My Project

Generated by Doxygen 1.8.15

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 files with brief descriptions:

Chapter 4

Class Documentation

4.1 back_screen Class Reference

Is responsible for creating and displaying the back screen of the game.

`#include <back_screen.hpp>`

Collaboration diagram for back_screen:

Public Member Functions

- void Show (sf::RenderWindow &window)

4.1.1 Detailed Description

Is responsible for creating and displaying the back screen of the game.

4.1.2 Member Function Documentation

4.1.2.1 Show()

```
void back_screen::Show (
            sf::RenderWindow & window )
```

Function load a back-screen image file and create a sprite to display it.

Parameters

window	is a created window.

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

- back_screen.hpp
- back_screen.cpp

4.2 Ball Class Reference

Responsible for creating a ball object and manipulating it.

`#include <Ball.hpp>`

Inheritance diagram for Ball:

Collaboration diagram for Ball:

Public Member Functions

- Ball ()
- virtual ~Ball ()
- void Update (float elapsedTime)
- float GetVelocity () const

Public Attributes

- bool score

Private Member Functions

- float LinearVelocityX (float angle)
- float LinearVelocityY (float angle)

Private Attributes

- float _velocity
- float _angle
- float _elapsedTimeSinceStart

Additional Inherited Members

4.2.1 Detailed Description

Responsible for creating a ball object and manipulating it.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Ball()

```
Ball::Ball ( )
```

Create the Ball object.

4.2.2.2 ~Ball()

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

Destroys objects.

4.2.3 Member Function Documentation

4.2.3.1 GetVelocity()

```
float Ball::GetVelocity ( ) const
```

Function gets the velocity of the object.

Returns

a float variable which represents a velocity of the ball.

4.2.3.2 LinearVelocityX()

```
float Ball::LinearVelocityX (
            float angle )  [private]
```

4.2.3.3 LinearVelocityY()

```
float Ball::LinearVelocityY (
            float angle )  [private]
```

4.2.3.4 Update()

```
void Ball::Update (
            float elapsedTime )  [virtual]
```

Function updates the elapsed time since the last frame and changes the speed and movement of the ball.

Parameters

elapsedTime	float variable which gives the elapsed time since the last frame.

Reimplemented from visible_obj.

4.2.4 Member Data Documentation

4.2.4.1 _angle

```
float Ball::_angle  [private]
```

4.2.4.2 _elapsedTimeSinceStart

```
float Ball::_elapsedTimeSinceStart  [private]
```

4.2.4.3 _velocity

```
float Ball::_velocity  [private]
```

4.2.4.4 score

```
bool Ball::score
```

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

- Ball.hpp
- Ball.cpp

4.3 Game Class Reference

Represents the game.

```
#include <game.hpp>
```

Collaboration diagram for Game:

Static Public Member Functions

- static void Start ()
- static sf::RenderWindow & GetWindow ()
- static const sf::Event & GetInput ()
- static const menager & GetGameObjectManager ()

Static Public Attributes

- static const int SCREEN_WIDTH = 1024
- static const int SCREEN_HEIGHT = 768

Private Types

- enum GameState {
  Uninitialized, ShowingSplash, Paused, ShowingMenu,
  Playing, Exiting }

Static Private Member Functions

- static bool IsExiting ()
- static void GameLoop ()
- static void show_back_screen ()
- static void ShowMenu ()

Private Attributes

- sf::Music music

Static Private Attributes

- static GameState _gameState = Uninitialized
- static sf::RenderWindow _mainWindow
- static menager _gameObjectManager
- static HUD _hud

4.3.1 Detailed Description

Represents the game.

4.3.2 Member Enumeration Documentation

4.3.2.1 GameState

```
enum Game::GameState  [private]
```

Represents the various states that game can be in.

Enumerator

Uninitialized	
ShowingSplash	
Paused	
ShowingMenu	
Playing	
Exiting	

4.3.3 Member Function Documentation

4.3.3.1 GameLoop()

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

Start the game loop.

