PICTURETETRIS

1.0.0

Generated by Doxygen 1.9.1

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

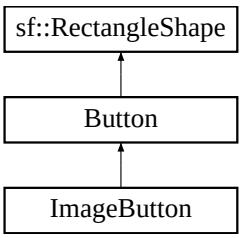
Class Documentation

3.1 Button Class Reference

Reprezentacja przyciskow w MENU.

`#include <Button.h>`

Inheritance diagram for Button:



Public Member Functions

- **Button** (float width, float height, const sf::String &text="", unsigned int size=TEXT_SIZE)

  Konstruuje przycisk o zadanych parametrach rozmiaru i tekstu.
- virtual void **selectThisButton** ()

  Funkcja ustawiajaca wybrany przycisk jako zaznaczony.
- virtual void **unselectThisButton** ()

  Funkcja ustawiajaca nie wybrane przyciski jako niezaznaczone.
- void **setStringForText** (const sf::String &string)

  Funkcja ustawiajaca tekst pokazywany na przycisku.
- void **setPositionOfTheButton** (float x, float y)

  Funkcja zmieniajaca poozenie przycisku w oknie.

Protected Member Functions

- bool **isTextPositionSet** ()
- void **draw** (sf::RenderTarget &target, sf::RenderStates states) const override

Protected Attributes

- sf::Text **_text**
- bool **_selected** = false

Static Protected Attributes

- static const sf::Color **DEFAULT_COLOR** { 73, 215, 245 }
- static const sf::Color **SELECTED_COLOR** {40, 146, 168}
- static const sf::Color **SELECTED_TXT_COLOR** {0, 0, 0}
- static const sf::Color **DEFAULT_TXT_COLOR** { 255, 255, 255 }
- static const unsigned int **TEXT_SIZE** = 28

3.1.1 Detailed Description

Reprezentacja przyciskow w MENU.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Button()

```
Button::Button (
            float width,
            float height,
            const sf::String & text = "",
            unsigned int size = TEXT_SIZE )
```

Konstruuje przycisk o zadanych parametrach rozmiaru i tekstu.

Parameters

width	Szerokos w pikselach
height	Wysokos w pikselach
text	Wyswietlany tekst, domyslnie pusty
size	Dugos tekstu

3.1.3 Member Function Documentation

3.1.3.1 selectThisButton()

```
void Button::selectThisButton ( )  [virtual]
```

Funkcja ustawiajaca wybrany przycisk jako zaznaczony.

Zmiana koloru ta i tekstu

Reimplemented in **ImageButton** (p. 23).

3.1.3.2 setPositionOfTheButton()

```
void Button::setPositionOfTheButton (
            float x,
            float y )
```

Funkcja zmieniajaca poozenie przycisku w oknie.

Parameters

x	Wsporzedna osi OX w px
y	Wsporzedna osi OY w px

3.1.3.3 setStringForText()

```
void Button::setStringForText (
            const sf::String & string )
```

Funkcja ustawiajaca tekst pokazywany na przycisku.

Automatyczne dopasowanie fontu do buttona, zapobieganie wychodzeniu tekstu poza obszar przycisku

Parameters

string	Tekst, ktory nalezy ustawi

3.1.3.4 unselectThisButton()

```
void Button::unselectThisButton ( )  [virtual]
```

Funkcja ustawiajaca nie wybrane przyciski jako niezaznaczone.

Default'owy kolor ta i tekstu

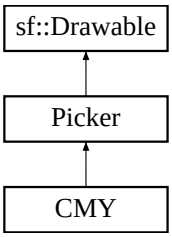
Reimplemented in **ImageButton** (p. 23).

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

- SFML/Button.h
- SFML/Button.cpp

3.2 CMY Class Reference

Inheritance diagram for CMY:



Public Member Functions

- void **draw_to_color_pixels** ()
- **CMY** (const wchar_t ∗Name, sf::Vector2f Position, const wchar_t ∗Prefix)
- void **update_color** (double Param)
- virtual void **draw** (sf::RenderTarget &target, sf::RenderStates states) const

Additional Inherited Members

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

- SFML/SFML.cpp

3.3 Config Class Reference

Reprezentacja ustawien gry.

