# My Project

Generated by Doxygen 1.8.17

# Chapter 1

# Source content

This folder should contain only hpp/cpp files of your implementation. You can also place hpp files in a separate directory `include`.

You can create a summary of files here. It might be useful to describe file relations, and brief summary of their content.

# Chapter 2

# Hierarchical Index

## 2.1 Class Hierarchy

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

# Chapter 3

# Class Index

## 3.1 Class List

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

# Chapter 4

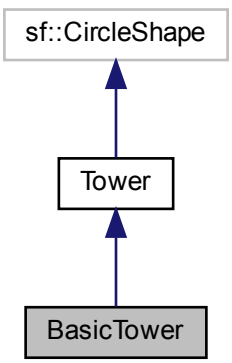
# Class Documentation

## 4.1 BasicTower Class Reference

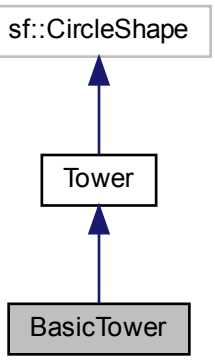
class BasicTower is the most basic type of tower. It can hit any type of enemy and does 1 HP of damage every hit.

`#include <basictower.hpp>`

Inheritance diagram for BasicTower:



Collaboration diagram for BasicTower:




## Public Member Functions

- BasicTower (sf::Vector2i position)

  *Construct a new Basic Tower object.*
- **BasicTower** (const BasicTower &tower)
- virtual std::shared_ptr< Projectile > **Shoot** (std::shared_ptr< Enemy > enemies)

## Additional Inherited Members

### 4.1.1 Detailed Description

class BasicTower is the most basic type of tower. It can hit any type of enemy and does 1 HP of damage every hit.

### 4.1.2 Constructor & Destructor Documentation

#### 4.1.2.1 BasicTower()

```
BasicTower::BasicTower (
            sf::Vector2i position )
```

Construct a new Basic Tower object.

**Parameters**

| | |
|---|---|
| *position* | |

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

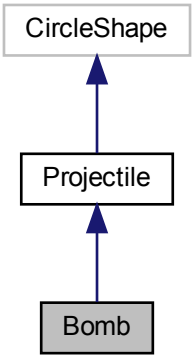
- src/basictower.hpp
- src/basictower.cpp

## 4.2 Bomb Class Reference

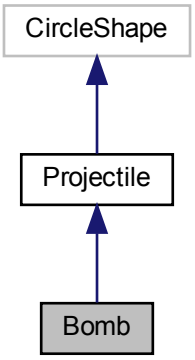
The projectile of the bomb tower.

`#include <bomb.hpp>`

Inheritance diagram for Bomb:




Collaboration diagram for Bomb:



### Public Member Functions

- **Bomb** (sf::Vector2f target, int damage, sf::Vector2f position)

### 4.2.1 Detailed Description

The projectile of the bomb tower.

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

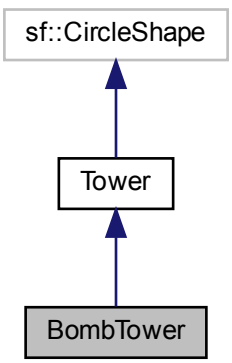
- src/bomb.hpp
- src/bomb.cpp

## 4.3 BombTower Class Reference

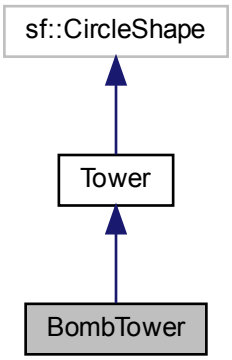
class BombTower throws bombs that damages all enemies within a certain range of the explosion. It can only hit grounded enemies and does 1 HP of damage to everyone hit.

`#include <bombtower.hpp>`

Inheritance diagram for BombTower:



Collaboration diagram for BombTower:




## Public Member Functions

- BombTower (sf::Vector2i position)

  *Construct a new Bomb Tower object.*
- **BombTower** (const BombTower &tower)
- virtual std::shared_ptr< Projectile > **Shoot** (std::shared_ptr< Enemy > enemy)

## Additional Inherited Members

### 4.3.1 Detailed Description

class BombTower throws bombs that damages all enemies within a certain range of the explosion. It can only hit grounded enemies and does 1 HP of damage to everyone hit.

