

Space Defender

Generated by Doxygen 1.13.2

1 Todo List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Class Documentation	9
5.1 Bomb Class Reference	9
5.1.1 Detailed Description	10
5.1.2 Member Function Documentation	10
5.1.2.1 fireWeapon()	10
5.2 Bullet Class Reference	10
5.2.1 Detailed Description	11
5.2.2 Member Function Documentation	11
5.2.2.1 draw()	11
5.2.2.2 fireWeapon()	11
5.3 Laser Class Reference	12
5.3.1 Detailed Description	13
5.3.2 Member Function Documentation	13
5.3.2.1 fireWeapon()	13
5.4 Screen Class Reference	13
5.4.1 Detailed Description	13
5.4.2 Constructor & Destructor Documentation	14
5.4.2.1 ~Screen()	14
5.4.3 Member Function Documentation	14
5.4.3.1 draw()	14
5.5 ScreenGame Class Reference	14
5.5.1 Detailed Description	15
5.5.2 Member Function Documentation	15
5.5.2.1 draw()	15
5.6 ScreenHighscore Class Reference	15
5.6.1 Detailed Description	16
5.6.2 Member Function Documentation	16
5.6.2.1 draw()	16
5.7 ScreenMenu Class Reference	16
5.7.1 Detailed Description	17
5.7.2 Member Function Documentation	17
5.7.2.1 draw()	17
5.8 ScreenSettings Class Reference	17

5.8.1 Detailed Description	18
5.8.2 Member Function Documentation	18
5.8.2.1 draw()	18
5.9 SpaceDefender Class Reference	18
5.9.1 Detailed Description	19
5.9.2 Constructor & Destructor Documentation	19
5.9.2.1 SpaceDefender()	19
5.9.3 Member Function Documentation	20
5.9.3.1 run()	20
5.9.3.2 setScreen()	20
5.10 SpaceShip Class Reference	21
5.10.1 Detailed Description	21
5.10.2 Constructor & Destructor Documentation	22
5.10.2.1 SpaceShip()	22
5.10.2.2 ~SpaceShip()	22
5.10.3 Member Function Documentation	22
5.10.3.1 getHealth()	22
5.10.3.2 getPositionX()	22
5.10.3.3 getPositionY()	22
5.10.3.4 getShipHeight()	22
5.10.3.5 getShipWidth()	23
5.10.3.6 healthReduction()	23
5.10.3.7 movements()	23
5.10.3.8 setShipSpeed()	23
5.10.3.9 shooting()	23
5.11 SpaceShipEnemy Class Reference	23
5.11.1 Detailed Description	24
5.11.2 Member Function Documentation	24
5.11.2.1 movements()	24
5.11.2.2 shooting()	25
5.12 SpaceShipPlayer Class Reference	25
5.12.1 Detailed Description	26
5.12.2 Member Function Documentation	26
5.12.2.1 movements()	26
5.12.2.2 shooting()	27
5.13 Weapon Class Reference	27
5.13.1 Detailed Description	28
5.13.2 Constructor & Destructor Documentation	28
5.13.2.1 Weapon()	28
5.13.2.2 ~Weapon()	28
5.13.3 Member Function Documentation	29
5.13.3.1 draw()	29

5.13.3.2 fireWeapon()	29
5.13.3.3 getDamage()	29
5.13.3.4 getSpeed()	29
5.13.3.5 move()	29
6 File Documentation	31
6.1 main.cpp File Reference	31
6.1.1 Detailed Description	31
6.1.2 Function Documentation	31
6.1.2.1 main()	31
6.2 Screen.cpp File Reference	32
6.2.1 Detailed Description	32
6.3 Screen.h File Reference	32
6.3.1 Detailed Description	33
6.4 Screen.h	33
6.5 SpaceDefender.cpp File Reference	34
6.5.1 Detailed Description	34
6.6 SpaceDefender.h File Reference	34
6.6.1 Detailed Description	35
6.7 SpaceDefender.h	35
6.8 SpaceShip.cpp File Reference	36
6.8.1 Detailed Description	36
6.9 SpaceShip.h File Reference	36
6.9.1 Detailed Description	37
6.10 SpaceShip.h	37
6.11 Weapon.cpp File Reference	38
6.11.1 Detailed Description	38
6.12 Weapon.h File Reference	38
6.12.1 Detailed Description	39
6.13 Weapon.h	39

Chapter 1

Todo List

Class `Bomb`

Add `Bomb` to the game

Member `Bomb::fireWeapon (SpaceDefender &>window)` override

Add this function

Member `formatPlayerInfo (const Player &player)`

Make better, width depends on characters

Class `Laser`

Add `Laser` to the game

Member `Laser::fireWeapon (SpaceDefender &>window)` override

Add this function

Member `ScreenHighscore::draw (SpaceDefender &>window)` override

Add highscores to the screen that is read from a json file

Member `ScreenSettings::draw (SpaceDefender &>window)` override

Add settings to the screen

Member `SpaceDefender::SpaceDefender (TDT4102::Point position={100, 100}, int width=600, int height=800, const std::string &title="Space Defender")`

Fix how enemies spawn

Need dynamic

Member `SpaceShipEnemy::movements (SpaceDefender &>window)` override

Fix the movement of the enemy

Member `SpaceShipEnemy::shooting (SpaceDefender &>window)` override

Fix shooting of the enemy

Method to find lowest ship in each column

One of said ships fire randomly

Firerate approximatly same as playership

Class `SpaceShipPlayer`

firerate in regards to different weapomn types

Member `SpaceShipPlayer::shooting (SpaceDefender &>window)` override

Only bullets are fired, consider making it more general. For example having a set weapon type