`#include <Config.h>`

Public Member Functions

- void **setNumberOfBlocksPerPictureSide** (unsigned int amount)

  Ustawia liczbe klockow na bok obrazka.
- unsigned int **getsetNumberOfBlocksPerPictureSide** () const

  Zwraca liczbe klockow na bok.
- double **getSpeedOfFallingBlocks** () const

  Zwraca szybkos spadania.

Public Attributes

- int **level** = 0

  Aktualny poziom trudnosci gry.

3.3.1 Detailed Description

Reprezentacja ustawien gry.

3.3.2 Member Function Documentation

3.3.2.1 getsetNumberOfBlocksPerPictureSide()

```
unsigned int Config::getsetNumberOfBlocksPerPictureSide ( ) const
```

Zwraca liczbe klockow na bok.

Returns

Liczba klockow na bok

3.3.2.2 getSpeedOfFallingBlocks()

```
double Config::getSpeedOfFallingBlocks ( ) const
```

Zwraca szybkos spadania.

Returns

Szybkos spadanie klockow

3.3.2.3 setNumberOfBlocksPerPictureSide()

```
void Config::setNumberOfBlocksPerPictureSide (
            unsigned int amount )
```

Ustawia liczbe klockow na bok obrazka.

Parameters

amount	Liczba klockow na bok

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

- SFML/Config.h
- SFML/Config.cpp

3.4 Game Class Reference

Centralna klasa w ktorej toczy sie cae dziaanie porgramu.

`#include <Game.h>`

Public Member Functions

- sf::RenderWindow & **getRenderedWindow** ()
  Zwraca okno stworzone za pomoca SFML.
- sf::Font & **getTextFont** ()
  Zwraca czcionke wykorzystywana w grze.
- const sf::Color & **getColorOfBackground** () const
  Zwraca kolor ta.
- const sf::Color & **getColorOfText** () const
  Zwraca kolor napisow.
- const sf::Color & **getHighlightColor** () const
  Zwraca kolor podswitlonej opcji.
- void **loadImageFromFile** (unsigned int idx)
  Funkcja wczytujaca z pliku obrazek.
- void **transformImagesToTextures** ()
  Przetwarza obrazki na tekstury.
- sf::Sprite **transformTextureToSpriteWithResizing** (sf::Texture &texture, float width, float height)
  Funkcja zamieniajaca texture na sprite o zadanych wymiarach.
- unsigned int **getWidthOfGameWindow** () const
  Funkcja staa zwracajaca szerokos okna gry.
- unsigned int **getHeightOfGameWindow** () const
  Funkcja staa zwracajaca wysokos okna gry.
- sf::Texture & **getTextureOfGivenImageNumber** (int n= **selected**)
  Zwraca texture obrazka wczytanego do pamieci o wybranym indeksie.
- void **runProgram** ()
  Funkcja inicjujca cae dziaanie programu.
- void **setVolumeOfSound** (int val)
  Funkcja sterujaca gosnoscia muzyki w grze.
- void **playWinSoundWithVolumeOf** (int val)
  Uruchomienie dwieku przy zwyciestwie gry.

Static Public Member Functions

- static **Game** & **getInstanceOfGame** ()
  *Zwraca instancje obiektu **Game** (p. 10).*
- static **Config** & **getConfig** ()
  Zwraca obecne ustawienia gry.
- static **GameController** & **getGameController** ()
  Zwraca menadzer stanow GameStateManager.

Public Attributes

- float **gameTime** = 0
  Zmienna przechowujaca czas pojedynczej gry.

Static Public Attributes

- static const unsigned int **IMAGES_COUNT** = 4

  Zmienna statyczna przechowujaca liczbe obrazkow.
- static const std::string **TITLE** = "Picture Tetris"

  Statyczny string przechowujacy tytu gry.
- static int **selected** = 0

  Zmienna statyczna przechowujaca numer wybranego obrazka.

3.4.1 Detailed Description

Centralna klasa w ktorej toczy sie cae dziaanie porgramu.

Singleton - jednoczesnie moze wystepowa tylko jedna instancja gry.

3.4.2 Member Function Documentation

3.4.2.1 getColorOfBackground()

```
const sf::Color & Game::getColorOfBackground ( ) const
```

Zwraca kolor ta.

