Arcade

Generated by Doxygen 1.9.1

Wed Apr 5 2023 19:31:50

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

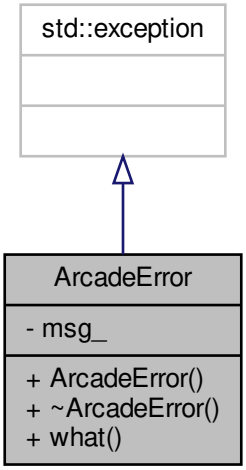
Chapter 4

Data Structure Documentation

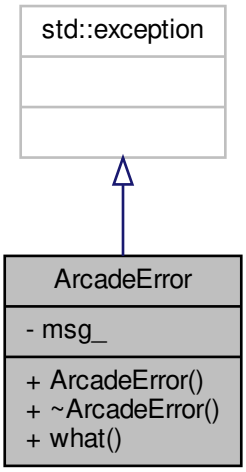
4.1 ArcadeError Class Reference

`#include <Error.hpp>`

Inheritance diagram for ArcadeError:



Collaboration diagram for ArcadeError:



Public Member Functions

- ArcadeError (const char ∗msg)
- ∼ArcadeError () override=default
- const char ∗ what () const noexcept override

Private Attributes

- const char ∗ msg_

4.1.1 Constructor & Destructor Documentation

4.1.1.1 ArcadeError()

```
ArcadeError::ArcadeError (
            const char * msg )
```

4.1.1.2 ∼ArcadeError()

```
ArcadeError::~ArcadeError ( )  [override], [default]
```

4.1.2 Member Function Documentation

4.1.2.1 what()

```
const char * ArcadeError::what ( ) const  [override], [noexcept]
```

4.1.3 Field Documentation

4.1.3.1 msg_

```
const char* ArcadeError::msg_  [private]
```

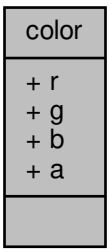
The documentation for this class was generated from the following files:

- include/error/Error.hpp
- src/error/Error.cpp

4.2 color Struct Reference

```
#include <IGraphic.hpp>
```

Collaboration diagram for color:



Data Fields

- unsigned short r
- unsigned short g
- unsigned short b
- unsigned short a

4.2.1 Field Documentation

4.2.1.1 a

```
unsigned short color::a
```

4.2.1.2 b

```
unsigned short color::b
```

4.2.1.3 g

```
unsigned short color::g
```

4.2.1.4 r

```
unsigned short color::r
```

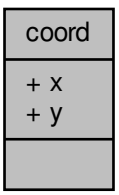
The documentation for this struct was generated from the following file:

- include/graphics/IGraphic.hpp

4.3 coord Struct Reference

```
#include <IGraphic.hpp>
```

Collaboration diagram for coord:



Data Fields

- int x
- int y

4.3.1 Field Documentation

4.3.1.1 x

```
int coord::x
```

4.3.1.2 y

```
int coord::y
```

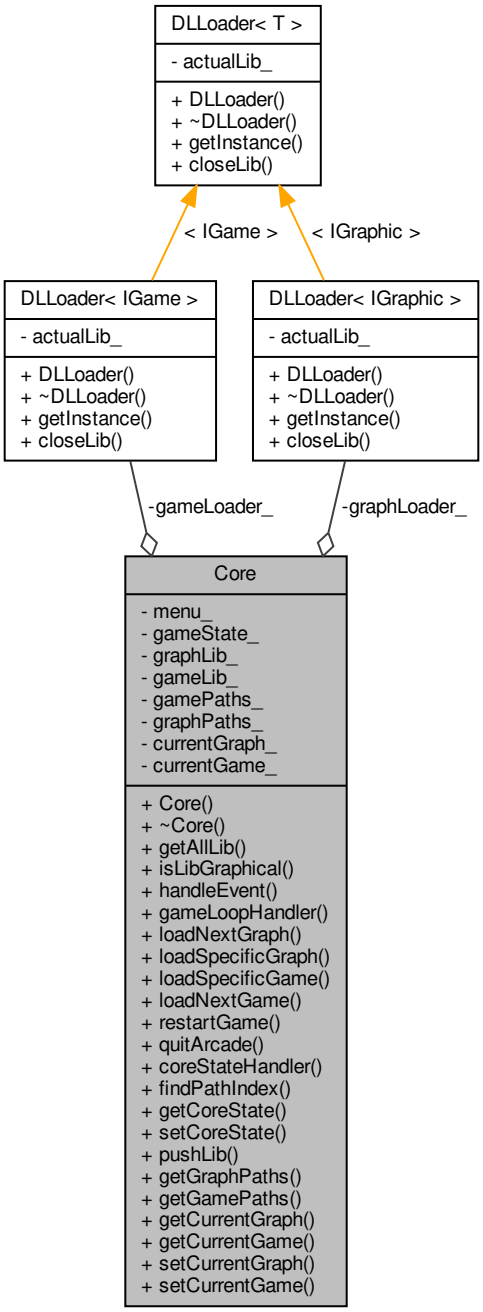
The documentation for this struct was generated from the following file:

- include/graphics/IGraphic.hpp

4.4 Core Class Reference

```
#include <Core.hpp>
```

Collaboration diagram for Core:



Public Member Functions

- Core (const char ∗libName)
- ∼Core ()=default
- void getAllLib ()
- void isLibGraphical (const std::string &libName)
- void handleEvent ()

- void gameLoopHandler ()
- void loadNextGraph ()
- void loadSpecificGraph (std::string path)
- void loadSpecificGame (std::string path)
- void loadNextGame ()
- void restartGame ()
- void quitArcade ()
- void coreStateHandler ()
- int findPathIndex (const std::string &path, const std::vector< std::string > &vec) const
- GState getCoreState () const
- void setCoreState (const GState &state)
- void pushLib (const std::string &path, std::vector< std::string > &container)
- std::vector< std::string > getGraphPaths ()
- std::vector< std::string > getGamePaths ()
- std::string getCurrentGraph () const
- std::string getCurrentGame () const
- void setCurrentGraph (const std::string &currentGraph)
- void setCurrentGame (const std::string &currentGame)

Private Attributes

- std::unique_ptr< Menu > menu_
- GState gameState_
- std::unique_ptr< IGraphic > graphLib_
- std::unique_ptr< IGame > gameLib_
- DLLoader< IGraphic > graphLoader_ { DLLoader<IGraphic>() }
- DLLoader< IGame > gameLoader_ { DLLoader<IGame>() }
- std::vector< std::string > gamePaths_
- std::vector< std::string > graphPaths_
- std::string currentGraph_
- std::string currentGame_

4.4.1 Constructor & Destructor Documentation

4.4.1.1 Core()

```
Core::Core (
            const char * libName )
```

4.4.1.2 ~Core()

```
Core::~Core ( )  [default]
```

4.4.2 Member Function Documentation

4.4.2.1 coreStateHandler()

```
void Core::coreStateHandler ( )
```

4.4.2.2 findPathIndex()

```
int Core::findPathIndex (
            const std::string & path,
            const std::vector< std::string > & vec ) const
```

4.4.2.3 gameLoopHandler()

```
void Core::gameLoopHandler ( )
```

4.4.2.4 getAllLib()

```
void Core::getAllLib ( )
```

4.4.2.5 getCoreState()

```
GState Core::getCoreState ( ) const
```

4.4.2.6 getCurrentGame()

```
std::string Core::getCurrentGame ( ) const
```

4.4.2.7 getCurrentGraph()

```
std::string Core::getCurrentGraph ( ) const
```

4.4.2.8 getGamePaths()

```
std::vector< std::string > Core::getGamePaths ( )
```

4.4.2.9 getGraphPaths()

```
std::vector< std::string > Core::getGraphPaths ( )
```

4.4.2.10 handleEvent()

```
void Core::handleEvent ( )
```

4.4.2.11 isLibGraphical()

```
void Core::isLibGraphical (
            const std::string & libName )
```

4.4.2.12 loadNextGame()

```
void Core::loadNextGame ( )
```

4.4.2.13 loadNextGraph()

```
void Core::loadNextGraph ( )
```

4.4.2.14 loadSpecificGame()

```
void Core::loadSpecificGame (
            std::string path )
```

4.4.2.15 loadSpecificGraph()

```
void Core::loadSpecificGraph (
            std::string path )
```

4.4.2.16 pushLib()

```
void Core::pushLib (
            const std::string & path,
            std::vector< std::string > & container )
```

4.4.2.17 quitArcade()

```
void Core::quitArcade ( )
```

4.4.2.18 restartGame()

```
void Core::restartGame ( )
```

4.4.2.19 setCoreState()

```
void Core::setCoreState (
            const GState & state )
```

4.4.2.20 setCurrentGame()

```
void Core::setCurrentGame (
            const std::string & currentGame )
```

4.4.2.21 setCurrentGraph()

```
void Core::setCurrentGraph (
            const std::string & currentGraph )
```

4.4.3 Field Documentation

4.4.3.1 currentGame_

```
std::string Core::currentGame_  [private]
```

4.4.3.2 currentGraph_

```
std::string Core::currentGraph_  [private]
```

4.4.3.3 gameLib_

```
std::unique_ptr<IGame> Core::gameLib_  [private]
```

4.4.3.4 gameLoader_

```
DLLoader<IGame> Core::gameLoader_ { DLLoader<IGame>() }  [private]
```

4.4.3.5 gamePaths_

```
std::vector<std::string> Core::gamePaths_  [private]
```

4.4.3.6 gameState_

```
GState Core::gameState_  [private]
```

4.4.3.7 graphLib_

```
std::unique_ptr<IGraphic> Core::graphLib_  [private]
```

4.4.3.8 graphLoader_

```
DLLoader<IGraphic> Core::graphLoader_ { DLLoader<IGraphic>() }  [private]
```

4.4.3.9 graphPaths_

```
std::vector<std::string> Core::graphPaths_  [private]
```

4.4.3.10 menu_

```
std::unique_ptr<Menu> Core::menu_  [private]
```

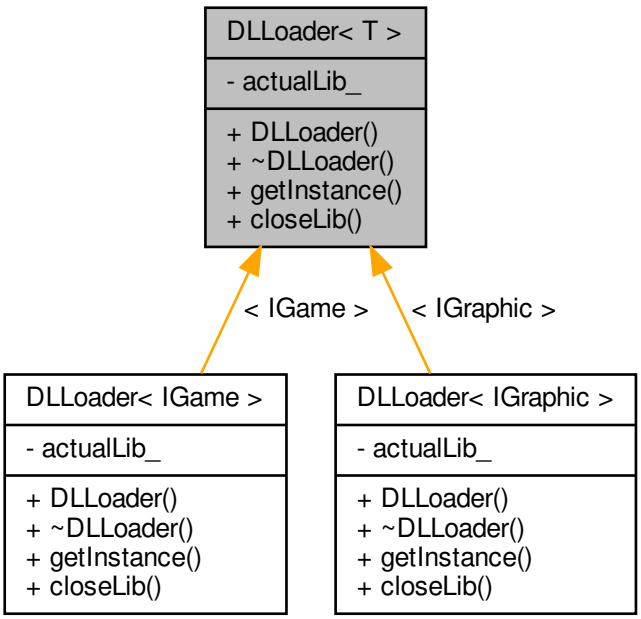
The documentation for this class was generated from the following files:

- include/core/Core.hpp
- src/core/Core.cpp

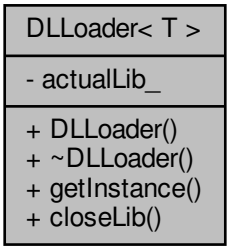
4.5 DLLoader< T > Class Template Reference

```
#include <DLLoader.hpp>
```

Inheritance diagram for DLLoader< T >:



Collaboration diagram for DLLoader< T >:



Public Member Functions

- DLLoader ()=default
- ~DLLoader ()=default
- std::unique_ptr< T > getInstance (std::string filename)
- void closeLib (void)

Private Attributes

- void * actualLib_ { nullptr }

4.5.1 Constructor & Destructor Documentation

4.5.1.1 DLLoader()

```
template<typename T >
DLLoader< T >::DLLoader ( )  [default]
```

4.5.1.2 ~DLLoader()

```
template<typename T >
DLLoader< T >::~DLLoader ( )  [default]
```

4.5.2 Member Function Documentation

4.5.2.1 closeLib()

```
template<typename T >
void DLLoader< T >::closeLib (
             void  )  [inline]
```

4.5.2.2 getInstance()

```
template<typename T >
std::unique_ptr<T> DLLoader< T >::getInstance (
             std::string filename )  [inline]
```

4.5.3 Field Documentation

4.5.3.1 actualLib_

```
template<typename T >
void* DLLoader< T >::actualLib_ { nullptr }  [private]
```

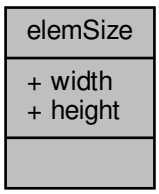
The documentation for this class was generated from the following file:

- include/core/DLLoader.hpp

4.6 elemSize Struct Reference

`#include <IGraphic.hpp>`

Collaboration diagram for elemSize:



Data Fields

- int width
- int height

4.6.1 Field Documentation

4.6.1.1 height

```
int elemSize::height
```

4.6.1.2 width

```
int elemSize::width
```

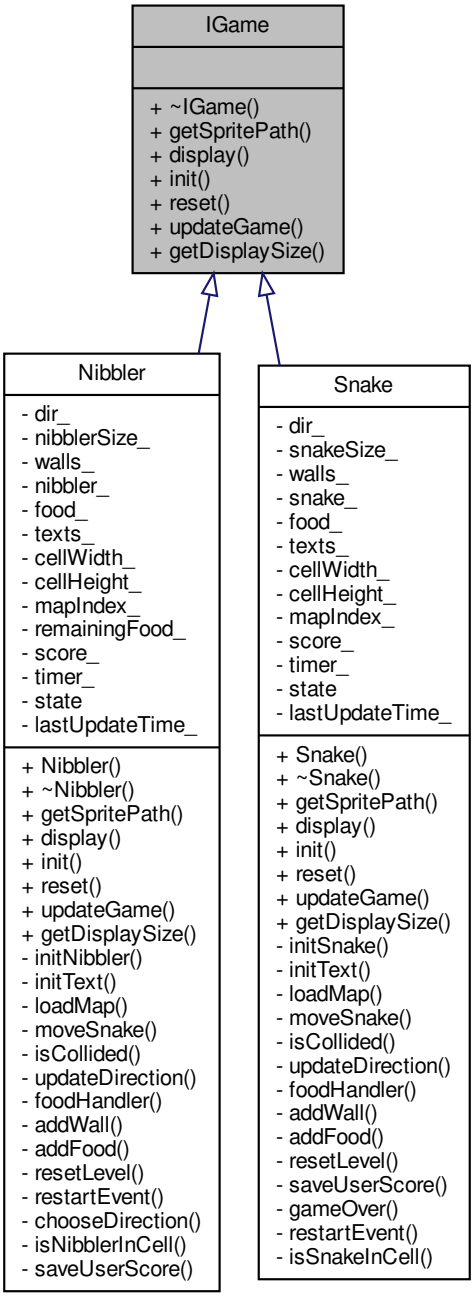
The documentation for this struct was generated from the following file:

- include/graphics/IGraphic.hpp

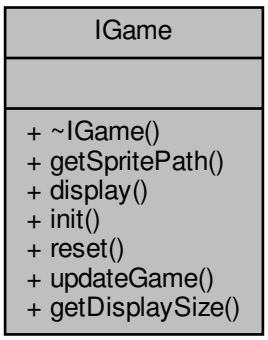
4.7 IGame Class Reference

```
#include <IGame.hpp>
```

Inheritance diagram for IGame:



Collaboration diagram for IGame:



Public Member Functions

- virtual ∼IGame () noexcept=default
- virtual std::string getSpritePath ()=0
- virtual void display (IGraphic &)=0
- virtual int init ()=0
- virtual void reset ()=0
- virtual int updateGame (eventKey evtKey)=0
- virtual elemSize getDisplaySize ()=0

4.7.1 Constructor & Destructor Documentation

4.7.1.1 ∼IGame()

```
virtual IGame::∼IGame ( )  [virtual], [default], [noexcept]
```

4.7.2 Member Function Documentation

4.7.2.1 display()

```
virtual void IGame::display (
            IGraphic &  )  [pure virtual]
```

Implemented in Snake, and Nibbler.

4.7.2.2 getDisplaySize()

```
virtual elemSize IGame::getDisplaySize ( )  [pure virtual]
```

Implemented in Snake, and Nibbler.

4.7.2.3 getSpritePath()

```
virtual std::string IGame::getSpritePath ( )  [pure virtual]
```

Implemented in Snake, and Nibbler.

4.7.2.4 init()

```
virtual int IGame::init ( )  [pure virtual]
```

Implemented in Snake, and Nibbler.

4.7.2.5 reset()

```
virtual void IGame::reset ( )  [pure virtual]
```

Implemented in Snake, and Nibbler.

4.7.2.6 updateGame()

```
virtual int IGame::updateGame (
            eventKey evtKey )  [pure virtual]
```

Implemented in Snake, and Nibbler.

