

AIBattleSimulator3000

Généré par Doxygen 1.8.13



# Table des matières

<b>1</b>	<b>Index des espaces de nommage</b>	<b>1</b>
1.1	Liste des espaces de nommage . . . . .	1
<b>2</b>	<b>Index hiérarchique</b>	<b>3</b>
2.1	Hiérarchie des classes . . . . .	3
<b>3</b>	<b>Index des classes</b>	<b>5</b>
3.1	Liste des classes . . . . .	5
<b>4</b>	<b>Index des fichiers</b>	<b>7</b>
4.1	Liste des fichiers . . . . .	7
<b>5</b>	<b>Documentation des espaces de nommage</b>	<b>9</b>
5.1	Référence de l'espace de nommage AttackType . . . . .	9
5.1.1	Documentation du type de l'énumération . . . . .	9
5.1.1.1	Type . . . . .	9
5.1.2	Documentation des variables . . . . .	9
5.1.2.1	All . . . . .	10
5.2	Référence de l'espace de nommage Direction . . . . .	10
5.2.1	Documentation du type de l'énumération . . . . .	10
5.2.1.1	Dir . . . . .	10
5.2.2	Documentation des variables . . . . .	10
5.2.2.1	All . . . . .	10
5.3	Référence de l'espace de nommage DommageType . . . . .	10
5.3.1	Documentation du type de l'énumération . . . . .	11
5.3.1.1	Type . . . . .	11
5.3.2	Documentation des variables . . . . .	11
5.3.2.1	All . . . . .	11

<b>6</b>	<b>Documentation des classes</b>	<b>13</b>
6.1	Référence de la structure <code>ArmorComponent</code>	13
6.1.1	Documentation des constructeurs et destructeur	13
6.1.1.1	<code>ArmorComponent()</code>	13
6.1.2	Documentation des fonctions membres	14
6.1.2.1	<code>accept()</code>	14
6.1.3	Documentation des données membres	14
6.1.3.1	<code>magicArmor</code>	14
6.1.3.2	<code>pierceArmor</code>	14
6.1.3.3	<code>shockArmor</code>	14
6.2	Référence de la structure <code>AttackComponent</code>	14
6.2.1	Documentation des constructeurs et destructeur	15
6.2.1.1	<code>AttackComponent()</code>	15
6.2.2	Documentation des fonctions membres	15
6.2.2.1	<code>accept()</code>	15
6.2.3	Documentation des données membres	15
6.2.3.1	<code>amount</code>	15
6.2.3.2	<code>attackType</code>	16
6.2.3.3	<code>dommageType</code>	16
6.3	Référence de la structure <code>Component</code>	16
6.3.1	Documentation des fonctions membres	16
6.3.1.1	<code>accept()</code>	16
6.4	Référence de la classe <code>ComponentStorer</code>	17
6.4.1	Documentation des fonctions membres	17
6.4.1.1	<code>addComponent()</code>	17
6.4.1.2	<code>getComponent()</code>	17
6.4.1.3	<code>getComponentsById()</code>	17
6.4.1.4	<code>getComponentsByType()</code>	17
6.5	Référence de la classe <code>EntityCreator</code>	18
6.5.1	Description détaillée	18

6.5.2	Documentation des constructeurs et destructeur . . . . .	18
6.5.2.1	EntityCreator() . . . . .	18
6.5.3	Documentation des fonctions membres . . . . .	19
6.5.3.1	addEntity() . . . . .	19
6.5.3.2	visit() [1/5] . . . . .	19
6.5.3.3	visit() [2/5] . . . . .	19
6.5.3.4	visit() [3/5] . . . . .	20
6.5.3.5	visit() [4/5] . . . . .	20
6.5.3.6	visit() [5/5] . . . . .	20
6.6	Référence de la classe EntityManager . . . . .	21
6.6.1	Documentation des fonctions membres . . . . .	21
6.6.1.1	getEntitiesId() . . . . .	21
6.6.1.2	getNextId() . . . . .	21
6.6.1.3	removeEntity() . . . . .	21
6.7	Référence de la structure HealthComponent . . . . .	21
6.7.1	Documentation des constructeurs et destructeur . . . . .	22
6.7.1.1	HealthComponent() . . . . .	22
6.7.2	Documentation des fonctions membres . . . . .	22
6.7.2.1	accept() . . . . .	22
6.7.3	Documentation des données membres . . . . .	22
6.7.3.1	health . . . . .	22
6.8	Référence de la classe LuaScript . . . . .	22
6.8.1	Documentation des constructeurs et destructeur . . . . .	23
6.8.1.1	LuaScript() . . . . .	23
6.8.1.2	~LuaScript() . . . . .	23
6.8.2	Documentation des fonctions membres . . . . .	23
6.8.2.1	get() . . . . .	23
6.8.2.2	getKeyValueMap() . . . . .	23
6.9	Référence de la classe MoveSystem . . . . .	24
6.9.1	Documentation des constructeurs et destructeur . . . . .	24