Returns

Referencja do kolor ta (SFML class)

3.4.2.2 getColorOfText()

```
const sf::Color & Game::getColorOfText ( ) const
```

Zwraca kolor napisow.

Returns

Referencja do koloru napisow (SFML class)

3.4.2.3 getConfig()

```
Config & Game::getConfig ( )  [static]
```

Zwraca obecne ustawienia gry.

Returns

Referencja do **Config** (p. 8)

3.4.2.4 getGameController()

```
GameController & Game::getGameController ( )  [static]
```

Zwraca menadzer stanow GameStateManager.

Returns

Referencja do **GameController** (p. 16)

3.4.2.5 getHeightOfGameWindow()

```
unsigned int Game::getHeightOfGameWindow ( ) const
```

Funkcja staa zwracajaca wysokos okna gry.

Returns

Wysokos okna wyrazona w dodatniej liczbie pikseli

3.4.2.6 getHighlightColor()

```
const sf::Color & Game::getHighlightColor ( ) const
```

Zwraca kolor podswitlonej opcji.

Returns

Referencja do koloru podswietlenia (SFML class)

3.4.2.7 getInstanceOfGame()

```
Game & Game::getInstanceOfGame ( )  [static]
```

Zwraca instancje obiektu **Game** (p. 10).

Returns

Referencja do **Game** (p. 10)

3.4.2.8 getRenderedWindow()

```
sf::RenderWindow & Game::getRenderedWindow ( )
```

Zwraca okno stworzone za pomoca SFML.

Returns

Referencja do klasy RenderWindow (SFML class)

3.4.2.9 getTextFont()

```
sf::Font & Game::getTextFont ( )
```

Zwraca czcionke wykorzystywana w grze.

Returns

Referencja do Font (SFML class)

3.4.2.10 getTextureOfGivenImageNumber()

```
sf::Texture & Game::getTextureOfGivenImageNumber (
            int n =  selected )
```

Zwraca texture obrazka wczytanego do pamieci o wybranym indeksie.

Domyslny numer indeksu to ten ktory, ktory obrazek zostal wybrany

Parameters

int	Indeks tablicy i koncowka nazwy pliku z obrazkiem

Returns

Referencja do Tekstury (SFML class)

3.4.2.11 getWidthOfGameWindow()

```
unsigned int Game::getWidthOfGameWindow ( ) const
```

Funkcja staa zwracajaca szerokos okna gry.

Returns

Szerokos okna wyrazona w dodatniej liczbie pikseli

3.4.2.12 loadImageFromFile()

```
void Game::loadImageFromFile (
            unsigned int idx )
```

Funkcja wczytujaca z pliku obrazek.

Obrazki sa przechowywane w katalogu data. Ich nazwa to numer indeksu, ktory beda zajmowa w tablicy

Parameters

Dodatnia	wartos reprezentujaca indeks obrazka

3.4.2.13 playWinSoundWithVolumeOf()

```
void Game::playWinSoundWithVolumeOf (
            int val )
```

Uruchomienie dwieku przy zwyciestwie gry.

Parameters

Wartos	dzwieku we wsporzednych cakowitych z przedziau 0-100

3.4.2.14 setVolumeOfSound()

```
void Game::setVolumeOfSound (
            int val )
```

Funkcja sterujaca gosnoscia muzyki w grze.

Parameters

Wartos	dzwieku we wsporzednych cakowitych z przedziau 0-100

3.4.2.15 transformImagesToTextures()

```
void Game::transformImagesToTextures ( )
```

Przetwarza obrazki na tekstury.

Parameters

Dodatnia	wartos reprezentujaca indeks tablicy

3.4.2.16 transformTextureToSpriteWithResizing()

```
sf::Sprite Game::transformTextureToSpriteWithResizing (
            sf::Texture & texture,
            float width,
            float height )
```

Funkcja zamieniajaca texture na sprite o zadanych wymiarach.

Parameters

texture	Tekstura
width	Wyjsciowa szerokos tekstury wyrazona w pikselach
height	Wyjsciowa wysokos tekstury wyrazona w pikselach

Returns

Obiekt Sprite z Textura wejsciowa

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

- SFML/Game.h
- SFML/Game.cpp

3.5 GameController Class Reference

Reprezentuje kontroler stanow aplikacji.