### 4.3.2 Constructor & Destructor Documentation

#### 4.3.2.1 BombTower()

```
BombTower::BombTower (
            sf::Vector2i position )
```

Construct a new Bomb Tower object.

**Parameters**

| | |
|---|---|
| *position* | |

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

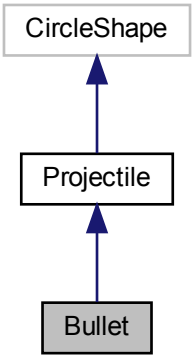
- src/bombtower.hpp
- src/bombtower.cpp

## 4.4 Bullet Class Reference

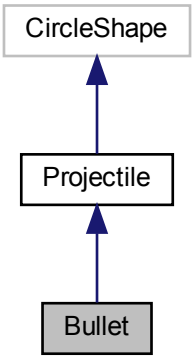
The projectile of the basic tower.

`#include <bullet.hpp>`

Inheritance diagram for Bullet:




Collaboration diagram for Bullet:



### Public Member Functions

- **Bullet** (sf::Vector2f target, int damage, sf::Vector2f position)

### 4.4.1 Detailed Description

The projectile of the basic tower.

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

- src/bullet.hpp
- src/bullet.cpp

## 4.5 Enemy Class Reference

### Public Member Functions

- **Enemy** (int type=EnemyType::BasicEnemy, float x=0, float y=150.f, float dir_x=1.f, float dir_y=0.f, float movementSpeed=1.f)
- const sf::Vector2f & **getPos** () const
- const sf::FloatRect **getBounds** () const
- const int & **getHp** () const
- const int & **getHpMax** () const
- void **setPosition** (const sf::Vector2f pos)
- void **setPosition** (const float x, const float y)
- void **setHp** (const int hp)
- void **loseHp** (const int value)
- int **getType** () const
- float **getDirX** () const
- float **getDirY** () const
- void **setMovementDirection** (float dir_x, float dir_y)
- void **move** (const float dirX, const float dirY)
- void **updateMovement** ()
- void **updateHp** ()
- void **update** ()
- void **render** (sf::RenderTarget &target)
- void **SlowDown** ()

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

- src/Enemy.h
- src/Enemy.cpp

## 4.6 Game Class Reference

### Public Member Functions

- const bool **running** () const
- void **pollEvents** ()
- void **updateGui** ()
- void **update** ()
- void **renderGui** (sf::RenderTarget ∗target)
- void render ()

### 4.6.1 Member Function Documentation

#### 4.6.1.1 render()

```
void Game::render ( )
```

**Returns**

void

- clear old frame
- render objects
- display frame in window

Renders the game objects.

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

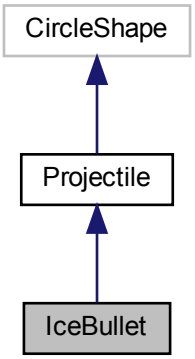
- src/Game.h
- src/Game.cpp

## 4.7 IceBullet Class Reference

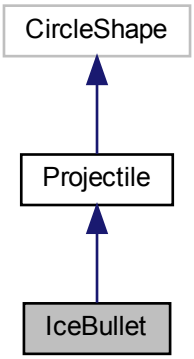
The projectile of the slow tower.

```
#include <icebullet.hpp>
```

Inheritance diagram for IceBullet:



Collaboration diagram for IceBullet:




### Public Member Functions

- **IceBullet** (sf::Vector2f target, int damage, sf::Vector2f position)

### 4.7.1 Detailed Description

The projectile of the slow tower.

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

- src/icebullet.hpp
- src/icebullet.cpp

## 4.8 Level Class Reference

### Public Member Functions

- **Level** (std::string, std::string)
- sf::Vector2f **getHomePosition** () const
- std::vector< sf::Vector2f > **getSpwanPoints** () const
- int **getLeftEmenies** () const
- int **getGameStatus** () const
- void **AddTower** (std::shared_ptr< Tower >)
- int **GetTowerAmount** () const
- std::vector< std::shared_ptr< Tower > > **GetTowers** () const
- void **AddEnemy** (std::shared_ptr< Enemy >)
- std::vector< std::shared_ptr< Enemy > > **GetEnemies** () const
- void **AddProjectile** (std::shared_ptr< Projectile >)
- int **GetProjectileAmount** () const

