Arcade

Generated by Doxygen 1.9.1

Chapter 1

README

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

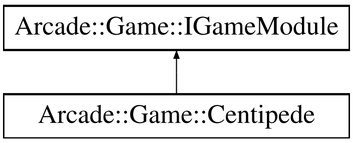
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Class Documentation

4.1 Arcade::Game::Centipede Class Reference

Inheritance diagram for Arcade::Game::Centipede:



Public Member Functions

- std::pair< int, int > getResolution () const

  Get the Resolution of the game.
- void start ()

  Start the game.
- void end ()

  End the game.
- const size_t & getScore () const

  Get the Score.
- void setInputs (std::vector< Arcade::Graphics::IDisplayModule::Events >)

  Set the User's Keyboard Inputs.
- const std::unordered_map< std::pair< int, int >, unsigned int > & **getPixels** ()
- void refresh ()

  Refresh the game.
- void reset ()

  Restart the game.
- unsigned int **compresFromRgba** (unsigned char r, unsigned char g, unsigned char b, unsigned char a)
- rgbaColors **decompressFromRgba** (unsigned int color)
- bool **isColision** (std::pair< int, int > pos1, std::pair< int, int > pos2)
- void **ennemySpawn** (unsigned int lenght)
- void **obstaclesSpawn** (int x, int y)

Protected Attributes

- std::pair< int, int > **_resolution**
- long unsigned int **_score**
- struct obstacle ∗ **_obstacles**
- std::unique_ptr< struct ennemy[ ]> **_ennemy**
- struct laser **_laser**
- struct player **_player**
- int **_ennemyCptr**

4.1.1 Member Function Documentation

4.1.1.1 getResolution()

```
std::pair<int, int> Arcade::Game::Centipede::getResolution ( ) const  [virtual]
```

Get the Resolution of the game.

Returns

std::pair<int x, int y>

Implements Arcade::Game::IGameModule.

4.1.1.2 getScore()

```
const size_t& Arcade::Game::Centipede::getScore ( ) const  [virtual]
```

Get the Score.

Returns

const size_t& score

Implements Arcade::Game::IGameModule.

4.1.1.3 setInputs()

```
void Arcade::Game::Centipede::setInputs (
             std::vector< Arcade::Graphics::IDisplayModule::Events > e )  [virtual]
```

Set the User's Keyboard Inputs.

Parameters

Events	key

Implements Arcade::Game::IGameModule.

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

- includes/Games/Centipede.hpp
- src/Games/Centipede/Centipede.cpp
- src/Games/Centipede/Ennemy.cpp
- src/Games/Centipede/Obstacles.cpp
- src/Games/Centipede/Screen.cpp

4.2 Arcade::Core Class Reference

Public Member Functions

- Core (std::string firstLib)

  Construct a new Core and open the first graphical lib.
- ∼Core ()

  Destroy the Core.
- void loop ()

  Main loop of the project.
- void displayMenu ()

  Display the Arcade's menu.
- unsigned int compresFromRgba (unsigned char r, unsigned char g, unsigned char b, unsigned char a)

  Compress an Rgba Color.
- void getLibs ()

  Open /lib and store the libs.
- void menuEventHandling ()

  Handle Events in the main Menu.
- void gameEventHandling ()

  Handle Events in games.
- void changeGraphicalLib (std::string libName)

  Load a new graphical lib.
- void changeGameLib (std::string libName)

  Load a new game lib.
- void setUserName (void)

  Set the User Name.
- void launchGame (void)

  Launch the selected game.
- void putMapsPixels (const std::unordered_map< long, unsigned int > &pixels)

  Put each pixels of the map.