`#include <GameController.h>`

Public Member Functions

- void **eventListener** (sf::Event &event)
  Reaguje na event od gracza.
- void **updateCurrentGameLevel** (float time)
  Aktualizuje stan.
- void **renderCurrentGameLevel** ()
  Renderuje aktualny stan.
- void **setNewGameLevel** (std::unique_ptr< **Level** > state)
  Zamyka poprzednie stany aplikacji i ustawia nowy.

3.5.1 Detailed Description

Reprezentuje kontroler stanow aplikacji.

3.5.2 Member Function Documentation

3.5.2.1 eventListener()

```
void GameController::eventListener (
            sf::Event & event )
```

Reaguje na event od gracza.

Parameters

event	Zdarzenie od gracza

3.5.2.2 setNewGameLevel()

```
void GameController::setNewGameLevel (
            std::unique_ptr< Level > state )
```

Zamyka poprzednie stany aplikacji i ustawia nowy.

Parameters

state	Nowy stan do ustawienia.

3.5.2.3 updateCurrentGameLevel()

```
void GameController::updateCurrentGameLevel (
            float time )
```

Aktualizuje stan.

Parameters

time	Czas renderowania i przetwarzania poprzedniej klatki

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

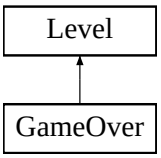
- SFML/GameController.h
- SFML/GameController.cpp

3.6 GameOver Class Reference

Reprezentacja konca gry.

```
#include <GameOver.h>
```

Inheritance diagram for GameOver:



Public Member Functions

- **GameOver** ()

  *Konstruktor **GameOver** (p. 17).*
- void **eventListener** (sf::Event &event) override

  Reaguje na event od gracza.
- void **updateCurrentStateOfCurrentGameLevel** (float dtime) override

  Aktualizuje stan.
- void **renderCurrentStateOfCurrentGameLevel** () override

  Renderuje aktualny stan.

Additional Inherited Members

3.6.1 Detailed Description

Reprezentacja konca gry.

3.6.2 Member Function Documentation

3.6.2.1 eventListener()

```
void GameOver::eventListener (
            sf::Event & event )  [override], [virtual]
```

Reaguje na event od gracza.

Parameters

event	Zdarzenie od gracza

Implements **Level** (p. 24).

3.6.2.2 updateCurrentStateOfCurrentGameLevel()

```
void GameOver::updateCurrentStateOfCurrentGameLevel (
            float dtime )  [override], [virtual]
```

Aktualizuje stan.

Parameters

dtime	Czas renderowania i przetwarzania poprzedniej klatki

Implements **Level** (p. 25).

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

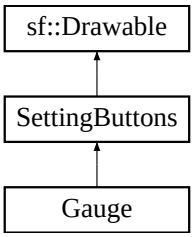
- SFML/GameOver.h
- SFML/GameOver.cpp

3.7 Gauge Class Reference

Reprezentacja slidera do wyboru szybkosci spadania.

`#include <Gauge.h>`

Inheritance diagram for Gauge:



Public Member Functions

- **Gauge** (std::string name, int posx, int posy, int sizex, int sizey, int padding)
  *Konstruktor **Gauge** (p. 18).*
- void **selectSetting** (int x) override
- void **selectRightSetting** () override
  Funkcja realizujaca przesuwanie slidera w prawo.
- void **selectLeftSetting** () override
  Funkcja realizujaca przesuwanie slidera w lewo.
- int **getSelectedSetting** () const override
  Funkcja zwracajaca wybrana pozycje slidera.

Additional Inherited Members

3.7.1 Detailed Description

Reprezentacja slidera do wyboru szybkosci spadania.

3.7.2 Constructor & Destructor Documentation

3.7.2.1 Gauge()

```
Gauge::Gauge (
            std::string name,
            int posx,
            int posy,
            int sizex,
            int sizey,
            int padding )
```

Konstruktor **Gauge** (p. 18).

