KText Editor

Generated by Doxygen 1.9.6

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

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

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 Command Namespace Reference

A stack in the Command namespace.

Classes

- class Stack

5.1.1 Detailed Description

A stack in the Command namespace.

5.2 KEYS Namespace Reference

An enum for Keyboard characters in hex form.

Enumerations

- enum {
  ESCAPE = 0x1B , ENTER = 0xD , BS = 0x8 , Shift_A = 0x41 ,
  CTRL = 0x11 , DELETE = 0x7f }

5.2.1 Detailed Description

An enum for Keyboard characters in hex form.

5.2.2 Enumeration Type Documentation

5.2.2.1 anonymous enum

```
anonymous enum
```

Enumerator

ESCAPE	
ENTER	
BS	
Shift_A	
CTRL	
DELETE	

Chapter 6

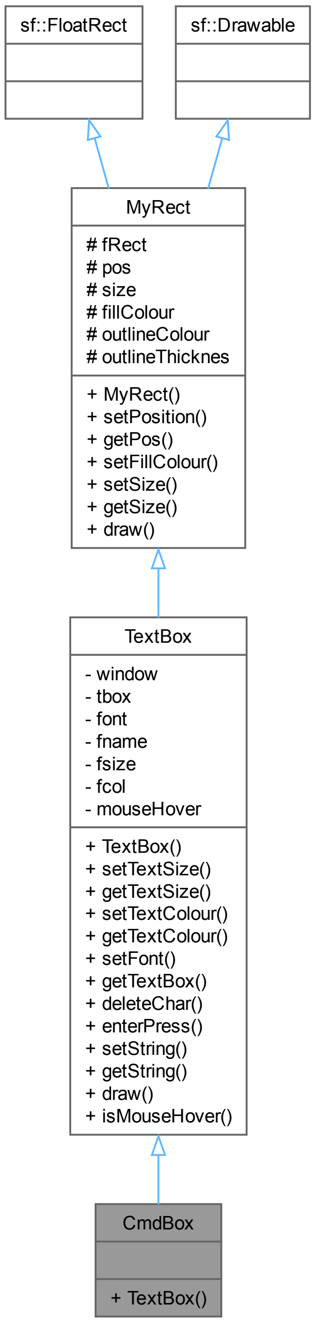
Class Documentation

6.1 CmdBox Class Reference

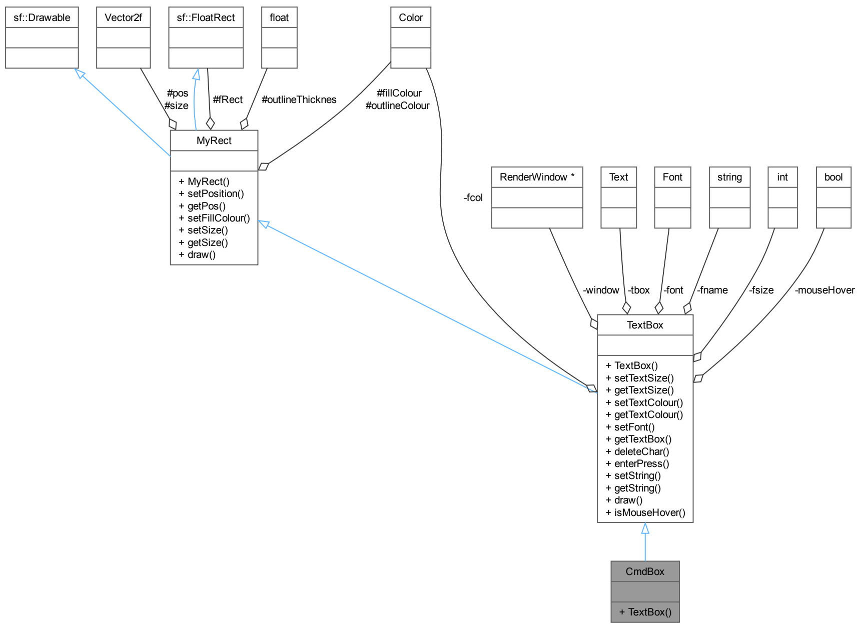
Class to handle the command TextBox.

```
#include <CmdBox.h>
```

Inheritance diagram for CmdBox:



Collaboration diagram for CmdBox:



Public Member Functions

- TextBox (sf::RenderWindow ∗win, sf::Vector2f pos, sf::Vector2f size, std::string sfont, int fsize, sf::Color fcol, sf::Color background, float thicc)

    Using teh Parent class constructor.

Public Member Functions inherited from TextBox

- TextBox (sf::RenderWindow ∗win, sf::Vector2f pos, sf::Vector2f size, std::string sfont, int fsize, sf::Color fcol, sf::Color background, float thicc)

    Constructor for TextBox.
- void setTextSize (int size)

    Set the size of the text.
- int getTextSize () const

    Get the size of the text.
- void setTextColour (sf::Color colour)

    Set the colour of the text.
- sf::Color getTextColour () const

    Get the colour of the text.
- void setFont (sf::Font &font)

    set what font you use
- sf::Text getTextBox () const

    Get both the main textbox and the cmd textbox in a vector.
- void deleteChar ()

    Delete last character entered.
- void enterPress ()

    Handles Enter key press.

- void setString (std::string nstring)

    Sets the string.
- std::string getString () const

    returns the text in tbox
- void draw (sf::RenderTarget &target, sf::RenderStates states) const override

    used to draw to the screen virutal method inherited from MyRect -> sf::Drawable thats overrided here is what allows us to draw to window using window.draw(TextBox)
- bool isMouseHover ()

    check if mouse is hovering over current textbox

Public Member Functions inherited from MyRect

- MyRect (sf::Vector2f pos, sf::Vector2f size, sf::Color fillColour, sf::Color outlineColour, float outlineThicknes)

    constructor for MyRect
- void setPosition (sf::Vector2f pos)

    sets the position of rect
- sf::Vector2f getPos () const

    get the position of rect
- void setFillColour (sf::Color colour)

    set the fill colour of the rect
- void setSize (sf::Vector2f size)

    set the size of the rect
- sf::Vector2f getSize () const

    get the size of the rect
- void draw (sf::RenderTarget &target, sf::RenderStates states) const override

    virutal method to draw to window

Additional Inherited Members

Protected Attributes inherited from MyRect

- sf::FloatRect fRect
- sf::Vector2f pos
- sf::Vector2f size
- sf::Color fillColour
- sf::Color outlineColour
- float outlineThicknes

6.1.1 Detailed Description

Class to handle the command TextBox.

6.1.2 Member Function Documentation

6.1.2.1 TextBox()

```
TextBox::TextBox (
             sf::RenderWindow * win,
             sf::Vector2f pos,
             sf::Vector2f size,
             std::string sfont,
             int fsize,
             sf::Color fcol,
             sf::Color background,
             float thicc )
```

Using teh Parent class constructor.