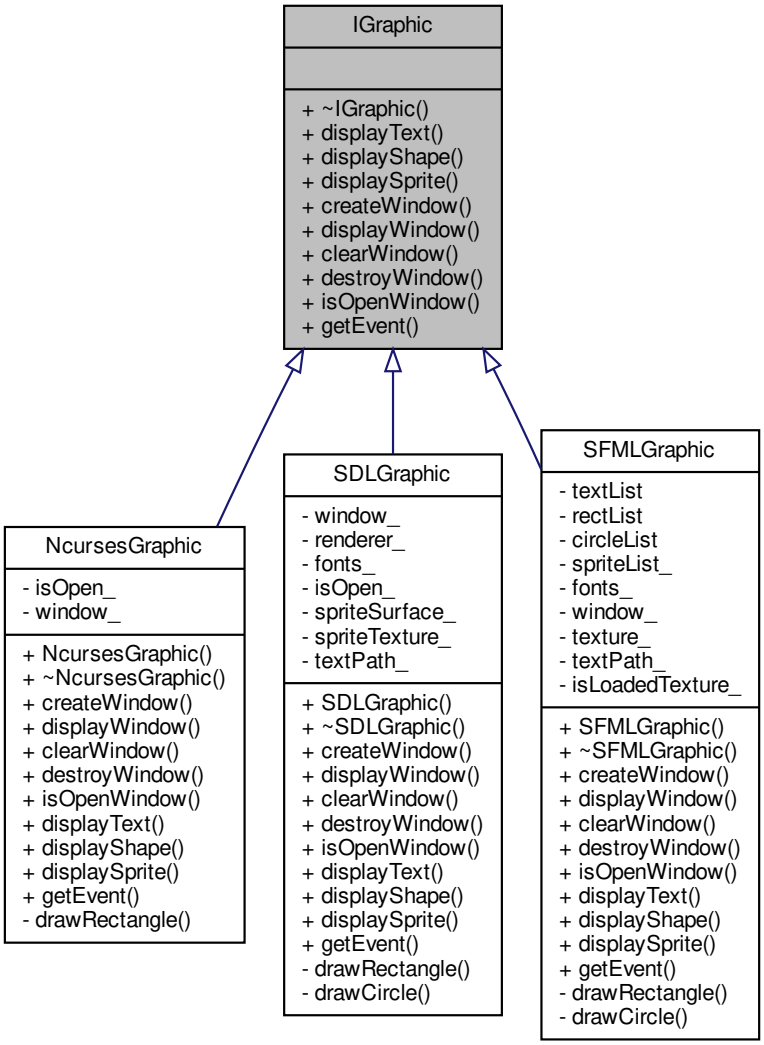
The documentation for this class was generated from the following file:

- include/games/IGame.hpp

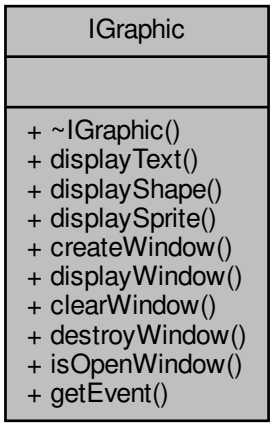
4.8 IGraphic Class Reference

`#include <IGraphic.hpp>`

Inheritance diagram for IGraphic:



Collaboration diagram for IGraphic:



Public Member Functions

- virtual ~IGraphic () noexcept=default
- virtual void displayText (const text &text)=0
- virtual void displayShape (const shape &shape)=0
- virtual void displaySprite (const sprite &sprite)=0
- virtual void createWindow (std::string title, int width, int height)=0
- virtual void displayWindow ()=0
- virtual void clearWindow ()=0
- virtual void destroyWindow ()=0
- virtual bool isOpenWindow ()=0
- virtual eventKey getEvent ()=0

4.8.1 Constructor & Destructor Documentation

4.8.1.1 ~IGraphic()

```
virtual IGraphic::~IGraphic ( )  [virtual], [default], [noexcept]
```

4.8.2 Member Function Documentation

4.8.2.1 clearWindow()

```
virtual void IGraphic::clearWindow ( )  [pure virtual]
```

Implemented in SFMLGraphic, SDLGraphic, and NcursesGraphic.

4.8.2.2 createWindow()

```
virtual void IGraphic::createWindow (
            std::string title,
            int width,
            int height )  [pure virtual]
```

Implemented in SFMLGraphic, SDLGraphic, and NcursesGraphic.

4.8.2.3 destroyWindow()

```
virtual void IGraphic::destroyWindow ( )  [pure virtual]
```

Implemented in SFMLGraphic, SDLGraphic, and NcursesGraphic.

4.8.2.4 displayShape()

```
virtual void IGraphic::displayShape (
            const shape & shape )  [pure virtual]
```

Implemented in SFMLGraphic, SDLGraphic, and NcursesGraphic.

4.8.2.5 displaySprite()

```
virtual void IGraphic::displaySprite (
            const sprite & sprite )  [pure virtual]
```

Implemented in SFMLGraphic, SDLGraphic, and NcursesGraphic.

4.8.2.6 displayText()

```
virtual void IGraphic::displayText (
            const text & text )  [pure virtual]
```

Implemented in SFMLGraphic, SDLGraphic, and NcursesGraphic.

4.8.2.7 displayWindow()

```
virtual void IGraphic::displayWindow ( )  [pure virtual]
```

Implemented in SFMLGraphic, SDLGraphic, and NcursesGraphic.

4.8.2.8 getEvent()

```
virtual eventKey IGraphic::getEvent ( )  [pure virtual]
```

Implemented in SFMLGraphic, SDLGraphic, and NcursesGraphic.

4.8.2.9 isOpenWindow()

```
virtual bool IGraphic::isOpenWindow ( )  [pure virtual]
```

Implemented in SFMLGraphic, SDLGraphic, and NcursesGraphic.

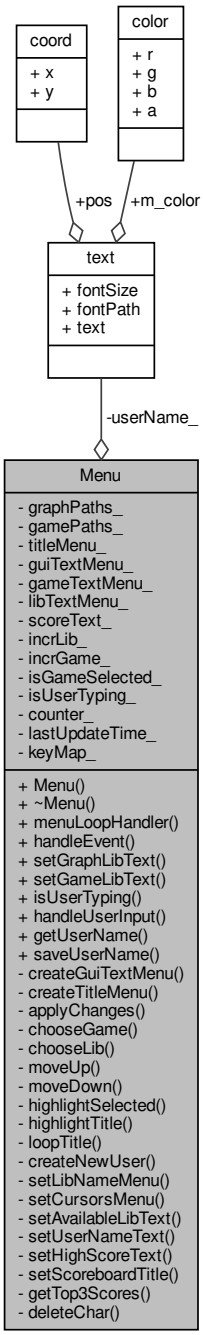
The documentation for this class was generated from the following file:

- include/graphics/IGraphic.hpp

4.9 Menu Class Reference

```
#include <Menu.hpp>
```

Collaboration diagram for Menu:



Public Member Functions

- Menu ()
- ~Menu ()=default
- void menuLoopHandler (IGraphic &graphLib, Core &core)
- void handleEvent (eventKey evt, Core &core)
- void setGraphLibText ()

- void setGameLibText ()
- bool isUserTyping () const
- void handleUserInput (eventKey evt, Core &core)
- std::string getUserName () const
- void saveUserName ()

Private Member Functions

- void createGuiTextMenu ()
- void createTitleMenu ()
- void applyChanges (Core &core)
- void chooseGame ()
- void chooseLib ()
- void moveUp (Core &core)
- void moveDown (Core &core)
- void highlightSelected (Core &core)
- void highlightTitle (const int count)
- void loopTitle ()
- void createNewUser ()
- void setLibNameMenu ()
- void setCursorsMenu ()
- void setAvailableLibText ()
- void setUserNameText ()
- void setHighScoreText ()
- void setScoreboardTitle ()
- std::vector< std::string > getTop3Scores ()
- void deleteChar ()

Private Attributes

- std::vector< std::string > graphPaths_
- std::vector< std::string > gamePaths_
- std::vector< text > titleMenu_
- std::vector< text > guiTextMenu_
- std::vector< text > gameTextMenu_
- std::vector< text > libTextMenu_
- std::vector< text > scoreText_
- text userName_
- int incrLib_ {0}
- int incrGame_ {0}
- bool isGameSelected_ {false}
- bool isUserTyping_ {true}
- int counter_ {0}
- std::chrono::steady_clock::time_point lastUpdateTime_
- std::unordered_map< eventKey, char > keyMap_

4.9.1 Constructor & Destructor Documentation

4.9.1.1 Menu()

```
Menu::Menu ( )
```

4.9.1.2 ~Menu()

```
Menu::~Menu ( )  [default]
```

4.9.2 Member Function Documentation

4.9.2.1 applyChanges()

```
void Menu::applyChanges (
            Core & core )  [private]
```

4.9.2.2 chooseGame()

```
void Menu::chooseGame ( )  [private]
```

4.9.2.3 chooseLib()

```
void Menu::chooseLib ( )  [private]
```

4.9.2.4 createGuiTextMenu()

```
void Menu::createGuiTextMenu ( )  [private]
```

4.9.2.5 createNewUser()

```
void Menu::createNewUser ( )  [private]
```

4.9.2.6 createTitleMenu()

```
void Menu::createTitleMenu ( )  [private]
```

4.9.2.7 deleteChar()

```
void Menu::deleteChar ( )  [private]
```

4.9.2.8 getTop3Scores()

```
std::vector< std::string > Menu::getTop3Scores ( )  [private]
```

4.9.2.9 getUserName()

```
std::string Menu::getUserName ( ) const
```

4.9.2.10 handleEvent()

```
void Menu::handleEvent (
            eventKey evt,
            Core & core )
```

4.9.2.11 handleUserInput()

```
void Menu::handleUserInput (
            eventKey evt,
            Core & core )
```

4.9.2.12 highlightSelected()

```
void Menu::highlightSelected (
            Core & core )  [private]
```

4.9.2.13 highlightTitle()

```
void Menu::highlightTitle (
            const int count )  [private]
```

4.9.2.14 isUserTyping()

```
bool Menu::isUserTyping ( ) const
```

4.9.2.15 loopTitle()

```
void Menu::loopTitle ( )  [private]
```

4.9.2.16 menuLoopHandler()

```
void Menu::menuLoopHandler (
            IGraphic & graphLib,
            Core & core )
```

4.9.2.17 moveDown()

```
void Menu::moveDown (
            Core & core )  [private]
```

4.9.2.18 moveUp()

```
void Menu::moveUp (
            Core & core )  [private]
```

4.9.2.19 saveUserName()

```
void Menu::saveUserName ( )
```

4.9.2.20 setAvailableLibText()

```
void Menu::setAvailableLibText ( )  [private]
```

4.9.2.21 setCursorsMenu()

```
void Menu::setCursorsMenu ( )  [private]
```

4.9.2.22 setGameLibText()

```
void Menu::setGameLibText ( )
```

4.9.2.23 setGraphLibText()

```
void Menu::setGraphLibText ( )
```

4.9.2.24 setHighScoreText()

```
void Menu::setHighScoreText ( )  [private]
```

4.9.2.25 setLibNameMenu()

```
void Menu::setLibNameMenu ( )  [private]
```

4.9.2.26 setScoreboardTitle()

```
void Menu::setScoreboardTitle ( )  [private]
```

4.9.2.27 setUserNameText()

```
void Menu::setUserNameText ( )  [private]
```

4.9.3 Field Documentation

4.9.3.1 counter_

```
int Menu::counter_ {0}  [private]
```

4.9.3.2 gamePaths_

```
std::vector<std::string> Menu::gamePaths_  [private]
```

4.9.3.3 gameTextMenu_

```
std::vector<text> Menu::gameTextMenu_  [private]
```

4.9.3.4 graphPaths_

```
std::vector<std::string> Menu::graphPaths_  [private]
```

4.9.3.5 guiTextMenu_

```
std::vector<text> Menu::guiTextMenu_  [private]
```

4.9.3.6 incrGame_

```
int Menu::incrGame_ {0}  [private]
```

4.9.3.7 incrLib_

```
int Menu::incrLib_ {0}  [private]
```

4.9.3.8 isGameSelected_

```
bool Menu::isGameSelected_ {false}  [private]
```

4.9.3.9 isUserTyping_

```
bool Menu::isUserTyping_ {true}  [private]
```

4.9.3.10 keyMap_

```
std::unordered_map<eventKey, char> Menu::keyMap_  [private]
```

4.9.3.11 lastUpdateTime_

```
std::chrono::steady_clock::time_point Menu::lastUpdateTime_  [private]
```

4.9.3.12 libTextMenu_

```
std::vector<text> Menu::libTextMenu_  [private]
```

4.9.3.13 scoreText_

```
std::vector<text> Menu::scoreText_  [private]
```

4.9.3.14 titleMenu_

```
std::vector<text> Menu::titleMenu_  [private]
```

4.9.3.15 userName_

```
text Menu::userName_  [private]
```

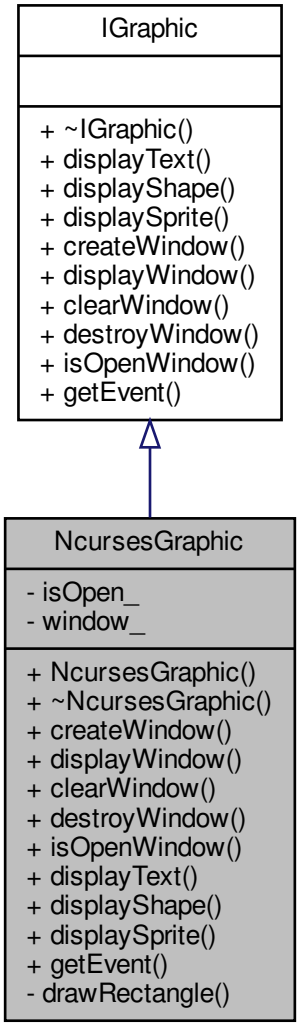
The documentation for this class was generated from the following files:

- include/core/Menu.hpp
- src/core/ActionsMenu.cpp
- src/core/Menu.cpp
- src/core/SetMenuText.cpp
- src/core/UserNameMenu.cpp

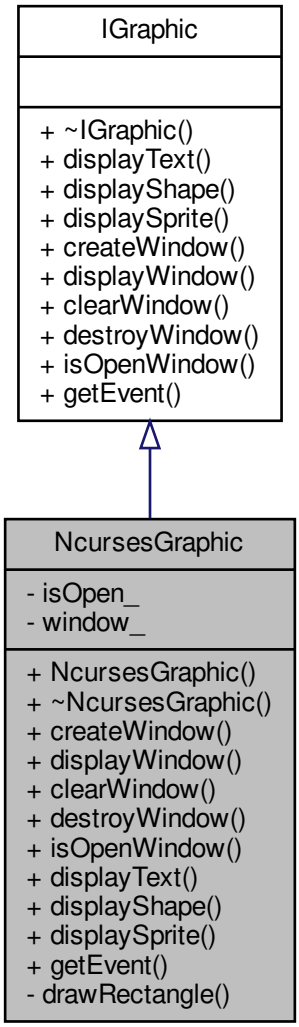
4.10 NcursesGraphic Class Reference

```
#include <NcursesGraphic.hpp>
```

Inheritance diagram for NcursesGraphic:



Collaboration diagram for NcursesGraphic:



Public Member Functions

- NcursesGraphic ()=default
- ~NcursesGraphic () noexcept=default
- void createWindow (std::string title, int width, int height) override
- void displayWindow () override
- void clearWindow () override
- void destroyWindow () override
- bool isOpenWindow () override
- void displayText (const text &text) override
- void displayShape (const shape &shape) override
- void displaySprite (const sprite &sprite) override
- eventKey getEvent () override

Private Member Functions

- void drawRectangle (const shape &shape)

Private Attributes

- bool isOpen_ { false }
- WINDOW * window_

4.10.1 Constructor & Destructor Documentation

4.10.1.1 NcursesGraphic()

```
NcursesGraphic::NcursesGraphic ( )  [default]
```

4.10.1.2 ~NcursesGraphic()

```
NcursesGraphic::~NcursesGraphic ( )  [default], [noexcept]
```

4.10.2 Member Function Documentation

4.10.2.1 clearWindow()

```
void NcursesGraphic::clearWindow ( )  [override], [virtual]
```

Implements IGraphic.

4.10.2.2 createWindow()

```
void NcursesGraphic::createWindow (
            std::string title,
            int width,
            int height )  [override], [virtual]
```

Implements IGraphic.

4.10.2.3 destroyWindow()

```
void NcursesGraphic::destroyWindow ( )  [override], [virtual]
```

Implements IGraphic.

4.10.2.4 displayShape()

```
void NcursesGraphic::displayShape (
            const shape & shape )  [override], [virtual]
```

Implements IGraphic.

4.10.2.5 displaySprite()

```
void NcursesGraphic::displaySprite (
            const sprite & sprite )  [override], [virtual]
```

Implements IGraphic.

4.10.2.6 displayText()

```
void NcursesGraphic::displayText (
            const text & text )  [override], [virtual]
```

Implements IGraphic.

4.10.2.7 displayWindow()

```
void NcursesGraphic::displayWindow ( )  [override], [virtual]
```

Implements IGraphic.