Parameters

name	Opis slidera
posx	Pozycja slidera na osi OX
posy	Pozycja slidera na osi OY
sizex	Rozmiar w p^{3}aszczy§nie OX
sizey	Rozmiar w p^{3}aszczy§nie OY
padding	Rozmiar marginesu

3.7.3 Member Function Documentation

3.7.3.1 getSelectedSetting()

```
int Gauge::getSelectedSetting ( ) const  [override], [virtual]
```

Funkcja zwracajaca wybrana pozycje slidera.

Returns

Wybrana pozycje slidera

Reimplemented from **SettingButtons** (p. 32).

3.7.3.2 selectLeftSetting()

```
void Gauge::selectLeftSetting ( )  [override], [virtual]
```

Funkcja realizujaca przesuwanie slidera w lewo.

Przesuwa pozycje suwaka w lewo

Reimplemented from **SettingButtons** (p. 30).

3.7.3.3 selectRightSetting()

```
void Gauge::selectRightSetting ( )  [override], [virtual]
```

Funkcja realizujaca przesuwanie slidera w prawo.

Przesuwa pozycje suwaka w prawo

Reimplemented from **SettingButtons** (p. 30).

3.7.3.4 selectSetting()

```
void Gauge::selectSetting (
            int x )  [override], [virtual]
```

Parameters

x	

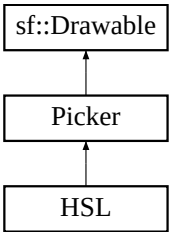
Reimplemented from **SettingButtons** (p. 32).

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

- SFML/Gauge.h
- SFML/Gauge.cpp

3.8 HSL Class Reference

Inheritance diagram for HSL:



Public Member Functions

- void **draw_to_color_pixels** ()
- **HSL** (const wchar_t ∗Name, sf::Vector2f Position, const wchar_t ∗Prefix)
- void **update_color** (double Param)

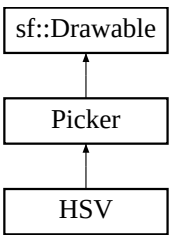
Additional Inherited Members

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

- SFML/SFML.cpp

3.9 HSV Class Reference

Inheritance diagram for HSV:



Public Member Functions

- void **draw_to_color_pixels** ()
- **HSV** (const wchar_t ∗Name, sf::Vector2f Position, const wchar_t ∗Prefix)
- void **update_color** (double Param)

Additional Inherited Members

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

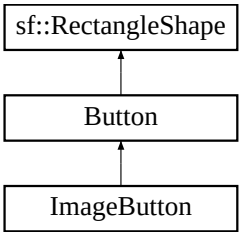
- SFML/SFML.cpp

3.10 ImageButton Class Reference

Reprezentacja wyboru obrazka w MENU.

`#include <ImageButton.h>`

Inheritance diagram for ImageButton:



Public Member Functions

- **ImageButton** (float width, float height, const sf::String &text, unsigned int size, const sf::Texture &tx)

  *Konstruuje **ImageButton** (p. 22).*
- void **selectThisButton** () override

  Funkcja ustawiajaca wybranego obrazek jako zaznaczony.
- void **unselectThisButton** () override

  Funkcja ustawiajaca nie wybrany obrazek jako niezaznaczone.

Additional Inherited Members

3.10.1 Detailed Description

Reprezentacja wyboru obrazka w MENU.

3.10.2 Constructor & Destructor Documentation

3.10.2.1 ImageButton()

```
ImageButton::ImageButton (
            float width,
            float height,
            const sf::String & text,
            unsigned int size,
            const sf::Texture & tx )
```

Konstruuje **ImageButton** (p. 22).

Parameters

width	Szerokos w pikselach
height	Wysokos w pikselach
text	Wyswietlany tekst
size	D^{3}ugos tekstu
tx	Textura

3.10.3 Member Function Documentation

3.10.3.1 selectThisButton()

```
void ImageButton::selectThisButton ( )  [override], [virtual]
```

Funkcja ustawiajaca wybranego obrazek jako zaznaczony.

Zmiana koloru t^{3}a

Reimplemented from **Button** (p. 6).

3.10.3.2 unselectThisButton()

```
void ImageButton::unselectThisButton ( )  [override], [virtual]
```

Funkcja ustawiajaca nie wybrany obrazek jako niezaznaczone.

Default'owy kolor t^{3}a

Reimplemented from **Button** (p. 7).