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

TDT4102::AnimationWindow	
SpaceDefender	18
Player	??
Screen	13
ScreenGame	14
ScreenHighscore	15
ScreenMenu	16
ScreenSettings	17
SpaceShip	21
SpaceShipEnemy	23
SpaceShipPlayer	25
Weapon	27
Bomb	9
Bullet	10
Laser	12

Chapter 3

Class Index

3.1 Class List

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

Bomb	Class for Bomb . Inherits from Weapon	9
Bullet	Class for Bullet . Inherits from Weapon	10
Laser	Class for Laser . Inherits from Weapon	12
Player	An object representing a player in regards to highscores	??
Screen	Abstract base class for different screens	13
ScreenGame	The game screen	14
ScreenHighscore	The highscore screen	15
ScreenMenu	The menu screen	16
ScreenSettings	The settings screen	17
SpaceDefender	The main game class which runs the game. Uses AnimationWindow as base class	18
SpaceShip	Abstract base class for different spaceships	21
SpaceShipEnemy	Class for enemy ship	23
SpaceShipPlayer	Class for player ship	25
Weapon	Abstract base class for different weapons	27

Chapter 4

File Index

4.1 File List

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

main.cpp	Main file	31
Screen.cpp	The cpp file for the Screen class	32
Screen.h	The header file for the Screen class	32
SpaceDefender.cpp	The cpp file for the SpaceDefender class	34
SpaceDefender.h	The header file for the SpaceDefender class	34
SpaceShip.cpp	The cpp file for the SpaceShip class	36
SpaceShip.h	The header file for the SpaceShip class	36
Weapon.cpp	The cpp file for the Weapon class	38
Weapon.h	The header file for the Weapon class	38

Chapter 5

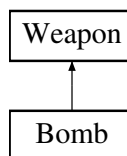
Class Documentation

5.1 Bomb Class Reference

Class for [Bomb](#). Inherits from [Weapon](#).

```
#include <Weapon.h>
```

Inheritance diagram for Bomb:



Public Member Functions

- **Bomb** (int speed, int damage)
- void [fireWeapon](#) ([SpaceDefender](#) &>window) override

Public Member Functions inherited from [Weapon](#)

- [Weapon](#) (int speed, int damage)
- virtual [~Weapon](#) ()=default
- int [getSpeed](#) ()
- int [getDamage](#) ()
- virtual void [move](#) ()
- virtual void [draw](#) ([SpaceDefender](#) &>window)=0

Additional Inherited Members

Protected Attributes inherited from [Weapon](#)

- int **speed**
- int **damage**
- int **xProjectile**
- int **yProjectile**

5.1.1 Detailed Description

Class for [Bomb](#). Inherits from [Weapon](#).

Todo Add [Bomb](#) to the game

5.1.2 Member Function Documentation

5.1.2.1 fireWeapon()

```
void Bomb::fireWeapon (
    SpaceDefender & window) [override], [virtual]
```

Todo Add this function

Parameters

<i>window</i>	
---------------	--

Implements [Weapon](#).

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

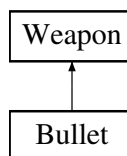
- [Weapon.h](#)
- [Weapon.cpp](#)

5.2 Bullet Class Reference

Class for [Bullet](#). Inherits from [Weapon](#).

```
#include <Weapon.h>
```

Inheritance diagram for Bullet:



Public Member Functions

- **Bullet** (int speed, int damage)
- void [fireWeapon](#) ([SpaceDefender](#) &window) override
Fire the weapon.
- void [draw](#) ([SpaceDefender](#) &window) override
Draw the bullet on the screen.

Public Member Functions inherited from [Weapon](#)

- [Weapon](#) (int speed, int damage)
- virtual [~Weapon](#) ()=default
- int [getSpeed](#) ()
- int [getDamage](#) ()
- virtual void [move](#) ()

Additional Inherited Members

Protected Attributes inherited from [Weapon](#)

- int **speed**
- int **damage**
- int **xProjectile**
- int **yProjectile**

5.2.1 Detailed Description

Class for [Bullet](#). Inherits from [Weapon](#).

Parameters

<i>radius</i>	Radius of the bullet
---------------	----------------------

5.2.2 Member Function Documentation

5.2.2.1 [draw\(\)](#)

```
void Bullet::draw (
    SpaceDefender & window) [override], [virtual]
```

Draw the bullet on the screen.

Parameters

<i>window</i>	SpaceDefender object
<i>radius</i>	Radius of the bullet
<i>location</i>	Location of the bullet. Updated in Bullet::fireWeapon(SpaceDefender& window)

Implements [Weapon](#).

5.2.2.2 [fireWeapon\(\)](#)

```
void Bullet::fireWeapon (
    SpaceDefender & window) [override], [virtual]
```

Fire the weapon.

Sets the position of the projectile to the position of the player. The movement of the projectile is handled in [Bullet::move\(\)](#).

Parameters

<i>window</i>	SpaceDefender object
<i>xProjectile</i>	Position of the projectile in the x-axis
<i>yProjectile</i>	Position of the projectile in the y-axis

Implements [Weapon](#).