4.3.3.2 GetGameObjectManager()

```
const menager & Game::GetGameObjectManager ( )  [static]
```

Returns a member variables - 'menager' object.

4.3.3.3 GetInput()

```
const sf::Event & Game::GetInput ( )  [static]
```

Wait for an event and return it.

4.3.3.4 GetWindow()

```
sf::RenderWindow & Game::GetWindow ( )  [static]
```

Returns rendered window.

4.3.3.5 IsExiting()

```
bool Game::IsExiting ( )  [static], [private]
```

Checks if the game status has been changed to 'Exiting'.

Returns

The information about whether the game state is changed.

4.3.3.6 show_back_screen()

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

The function is responsible for creating and displaying the back screen.

4.3.3.7 ShowMenu()

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

The function is responsible for creating and displaying the main menu.

4.3.3.8 Start()

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

Launch the game loop.

4.3.4 Member Data Documentation

4.3.4.1 _gameObjectManager

```
menager Game::_gameObjectManager  [static], [private]
```

4.3.4.2 _gameState

```
Game::GameState Game::_gameState = Uninitialized  [static], [private]
```

4.3.4.3 _hud

```
HUD Game::_hud  [static], [private]
```

4.3.4.4 _mainWindow

```
sf::RenderWindow Game::_mainWindow  [static], [private]
```

4.3.4.5 music

```
sf::Music Game::music  [private]
```

4.3.4.6 SCREEN_HEIGHT

```
const int Game::SCREEN_HEIGHT = 768  [static]
```

4.3.4.7 SCREEN_WIDTH

```
const int Game::SCREEN_WIDTH = 1024  [static]
```

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

- game.hpp
- game.cpp

4.4 HUD Class Reference

Responsible for creating and displaying the results of each player after losing the game by one of them.

```
#include <HUD.hpp>
```

Collaboration diagram for HUD:

Public Member Functions

- HUD ()
- ~HUD ()
- void SetPlayer (player_paletka *player)
- void SetPaletka (Paletka_2 *player2)
- void Show (sf::RenderWindow &window)

Private Attributes

- sf::Font font
- sf::Text text
- player_paletka * player
- Paletka_2 * player2

4.4.1 Detailed Description

Responsible for creating and displaying the results of each player after losing the game by one of them.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 HUD()

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

Create the HUD object.

4.4.2.2 ~HUD()

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

Destroys objects.

4.4.3 Member Function Documentation

4.4.3.1 SetPaletka()

```
void HUD::SetPaletka (
            Paletka_2 * player2 )
```

Sets a Paletka_2 pointer to an Paletka_2 object.

Parameters

player2	pointer to an object.

4.4.3.2 SetPlayer()

```
void HUD::SetPlayer (
            player_paletka * player )
```

Sets a player_paletka pointer to an player_paletka object.

Parameters

player	pointer to an object.

4.4.3.3 Show()

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

Function that draws and displays text in the background window after a lost game.

Parameters

window	is a created window.

4.4.4 Member Data Documentation

4.4.4.1 font

```
sf::Font HUD::font  [private]
```

4.4.4.2 player

```
player_paletka* HUD::player  [private]
```

4.4.4.3 player2

```
Paletka_2* HUD::player2  [private]
```

4.4.4.4 text

```
sf::Text HUD::text  [private]
```

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

- HUD.hpp
- HUD.cpp

4.5 MainMenu Class Reference

Is responsible for creating and displaying the main menu screen of the game and catches the mouse movement.

`#include <MainMenu.hpp>`

Collaboration diagram for MainMenu:

Classes

- struct MenuItem

Public Types

- enum MenuResult { Nothing, Exit, Play }

Public Member Functions

- MenuResult Show (sf::RenderWindow &window)

Private Member Functions

- MenuResult GetMenuResponse (sf::RenderWindow &window)
- MenuResult HandleClick (int x, int y)

Private Attributes

- std::list< MenuItem > _menuItems

4.5.1 Detailed Description

Is responsible for creating and displaying the main menu screen of the game and catches the mouse movement.