4.10.2.8 drawRectangle()

```
void NcursesGraphic::drawRectangle (
            const shape & shape )  [private]
```

4.10.2.9 getEvent()

```
eventKey NcursesGraphic::getEvent ( )  [override], [virtual]
```

Implements IGraphic.

4.10.2.10 isOpenWindow()

```
bool NcursesGraphic::isOpenWindow ( )  [override], [virtual]
```

Implements IGraphic.

4.10.3 Field Documentation

4.10.3.1 isOpen_

```
bool NcursesGraphic::isOpen_ { false }  [private]
```

4.10.3.2 window_

```
WINDOW* NcursesGraphic::window_  [private]
```

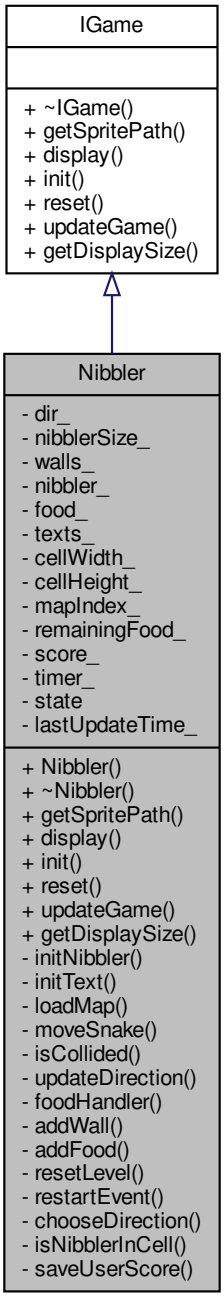
The documentation for this class was generated from the following files:

- include/graphics/ncurses/NcursesGraphic.hpp
- src/graphics/ncurses/NcursesEvent.cpp
- src/graphics/ncurses/NcursesGraphic.cpp

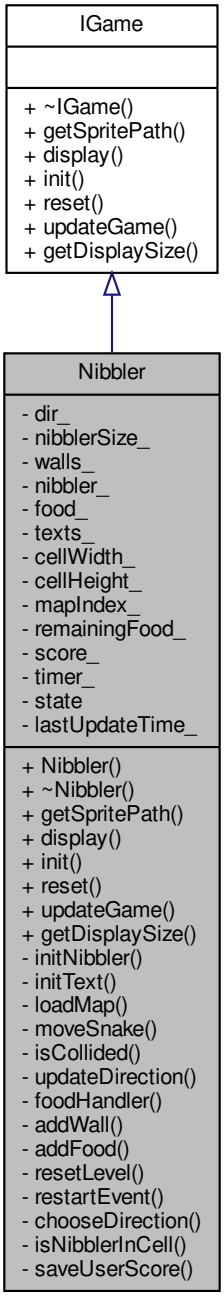
4.11 Nibbler Class Reference

`#include <Nibbler.hpp>`

Inheritance diagram for Nibbler:



Collaboration diagram for Nibbler:



Public Member Functions

- Nibbler ()=default
- ~Nibbler ()=default
- std::string getSpritePath () override
- void display (IGraphic &) override
- int init () override

- void reset () override
- int updateGame (eventKey evtKey) override
- elemSize getDisplaySize () override

Private Member Functions

- void initNibbler (int x, int y)
- void initText ()
- void loadMap (int index)
- void moveSnake (std::vector< shape > &tmp)
- bool isCollided (const shape &s1, const shape &s2)
- void updateDirection (eventKey evtKey)
- void foodHandler ()
- void addWall (int x, int y)
- void addFood (int x, int y)
- void resetLevel ()
- void restartEvent (eventKey evtKey)
- void chooseDirection ()
- bool isNibblerInCell (int x, int y)
- void saveUserScore ()

Private Attributes

- direction dir_ { direction::RIGHT }
- int nibblerSize_ { 4 }
- std::vector< shape > walls_
- std::vector< shape > nibbler_
- std::vector< shape > food_
- std::vector< text > texts_
- int cellWidth_ { 20 }
- int cellHeight_ { 20 }
- int mapIndex_ { 0 }
- int remainingFood_ { 0 }
- int score_ { 0 }
- int timer_ { 0 }
- playerState state { playerState::ALIVE }
- std::chrono::steady_clock::time_point lastUpdateTime_

4.11.1 Constructor & Destructor Documentation

4.11.1.1 Nibbler()

```
Nibbler::Nibbler ( )  [default]
```

4.11.1.2 ~Nibbler()

```
Nibbler::~Nibbler ( )  [default]
```

4.11.2 Member Function Documentation

4.11.2.1 addFood()

```
void Nibbler::addFood (
            int x,
            int y )  [private]
```

4.11.2.2 addWall()

```
void Nibbler::addWall (
            int x,
            int y )  [private]
```

4.11.2.3 chooseDirection()

```
void Nibbler::chooseDirection ( )  [private]
```

4.11.2.4 display()

```
void Nibbler::display (
            IGraphic & graphLib )  [override], [virtual]
```

Implements IGame.

4.11.2.5 foodHandler()

```
void Nibbler::foodHandler ( )  [private]
```

4.11.2.6 getDisplaySize()

```
elemSize Nibbler::getDisplaySize ( )  [override], [virtual]
```

Implements IGame.

4.11.2.7 getSpritePath()

```
std::string Nibbler::getSpritePath ( )  [override], [virtual]
```

Implements IGame.

4.11.2.8 init()

```
int Nibbler::init ( )  [override], [virtual]
```

Implements IGame.

4.11.2.9 initNibbler()

```
void Nibbler::initNibbler (
            int x,
            int y )  [private]
```

4.11.2.10 initText()

```
void Nibbler::initText ( )  [private]
```

4.11.2.11 isCollided()

```
bool Nibbler::isCollided (
            const shape & s1,
            const shape & s2 )  [private]
```

4.11.2.12 isNibblerInCell()

```
bool Nibbler::isNibblerInCell (
            int x,
            int y )  [private]
```

4.11.2.13 loadMap()

```
void Nibbler::loadMap (
            int index )  [private]
```

4.11.2.14 moveSnake()

```
void Nibbler::moveSnake (
            std::vector< shape > & tmp )  [private]
```

4.11.2.15 reset()

```
void Nibbler::reset ( )  [override], [virtual]
```

Implements IGame.

4.11.2.16 resetLevel()

```
void Nibbler::resetLevel ( )  [private]
```

4.11.2.17 restartEvent()

```
void Nibbler::restartEvent (
            eventKey evtKey )  [private]
```

4.11.2.18 saveUserScore()

```
void Nibbler::saveUserScore ( )  [private]
```

4.11.2.19 updateDirection()

```
void Nibbler::updateDirection (
            eventKey evtKey )  [private]
```

4.11.2.20 updateGame()

```
int Nibbler::updateGame (
            eventKey evtKey )  [override], [virtual]
```

Implements IGame.

4.11.3 Field Documentation

4.11.3.1 cellHeight_

```
int Nibbler::cellHeight_ { 20 }  [private]
```

4.11.3.2 cellWidth_

```
int Nibbler::cellWidth_ { 20 }  [private]
```

4.11.3.3 dir_

```
direction Nibbler::dir_ { direction::RIGHT }  [private]
```

4.11.3.4 food_

```
std::vector<shape> Nibbler::food_  [private]
```

4.11.3.5 lastUpdateTime_

```
std::chrono::steady_clock::time_point Nibbler::lastUpdateTime_  [private]
```

4.11.3.6 mapIndex_

```
int Nibbler::mapIndex_ { 0 }  [private]
```

4.11.3.7 nibbler_

```
std::vector<shape> Nibbler::nibbler_  [private]
```

4.11.3.8 nibblerSize_

```
int Nibbler::nibblerSize_ { 4 }  [private]
```

4.11.3.9 remainingFood_

```
int Nibbler::remainingFood_ { 0 }  [private]
```

4.11.3.10 score_

```
int Nibbler::score_ { 0 }  [private]
```

4.11.3.11 state

```
playerState Nibbler::state { playerState::ALIVE }  [private]
```

4.11.3.12 texts_

```
std::vector<text> Nibbler::texts_  [private]
```

4.11.3.13 timer_

```
int Nibbler::timer_ { 0 }  [private]
```

4.11.3.14 walls_

```
std::vector<shape> Nibbler::walls_  [private]
```

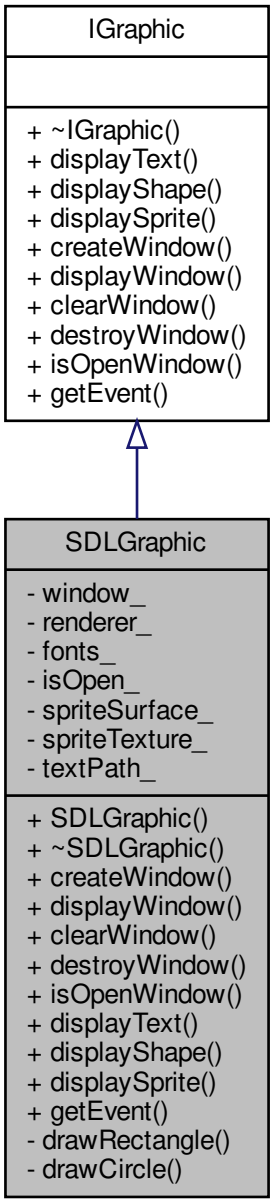
The documentation for this class was generated from the following files:

- include/games/nibbler/Nibbler.hpp
- src/games/nibbler/Compute.cpp
- src/games/nibbler/Display.cpp
- src/games/nibbler/Nibbler.cpp

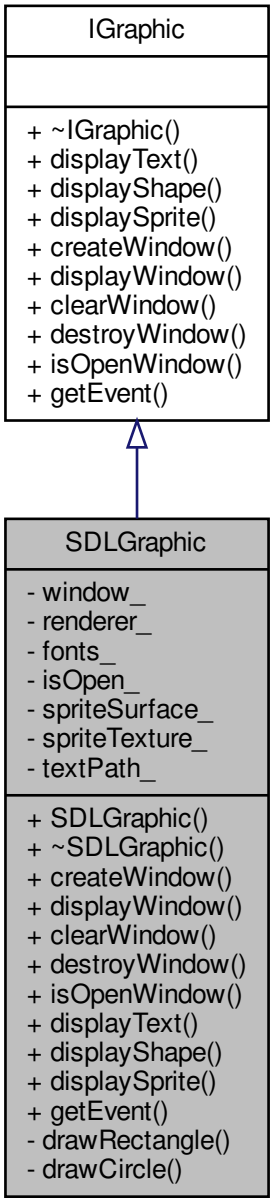
4.12 SDLGraphic Class Reference

```
#include <SDLGraphic.hpp>
```

Inheritance diagram for SDLGraphic:



Collaboration diagram for SDLGraphic:



Public Member Functions

- SDLGraphic ()=default
- ~SDLGraphic () noexcept=default
- void createWindow (std::string title, int width, int height) override
- void displayWindow () override
- void clearWindow () override
- void destroyWindow () override
- bool isOpenWindow () override

- void displayText (const text &text) override
- void displayShape (const shape &shape) override
- void displaySprite (const sprite &sprite) override
- eventKey getEvent () override

Private Member Functions

- void drawRectangle (const shape &shape)
- void drawCircle (const shape &shape)

Private Attributes

- SDL_Window * window_
- SDL_Renderer * renderer_
- std::unordered_map< int, TTF_Font * > fonts_
- bool isOpen_ { false }
- SDL_Surface * spriteSurface_ { NULL }
- SDL_Texture * spriteTexture_ { NULL }
- std::string textPath_

4.12.1 Constructor & Destructor Documentation

4.12.1.1 SDLGraphic()

```
SDLGraphic::SDLGraphic ( )  [default]
```

4.12.1.2 ~SDLGraphic()

```
SDLGraphic::~SDLGraphic ( )  [default], [noexcept]
```

4.12.2 Member Function Documentation

4.12.2.1 clearWindow()

```
void SDLGraphic::clearWindow ( )  [override], [virtual]
```

Implements IGraphic.

4.12.2.2 createWindow()

```
void SDLGraphic::createWindow (
            std::string title,
            int width,
            int height )  [override], [virtual]
```

Implements IGraphic.

4.12.2.3 destroyWindow()

```
void SDLGraphic::destroyWindow ( )  [override], [virtual]
```

Implements IGraphic.

4.12.2.4 displayShape()

```
void SDLGraphic::displayShape (
            const shape & shape )  [override], [virtual]
```

Implements IGraphic.

4.12.2.5 displaySprite()

```
void SDLGraphic::displaySprite (
            const sprite & sprite )  [override], [virtual]
```

Implements IGraphic.

4.12.2.6 displayText()

```
void SDLGraphic::displayText (
            const text & text )  [override], [virtual]
```

Implements IGraphic.

4.12.2.7 displayWindow()

```
void SDLGraphic::displayWindow ( )  [override], [virtual]
```

Implements IGraphic.

4.12.2.8 drawCircle()

```
void SDLGraphic::drawCircle (
            const shape & shape )  [private]
```

4.12.2.9 drawRectangle()

```
void SDLGraphic::drawRectangle (
            const shape & shape )  [private]
```

4.12.2.10 getEvent()

```
eventKey SDLGraphic::getEvent ( )  [override], [virtual]
```

Implements IGraphic.

4.12.2.11 isOpenWindow()

```
bool SDLGraphic::isOpenWindow ( )  [override], [virtual]
```

Implements IGraphic.

4.12.3 Field Documentation

4.12.3.1 fonts_

```
std::unordered_map<int, TTF_Font*> SDLGraphic::fonts_  [private]
```

4.12.3.2 isOpen_

```
bool SDLGraphic::isOpen_ { false }  [private]
```

4.12.3.3 renderer_

```
SDL_Renderer* SDLGraphic::renderer_  [private]
```

4.12.3.4 spriteSurface_

```
SDL_Surface* SDLGraphic::spriteSurface_ { NULL }  [private]
```

4.12.3.5 spriteTexture_

```
SDL_Texture* SDLGraphic::spriteTexture_ { NULL }  [private]
```

4.12.3.6 textPath_

```
std::string SDLGraphic::textPath_  [private]
```

4.12.3.7 window_

```
SDL_Window* SDLGraphic::window_  [private]
```

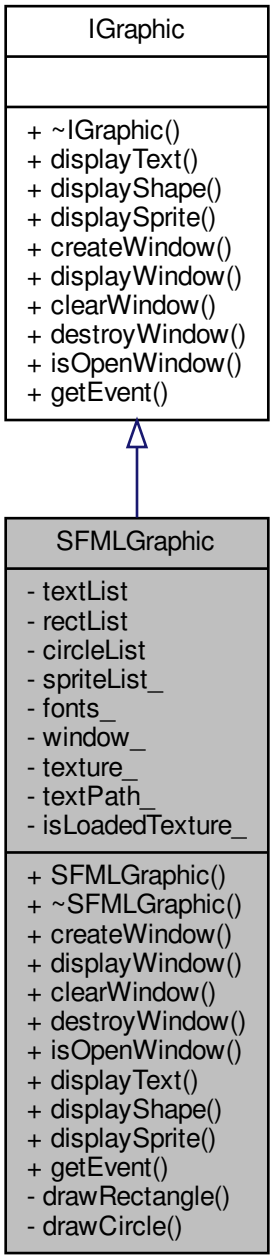
The documentation for this class was generated from the following files:

- include/graphics/sdl/SDLGraphic.hpp
- src/graphics/sdl/SDLEvent.cpp
- src/graphics/sdl/SDLGraphic.cpp

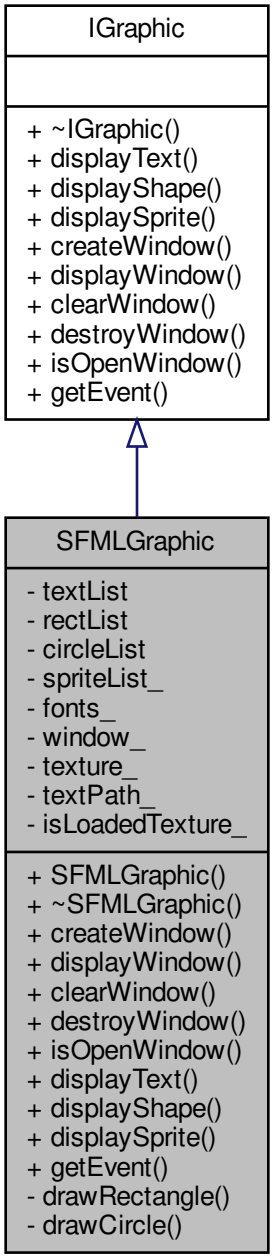
4.13 SFMLGraphic Class Reference

`#include <SFMLGraphic.hpp>`

Inheritance diagram for SFMLGraphic:



Collaboration diagram for SFMLGraphic:



Public Member Functions

- SFMLGraphic ()=default
- ~SFMLGraphic () noexcept=default
- void createWindow (std::string title, int width, int height) override
- void displayWindow () override
- void clearWindow () override

- void destroyWindow () override
- bool isOpenWindow () override
- void displayText (const text &text) override
- void displayShape (const shape &shape) override
- void displaySprite (const sprite &sprite) override
- eventKey getEvent () override

Private Member Functions

- void drawRectangle (const shape &shape)
- void drawCircle (const shape &shape)

Private Attributes

- std::unordered_map< const text *, sf::Text > textList
- std::unordered_map< const shape *, sf::RectangleShape > rectList
- std::unordered_map< const shape *, sf::CircleShape > circleList
- std::unordered_map< const sprite *, sf::Sprite > spriteList_
- std::unordered_map< int, sf::Font > fonts_
- sf::RenderWindow window_
- sf::Texture texture_
- std::string textPath_
- bool isLoadedTexture_ = true

4.13.1 Constructor & Destructor Documentation

4.13.1.1 SFMLGraphic()

```
SFMLGraphic::SFMLGraphic ( )  [default]
```

4.13.1.2 ~SFMLGraphic()

```
SFMLGraphic::~SFMLGraphic ( )  [default], [noexcept]
```

4.13.2 Member Function Documentation

4.13.2.1 clearWindow()

```
void SFMLGraphic::clearWindow ( )  [override], [virtual]
```

Implements IGraphic.

