

Ssssnahe

Generated by Doxygen 1.8.17

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 Apple Class Reference	7
4.1.1 Detailed Description	8
4.1.2 Constructor & Destructor Documentation	8
4.1.2.1 Apple()	8
4.1.2.2 ~Apple()	8
4.1.3 Member Function Documentation	8
4.1.3.1 draw()	8
4.1.3.2 position()	9
4.1.3.3 update()	9
4.2 Background Class Reference	9
4.2.1 Constructor & Destructor Documentation	10
4.2.1.1 Background()	10
4.2.1.2 ~Background()	10
4.2.2 Member Function Documentation	10
4.2.2.1 draw()	10
4.3 background Class Reference	11
4.3.1 Detailed Description	11
4.4 Snake::Destroy Struct Reference	11
4.4.1 Detailed Description	11
4.4.2 Member Function Documentation	11
4.4.2.1 operator()()	11
4.5 Game Class Reference	12
4.5.1 Detailed Description	13
4.5.2 Member Enumeration Documentation	13
4.5.2.1 State	13
4.5.3 Member Function Documentation	13
4.5.3.1 initialize()	13
4.5.3.2 loop()	13
4.6 HUD Class Reference	14
4.6.1 Detailed Description	14
4.6.2 Constructor & Destructor Documentation	15
4.6.2.1 HUD()	15
4.6.2.2 ~HUD()	15

4.6.3 Member Function Documentation	15
4.6.3.1 operator=()	15
4.6.3.2 showScore()	15
4.7 intro Class Reference	16
4.7.1 Detailed Description	16
4.7.2 Constructor & Destructor Documentation	16
4.7.2.1 intro()	16
4.7.2.2 ~intro()	16
4.7.3 Member Function Documentation	16
4.7.3.1 show()	16
4.8 Object Class Reference	17
4.8.1 Detailed Description	18
4.8.2 Constructor & Destructor Documentation	18
4.8.2.1 Object()	18
4.8.2.2 ~Object()	18
4.8.3 Member Function Documentation	18
4.8.3.1 draw()	18
4.8.3.2 load()	18
4.8.3.3 update()	19
4.9 Snake Class Reference	19
4.9.1 Constructor & Destructor Documentation	21
4.9.1.1 Snake()	21
4.9.1.2 ~Snake()	21
4.9.2 Member Function Documentation	21
4.9.2.1 apple()	22
4.9.2.2 bounds()	22
4.9.2.3 collision()	22
4.9.2.4 draw()	22
4.9.2.5 GetScore()	22
4.9.2.6 input()	22
4.9.2.7 positions()	23
4.9.2.8 reset()	23
4.9.2.9 update()	23
4.10 Snake_Body Class Reference	23
4.10.1 Detailed Description	24
4.10.2 Constructor & Destructor Documentation	24
4.10.2.1 Snake_Body()	24
4.10.2.2 ~Snake_Body()	25
4.10.3 Member Function Documentation	25
4.10.3.1 draw()	25
4.10.3.2 operator=()	25
4.10.3.3 operator==()	25

5 File Documentation	27
5.1 Apple.h File Reference	27
5.1.1 Detailed Description	28
5.2 Background.h File Reference	28
5.2.1 Detailed Description	29
5.3 Game.h File Reference	29
5.3.1 Detailed Description	30
5.4 HUD.h File Reference	31
5.4.1 Detailed Description	32
5.5 intro.h File Reference	32
5.5.1 Detailed Description	33
5.6 Object.h File Reference	33
5.6.1 Detailed Description	34
5.7 Snake.h File Reference	34
5.7.1 Detailed Description	35
5.8 Snake_Body.h File Reference	36
5.8.1 Detailed Description	36
Index	37

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

background	11
Snake::Destroy	11
Game	12
HUD	14
intro	16
Object	17
Apple	7
Background	9
Snake	19
Snake_Body	23

Chapter 2

Class Index

2.1 Class List

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

Apple	7
Background	9
background	11
Snake::Destroy	11
Game	12
HUD	14
intro	16
Object	17
Snake	19
Snake_Body	23

Chapter 3

File Index

3.1 File List

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

Apple.h	Object for eating class	27
Background.h	Rendered window class	28
Game.h	Game class	29
HUD.h	HUD class	31
intro.h	Intro handling file	32
Object.h	Object class	33
Snake.h	Snake class file	34
Snake_Body.h	File handling body of the snake	36