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

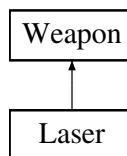
- [Weapon.h](#)
- [Weapon.cpp](#)

5.3 Laser Class Reference

Class for [Laser](#). Inherits from [Weapon](#).

```
#include <Weapon.h>
```

Inheritance diagram for Laser:



Public Member Functions

- **Laser** (int speed, int damage)
- void [fireWeapon](#) ([SpaceDefender](#) &>window) override

Public Member Functions inherited from [Weapon](#)

- [Weapon](#) (int speed, int damage)
- virtual [~Weapon](#) ()=default
- int [getSpeed](#) ()
- int [getDamage](#) ()
- virtual void [move](#) ()
- virtual void [draw](#) ([SpaceDefender](#) &>window)=0

Additional Inherited Members

Protected Attributes inherited from [Weapon](#)

- int **speed**
- int **damage**
- int **xProjectile**
- int **yProjectile**

5.3.1 Detailed Description

Class for [Laser](#). Inherits from [Weapon](#).

Todo Add [Laser](#) to the game

5.3.2 Member Function Documentation

5.3.2.1 fireWeapon()

```
void Laser::fireWeapon (
    SpaceDefender & window) [override], [virtual]
```

Todo Add this function

Parameters

<i>window</i>	
---------------	--

Implements [Weapon](#).

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

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

5.4 Player Struct Reference

An object representing a player in regards to highscores.

```
#include <Screen.h>
```

Public Attributes

- `std::string rank`
- `std::string name`
- `int score`
- `int round`

5.4.1 Detailed Description

An object representing a player in regards to highscores.

Parameters

<i>rank</i>	The rank of the player
<i>name</i>	The name of the player
<i>score</i>	The score of the player
<i>round</i>	The round of the player

The documentation for this struct was generated from the following file:

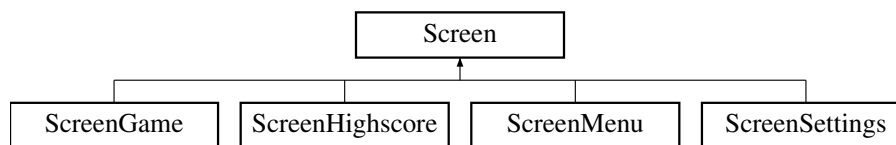
- [Screen.h](#)

5.5 Screen Class Reference

Abstract base class for different screens.

```
#include <Screen.h>
```

Inheritance diagram for Screen:



Public Member Functions

- virtual [~Screen](#) ()=default
- virtual void [draw](#) ([SpaceDefender](#) &window)=0

5.5.1 Detailed Description

Abstract base class for different screens.

Parameters

<i>window</i>	SpaceDefender object
---------------	--------------------------------------

5.5.2 Constructor & Destructor Documentation

5.5.2.1 ~Screen()

```
virtual Screen::~~Screen () [virtual], [default]
```

Virtual destructor to ensure proper cleanup

5.5.3 Member Function Documentation

5.5.3.1 draw()

```
virtual void Screen::draw (
    SpaceDefender & window) [pure virtual]
```

Pure virtual function. Is supposed to draw the screen

Implemented in [ScreenGame](#), [ScreenHighscore](#), [ScreenMenu](#), and [ScreenSettings](#).

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

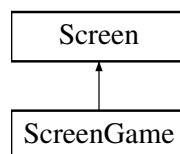
- [Screen.h](#)

5.6 ScreenGame Class Reference

The game screen.

```
#include <Screen.h>
```

Inheritance diagram for ScreenGame:



Public Member Functions

- void [draw](#) ([SpaceDefender](#) &window) override
Draws the screencontent of the Game.

Public Member Functions inherited from [Screen](#)

- virtual [~Screen](#) ()=default

5.6.1 Detailed Description

The game screen.

5.6.2 Member Function Documentation

5.6.2.1 draw()

```
void ScreenGame::draw (
    SpaceDefender & window) [override], [virtual]
```

Draws the screencontent of the Game.

Draws and updates, enemie ships, the player ship and fired weapons.

Parameters

<i>window</i>	SpaceDefender object
---------------	--------------------------------------

Implements [Screen](#).

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

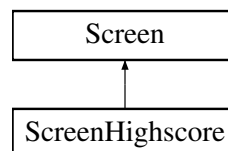
- [Screen.h](#)
- [Screen.cpp](#)

5.7 ScreenHighscore Class Reference

The highscore screen.

```
#include <Screen.h>
```

Inheritance diagram for ScreenHighscore:



Public Member Functions

- void [draw](#) ([SpaceDefender](#) &window) override
Draws the screencontent of the Highscore.

Public Member Functions inherited from [Screen](#)

- virtual [~Screen](#) ()=default

5.7.1 Detailed Description

The highscore screen.

5.7.2 Member Function Documentation

5.7.2.1 draw()

```
void ScreenHighscore::draw (  
    SpaceDefender & window) [override], [virtual]
```

Draws the screencontent of the Highscore.

Draws the highscore screen, and show the back button.

Todo Add highscores to the screen that is read from a json file

Parameters

<i>window</i>	SpaceDefender object
<i>file</i>	The json file variable which contains the highscores

Implements [Screen](#).

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

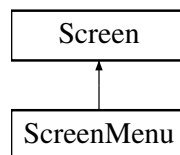
- [Screen.h](#)
- [Screen.cpp](#)

5.8 ScreenMenu Class Reference

The menu screen.

```
#include <Screen.h>
```

Inheritance diagram for ScreenMenu:



Public Member Functions

- void [draw](#) ([SpaceDefender](#) &window) override
Draws the screencontent of the Menu.

Public Member Functions inherited from [Screen](#)

- virtual [~Screen](#) ()=default

5.8.1 Detailed Description

The menu screen.

5.8.2 Member Function Documentation

5.8.2.1 draw()

```
void ScreenMenu::draw (
    SpaceDefender & window) [override], [virtual]
```

Draws the screencontent of the Menu.

Draws the menu screen where it shows all the buttons exept the back button.

Parameters

<i>window</i>	SpaceDefender object
---------------	--------------------------------------

Implements [Screen](#).