4.13.2.2 createWindow()

```
void SFMLGraphic::createWindow (
            std::string title,
            int width,
            int height )  [override], [virtual]
```

Implements IGraphic.

4.13.2.3 destroyWindow()

```
void SFMLGraphic::destroyWindow ( )  [override], [virtual]
```

Implements IGraphic.

4.13.2.4 displayShape()

```
void SFMLGraphic::displayShape (
            const shape & shape )  [override], [virtual]
```

Implements IGraphic.

4.13.2.5 displaySprite()

```
void SFMLGraphic::displaySprite (
            const sprite & sprite )  [override], [virtual]
```

Implements IGraphic.

4.13.2.6 displayText()

```
void SFMLGraphic::displayText (
            const text & text )  [override], [virtual]
```

Implements IGraphic.

4.13.2.7 displayWindow()

```
void SFMLGraphic::displayWindow ( )  [override], [virtual]
```

Implements IGraphic.

4.13.2.8 drawCircle()

```
void SFMLGraphic::drawCircle (
            const shape & shape )  [private]
```

4.13.2.9 drawRectangle()

```
void SFMLGraphic::drawRectangle (
            const shape & shape )  [private]
```

4.13.2.10 getEvent()

```
eventKey SFMLGraphic::getEvent ( )  [override], [virtual]
```

Implements IGraphic.

4.13.2.11 isOpenWindow()

```
bool SFMLGraphic::isOpenWindow ( )  [override], [virtual]
```

Implements IGraphic.

4.13.3 Field Documentation

4.13.3.1 circleList

```
std::unordered_map<const shape*, sf::CircleShape> SFMLGraphic::circleList  [private]
```

4.13.3.2 fonts_

```
std::unordered_map<int, sf::Font> SFMLGraphic::fonts_  [private]
```

4.13.3.3 isLoadedTexture_

```
bool SFMLGraphic::isLoadedTexture_ = true  [private]
```

4.13.3.4 rectList

```
std::unordered_map<const shape*, sf::RectangleShape> SFMLGraphic::rectList  [private]
```

4.13.3.5 spriteList_

```
std::unordered_map<const sprite*, sf::Sprite> SFMLGraphic::spriteList_  [private]
```

4.13.3.6 textList

```
std::unordered_map<const text*, sf::Text> SFMLGraphic::textList  [private]
```

4.13.3.7 textPath_

```
std::string SFMLGraphic::textPath_  [private]
```

4.13.3.8 texture_

```
sf::Texture SFMLGraphic::texture_  [private]
```

4.13.3.9 window_

```
sf::RenderWindow SFMLGraphic::window_  [private]
```

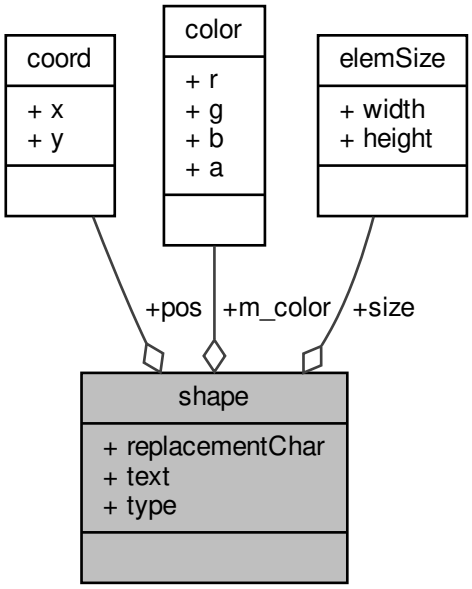
The documentation for this class was generated from the following files:

- include/graphics/sfml/SFMLGraphic.hpp
- src/graphics/sfml/SFMLEvent.cpp
- src/graphics/sfml/SFMLGraphic.cpp

4.14 shape Struct Reference

```
#include <IGraphic.hpp>
```

Collaboration diagram for shape:



Data Fields

- coord pos
- elemSize size
- color m_color
- char replacementChar
- std::string text
- shapeType type

4.14.1 Field Documentation

4.14.1.1 m_color

```
color shape::m_color
```

4.14.1.2 pos

```
coord shape::pos
```

4.14.1.3 replacementChar

```
char shape::replacementChar
```

4.14.1.4 size

```
elemSize shape::size
```

4.14.1.5 text

```
std::string shape::text
```

4.14.1.6 type

```
shapeType shape::type
```

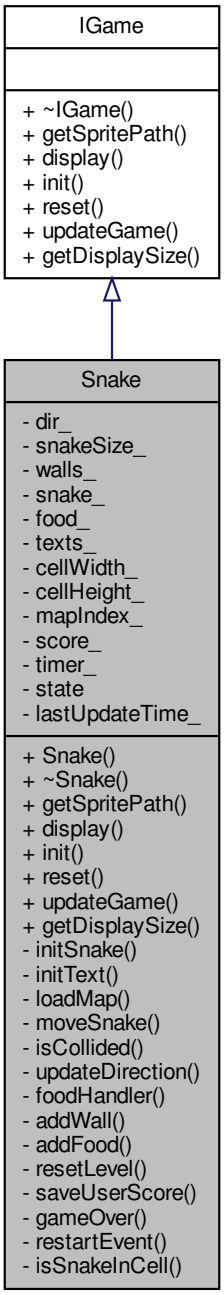
The documentation for this struct was generated from the following file:

- include/graphics/IGraphic.hpp

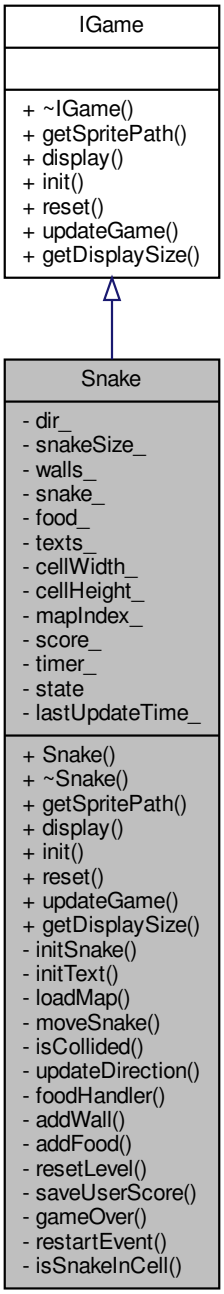
4.15 Snake Class Reference

`#include <Snake.hpp>`

Inheritance diagram for Snake:



Collaboration diagram for Snake:



Public Member Functions

- Snake ()=default
- ~Snake ()=default
- std::string getSpritePath () override
- void display (IGraphic &) override
- int init () override

- void reset () override
- int updateGame (eventKey evtKey) override
- elemSize getDisplaySize () override

Private Member Functions

- void initSnake (int x, int y)
- void initText ()
- void loadMap (int index)
- void moveSnake (std::vector< shape > &tmp)
- bool isCollided (const shape &s1, const shape &s2)
- void updateDirection (eventKey evtKey)
- void foodHandler ()
- void addWall (int x, int y)
- void addFood (int x, int y)
- void resetLevel ()
- void saveUserScore ()
- void gameOver ()
- void restartEvent (eventKey evtKey)
- bool isSnakeInCell (int x, int y)

Private Attributes

- direction dir_ { direction::RIGHT }
- int snakeSize_ { 4 }
- std::vector< shape > walls_
- std::vector< shape > snake_
- std::vector< shape > food_
- std::vector< text > texts_
- int cellWidth_ { 20 }
- int cellHeight_ { 20 }
- int mapIndex_ { 0 }
- int score_ { 0 }
- int timer_ { 0 }
- playerState state { playerState::ALIVE }
- std::chrono::steady_clock::time_point lastUpdateTime_

4.15.1 Constructor & Destructor Documentation

4.15.1.1 Snake()

```
Snake::Snake ( )  [default]
```

4.15.1.2 ~Snake()

```
Snake::~Snake ( )  [default]
```

4.15.2 Member Function Documentation

4.15.2.1 addFood()

```
void Snake::addFood (
            int x,
            int y )  [private]
```

4.15.2.2 addWall()

```
void Snake::addWall (
            int x,
            int y )  [private]
```

4.15.2.3 display()

```
void Snake::display (
            IGraphic & graphLib )  [override], [virtual]
```

Implements IGame.

4.15.2.4 foodHandler()

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

4.15.2.5 gameOver()

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

4.15.2.6 getDisplaySize()

```
elemSize Snake::getDisplaySize ( )  [override], [virtual]
```

Implements IGame.

4.15.2.7 getSpritePath()

```
std::string Snake::getSpritePath ( )  [override], [virtual]
```

Implements IGame.

4.15.2.8 init()

```
int Snake::init ( )  [override], [virtual]
```

Implements IGame.

4.15.2.9 initSnake()

```
void Snake::initSnake (
            int x,
            int y )  [private]
```

4.15.2.10 initText()

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

4.15.2.11 isCollided()

```
bool Snake::isCollided (
            const shape & s1,
            const shape & s2 )  [private]
```

4.15.2.12 isSnakeInCell()

```
bool Snake::isSnakeInCell (
            int x,
            int y )  [private]
```

4.15.2.13 loadMap()

```
void Snake::loadMap (
            int index )  [private]
```

4.15.2.14 moveSnake()

```
void Snake::moveSnake (
            std::vector< shape > & tmp )  [private]
```

4.15.2.15 reset()

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

Implements IGame.

4.15.2.16 resetLevel()

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

4.15.2.17 restartEvent()

```
void Snake::restartEvent (
            eventKey evtKey )  [private]
```

4.15.2.18 saveUserScore()

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

4.15.2.19 updateDirection()

```
void Snake::updateDirection (
            eventKey evtKey )  [private]
```

4.15.2.20 updateGame()

```
int Snake::updateGame (
            eventKey evtKey )  [override], [virtual]
```

Implements IGame.

4.15.3 Field Documentation

4.15.3.1 cellHeight_

```
int Snake::cellHeight_ { 20 }  [private]
```

4.15.3.2 cellWidth_

```
int Snake::cellWidth_ { 20 }  [private]
```

4.15.3.3 dir_

```
direction Snake::dir_ { direction::RIGHT }  [private]
```

4.15.3.4 food_

```
std::vector<shape> Snake::food_  [private]
```

4.15.3.5 lastUpdateTime_

```
std::chrono::steady_clock::time_point Snake::lastUpdateTime_  [private]
```

4.15.3.6 mapIndex_

```
int Snake::mapIndex_ { 0 }  [private]
```

4.15.3.7 score_

```
int Snake::score_ { 0 }  [private]
```

4.15.3.8 snake_

```
std::vector<shape> Snake::snake_  [private]
```

4.15.3.9 snakeSize_

```
int Snake::snakeSize_ { 4 }  [private]
```

4.15.3.10 state

```
playerState Snake::state { playerState::ALIVE }  [private]
```

4.15.3.11 texts_

```
std::vector<text> Snake::texts_  [private]
```

4.15.3.12 timer_

```
int Snake::timer_ { 0 }  [private]
```

4.15.3.13 walls_

```
std::vector<shape> Snake::walls_  [private]
```

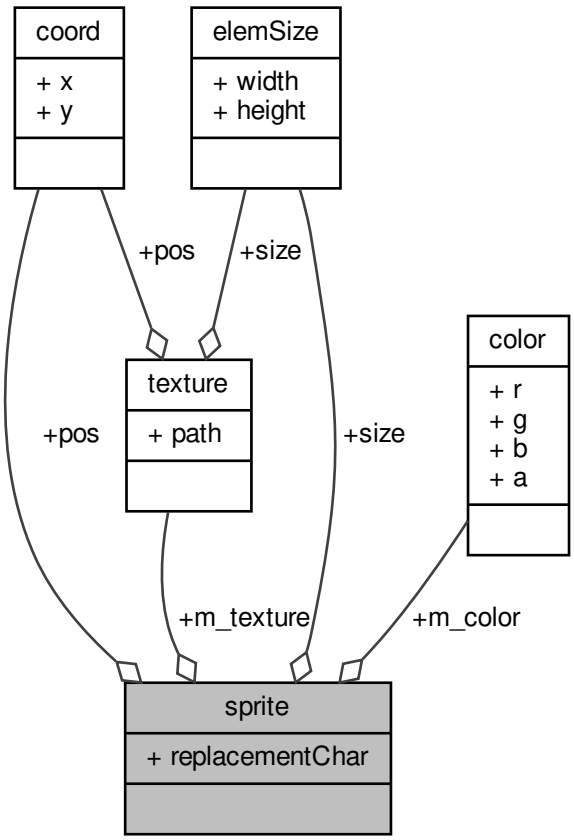
The documentation for this class was generated from the following files:

- include/games/snake/Snake.hpp
- src/games/snake/Compute.cpp
- src/games/snake/Display.cpp
- src/games/snake/Snake.cpp

4.16 sprite Struct Reference

```
#include <IGraphic.hpp>
```

Collaboration diagram for sprite:



Data Fields

- coord pos
- elemSize size
- texture m_texture
- char replacementChar
- color m_color

4.16.1 Field Documentation

4.16.1.1 m_color

```
color sprite::m_color
```

4.16.1.2 m_texture

```
texture sprite::m_texture
```

4.16.1.3 pos

```
coord sprite::pos
```

4.16.1.4 replacementChar

```
char sprite::replacementChar
```

4.16.1.5 size

```
elemSize sprite::size
```

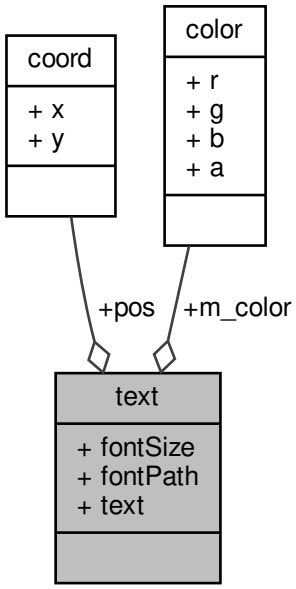
The documentation for this struct was generated from the following file:

- include/graphics/IGraphic.hpp

4.17 text Struct Reference

`#include <IGraphic.hpp>`

Collaboration diagram for text:



Data Fields

- coord pos
- int fontSize
- std::string fontPath
- color m_color
- std::string text

4.17.1 Field Documentation

4.17.1.1 fontPath

`std::string text::fontPath`

4.17.1.2 fontSize

```
int text::fontSize
```

4.17.1.3 m_color

```
color text::m_color
```

4.17.1.4 pos

```
coord text::pos
```

4.17.1.5 text

```
std::string text::text
```

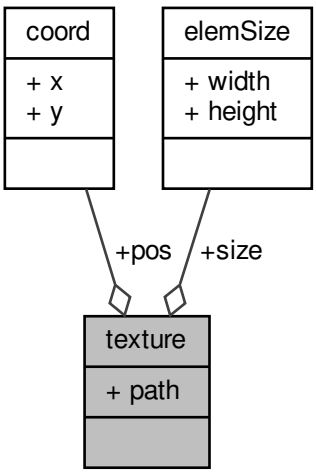
The documentation for this struct was generated from the following file:

- include/graphics/IGraphic.hpp

4.18 texture Struct Reference

```
#include <IGraphic.hpp>
```

Collaboration diagram for texture:



Data Fields

- coord pos
- elemSize size
- std::string path

4.18.1 Field Documentation

4.18.1.1 path

```
std::string texture::path
```

4.18.1.2 pos

```
coord texture::pos
```

4.18.1.3 size

```
elemSize texture::size
```

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

- include/graphics/IGraphic.hpp

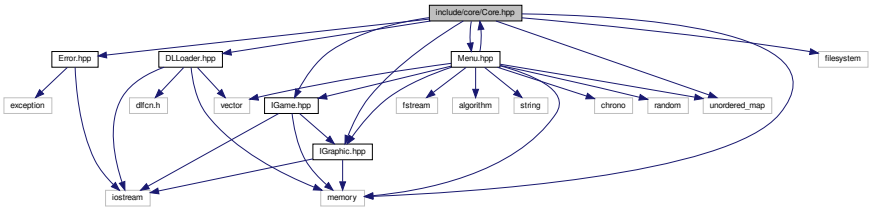
Chapter 5

File Documentation

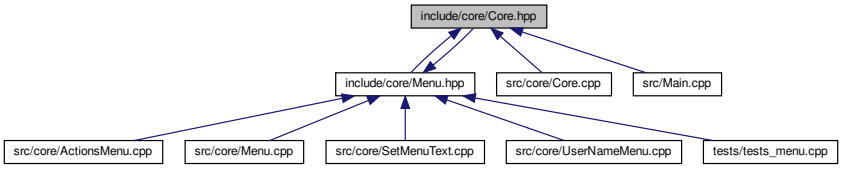
5.1 include/core/Core.hpp File Reference

```
#include "DLLoader.hpp"
#include "Error.hpp"
#include "IGame.hpp"
#include "IGraphic.hpp"
#include "Menu.hpp"
#include <filesystem>
#include <memory>
#include <unordered_map>
```

Include dependency graph for Core.hpp:



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



Data Structures

- class Core

Enumerations

- enum class GState {
  PLAY , PAUSE , MENU , QUIT ,
  CONTINUE }
- enum class libType { GRAPHICAL , GAME }

Variables

- static const std::unordered_map< std::string, libType > validLibs

5.1.1 Enumeration Type Documentation

5.1.1.1 GState

```
enum GState  [strong]
```

Enumerator

PLAY	
PAUSE	
MENU	
QUIT	
CONTINUE	

5.1.1.2 libType

```
enum libType  [strong]
```

Enumerator

GRAPHICAL	
GAME	

5.1.2 Variable Documentation

5.1.2.1 validLibs

```
const std::unordered_map<std::string, libType> validLibs  [static]
```

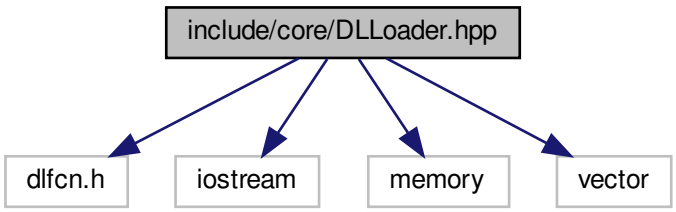
Initial value:

```
{
  {"arcade_ncurses.so", libType::GRAPHICAL},
  {"arcade_sdl2.so", libType::GRAPHICAL},
  {"arcade_ndk++.so", libType::GRAPHICAL},
  {"arcade_aalib.so", libType::GRAPHICAL},
  {"arcade_libcaca.so", libType::GRAPHICAL},
  {"arcade_allegro5.so", libType::GRAPHICAL},
  {"arcade_xlib.so", libType::GRAPHICAL},
  {"arcade_gtk+.so", libType::GRAPHICAL},
  {"arcade_sfml.so", libType::GRAPHICAL},
  {"arcade_irrlicht.so", libType::GRAPHICAL},
  {"arcade_opengl.so", libType::GRAPHICAL},
  {"arcade_vulkan.so", libType::GRAPHICAL},
  {"arcade_qt5.so", libType::GRAPHICAL},
  {"arcade_snake.so", libType::GAME},
  {"arcade_nibbler.so", libType::GAME},
  {"test_game.so", libType::GAME},
  {"arcade_pacman.so", libType::GAME},
  {"arcade_qix.so", libType::GAME},
  {"arcade_centipede.so", libType::GAME},
  {"arcade_solarfox.so", libType::GAME}
}
```

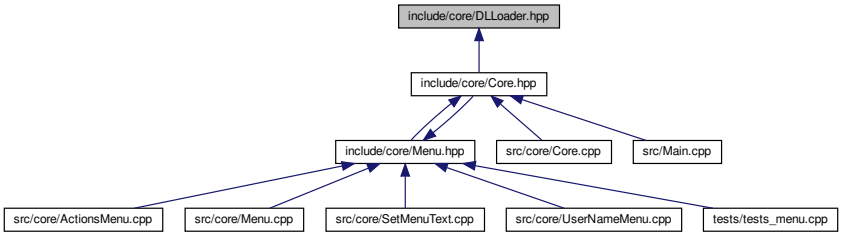
5.2 include/core/DLLoader.hpp File Reference

```
#include <dlfcn.h>
#include <iostream>
#include <memory>
#include <vector>
```

Include dependency graph for DLLoader.hpp:



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



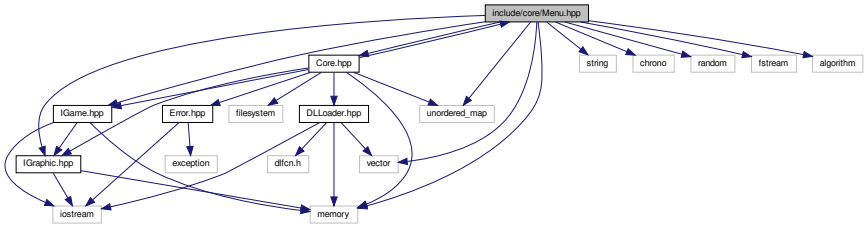
Data Structures

- class DLLoader< T >

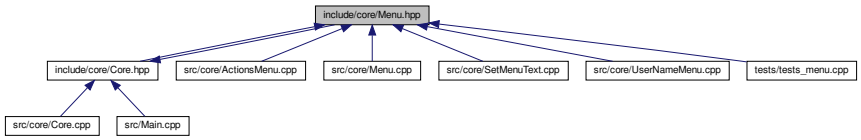
5.3 include/core/Menu.hpp File Reference

```
#include "IGraphic.hpp"
#include "IGame.hpp"
#include "Core.hpp"
#include <string>
#include <vector>
#include <memory>
#include <chrono>
#include <random>
#include <fstream>
#include <algorithm>
#include <unordered_map>
```

Include dependency graph for Menu.hpp:



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

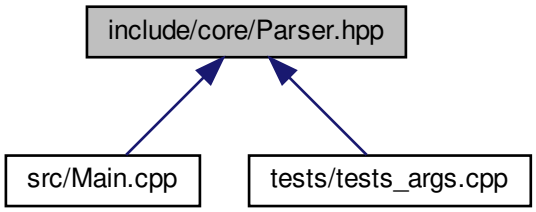


Data Structures

- class Menu

5.4 include/core/Parser.hpp File Reference

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



Functions

- int checkArgs (int ac, char ∗∗av)
- int checkEnv (char ∗∗env)
- void displayUsage ()
- int checkLibrary (const char ∗library)

5.4.1 Function Documentation

5.4.1.1 checkArgs()

```
int checkArgs (
            int ac,
            char ** av )
```

5.4.1.2 checkEnv()

```
int checkEnv (
            char ** env )
```

5.4.1.3 checkLibrary()

```
int checkLibrary (
            const char * library )
```

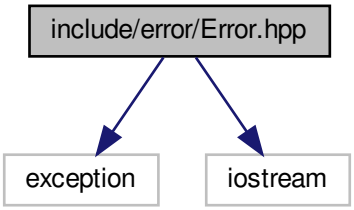
5.4.1.4 displayUsage()

```
void displayUsage ( )
```

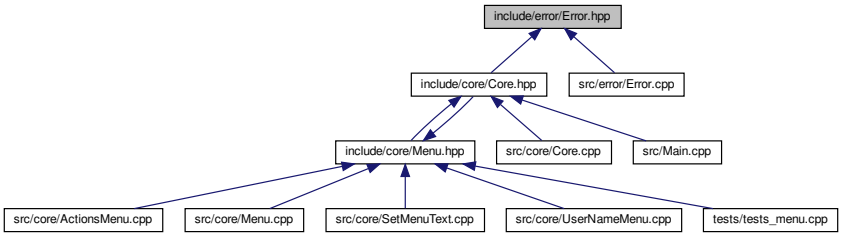
5.5 include/error/Error.hpp File Reference

```
#include <exception>
#include <iostream>
```

Include dependency graph for Error.hpp:



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



Data Structures

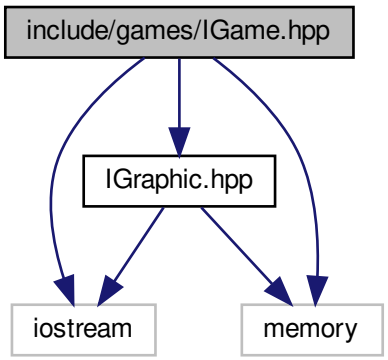
- class ArcadeError

5.6 include/games/IGame.hpp File Reference

```
#include <iostream>
#include <memory>
```

```
#include "IGraphic.hpp"
```

Include dependency graph for IGame.hpp:



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

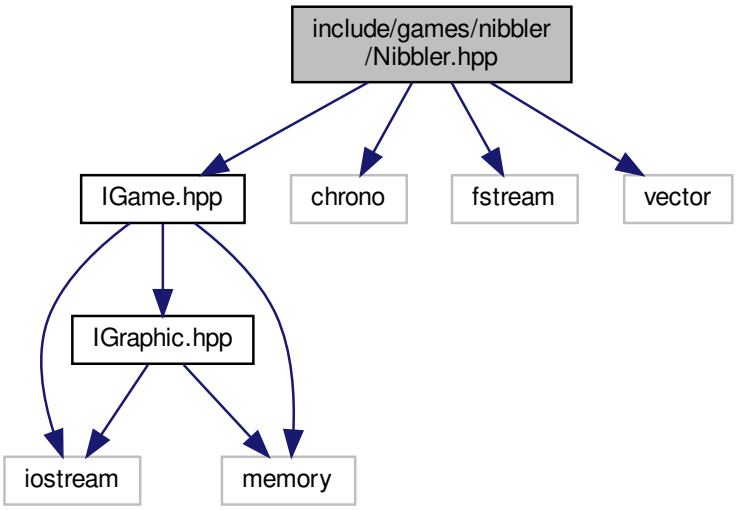
Data Structures

- class IGame

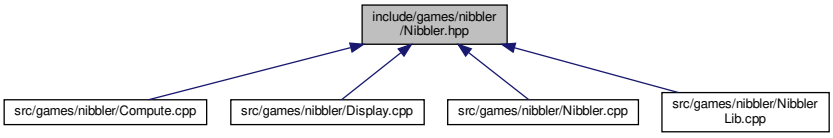
5.7 include/games/nibbler/Nibbler.hpp File Reference

```
#include "IGame.hpp"
#include <chrono>
#include <fstream>
#include <vector>
```

Include dependency graph for Nibbler.hpp:



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



Data Structures

- class Nibbler

Enumerations

- enum class direction {
  UP , DOWN , LEFT , RIGHT ,
  UP , DOWN , LEFT , RIGHT }
- enum class playerState {
  ALIVE , DEAD , STOP , WON ,
  ALIVE , DEAD , STOP }

Variables

- const std::vector< std::vector< std::string > > allMaps

5.7.1 Enumeration Type Documentation

5.7.1.1 direction

```
enum direction  [strong]
```

Enumerator

UP	
DOWN	
LEFT	
RIGHT	
UP	
DOWN	
LEFT	
RIGHT	

5.7.1.2 playerState

```
enum playerState  [strong]
```

Enumerator

ALIVE	
DEAD	
STOP	
WON	
ALIVE	
DEAD	
STOP	

5.7.2 Variable Documentation

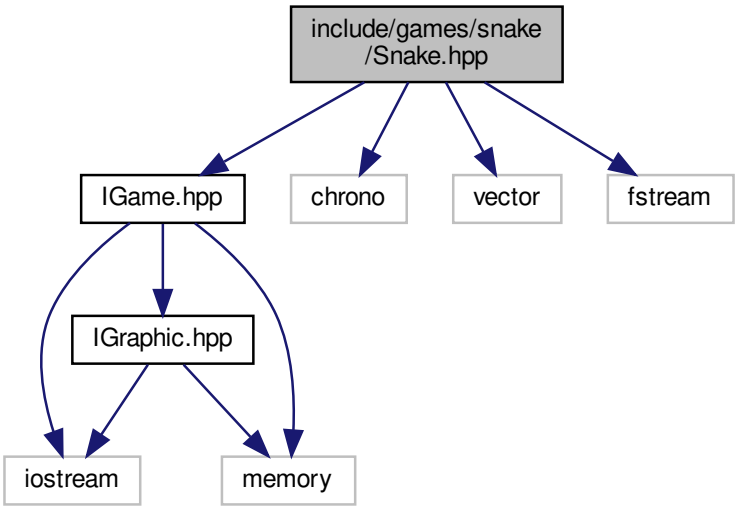
5.7.2.1 allMaps

```
const std::vector<std::vector<std::string> > allMaps
```

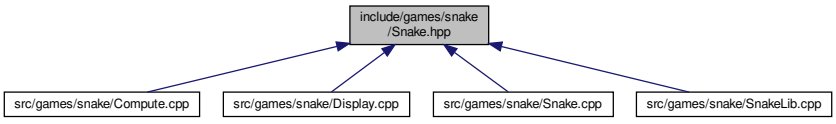
5.8 include/games/snake/Snake.hpp File Reference

```
#include "IGame.hpp"
#include <chrono>
#include <vector>
#include <fstream>
```

Include dependency graph for Snake.hpp:



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



Data Structures

- class Snake

Enumerations

- enum class direction {
  UP , DOWN , LEFT , RIGHT ,
  UP , DOWN , LEFT , RIGHT }
- enum class playerState {
  ALIVE , DEAD , STOP , WON ,
  ALIVE , DEAD , STOP }

Variables

- const std::vector< std::vector< std::string > > allMaps

5.8.1 Enumeration Type Documentation

5.8.1.1 direction

```
enum direction  [strong]
```

Enumerator

UP	
DOWN	
LEFT	
RIGHT	
UP	
DOWN	
LEFT	
RIGHT	

5.8.1.2 playerState

```
enum playerState  [strong]
```

Enumerator

ALIVE	
DEAD	
STOP	
WON	
ALIVE	
DEAD	
STOP	

5.8.2 Variable Documentation

5.8.2.1 allMaps

```
const std::vector<std::vector<std::string> > allMaps
```

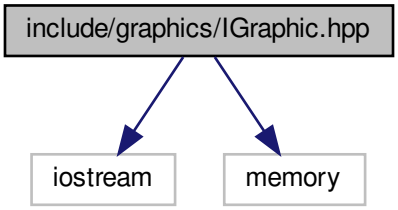
Initial value:

```
= {
    {
        "1111111111111111111",
        "1000000000000000001",
        "1000000000000000001",
        "1002000000000000001",
        "1000000000000000001",
        "1000000000000000001",
        "1000000000000000001",
        "1000000000000000001",
        "1000000000000000001",
        "1000000003000000001",
        "1000000000000000001",
        "1000000000000000001",
        "1000000000000000001",
        "1000000000000000001",
        "1000000000000000001",
        "1000000000000000001",
        "1000000000000000001",
        "1000000000000000001",
        "1111111111111111111",
    },
}
```

5.9 include/graphics/IGraphic.hpp File Reference

```
#include <iostream>
#include <memory>
```

Include dependency graph for IGraphic.hpp:



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

Data Structures

- struct coord
- struct elemSize
- struct texture
- struct color
- struct sprite
- struct text
- struct shape
- class IGraphic

Enumerations

- enum class eventKey {
  NULL_EVENT , A , Z , E ,
  R , T , Y , U ,
  I , O , P , Q ,
  S , D , F , G ,
  H , J , K , L ,
  M , W , X , C ,
  V , B , N , SPACE ,
  ESCAPE , ENTER , LARROW , RARROW ,
  BARROW , UARROW , DELETE , SUPR ,
  TAB , ONE , TWO , THREE ,
  FOUR , FIVE , SIX , SEVEN ,
  EIGHT , NINE , ZERO , CTRL ,
  ALT , SHIFT , QUIT }
- enum class shapeType { RECTANGLE , CIRCLE , TRIANGLE }

5.9.1 Enumeration Type Documentation

5.9.1.1 eventKey

```
enum eventKey  [strong]
```

Enumerator

NULL_EVENT	
A	
Z	
E	
R	
T	
Y	
U	
I	
O	
P	
Q	
S	
D	
F	
G	
H	
J	
K	
L	
M	
W	
X	
C	

Enumerator

V	
B	
N	
SPACE	
ESCAPE	
ENTER	
LARROW	
RARROW	
BARROW	
UARROW	
DELETE	
SUPR	
TAB	
ONE	
TWO	
THREE	
FOUR	
FIVE	
SIX	
SEVEN	
EIGHT	
NINE	
ZERO	
CTRL	
ALT	
SHIFT	
QUIT	