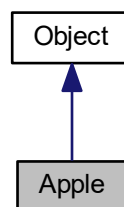
Chapter 4

Class Documentation

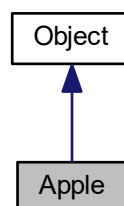
4.1 Apple Class Reference

```
#include <Apple.h>
```

Inheritance diagram for Apple:



Collaboration diagram for Apple:



Public Member Functions

- [Apple](#) ()
- [~Apple](#) ()
- void [update](#) () override
- void [draw](#) (sf::RenderWindow &window) override
- bool [GetEaten](#) () const
getter
- void [SetEaten](#) (bool a)

Private Member Functions

- void [position](#) ()

Private Attributes

- bool [_wasEaten](#)

Additional Inherited Members

4.1.1 Detailed Description

[Apple](#) Class handles functions for objects designed to be eaten by the snake. Public inheritance from [Object](#) class.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Apple()

```
Apple::Apple ( )
```

Constructor

4.1.2.2 ~Apple()

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

Destructor

4.1.3 Member Function Documentation

4.1.3.1 draw()

```
void Apple::draw (
    sf::RenderWindow & window ) [override], [virtual]
```

Draws sprite to the game window.

Parameters

<i>window</i>	render window.
---------------	----------------

Reimplemented from [Object](#).

4.1.3.2 position()

```
void Apple::position ( ) [private]
```

Sets place for the apple inside the board.

4.1.3.3 update()

```
void Apple::update ( ) [override], [virtual]
```

Updates state of the game.

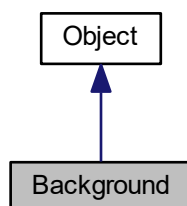
Reimplemented from [Object](#).

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

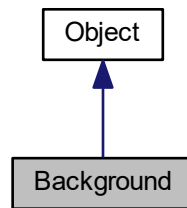
- [Apple.h](#)

4.2 Background Class Reference

Inheritance diagram for Background:



Collaboration diagram for Background:



Public Member Functions

- [Background](#) ()
- [~Background](#) ()
- void [draw](#) (sf::RenderWindow &window) override

Additional Inherited Members

4.2.1 Constructor & Destructor Documentation

4.2.1.1 Background()

```
Background::Background ( )
```

Constructor

4.2.1.2 ~Background()

```
Background::~~Background ( )
```

Destructor

4.2.2 Member Function Documentation

4.2.2.1 draw()

```
void Background::draw (
    sf::RenderWindow & window ) [override], [virtual]
```

Draws sprite to the render window.

Parameters

<i>window</i>	render window.
---------------	----------------

Reimplemented from [Object](#).

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

- [Background.h](#)

4.3 background Class Reference

```
#include <Background.h>
```

4.3.1 Detailed Description

Class shows game screen Public inheritance from [Object](#) class

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

- [Background.h](#)

4.4 Snake::Destroy Struct Reference

Public Member Functions

- void [operator\(\)](#) (const [Snake_Body](#) *a) const

4.4.1 Detailed Description

Struct for functor for deallocating elements

4.4.2 Member Function Documentation

4.4.2.1 operator>()

```
void Snake::Destroy::operator() (
    const Snake\_Body * a ) const [inline]
```

Overloaded function operator

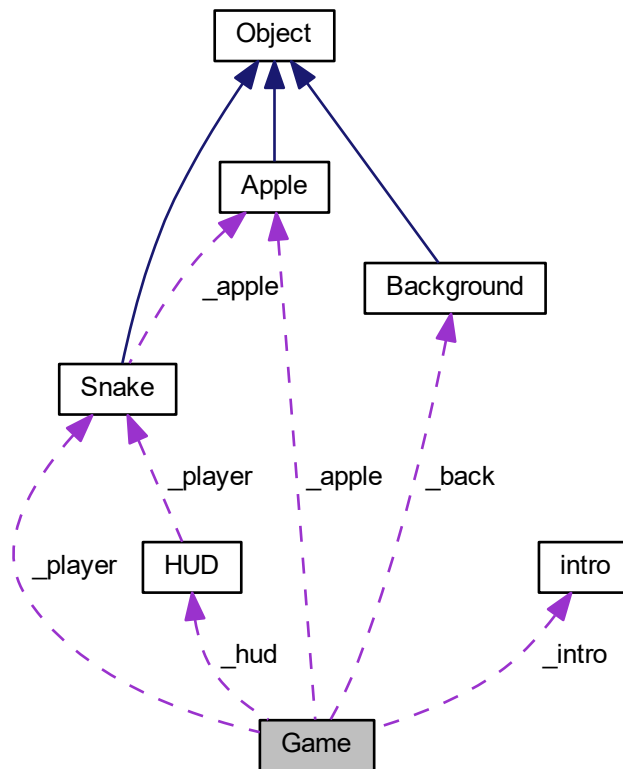
The documentation for this struct was generated from the following file:

- [Snake.h](#)

4.5 Game Class Reference

```
#include <Game.h>
```

Collaboration diagram for Game:



Static Public Member Functions

- static void [initialize](#) ()

Static Public Attributes

- const static int [SCREEN_WIDTH](#) = 640
game screen width.
- const static int [SCREEN_HEIGHT](#) = 480
game screen height.

Private Types

- enum [State](#) { [start](#), [play](#), [exit](#), [showintro](#) }

Static Private Member Functions

- static void [loop](#) ()

Static Private Attributes

- static [State](#) [_state](#)
- static sf::RenderWindow [_mainWindow](#)
rendering game window.
- static [Snake](#) * [_player](#)
- static [Apple](#) * [_apple](#)
- static [Background](#) [_back](#)
- static [HUD](#) [_hud](#)
- static [intro](#) [_intro](#)

4.5.1 Detailed Description

Most important functions for the game are handled by this class-initialization, game loop, status of the game.

4.5.2 Member Enumeration Documentation

4.5.2.1 State

```
enum Game::State [private]
```

Enumerator representing game states.

4.5.3 Member Function Documentation

4.5.3.1 initialize()

```
static void Game::initialize ( ) [static]
```

Initializing game objects.

4.5.3.2 loop()

```
static void Game::loop ( ) [static], [private]
```

[Game](#) loop which calls functions depending on current game state.

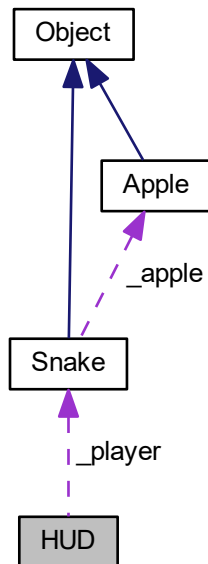
The documentation for this class was generated from the following file:

- [Game.h](#)

4.6 HUD Class Reference

```
#include <HUD.h>
```

Collaboration diagram for HUD:



Public Member Functions

- [HUD](#) ()
- [~HUD](#) ()
- void [operator=](#) ([Snake](#) *player)
- void [showScore](#) (sf::RenderWindow &>window)

Private Attributes

- sf::Font **font**
- sf::Text **text**
- sf::Text **textBest**
- [Snake](#) * [_player](#)
_player pointer to the [Snake](#) object.6
- int **best**
best score-also displayed on the screen

4.6.1 Detailed Description

Displays score.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 HUD()

```
HUD::HUD ( )
```

Constructor

4.6.2.2 ~HUD()

```
HUD::~~HUD ( )
```

Destructor

4.6.3 Member Function Documentation

4.6.3.1 operator=()

```
void HUD::operator= (
    Snake * player )
```

Overloaded operator. Sets pointer to [Snake](#)

Parameters

<i>player</i>	pointer to the Snake
---------------	--------------------------------------

4.6.3.2 showScore()

```
void HUD::showScore (
    sf::RenderWindow & window )
```

Shows score on the screen

Parameters

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

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

- [HUD.h](#)

4.7 intro Class Reference

```
#include <intro.h>
```

Public Member Functions

- [intro](#) ()
- [~intro](#) ()
- void [show](#) (sf::RenderWindow &window)

Private Attributes

- sf::Sprite [_sprite](#)
- sf::Texture [_texture](#)
- bool [_isLoading](#)

4.7.1 Detailed Description

Class showing intro and pause screen.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 intro()

```
intro::intro ( )
```

Constructor.

4.7.2.2 ~intro()

```
intro::~~intro ( )
```

Destructor.

4.7.3 Member Function Documentation

4.7.3.1 show()

```
void intro::show (
    sf::RenderWindow & window )
```

Draws sprite to the game window.