4.5.2 Member Enumeration Documentation

4.5.2.1 MenuResult

`enum MainMenu::MenuResult`

Represents various possible return values the menu could return.

Enumerator

Nothing	
Exit	
Play	

4.5.3 Member Function Documentation

4.5.3.1 GetMenuResponse()

```
MainMenu::MenuResult MainMenu::GetMenuResponse (
            sf::RenderWindow & window )  [private]
```

Function which correspond with the location of buttons physical locations within the main-menu image file.

Parameters

window	c

Returns

menu state.

4.5.3.2 HandleClick()

```
MainMenu::MenuResult MainMenu::HandleClick (
            int x,
            int y )  [private]
```

Function checks if any button has been pressed.If not, returns exit.

Parameters

x	an integer variable which informs of the location of the mouse.
y	an integer variable which informs of the location of the mouse.

Returns

menu state.

4.5.3.3 Show()

```
MainMenu::MenuResult MainMenu::Show (
            sf::RenderWindow & window )
```

Function load a main-menu and create a sprite to display it.

Parameters

window	is a created window.

Returns

the main-menu state from the GetMenuResponse function for a created window.

4.5.4 Member Data Documentation

4.5.4.1 _menuItems

```
std::list<MenuItem> MainMenu::_menuItems  [private]
```

A list of type MenuItem, which holds the various MenuItems that compose MainMenu.

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

- MainMenu.hpp
- MainMenu.cpp

4.6 menager Class Reference

Responspible for holding all of our visibe objects, being in charge of updating, drawing and then finally removing them.

```
#include <Menager.hpp>
```

Collaboration diagram for menager:

Classes

- struct object_delocation

Public Member Functions

- menager ()
- ∼menager ()
- void Add (std::string name, visible_obj ∗gameObject)
- void Remove (std::string name)
- int GetObjectCount () const
- visible_obj ∗ Get (std::string name) const
- void Update_all ()
- void DrawAll (sf::RenderWindow &renderWindow)

Private Attributes

- std::map< std::string, visible_obj ∗ > _gameObjects
- sf::Clock clock

4.6.1 Detailed Description

Responspible for holding all of our visibe objects, being in charge of updating, drawing and then finally removing them.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 menager()

```
menager::menager ( )
```

Create the menager object.

4.6.2.2 ∼menager()

```
menager::∼menager ( )
```

Destroys objects.

4.6.3 Member Function Documentation

4.6.3.1 Add()

```
void menager::Add (
            std::string name,
            visible_obj * gameObject )
```

Function creates map _gameObjects which is composed of a collection of std::pair<> objects and adds one more.

Parameters

name	a string which holds the name as the identifier.
gameObject	a Visible_obj pointer.

4.6.3.2 DrawAll()

```
void menager::DrawAll (
            sf::RenderWindow & renderWindow )
```

Function loops through all of the items stored within map and calls the Visible_obj's Draw method.

Parameters

renderWindow	is a created window.

4.6.3.3 Get()

```
visible_obj * menager::Get (
            std::string name ) const
```

Function iterates through the map.

Parameters

name	a string which holds the name as the identifier.

Returns

second value from a pair collected in a map.

4.6.3.4 GetObjectCount()

```
int menager::GetObjectCount ( ) const
```

Function returns the size of the map.

Returns

the number of items contained in the map.

4.6.3.5 Remove()

```
void menager::Remove (
            std::string name )
```

Function deletes the pointer to the Visible_obj* referred to in results->second.

Parameters

name	a string which holds the name as the identifier.

4.6.3.6 Update_all()

```
void menager::Update_all ( )
```

Function updates elapsed time of visible objects from a map.