5.9.1.2 shapeType

```
enum shapeType  [strong]
```

Enumerator

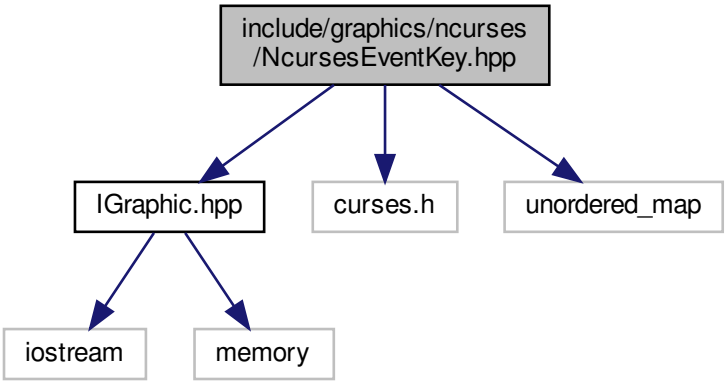
RECTANGLE	
CIRCLE	
TRIANGLE	

5.10 include/graphics/ncurses/NcursesEventKey.hpp File Reference

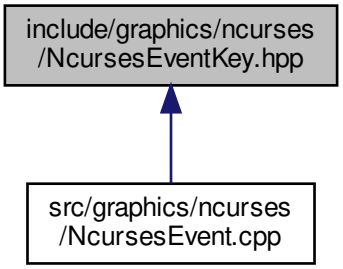
```
#include "IGraphic.hpp"
#include <curses.h>
```

```
#include <unordered_map>
```

Include dependency graph for NcursesEventKey.hpp:



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



Variables

- std::unordered_map< char, eventKey > keyEvent

5.10.1 Variable Documentation

5.10.1.1 keyEvent

```
std::unordered_map<char, eventKey> keyEvent
```

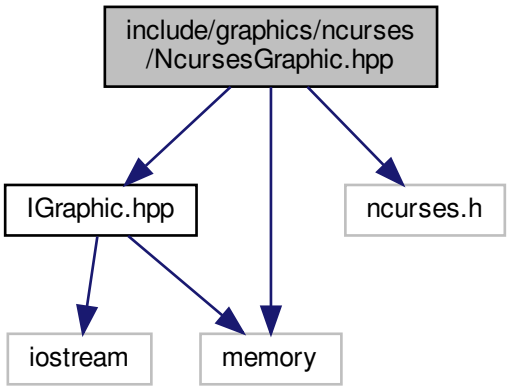
Initial value:

```
= { { 'a', eventKey::A },
    { 'b', eventKey::B }, { 'c', eventKey::C }, { 'd', eventKey::D },
    { 'e', eventKey::E }, { 'f', eventKey::F }, { 'g', eventKey::G },
    { 'h', eventKey::H }, { 'i', eventKey::I }, { 'j', eventKey::J },
    { 'k', eventKey::K }, { 'l', eventKey::L }, { 'm', eventKey::M },
    { 'n', eventKey::N }, { 'o', eventKey::O }, { 'p', eventKey::P },
    { 'q', eventKey::Q }, { 'r', eventKey::R }, { 's', eventKey::S },
    { 't', eventKey::T }, { 'u', eventKey::U }, { 'v', eventKey::V },
    { 'w', eventKey::W }, { 'x', eventKey::X }, { 'y', eventKey::Y },
    { 'z', eventKey::Z }, { KEY_F0, eventKey::SPACE }, { '\n', eventKey::ENTER },
    { KEY_LEFT, eventKey::LARROW }, { KEY_RIGHT, eventKey::RARROW }, { KEY_DOWN, eventKey::BARROW },
    { KEY_UP, eventKey::UARROW }, { KEY_BACKSPACE, eventKey::DELETE }, { KEY_STAB, eventKey::TAB } }
```

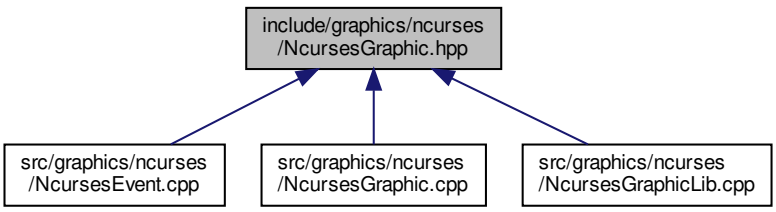
5.11 include/graphics/ncurses/NcursesGraphic.hpp File Reference

```
#include "IGraphic.hpp"
#include <memory>
#include <ncurses.h>
```

Include dependency graph for NcursesGraphic.hpp:



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



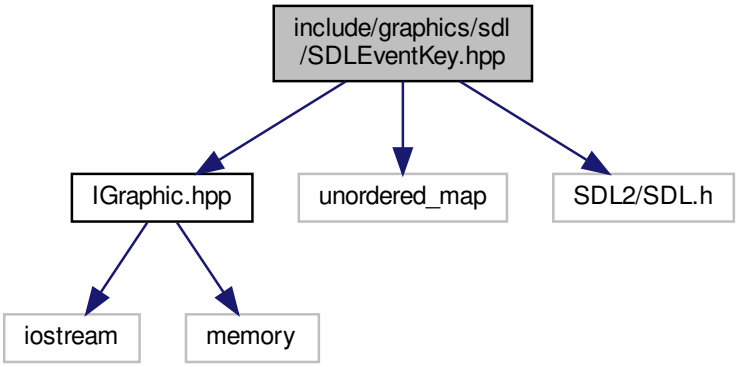
Data Structures

- class NcursesGraphic

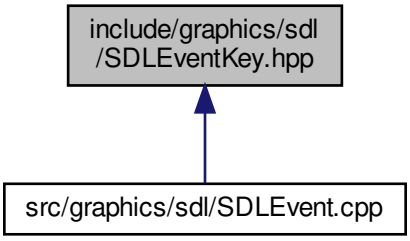
5.12 include/graphics/sdl/SDLEventKey.hpp File Reference

```
#include "IGraphic.hpp"
#include <unordered_map>
#include <SDL2/SDL.h>
```

Include dependency graph for SDLEventKey.hpp:



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



Variables

- std::unordered_map< char, eventKey > keyEvent

5.12.1 Variable Documentation

5.12.1.1 keyEvent

```
std::unordered_map<char, eventKey> keyEvent
```

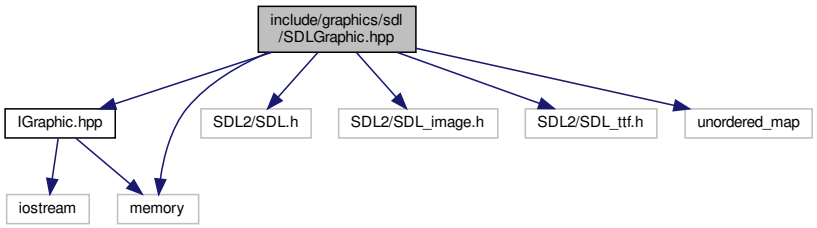
Initial value:

```
= { { 'a', eventKey::A },
    { 'b', eventKey::B }, { 'c', eventKey::C }, { 'd', eventKey::D },
    { 'e', eventKey::E }, { 'f', eventKey::F }, { 'g', eventKey::G },
    { 'h', eventKey::H }, { 'i', eventKey::I }, { 'j', eventKey::J },
    { 'k', eventKey::K }, { 'l', eventKey::L }, { 'm', eventKey::M },
    { 'n', eventKey::N }, { 'o', eventKey::O }, { 'p', eventKey::P },
    { 'q', eventKey::Q }, { 'r', eventKey::R }, { 's', eventKey::S },
    { 't', eventKey::T }, { 'u', eventKey::U }, { 'v', eventKey::V },
    { 'w', eventKey::W }, { 'x', eventKey::X }, { 'y', eventKey::Y },
    { 'z', eventKey::Z }, { SDLK_SPACE, eventKey::SPACE },
    { SDLK_ESCAPE, eventKey::ESCAPE }, { SDLK_RETURN, eventKey::ENTER },
    { SDLK_LEFT, eventKey::LARROW }, { SDLK_RIGHT, eventKey::RARROW },
    { SDLK_DOWN, eventKey::BARROW }, { SDLK_UP, eventKey::UARROW },
    { SDLK_BACKSPACE, eventKey::DELETE }, { SDLK_DELETE, eventKey::SUPR },
    { SDLK_KP_TAB, eventKey::TAB }, { SDLK_1, eventKey::ONE },
    {SDLK_2, eventKey::TWO }, {SDLK_3, eventKey::THREE },
    {SDLK_4, eventKey::FOUR }, {SDLK_5, eventKey::FIVE },
    {SDLK_6, eventKey::SIX }, {SDLK_7, eventKey::SEVEN },
    {SDLK_8, eventKey::EIGHT }, {SDLK_9, eventKey::NINE },
    {SDLK_0, eventKey::ZERO }, { SDLK_LCTRL, eventKey::CTRL },
    { SDLK_LALT, eventKey::ALT }, { SDLK_LSHIFT, eventKey::SHIFT } }
```

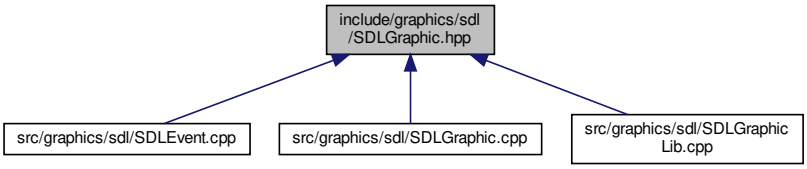
5.13 include/graphics/sdl/SDLGraphic.hpp File Reference

```
#include "IGraphic.hpp"
#include <SDL2/SDL.h>
#include <SDL2/SDL_image.h>
#include <SDL2/SDL_ttf.h>
#include <memory>
#include <unordered_map>
```

Include dependency graph for SDLGraphic.hpp:



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



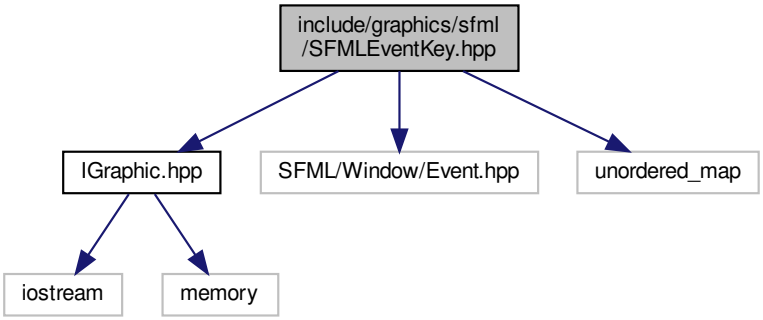
Data Structures

- class SDLGraphic

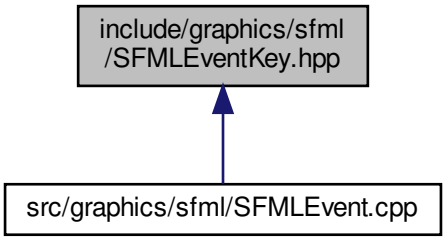
5.14 include/graphics/sfml/SFMLEventKey.hpp File Reference

```
#include "IGraphic.hpp"
#include <SFML/Window/Event.hpp>
#include <unordered_map>
```

Include dependency graph for SFMLEventKey.hpp:



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



Variables

- std::unordered_map< sf::Keyboard::Key, eventKey > keyEvent

5.14.1 Variable Documentation

5.14.1.1 keyEvent

```
std::unordered_map<sf::Keyboard::Key, eventKey> keyEvent
```

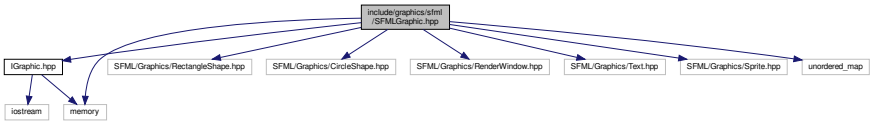
Initial value:

```
= { { sf::Keyboard::Key::A, eventKey::A },
    { sf::Keyboard::Key::B, eventKey::B }, { sf::Keyboard::Key::C, eventKey::C }, { sf::Keyboard::Key::D,
      eventKey::D },
    { sf::Keyboard::Key::E, eventKey::E }, { sf::Keyboard::Key::F, eventKey::F }, { sf::Keyboard::Key::G,
      eventKey::G },
    { sf::Keyboard::Key::H, eventKey::H }, { sf::Keyboard::Key::I, eventKey::I }, { sf::Keyboard::Key::J,
      eventKey::J },
    { sf::Keyboard::Key::K, eventKey::K }, { sf::Keyboard::Key::L, eventKey::L }, { sf::Keyboard::Key::M,
      eventKey::M },
    { sf::Keyboard::Key::N, eventKey::N }, { sf::Keyboard::Key::O, eventKey::O }, { sf::Keyboard::Key::P,
      eventKey::P },
    { sf::Keyboard::Key::Q, eventKey::Q }, { sf::Keyboard::Key::R, eventKey::R }, { sf::Keyboard::Key::S,
      eventKey::S },
    { sf::Keyboard::Key::T, eventKey::T }, { sf::Keyboard::Key::U, eventKey::U }, { sf::Keyboard::Key::V,
      eventKey::V },
    { sf::Keyboard::Key::W, eventKey::W }, { sf::Keyboard::Key::X, eventKey::X }, { sf::Keyboard::Key::Y,
      eventKey::Y },
    { sf::Keyboard::Key::Z, eventKey::Z }, { sf::Keyboard::Key::Space, eventKey::SPACE },
    { sf::Keyboard::Key::Escape, eventKey::ESCAPE }, { sf::Keyboard::Key::Enter, eventKey::ENTER },
    { sf::Keyboard::Key::Left, eventKey::LARROW }, { sf::Keyboard::Key::Right, eventKey::RARROW },
    { sf::Keyboard::Key::Down, eventKey::BARROW }, { sf::Keyboard::Key::Up, eventKey::UARROW },
    { sf::Keyboard::Key::BackSpace, eventKey::DELETE }, { sf::Keyboard::Key::Delete, eventKey::SUPR },
    { sf::Keyboard::Key::Tab, eventKey::TAB }, { sf::Keyboard::Key::Num1, eventKey::ONE },
    { sf::Keyboard::Key::Num2, eventKey::TWO }, { sf::Keyboard::Key::Num3, eventKey::THREE },
    { sf::Keyboard::Key::Num4, eventKey::FOUR }, { sf::Keyboard::Key::Num5, eventKey::FIVE },
    { sf::Keyboard::Key::Num6, eventKey::SIX }, { sf::Keyboard::Key::Num7, eventKey::SEVEN },
    { sf::Keyboard::Key::Num8, eventKey::EIGHT }, { sf::Keyboard::Key::Num9, eventKey::NINE },
    { sf::Keyboard::Key::Num0, eventKey::ZERO }, { sf::Keyboard::Key::LControl, eventKey::CTRL },
    { sf::Keyboard::Key::LAlt, eventKey::ALT }, { sf::Keyboard::Key::LShift, eventKey::SHIFT } }
```

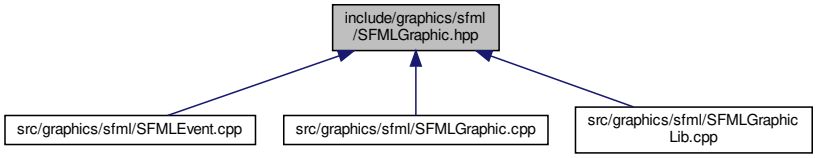
5.15 include/graphics/sfml/SFMLGraphic.hpp File Reference

```
#include "IGraphic.hpp"
#include <SFML/Graphics/RectangleShape.hpp>
#include <SFML/Graphics/CircleShape.hpp>
#include <SFML/Graphics/RenderWindow.hpp>
#include <SFML/Graphics/Text.hpp>
#include <SFML/Graphics/Sprite.hpp>
#include <memory>
#include <unordered_map>
```

Include dependency graph for SFMLGraphic.hpp:



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



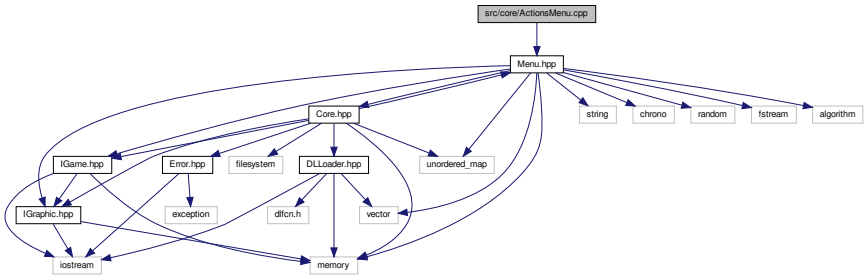
Data Structures

- class SFMLGraphic

5.16 src/core/ActionsMenu.cpp File Reference

```
#include "Menu.hpp"
```

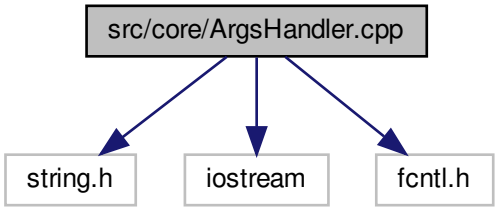
Include dependency graph for ActionsMenu.cpp:



5.17 src/core/ArgsHandler.cpp File Reference

```
#include <string.h>
#include <iostream>
#include <fcntl.h>
```

Include dependency graph for ArgsHandler.cpp:



Functions

- int checkLibrary (const char *library)
- void displayUsage ()
- int checkArgs (int ac, char **av)
- int checkEnv (char **env)

