Space Defender

Generated by Doxygen 1.13.2

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:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

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

Chapter 5

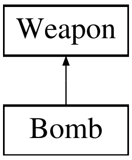
Class Documentation

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

window	

Implements Weapon.

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

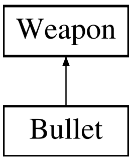
- Weapon.h
- Weapon.cpp

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

radius	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

window	SpaceDefender object
radius	Radius of the bullet
location	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

window	SpaceDefender object
xProjectile	Position of the projectile in the x-axis
yProjectile	Position of the projectile in the y-axis

Implements Weapon.

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

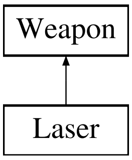
- Weapon.h
- Weapon.cpp

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

window	

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

rank	The rank of the player
name	The name of the player
score	The score of the player
round	The round of the player

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

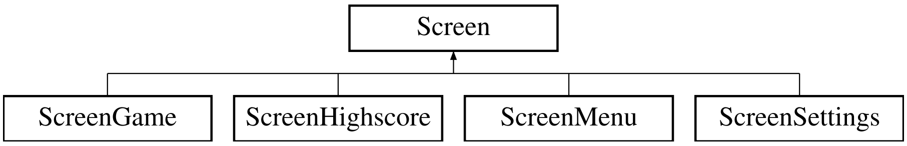
- Screen.h

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

window	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:

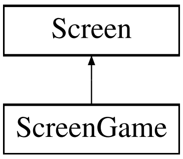
- Screen.h

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

window	SpaceDefender object

Implements Screen.

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

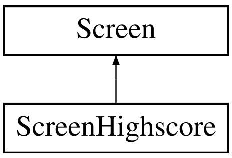
- Screen.h
- Screen.cpp

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

window	SpaceDefender object
file	The json file variable which contains the highscores

Implements Screen.

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

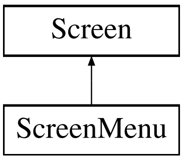
- Screen.h
- Screen.cpp

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

window	SpaceDefender object

Implements Screen.

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

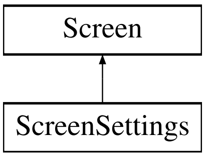
- Screen.h
- Screen.cpp

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

window	SpaceDefender object

Implements Screen.

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

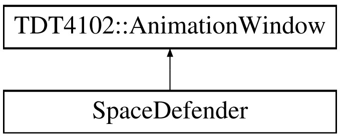
- Screen.h
- Screen.cpp

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

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

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

currentScreen	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

newScreen	The new screen that we want to point to
currentScreen	The current screen that the pointer points to

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

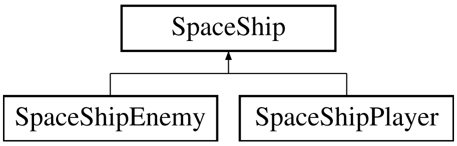
- SpaceDefender.h
- SpaceDefender.cpp

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

window	SpaceDefender object
x	Position in the x-axis
y	Position in the y-axis
health	Health of the ship
shipHeight	Height of the ship
shipWidth	Width of the ship
shipSpeed	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

startX	
startY	
startHealth	

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:

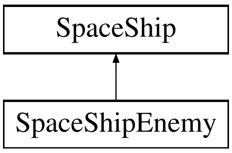
- SpaceShip.h

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

alienImage	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

window	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 enenmy fires a bullet and stores it in a vector

Parameters

window	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:

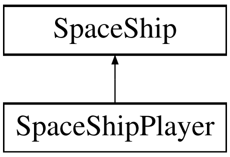
- SpaceShip.h
- SpaceShip.cpp

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

playerImage	Image of the playership
lastShotTime	The time of the last
fireRate	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

window	SpaceDefender object
x	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

window	SpaceDefender object
newBullet	Creates a new Bullet
new	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:

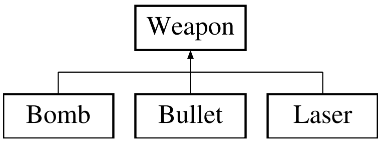
- SpaceShip.h
- SpaceShip.cpp

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

speed	Speed of the projectile
damage	Damage the projectile does
xProjectile	Position of the projectile in the x-axis
yProjectile	Position of the projectile in the y-axis
window	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 postion 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

6.1.2 Function Documentation

6.1.2.1 main()

```
int main ()
```

Starts the game.

