Ssssnake

Generated by Doxygen 1.8.17

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

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

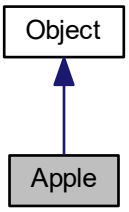
Chapter 4

Class Documentation

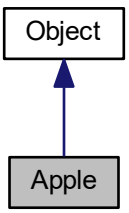
4.1 Apple Class Reference

`#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

window	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.

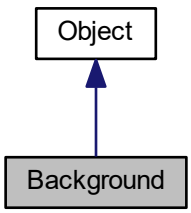
Reimplemented from Object.

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

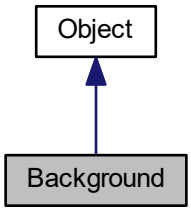
- Apple.h

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

window	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

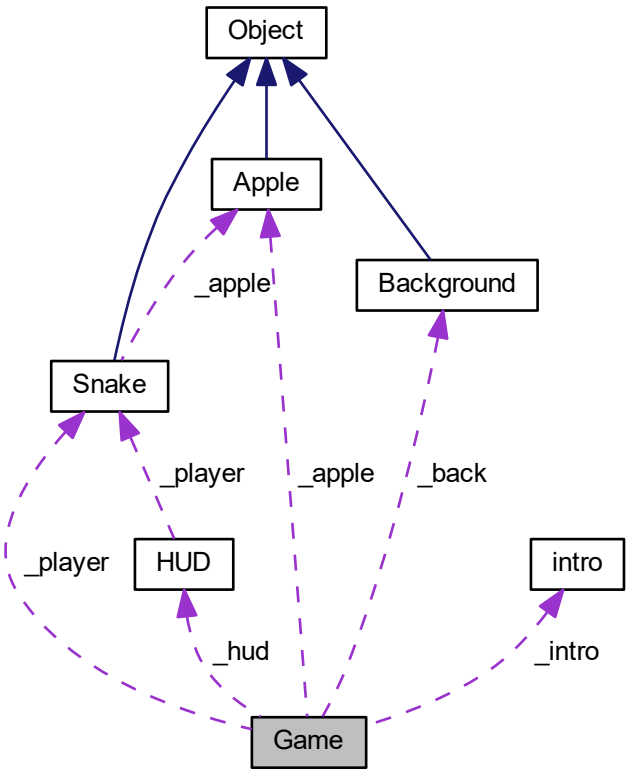
The documentation for this struct was generated from the following file:

- 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.

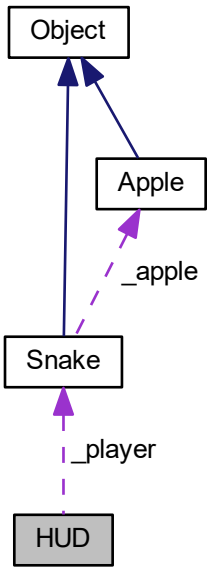
The documentation for this class was generated from the following file:

- 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

player	pointer to the Snake

4.6.3.2 showScore()

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

Shows score on the screen

Parameters

window	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 **_isLoaded**

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

window	render window.

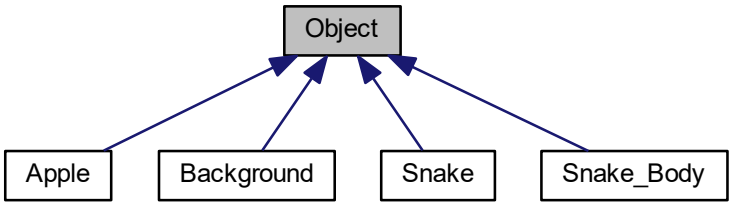
The documentation for this class was generated from the following file:

- 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 **_isLoaded**
- 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

window	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

string	filename

4.8.3.3 update()

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

Calls update function

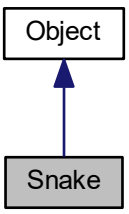
Reimplemented in Apple, and Snake.