Protected Attributes

- DLLoader **libsLoader**
- std::unique_ptr< Graphics::IDisplayModule > **graphicalLib**
- std::vector< std::string > **graphicalLibs**
- std::string **graphicalName**
- std::string **graphicalNameCleared**
- int **graphicalLibUsed**
- std::unique_ptr< Game::IGameModule > **gameLib**
- std::vector< std::string > **gameLibs**
- std::string **gameName**
- std::string **gameNameCleared**
- int **gameLibUsed**
- bool **openMenu**
- std::vector< Graphics::IDisplayModule::Events > **eventInputs**
- Graphics::IDisplayModule::Events **event**
- int **select** = 0
- std::string **userName**

4.2.1 Constructor & Destructor Documentation

4.2.1.1 Core()

```
Arcade::Core::Core (
             std::string firstLib )
```

Construct a new Core and open the first graphical lib.

Parameters

firstLib	

4.2.2 Member Function Documentation

4.2.2.1 changeGameLib()

```
void Arcade::Core::changeGameLib (
             std::string libName )
```

Load a new game lib.

Parameters

libName	

4.2.2.2 changeGraphicalLib()

```
void Arcade::Core::changeGraphicalLib (
            std::string libName )
```

Load a new graphical lib.

Parameters

libName	

4.2.2.3 compresFromRgba()

```
unsigned int Arcade::Core::compresFromRgba (
            unsigned char r,
            unsigned char g,
            unsigned char b,
            unsigned char a )
```

Compress an Rgba Color.

Parameters

r	red
g	green
b	blue
a	alpha

Returns

unsigned int compressedColor

4.2.2.4 putMapsPixels()

```
void Arcade::Core::putMapsPixels (
            const std::unordered_map< long, unsigned int > & pixels )
```

Put each pixels of the map.

Parameters

pixels	

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

- includes/Core.hpp
- src/Core.cpp

4.3 DLLoader Class Reference

Public Member Functions

- DLLoader ()

  Construct a new DLLoader object.
- void loadGraphicalHandler (std::string libname)

  Open and store a graphical dynamic lib.
- void loadGameHandler (std::string libname)

  Open and store a game dynamic lib.
- template<typename T >
  T ∗ getGameInstance (std::string func)

  Get the Game Instance.
- ∼DLLoader ()

  Destroy the DLLoader.
- template<typename T >
  T ∗ getGraphicalInstance (std::string func)

  Get the Graphical Instance.
- bool checkGraphical (std::string libname)

  Check if a dynamic lib is whether a game or graphical lib.

Protected Attributes

- void ∗ **graphicalHandler** = nullptr
- void ∗ **gameHandler** = nullptr
- void ∗ **checkHandler** = nullptr

4.3.1 Member Function Documentation

4.3.1.1 checkGraphical()

```
bool DLLoader::checkGraphical (
             std::string libname )  [inline]
```

Check if a dynamic lib is whether a game or graphical lib.

Parameters

libname	

Returns

true if the lib is a graphical lib

false if the lib is a game lib

4.3.1.2 getGameInstance()

```
template<typename T >
T* DLLoader::getGameInstance (
             std::string func )  [inline]
```

Get the Game Instance.

Template Parameters

T	func prototype

Parameters

func	funcName

Returns

T∗ func(T)

4.3.1.3 getGraphicalInstance()

```
template<typename T >
T* DLLoader::getGraphicalInstance (
             std::string func )  [inline]
```

Get the Graphical Instance.

Template Parameters

T	func prototype

Parameters

func	funcName

Returns

T∗ func(T)

4.3.1.4 loadGameHandler()

```
void DLLoader::loadGameHandler (
            std::string libname )  [inline]
```

Open and store a game dynamic lib.

Parameters

libname	

4.3.1.5 loadGraphicalHandler()

```
void DLLoader::loadGraphicalHandler (
            std::string libname )  [inline]
```

Open and store a graphical dynamic lib.