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

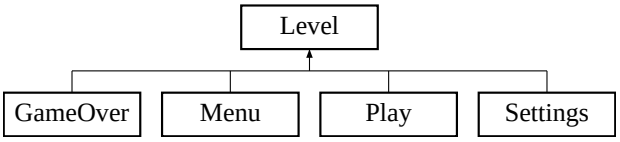
- SFML/ImageButton.h
- SFML/ImageButton.cpp

3.11 Level Class Reference

Reprezentacja stanu abstrakcyjnego.

`#include <Level.h>`

Inheritance diagram for Level:



Public Member Functions

- **Level** ()=default

  Domyslny konstruktor obiektu stanu abstrakcyjnego.
- virtual ~**Level** ()

  Wirtualny destruktor stanu abstrakcyjnego.
- virtual void **eventListener** (sf::Event &event)=0

  Reaguje na event od gracza. Czysto abstrakcyjna metoda.
- virtual void **updateCurrentStateOfCurrentGameLevel** (float dtime)=0

  Aktualizuje stan. Czysto abstrakcyjna metoda.
- virtual void **renderCurrentStateOfCurrentGameLevel** ()=0

  Renderuje aktualny stan. Czysto abstrakcyjna metoda.

Protected Attributes

- bool **_isCurrentLevelUpdated** = true

3.11.1 Detailed Description

Reprezentacja stanu abstrakcyjnego.

3.11.2 Member Function Documentation

3.11.2.1 eventListener()

```
virtual void Level::eventListener (
            sf::Event & event )  [pure virtual]
```

Reaguje na event od gracza. Czysto abstrakcyjna metoda.

Parameters

event	Zdarzenie od gracza

Implemented in **Settings** (p. 33), **Play** (p. 28), **Menu** (p. 26), and **GameOver** (p. 18).

3.11.2.2 updateCurrentStateOfCurrentGameLevel()

```
virtual void Level::updateCurrentStateOfCurrentGameLevel (
            float dtime )  [pure virtual]
```

Aktualizuje stan. Czysto abstrakcyjna metoda.

Parameters

dtime	Czas renderowania i przetwarzania poprzedniej klatki

Implemented in **Menu** (p. 26), **GameOver** (p. 18), **Settings** (p. 33), and **Play** (p. 29).

3.11.3 Member Data Documentation

3.11.3.1 _isCurrentLevelUpdated

```
bool Level::_isCurrentLevelUpdated = true  [protected]
```

Informacja czy stan zosta zaktualizowany

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

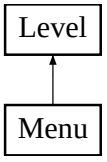
- SFML/Level.h

3.12 Menu Class Reference

Reprezentuje MENU gowne programu.

```
#include <Menu.h>
```

Inheritance diagram for Menu:



Public Member Functions

- **Menu** ()

  *Konstruuje **Menu** (p. 25).*
- void **eventListener** (sf::Event &event) override

  Reaguje na event od gracza. Czysto abstrakcyjna metoda.
- void **updateCurrentStateOfCurrentGameLevel** (float dtime) override

  Aktualizuje stan. Czysto abstrakcyjna metoda.
- void **renderCurrentStateOfCurrentGameLevel** () override

  Renderuje aktualny stan. Czysto abstrakcyjna metoda.

Static Public Attributes

- static bool **isSoundMuted** = true

  Czy wyciszy dwiek.

Additional Inherited Members

3.12.1 Detailed Description

Reprezentuje MENU gowne programu.

3.12.2 Member Function Documentation

3.12.2.1 eventListener()

```
void Menu::eventListener (
            sf::Event & event )  [override], [virtual]
```

Reaguje na event od gracza. Czysto abstrakcyjna metoda.

Parameters

event	Zdarzenie od gracza

Implements **Level** (p. 24).

3.12.2.2 updateCurrentStateOfCurrentGameLevel()

```
void Menu::updateCurrentStateOfCurrentGameLevel (
            float dtime )  [override], [virtual]
```

Aktualizuje stan. Czysto abstrakcyjna metoda.