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

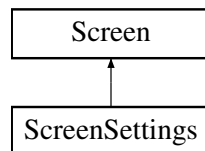
- [Screen.h](#)
- [Screen.cpp](#)

5.9 ScreenSettings Class Reference

The settings screen.

```
#include <Screen.h>
```

Inheritance diagram for ScreenSettings:



Public Member Functions

- void [draw](#) ([SpaceDefender](#) &window) override
Draws the screencontent of the Settings.

Public Member Functions inherited from [Screen](#)

- virtual [~Screen](#) ()=default

5.9.1 Detailed Description

The settings screen.

5.9.2 Member Function Documentation

5.9.2.1 draw()

```
void ScreenSettings::draw (  
    SpaceDefender & window) [override], [virtual]
```

Draws the screencontent of the Settings.

Draws the settings screen, where you can change the game settings

Todo Add settings to the screen

Parameters

<i>window</i>	SpaceDefender object
---------------	--------------------------------------

Implements [Screen](#).

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

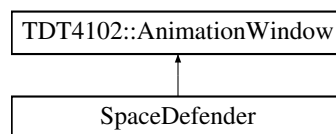
- [Screen.h](#)
- [Screen.cpp](#)

5.10 SpaceDefender Class Reference

The main game class which runs the game. Uses AnimationWindow as base class.

```
#include <SpaceDefender.h>
```

Inheritance diagram for SpaceDefender:



Public Member Functions

- [SpaceDefender](#) (TDT4102::Point position={100, 100}, int width=600, int height=800, const std::string &title="Space Defender")
Construct a new [SpaceDefender::SpaceDefender](#) object.
- void [setScreen](#) (std::unique_ptr< [Screen](#) > newScreen)
Set the current screen.
- void [run](#) ()
Game loop that runs the game until the window is closed.

Public Attributes

- TDT4102::Button **StartGameBtn**
- TDT4102::Button **HighscoresBtn**
- TDT4102::Button **SettingsBtn**
- TDT4102::Button **EndGameBtn**
- TDT4102::Button **GoToMenuBtn**
- [SpaceShipPlayer](#) **playerShip**
- std::vector< [SpaceShipEnemy](#) > **enemyShips**
- std::vector< std::unique_ptr< [Weapon](#) > > **firedWeapons**

5.10.1 Detailed Description

The main game class which runs the game. Uses AnimationWindow as base class.

Parameters

<i>currentScreen</i>	Pointer to the current screen
<i>btnWidth</i>	Width of the buttons. Relative to window width
<i>btnHeight</i>	Height of the buttons. Relative to window height
<i>playerShip</i>	PlayerShip object
<i>enemyShips</i>	Vector of EnemyShip objects
<i>firedWeapons</i>	Vector of Weapon objects

5.10.2 Constructor & Destructor Documentation

5.10.2.1 SpaceDefender()

```
SpaceDefender::SpaceDefender (
    TDT4102::Point position = {100, 100},
    int width = 600,
    int height = 800,
    const std::string & title = "Space Defender")
```

Construct a new [SpaceDefender::SpaceDefender](#) object.

Parameters

<i>position</i>	Position of where the window starts in upper left corner
<i>width</i>	The width of the window
<i>height</i>	The height of the window
<i>title</i>	The title of the window
<i>numEnemiesHeight</i>	The number of enemies in the height of the window
<i>numEnemiesWidth</i>	The number of enemies in the width of the window

Todo Fix how enemies spawn

Todo Need dynamic

5.10.3 Member Function Documentation

5.10.3.1 run()

```
void SpaceDefender::run ()
```

Game loop that runs the game until the window is closed.

Updates the game state and draws the current screen

Parameters

<i>currentScreen</i>	Draws the current screen as long as its not a null pointer
----------------------	--

5.10.3.2 setScreen()

```
void SpaceDefender::setScreen (  
    std::unique_ptr< Screen > newScreen)
```

Set the current screen.

Replaces the current screen with `std::move(newScreen)` of the `unique_ptr<Screen>`

Parameters

<i>newScreen</i>	The new screen that we want to point to
<i>currentScreen</i>	The current screen that the pointer points to

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

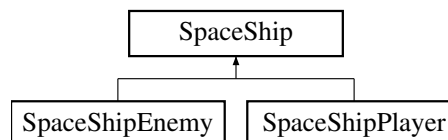
- [SpaceDefender.h](#)
- [SpaceDefender.cpp](#)

5.11 SpaceShip Class Reference

Abstract base class for different spaceships.

```
#include <SpaceShip.h>
```

Inheritance diagram for SpaceShip:



Public Member Functions

- [SpaceShip](#) (int startX, int startY, int startHealth)
Constructor that initializes x, y and health.
- virtual [~SpaceShip](#) ()=default
- virtual void [movements](#) ([SpaceDefender](#) &>window)=0
- virtual void [shooting](#) ([SpaceDefender](#) &>window)=0
- void [healthReduction](#) ()
- int [getHealth](#) () const
- int [getPositionX](#) () const
- int [getPositionY](#) () const
- int [getShipHeight](#) () const
- int [getShipWidth](#) () const
- void [setShipSpeed](#) (const int &newSpeed)

Protected Attributes

- int **x**
- int **y**
- int **health**
- const int **shipHeight** = 20
- const int **shipWidth** = 20
- int **shipSpeed**

5.11.1 Detailed Description

Abstract base class for different spaceships.