Parameters

libname	

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

- includes/DLLoader.hpp

4.4 ennemy Struct Reference

Public Attributes

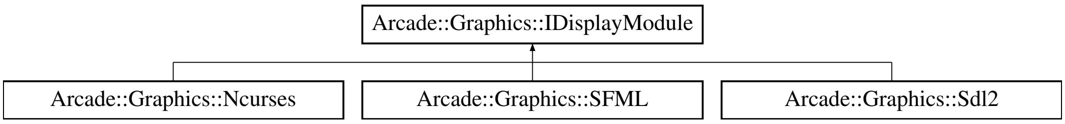
- std::pair< int, int > **_pos** = {0, 0}
- bool **_alive** = true
- bool **_head** = false
- char **direction** = 'r'

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

- includes/Games/Centipede.hpp

4.5 Arcade::Graphics::IDisplayModule Class Reference

Inheritance diagram for Arcade::Graphics::IDisplayModule:



Public Types

- enum **Events** {
  A , **B** , **C** , **D** ,
  E , **F** , **G** , **H** ,
  I , **J** , **K** , **L** ,
  M , **N** , **O** , **P** ,
  Q , **R** , **S** , **T** ,
  U , **V** , **W** , **X** ,
  Y , **Z** , **keyUp** , **keyDown** ,
  keyLeft , **keyRight** , **key1** , **key2** ,
  key3 , **key4** , **key5** , **key6** ,
  key7 , **key8** , **key9** , **key0** ,
  keySpace , **keyEscape** , **keyEnter** , **rightClic** ,
  leftClic , **backSpace** , **close** }

Public Member Functions

- virtual void setFrameRateLimit (int limit)=0

  Set the window's Frame Rate Limit.
- virtual void putpixel (int x, int y, unsigned int color)=0

  Put a pixel on the window.
- virtual void puttext (int x, int y, unsigned int color, const std::string &str)=0

  Put text on the window.
- virtual std::vector< Events > getEvents ()=0

  Get the window's events.
- virtual std::pair< int, int > getMousePos ()=0

  Get the Mouse Position.
- virtual bool isOpen ()=0

  Get the window's status.
- virtual void clearWin ()=0

  Clear the window.
- virtual void refreshWin ()=0

  Refresh the window.

4.5.1 Member Function Documentation

4.5.1.1 getEvents()

```
virtual std::vector<Events> Arcade::Graphics::IDisplayModule::getEvents ( )  [pure virtual]
```

Get the window's events.

Returns

std::vector<Events>

Implemented in Arcade::Graphics::SFML, Arcade::Graphics::Sdl2, and Arcade::Graphics::Ncurses.

4.5.1.2 getMousePos()

```
virtual std::pair<int, int> Arcade::Graphics::IDisplayModule::getMousePos ( )  [pure virtual]
```

Get the Mouse Position.

Returns

std::pair<x, y>

Implemented in Arcade::Graphics::SFML, Arcade::Graphics::Sdl2, and Arcade::Graphics::Ncurses.

4.5.1.3 isOpen()

```
virtual bool Arcade::Graphics::IDisplayModule::isOpen ( )  [pure virtual]
```

Get the window's status.

Returns

true Opened

false Closed

Implemented in Arcade::Graphics::SFML, Arcade::Graphics::Sdl2, and Arcade::Graphics::Ncurses.

4.5.1.4 putpixel()

```
virtual void Arcade::Graphics::IDisplayModule::putpixel (
             int x,
             int y,
             unsigned int color )  [pure virtual]
```

Put a pixel on the window.

Parameters

x	x pos
y	y pos
color	color of the pixel

Implemented in Arcade::Graphics::SFML, Arcade::Graphics::Sdl2, and Arcade::Graphics::Ncurses.

4.5.1.5 puttext()

```
virtual void Arcade::Graphics::IDisplayModule::puttext (
             int x,
             int y,
             unsigned int color,
             const std::string & str )  [pure virtual]
```

Put text on the window.

Parameters

x	x pos
y	y pos
color	color of the text
str	text

Implemented in Arcade::Graphics::SFML, Arcade::Graphics::Sdl2, and Arcade::Graphics::Ncurses.

4.5.1.6 setFrameRateLimit()

```
virtual void Arcade::Graphics::IDisplayModule::setFrameRateLimit (
             int limit )  [pure virtual]
```

Set the window's Frame Rate Limit.