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

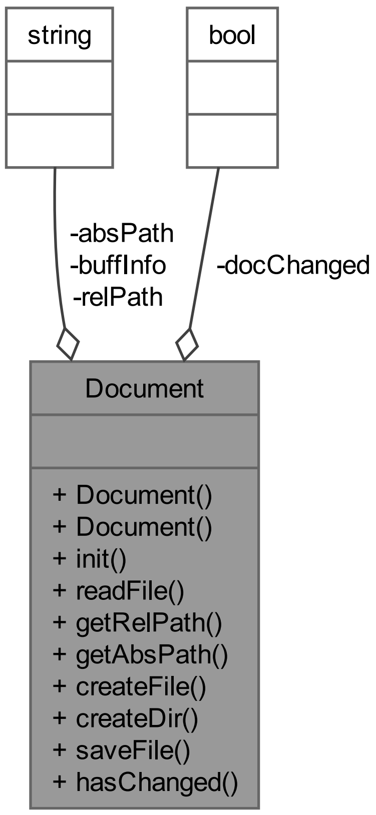
- include/Kamil/CmdBox.h

6.2 Document Class Reference

Document class.

```
#include <Document.h>
```

Collaboration diagram for Document:



Public Member Functions

- Document ()

  Constructor for Document class.
- Document (std::string fileP)

  Constructor for Document class.
- void init ()

  initialise the file TODO: overload this with file path
- void readFile ()

  read the file
- std::string getRelPath ()

  get the relative path
- std::string getAbsPath ()

  get the relative path
- void createFile ()

  create the file
- void createDir ()

  create a directory
- bool saveFile (const std::string &filename)

  save to a file
- bool hasChanged ()

  if the file has changed

Private Attributes

- std::string relPath
- std::string absPath
- std::string buffInfo
- bool docChanged

6.2.1 Detailed Description

Document class.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 Document() [1/2]

```
Document::Document ( )
```

Constructor for Document class.

6.2.2.2 Document() [2/2]

```
Document::Document (
            std::string fileP )
```

Constructor for Document class.

Parameters

fileP	- file path

6.2.3 Member Function Documentation

6.2.3.1 createDir()

```
void Document::createDir ( )
```

create a directory

6.2.3.2 createFile()

```
void Document::createFile ( )
```

create the file

6.2.3.3 getAbsPath()

```
std::string Document::getAbsPath ( )
```

get the relative path

6.2.3.4 getRelPath()

```
std::string Document::getRelPath ( )
```

get the relative path

6.2.3.5 hasChanged()

```
bool Document::hasChanged ( )
```

if the file has changed

Returns

bool - true if file has changed

6.2.3.6 init()

```
void Document::init ( )
```

initialise the file TODO: overload this with file path

6.2.3.7 readFile()

```
void Document::readFile ( )
```

read the file

6.2.3.8 saveFile()

```
bool Document::saveFile (
             const std::string & filename )
```

save to a file

6.2.4 Member Data Documentation

6.2.4.1 absPath

```
std::string Document::absPath  [private]
```

absolute path

6.2.4.2 buffInfo

```
std::string Document::buffInfo  [private]
```

6.2.4.3 docChanged

```
bool Document::docChanged  [private]
```

buffer information (the file text) if the file has changed

6.2.4.4 relPath

```
std::string Document::relPath  [private]
```

relative path

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

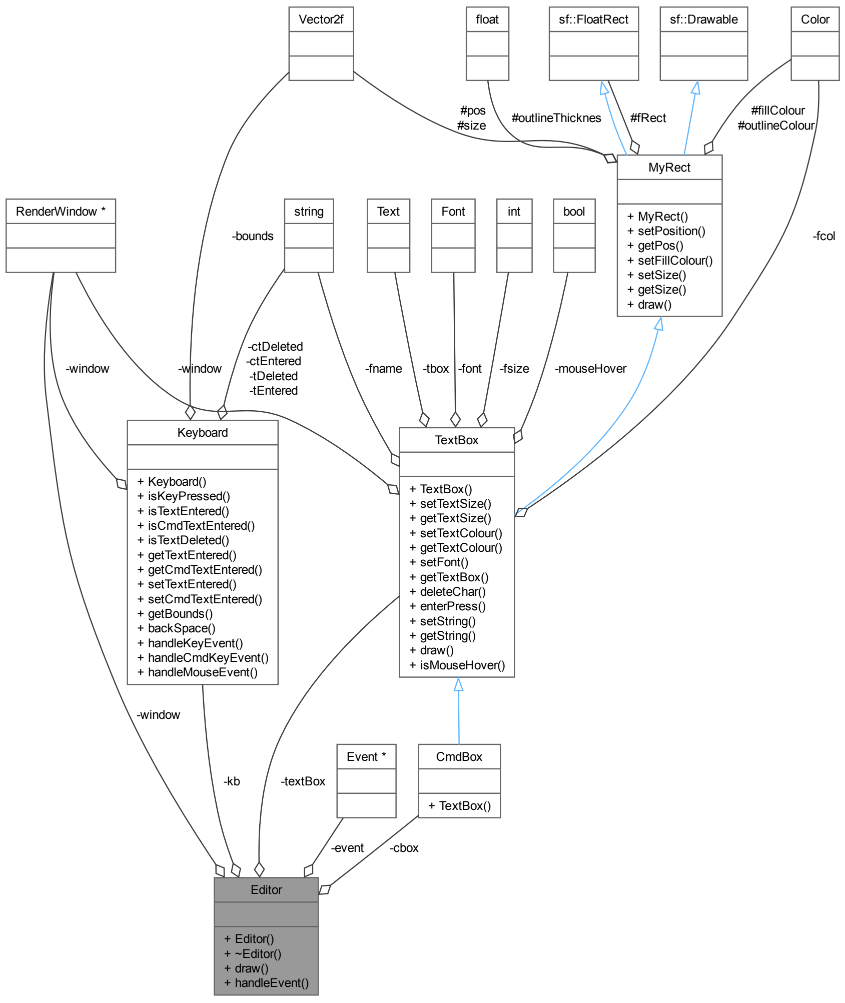
- include/Kamil/Document.h

6.3 Editor Class Reference

Class that handles and draws everything in the Editor.

```
#include <Editor.h>
```

Collaboration diagram for Editor:



Public Member Functions

- Editor (sf::RenderWindow ∗, sf::Event ∗)

    Constructor for Editor.
- ∼Editor ()

    Destructor for Editor class.
- void draw ()

    function that draws everything to RenderWindow
- void handleEvent ()

    handle the events for the Editor

Private Attributes

- TextBox ∗ textBox
- CmdBox ∗ cbox
- sf::RenderWindow ∗ window
- sf::Event ∗ event
- Keyboard kb

6.3.1 Detailed Description

Class that handles and draws everything in the Editor.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 Editor()

```
Editor::Editor (
            sf::RenderWindow * ,
            sf::Event *  )
```

Constructor for Editor.

Parameters

window	- reference to main RenderWindow
event	- reference to main event

6.3.2.2 ∼Editor()

```
Editor::~Editor ( )
```

Destructor for Editor class.