4.6.4 Member Data Documentation

4.6.4.1 _gameObjects

```
std::map<std::string, visible_obj*> menager::_gameObjects  [private]
```

4.6.4.2 clock

```
sf::Clock menager::clock  [private]
```

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

- Menager.hpp
- Menager.cpp

4.7 MainMenu::MenuItem Struct Reference

```
#include <MainMenu.hpp>
```

Collaboration diagram for MainMenu::MenuItem:

Public Attributes

- sf::Rect< int > rect
- MenuResult action

4.7.1 Detailed Description

Struct that represents the individual menu items in the menu.

4.7.2 Member Data Documentation

4.7.2.1 action

```
MenuResult MainMenu::MenuItem::action
```

4.7.2.2 rect

```
sf::Rect<int> MainMenu::MenuItem::rect
```

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

- MainMenu.hpp

4.8 menager::object_delocation Struct Reference

Collaboration diagram for menager::object_delocation:

Public Member Functions

- void operator() (const std::pair< std::string, visible_obj * > &p) const

4.8.1 Detailed Description

A struct which holds a functor to delete a Visible_obj pointer.

4.8.2 Member Function Documentation

4.8.2.1 operator()()

```
void menager::object_delocation::operator() (
            const std::pair< std::string, visible_obj * > & p ) const  [inline]
```

Overloaded function operator ().

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

- Menager.hpp

4.9 Paletka_2 Class Reference

Creates the AI player's paddle.

`#include <paletka_2.hpp>`

Inheritance diagram for Paletka_2:

Collaboration diagram for Paletka_2:

Public Member Functions

- Paletka_2 (void)
- ~Paletka_2 (void)
- void Update (float elapsedTime)
- void Draw (sf::RenderWindow &rw)
- float GetVelocity () const

Public Attributes

- int score

Private Attributes

- float _velocity
- float _maxVelocity
- float _elapsedTimeSinceStart

Additional Inherited Members

4.9.1 Detailed Description

Creates the AI player's paddle.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 Paletka_2()

```
Paletka_2::Paletka_2 (
            void  )
```

Create the Paletka_2 object.

4.9.2.2 ~Paletka_2()

```
Paletka_2::~Paletka_2 (
            void  )
```

Destroys objects.

4.9.3 Member Function Documentation

4.9.3.1 Draw()

```
void Paletka_2::Draw (
            sf::RenderWindow & rw )
```

Function draw a rendered window.

Parameters

rw	is a created window.

4.9.3.2 GetVelocity()

```
float Paletka_2::GetVelocity ( ) const
```

Function gets the velocity of the object.

Returns

a float variable which represents a velocity of the paddle.

4.9.3.3 Update()

```
void Paletka_2::Update (
            float elapsedTime )  [virtual]
```

Override Update() function from the base class visible_obj. Checks if the arrows are pressed, changes the speed of the paddle.

Parameters

elapsedTime	elapsed since the last frame.

Reimplemented from visible_obj.

4.9.4 Member Data Documentation

4.9.4.1 _elapsedTimeSinceStart

```
float Paletka_2::_elapsedTimeSinceStart  [private]
```

4.9.4.2 _maxVelocity

```
float Paletka_2::_maxVelocity  [private]
```

4.9.4.3 _velocity

```
float Paletka_2::_velocity  [private]
```

4.9.4.4 score

```
int Paletka_2::score
```

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

- paletka_2.hpp
- paletka_2.cpp

4.10 player_paletka Class Reference

Creates the player's paddle.

`#include <player_paletka.hpp>`

Inheritance diagram for player_paletka:

Collaboration diagram for player_paletka:

Public Member Functions

- player_paletka ()
- ~player_paletka ()
- void Update (float elapsedTime)
- void Draw (sf::RenderWindow &rw)
- float GetVelocity () const

Public Attributes

- int score

Private Attributes

- float _velocity
- float _maxVelocity

Additional Inherited Members

4.10.1 Detailed Description

Creates the player's paddle.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 player_paletka()

`player_paletka::player_paletka ( )`

Create the player_paddle object.