6.9.1.1	MoveSystem()	24
6.9.2	Documentation des fonctions membres	24
6.9.2.1	update()	24
6.9.2.2	visit()	24
6.10	Référence de la structure Position	25
6.10.1	Documentation des fonctions membres	25
6.10.1.1	operator*=( )	25
6.10.1.2	operator++( ) [1/2]	25
6.10.1.3	operator++( ) [2/2]	25
6.10.1.4	operator+=( )	25
6.10.2	Documentation des données membres	26
6.10.2.1	x	26
6.10.2.2	y	26
6.11	Référence de la structure PositionComponent	26
6.11.1	Documentation des constructeurs et destructeur	26
6.11.1.1	PositionComponent()	27
6.11.2	Documentation des fonctions membres	27
6.11.2.1	accept()	27
6.11.3	Documentation des données membres	27
6.11.3.1	position	27
6.12	Référence de la structure SpeedComponent	27
6.12.1	Documentation des constructeurs et destructeur	28
6.12.1.1	SpeedComponent()	28
6.12.2	Documentation des fonctions membres	28
6.12.2.1	accept()	28
6.12.3	Documentation des données membres	28
6.12.3.1	speed	28
6.13	Référence de la classe Visitor	28
6.13.1	Documentation des constructeurs et destructeur	29
6.13.1.1	~Visitor()	29
6.13.2	Documentation des fonctions membres	29
6.13.2.1	visit() [1/5]	29
6.13.2.2	visit() [2/5]	29
6.13.2.3	visit() [3/5]	29
6.13.2.4	visit() [4/5]	30
6.13.2.5	visit() [5/5]	30

<b>7</b>	<b>Documentation des fichiers</b>	<b>31</b>
7.1	Référence du fichier include/Component.hpp . . . . .	31
7.2	Référence du fichier include/ComponentStorer.hpp . . . . .	31
7.3	Référence du fichier include/Define.hpp . . . . .	32
7.3.1	Documentation des fonctions . . . . .	32
7.3.1.1	operator*() . . . . .	32
7.3.1.2	operator+() . . . . .	33
7.3.1.3	operator==() . . . . .	33
7.4	Référence du fichier include/ECS.hpp . . . . .	33
7.5	Référence du fichier include/ECSEntity.hpp . . . . .	33
7.6	Référence du fichier include/EntityCreator.hpp . . . . .	33
7.6.1	Description détaillée . . . . .	34
7.7	Référence du fichier include/EntityManager.hpp . . . . .	34
7.8	Référence du fichier include/LuaScript.hpp . . . . .	34
7.9	Référence du fichier include/MoveSystem.hpp . . . . .	34
7.10	Référence du fichier src/Component.cpp . . . . .	35
7.11	Référence du fichier src/ComponentStorer.cpp . . . . .	35
7.12	Référence du fichier src/Define.cpp . . . . .	35
7.12.1	Documentation des fonctions . . . . .	35
7.12.1.1	operator*() . . . . .	35
7.12.1.2	operator+() . . . . .	35
7.12.1.3	operator==() . . . . .	35
7.13	Référence du fichier src/EntityCreator.cpp . . . . .	36
7.14	Référence du fichier src/EntityManager.cpp . . . . .	36
7.15	Référence du fichier src/LuaScript.cpp . . . . .	36
7.16	Référence du fichier src/main.cpp . . . . .	36
7.16.1	Documentation des fonctions . . . . .	36
7.16.1.1	main() . . . . .	36
7.17	Référence du fichier src/MoveSystem.cpp . . . . .	36
	<b>Index</b>	<b>37</b>





# Chapitre 1

## Index des espaces de nommage

### 1.1 Liste des espaces de nommage

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

<a href="#">AttackType</a>	9
<a href="#">Direction</a>	10
<a href="#">DommageType</a>	10



## Chapitre 2

# Index hiérarchique

### 2.1 Hiérarchie des classes

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

Component . . . . .	16
ArmorComponent . . . . .	13
AttackComponent . . . . .	14
HealthComponent . . . . .	21
PositionComponent . . . . .	26
SpeedComponent . . . . .	27
ComponentStorer . . . . .	17
EntityManager . . . . .	21
LuaScript . . . . .	22
Position . . . . .	25
Visitor . . . . .	28
EntityCreator . . . . .	18
MoveSystem . . . . .	24



## Chapitre 3

# Index des classes

### 3.1 Liste des classes

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

ArmorComponent	13
AttackComponent	14
Component	16
ComponentStorer	17
EntityCreator	
Cette classe a pour but de creer les entites sur demande	18
EntityManager	21
HealthComponent	21
LuaScript	22
MoveSystem	24
Position	25
PositionComponent	26
SpeedComponent	27
Visitor	28



# Chapitre 4

## Index des fichiers

### 4.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

include/Component.hpp	31
include/ComponentStorer.hpp	31
include/Define.hpp	32
include/ECS.hpp	33
include/ECSEntity.hpp	33
include/EntityCreator.hpp	
Ici est declare la class EntityCreator	33
include/EntityManager.hpp	34
include/LuaScript.hpp	34
include/MoveSystem.hpp	34
src/Component.cpp	35
src/ComponentStorer.cpp	35
src/Define.cpp	35
src/EntityCreator.cpp	36
src/EntityManager.cpp	36
src/LuaScript.cpp	36
src/main.cpp	36
src/MoveSystem.cpp	36





