File Reference

Classe du [Monster](#).

```
#include "Mob.h"
```

Classes

- class [Monster](#)

classe representant un [Monster](#), hérite de [Mob](#)

7.18.1 Detailed Description

Classe du [Monster](#).

Author

SINET Theo

Version

0.1

Date

09 avril 2022

7.19 Monster.h

[Go to the documentation of this file.](#)

```
1 #ifndef __MONSTER_H__
2 #define __MONSTER_H__
3
11 #include "Mob.h"
12
19 class Monster : public Mob
20 {
21     public:
27     Monster();
34     bool isBowman() {return false;}
35 };
36
37
38 #endif
```

7.20 MoreBomb.h File Reference

Classe de l'item [MoreBomb](#).

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

Classes

- class [MoreBomb](#)
classe representant l'item [MoreBomb](#), hérite de [Item](#)

7.20.1 Detailed Description

Classe de l'item [MoreBomb](#).

Author

SINET Theo

Version

0.1

Date

09 avril 2022

7.21 MoreBomb.h

[Go to the documentation of this file.](#)

```
1 #ifndef __MOREBOMB_H__
2 #define __MOREBOMB_H__
3
11 #include "Item.h"
12
20 class MoreBomb : public Item
21 {
22     public:
28     MoreBomb();
35     void ajouterBombe(Joueur* joueur);
36 };
37
38 #endif
```

7.22 MoreLife.h File Reference

Classe de l'item [MoreLife](#).

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

Classes

- class [MoreLife](#)
classe representant l'item [MoreLife](#), hérite de [Item](#)

7.22.1 Detailed Description

Classe de l'item [MoreLife](#).

Author

SINET Theo

Version

0.1

Date

09 avril 2022

7.23 MoreLife.h

[Go to the documentation of this file.](#)

```
1 #ifndef __MORELIFE_H__
2 #define __MORELIFE_H__
3
11 #include "Item.h"
12
20 class MoreLife : public Item
21 {
22     public:
28     MoreLife();
35     void ajouterVie(Joueur* joueur);
36 };
37
38 #endif
```

7.24 Objectif.h File Reference

Classe de la bombe.

Classes

- class [Objectif](#)
classe representant l'[Objectif](#)

7.24.1 Detailed Description

Classe de la bombe.

Author

SINET Theo

Version

0.1

Date

09 avril 2022

7.25 Objectif.h

[Go to the documentation of this file.](#)

```
1 #ifndef __OBJECTIF__H
2 #define __OBJECTIF__H
18 class Objectif
19 {
20     private:
21         int hp;
22         int position;
23         bool sortieValide;
24     public:
25         Objectif(int hp=2, int position=-1, bool sortieValide=false);
26
27         int getHP() {return hp;}
28         int getPosition() {return position;}
29         bool getSortieValide() {return sortieValide;}
30
31         void setHP(int hp) {this->hp = hp;}
32         void setPosition(int position) {this->position = position;}
33         void setSortieValide(bool sortieValide) {this->sortieValide=sortieValide;}
34 };
35
36 #endif
```

7.26 PowerUp.h File Reference

Classe de l'item [PowerUp](#).

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

Classes

- class [PowerUp](#)

classe representant l'item [PowerUp](#), hérite de [Item](#)

7.26.1 Detailed Description

Classe de l'item [PowerUp](#).

Author

SINET Theo

Version

0.1

Date

09 avril 2022

7.27 PowerUp.h

[Go to the documentation of this file.](#)

```
1 #ifndef __POWERUP_H__
2 #define __POWERUP_H__
3
11 #include "Item.h"
12
20 class PowerUp : public Item
21 {
22     public:
28     PowerUp();
35     void augmenterPuissance(Joueur* joueur);
36 };
37
38
39 #endif
```

7.28 ScaleUp.h File Reference

Classe de l'item [ScaleUp](#).

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

Classes

- class [ScaleUp](#)

classe representant l'item [ScaleUp](#), hérite de [Item](#)

7.28.1 Detailed Description

Classe de l'item [ScaleUp](#).

Author

SINET Theo

Version

0.1

Date

09 avril 2022

7.29 ScaleUp.h

[Go to the documentation of this file.](#)

```
1 #ifndef __SCALEUP_H__
2 #define __SCALEUP_H__
3
11 #include "Item.h"
12
20 class ScaleUp : public Item
21 {
22     public:
28         ScaleUp();
35         void augmenterPortee(Joueur* joueur);
36 };
37
38 #endif
```

7.30 SpeedUp.h File Reference

Classe de l'item [SpeedUp](#).

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

Classes

- class [SpeedUp](#)
classe representant l'item [SpeedUp](#), hérite de [Item](#)

7.30.1 Detailed Description

Classe de l'item [SpeedUp](#).