6.3.3 Member Function Documentation

6.3.3.1 draw()

```
void Editor::draw ( )
```

function that draws everything to RenderWindow

DEPRECATED

6.3.3.2 handleEvent()

```
void Editor::handleEvent ( )
```

handle the events for the Editor

where all event handles are called when interacting with other classes e.g. kb.handleEvent(); kb.handleMouse←Events();

6.3.4 Member Data Documentation

6.3.4.1 cbox

```
CmdBox* Editor::cbox [private]
```

reference to command box that we draw

6.3.4.2 event

```
sf::Event* Editor::event [private]
```

refernce to event

6.3.4.3 kb

```
Keyboard Editor::kb [private]
```

handles keyboard events

6.3.4.4 textBox

```
TextBox* Editor::textBox  [private]
```

reference to textbox that we draw

6.3.4.5 window

```
sf::RenderWindow* Editor::window  [private]
```

refernce to RenderWindow

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

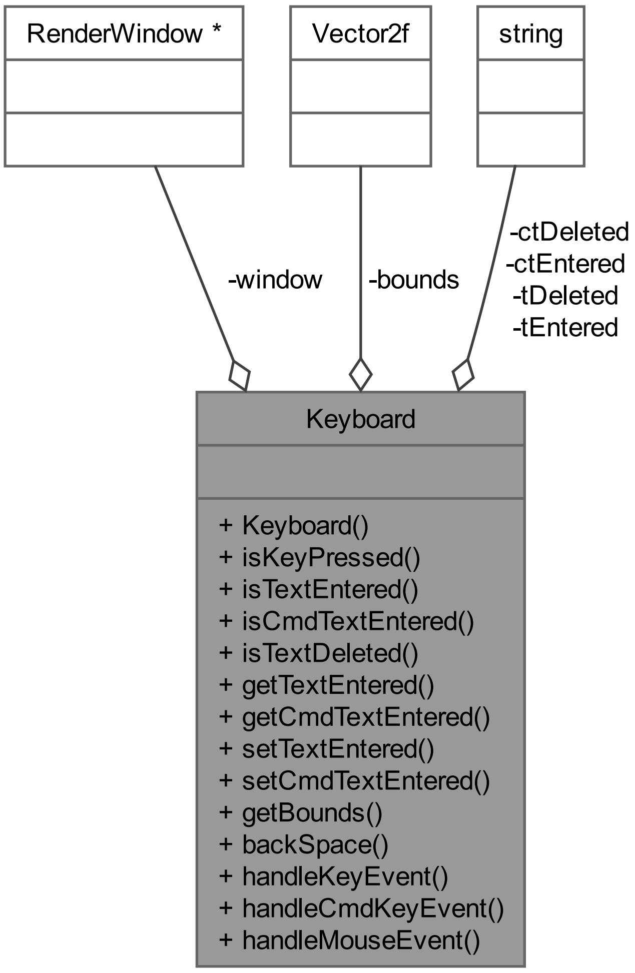
- include/Kamil/Editor.h

6.4 Keyboard Class Reference

A class to handle Keyboard input.

```
#include <Keyboard.h>
```

Collaboration diagram for Keyboard:



Public Member Functions

- Keyboard (sf::RenderWindow ∗win, sf::Vector2f bounds)

  Constructor for Keyboard class.
- bool isKeyPressed (sf::Keyboard::Key)

  checks if a key is pressed
- bool isTextEntered ()

  checks if a text is entered
- bool isCmdTextEntered ()

  checks if text is entered to the command box
- bool isTextDeleted ()

  check if text is being deleted
- std::string getTextEntered ()

  returns text entered

- std::string getCmdTextEntered ()

  returns text entered
- void setTextEntered (std::string)

  sets text
- void setCmdTextEntered (std::string)

  sets text
- sf::Vector2f getBounds () const

  get the bounds of the area we are in
- void backSpace ()

  when we backspace on teh text
- void handleKeyEvent (sf::Event &event)

  handle keyboard events
- void handleCmdKeyEvent ()

  handle keyboard events
- void handleMouseEvent (sf::Event &event)

  mouse keyboard events

Private Attributes

- sf::RenderWindow ∗ window
- sf::Vector2f bounds
- std::string tEntered
- std::string tDeleted
- std::string ctEntered
- std::string ctDeleted

6.4.1 Detailed Description

A class to handle Keyboard input.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 Keyboard()

```
Keyboard::Keyboard (
            sf::RenderWindow * win,
            sf::Vector2f bounds )
```

Constructor for Keyboard class.

Parameters

win	- reference to main window
bounds	- bounds of the window we are working in

6.4.3 Member Function Documentation

6.4.3.1 backSpace()

```
void Keyboard::backSpace ( )
```

when we backspace on teh text

6.4.3.2 getBounds()

```
sf::Vector2f Keyboard::getBounds ( ) const
```

get the bounds of the area we are in

Returns

sf::Vector2f bounded area

6.4.3.3 getCmdTextEntered()

```
std::string Keyboard::getCmdTextEntered ( )
```

returns text entered

Returns

std::string text entered

6.4.3.4 getTextEntered()

```
std::string Keyboard::getTextEntered ( )
```

returns text entered

Returns

std::string text entered

6.4.3.5 handleCmdKeyEvent()

```
void Keyboard::handleCmdKeyEvent ( )
```

handle keyboard events

Parameters

event	- to get text entered from events

6.4.3.6 handleKeyEvent()

```
void Keyboard::handleKeyEvent (
             sf::Event & event )
```

handle keyboard events

Parameters

event	- to get text entered from events

6.4.3.7 handleMouseEvent()

```
void Keyboard::handleMouseEvent (
             sf::Event & event )
```

mouse keyboard events

Parameters

event	- to get text entered from events

6.4.3.8 isCmdTextEntered()

```
bool Keyboard::isCmdTextEntered ( )
```

checks if text is entered to the command box

Returns

bool tru eif key is pressed false if not

6.4.3.9 isKeyPressed()

```
bool Keyboard::isKeyPressed (
             sf::Keyboard::Key  )
```

checks if a key is pressed

Returns

bool true if key is pressed false if not

6.4.3.10 isTextDeleted()

```
bool Keyboard::isTextDeleted ( )
```

check if text is being deleted

Returns

bool true if text is being deleted

6.4.3.11 isTextEntered()

```
bool Keyboard::isTextEntered ( )
```

checks if a text is entered

Returns

bool true if key is pressed false if not

6.4.3.12 setCmdTextEntered()

```
void Keyboard::setCmdTextEntered (
             std::string  )
```

sets text

Parameters

nstring	- new string

6.4.3.13 setTextEntered()

```
void Keyboard::setTextEntered (
             std::string  )
```