## Chapitre 5

# Documentation des espaces de nommage

### 5.1 Référence de l'espace de nommage AttackType

#### Énumérations

— enum `Type` { `Melee`, `Distance`, `NONE` }

#### Variables

— const std : :array< `Type`, 2 > `All` = {`Melee`, `Distance`}

#### 5.1.1 Documentation du type de l'énumération

##### 5.1.1.1 Type

enum `AttackType::Type`

##### Valeurs énumérées

Melee	
Distance	
NONE	

#### 5.1.2 Documentation des variables

### 5.1.2.1 All

```
const std::array<Type, 2> AttackType::All = {Melee, Distance}
```

## 5.2 Référence de l'espace de nommage Direction

### Énumérations

```
— enum Dir {
    Up, Down, Left, Right,
    None }
```

### Variables

```
— const std : :array< Dir, 4 > All = {Up, Down, Left, Right}
```

### 5.2.1 Documentation du type de l'énumération

#### 5.2.1.1 Dir

```
enum Direction::Dir
```

#### Valeurs énumérées

Up	
Down	
Left	
Right	
None	

### 5.2.2 Documentation des variables

#### 5.2.2.1 All

```
const std::array<Dir, 4> Direction::All = {Up, Down, Left, Right}
```

## 5.3 Référence de l'espace de nommage DommageType

### Énumérations

```
— enum Type { Pierce, Shock, Magic, NONE }
```

## Variables

— `const std : :array< Type, 3 > All = {Shock, Pierce, Magic}`

### 5.3.1 Documentation du type de l'énumération

#### 5.3.1.1 Type

```
enum DommageType::Type
```

##### Valeurs énumérées

Pierce	
Shock	
Magic	
NONE	

### 5.3.2 Documentation des variables

#### 5.3.2.1 All

```
const std::array<Type, 3> DommageType::All = {Shock, Pierce, Magic}
```



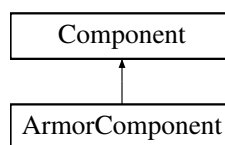
## Chapitre 6

# Documentation des classes

### 6.1 Référence de la structure ArmorComponent

```
#include <Component.hpp>
```

Graphe d'héritage de ArmorComponent :



#### Fonctions membres publiques

- `ArmorComponent` (int p, int s, int m)
- virtual void `accept` (Visitor &v)

#### Attributs publics

- int `pierceArmor`
- int `shockArmor`
- int `magicArmor`

#### 6.1.1 Documentation des constructeurs et destructeur

##### 6.1.1.1 ArmorComponent()

```
ArmorComponent::ArmorComponent (  
    int p,  
    int s,  
    int m ) [inline]
```

## 6.1.2 Documentation des fonctions membres

### 6.1.2.1 accept()

```
void ArmorComponent::accept (
    Visitor & v ) [virtual]
```

Implémente [Component](#).

## 6.1.3 Documentation des données membres

### 6.1.3.1 magicArmor

```
int ArmorComponent::magicArmor
```

### 6.1.3.2 pierceArmor

```
int ArmorComponent::pierceArmor
```

### 6.1.3.3 shockArmor

```
int ArmorComponent::shockArmor
```

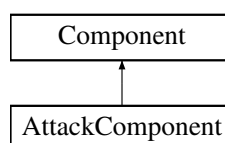
La documentation de cette structure a été générée à partir des fichiers suivants :

- [include/Component.hpp](#)
- [src/Component.cpp](#)

## 6.2 Référence de la structure AttackComponent

```
#include <Component.hpp>
```

Graphe d'héritage de AttackComponent :



### Fonctions membres publiques

- `AttackComponent` (int `a`, `DomDamageType` : :Type `d`, `AttackType` : :Type `at`)
- virtual void `accept` (`Visitor` &`v`)

### Attributs publics

- int `amount`
- `DomDamageType` : :Type `domDamageType`
- `AttackType` : :Type `attackType`

## 6.2.1 Documentation des constructeurs et destructeur

### 6.2.1.1 `AttackComponent()`

```
AttackComponent::AttackComponent (
    int a,
    DomDamageType::Type d,
    AttackType::Type at ) [inline]
```

## 6.2.2 Documentation des fonctions membres

### 6.2.2.1 `accept()`

```
void AttackComponent::accept (
    Visitor & v ) [virtual]
```

Implémente `Component`.