4.10.2.2 ~player_paletka()

```
player_paletka::~player_paletka ( )
```

Destroys objects.

4.10.3 Member Function Documentation

4.10.3.1 Draw()

```
void player_paletka::Draw (
            sf::RenderWindow & rw )
```

Function draw a rendered window.

Parameters

rw	is a created window.

4.10.3.2 GetVelocity()

```
float player_paletka::GetVelocity ( ) const
```

Function gets the velocity of the object.

Returns

a float variable which represents a velocity of the paddle.

4.10.3.3 Update()

```
void player_paletka::Update (
            float elapsedTime )  [virtual]
```

Override Update() function from the base class visible_obj. Checks if the arrows are pressed, changes the speed of the paddle.

Parameters

elapsedTime	elapsed time since the last frame.

Reimplemented from visible_obj.

4.10.4 Member Data Documentation

4.10.4.1 _maxVelocity

```
float player_paletka::_maxVelocity  [private]
```

4.10.4.2 _velocity

```
float player_paletka::_velocity  [private]
```

4.10.4.3 score

```
int player_paletka::score
```

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

- player_paletka.hpp
- player_paletka.cpp

4.11 visible_obj Class Reference

Represents an item in the world that needs to be drawn, such as the player's paddle or the game's ball.

```
#include <Visible_obj.hpp>
```

Inheritance diagram for visible_obj:

Collaboration diagram for visible_obj:

Public Member Functions

- visible_obj ()
- virtual ~visible_obj ()
- virtual void load (std::string filename)
- virtual void drawing (sf::RenderWindow &window)
- virtual void Update (float elapsedTime)
- virtual void set_position (float x, float y)
- virtual sf::Vector2f GetPosition () const
- virtual bool IsLoaded () const
- virtual float GetWidth () const
- virtual float GetHeight () const
- virtual sf::Rect< float > GetBoundingRect () const

Protected Member Functions

- sf::Sprite & GetSprite ()

Private Attributes

- sf::Sprite sprite_
- sf::Texture image_
- std::string file_name
- bool is_loaded

4.11.1 Detailed Description

Represents an item in the world that needs to be drawn, such as the player's paddle or the game's ball.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 visible_obj()

```
visible_obj::visible_obj ( )
```

Create the visible_obj object.

4.11.2.2 ~visible_obj()

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

Destroys objects.

4.11.3 Member Function Documentation

4.11.3.1 drawing()

```
void visible_obj::drawing (
            sf::RenderWindow & window )  [virtual]
```

Function draw a sprite if possible.

Parameters

window	is a created window.

4.11.3.2 GetBoundingRect()

```
sf::Rect< float > visible_obj::GetBoundingRect ( ) const  [virtual]
```

Gets the global bounding rectangle of the sprite.

Returns

the bounds of the sprite in the coordinate system.

4.11.3.3 GetHeight()

```
float visible_obj::GetHeight ( ) const  [virtual]
```

Gets the height of the sprite.

Returns

the bounds of the sprite in the coordinate system.

4.11.3.4 GetPosition()

```
sf::Vector2f visible_obj::GetPosition ( ) const  [virtual]
```

Gets position of the sprite.

Returns

a mathematical vector with two coordinates (x and y) or current rotation of the object, in degrees.

4.11.3.5 GetSprite()

```
sf::Sprite & visible_obj::GetSprite ( )  [protected]
```

Gets the sprite.

Returns

the sprite.

4.11.3.6 GetWidth()

```
float visible_obj::GetWidth ( ) const  [virtual]
```

Gets the width of the sprite.

Returns

the bounds of the sprite in the coordinate system.

4.11.3.7 IsLoaded()

```
bool visible_obj::IsLoaded ( ) const  [virtual]
```

Informs if the file was loaded correctly.