Parameters

dtime	Czas renderowania i przetwarzania poprzedniej klatki

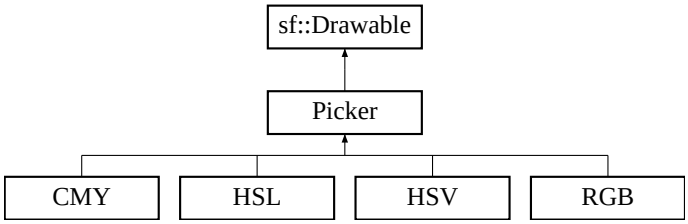
Implements **Level** (p. 25).

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

- SFML/Menu.h
- SFML/Menu.cpp

3.13 Picker Class Reference

Inheritance diagram for Picker:



Public Member Functions

- **Picker** (const wchar_t ∗Name, sf::Vector2f Position, const wchar_t ∗Prefix)
- void **outtextxy** (sf::RenderTarget &target, float x, float y, const wchar_t ∗str) const
- virtual void **draw** (sf::RenderTarget &target, sf::RenderStates states) const

Protected Member Functions

- virtual void **draw_to_color_pixels** ()

Protected Attributes

- sf::Font **font**
- sf::Uint8 ∗ **colors_pixels**
- sf::Text ∗ **text**
- sf::Texture ∗ **colors_texture**
- sf::Sprite ∗ **colors_sprite**
- double **param** =0.5
- const wchar_t ∗ **name**
- const wchar_t ∗ **prefix**
- sf::Vector2f **position**
- const int **colors_size** = 200
- const int **radius** = colors_size/2
- double ∗∗ **shadow**

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

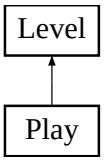
- SFML/SFML.cpp

3.14 Play Class Reference

Gowna funkcja gry.

`#include <Play.h>`

Inheritance diagram for Play:



Public Member Functions

- **Play** ()
  Konstruktor Gry.
- void **eventListener** (sf::Event &event) override
  Funkcja odpowiedzialna za rejstrownie zdarzen klawiatury.
- void **updateCurrentStateOfCurrentGameLevel** (float dt) override
  Aktualizuje stan gry.
- void **renderCurrentStateOfCurrentGameLevel** () override
  Renderuje stan gry.

Additional Inherited Members

3.14.1 Detailed Description

Gowna funkcja gry.

3.14.2 Member Function Documentation

3.14.2.1 eventListener()

```
void Play::eventListener (
            sf::Event & event )  [override], [virtual]
```

Funkcja odpowiedzialna za rejstrownie zdarzen klawiatury.

Parameters

event	Zdarzenie klawiatury

Implements **Level** (p. 24).

3.14.2.2 updateCurrentStateOfCurrentGameLevel()

```
void Play::updateCurrentStateOfCurrentGameLevel (
            float dt )  [override], [virtual]
```

Aktualizuje stan gry.

Parameters

dt	Czas renderowania poprzedniej klatki gry

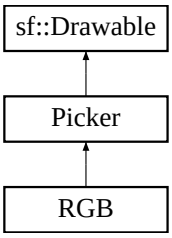
Implements **Level** (p. 25).

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

- SFML/Play.h
- SFML/Play.cpp

3.15 RGB Class Reference

Inheritance diagram for RGB:



Public Member Functions

- void **draw_to_color_pixels** ()
- **RGB** (const wchar_t ∗Name, sf::Vector2f Position, const wchar_t ∗Prefix)
- void **update_color** (double Param)
- virtual void **draw** (sf::RenderTarget &target, sf::RenderStates states) const

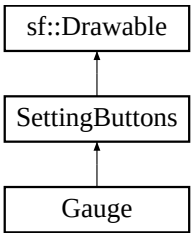
Additional Inherited Members

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

- SFML/SFML.cpp

3.16 SettingButtons Class Reference

Inheritance diagram for SettingButtons:



Public Member Functions

- **SettingButtons** (std::string name, int posx, int posy, int sizex, int sizey, int padding)

  1. Konstruktor zestwau przyciskow
- **SettingButtons** (std::string name, std::vector< std::string > opt, int posx, int posy, int sizex, int sizey, int padding)

  1. Konstruktor zestwau przyciskow
- virtual void **selectSetting** (int x)

  Funckja pozwalajaca na zaznaczenie wybranego przycisku.
- virtual void **selectRightSetting** ()

  Przesuniecie zaznaczenia przycisku w lewo.
- virtual void **selectLeftSetting** ()

  Przesuniecie zaznaczenia przycisku w prawo.
- virtual int **getSelectedSetting** () const

  Funkcja zwracajaca indeks aktywnej opcji.
- void **highlightSelectedSetting** ()

  Funkcja zaznaczajaca zestaw przycikow jako aktywny.
- void **unhighlightSelectedSetting** ()

  Funkcja zaznaczajaca zestaw przycikow jako nieaktywny.