## 6.2.3 Documentation des données membres

### 6.2.3.1 `amount`

```
int AttackComponent::amount
```

### 6.2.3.2 attackType

`AttackType::Type` `AttackComponent::attackType`

### 6.2.3.3 dommageType

`DommageType::Type` `AttackComponent::dommageType`

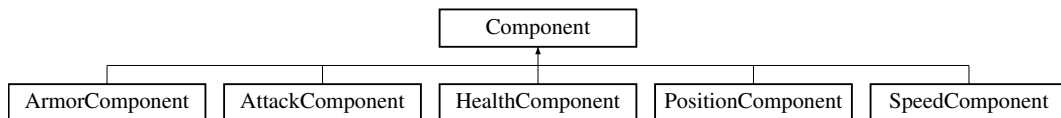
La documentation de cette structure a été générée à partir des fichiers suivants :

- `include/Component.hpp`
- `src/Component.cpp`

## 6.3 Référence de la structure Component

```
#include <Component.hpp>
```

Graphe d'héritage de Component :



### Fonctions membres publiques

- virtual void `accept` (`Visitor` &`v`)=0

### 6.3.1 Documentation des fonctions membres

#### 6.3.1.1 accept()

```
virtual void Component::accept (
    Visitor & v ) [pure virtual]
```

Implémenté dans `ArmorComponent`, `AttackComponent`, `SpeedComponent`, `HealthComponent`, et `PositionComponent`.

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

- `include/Component.hpp`



## 6.4 Référence de la classe ComponentStorer

```
#include <ComponentStorer.hpp>
```

### Fonctions membres publiques

```
— template<typename T >
  std::map< int, std::shared_ptr< Component > > getComponentsByType () const
— template<typename T >
  std::shared_ptr< Component > getComponent (int id) const
— std::map< std::type_index, std::shared_ptr< Component > > getComponentsById (int id) const
— void addComponent (int id, std::shared_ptr< Component > c)
```

### 6.4.1 Documentation des fonctions membres

#### 6.4.1.1 addComponent()

```
void ComponentStorer::addComponent (
    int id,
    std::shared_ptr< Component > c )
```

#### 6.4.1.2 getComponent()

```
template<typename T >
std::shared_ptr<Component> ComponentStorer::getComponent (
    int id ) const [inline]
```

#### 6.4.1.3 getComponentsById()

```
std::map< std::type_index, std::shared_ptr< Component > > ComponentStorer::getComponentsById
(
    int id ) const
```

#### 6.4.1.4 getComponentsByType()

```
template<typename T >
std::map<int, std::shared_ptr<Component> > ComponentStorer::getComponentsByType ( ) const
[inline]
```

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

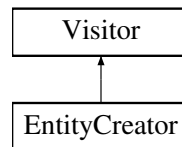
- include/ComponentStorer.hpp
- src/ComponentStorer.cpp

## 6.5 Référence de la classe EntityCreator

Cette classe a pour but de creer les entites sur demande.

```
#include <EntityCreator.hpp>
```

Graphe d'héritage de EntityCreator :



### Fonctions membres publiques

- [EntityCreator](#) ([ComponentStorer](#) &compS, [EntityManager](#) &entM, const std::string &scriptPath)  
*Constructor.*
- int [addEntity](#) (const std::string &entName)
- void [visit](#) ([PositionComponent](#) &)
- void [visit](#) ([HealthComponent](#) &)
- void [visit](#) ([AttackComponent](#) &)
- void [visit](#) ([ArmorComponent](#) &)
- void [visit](#) ([SpeedComponent](#) &)

### 6.5.1 Description détaillée

Cette classe a pour but de creer les entites sur demande.

La classe va charger et stocker des templates de chaque entites et va les copier dans [ComponentStorer](#) quand on creer l'objet

### 6.5.2 Documentation des constructeurs et destructeur

#### 6.5.2.1 EntityCreator()

```
EntityCreator::EntityCreator (
    ComponentStorer & compS,
    EntityManager & entM,
    const std::string & scriptPath )
```

Constructor.

Constructeur.

#### Paramètres

<i>compStorer</i>	le stockage des composants
<i>entManager</i>	le gestionnaire des entites
<i>scriptPath</i>	le chemin vers le fichier lua contenant les templates des entites

### 6.5.3 Documentation des fonctions membres

#### 6.5.3.1 addEntity()

```
int EntityCreator::addEntity (
    const std::string & entName )
```

Cette fonction ajoute une entite dans le gestionnaire et les composant dans le stockage a partir d'un template

##### Paramètres

<i>entName</i>	le nom de l'entite a creer
----------------	----------------------------

##### Renvoie

l'id de l'entite creer, -1 si erreur

#### 6.5.3.2 visit() [1/5]

```
void EntityCreator::visit (
    PositionComponent & comp ) [virtual]
```

Permet aux composants de visiter la classe

##### Paramètres

<i>comp</i>	le composant
-------------	--------------

Réimplémentée à partir de [Visitor](#).

#### 6.5.3.3 visit() [2/5]

```
void EntityCreator::visit (
    HealthComponent & comp ) [virtual]
```

Permet aux composants de visiter la classe

##### Paramètres

<i>comp</i>	le composant
-------------	--------------

Réimplémentée à partir de [Visitor](#).

#### 6.5.3.4 visit() [3/5]

```
void EntityCreator::visit (  
    AttackComponent & comp ) [virtual]
```

Permet aux composants de visiter la classe

##### Paramètres

<i>comp</i>	le composant
-------------	--------------

Réimplémentée à partir de [Visitor](#).

#### 6.5.3.5 visit() [4/5]

```
void EntityCreator::visit (  
    ArmorComponent & comp ) [virtual]
```

Permet aux composants de visiter la classe

##### Paramètres

<i>comp</i>	le composant
-------------	--------------

Réimplémentée à partir de [Visitor](#).