Returns

a bool variable, which is true or false, depending on whether the file was loaded without any problem.

4.11.3.8 load()

```
void visible_obj::load (
            std::string filename )  [virtual]
```

Function checks if loading a file is possible.

Parameters

filename	the name of the file that is loaded into the image.

4.11.3.9 set_position()

```
void visible_obj::set_position (
            float x,
            float y )  [virtual]
```

Sets position of the sprite.

Parameters

x	is a float variable which informs of the location of the sprite.
y	is a float variable which informs of the location of the sprite.

4.11.3.10 Update()

```
void visible_obj::Update (
            float elapsedTime )  [virtual]
```

Call update for all game objects

Parameters

elapsedTime	elapsed time since the last frame

Reimplemented in Paletka_2, player_paletka, and Ball.

4.11.4 Member Data Documentation

4.11.4.1 file_name

```
std::string visible_obj::file_name  [private]
```

4.11.4.2 image_

```
sf::Texture visible_obj::image_  [private]
```

4.11.4.3 is_loaded

```
bool visible_obj::is_loaded  [private]
```

4.11.4.4 sprite_

```
sf::Sprite visible_obj::sprite_  [private]
```

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

- Visible_obj.hpp
- Visible_obj.cpp

Chapter 5

File Documentation

5.1 back_screen.cpp File Reference

```
#include <SFML/System.hpp>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
#include <map>
#include <iostream>
#include <cassert>
#include "back_screen.hpp"
```

Include dependency graph for back_screen.cpp:

5.2 back_screen.hpp File Reference

The file contains class which displays the back screen of the game.

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

Include dependency graph for back_screen.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class back_screen

  Is responsible for creating and displaying the back screen of the game.

5.2.1 Detailed Description

The file contains class which displays the back screen of the game.

Author

Oliwia Mlonek

5.3 Ball.cpp File Reference

```
#include "Ball.hpp"
#include "game.hpp"
#include "Menager.hpp"
#include "Visible_obj.hpp"
#include <cmath>
#include <SFML/Audio.hpp>
```
Include dependency graph for Ball.cpp:

5.4 Ball.hpp File Reference

The file contains class which manipulate the game ball object.

```
#include <iostream>
#include "Visible_obj.hpp"
#include "Menager.hpp"
#include "game.hpp"
```
Include dependency graph for Ball.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Ball

  Responsible for creating a ball object and manipulating it.

5.4.1 Detailed Description

The file contains class which manipulate the game ball object.

Author

Oliwia Mlonek

5.5 game.cpp File Reference

```
#include "MainMenu.hpp"
#include "back_screen.hpp"
#include "player_paletka.hpp"
#include "Visible_obj.hpp"
#include "Ball.hpp"
#include "paletka_2.hpp"
#include <SFML/System.hpp>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
#include <map>
#include <iostream>
#include <cassert>
```
Include dependency graph for game.cpp:

5.6 game.hpp File Reference

The file contains entirely static class 'Game'.

```
#include <SFML/Window.hpp>
#include <SFML/Graphics.hpp>
#include <SFML/Audio.hpp>
#include "player_paletka.hpp"
#include "paletka_2.hpp"
#include "Visible_obj.hpp"
#include "Menager.hpp"
#include "HUD.hpp"
```

Include dependency graph for game.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Game

  Represents the game.

5.6.1 Detailed Description

The file contains entirely static class 'Game'.

Author

Oliwia Mlonek

5.7 HUD.cpp File Reference

```
#include "HUD.hpp"
#include "game.hpp"
#include "player_paletka.hpp"
#include "paletka_2.hpp"
#include <SFML/System.hpp>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
#include <iostream>
```

Include dependency graph for HUD.cpp:

5.8 HUD.hpp File Reference

```
#include <iostream>
#include "player_paletka.hpp"
#include "paletka_2.hpp"
#include <SFML/System.hpp>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
```

Include dependency graph for HUD.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class HUD

  Responsible for creating and displaying the results of each player after losing the game by one of them.