sets text

Parameters

nstring	- new string

6.4.4 Member Data Documentation

6.4.4.1 bounds

```
sf::Vector2f Keyboard::bounds  [private]
```

store the bounded area

6.4.4.2 ctDeleted

```
std::string Keyboard::ctDeleted  [private]
```

tmp for text deleted to cmd

6.4.4.3 ctEntered

```
std::string Keyboard::ctEntered  [private]
```

tmp for text enterd to cmd

6.4.4.4 tDeleted

```
std::string Keyboard::tDeleted  [private]
```

the text deleted from main box

6.4.4.5 tEntered

```
std::string Keyboard::tEntered  [private]
```

the text entered to main box

6.4.4.6 window

```
sf::RenderWindow* Keyboard::window  [private]
```

refernce to window

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

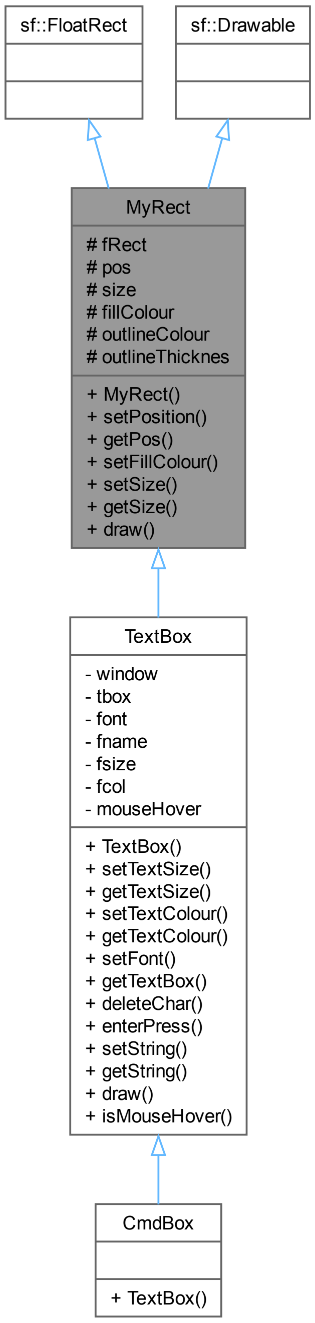
- include/Kamil/Keyboard.h

6.5 MyRect Class Reference

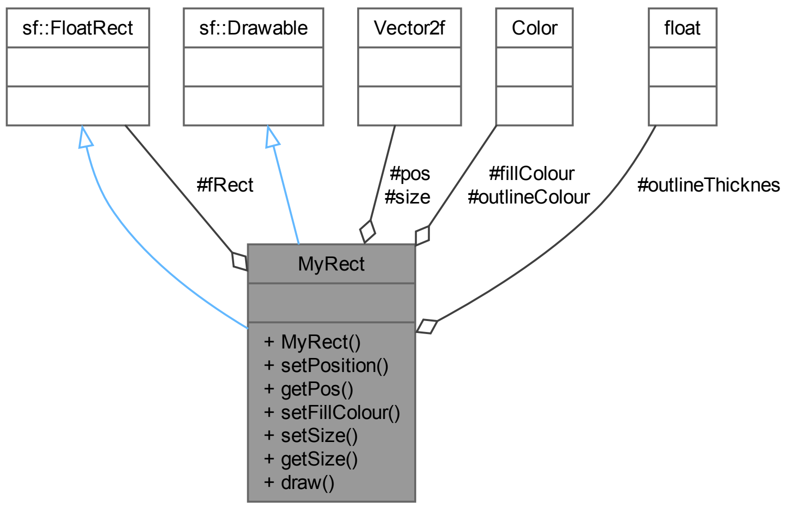
gives extra functionality to FloatRect

```
#include <MyRect.h>
```

Inheritance diagram for MyRect:



Collaboration diagram for MyRect:



Public Member Functions

- MyRect (sf::Vector2f pos, sf::Vector2f size, sf::Color fillColour, sf::Color outlineColour, float outlineThicknes)

  constructor for MyRect
- void setPosition (sf::Vector2f pos)

  sets the position of rect
- sf::Vector2f getPos () const

  get the position of rect
- void setFillColour (sf::Color colour)

  set the fill colour of the rect
- void setSize (sf::Vector2f size)

  set the size of the rect
- sf::Vector2f getSize () const

  get the size of the rect
- void draw (sf::RenderTarget &target, sf::RenderStates states) const override

  virutal method to draw to window

Protected Attributes

- sf::FloatRect fRect
- sf::Vector2f pos
- sf::Vector2f size
- sf::Color fillColour
- sf::Color outlineColour
- float outlineThicknes

6.5.1 Detailed Description

gives extra functionality to FloatRect

Uses FloatRect for the ability to collision detect better than RectangleShape and inherits from Drawable so we are able to keep uniform sytax of window.draw(Drawable object)

6.5.2 Constructor & Destructor Documentation

6.5.2.1 MyRect()

```
MyRect::MyRect (
            sf::Vector2f pos,
            sf::Vector2f size,
            sf::Color fillColour,
            sf::Color outlineColour,
            float outlineThicknes )
```

constructor for MyRect

Parameters

pos	- position of rect
size	- size of rect
fillColour	- fill colour of rect
outlineColour	- ouline colour of rect
outlineThicknes	- outline thickness of rect

6.5.3 Member Function Documentation

6.5.3.1 draw()

```
void MyRect::draw (
            sf::RenderTarget & target,
            sf::RenderStates states ) const  [override]
```

virutal method to draw to window

Inherited from sf::Drawable it is what allows us to draw to the screen using window.draw(MyRect); instead of My←
Rect.draw(window) keeping similar drawing standard to base SFML code making our class more modular and familiar to those who use SFML

Example of polymorphism

6.5.3.2 getPos()

```
sf::Vector2f MyRect::getPos ( ) const
```

get the position of rect

Returns

sf::Vector2f pos

6.5.3.3 getSize()

```
sf::Vector2f MyRect::getSize ( ) const
```

get the size of the rect

Returns

sf::Vector2f size

6.5.3.4 setFillColour()

```
void MyRect::setFillColour (
             sf::Color colour )
```

set the fill colour of the rect

Parameters

sf::Color	colour

6.5.3.5 setPosition()

```
void MyRect::setPosition (
             sf::Vector2f pos )
```

sets the position of rect

Parameters

sf::Vector2f	pos

6.5.3.6 setSize()

```
void MyRect::setSize (
             sf::Vector2f size )
```

set the size of the rect

Parameters

sf::Vector2f	size

6.5.4 Member Data Documentation

6.5.4.1 fillColour

```
sf::Color MyRect::fillColour  [protected]
```

colour of rect

6.5.4.2 fRect

```
sf::FloatRect MyRect::fRect  [protected]
```

for collision checking

6.5.4.3 outlineColour

```
sf::Color MyRect::outlineColour  [protected]
```

outline colour of rect

6.5.4.4 outlineThicknes

```
float MyRect::outlineThicknes  [protected]
```

outline thickness of rect

6.5.4.5 pos

```
sf::Vector2f MyRect::pos  [protected]
```

position of rect

6.5.4.6 size

```
sf::Vector2f MyRect::size  [protected]
```

size of rect

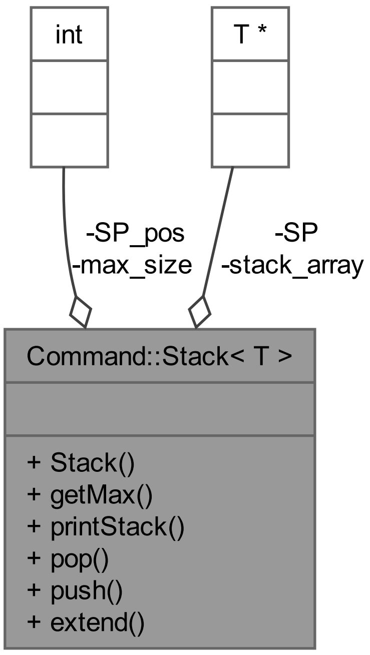
The documentation for this class was generated from the following file:

- include/Kamil/MyRect.h

6.6 Command::Stack< T > Class Template Reference

```
#include <Stack.h>
```

Collaboration diagram for Command::Stack< T >:



Public Member Functions

- Stack (int)
- int getMax () const
- void printStack () const
- int pop ()
- int push (T)
- void extend (int)