Protected Attributes

- int **_posx**
- int **_posy**
- int **_selected**
- sf::RectangleShape **_high**
- sf::Text **_name**

3.16.1 Constructor & Destructor Documentation

3.16.1.1 SettingButtons() [1/2]

```
SettingButtons::SettingButtons (
            std::string name,
            int posx,
            int posy,
            int sizex,
            int sizey,
            int padding )
```

1. Konstruktor zestwau przyciskow

Parameters

name	Tekst wyswietlany przy zestawie przyciskow
posx	Pozycja x w pikselach
posy	Pozycja y w pikselach
sizex	Szerokos pojedynczego przycisku
sizey	Wysokos pojedynczego przycisku
padding	Odstepy pomiedzy przyciskami w zestwie

3.16.1.2 SettingButtons() [2/2]

```
SettingButtons::SettingButtons (
            std::string name,
            std::vector< std::string > opt,
            int posx,
            int posy,
            int sizex,
            int sizey,
            int padding )
```

1. Konstruktor zestwau przyciskow

Parameters

name	Tekst wyswietlany przy zestawie przyciskow
opt	Wektor stringow przechowujacy nazwy przyciskow, ktore tworzymy
posx	Pozycja x w pikselach
posy	Pozycja y w pikselach
sizex	Szerokos pojedynczego przycisku
sizey	Wysokos pojedynczego przycisku
padding	Odstepy pomiedzy przyciskami w zestwie

3.16.2 Member Function Documentation

3.16.2.1 getSelectedSetting()

```
int SettingButtons::getSelectedSetting ( ) const  [virtual]
```

Funkcja zwracajaca indeks aktywnej opcji.

Returns

Indeks aktywnej opcji

Reimplemented in **Gauge** (p. 20).

3.16.2.2 selectSetting()

```
void SettingButtons::selectSetting (
            int x )  [virtual]
```

Funckja pozwalajaca na zaznaczenie wybranego przycisku.

Parameters

Indeks	elementu, ktory ma by zaznaczony

Reimplemented in **Gauge** (p. 20).

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

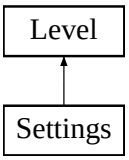
- SFML/SettingButtons.h
- SFML/OptionButtons.cpp

3.17 Settings Class Reference

Reprezentuje stan bycia w konfiguracji gry.

```
#include <Settings.h>
```

Inheritance diagram for Settings:



Public Member Functions

- **Settings** ()

  *Konstruktor bezargumentowy obiektu **Settings** (p. 32).*
- void **eventListener** (sf::Event &event) override

  Funkcja odpowiedzialna za rejstrownie zdarzen klawiatury.
- void **updateCurrentStateOfCurrentGameLevel** (float dt) override

  Aktualizuja stanu menu.
- void **renderCurrentStateOfCurrentGameLevel** () override

  Renderowanie stanu okna.

Additional Inherited Members

3.17.1 Detailed Description

Reprezentuje stan bycia w konfiguracji gry.

3.17.2 Member Function Documentation

3.17.2.1 eventListener()

```
void Settings::eventListener (
            sf::Event & event )  [override], [virtual]
```

Funkcja odpowiedzialna za rejstrownie zdarzen klawiatury.

Parameters

event	Zdarzenie klawiatury

Implements **Level** (p. 24).

3.17.2.2 updateCurrentStateOfCurrentGameLevel()

```
void Settings::updateCurrentStateOfCurrentGameLevel (
            float dt )  [override], [virtual]
```

Aktualizuja stanu menu.

Parameters

dt	Czas renderowania poprzedniej klatki gry

Implements **Level** (p. 25).

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

- SFML/Settings.h
- SFML/Settings.cpp