#### 6.5.3.6 visit() [5/5]

```
void EntityCreator::visit (  
    SpeedComponent & comp ) [virtual]
```

Permet aux composants de visiter la classe

##### Paramètres

<i>comp</i>	le composant
-------------	--------------

Réimplémentée à partir de [Visitor](#).

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

- include/[EntityCreator.hpp](#)
- src/[EntityCreator.cpp](#)

## 6.6 Référence de la classe EntityManager

```
#include <EntityManager.hpp>
```

### Fonctions membres publiques

- unsigned int getNextId ()
- void removeEntity (unsigned int id)
- std::vector< unsigned int > getEntitiesId () const

### 6.6.1 Documentation des fonctions membres

#### 6.6.1.1 getEntitiesId()

```
std::vector<unsigned int> EntityManager::getEntitiesId ( ) const [inline]
```

#### 6.6.1.2 getNextId()

```
unsigned int EntityManager::getNextId ( )
```

#### 6.6.1.3 removeEntity()

```
void EntityManager::removeEntity (
    unsigned int id )
```

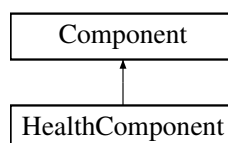
La documentation de cette classe a été générée à partir des fichiers suivants :

- include/EntityManager.hpp
- src/EntityManager.cpp

## 6.7 Référence de la structure HealthComponent

```
#include <Component.hpp>
```

Graphe d'héritage de HealthComponent :



## Fonctions membres publiques

- [HealthComponent](#) (int h)
- virtual void [accept](#) ([Visitor](#) &v)

## Attributs publics

- int [health](#)

### 6.7.1 Documentation des constructeurs et destructeur

#### 6.7.1.1 HealthComponent()

```
HealthComponent::HealthComponent (  
    int h ) [inline]
```

### 6.7.2 Documentation des fonctions membres

#### 6.7.2.1 accept()

```
void HealthComponent::accept (  
    Visitor & v ) [virtual]
```

Implémente [Component](#).

### 6.7.3 Documentation des données membres

#### 6.7.3.1 health

```
int HealthComponent::health
```

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

- include/[Component.hpp](#)
- src/[Component.cpp](#)

## 6.8 Référence de la classe LuaScript

```
#include <LuaScript.hpp>
```

## Fonctions membres publiques

- [LuaScript](#) (const std : :string &scriptName)
- [~LuaScript](#) ()
- luabridge : :LuaRef [get](#) (const std : :string &varName)
- std : :unordered\_map< std : :string, luabridge : :LuaRef > [getKeyValueMap](#) (const luabridge : :LuaRef &table)

### 6.8.1 Documentation des constructeurs et destructeur

#### 6.8.1.1 LuaScript()

```
LuaScript::LuaScript (
    const std::string & scriptName )
```

#### 6.8.1.2 ~LuaScript()

```
LuaScript::~~LuaScript ( )
```

### 6.8.2 Documentation des fonctions membres

#### 6.8.2.1 get()

```
luabridge::LuaRef LuaScript::get (
    const std::string & varName )
```

#### 6.8.2.2 getKeyValueMap()

```
std::unordered_map< std::string, luabridge::LuaRef > LuaScript::getKeyValueMap (
    const luabridge::LuaRef & table )
```

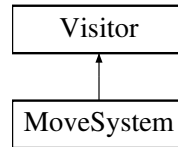
La documentation de cette classe a été générée à partir des fichiers suivants :

- include/[LuaScript.hpp](#)
- src/[LuaScript.cpp](#)

## 6.9 Référence de la classe MoveSystem

```
#include <MoveSystem.hpp>
```

Graphe d'héritage de MoveSystem :



### Fonctions membres publiques

- `MoveSystem (ComponentStorer &comps, EntityManager &entM)`
- `void update ()`
- `void visit (PositionComponent &pComp)`

### 6.9.1 Documentation des constructeurs et destructeur

#### 6.9.1.1 MoveSystem()

```
MoveSystem::MoveSystem (
    ComponentStorer & comps,
    EntityManager & entM )
```

### 6.9.2 Documentation des fonctions membres

#### 6.9.2.1 update()

```
void MoveSystem::update ( )
```

#### 6.9.2.2 visit()

```
void MoveSystem::visit (
    PositionComponent & pComp ) [virtual]
```

Réimplémentée à partir de [Visitor](#).