5.17.1 Function Documentation

5.17.1.1 checkArgs()

```
int checkArgs (
            int ac,
            char ** av )
```

5.17.1.2 checkEnv()

```
int checkEnv (
            char ** env )
```

5.17.1.3 checkLibrary()

```
int checkLibrary (
            const char * library )
```

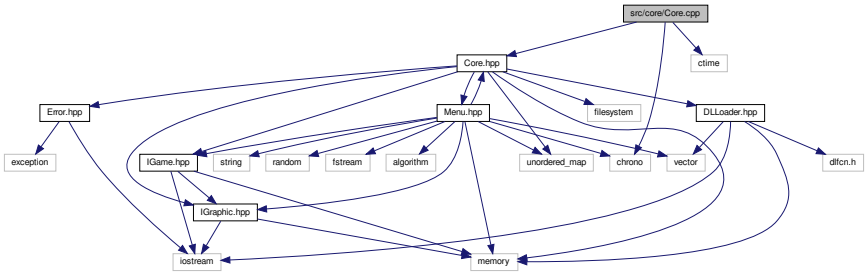
5.17.1.4 displayUsage()

```
void displayUsage ( )
```

5.18 src/core/Core.cpp File Reference

```
#include "Core.hpp"
#include <chrono>
#include <ctime>
```

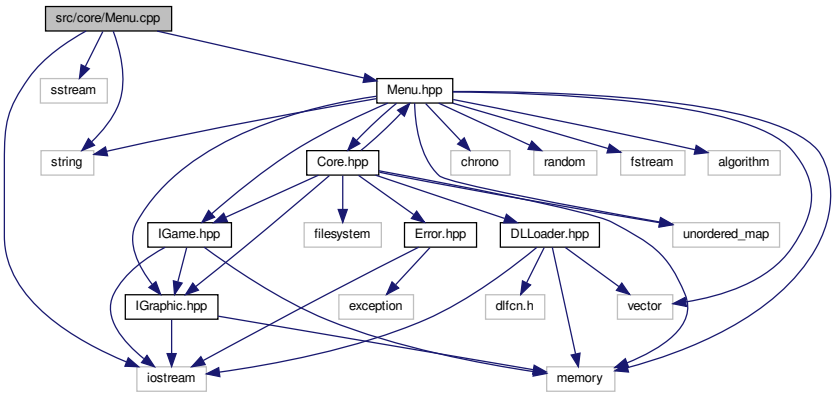
Include dependency graph for Core.cpp:



5.19 src/core/Menu.cpp File Reference

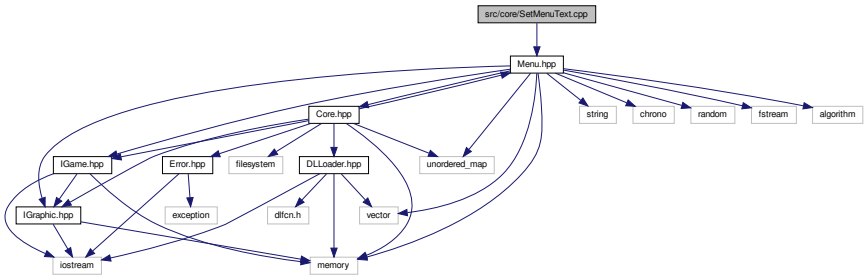
```
#include <iostream>
#include <sstream>
#include <string>
#include "Menu.hpp"
```

Include dependency graph for Menu.cpp:



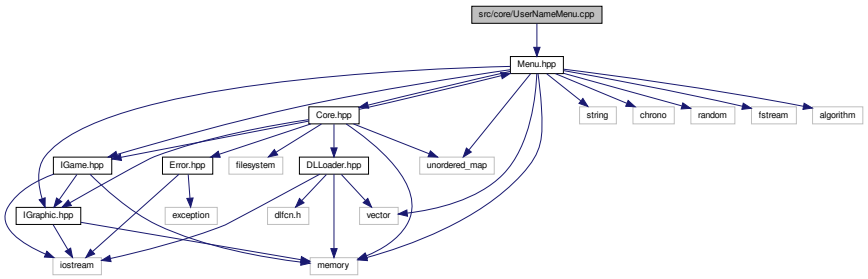
5.20 src/core/SetMenuText.cpp File Reference

```
#include "Menu.hpp"
```
Include dependency graph for SetMenuText.cpp:



5.21 src/core/UserNameMenu.cpp File Reference

```
#include "Menu.hpp"
```
Include dependency graph for UserNameMenu.cpp:



Functions

- int random (int low, int high)

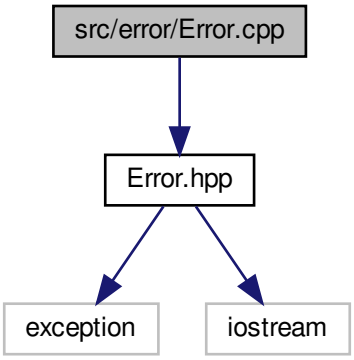
5.21.1 Function Documentation

5.21.1.1 random()

```
int random (
            int low,
            int high )
```

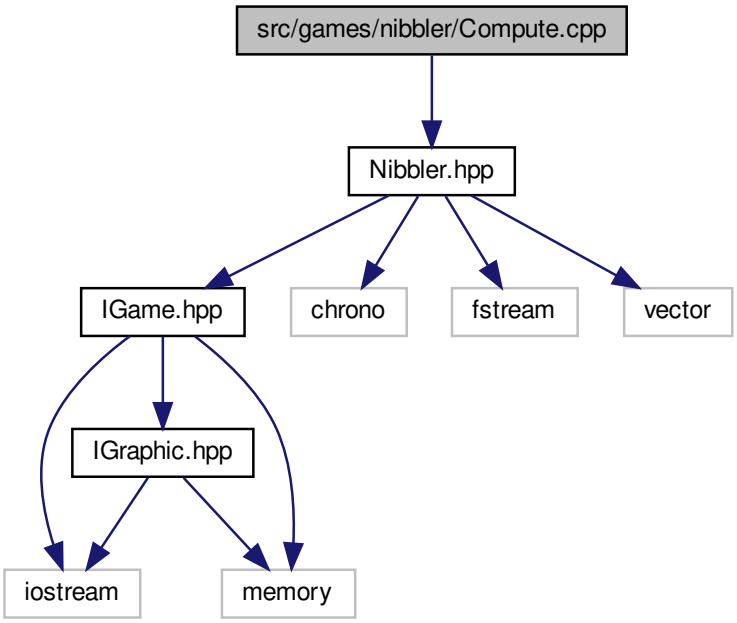
5.22 src/error/Error.cpp File Reference

`#include "Error.hpp"`
Include dependency graph for Error.cpp:



5.23 src/games/nibbler/Compute.cpp File Reference

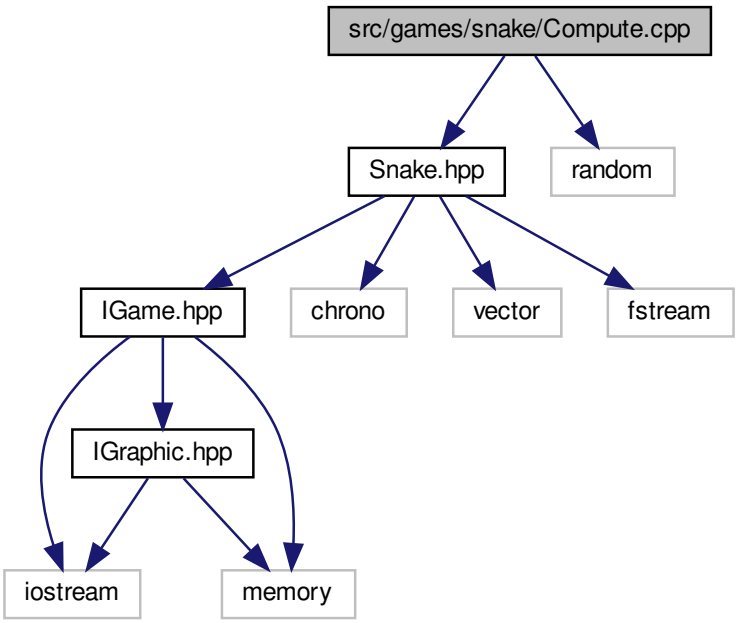
`#include "Nibbler.hpp"`
Include dependency graph for Compute.cpp:



5.24 src/games/snake/Compute.cpp File Reference

```
#include "Snake.hpp"
#include <random>
```

Include dependency graph for Compute.cpp:



Functions

- int random (int low, int high)

5.24.1 Function Documentation

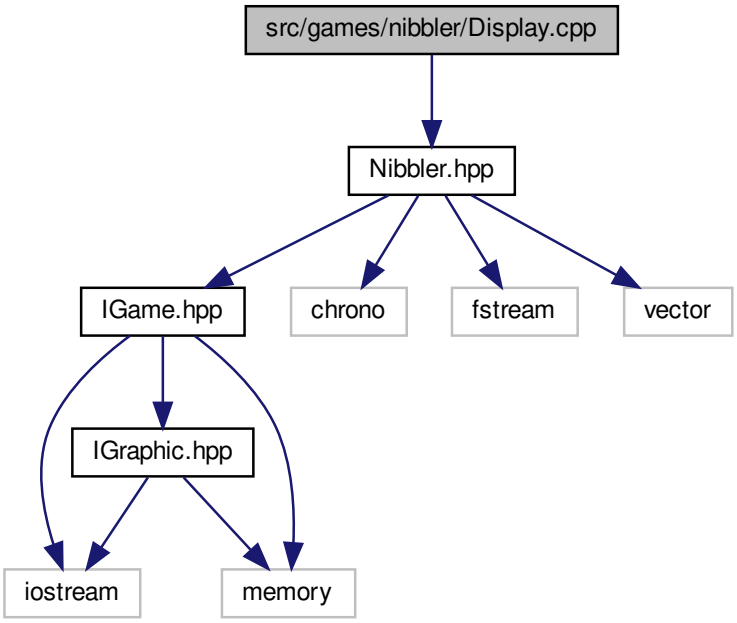
5.24.1.1 random()

```
int random (
            int low,
            int high )
```

5.25 src/games/nibbler/Display.cpp File Reference

```
#include "Nibbler.hpp"
```

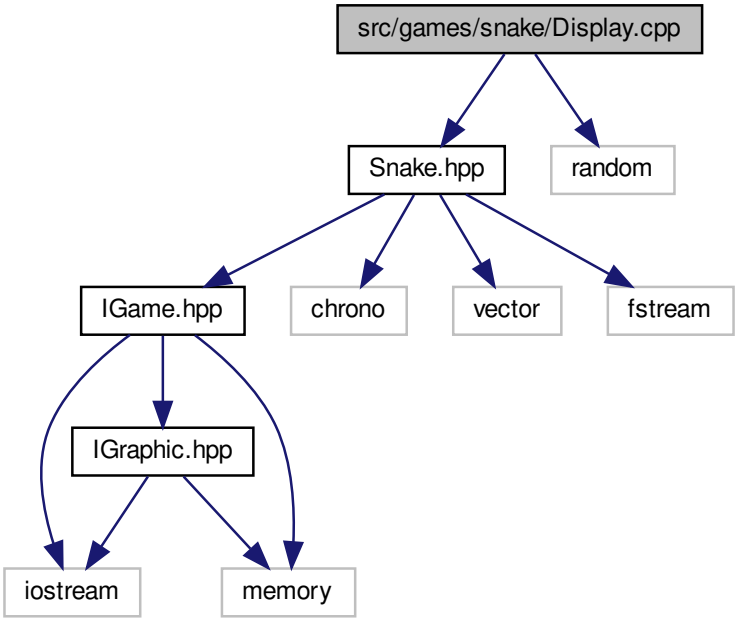
Include dependency graph for Display.cpp:



5.26 src/games/snake/Display.cpp File Reference

```
#include "Snake.hpp"
#include <random>
```

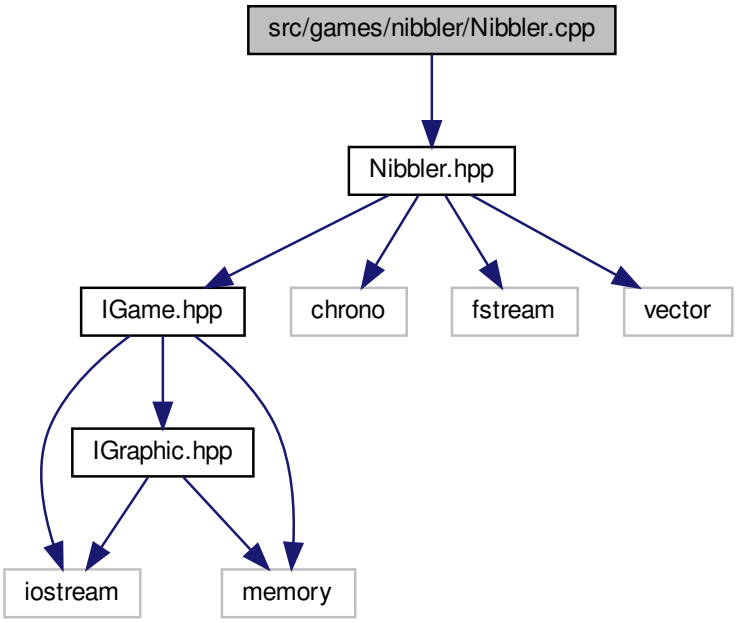
Include dependency graph for Display.cpp:



5.27 src/games/nibbler/Nibbler.cpp File Reference

```
#include "Nibbler.hpp"
```

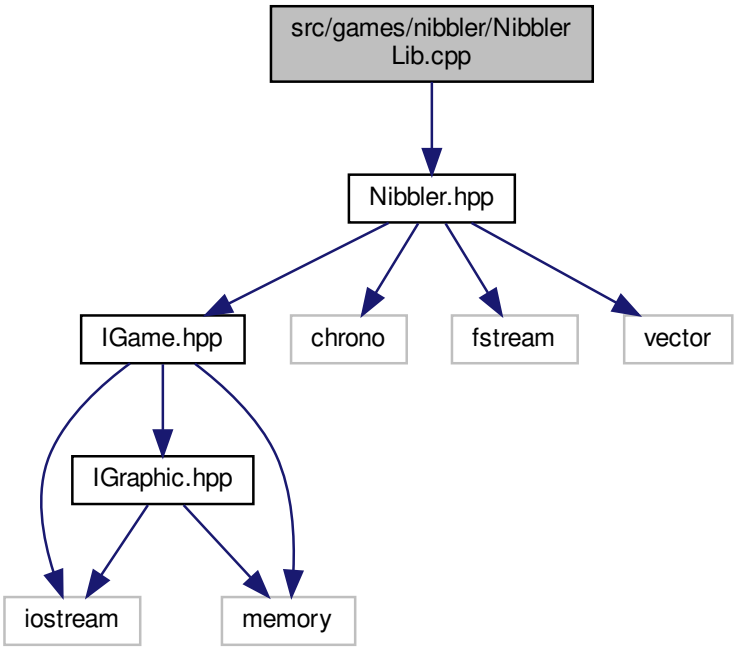
Include dependency graph for Nibbler.cpp:



5.28 src/games/nibbler/NibblerLib.cpp File Reference

```
#include "Nibbler.hpp"
```

Include dependency graph for NibblerLib.cpp:



Functions

- std::unique_ptr< IGame > entryPoint (void)

5.28.1 Function Documentation

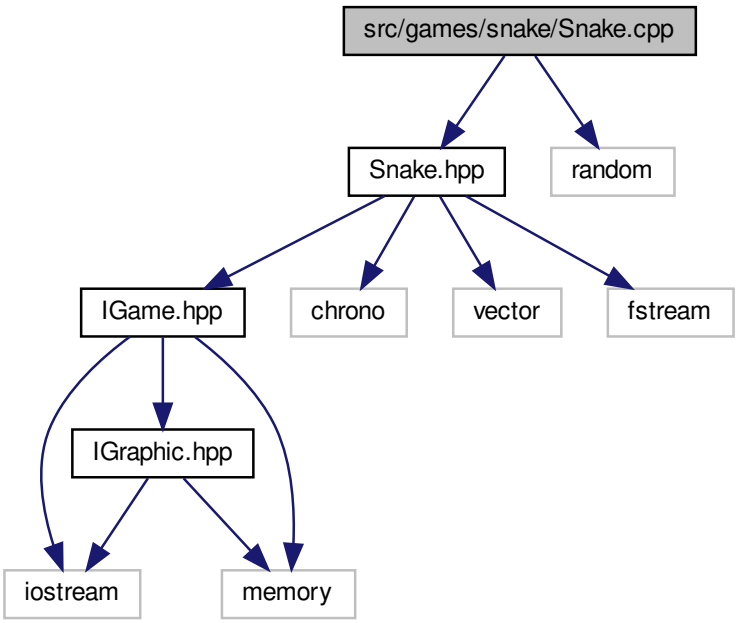
5.28.1.1 entryPoint()

```
std::unique_ptr<IGame> entryPoint (
            void  )
```

5.29 src/games/snake/Snake.cpp File Reference

```
#include "Snake.hpp"
#include <random>
```