Parameters

<code>window</code>	render window.
---------------------	----------------

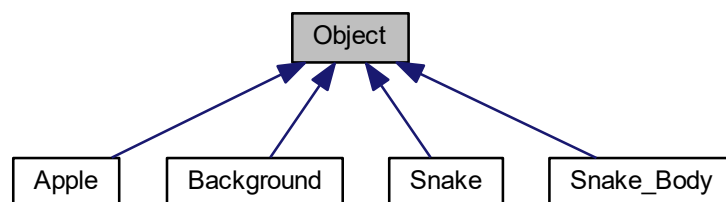
The documentation for this class was generated from the following file:

- [intro.h](#)

4.8 Object Class Reference

```
#include <Object.h>
```

Inheritance diagram for Object:



Public Member Functions

- [Object](#) ()
- virtual [~Object](#) ()
- virtual void [update](#) ()
- virtual void [draw](#) (sf::RenderWindow &window)
- virtual void [SetPosition](#) (float x, float y)
- virtual sf::Vector2f [Getposition](#) () const
- sf::Sprite & [GetSprite](#) ()

Protected Member Functions

- void [load](#) (std::string name)

Private Attributes

- bool [_isLoading](#)
- sf::Sprite [_sprite](#)
- sf::Texture [_texture](#)

4.8.1 Detailed Description

Abstract class for objects displayed on the screen. Handles loading images, updates and sets position of the object.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 Object()

```
Object::Object ( )
```

Constructor.

4.8.2.2 ~Object()

```
virtual Object::~~Object ( ) [virtual]
```

Virtual destructor.

4.8.3 Member Function Documentation

4.8.3.1 draw()

```
virtual void Object::draw (
    sf::RenderWindow & window ) [virtual]
```

Draws sprite to the game window.

Parameters

<i>window</i>	render window.
---------------	----------------

Reimplemented in [Apple](#), [Snake](#), [Background](#), and [Snake_Body](#).

4.8.3.2 load()

```
void Object::load (
    std::string name ) [protected]
```

Loading texture from the file.

Parameters

<i>string</i>	filename
---------------	----------

4.8.3.3 update()

```
virtual void Object::update ( ) [virtual]
```

Calls update function

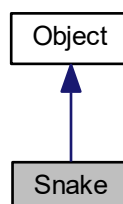
Reimplemented in [Apple](#), and [Snake](#).

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

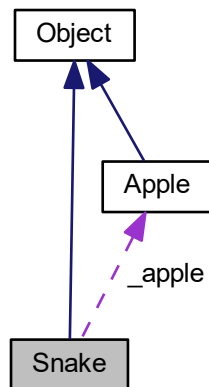
- [Object.h](#)

4.9 Snake Class Reference

Inheritance diagram for Snake:



Collaboration diagram for Snake:



Classes

- struct [Destroy](#)

Public Member Functions

- [Snake](#) ([Apple](#) *[apple](#))
- [~Snake](#) ()
- void [update](#) () override
- void [draw](#) (sf::RenderWindow &window) override
- int [GetScore](#) () const

Private Types

- enum [dir](#) { [left](#), [right](#), [up](#), [down](#) }

Private Member Functions

- void [input](#) ()
- void [bounds](#) ()
- void [apple](#) ()
- void [reset](#) ()
- void [collision](#) ()
- void [positions](#) ()

Private Attributes

- int `score`
score
- int `delay`
necessary delay on the start and on the reset states
- double `velocity`
movement speed
- `std::vector< Snake_Body * > _body`
vector contains pointers on objects from which snake is builded
- `Apple * _apple`
Apple pointer.
- `dir _dir`
object for using above enumerator

Additional Inherited Members

4.9.1 Constructor & Destructor Documentation

4.9.1.1 Snake()

```
Snake::Snake (
    Apple * apple )
```

One argument constructor.

Parameters

<code>apple</code>	pointer to the <code>Apple</code> .
--------------------	-------------------------------------

4.9.1.2 ~Snake()

```
Snake::~Snake ( )
```

Destructor where the functor is used.

4.9.2 Member Function Documentation

4.9.2.1 apple()

```
void Snake::apple ( ) [private]
```

Checks whether snake ate the apple. If yes, increases its length.

4.9.2.2 bounds()

```
void Snake::bounds ( ) [private]
```

Checks collision of the snake with the wall.

4.9.2.3 collision()

```
void Snake::collision ( ) [private]
```

Checks collision of the snake with himself.

4.9.2.4 draw()

```
void Snake::draw (
    sf::RenderWindow & window ) [override], [virtual]
```

Draws sprite to the game window.

Parameters

<i>window</i>	render window.
---------------	----------------

Reimplemented from [Object](#).