Parameters

limit	

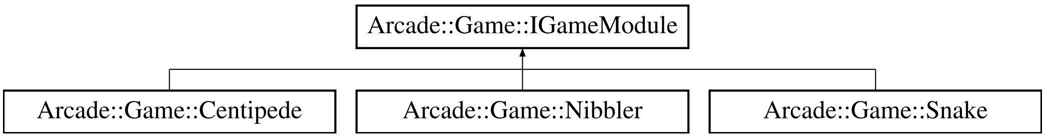
Implemented in Arcade::Graphics::SFML, Arcade::Graphics::Sdl2, and Arcade::Graphics::Ncurses.

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

- includes/Graphical/IDisplayModule.hpp

4.6 Arcade::Game::IGameModule Class Reference

Inheritance diagram for Arcade::Game::IGameModule:



Public Member Functions

- virtual std::pair< int, int > getResolution () const =0

  Get the Resolution of the game.
- virtual void start ()=0

  Start the game.
- virtual void end ()=0

  End the game.
- virtual const size_t & getScore () const =0

  Get the Score.
- virtual void setInputs (std::vector< Arcade::Graphics::IDisplayModule::Events > e)=0

  Set the User's Keyboard Inputs.
- virtual const std::unordered_map< long, unsigned int > & getPixels () const =0

  Get a map of pixels.
- virtual void refresh ()=0

  Refresh the game.
- virtual void reset ()=0

  Restart the game.

4.6.1 Member Function Documentation

4.6.1.1 getPixels()

```
virtual const std::unordered_map<long, unsigned int>& Arcade::Game::IGameModule::getPixels ( )
const [pure virtual]
```

Get a map of pixels.

Returns

const std::unordered_map<long position, unsigned int color>&

Implemented in Arcade::Game::Snake, and Arcade::Game::Nibbler.

4.6.1.2 getResolution()

```
virtual std::pair<int, int> Arcade::Game::IGameModule::getResolution ( ) const  [pure virtual]
```

Get the Resolution of the game.

Returns

std::pair<int x, int y>

Implemented in Arcade::Game::Snake, Arcade::Game::Nibbler, and Arcade::Game::Centipede.

4.6.1.3 getScore()

```
virtual const size_t& Arcade::Game::IGameModule::getScore ( ) const  [pure virtual]
```

Get the Score.

Returns

const size_t& score

Implemented in Arcade::Game::Snake, Arcade::Game::Nibbler, and Arcade::Game::Centipede.

4.6.1.4 setInputs()

```
virtual void Arcade::Game::IGameModule::setInputs (
             std::vector< Arcade::Graphics::IDisplayModule::Events > e )  [pure virtual]
```

Set the User's Keyboard Inputs.

Parameters

Events	key

Implemented in Arcade::Game::Nibbler, Arcade::Game::Centipede, and Arcade::Game::Snake.

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

- includes/Games/IGameModule.hpp

4.7 laser Struct Reference

Public Attributes

- std::pair< int, int > **_pos** = {0, 0}

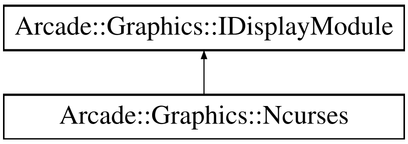
- bool **_alive** = false

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

- includes/Games/Centipede.hpp

4.8 Arcade::Graphics::Ncurses Class Reference

Inheritance diagram for Arcade::Graphics::Ncurses:



Public Member Functions

- Ncurses ()

  Init Ncurses "window".
- ~Ncurses ()

  Destroy the Ncurses "window".
- void setFrameRateLimit (int limit)

  Set the window's Frame Rate Limit.
- void putpixel (int x, int y, unsigned int color)

  Put a pixel on the window.
- void puttext (int x, int y, unsigned int color, const std::string &str)

  Put text on the window.
- std::vector< Events > getEvents ()

  Get the window's events.
- std::pair< int, int > getMousePos ()

  Get the Mouse Position.
- bool isOpen ()

  Get the window's status.
- void clearWin ()

  Clear the window.
- void refreshWin ()

  Refresh the window.