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

- `include/MoveSystem.hpp`
- `src/MoveSystem.cpp`



## 6.10 Référence de la structure Position

```
#include <Define.hpp>
```

### Fonctions membres publiques

- `Position & operator+=` (const `Position` &r)
- `Position & operator*=` (int d)
- `Position & operator++` ()
- `Position operator++` (int)

### Attributs publics

- int x
- int y

## 6.10.1 Documentation des fonctions membres

### 6.10.1.1 `operator*=()`

```
Position & Position::operator*= (  
    int d )
```

### 6.10.1.2 `operator++()` [1/2]

```
Position & Position::operator++ ( )
```

### 6.10.1.3 `operator++()` [2/2]

```
Position Position::operator++ (  
    int )
```

### 6.10.1.4 `operator+=()`

```
Position & Position::operator+= (  
    const Position & r )
```

## 6.10.2 Documentation des données membres

### 6.10.2.1 x

```
int Position::x
```

### 6.10.2.2 y

```
int Position::y
```

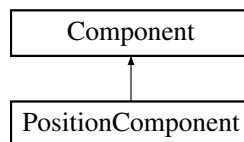
La documentation de cette structure a été générée à partir des fichiers suivants :

- [include/Define.hpp](#)
- [src/Define.cpp](#)

## 6.11 Référence de la structure PositionComponent

```
#include <Component.hpp>
```

Graphe d'héritage de PositionComponent :



### Fonctions membres publiques

- [PositionComponent](#) ([Position](#) p)
- virtual void [accept](#) ([Visitor](#) &v)

### Attributs publics

- [Position](#) position

### 6.11.1 Documentation des constructeurs et destructeur

### 6.11.1.1 PositionComponent()

```
PositionComponent::PositionComponent (
    Position p ) [inline]
```

## 6.11.2 Documentation des fonctions membres

### 6.11.2.1 accept()

```
void PositionComponent::accept (
    Visitor & v ) [virtual]
```

Implémente [Component](#).

## 6.11.3 Documentation des données membres

### 6.11.3.1 position

```
Position PositionComponent::position
```

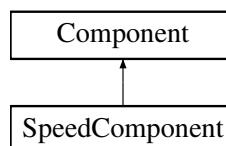
La documentation de cette structure a été générée à partir des fichiers suivants :

- [include/Component.hpp](#)
- [src/Component.cpp](#)

## 6.12 Référence de la structure SpeedComponent

```
#include <Component.hpp>
```

Graphe d'héritage de SpeedComponent :



### Fonctions membres publiques

- [SpeedComponent](#) (int s)
- virtual void [accept](#) ([Visitor](#) &v)

## Attributs publics

— int [speed](#)

### 6.12.1 Documentation des constructeurs et destructeur

#### 6.12.1.1 SpeedComponent()

```
SpeedComponent::SpeedComponent (  
    int s ) [inline]
```

### 6.12.2 Documentation des fonctions membres

#### 6.12.2.1 accept()

```
void SpeedComponent::accept (  
    Visitor & v ) [virtual]
```

Implémente [Component](#).

### 6.12.3 Documentation des données membres

#### 6.12.3.1 speed

```
int SpeedComponent::speed
```

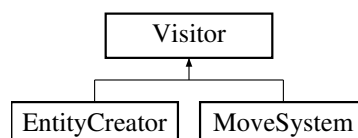
La documentation de cette structure a été générée à partir des fichiers suivants :

- include/[Component.hpp](#)
- src/[Component.cpp](#)

## 6.13 Référence de la classe Visitor

```
#include <Component.hpp>
```

Graphe d'héritage de Visitor :



## Fonctions membres publiques

- virtual [~Visitor](#) ()=0
- virtual void [visit](#) ([PositionComponent](#) &comp)
- virtual void [visit](#) ([HealthComponent](#) &comp)
- virtual void [visit](#) ([AttackComponent](#) &comp)
- virtual void [visit](#) ([ArmorComponent](#) &comp)
- virtual void [visit](#) ([SpeedComponent](#) &comp)

### 6.13.1 Documentation des constructeurs et destructeur

#### 6.13.1.1 [~Visitor\(\)](#)

```
Visitor::~Visitor ( ) [inline], [pure virtual]
```

### 6.13.2 Documentation des fonctions membres

#### 6.13.2.1 [visit\(\)](#) [1/5]

```
virtual void Visitor::visit (
    PositionComponent & comp ) [inline], [virtual]
```

Réimplémentée dans [EntityCreator](#), et [MoveSystem](#).

#### 6.13.2.2 [visit\(\)](#) [2/5]

```
virtual void Visitor::visit (
    HealthComponent & comp ) [inline], [virtual]
```

Réimplémentée dans [EntityCreator](#).

#### 6.13.2.3 [visit\(\)](#) [3/5]

```
virtual void Visitor::visit (
    AttackComponent & comp ) [inline], [virtual]
```

Réimplémentée dans [EntityCreator](#).

**6.13.2.4 visit()** [4/5]

```
virtual void Visitor::visit (  
    ArmorComponent & comp ) [inline], [virtual]
```

Réimplémentée dans [EntityCreator](#).

**6.13.2.5 visit()** [5/5]

```
virtual void Visitor::visit (  
    SpeedComponent & comp ) [inline], [virtual]
```

Réimplémentée dans [EntityCreator](#).