- std::vector< std::shared_ptr< Projectile > > **GetProjectiles** () const
- void **UpdateTowerType** (std::string)
- std::string **GetTowerType** () const
- bool **UpdateMoney** (int)
- int **GetMoney** () const
- int getTileTypeByPos (const float x, const float y) const

  *Get the Tile Type By Pos object.*
- void **updateEnemies** ()
- int **update** (sf::RenderWindow &window)
- void **render** (sf::RenderTarget &target)
- void **UpdateEvent** (sf::Event event)
- std::string **GetMap** () const

## Public Attributes

- std::vector< std::vector< std::shared_ptr< levelTile > > > **tiles**
- int **gridLenght**

### 4.8.1 Member Function Documentation

#### 4.8.1.1 getTileTypeByPos()

```
int Level::getTileTypeByPos (
            const float x,
            const float y ) const
```

Get the Tile Type By Pos object.

**Parameters**

| | |
|---|---|
| *x* | |
| *y* | |

**Returns**

int

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

- src/level.h
- src/level.cpp

## 4.9 levelTile Class Reference

### Public Member Functions

- **levelTile** (std::string, float, float, int)

- bool **setUpSprite** (std::string)
- const sf::FloatRect **getGlobalBounds** () const
- int **getType** () const

### Public Attributes

- sf::Texture **texture**
- sf::Sprite **sprite**
- int **type**

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

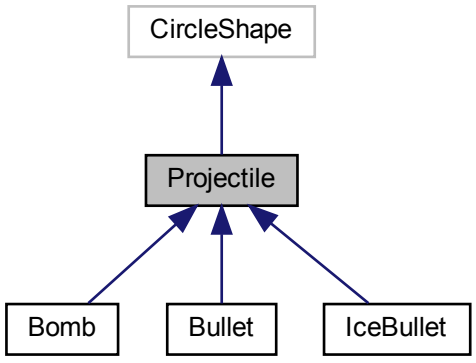
- src/levelTile.h
- src/levelTile.cpp

## 4.10 Projectile Class Reference

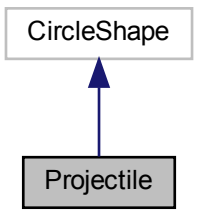
The projectile class defines objects that the towers shoot.

`#include <projectile.hpp>`

Inheritance diagram for Projectile:




Collaboration diagram for Projectile:



### Public Member Functions

- **Projectile** (sf::Vector2f target, float speed, int damage, sf::Vector2f position, bool aoe)
- **Projectile** (const Projectile &projectile)
- sf::Vector2f **GetTarget** () const
- float **GetSpeed** () const
- int **GetDamage** () const
- void **UpdatePosition** ()
- float **DistancteToTarget** () const
- float **DistanceTo** (sf::Vector2f target) const
- bool **IsAOE** () const
- Projectile * **Clone** ()

### 4.10.1 Detailed Description

The projectile class defines objects that the towers shoot.

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

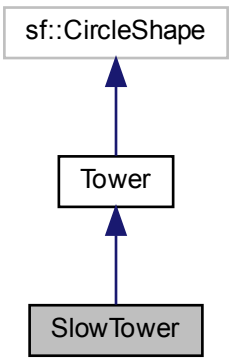
- src/projectile.hpp
- src/projectile.cpp

## 4.11 SlowTower Class Reference

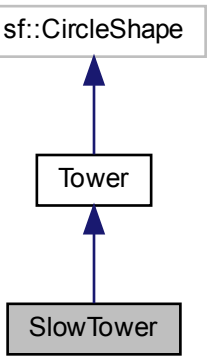
class SlowTower slows down the enemy that it hits. It can hit any type of enemy and does no damage.

`#include <slowtower.hpp>`

Inheritance diagram for SlowTower:



Collaboration diagram for SlowTower:




## Public Member Functions

- SlowTower (sf::Vector2i position)

  *Construct a new Slow Tower object.*
- **SlowTower** (const SlowTower &tower)
- virtual std::shared_ptr< Projectile > **Shoot** (std::shared_ptr< Enemy > enemy)

## Additional Inherited Members

### 4.11.1 Detailed Description

class SlowTower slows down the enemy that it hits. It can hit any type of enemy and does no damage.