Private Attributes

- int max_size {}
- T ∗ stack_array {new T[max_size]}
- T ∗ SP = &stack_array[max_size]
- int SP_pos = max_size

6.6.1 Constructor & Destructor Documentation

6.6.1.1 Stack()

```
template<typename T >
Command::Stack< T >::Stack (
             int   )
```

6.6.2 Member Function Documentation

6.6.2.1 extend()

```
template<typename T >
void Command::Stack< T >::extend (
             int   )
```

6.6.2.2 getMax()

```
template<typename T >
int Command::Stack< T >::getMax ( ) const
```

6.6.2.3 pop()

```
template<typename T >
int Command::Stack< T >::pop ( )
```

6.6.2.4 printStack()

```
template<typename T >
void Command::Stack< T >::printStack ( ) const
```

6.6.2.5 push()

```
template<typename T >
int Command::Stack< T >::push (
             T  )
```

6.6.3 Member Data Documentation

6.6.3.1 max_size

```
template<typename T >
int Command::Stack< T >::max_size {}  [private]
```

6.6.3.2 SP

```
template<typename T >
T* Command::Stack< T >::SP = &stack_array[max_size]  [private]
```

6.6.3.3 SP_pos

```
template<typename T >
int Command::Stack< T >::SP_pos = max_size  [private]
```

6.6.3.4 stack_array

```
template<typename T >
T* Command::Stack< T >::stack_array {new T[max_size]}  [private]
```

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

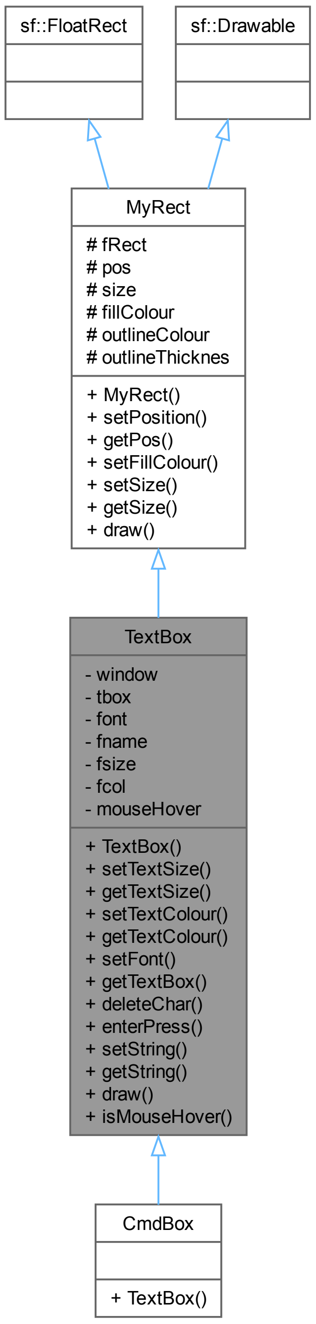
- include/Kamil/Commands.h
- include/Kamil/Utils/Stack.h

6.7 TextBox Class Reference

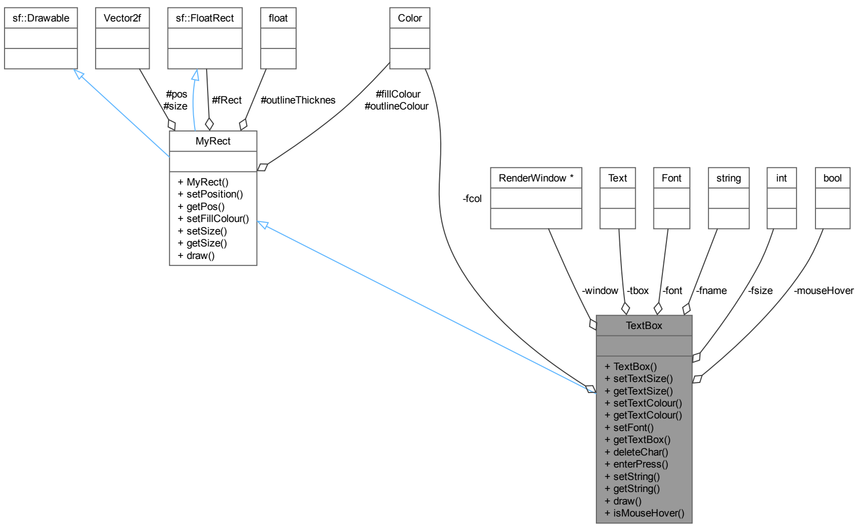
A class that makes a Textbox in SFML.

```
#include <TextBox.h>
```

Inheritance diagram for TextBox:



Collaboration diagram for TextBox:



Public Member Functions

- TextBox (sf::RenderWindow ∗win, sf::Vector2f pos, sf::Vector2f size, std::string sfont, int fsize, sf::Color fcol, sf::Color background, float thicc)

  Constructor for TextBox.
- void setTextSize (int size)

  Set the size of the text.
- int getTextSize () const

  Get the size of the text.
- void setTextColour (sf::Color colour)

  Set the colour of the text.
- sf::Color getTextColour () const

  Get the colour of the text.
- void setFont (sf::Font &font)

  set what font you use
- sf::Text getTextBox () const

  Get both the main textbox and the cmd textbox in a vector.
- void deleteChar ()

  Delete last character entered.
- void enterPress ()

  Handles Enter key press.
- void setString (std::string nstring)

  Sets the string.
- std::string getString () const

  returns the text in tbox
- void draw (sf::RenderTarget &target, sf::RenderStates states) const override

  used to draw to the screen virutal method inherited from MyRect -> sf::Drawable thats overrided here is what allows us to draw to window using window.draw(TextBox)
- bool isMouseHover ()

  check if mouse is hovering over current textbox

Public Member Functions inherited from MyRect

- MyRect (sf::Vector2f pos, sf::Vector2f size, sf::Color fillColour, sf::Color outlineColour, float outlineThicknes)

  constructor for MyRect
- void setPosition (sf::Vector2f pos)

  sets the position of rect
- sf::Vector2f getPos () const

  get the position of rect
- void setFillColour (sf::Color colour)

  set the fill colour of the rect
- void setSize (sf::Vector2f size)

  set the size of the rect
- sf::Vector2f getSize () const

  get the size of the rect
- void draw (sf::RenderTarget &target, sf::RenderStates states) const override

  virutal method to draw to window

Private Attributes

- sf::RenderWindow ∗ window
- sf::Text tbox {}
- sf::Font font {}
- std::string fname {}
- int fsize {}
- sf::Color fcol {}
- bool mouseHover

Additional Inherited Members

Protected Attributes inherited from MyRect

- sf::FloatRect fRect
- sf::Vector2f pos
- sf::Vector2f size
- sf::Color fillColour
- sf::Color outlineColour
- float outlineThicknes

6.7.1 Detailed Description

A class that makes a Textbox in SFML.

The class creates a textbox for inputting and handling text and Keyboard commands and allows the use of commands in the secondary textbox cmdbox

6.7.2 Constructor & Destructor Documentation

6.7.2.1 TextBox()

```
TextBox::TextBox (
            sf::RenderWindow * win,
            sf::Vector2f pos,
            sf::Vector2f size,
            std::string sfont,
            int fsize,
            sf::Color fcol,
            sf::Color background,
            float thicc )
```

Constructor for TextBox.