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

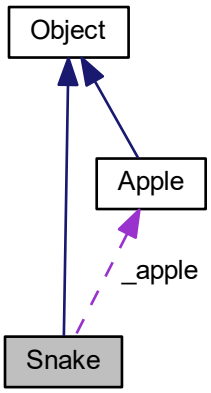
- Object.h

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

apple	pointer to the Apple.

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

window	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.

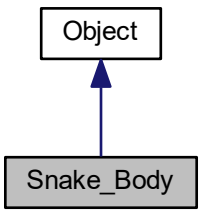
The documentation for this class was generated from the following file:

- Snake.h

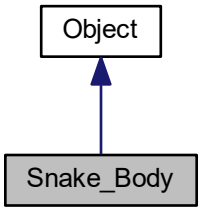
4.10 Snake_Body Class Reference

```
#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

pointer	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

window	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

vector	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

check	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

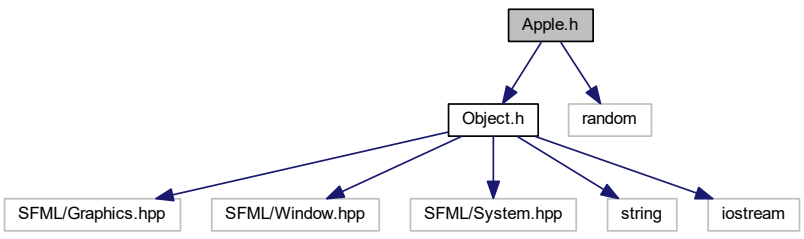
File Documentation

5.1 Apple.h File Reference

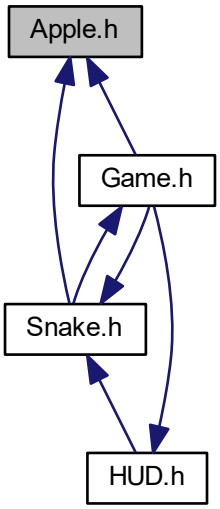
Object for eating class.

```
#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

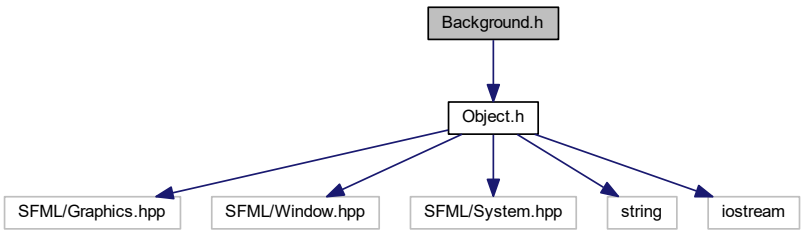
Pawel Szafraniec

5.2 Background.h File Reference

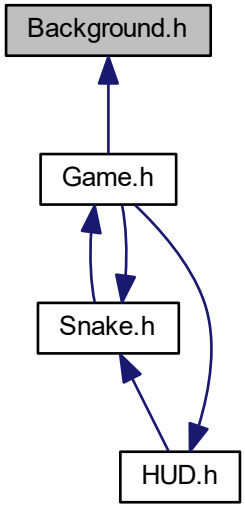
Rendered window class.

```
#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

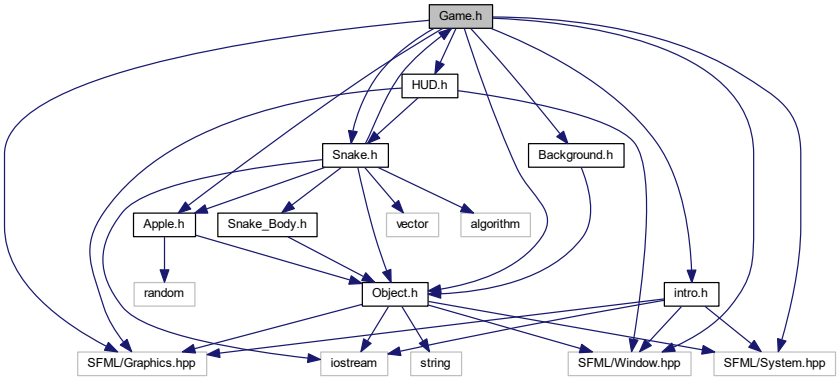
5.3 Game.h File Reference

Game class.

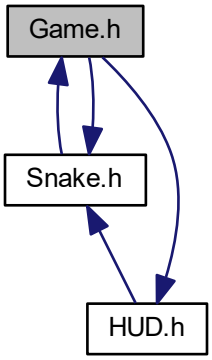
```
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/System.hpp>
#include "Object.h"
#include "Apple.h"
#include "Snake.h"
#include "Background.h"
#include "HUD.h"
```