Additional Inherited Members

4.8.1 Member Function Documentation

4.8.1.1 getEvents()

```
std::vector< Arcade::Graphics::Ncurses::Events > Arcade::Graphics::Ncurses::getEvents ( )
[virtual]
```

Get the window's events.

Returns

std::vector<Events>

Implements Arcade::Graphics::IDisplayModule.

4.8.1.2 getMousePos()

```
std::pair< int, int > Arcade::Graphics::Ncurses::getMousePos ( ) [virtual]
```

Get the Mouse Position.

Returns

std::pair<x, y>

Implements Arcade::Graphics::IDisplayModule.

4.8.1.3 isOpen()

```
bool Arcade::Graphics::Ncurses::isOpen ( ) [virtual]
```

Get the window's status.

Returns

true Opened

false Closed

Implements Arcade::Graphics::IDisplayModule.

4.8.1.4 putpixel()

```
void Arcade::Graphics::Ncurses::putpixel (
            int x,
            int y,
            unsigned int color ) [virtual]
```

Put a pixel on the window.

Parameters

x	x pos
y	y pos
color	color of the pixel

Implements Arcade::Graphics::IDisplayModule.

4.8.1.5 puttext()

```
void Arcade::Graphics::Ncurses::puttext (
             int x,
             int y,
             unsigned int color,
             const std::string & str )  [virtual]
```

Put text on the window.

Parameters

x	x pos
y	y pos
color	color of the text
str	text

Implements Arcade::Graphics::IDisplayModule.

4.8.1.6 setFrameRateLimit()

```
void Arcade::Graphics::Ncurses::setFrameRateLimit (
             int limit )  [virtual]
```

Set the window's Frame Rate Limit.

Parameters

limit	

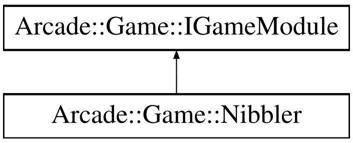
Implements Arcade::Graphics::IDisplayModule.

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

- includes/Graphical/Ncurses.hpp
- src/Graphical/Ncurses.cpp

4.9 Arcade::Game::Nibbler Class Reference

Inheritance diagram for Arcade::Game::Nibbler:



Public Member Functions

- Nibbler ()

    Init Nibbler.
- ∼Nibbler ()

    Destroy Nibbler.
- std::pair< int, int > getResolution () const

    Get the Resolution of the game.
- void start ()

    Start the game.
- void end ()

    End the game.
- const size_t & getScore () const

    Get the Score.
- void setInputs (std::vector< Arcade::Graphics::IDisplayModule::Events >)

    Set the User's Keyboard Inputs.
- const std::unordered_map< long, unsigned int > & getPixels () const

    Get a map of pixels.
- void refresh ()

    Refresh the game.
- void reset ()

    Restart the game.
- unsigned int compresFromRgba (unsigned char r, unsigned char g, unsigned char b, unsigned char a)

    Compress a RGBA color into a single unsigned int.

4.9.1 Member Function Documentation

4.9.1.1 compresFromRgba()

```
unsigned int Arcade::Game::Nibbler::compresFromRgba (
             unsigned char r,
             unsigned char g,
             unsigned char b,
             unsigned char a )
```

Compress a RGBA color into a single unsigned int.

Parameters

r	Red
g	Green
b	Blue
a	Alpha

Returns

unsigned int color

4.9.1.2 getPixels()

```
const std::unordered_map< long, unsigned int > & Arcade::Game::Nibbler::getPixels ( ) const
[virtual]
```

Get a map of pixels.

Returns

const std::unordered_map<long position, unsigned int color>&

Implements Arcade::Game::IGameModule.

4.9.1.3 getResolution()

```
std::pair< int, int > Arcade::Game::Nibbler::getResolution ( ) const  [virtual]
```

Get the Resolution of the game.