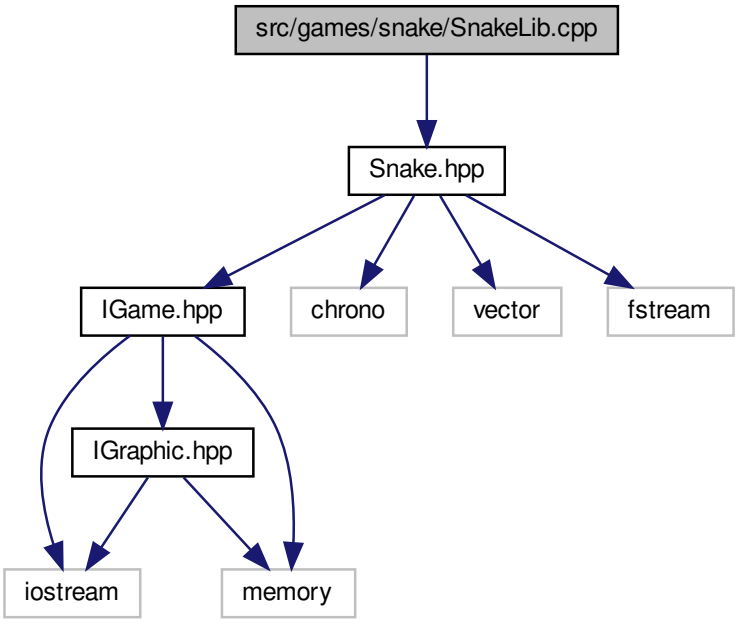
Include dependency graph for Snake.cpp:



5.30 src/games/snake/SnakeLib.cpp File Reference

```
#include "Snake.hpp"
```

Include dependency graph for SnakeLib.cpp:



Functions

- std::unique_ptr< IGame > entryPoint (void)

5.30.1 Function Documentation

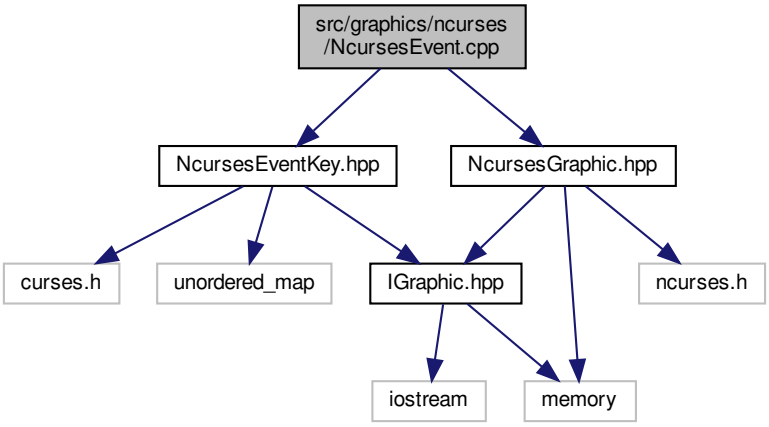
5.30.1.1 entryPoint()

```
std::unique_ptr<IGame> entryPoint (
            void  )
```

5.31 src/graphics/ncurses/NcursesEvent.cpp File Reference

```
#include "NcursesEventKey.hpp"
#include "NcursesGraphic.hpp"
```

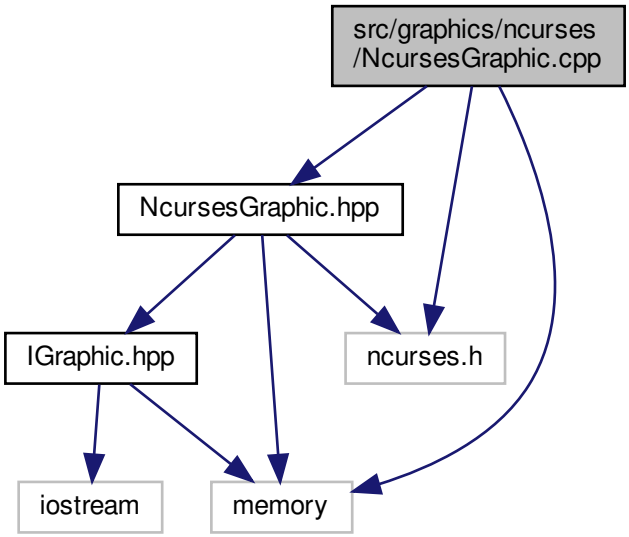
Include dependency graph for NcursesEvent.cpp:



5.32 src/graphics/ncurses/NcursesGraphic.cpp File Reference

```
#include "NcursesGraphic.hpp"
#include <memory>
#include <ncurses.h>
```

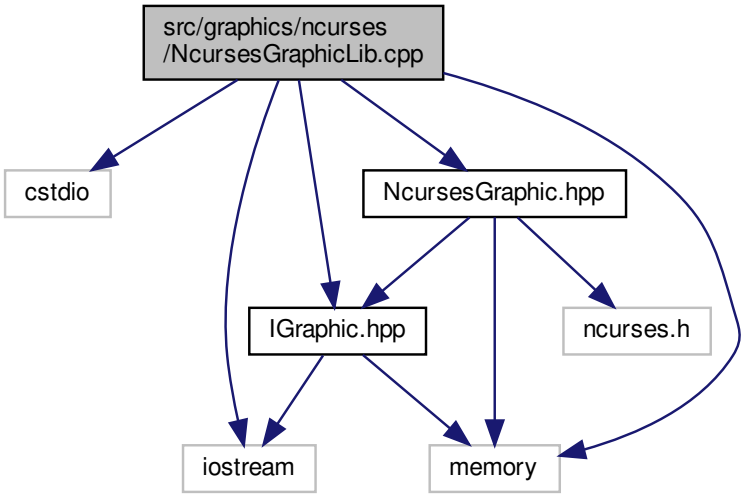
Include dependency graph for NcursesGraphic.cpp:



5.33 src/graphics/ncurses/NcursesGraphicLib.cpp File Reference

```
#include <cstdio>
#include <iostream>
#include <memory>
#include "IGraphic.hpp"
#include "NcursesGraphic.hpp"
```

Include dependency graph for NcursesGraphicLib.cpp:



Functions

- std::unique_ptr< IGraphic > entryPoint (void)

5.33.1 Function Documentation

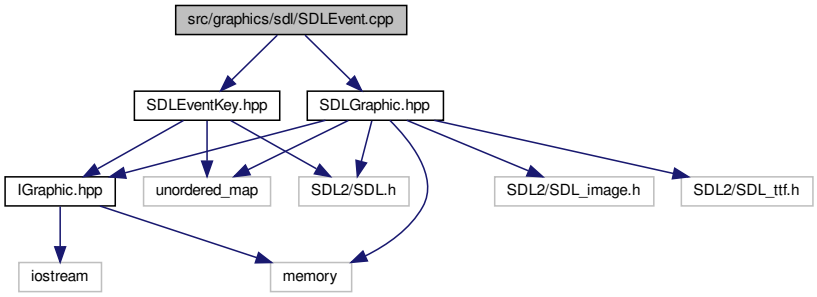
5.33.1.1 entryPoint()

```
std::unique_ptr<IGraphic> entryPoint (
            void  )
```

5.34 src/graphics/sdl/SDLEvent.cpp File Reference

```
#include "SDLEventKey.hpp"
#include "SDLGraphic.hpp"
```

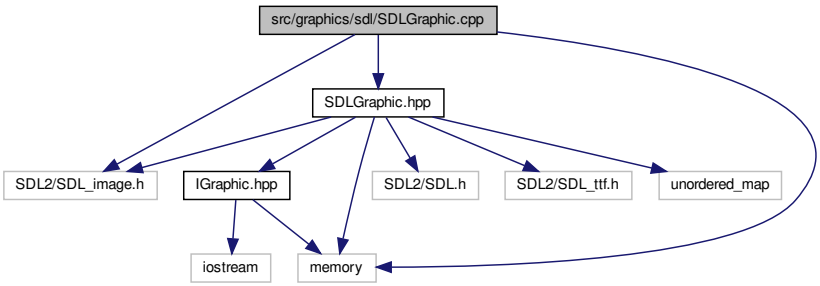
Include dependency graph for SDLEvent.cpp:



5.35 src/graphics/sdl/SDLGraphic.cpp File Reference

```
#include <SDL2/SDL_image.h>
#include <SDLGraphic.hpp>
#include <memory>
```

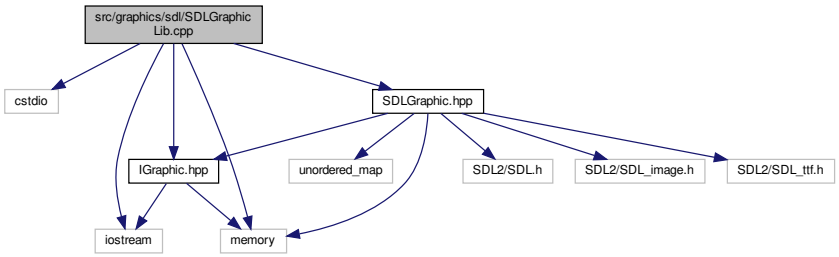
Include dependency graph for SDLGraphic.cpp:



5.36 src/graphics/sdl/SDLGraphicLib.cpp File Reference

```
#include <cstdio>
#include <iostream>
#include <memory>
#include "IGraphic.hpp"
#include "SDLGraphic.hpp"
```

Include dependency graph for SDLGraphicLib.cpp:



Functions

- std::unique_ptr< IGraphic > entryPoint (void)

5.36.1 Function Documentation

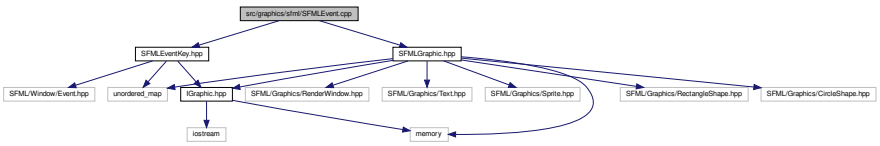
5.36.1.1 entryPoint()

```
std::unique_ptr<IGraphic> entryPoint (
            void  )
```

5.37 src/graphics/sfml/SFMLEvent.cpp File Reference

```
#include "SFMLEventKey.hpp"
#include "SFMLGraphic.hpp"
```

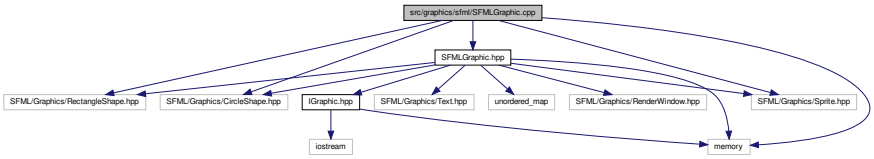
Include dependency graph for SFMLEvent.cpp:



5.38 src/graphics/sfml/SFMLGraphic.cpp File Reference

```
#include "SFMLGraphic.hpp"
#include <SFML/Graphics/CircleShape.hpp>
#include <SFML/Graphics/RectangleShape.hpp>
#include <SFML/Graphics/Sprite.hpp>
#include <memory>
```

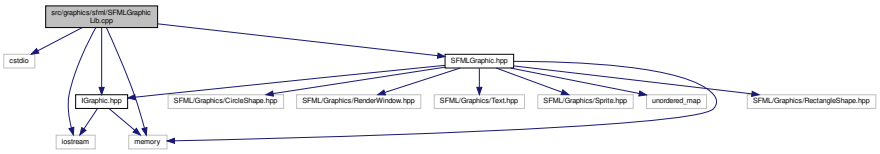
Include dependency graph for SFMLGraphic.cpp:



5.39 src/graphics/sfml/SFMLGraphicLib.cpp File Reference

```
#include <cstdio>
#include <iostream>
#include <memory>
#include "IGraphic.hpp"
#include "SFMLGraphic.hpp"
```

Include dependency graph for SFMLGraphicLib.cpp:



Functions

- std::unique_ptr< IGraphic > entryPoint (void)

5.39.1 Function Documentation

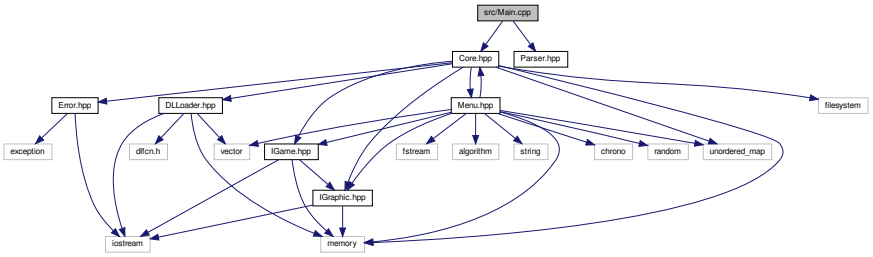
5.39.1.1 entryPoint()

```
std::unique_ptr<IGraphic> entryPoint (
            void  )
```

5.40 src/Main.cpp File Reference

```
#include "Core.hpp"
#include "Parser.hpp"
```

Include dependency graph for Main.cpp:



Functions

- int main (int ac, char **av, char **env)

5.40.1 Function Documentation

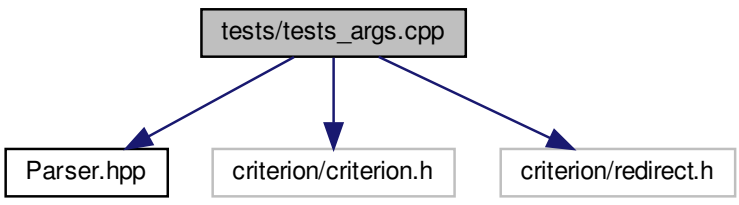
5.40.1.1 main()

```
int main (
            int ac,
            char ** av,
            char ** env )
```

5.41 tests/tests_args.cpp File Reference

```
#include "Parser.hpp"
#include <criterion/criterion.h>
#include <criterion/redirect.h>
```

Include dependency graph for tests_args.cpp:



Functions

- Test (checkArgs, no_args)
- Test (checkArgs, no_args2)
- Test (checkArgs, bad_args)
- Test (checkArgs, wrong_lib)
- Test (checkArgs, wrong_lib2)
- Test (checkArgs, one_arg)
- Test (checkArgs, two_args)
- Test (checkArgs, help)
- Test (checkEnv, no_env)

5.41.1 Function Documentation

5.41.1.1 Test() [1/9]

```
Test (
            checkArgs ,
            bad_args  )
```

5.41.1.2 Test() [2/9]

```
Test (
            checkArgs ,
            help  )
```

5.41.1.3 Test() [3/9]

```
Test (
            checkArgs ,
            no_args  )
```

5.41.1.4 Test() [4/9]

```
Test (
            checkArgs ,
            no_args2  )
```

5.41.1.5 Test() [5/9]

```
Test (
            checkArgs ,
            one_arg  )
```

5.41.1.6 Test() [6/9]

```
Test (
            checkArgs ,
            two_args  )
```

5.41.1.7 Test() [7/9]

```
Test (
            checkArgs ,
            wrong_lib  )
```

5.41.1.8 Test() [8/9]

```
Test (
            checkArgs ,
            wrong_lib2  )
```

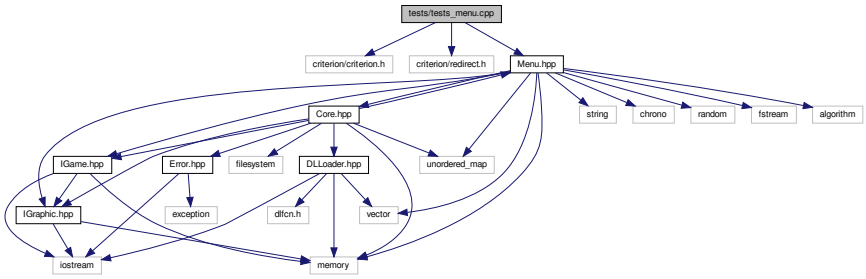
5.41.1.9 Test() [9/9]

```
Test (
            checkEnv ,
            no_env  )
```

5.42 tests/tests_menu.cpp File Reference

```
#include <criterion/criterion.h>
#include <criterion/redirect.h>
#include "Menu.hpp"
```

Include dependency graph for tests_menu.cpp:



Functions

- Test (MenuTest, coreStateNotModifiedWhenNotTyping)
- Test (MenuTest, deleteCharIsCalledOnDeleteKey)
- Test (MenuTest, coreStateIsSetToQuitOnQuitKey)
- Test (MenuTest, characterIsAddedToUserNameOnValidKey)

5.42.1 Function Documentation

5.42.1.1 Test() [1/4]

```
Test (
            MenuTest ,
            characterIsAddedToUserNameOnValidKey  )
```

5.42.1.2 Test() [2/4]

```
Test (
            MenuTest ,
            coreStateIsSetToQuitOnQuitKey  )
```

5.42.1.3 Test() [3/4]

```
Test (
            MenuTest ,
            coreStateNotModifiedWhenNotTyping  )
```

5.42.1.4 Test() [4/4]

```
Test (
            MenuTest ,
            deleteCharIsCalledOnDeleteKey  )
```

Index