### 4.11.2 Constructor & Destructor Documentation

#### 4.11.2.1 SlowTower()

```
SlowTower::SlowTower (
            sf::Vector2i position )
```

Construct a new Slow Tower object.

**Parameters**

| | |
|---|---|
| *position* | |

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

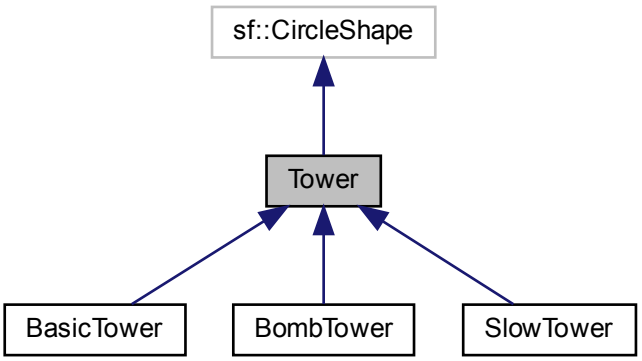
- src/slowtower.hpp
- src/slowtower.cpp

## 4.12 Tower Class Reference

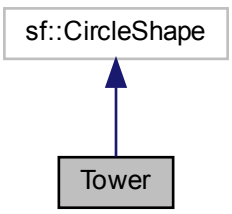
class Tower is the abstract parent class of all different types of towers. Each Tower object has a damage, a range, a position, and a level.

`#include <tower.hpp>`

Inheritance diagram for Tower:




Collaboration diagram for Tower:



## Public Member Functions

- Tower (int damage, int range, int firerate, int upgrade_cost, int max_level, sf::Vector2i position)

  *Construct a new Tower object.*
- ~Tower ()

  *Destroy the Tower object.*
- Tower (const Tower &tower)

  *Copy constructor for a new Tower object.*
- int GetDamage () const

  *Get the Damage object.*
- int GetRange () const

  *Get the Range object.*
- int GetLevel () const

  *Get the Level object.*
- int **GetUpgradeCost** () const
- bool LevelUp ()

  *Upgrade the tower. Returns true if upgrade was succesful, i.e. not already max level and player has enough money.*
- virtual std::shared_ptr< Projectile > **Shoot** (std::shared_ptr< Enemy >)=0
- Tower & operator= (const Tower &tower)

  *Copy assignment operator.*
- float **DistanceTo** (sf::Vector2f target) const

## Protected Attributes

- int **damage_**
- int **range_**
- int **firerate_**
- int **max_level_**
- int **level_**
- int **damage_buff_**
- int **upgrade_cost_**

### 4.12.1 Detailed Description

class Tower is the abstract parent class of all different types of towers. Each Tower object has a damage, a range, a position, and a level.

### 4.12.2 Constructor & Destructor Documentation

#### 4.12.2.1 Tower() [1/2]

```
Tower::Tower (
            int damage,
            int range,
            int firerate,
            int upgrade_cost,
            int max_level,
            sf::Vector2i position )
```

Construct a new Tower object.

**Parameters**

| | |
|---|---|
| *damage* | |
| *range* | |
| *position* | |

#### 4.12.2.2 ~Tower()

```
Tower::~Tower ( )  [default]
```

Destroy the Tower object.

#### 4.12.2.3 Tower() [2/2]

```
Tower::Tower (
            const Tower & tower )
```

Copy constructor for a new Tower object.

**Parameters**

| | |
|---|---|
| *tower* | |

### 4.12.3 Member Function Documentation

#### 4.12.3.1 GetDamage()

```
int Tower::GetDamage ( ) const
```

Get the Damage object.

**Returns**

int

#### 4.12.3.2 GetLevel()

```
int Tower::GetLevel ( ) const
```

Get the Level object.

**Returns**

int

#### 4.12.3.3 GetRange()

```
int Tower::GetRange ( ) const
```

Get the Range object.

**Returns**

int

#### 4.12.3.4 LevelUp()

```
bool Tower::LevelUp ( )
```

Upgrade the tower. Returns true if upgrade was succesful, i.e. not already max level and player has enough money.

**Returns**

true

false

#### 4.12.3.5 operator=()

```
Tower & Tower::operator= (
            const Tower & tower )
```

Copy assignment operator.

**Parameters**

| | |
|---|---|
| *tower* | |

**Returns**

Tower&

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

- src/tower.hpp
- src/tower.cpp

# Index