Parameters

<i>window</i>	SpaceDefender object
<i>x</i>	Position in the x-axis
<i>y</i>	Position in the y-axis
<i>health</i>	Health of the ship
<i>shipHeight</i>	Height of the ship
<i>shipWidth</i>	Width of the ship
<i>shipSpeed</i>	Speed of the ship

5.11.2 Constructor & Destructor Documentation

5.11.2.1 SpaceShip()

```
SpaceShip::SpaceShip (
    int startX,
    int startY,
    int startHealth) [inline]
```

Constructor that initializes x, y and health.

Parameters

<i>startX</i>	
<i>startY</i>	
<i>startHealth</i>	

5.11.2.2 ~SpaceShip()

```
virtual SpaceShip::~SpaceShip () [virtual], [default]
```

Virtual destructor to ensure proper cleanup

5.11.3 Member Function Documentation

5.11.3.1 getHealth()

```
int SpaceShip::getHealth () const [inline]
```

Getter for health

5.11.3.2 getPositionX()

```
int SpaceShip::getPositionX () const [inline]
```

Getter for position in x-axis

5.11.3.3 getPositionY()

```
int SpaceShip::getPositionY () const [inline]
```

Getter for position in y-axis

5.11.3.4 getShipHeight()

```
int SpaceShip::getShipHeight () const [inline]
```

Getter for ship height

5.11.3.5 getShipWidth()

```
int SpaceShip::getShipWidth () const [inline]
```

Getter for ship width

5.11.3.6 healthReduction()

```
void SpaceShip::healthReduction () [inline]
```

Reduces the health of the ship

5.11.3.7 movements()

```
virtual void SpaceShip::movements (  
    SpaceDefender & window) [pure virtual]
```

Pure virtual function. Is supposed to move the spaceship

Implemented in [SpaceShipEnemy](#), and [SpaceShipPlayer](#).

5.11.3.8 setShipSpeed()

```
void SpaceShip::setShipSpeed (  
    const int & newSpeed) [inline]
```

Setter for ship speed

5.11.3.9 shooting()

```
virtual void SpaceShip::shooting (  
    SpaceDefender & window) [pure virtual]
```

Pure virtual function. Is supposed do shooting

Implemented in [SpaceShipEnemy](#), and [SpaceShipPlayer](#).

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

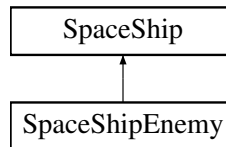
- [SpaceShip.h](#)

5.12 SpaceShipEnemy Class Reference

Class for enemy ship.

```
#include <SpaceShip.h>
```

Inheritance diagram for SpaceShipEnemy:



Public Member Functions

- **SpaceShipEnemy** (int startX, int startY)
- void **movements** ([SpaceDefender](#) &>window) override
Moves the spaceship of the enemy.
- void **shooting** ([SpaceDefender](#) &>window) override
Fires the weapon at some interval.

Public Member Functions inherited from [SpaceShip](#)

- [SpaceShip](#) (int startX, int startY, int startHealth)
Constructor that initializes x, y and health.
- virtual [~SpaceShip](#) ()=default
- void [healthReduction](#) ()
- int [getHealth](#) () const
- int [getPositionX](#) () const
- int [getPositionY](#) () const
- int [getShipHeight](#) () const
- int [getShipWidth](#) () const
- void [setShipSpeed](#) (const int &newSpeed)

Public Attributes

- TDT4102::Image **alienImage**

Additional Inherited Members

Protected Attributes inherited from [SpaceShip](#)

- int **x**
- int **y**
- int **health**
- const int **shipHeight** = 20
- const int **shipWidth** = 20
- int **shipSpeed**

5.12.1 Detailed Description

Class for enemy ship.

Parameters

<i>alienImage</i>	Image of the alienship
-------------------	------------------------

5.12.2 Member Function Documentation

5.12.2.1 movements()

```
void SpaceShipEnemy::movements (
    SpaceDefender & window) [override], [virtual]
```

Moves the spaceship of the enemy.

Make a bullet, fire the and stores it in a vector

Parameters

<i>window</i>	SpaceDefender object
---------------	--------------------------------------

Todo Fix the movement of the enemy

Implements [SpaceShip](#).

5.12.2.2 shooting()

```
void SpaceShipEnemy::shooting (
    SpaceDefender & window) [override], [virtual]
```

Fires the weapon at some interval.

A random enemy fires a bullet and stores it in a vector

Parameters

<i>window</i>	SpaceDefender object
---------------	--------------------------------------

Todo Fix shooting of the enemy

- Method to find lowest ship in each column

- One of said ships fire randomly

- Firerate approximatly same as playership

Implements [SpaceShip](#).

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

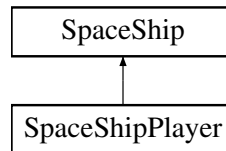
- [SpaceShip.h](#)
- [SpaceShip.cpp](#)

5.13 SpaceShipPlayer Class Reference

Class for player ship.

```
#include <SpaceShip.h>
```

Inheritance diagram for SpaceShipPlayer:



Public Member Functions

- **SpaceShipPlayer** (int startX, int startY)
- void **movements** ([SpaceDefender](#) &>window) override
Move the spaceship in x-axis using the arrow keys.
- void **shooting** ([SpaceDefender](#) &>window) override
Fires the weapon if the space key is pressed.

Public Member Functions inherited from [SpaceShip](#)

- [SpaceShip](#) (int startX, int startY, int startHealth)
Constructor that initializes x, y and health.
- virtual [~SpaceShip](#) ()=default
- void [healthReduction](#) ()
- int [getHealth](#) () const
- int [getPositionX](#) () const
- int [getPositionY](#) () const
- int [getShipHeight](#) () const
- int [getShipWidth](#) () const
- void [setShipSpeed](#) (const int &newSpeed)

Public Attributes

- TDT4102::Image **playerImage**

Additional Inherited Members

Protected Attributes inherited from [SpaceShip](#)

- int **x**
- int **y**
- int **health**
- const int **shipHeight** = 20
- const int **shipWidth** = 20
- int **shipSpeed**

5.13.1 Detailed Description

Class for player ship.