Parameters

win	- RenderWindow the TextBox is drawn onto
pos	- the initial position of the TextBox
size	- the initial size of the TextBox
sfont	- the initial font used by the TextBox
fsize	- the inital font size
fcol	- the initial font colour
background	- the initial background colour
thicc	- the padding for the RectangleShape

6.7.3 Member Function Documentation

6.7.3.1 deleteChar()

```
void TextBox::deleteChar ( )
```

Delete last character entered.

6.7.3.2 draw()

```
void TextBox::draw (
            sf::RenderTarget & target,
            sf::RenderStates states ) const  [override]
```

used to draw to the screen virutal method inherited from MyRect -> sf::Drawable thats overrided here is what allows us to draw to window using window.draw(TextBox)

Example of polymorphism

6.7.3.3 enterPress()

```
void TextBox::enterPress ( )
```

Handles Enter key press.

6.7.3.4 getString()

```
std::string TextBox::getString ( ) const
```

returns the text in tbox

Returns

type std::string

6.7.3.5 getTextBox()

```
sf::Text TextBox::getTextBox ( ) const
```

Get both the main textbox and the cmd textbox in a vector.

Returns

type Boxv2 that contains textbox and cmdbox

6.7.3.6 getTextColour()

```
sf::Color TextBox::getTextColour ( ) const
```

Get the colour of the text.

Returns

sf::Colour textColour

6.7.3.7 getTextSize()

```
int TextBox::getTextSize ( ) const
```

Get the size of the text.

Returns

an int of the text size

6.7.3.8 isMouseHover()

```
bool TextBox::isMouseHover ( )
```

check if mouse is hovering over current textbox

Returns

bool - yes if hovering

6.7.3.9 setFont()

```
void TextBox::setFont (
             sf::Font & font )
```

set what font you use

Parameters

font	file dir of font

6.7.3.10 setString()

```
void TextBox::setString (
             std::string nstring )
```

Sets the string.

Parameters

nstring	- new string placed on tbox

6.7.3.11 setTextColour()

```
void TextBox::setTextColour (
             sf::Color colour )
```

Set the colour of the text.

Parameters

fill	font colour

6.7.3.12 setTextSize()

```
void TextBox::setTextSize (
             int size )
```

Set the size of the text.

Parameters

size	text size

6.7.4 Member Data Documentation

6.7.4.1 fcol

```
sf::Color TextBox::fcol {}  [private]
```

the font colour

6.7.4.2 fname

```
std::string TextBox::fname {}  [private]
```

the name of the font used

6.7.4.3 font

```
sf::Font TextBox::font {}  [private]
```

the font that the TextBox uses

6.7.4.4 fsize

```
int TextBox::fsize {}  [private]
```

the font size

6.7.4.5 mouseHover

```
bool TextBox::mouseHover  [private]
```

if the mouse is hovering over

6.7.4.6 tbox

```
sf::Text TextBox::tbox {}  [private]
```

the text that everything is written onto

6.7.4.7 window

```
sf::RenderWindow* TextBox::window  [private]
```

pointer to the main RenderWindow variable

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

- include/Kamil/TextBox.h

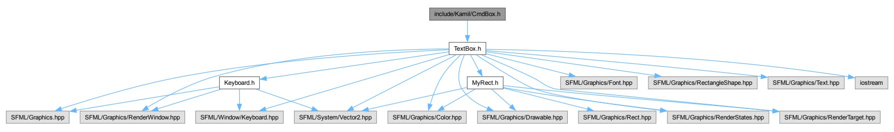
Chapter 7

File Documentation

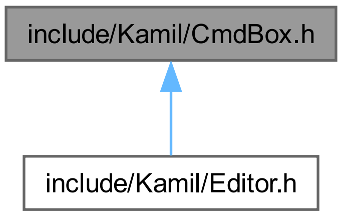
7.1 include/Kamil/CmdBox.h File Reference

```
#include "TextBox.h"
```

Include dependency graph for CmdBox.h:



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



Classes

- class CmdBox

  Class to handle the command TextBox.

7.2 CmdBox.h

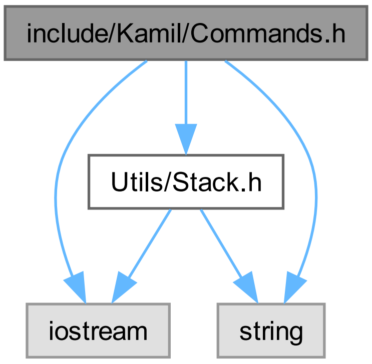
Go to the documentation of this file.

```
#ifndef KAMIL_CMDBOX_H
#define KAMIL_CMDBOX_H

#include "TextBox.h"

class CmdBox : public TextBox
{
public:
    using TextBox::TextBox;
};
#endif // KAMIL_CMDBOX_H
```

7.3 include/Kamil/Commands.h File Reference

```
#include <iostream>
#include <string>
#include "Utils/Stack.h"
```

Include dependency graph for Commands.h:



Namespaces

- namespace Command

  A stack in the Command namespace.

7.4 Commands.h

Go to the documentation of this file.

```
#ifndef KAMIL_COMMANDS_H
#define KAMIL_COMMANDS_H

#include <iostream>
#include <string>

#include "Utils/Stack.h"
```

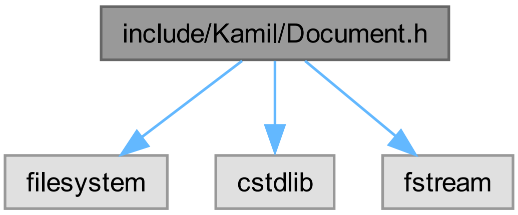
```

namespace Command{
    // template <typename T>
    // class Node{ // used for LinkedList
    //     public:
    //         Node();
    //         Node(T);
    //         T data;
    //         Node* next;
    // };

    // template <typename T>
    // class LinkedList{
    //     public:
    //         LinkedList();
    //         void insertNode(int);
    //         void printList();
    //         void deleteNode(int);
    //     private:
    //         Node<T>* head;
    // };
    template <typename>
    class Stack;

//    class Undo{};

//    class Redo{};
}

#endif // KAMIL_COMMANDS_H
```

7.5 include/Kamil/Document.h File Reference

Interface file for the Document class.

```
#include <filesystem>
#include <cstdlib>
#include <fstream>
```

Include dependency graph for Document.h:



Classes

- class Document

  Document class.

7.5.1 Detailed Description

Interface file for the Document class.

The Document.h file is responsible for all File I/O between the system and the program it can read and write files and will also push some work off to python scripts to handle config files

7.6 Document.h

Go to the documentation of this file.