4.9.2.5 GetScore()

```
int Snake::GetScore ( ) const
```

Gets score of the game.

4.9.2.6 input()

```
void Snake::input ( ) [private]
```

Reads input and changes direction of the object.

4.9.2.7 positions()

```
void Snake::positions ( ) [private]
```

Depending on current snake direction, moves him in a proper way

4.9.2.8 reset()

```
void Snake::reset ( ) [private]
```

Resets snake to the begginging state.

4.9.2.9 update()

```
void Snake::update ( ) [override], [virtual]
```

Updates state of the game.

Reimplemented from [Object](#).

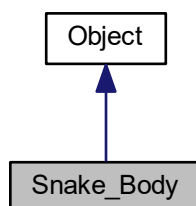
The documentation for this class was generated from the following file:

- [Snake.h](#)

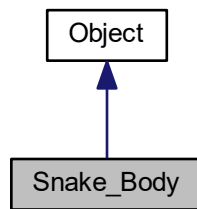
4.10 Snake_Body Class Reference

```
#include <Snake_Body.h>
```

Inheritance diagram for Snake_Body:



Collaboration diagram for Snake_Body:



Public Member Functions

- [Snake_Body](#) ([Snake_Body](#) *prev)
- [~Snake_Body](#) ()
- void [draw](#) (sf::RenderWindow &>window)
- void [operator=](#) (sf::Vector2f a)
- bool [operator==](#) ([Snake_Body](#) *check)

Additional Inherited Members

4.10.1 Detailed Description

Handles body elements of the snake. Public inheritance from [Object](#) class.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 Snake_Body()

```
Snake_Body::Snake_Body (
    Snake\_Body * prev )
```

Constructor

Parameters

<i>pointer</i>	to the Snake_Body object.
----------------	---

4.10.2.2 ~Snake_Body()

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

Destructor.

4.10.3 Member Function Documentation

4.10.3.1 draw()

```
void Snake_Body::draw (
    sf::RenderWindow & window ) [virtual]
```

Draws sprite to the game window.

Parameters

<i>window</i>	render window.
---------------	----------------

Reimplemented from [Object](#).

4.10.3.2 operator=()

```
void Snake_Body::operator= (
    sf::Vector2f a )
```

Overloaded = operator. Sets position of the snake.

Parameters

<i>vector</i>	for storing position coordinates.
---------------	-----------------------------------

4.10.3.3 operator==()

```
bool Snake_Body::operator== (
    Snake_Body * check )
```

Overloaded == operator. compares positions and checks their equality

Parameters

<i>check</i>	pointer to the Snake_Body object
--------------	--

Returns

true if yes otherwise false.

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

- [Snake_Body.h](#)

Chapter 5

File Documentation

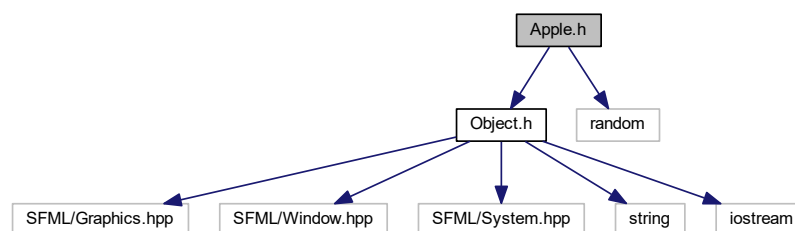
5.1 Apple.h File Reference

[Object](#) for eating class.

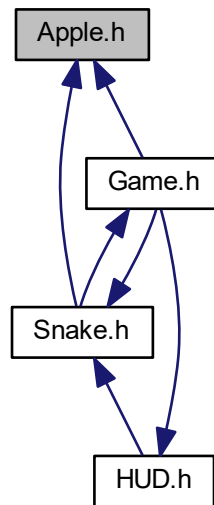
```
#include "Object.h"
```

```
#include <random>
```

Include dependency graph for Apple.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Apple](#)

5.1.1 Detailed Description

[Object](#) for eating class.