Parameters

<i>playerImage</i>	Image of the playership
<i>lastShotTime</i>	The time of the last
<i>fireRate</i>	How often the weapon can be fired

Todo firerate in regards to different weapomn types

5.13.2 Member Function Documentation

5.13.2.1 movements()

```
void SpaceShipPlayer::movements (
    SpaceDefender & window) [override], [virtual]
```

Move the spaceship in x-axis using the arrow keys.

Parameters

<i>window</i>	SpaceDefender object
<i>x</i>	Position of the spaceship in the x-axis

Implements [SpaceShip](#).

5.13.2.2 shooting()

```
void SpaceShipPlayer::shooting (
    SpaceDefender & window) [override], [virtual]
```

Fires the weapon if the space key is pressed.

Fires a bullet, fire the and stores it in a vector

Parameters

<i>window</i>	SpaceDefender object
<i>newBullet</i>	Creates a new Bullet
<i>new</i>	The time at the point when the function is called

Todo Only bullets are fired, consider making it more general. For example having a set weapon type

Implements [SpaceShip](#).

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

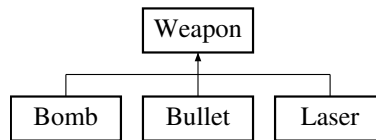
- [SpaceShip.h](#)
- [SpaceShip.cpp](#)

5.14 Weapon Class Reference

Abstract base class for different weapons.

```
#include <Weapon.h>
```

Inheritance diagram for Weapon:



Public Member Functions

- [Weapon](#) (int speed, int damage)
- virtual [~Weapon](#) ()=default
- virtual void [fireWeapon](#) ([SpaceDefender](#) &>window)=0
- int [getSpeed](#) ()
- int [getDamage](#) ()
- virtual void [move](#) ()
- virtual void [draw](#) ([SpaceDefender](#) &>window)=0

Protected Attributes

- int **speed**
- int **damage**
- int **xProjectile**
- int **yProjectile**

5.14.1 Detailed Description

Abstract base class for different weapons.

Parameters

<i>speed</i>	Speed of the projectile
<i>damage</i>	Damage the projectile does
<i>xProjectile</i>	Position of the projectile in the x-axis
<i>yProjectile</i>	Position of the projectile in the y-axis
<i>window</i>	SpaceDefender object

5.14.2 Constructor & Destructor Documentation

5.14.2.1 Weapon()

```
Weapon::Weapon (
    int speed,
    int damage) [inline]
```

Constructor that initializes speed and damage

5.14.2.2 ~Weapon()

```
virtual Weapon::~Weapon () [virtual], [default]
```

Virtual destructor to ensure proper cleanup

5.14.3 Member Function Documentation

5.14.3.1 draw()

```
virtual void Weapon::draw (  
    SpaceDefender & window) [pure virtual]
```

Pure virtual function. Is supposed to draw the projectile

Implemented in [Bullet](#).

5.14.3.2 fireWeapon()

```
virtual void Weapon::fireWeapon (  
    SpaceDefender & window) [pure virtual]
```

Pure virtual function. Is supposed to fire the projectile, aka get the position when fired.

Implemented in [Bomb](#), [Bullet](#), and [Laser](#).

5.14.3.3 getDamage()

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

Getter for damage

5.14.3.4 getSpeed()

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

Getter for speed

5.14.3.5 move()

```
virtual void Weapon::move () [inline], [virtual]
```

Move the projectile in y-axis

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

- [Weapon.h](#)

Chapter 6

File Documentation

6.1 main.cpp File Reference

Main file.

```
#include "std_lib_facilities.h"
#include "SpaceDefender.h"
#include <iostream>
#include <fstream>
#include <nlohmann/json.hpp>
```

Typedefs

- using **json** = nlohmann::json

Functions

- int **main** ()
Starts the game.

6.1.1 Detailed Description

Main file.

Author

Tor Gunnar Ravatn Hammer (tor.ravatn@gmail.com)
Gabriel Anton Norheim ()

Version

1.0

Date

2025-04-01

Copyright

Copyright (c) 2025

6.1.2 Function Documentation

6.1.2.1 main()

```
int main ()
```

Starts the game.

Parameters

<code>game</code>	<code>SpaceDefender</code> object
-------------------	-----------------------------------

Returns

Returns 0 on success

6.2 Screen.cpp File Reference

The cpp file for the `Screen` class.

```
#include "Screen.h"
#include "SpaceDefender.h"
#include <iostream>
#include <fstream>
#include <sstream>
#include <iomanip>
#include <nlohmann/json.hpp>
```

Functions

- `std::vector< Player > readScores` (const std::string &filename)
Reads the highscores from a json file.
- `std::string formatPlayerInfo` (const [Player](#) &player)
Function to format the text that is drawn for each highscore.

6.2.1 Detailed Description

The cpp file for the `Screen` class.

Author

Tor Gunnar Ravatn Hammer (tor.ravatn@gmail.com)

Version

1.0

Date

2025-04-01

Copyright

Copyright (c) 2025

6.2.2 Function Documentation

6.2.2.1 formatPlayerInfo()

```
std::string formatPlayerInfo (  
    const Player & player)
```

Function to format the text that is drawn for each highscore.