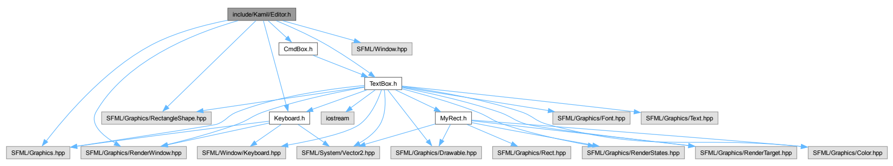
```
#ifndef KAMIL_DOCUMENT_H
#define KAMIL_DOCUMENT_H


#include <filesystem>
#include <cstdlib>
#include <fstream>

class Document{
public:
    Document();

    Document(std::string fileP);

    void init();

    void readFile();

    std::string getRelPath();

    std::string getAbsPath();

    void createFile();

    void createDir();

    bool saveFile(const std::string& filename);

    bool hasChanged();

    // void addTextToPos(std::string txt, int pos);

private:
    std::string relPath;
    std::string absPath;
    std::string buffInfo;
    bool docChanged;
};
#endif // KAMIL_DOCUMENT_H
```

7.7 include/Kamil/Editor.h File Reference

Interface file for the Editor class.

```
#include <SFML/Graphics.hpp>
#include <SFML/Graphics/RectangleShape.hpp>
#include <SFML/Graphics/RenderWindow.hpp>
#include <SFML/Window.hpp>
#include "TextBox.h"
#include "Keyboard.h"
#include "CmdBox.h"
```

Include dependency graph for Editor.h:



Classes

- class Editor

  Class that handles and draws everything in the Editor.

7.7.1 Detailed Description

Interface file for the Editor class.

The Editor class is responsible for the interaction between the different classes. All things outside the main while loop will be checked or initialise. Anything to do with the Editor Window will happen here

7.8 Editor.h

Go to the documentation of this file.

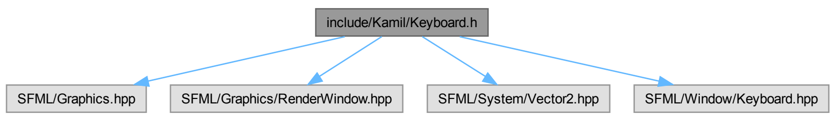
```
#ifndef KAMIL_EDITOR_WINDOW_HPP
#define KAMIL_EDITOR_WINDOW_HPP

#include <SFML/Graphics.hpp>
#include <SFML/Graphics/RectangleShape.hpp>
#include <SFML/Graphics/RenderWindow.hpp>
#include <SFML/Window.hpp>

#include "TextBox.h"
#include "Keyboard.h"
#include "CmdBox.h"


class Editor{
    public:
        Editor(sf::RenderWindow*, sf::Event*);

        ~Editor();

        void draw();

        void handleEvent();
    private:
        TextBox* textBox;
        CmdBox* cbox;
        sf::RenderWindow* window;
        sf::Event* event;
        Keyboard kb;
};

#endif // KAMIL_EDITOR_WINDOW_HPP
```

7.9 include/Kamil/Keyboard.h File Reference

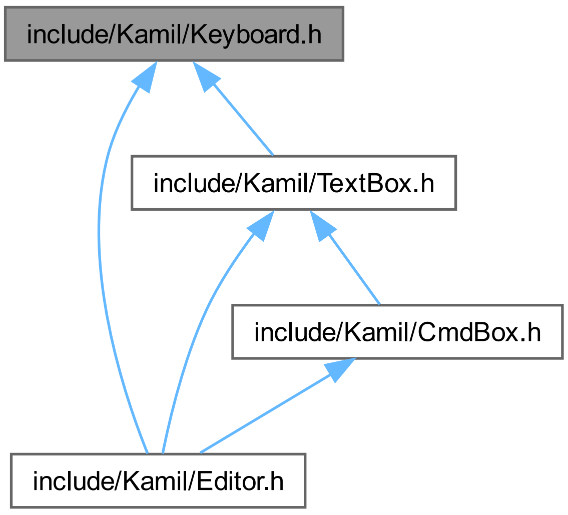
Interface file for Keyboard.h.

```
#include <SFML/Graphics.hpp>
#include <SFML/Graphics/RenderWindow.hpp>
#include <SFML/System/Vector2.hpp>
#include <SFML/Window/Keyboard.hpp>
```

Include dependency graph for Keyboard.h:



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



Classes

- class Keyboard

  A class to handle Keyboard input.

Namespaces

- namespace KEYS

  An enum for Keyboard characters in hex form.

Enumerations

- enum {
  KEYS::ESCAPE = 0x1B , KEYS::ENTER = 0xD , KEYS::BS = 0x8 , KEYS::Shift_A = 0x41 ,
  KEYS::CTRL = 0x11 , KEYS::DELETE = 0x7f }

7.9.1 Detailed Description

Interface file for Keyboard.h.

A class that handles all keyboard and mouse events for the editor is responsible for manging input of keyboard data and their corresponding command

7.10 Keyboard.h

Go to the documentation of this file.
```
#ifndef KAMIL_KEYBOARD_H
#define KAMIL_KEYBOARD_H


#include <SFML/Graphics.hpp>
#include <SFML/Graphics/RenderWindow.hpp>
#include <SFML/System/Vector2.hpp>
#include <SFML/Window/Keyboard.hpp>

namespace KEYS{
    enum {
        ESCAPE = 0x1B,
        ENTER = 0xD,
        BS = 0x8,
        Shift_A = 0x41,
        CTRL = 0x11,
        DELETE = 0x7f,
    };
}


class Keyboard{
    public:
        Keyboard(sf::RenderWindow* win, sf::Vector2f bounds);


        bool isKeyPressed(sf::Keyboard::Key);

        bool isTextEntered();

        bool isCmdTextEntered();

        bool isTextDeleted();

        std::string getTextEntered();


        std::string getCmdTextEntered();


        void setTextEntered(std::string);


        void setCmdTextEntered(std::string);


        sf::Vector2f getBounds() const;

        void backSpace();

        void handleKeyEvent(sf::Event& event);


        void handleCmdKeyEvent();

        void handleMouseEvent(sf::Event& event); // not implemented yet

    private:
        sf::RenderWindow* window;
        sf::Vector2f bounds;
        std::string tEntered;
        std::string tDeleted;
        std::string ctEntered;
        std::string ctDeleted;
};
#endif // KAMIL_KEYBOARD_H
```

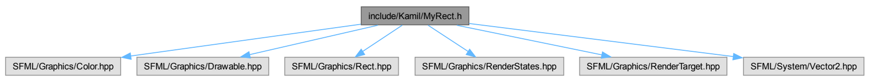
7.11 include/Kamil/MyRect.h File Reference

Interface file for the MyRect class.

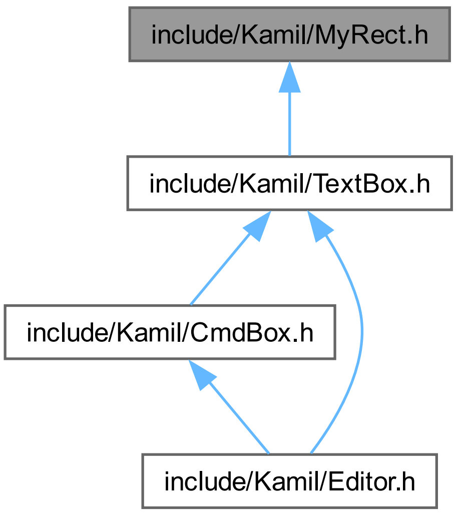
```
#include <SFML/Graphics/Color.hpp>
#include <SFML/Graphics/Drawable.hpp>
#include <SFML/Graphics/Rect.hpp>
```

```
#include <SFML/Graphics/RenderStates.hpp>
#include <SFML/Graphics/RenderTarget.hpp>
#include <SFML/System/Vector2.hpp>
```

Include dependency graph for MyRect.h:



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



Classes

- class MyRect

  gives extra functionality to FloatRect

7.11.1 Detailed Description

Interface file for the MyRect class.