Returns

std::pair<int x, int y>

Implements Arcade::Game::IGameModule.

4.9.1.4 getScore()

```
const size_t & Arcade::Game::Nibbler::getScore ( ) const  [virtual]
```

Get the Score.

Returns

const size_t& score

Implements Arcade::Game::IGameModule.

4.9.1.5 setInputs()

```
void Arcade::Game::Nibbler::setInputs (
             std::vector< Arcade::Graphics::IDisplayModule::Events >  )  [virtual]
```

Set the User's Keyboard Inputs.

Parameters

Events	key

Implements Arcade::Game::IGameModule.

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

- includes/Games/Nibbler.hpp
- src/Games/Nibbler.cpp

4.10 obstacle Struct Reference

Public Attributes

- std::pair< int, int > **_pos** = {0, 0}
- unsigned int **_hp** = 5
- obstacle ∗ **_next** = NULL

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

- includes/Games/Centipede.hpp

4.11 player Struct Reference

Public Attributes

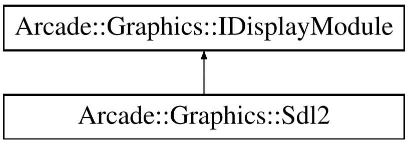
- std::pair< int, int > **_pos** = {0, 0}
- unsigned int **_lives** = 5

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

- includes/Games/Centipede.hpp

4.12 Arcade::Graphics::Sdl2 Class Reference

Inheritance diagram for Arcade::Graphics::Sdl2:



Public Member Functions

- Sdl2 (int width, int height)

  Init SDL2 window.
- ∼Sdl2 ()

  Destroy the SDL2 window.
- void setFrameRateLimit (int limit)

  Set the window's Frame Rate Limit.
- void putpixel (int x, int y, unsigned int color)

  Put a pixel on the window.
- void puttext (int x, int y, unsigned int color, const std::string &str)

  Put text on the window.
- std::vector< Events > getEvents ()

  Get the window's events.
- std::pair< int, int > getMousePos ()

  Get the Mouse Position.
- bool isOpen ()

  Get the window's status.
- void clearWin ()

  Clear the window.
- void refreshWin ()

  Refresh the window.

Protected Attributes

- SDL_Window ∗ **win**
- SDL_Renderer ∗ **rend**
- bool **_isOpen**
- int **frameRateLimit**
- TTF_Font ∗ **font**

Additional Inherited Members

4.12.1 Constructor & Destructor Documentation

4.12.1.1 Sdl2()

```
Arcade::Graphics::Sdl2::Sdl2 (
            int width,
            int height )
```

Init SDL2 window.

Parameters

width	
height	

4.12.2 Member Function Documentation

4.12.2.1 getEvents()

```
std::vector< Arcade::Graphics::IDisplayModule::Events > Arcade::Graphics::Sdl2::getEvents ( )
[virtual]
```

Get the window's events.

Returns

std::vector<Events>

Implements Arcade::Graphics::IDisplayModule.

4.12.2.2 getMousePos()

```
std::pair< int, int > Arcade::Graphics::Sdl2::getMousePos ( ) [virtual]
```

Get the Mouse Position.

Returns

std::pair<x, y>

Implements Arcade::Graphics::IDisplayModule.

4.12.2.3 isOpen()

```
bool Arcade::Graphics::Sdl2::isOpen ( ) [virtual]
```

Get the window's status.

Returns

true Opened

false Closed

Implements Arcade::Graphics::IDisplayModule.

4.12.2.4 putpixel()

```
void Arcade::Graphics::Sdl2::putpixel (
            int x,
            int y,
            unsigned int color ) [virtual]
```

Put a pixel on the window.

Parameters

x	x pos
y	y pos
color	color of the pixel

Implements Arcade::Graphics::IDisplayModule.

4.12.2.5 puttext()

```
void Arcade::Graphics::Sdl2::puttext (
             int x,
             int y,
             unsigned int color,
             const std::string & str )  [virtual]
```

Put text on the window.