Parameters

<code>player</code>	A single input of the Player struct
---------------------	---

Returns

std::string of the formatted player info

Todo Make better, width depends on characters

6.2.2.2 readScores()

```
std::vector< Player > readScores (  
    const std::string & filename)
```

Reads the highscores from a json file.

Parameters

<code>filename</code>	The name of the json file to be read
-----------------------	--------------------------------------

Returns

std::vector<Player>

6.3 Screen.h File Reference

The header file for the [Screen](#) class.

```
#include <string>
```

Classes

- struct [Player](#)
An object representing a player in regards to highscores.
- class [Screen](#)
Abstract base class for different screens.
- class [ScreenMenu](#)
The menu screen.
- class [ScreenGame](#)
The game screen.
- class [ScreenHighscore](#)
The highscore screen.
- class [ScreenSettings](#)
The settings screen.

Functions

- `std::vector< Player > readScores` (const std::string &filename="highscores.json")
Reads the highscores from a json file.

6.3.1 Detailed Description

The header file for the [Screen](#) class.

Author

Tor Gunnar Ravatn Hammer (tor.ravatn@gmail.com)

Version

1.0

Date

2025-04-01

Copyright

Copyright (c) 2025

6.3.2 Function Documentation

6.3.2.1 readScores()

```
std::vector< Player > readScores (  
    const std::string & filename)
```

Reads the highscores from a json file.

Parameters

<i>filename</i>	The name of the json file to be read
-----------------	--------------------------------------

Returns

std::vector<Player>

6.4 Screen.h

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

```

00001
00011 #pragma once
00012 #include <string>
00013
00014
00024 struct Player {
00025     std::string rank;
00026     std::string name;
00027     int score;
00028     int round;
00029 };
00030
00031 std::vector<Player> readScores(const std::string& filename = "highscores.json");
00032
00033 class SpaceDefender; // Forward declaration of SpaceDefender to avoid circular dependency
00034
00040 class Screen {
00041 public:
00042     virtual ~Screen() = default;
00043     virtual void draw(SpaceDefender& window) = 0;
00044 };
00045
00046
00051 class ScreenMenu : public Screen {
00052 public:
00053     void draw(SpaceDefender& window) override;
00054 };
00055
00056
00061 class ScreenGame : public Screen {
00062 public:
00063     void draw(SpaceDefender& window) override;
00064 };
00065
00066
00071 class ScreenHighscore : public Screen {
00072 public:
00073     void draw(SpaceDefender& window) override;
00074 };
00075
00076
00081 class ScreenSettings : public Screen {
00082 public:
00083     void draw(SpaceDefender& window) override;
00084 };

```

6.5 SpaceDefender.cpp File Reference

The cpp file for the [SpaceDefender](#) class.

```

#include "SpaceDefender.h"
#include <iostream>

```

6.5.1 Detailed Description

The cpp file for the [SpaceDefender](#) class.

Author

Tor Gunnar Ravatn Hammer (tor.ravatn@gmail.com)
Gabriel Anton Norheim ()

Version

1.0

Date

2025-04-01

Copyright

Copyright (c) 2025

6.6 SpaceDefender.h File Reference

The header file for the [SpaceDefender](#) class.

```
#include "AnimationWindow.h"  
#include "widgets/Button.h"  
#include "Screen.h"  
#include "SpaceShip.h"  
#include "Weapon.h"
```

Classes

- class [SpaceDefender](#)

The main game class which runs the game. Uses AnimationWindow as base class.

6.6.1 Detailed Description

The header file for the [SpaceDefender](#) class.

Author

Tor Gunnar Ravatn Hammer (tor.ravatn@gmail.com)

Version

1.0

Date

2025-04-01

Copyright

Copyright (c) 2025

6.7 SpaceDefender.h

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

```

00001
00011 #pragma once
00012
00013 #include "AnimationWindow.h"
00014 #include "widgets/Button.h"
00015
00016 #include "Screen.h"
00017 #include "SpaceShip.h"
00018 #include "Weapon.h"
00019
00031 class SpaceDefender : public TDT4102::AnimationWindow {
00032 private:
00033     std::unique_ptr<Screen> currentScreen;
00034     unsigned int btnWidth;
00035     unsigned int btnHeight;
00036
00037     // Callback functions for buttons
00038     void cb_endGame() {close();}
00039     void cb_startGame() {setScreen(std::make_unique<ScreenGame>());}
00040     void cb_showHighscores() {setScreen(std::make_unique<ScreenHighscore>());}
00041     void cb_settings() {setScreen(std::make_unique<ScreenSettings>());}
00042     void cb_menu() {setScreen(std::make_unique<ScreenMenu>()); }
00043
00044 public:
00045     SpaceDefender(TDT4102::Point position = {100, 100}, int width = 600, int height = 800, const
std::string& title = "Space Defender");
00046     void setScreen(std::unique_ptr<Screen> newScreen);
00047     void run();
00048
00049     // Buttons
00050     TDT4102::Button StartGameBtn;
00051     TDT4102::Button HighscoresBtn;
00052     TDT4102::Button SettingsBtn;
00053     TDT4102::Button EndGameBtn;
00054     TDT4102::Button GoToMenuBtn;
00055
00056     // Spaceships
00057     SpaceShipPlayer playerShip;
00058     std::vector<SpaceShipEnemy> enemyShips;
00059
00060     // Weapons
00061     std::vector<std::unique_ptr<Weapon>> firedWeapons;
00062 };

```

6.8 SpaceShip.cpp File Reference

The cpp file for the [SpaceShip](#) class.

```

#include "SpaceShip.h"
#include "SpaceDefender.h"
#include <iostream>

```

6.8.1 Detailed Description

The cpp file for the [SpaceShip](#) class.