Author

SINET Theo

Version

0.1

Date

09 avril 2022

7.31 SpeedUp.h

[Go to the documentation of this file.](#)

```
1 #ifndef __SPEEDUP_H__
2 #define __SPEEDUP_H__
3
11 #include "Item.h"
12
20 class SpeedUp : public Item
21 {
22     public:
28     SpeedUp();
35     void augmenterVitesse(Joueur* joueur);
36 };
37
38 #endif
```

7.32 System.h File Reference

Classe du [System](#).

```
#include "Joueur.h"
#include "Map.h"
#include "Mob.h"
#include "Item.h"
#include "Monster.h"
#include "Ghost.h"
#include "Bowman.h"
#include "Arrow.h"
#include "MoreLife.h"
#include "MoreBomb.h"
#include "SpeedUp.h"
#include "PowerUp.h"
#include "ScaleUp.h"
#include "Wall.h"
#include "Objectif.h"
```

Classes

- class [System](#)
classe representant le [System](#)

7.32.1 Detailed Description

Classe du [System](#).

Author

DARVILLE Killian, SINET Theo

Version

0.1

Date

09 avril 2022

7.33 System.h

[Go to the documentation of this file.](#)

```

1  #ifndef __SYSTEM_H__
2  #define __SYSTEM_H__
3
4
12 #include "Joueur.h"
13 #include "Map.h"
14 #include "Mob.h"
15 #include "Item.h"
16 #include "Monster.h"
17 #include "Ghost.h"
18 #include "Bowman.h"
19 #include "Arrow.h"
20 #include "MoreLife.h"
21 #include "MoreBomb.h"
22 #include "SpeedUp.h"
23 #include "PowerUp.h"
24 #include "ScaleUp.h"
25 #include "Wall.h"
26 #include "Objectif.h"
27
35 class System
36 {
37     private:
38         Map* map;
39         Joueur* joueur;
40         Mob** mob;
41         Item** item;
42         int nbMobMap;
43         Wall** wall;
44         Objectif* objectif;
45         bool joueurMort();
46         bool mobMort();
47         void joueurItemFlecheMonstre(int pos);
48         void gestionFlecheMob(int pos);
49         void gestionFlecheJoueur(int pos);
50         void gestionMoreLife(int i, bool trouve);
51         void gestionMoreBomb(int i, bool trouve);
52         void gestionPowerUp(int i, bool trouve);
53         void gestionSpeedUp(int i, bool trouve);
54         void gestionScaleUp(int i, bool trouve);
55         int conversionDeplacementArrow(char direction);
56         char conversionDirectionArrow(char direction);
57         void bombeInfligerDegats(int pos, Bombe* bombe);
58         void mortParBombe(int position);
59         void degatMurs(int position);
60         bool gagner();
61
62     public:
63         System(Map* map = new Map(), Joueur* joueur = new Joueur(), Mob** mob = new Mob*(), Item** item
64             = new Item*(), int nbMobMap=0, Wall** wall=new Wall*());
65
66         void afficherCarte(int nbHP, int puissanceBombe, int nbBombMax, int speed, int portee);
67         bool demanderAction();
68         void deplacerMonstres(int numMob);
69         void exploserBombe(Bombe* bombe);
70         void enleverExplosionBombe(int position);
71         void resetBomb(Bombe* bombe);
72         void actualiserCarte();
73         bool verifFuturPosition(int position, char deplacement, int futurPosition, std::string
74             lettreJoueur);
75         int nombreMobMap();
76         void alloueMob();
77         int nombreItemMap();
78         void alloueItem();
79         void alloueBomb();
80         void lirePositionJoueur();
81         void jouer(std::string nomFichier);
82
83         Map* getMap() {return map;}
84         Mob** getMob() {return mob;}
85         Joueur* getJoueur() {return joueur;}
86         int getNbMobMap() {return nbMobMap;}
87
88         void setNbMobMap(int nbMobMap) {this->nbMobMap=nbMobMap;}
89
90 };
91
92 #endif

```

7.34 Wall.h File Reference

Classe de la bombe.

Classes

- class [Wall](#)
classe representant le [Wall](#)

7.34.1 Detailed Description

Classe de la bombe.

Author

SINET Theo

Version

0.1

Date

09 avril 2022

7.35 Wall.h

[Go to the documentation of this file.](#)

```
1 #ifndef __WALL_H__
2 #define __WALL_H__
18 class Wall
19 {
20     private:
21         int hp;
22         int position;
24     public:
32         Wall(int hp=2, int position=-1);
33
40         int getHP(){return hp;}
47         int getPosition(){return position;}
48
55         void setHP(int hp){this->hp = hp;}
62         void setPosition(int position){this->position = position;}
63
64 };
65
66 #endif
```