```
#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

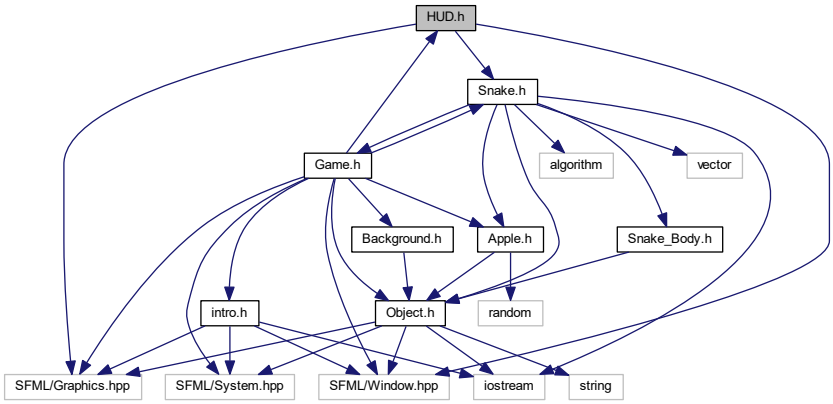
Pawel Szafraniec

5.4 HUD.h File Reference

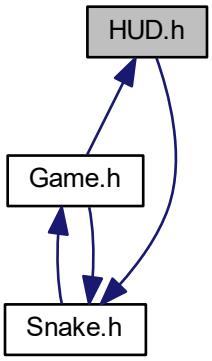
HUD class.

```
#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

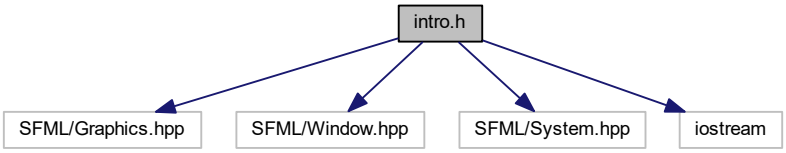
Pawel Szafraniec

5.5 intro.h File Reference

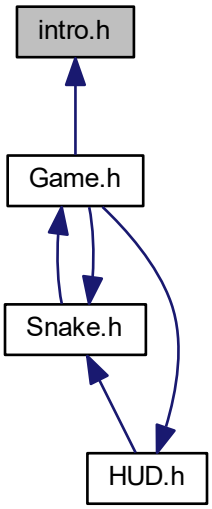
intro handling file.