Author

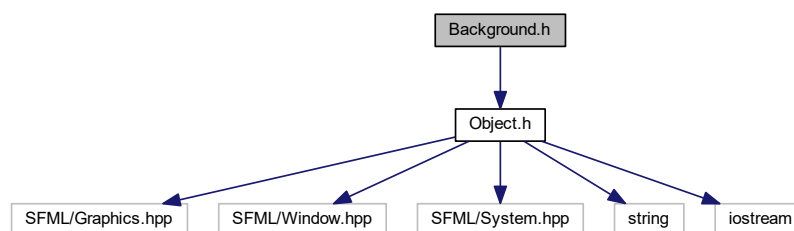
Pawel Szafraniec

5.2 Background.h File Reference

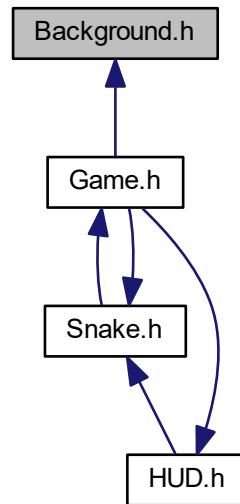
Rendered window class.

```
#include "Object.h"
```

Include dependency graph for `Background.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [Background](#)

5.2.1 Detailed Description

Rendered window class.

Author

Pawel Szafraniec

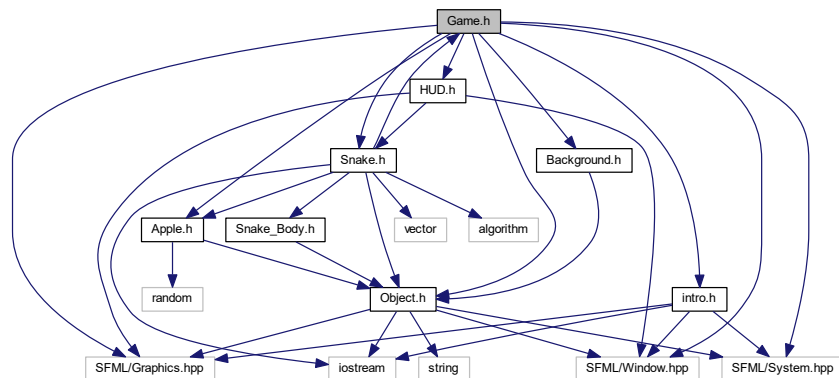
5.3 Game.h File Reference

[Game](#) class.

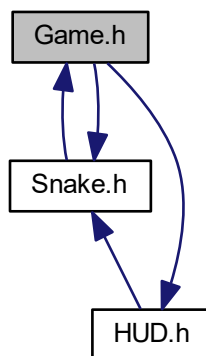
```
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/System.hpp>
#include "Object.h"
#include "Apple.h"
#include "Snake.h"
#include "Background.h"
#include "HUD.h"
```

```
#include "intro.h"
```

Include dependency graph for Game.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Game](#)

5.3.1 Detailed Description

[Game](#) class.