Parameters

game	SpaceDefender 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

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

player	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

filename	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

filename	The name of the json file to be read

Returns

std::vector<Player>

6.4 Screen.h

Go to the documentation of this file.

```

#pragma once
#include <string>


struct Player {
    std::string rank;
    std::string name;
    int score;
    int round;
};

std::vector<Player> readScores(const std::string& filename = "highscores.json");

class SpaceDefender;  // Forward declaration of SpaceDefender to avoid circular dependency

class Screen {
public:
    virtual ~Screen() = default;
    virtual void draw(SpaceDefender& window) = 0;
};


class ScreenMenu : public Screen {
public:
    void draw(SpaceDefender& window) override;
};


class ScreenGame : public Screen {
public:
    void draw(SpaceDefender& window) override;
};


class ScreenHighscore : public Screen {
public:
    void draw(SpaceDefender& window) override;
};


class ScreenSettings : public Screen {
public:
    void draw(SpaceDefender& window) override;
};
```

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.

```

#pragma once

#include "AnimationWindow.h"
#include "widgets/Button.h"

#include "Screen.h"
#include "SpaceShip.h"
#include "Weapon.h"

class SpaceDefender : public TDT4102::AnimationWindow {
private:
    std::unique_ptr<Screen> currentScreen;
    unsigned int btnWidth;
    unsigned int btnHeight;

    // Callback functions for buttons
    void cb_endGame()        {close();}
    void cb_startGame()      {setScreen(std::make_unique<ScreenGame>());}
    void cb_showHighscores() {setScreen(std::make_unique<ScreenHighscore>());}
    void cb_settings()       {setScreen(std::make_unique<ScreenSettings>());}
    void cb_menu()           {setScreen(std::make_unique<ScreenMenu>()); }

public:
    SpaceDefender(TDT4102::Point position = {100, 100}, int width = 600, int height = 800, const std::string& title = "Space Defender");
    void setScreen(std::unique_ptr<Screen> newScreen);
    void run();

    // Buttons
    TDT4102::Button StartGameBtn;
    TDT4102::Button HighscoresBtn;
    TDT4102::Button SettingsBtn;
    TDT4102::Button EndGameBtn;
    TDT4102::Button GoToMenuBtn;

    // Spaceships
    SpaceShipPlayer playerShip;
    std::vector<SpaceShipEnemy> enemyShips;

    // Weapons
    std::vector<std::unique_ptr<Weapon» firedWeapons;
};
```

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.

```

#pragma once
#include <chrono>
#include "subprojects/animationwindow/include/Image.h" //for image type

class SpaceDefender;  // Forward declaration of SpaceDefender to avoid circular dependency

class SpaceShip {
public:
    SpaceShip(int startX, int startY, int startHealth) : x(startX), y(startY), health(startHealth) {}
    virtual ~SpaceShip() = default;
    virtual void movements(SpaceDefender& window) = 0;
    virtual void shooting(SpaceDefender& window) = 0;
    void healthReduction() { --health; }
    int getHealth() const {return health;}
    int getPositionX() const {return x;}
    int getPositionY() const {return y;}
    int getShipHeight() const {return shipHeight;}
    int getShipWidth() const {return shipWidth;}
    void setShipSpeed(const int &newSpeed) {this->shipSpeed = newSpeed;}

protected:
    int x;
    int y;
    int health;
    const int shipHeight = 20;
    const int shipWidth = 20;
    int shipSpeed;
};

class SpaceShipPlayer : public SpaceShip {
    public:
        SpaceShipPlayer(int startX, int startY) :
            SpaceShip(startX, startY, 3),
            playerImage("bilder/ShipSprite.png") {
            setShipSpeed(10);
        }
        void movements(SpaceDefender& window) override;
        void shooting(SpaceDefender& window) override;
        TDT4102::Image playerImage;
    private:
        std::chrono::steady_clock::time_point lastShotTime;
        const std::chrono::milliseconds fireRate = std::chrono::milliseconds(500);
};


class SpaceShipEnemy : public SpaceShip {
    public:
        SpaceShipEnemy(int startX, int startY) :
            SpaceShip(startX, startY, 1),
            alienImage("bilder/ailienHead.png") {}
        void movements(SpaceDefender& window) override;
        void shooting(SpaceDefender& window) override;
        TDT4102::Image alienImage;
    };
```

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.

```
#pragma once

class SpaceDefender;  // Forward declaration of SpaceDefender to avoid circular dependency

class Weapon {
public:
    Weapon(int speed, int damage) : speed(speed), damage(damage) {}
    virtual ~Weapon() = default;
    virtual void fireWeapon(SpaceDefender& window) = 0;
    int getSpeed() {return speed;}
    int getDamage() {return damage;}
    virtual void move() {yProjectile -=speed;}
    virtual void draw(SpaceDefender& window) = 0;

protected:
    int speed;
    int damage;
    int xProjectile;
    int yProjectile;
};

class Bullet : public Weapon {
public:
    Bullet(int speed, int damage) : Weapon(speed, damage) {}
    void fireWeapon(SpaceDefender& window) override;
    void draw(SpaceDefender& window) override;
private:
    int radius = 5;
};

class Bomb : public Weapon {
public:
    Bomb(int speed, int damage) : Weapon(speed, damage) {}
    void fireWeapon(SpaceDefender& window) override;
};

class Laser : public Weapon {
public:
    Laser(int speed, int damage) : Weapon(speed, damage) {}
    void fireWeapon(SpaceDefender& window) override;
};
```