Inherits from sf::FloatRect and sf::Drawable. sf::FloatRect is a templated class of sf::Rect<float> and its primary use is for defining the border and creating a hollow rectangle, as such it only has methods for collision detection and intersections. The normal RectangleShape class creates a basic rectangle without the collision and intersections checking so we inherit this functionality from FloatRect and in effect add it to the instantiated RectangleShape in the MyRect class.

The sf::Drawable is only here to add a draw property to our class so when we draw to the RenderTarget, in this case RenderWindow, we can use the same code of window.draw(our_own_object) instead of the general our_own_↩ object.draw(window). This is done so when others use this code it makes it easier for them to follow a standard way of drawing to the RenderTarget and not having to worry about passing parameters into the objects.

7.12 MyRect.h

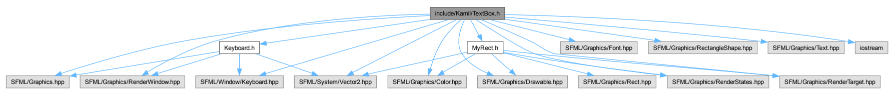
Go to the documentation of this file.

```
#ifndef KAMIL_MYRECT_H
#define KAMIL_MYRECT_H


#include <SFML/Graphics/Color.hpp>
#include <SFML/Graphics/Drawable.hpp>
#include <SFML/Graphics/Rect.hpp>
#include <SFML/Graphics/RenderStates.hpp>
#include <SFML/Graphics/RenderTarget.hpp>
#include <SFML/System/Vector2.hpp>


class MyRect : public sf::FloatRect
             , public sf::Drawable
{
    public:
        MyRect(sf::Vector2f pos, sf::Vector2f size, sf::Color fillColour, sf::Color outlineColour,
float outlineThicknes);

        void setPosition(sf::Vector2f pos);

        sf::Vector2f getPos()const;

        void setFillColour(sf::Color colour);

        void setSize(sf::Vector2f size);

        sf::Vector2f getSize()const;

        void draw(sf::RenderTarget& target, sf::RenderStates states)const override;

    protected:
        sf::FloatRect fRect;
        sf::Vector2f pos;
        sf::Vector2f size;
        sf::Color fillColour;
        sf::Color outlineColour;
        float outlineThicknes;
};

#endif // KAMIL_MYRECT_H

```

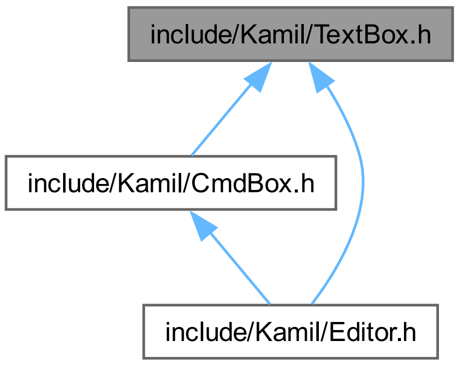
7.13 include/Kamil/TextBox.h File Reference

```
#include <SFML/Graphics.hpp>
#include <SFML/Graphics/Color.hpp>
#include <SFML/Graphics/Drawable.hpp>
#include <SFML/Graphics/Font.hpp>
#include <SFML/Graphics/RectangleShape.hpp>
#include <SFML/Graphics/RenderStates.hpp>
#include <SFML/Graphics/RenderTarget.hpp>
#include <SFML/Graphics/RenderWindow.hpp>
#include <SFML/Graphics/Text.hpp>
#include <SFML/System/Vector2.hpp>
#include <SFML/Window/Keyboard.hpp>
#include <iostream>
#include "Keyboard.h"
#include "MyRect.h"
```

Include dependency graph for TextBox.h:



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



Classes

- class TextBox

  A class that makes a Textbox in SFML.

7.14 TextBox.h

Go to the documentation of this file.

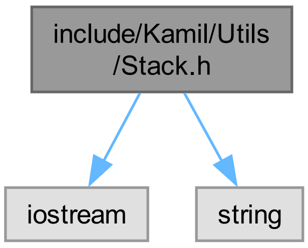
```
#ifndef KAMIL_TEXTBOX_HPP
#define KAMIL_TEXTBOX_HPP

#include <SFML/Graphics.hpp>
#include <SFML/Graphics/Color.hpp>
#include <SFML/Graphics/Drawable.hpp>
#include <SFML/Graphics/Font.hpp>
#include <SFML/Graphics/RectangleShape.hpp>
#include <SFML/Graphics/RenderStates.hpp>
#include <SFML/Graphics/RenderTarget.hpp>
#include <SFML/Graphics/RenderWindow.hpp>
#include <SFML/Graphics/Text.hpp>
#include <SFML/System/Vector2.hpp>
#include <SFML/Window/Keyboard.hpp>
#include <iostream>

#include "Keyboard.h"
#include "MyRect.h"


/*
 *
 * TODO:
 *       Make a RectangleShape that acts as the bounds of the TextBox
 *       then add limits to the textbox so it stays in the limits
 *
 *       Add the Keybord manager class here and use its methods
 *       to handle the key events
 */


class TextBox : public MyRect
{
    public:
        TextBox(sf::RenderWindow* win, sf::Vector2f pos, sf::Vector2f size, std::string sfont, int
fsize, sf::Color fcol, sf::Color background, float thicc);

```

```
        void setTextSize(int size);

        int getTextSize()const;

        void setTextColour(sf::Color colour);

        sf::Color getTextColour()const;

        void setFont(sf::Font& font);

        sf::Text getTextBox()const;

        void deleteChar();

        void enterPress();

        void setString(std::string nstring);

        std::string getString()const;

        void draw(sf::RenderTarget& target, sf::RenderStates states)const override;

        bool isMouseHover();

    private:
        sf::RenderWindow* window;
        sf::Text tbox{};
        sf::Font font{};
        std::string fname{};
        int fsize{};
        sf::Color fcol{};
        bool mouseHover;
};
#endif // KAMIL_TEXTBOX_HPP
```

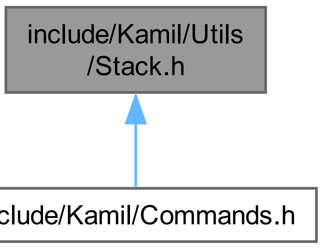
7.15 include/Kamil/Utils/Stack.h File Reference

```
#include <iostream>
#include <string>
```

Include dependency graph for Stack.h:



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



Classes

- class Command::Stack< T >

Namespaces

- namespace Command

  A stack in the Command namespace.

7.16 Stack.h

Go to the documentation of this file.

```
#ifndef KAMIL_STACK_H
#define KAMIL_STACK_H

#include <iostream>
#include <string>

namespace Command{

// Dynamic stack array
template<typename T>
class Stack{
public:
    Stack(int);
    int getMax()const;
    void printStack()const;
    int pop();
    int push(T);
    void extend(int);
private:
    int max_size{};
    T* stack_array{new T[max_size]};
    T* SP = &stack_array[max_size];
    int SP_pos = max_size;
};

#endif

} // Command
```

Index