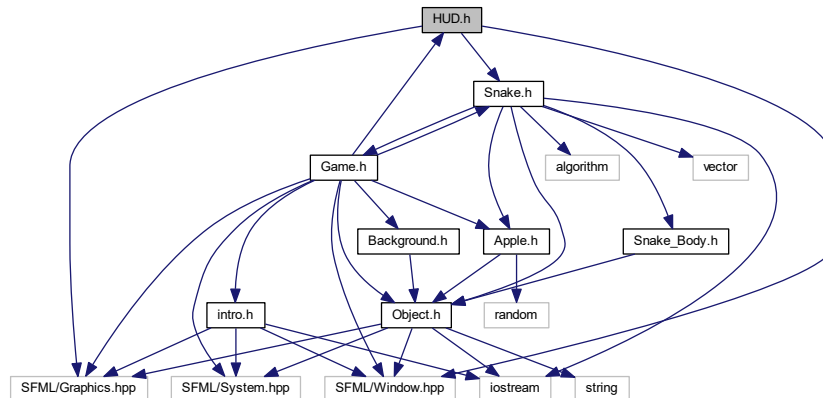
Author

Pawel Szafraniec

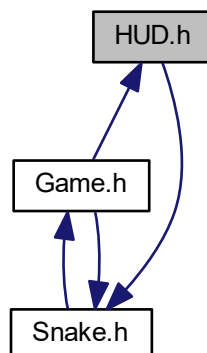
5.4 HUD.h File Reference

HUD class.

```
#include "SFML/Graphics.hpp"
#include "SFML/Window.hpp"
#include "Snake.h"
Include dependency graph for HUD.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [HUD](#)

5.4.1 Detailed Description

HUD class.

Author

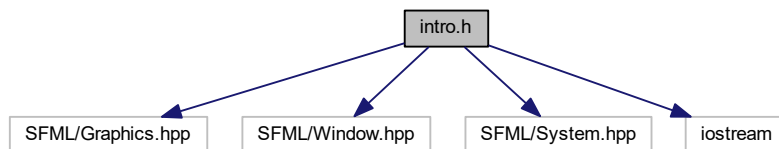
Pawel Szafraniec

5.5 intro.h File Reference

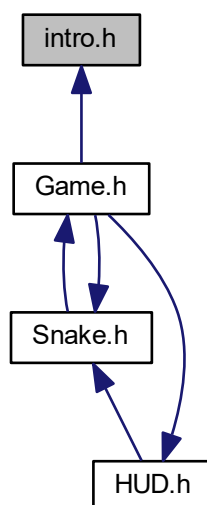
intro handling file.

```
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/System.hpp>
#include <iostream>
```

Include dependency graph for intro.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [intro](#)

5.5.1 Detailed Description

intro handling file.

Author

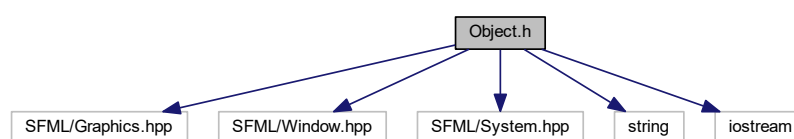
Pawel Szafraniec

5.6 Object.h File Reference

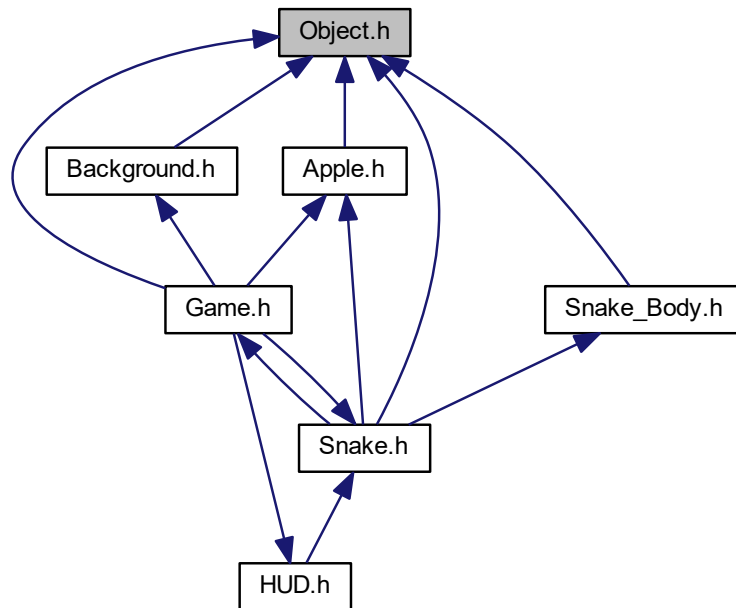
[Object](#) class.

```
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/System.hpp>
#include <string>
#include <iostream>
```

Include dependency graph for Object.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Object](#)

5.6.1 Detailed Description

[Object](#) class.

Author

Pawel Szafraniec

5.7 Snake.h File Reference

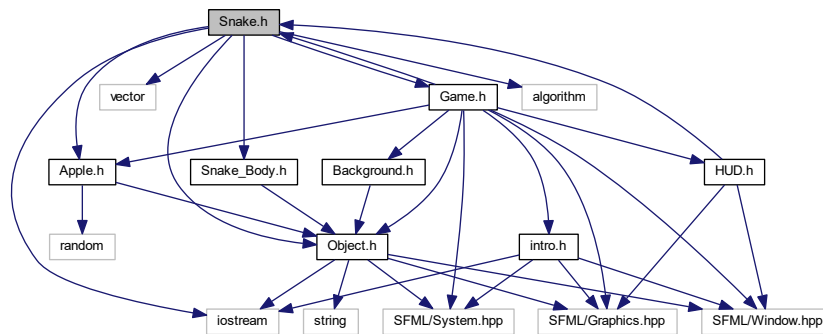
[Snake](#) class file.