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

— include/[Component.hpp](#)

## Chapitre 7

# Documentation des fichiers

### 7.1 Référence du fichier include/Component.hpp

```
#include "Define.hpp"  
#include <SFML/Graphics.hpp>  
#include <array>  
#include <memory>
```

#### Classes

- struct [Component](#)
- struct [PositionComponent](#)
- struct [HealthComponent](#)
- struct [SpeedComponent](#)
- struct [AttackComponent](#)
- struct [ArmorComponent](#)
- class [Visitor](#)

### 7.2 Référence du fichier include/ComponentStorer.hpp

```
#include "Component.hpp"  
#include <memory>  
#include <map>  
#include <vector>  
#include <typeindex>  
#include <iostream>
```

#### Classes

- class [ComponentStorer](#)

## 7.3 Référence du fichier include/Define.hpp

```
#include <array>
```

### Classes

- struct [Position](#)

### Espaces de nommage

- [DamageType](#)
- [AttackType](#)
- [Direction](#)

### Énumérations

- enum [DamageType](#) : :Type { [DamageType](#) : :Pierce, [DamageType](#) : :Shock, [DamageType](#) : :Magic, [DamageType](#) : :NONE }
- enum [AttackType](#) : :Type { [AttackType](#) : :Melee, [AttackType](#) : :Distance, [AttackType](#) : :NONE }
- enum [Direction](#) : :Dir { [Direction](#) : :Up, [Direction](#) : :Down, [Direction](#) : :Left, [Direction](#) : :Right, [Direction](#) : :None }

### Fonctions

- bool [operator==](#) (const [Position](#) &l, const [Position](#) &r)
- [Position](#) [operator+](#) (const [Position](#) &l, const [Position](#) &r)
- [Position](#) [operator\\*](#) (const [Position](#) &l, int d)

### Variables

- const std : :array< Type, 3 > [DamageType](#) : :All = {Shock, Pierce, Magic}
- const std : :array< Type, 2 > [AttackType](#) : :All = {Melee, Distance}
- const std : :array< Dir, 4 > [Direction](#) : :All = {Up, Down, Left, Right}

#### 7.3.1 Documentation des fonctions

##### 7.3.1.1 [operator\\*](#)()

```
Position operator\* (
    const Position & l,
    int d )
```



### 7.3.1.2 operator+()

```
Position operator+ (
    const Position & l,
    const Position & r )
```

### 7.3.1.3 operator==()

```
bool operator== (
    const Position & l,
    const Position & r )
```

## 7.4 Référence du fichier include/ECS.hpp

```
#include "MoveSystem.hpp"
```

## 7.5 Référence du fichier include/ECSEntity.hpp

```
#include "Component.hpp"
#include "ComponentStorer.hpp"
#include "EntityManager.hpp"
#include "EntityCreator.hpp"
```

## 7.6 Référence du fichier include/EntityCreator.hpp

Ici est declare la class [EntityCreator](#).

```
#include "ComponentStorer.hpp"
#include "EntityManager.hpp"
#include "LuaScript.hpp"
#include <map>
#include <memory>
#include <string>
#include <vector>
#include <cassert>
```

### Classes

- class [EntityCreator](#)  
*Cette classe a pour but de creer les entites sur demande.*

### 7.6.1 Description détaillée

Ici est declare la class [EntityCreator](#).

#### Auteur

PapyRedstone

#### Version

1.0

## 7.7 Référence du fichier include/EntityManager.hpp

```
#include <vector>
#include <algorithm>
```

### Classes

— class [EntityManager](#)

## 7.8 Référence du fichier include/LuaScript.hpp

```
#include <string>
#include <iostream>
#include <vector>
#include <lua.hpp>
#include <unordered_map>
#include <LuaBridge/LuaBridge.h>
```

### Classes

— class [LuaScript](#)

## 7.9 Référence du fichier include/MoveSystem.hpp

```
#include "ECSEntity.hpp"
```

### Classes

— class [MoveSystem](#)

## 7.10 Référence du fichier src/Component.cpp

```
#include "Component.hpp"
```

## 7.11 Référence du fichier src/ComponentStorer.cpp

```
#include "ComponentStorer.hpp"
```

## 7.12 Référence du fichier src/Define.cpp

```
#include "Define.hpp"
```

### Fonctions

- `bool operator==` (const `Position` &l, const `Position` &r)
- `Position operator+` (const `Position` &l, const `Position` &r)
- `Position operator*` (const `Position` &l, int d)

### 7.12.1 Documentation des fonctions

#### 7.12.1.1 `operator*()`

```
Position operator* (
    const Position & l,
    int d )
```

#### 7.12.1.2 `operator+()`

```
Position operator+ (
    const Position & l,
    const Position & r )
```

#### 7.12.1.3 `operator==()`

```
bool operator== (
    const Position & l,
    const Position & r )
```

### 7.13 Référence du fichier src/EntityCreator.cpp

```
#include "EntityCreator.hpp"
```

### 7.14 Référence du fichier src/EntityManager.cpp

```
#include "EntityManager.hpp"
```

### 7.15 Référence du fichier src/LuaScript.cpp

```
#include "LuaScript.hpp"
```

### 7.16 Référence du fichier src/main.cpp

```
#include "ECS.hpp"
```

#### Fonctions

— int `main` ()