5.9 main.cpp File Reference

```
#include "game.hpp"
#include "back_screen.hpp"
#include <SFML/System.hpp>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
#include <map>
#include <iostream>
#include <cassert>
```

Include dependency graph for main.cpp:

Functions

- int main (int argc, const char ∗argv[ ])

5.9.1 Function Documentation

5.9.1.1 main()

```
int main (
            int argc,
            const char * argv[] )
```

5.10 MainMenu.cpp File Reference

```
#include "MainMenu.hpp"
#include <SFML/System.hpp>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
#include <map>
#include <iostream>
#include <cassert>
```

Include dependency graph for MainMenu.cpp:

5.11 MainMenu.hpp File Reference

The file contains class which displays the main menu screen of the game.

```
#include <stdio.h>
#include <SFML/Window.hpp>
#include <SFML/Graphics.hpp>
#include <list>
```

Include dependency graph for MainMenu.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class MainMenu

  Is responsible for creating and displaying the main menu screen of the game and catches the mouse movement.
- struct MainMenu::MenuItem

5.11.1 Detailed Description

The file contains class which displays the main menu screen of the game.

Author

Oliwia Mlonek

5.12 Menager.cpp File Reference

```
#include "Menager.hpp"
#include "Visible_obj.hpp"
#include "game.hpp"
#include <iostream>
#include <map>
#include <cassert>
#include <SFML/System.hpp>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
```

Include dependency graph for Menager.cpp:

5.13 Menager.hpp File Reference

The file contains class which manipulate the visible obejcts.

```
#include "Visible_obj.hpp"
#include <iostream>
#include <map>
#include <cassert>
#include <SFML/System.hpp>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
```

Include dependency graph for Menager.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class menager

  Responspible for holding all of our visibe objects, being in charge of updating, drawing and then finally removing them.
- struct menager::object_delocation

5.13.1 Detailed Description

The file contains class which manipulate the visible obejcts.

Author

Oliwia Mlonek

5.14 paletka_2.cpp File Reference

```
#include "paletka_2.hpp"
#include "game.hpp"
#include "Ball.hpp"
#include "Visible_obj.hpp"
#include <SFML/System.hpp>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
```

Include dependency graph for paletka_2.cpp:

5.15 paletka_2.hpp File Reference

The file contains class which manipulate the AI player paddle object.

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

Include dependency graph for paletka_2.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Paletka_2

  Creates the AI player's paddle.

5.15.1 Detailed Description

The file contains class which manipulate the AI player paddle object.

Author

Oliwia Mlonek

5.16 player_paletka.cpp File Reference

```
#include "player_paletka.hpp"
#include "Visible_obj.hpp"
#include "game.hpp"
#include "MainMenu.hpp"
#include "back_screen.hpp"
#include <iostream>
#include <SFML/System.hpp>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
```

Include dependency graph for player_paletka.cpp:

5.17 player_paletka.hpp File Reference

The file contains class which manipulate the player paddle object.

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

Include dependency graph for player_paletka.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class player_paletka

  Creates the player's paddle.

5.17.1 Detailed Description

The file contains class which manipulate the player paddle object.

Author

Oliwia Mlonek

5.18 Visible_obj.cpp File Reference

```
#include "Visible_obj.hpp"
#include "game.hpp"
#include "MainMenu.hpp"
#include "back_screen.hpp"
#include <iostream>
#include <SFML/System.hpp>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
```

Include dependency graph for Visible_obj.cpp:

5.19 Visible_obj.hpp File Reference

The file contains class which displays the objects such as ball or paddles.

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

Include dependency graph for Visible_obj.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class visible_obj

  Represents an item in the world that needs to be drawn, such as the player's paddle or the game's ball.

5.19.1 Detailed Description

The file contains class which displays the objects such as ball or paddles.

Author

Oliwia Mlonek