```
#include "Object.h"
#include <vector>
#include "Apple.h"
#include "Snake_Body.h"
#include "Game.h"
#include <iostream>
```

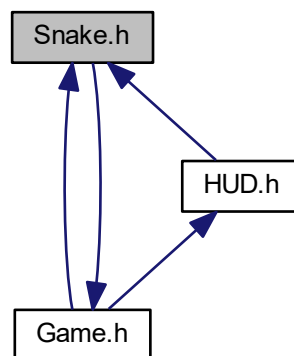


```
#include <algorithm>
```

Include dependency graph for Snake.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Snake](#)
- struct [Snake::Destroy](#)

5.7.1 Detailed Description

[Snake](#) class file.

Author

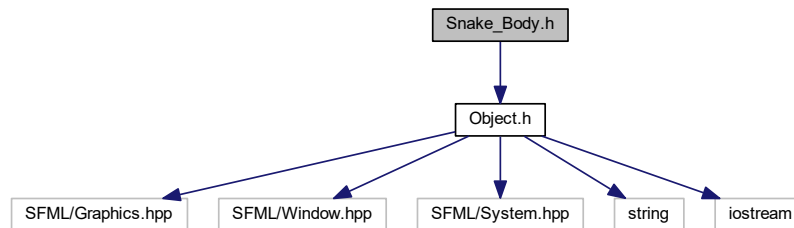
Pawel Szafranie

5.8 Snake_Body.h File Reference

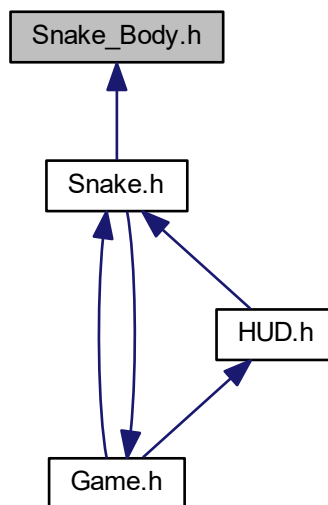
file handling body of the snake.

```
#include "Object.h"
```

Include dependency graph for Snake_Body.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Snake_Body](#)

5.8.1 Detailed Description

file handling body of the snake.

Author

Pawel Szafraniec

Index

- ~Apple
 - Apple, 8
- ~Background
 - Background, 10
- ~HUD
 - HUD, 15
- ~Object
 - Object, 18
- ~Snake
 - Snake, 21
- ~Snake_Body
 - Snake_Body, 24
- ~intro
 - intro, 16
- Apple, 7
 - ~Apple, 8
 - Apple, 8
 - draw, 8
 - position, 9
 - update, 9
- apple
 - Snake, 21
- Apple.h, 27
- Background, 9
 - ~Background, 10
 - Background, 10
 - draw, 10
- background, 11
- Background.h, 28
- bounds
 - Snake, 22
- collision
 - Snake, 22
- draw
 - Apple, 8
 - Background, 10
 - Object, 18
 - Snake, 22
 - Snake_Body, 25
- Game, 12
 - initialize, 13
 - loop, 13
 - State, 13
- Game.h, 29
- GetScore
 - Snake, 22
- HUD, 14
 - ~HUD, 15
 - HUD, 15
 - operator=, 15
 - showScore, 15
- HUD.h, 31
- initialize
 - Game, 13
- input
 - Snake, 22
- intro, 16
 - ~intro, 16
 - intro, 16
 - show, 16
- intro.h, 32
- load
 - Object, 18
- loop
 - Game, 13
- Object, 17
 - ~Object, 18
 - draw, 18
 - load, 18
 - Object, 18
 - update, 19
- Object.h, 33
- operator()
 - Snake::Destroy, 11
- operator=
 - HUD, 15
 - Snake_Body, 25
- operator==
 - Snake_Body, 25
- position
 - Apple, 9
- positions
 - Snake, 22
- reset
 - Snake, 23
- show
 - intro, 16
- showScore
 - HUD, 15
- Snake, 19
 - ~Snake, 21

- apple, [21](#)
- bounds, [22](#)
- collision, [22](#)
- draw, [22](#)
- GetScore, [22](#)
- input, [22](#)
- positions, [22](#)
- reset, [23](#)
- Snake, [21](#)
- update, [23](#)
- Snake.h, [34](#)
- Snake::Destroy, [11](#)
 - operator(), [11](#)
- Snake_Body, [23](#)
 - ~Snake_Body, [24](#)
 - draw, [25](#)
 - operator=, [25](#)
 - operator==, [25](#)
 - Snake_Body, [24](#)
- Snake_Body.h, [36](#)
- State
 - Game, [13](#)
- update
 - Apple, [9](#)
 - Object, [19](#)
 - Snake, [23](#)