Author

Tor Gunnar Ravatn Hammer (tor.ravatn@gmail.com)

Version

1.0

Date

2025-04-01

Copyright

Copyright (c) 2025

6.9 SpaceShip.h File Reference

The header file for the [SpaceShip](#) class.

```
#include <chrono>
#include "subprojects/animationwindow/include/Image.h"
```

Classes

- class [SpaceShip](#)
Abstract base class for different spaceships.
- class [SpaceShipPlayer](#)
Class for player ship.
- class [SpaceShipEnemy](#)
Class for enemy ship.

6.9.1 Detailed Description

The header file for the [SpaceShip](#) class.

Author

Tor Gunnar Ravatn Hammer (tor.ravatn@gmail.com)

Version

1.0

Date

2025-04-01

Copyright

Copyright (c) 2025

6.10 SpaceShip.h

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

```

00001
00011 #pragma once
00012 #include <chrono>
00013 #include "subprojects/animationwindow/include/Image.h" //for image type
00014
00015 class SpaceDefender; // Forward declaration of SpaceDefender to avoid circular dependency
00016
00029 class SpaceShip {
00030 public:
00037     SpaceShip(int startX, int startY, int startHealth) : x(startX), y(startY), health(startHealth) {}
00038     virtual ~SpaceShip() = default;
00039     virtual void movements(SpaceDefender& window) = 0;
00040     virtual void shooting(SpaceDefender& window) = 0;
00041     void healthReduction() { --health; }
00042     int getHealth() const {return health;}
00043     int getPositionX() const {return x;}
00044     int getPositionY() const {return y;}
00045     int getShipHeight() const {return shipHeight;}
00046     int getShipWidth() const {return shipWidth;}
00047     void setShipSpeed(const int &newSpeed) {this->shipSpeed = newSpeed;}
00048
00049 protected:
00050     int x;
00051     int y;
00052     int health;
00053     const int shipHeight = 20;
00054     const int shipWidth = 20;
00055     int shipSpeed;
00056 };
00057
00066 class SpaceShipPlayer : public SpaceShip {
00067 public:
00068     SpaceShipPlayer(int startX, int startY) :
00069         SpaceShip(startX, startY, 3),
00070         playerImage("bilder/ShipSprite.png") {
00071         setShipSpeed(10);
00072     }
00073     void movements(SpaceDefender& window) override;
00074     void shooting(SpaceDefender& window) override;
00075     TDT4102::Image playerImage;
00076 private:
00077     std::chrono::steady_clock::time_point lastShotTime;
00078     const std::chrono::milliseconds fireRate = std::chrono::milliseconds(500);
00079 };
00080
00081
00087 class SpaceShipEnemy : public SpaceShip {
00088 public:
00089     SpaceShipEnemy(int startX, int startY) :
00090         SpaceShip(startX, startY, 1),
00091         alienImage("bilder/aillenHead.png") {}
00092     void movements(SpaceDefender& window) override;
00093     void shooting(SpaceDefender& window) override;
00094     TDT4102::Image alienImage;
00095 };

```

6.11 Weapon.cpp File Reference

The cpp file for the [Weapon](#) class.

```

#include "Weapon.h"
#include "SpaceDefender.h"

```

6.11.1 Detailed Description

The cpp file for the [Weapon](#) class.

Author

Tor Gunnar Ravatn Hammer (tor.ravatn@gmail.com)

Version

1.0

Date

2025-04-01

Copyright

Copyright (c) 2025

6.12 Weapon.h File Reference

The header file for the [Weapon](#) class.

Classes

- class [Weapon](#)
Abstract base class for different weapons.
- class [Bullet](#)
Class for [Bullet](#). Inherits from [Weapon](#).
- class [Bomb](#)
Class for [Bomb](#). Inherits from [Weapon](#).
- class [Laser](#)
Class for [Laser](#). Inherits from [Weapon](#).

6.12.1 Detailed Description

The header file for the [Weapon](#) class.

Author

Tor Gunnar Ravatn Hammer (tor.ravatn@gmail.com)

Version

1.0

Date

2025-04-01

Copyright

Copyright (c) 2025

6.13 Weapon.h

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

```
00001
00011 #pragma once
00012
00013 class SpaceDefender; // Forward declaration of SpaceDefender to avoid circular dependency
00014
00024 class Weapon {
00025 public:
00026     Weapon(int speed, int damage) : speed(speed), damage(damage) {}
00027     virtual ~Weapon() = default;
00028     virtual void fireWeapon(SpaceDefender& window) = 0;
00029     int getSpeed() {return speed;}
00030     int getDamage() {return damage;}
00031     virtual void move() {yProjectile -= speed;}
00032     virtual void draw(SpaceDefender& window) = 0;
00033
00034 protected:
00035     int speed;
00036     int damage;
00037     int xProjectile;
00038     int yProjectile;
00039 };
00040
00046 class Bullet : public Weapon {
00047 public:
00048     Bullet(int speed, int damage) : Weapon(speed, damage) {}
00049     void fireWeapon(SpaceDefender& window) override;
00050     void draw(SpaceDefender& window) override;
00051 private:
00052     int radius = 5;
00053 };
00054
00060 class Bomb : public Weapon {
00061 public:
00062     Bomb(int speed, int damage) : Weapon(speed, damage) {}
00063     void fireWeapon(SpaceDefender& window) override;
00064 };
00065
00071 class Laser : public Weapon {
00072 public:
00073     Laser(int speed, int damage) : Weapon(speed, damage) {}
00074     void fireWeapon(SpaceDefender& window) override;
00075 };
```