```
#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

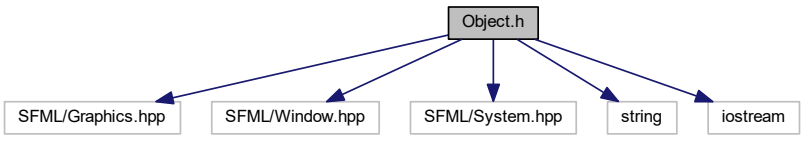
Pawel Szafraniec

5.6 Object.h File Reference

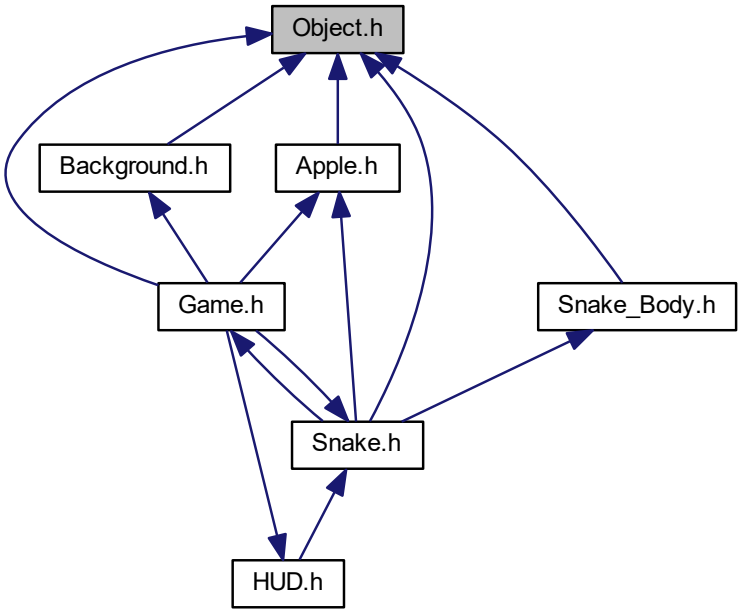
Object class.

```
#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

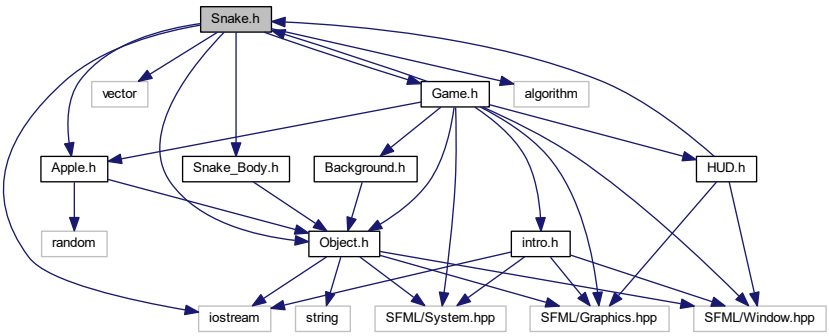
5.7 Snake.h File Reference

Snake class file.

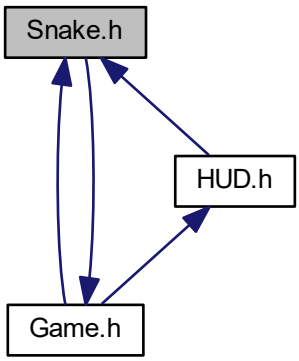
```
#include "Object.h"
#include <vector>
#include "Apple.h"
#include "Snake_Body.h"
#include "Game.h"
#include <iostream>
```

```
#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

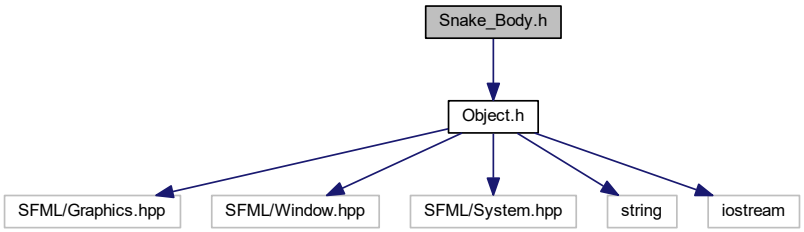
Pawel Szafranie

5.8 Snake_Body.h File Reference

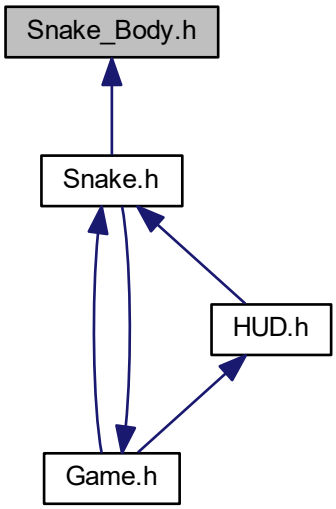
file handling body of the snake.

```
#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