Parameters

x	x pos
y	y pos
color	color of the text
str	text

Implements Arcade::Graphics::IDisplayModule.

4.12.2.6 setFrameRateLimit()

```
void Arcade::Graphics::Sdl2::setFrameRateLimit (
             int limit )  [virtual]
```

Set the window's Frame Rate Limit.

Parameters

limit	

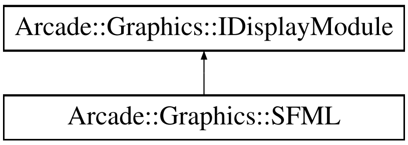
Implements Arcade::Graphics::IDisplayModule.

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

- includes/Graphical/Sdl2.hpp
- src/Graphical/Sdl2.cpp

4.13 Arcade::Graphics::SFML Class Reference

Inheritance diagram for Arcade::Graphics::SFML:



Public Member Functions

- SFML (int width, int height)

  Init SFML window.
- void setFrameRateLimit (int limit)

  Set the window's Frame Rate Limit.
- void putpixel (int x, int y, unsigned int color)

  Put a pixel on the window.
- void puttext (int x, int y, unsigned int color, const std::string &str)

  Put text on the window.
- std::vector< Events > getEvents ()

  Get the window's events.
- std::pair< int, int > getMousePos ()

  Get the Mouse Position.
- bool isOpen ()

  Get the window's status.
- void clearWin ()

  Clear the window.
- void refreshWin ()

  Refresh the window.
- std::vector< Events > getEventsMouse (std::vector< Arcade::Graphics::SFML::Events > events, sf::Event event)

  Get the Events Mouse object.
- std::vector< Events > getEventsKeys (std::vector< Arcade::Graphics::SFML::Events > events, sf::Event event)

  Get the Events Keys object.

Protected Attributes

- std::unique_ptr< sf::RenderWindow > **_window**
- sf::Font **_font**
- bool **_isOpen**

Additional Inherited Members

4.13.1 Constructor & Destructor Documentation

4.13.1.1 SFML()

```
Arcade::Graphics::SFML::SFML (
              int width,
              int height )
```

Init SFML window.

Parameters

width	
height	

4.13.2 Member Function Documentation

4.13.2.1 getEvents()

```
std::vector< Arcade::Graphics::SFML::Events > Arcade::Graphics::SFML::getEvents ( )  [virtual]
```

Get the window's events.

Returns

std::vector<Events>

Implements Arcade::Graphics::IDisplayModule.

4.13.2.2 getEventsKeys()

```
std::vector< Arcade::Graphics::SFML::Events > Arcade::Graphics::SFML::getEventsKeys (
              std::vector< Arcade::Graphics::SFML::Events > events,
              sf::Event event )
```

Get the Events Keys object.

Parameters

events	
event	

Returns

std::vector<Events>

4.13.2.3 getEventsMouse()

```
std::vector< Arcade::Graphics::SFML::Events > Arcade::Graphics::SFML::getEventsMouse (
            std::vector< Arcade::Graphics::SFML::Events > events,
            sf::Event event )
```

Get the Events Mouse object.

Parameters

events	
event	

Returns

std::vector<Events>

4.13.2.4 getMousePos()

```
std::pair< int, int > Arcade::Graphics::SFML::getMousePos ( ) [virtual]
```

Get the Mouse Position.

Returns

std::pair<x, y>

Implements Arcade::Graphics::IDisplayModule.

4.13.2.5 isOpen()

```
bool Arcade::Graphics::SFML::isOpen ( ) [virtual]
```

Get the window's status.

Returns

true Opened

false Closed

Implements Arcade::Graphics::IDisplayModule.

4.13.2.6 putpixel()

```
void Arcade::Graphics::SFML::putpixel (
            int x,
            int y,
            unsigned int color ) [virtual]
```

Put a pixel on the window.