#### 7.16.1 Documentation des fonctions

##### 7.16.1.1 `main()`

```
int main ( )
```

### 7.17 Référence du fichier src/MoveSystem.cpp

```
#include "MoveSystem.hpp"
```

# Index

- ~LuaScript
  - LuaScript, [23](#)
- ~Visitor
  - Visitor, [29](#)
- accept
  - ArmorComponent, [14](#)
  - AttackComponent, [15](#)
  - Component, [16](#)
  - HealthComponent, [22](#)
  - PositionComponent, [27](#)
  - SpeedComponent, [28](#)
- addComponent
  - ComponentStorer, [17](#)
- addEntity
  - EntityCreator, [19](#)
- All
  - AttackType, [9](#)
  - Direction, [10](#)
  - DommmageType, [11](#)
- amount
  - AttackComponent, [15](#)
- ArmorComponent, [13](#)
  - accept, [14](#)
  - ArmorComponent, [13](#)
  - magicArmor, [14](#)
  - pierceArmor, [14](#)
  - shockArmor, [14](#)
- AttackComponent, [14](#)
  - accept, [15](#)
  - amount, [15](#)
  - AttackComponent, [15](#)
  - attackType, [15](#)
  - dommmageType, [16](#)
- AttackType, [9](#)
  - All, [9](#)
  - Type, [9](#)
- attackType
  - AttackComponent, [15](#)
- Component, [16](#)
  - accept, [16](#)
- ComponentStorer, [17](#)
  - addComponent, [17](#)
  - getComponent, [17](#)
  - getComponentsByld, [17](#)
  - getComponentsByType, [17](#)
- Define.cpp
  - operator\*, [35](#)
- operator+, [35](#)
- operator==, [35](#)
- Define.hpp
  - operator\*, [32](#)
  - operator+, [32](#)
  - operator==, [33](#)
- Dir
  - Direction, [10](#)
- Direction, [10](#)
  - All, [10](#)
  - Dir, [10](#)
  - Down, [10](#)
  - Left, [10](#)
  - None, [10](#)
  - Right, [10](#)
  - Up, [10](#)
- DommmageType, [10](#)
  - All, [11](#)
  - Type, [11](#)
- dommmageType
  - AttackComponent, [16](#)
- Down
  - Direction, [10](#)
- EntityCreator, [18](#)
  - addEntity, [19](#)
  - EntityCreator, [18](#)
  - visit, [19](#), [20](#)
- EntityManager, [21](#)
  - getEntitiesld, [21](#)
  - getNextld, [21](#)
  - removeEntity, [21](#)
- get
  - LuaScript, [23](#)
- getComponent
  - ComponentStorer, [17](#)
- getComponentsByld
  - ComponentStorer, [17](#)
- getComponentsByType
  - ComponentStorer, [17](#)
- getEntitiesld
  - EntityManager, [21](#)
- getKeyValueMap
  - LuaScript, [23](#)
- getNextld
  - EntityManager, [21](#)
- health
  - HealthComponent, [22](#)

- HealthComponent, 21
  - accept, 22
  - health, 22
  - HealthComponent, 22
- include/Component.hpp, 31
- include/ComponentStorer.hpp, 31
- include/Define.hpp, 32
- include/ECS.hpp, 33
- include/ECSEntity.hpp, 33
- include/EntityCreator.hpp, 33
- include/EntityManager.hpp, 34
- include/LuaScript.hpp, 34
- include/MoveSystem.hpp, 34
- Left
  - Direction, 10
- LuaScript, 22
  - ~LuaScript, 23
  - get, 23
  - getKeyValueMap, 23
  - LuaScript, 23
- magicArmor
  - ArmorComponent, 14
- main
  - main.cpp, 36
- main.cpp
  - main, 36
- MoveSystem, 24
  - MoveSystem, 24
  - update, 24
  - visit, 24
- None
  - Direction, 10
- operator\*
  - Define.cpp, 35
  - Define.hpp, 32
- operator\*=
  - Position, 25
- operator+
  - Define.cpp, 35
  - Define.hpp, 32
- operator++
  - Position, 25
- operator+=
  - Position, 25
- operator==
  - Define.cpp, 35
  - Define.hpp, 33
- pierceArmor
  - ArmorComponent, 14
- Position, 25
  - operator\*=, 25
  - operator++, 25
  - operator+=, 25
  - x, 26
  - y, 26
- position
  - PositionComponent, 27
- PositionComponent, 26
  - accept, 27
  - position, 27
  - PositionComponent, 26
- removeEntity
  - EntityManager, 21
- Right
  - Direction, 10
- shockArmor
  - ArmorComponent, 14
- speed
  - SpeedComponent, 28
- SpeedComponent, 27
  - accept, 28
  - speed, 28
  - SpeedComponent, 28
- src/Component.cpp, 35
- src/ComponentStorer.cpp, 35
- src/Define.cpp, 35
- src/EntityCreator.cpp, 36
- src/EntityManager.cpp, 36
- src/LuaScript.cpp, 36
- src/MoveSystem.cpp, 36
- src/main.cpp, 36
- Type
  - AttackType, 9
  - DommageType, 11
- Up
  - Direction, 10
- update
  - MoveSystem, 24
- visit
  - EntityCreator, 19, 20
  - MoveSystem, 24
  - Visitor, 29, 30
- Visitor, 28
  - ~Visitor, 29
  - visit, 29, 30
- x
  - Position, 26
- y
  - Position, 26