Parameters

x	x pos
y	y pos
color	color of the pixel

Implements Arcade::Graphics::IDisplayModule.

4.13.2.7 puttext()

```
void Arcade::Graphics::SFML::puttext (
             int x,
             int y,
             unsigned int color,
             const std::string & str )  [virtual]
```

Put text on the window.

Parameters

x	x pos
y	y pos
color	color of the text
str	text

Implements Arcade::Graphics::IDisplayModule.

4.13.2.8 setFrameRateLimit()

```
void Arcade::Graphics::SFML::setFrameRateLimit (
             int limit )  [virtual]
```

Set the window's Frame Rate Limit.

Parameters

limit	

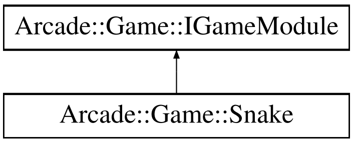
Implements Arcade::Graphics::IDisplayModule.

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

- includes/Graphical/SFML.hpp
- src/Graphical/SFML.cpp

4.14 Arcade::Game::Snake Class Reference

Inheritance diagram for Arcade::Game::Snake:



Public Member Functions

- Snake ()

  Init Snake.
- ~Snake ()

  Destroy Snake.
- std::pair< int, int > getResolution () const

  Get the Resolution of the game.
- void start ()

  Start the game.
- void end ()

  End the game.
- const size_t & getScore () const

  Get the Score.
- void setInputs (std::vector< Arcade::Graphics::IDisplayModule::Events > e)

  Set the User's Keyboard Inputs.
- const std::unordered_map< long, unsigned int > & getPixels () const

  Get a map of pixels.
- void refresh ()

  Refresh the game.
- void reset ()

  Restart the game.
- unsigned int compresFromRgba (unsigned char r, unsigned char g, unsigned char b, unsigned char a)

  Compress a RGBA color into a single unsigned int.
- void convertMap (void)

  Convert the snake and fruit into an std::unordered_map<long, unsigned int>
- void moveSnake (void)

  Move the snake.
- void eventHandling (void)

  Handle the events.
- void checkFruit (void)

  Check fruit and snake colisions.
- void checkDeath (void)

  Check snake colisions with walls and himself.

Protected Attributes

- int **snakeLength**
- bool **alive**
- size_t **score**
- char **direction**
- std::vector< long > **snake**
- std::vector< Arcade::Graphics::IDisplayModule::Events > **events**
- std::unordered_map< long, unsigned int > **map**
- long **fruit**

4.14.1 Member Function Documentation

4.14.1.1 compresFromRgba()

```
unsigned int Arcade::Game::Snake::compresFromRgba (
            unsigned char r,
            unsigned char g,
            unsigned char b,
            unsigned char a )
```

Compress a RGBA color into a single unsigned int.

Parameters

r	Red
g	Green
b	Blue
a	Alpha

Returns

unsigned int color

4.14.1.2 getPixels()

```
const std::unordered_map< long, unsigned int > & Arcade::Game::Snake::getPixels ( ) const
[virtual]
```

Get a map of pixels.

Returns

const std::unordered_map<long position, unsigned int color>&

Implements Arcade::Game::IGameModule.

4.14.1.3 getResolution()

```
std::pair< int, int > Arcade::Game::Snake::getResolution ( ) const  [virtual]
```

Get the Resolution of the game.

Returns

std::pair<int x, int y>

Implements Arcade::Game::IGameModule.

4.14.1.4 getScore()

```
const size_t & Arcade::Game::Snake::getScore ( ) const  [virtual]
```

Get the Score.

Returns

const size_t& score

Implements Arcade::Game::IGameModule.

4.14.1.5 setInputs()

```
void Arcade::Game::Snake::setInputs (
             std::vector< Arcade::Graphics::IDisplayModule::Events > e )  [virtual]
```

Set the User's Keyboard Inputs.

Parameters

Events	key

Implements Arcade::Game::IGameModule.

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

- includes/Games/Snake.hpp
- src/Games/Snake.cpp

Index