

光通信系统仿真平台搭建的探索 Version 1.0.0 基础版本

开发文档

1 命名空间索引	1
1.1 命名空间列表	1
2 继承关系索引	3
2.1 类继承关系	3
3 结构体索引	5
3.1 结构体	5
4 文件索引	7
4.1 文件列表	7
5 命名空间文档	9
5.1 Ui 命名空间参考	9
6 结构体说明	11
6.1 Arrow类 参考	11
6.1.1 详细描述	13
6.1.2 成员枚举类型说明	13
6.1.2.1 anonymous enum	13
6.1.3 构造及析构函数说明	13
6.1.3.1 Arrow()	13
6.1.4 成员函数说明	14
6.1.4.1 boundingRect()	14
6.1.4.2 endItem()	14
6.1.4.3 paint()	14
6.1.4.4 shape()	15
6.1.4.5 startItem()	15
6.1.4.6 type()	16
6.1.4.7 updatePosition()	16
6.1.5 结构体成员变量说明	16
6.1.5.1 arrowHead	16
6.1.5.2 myColor	16
6.1.5.3 myEndItem	17
6.1.5.4 myStartItem	17
6.2 IOset类 参考	17
6.2.1 详细描述	18
6.2.2 构造及析构函数说明	18
6.2.2.1 IOset()	18
6.2.2.2 ~IOset()	19
6.2.3 成员函数说明	19
6.2.3.1 getOutputValue	19
6.2.3.2 on_SetInput_clicked	19
6.2.3.3 sendInputValue	19

6.2.4 结构体成员变量说明	20
6.2.4.1 ui	20
6.3 Items类 参考	20
6.3.1 详细描述	22
6.3.2 成员枚举类型说明	22
6.3.2.1 anonymous enum	22
6.3.2.2 ItemType	22
6.3.3 构造及析构函数说明	22
6.3.3.1 Items()	23
6.3.4 成员函数说明	23
6.3.4.1 addArrow()	23
6.3.4.2 calculateResult()	24
6.3.4.3 contextMenuEvent()	24
6.3.4.4 getInputNum()	24
6.3.4.5 getItemType()	25
6.3.4.6 getOutputNum()	26
6.3.4.7 image()	26
6.3.4.8 itemChange()	26
6.3.4.9 itemType()	27
6.3.4.10 polygon()	28
6.3.4.11 removeArrow()	28
6.3.4.12 removeArrows()	29
6.3.4.13 setInputNum()	29
6.3.4.14 type()	29
6.3.5 结构体成员变量说明	30
6.3.5.1 arrows	30
6.3.5.2 inputNum	30
6.3.5.3 myContextMenu	30
6.3.5.4 myItemType	30
6.3.5.5 myPolygon	30
6.3.5.6 outputNum	31
6.4 MainWindow类 参考	31
6.4.1 详细描述	34
6.4.2 构造及析构函数说明	34
6.4.2.1 MainWindow()	34
6.4.3 成员函数说明	34
6.4.3.1 about	34
6.4.3.2 bringToFront	34
6.4.3.3 buttonGroupClicked	34
6.4.3.4 createActions()	35
6.4.3.5 createBackgroundCellWidget()	35
6.4.3.6 createCellWidget()	35

6.4.3.7 createColorIcon()	36
6.4.3.8 createColorMenu()	36
6.4.3.9 createColorToolButtonIcon()	36
6.4.3.10 createMenus()	36
6.4.3.11 createToolbars()	36
6.4.3.12 createToolBox()	36
6.4.3.13 currentFontChanged	36
6.4.3.14 deleteItem	37
6.4.3.15 fontSizeChanged	37
6.4.3.16 getInputValue	38
6.4.3.17 handleFontChange	38
6.4.3.18 itemInserted	38
6.4.3.19 itemSelected	39
6.4.3.20 pointerGroupClicked	39
6.4.3.21 sceneScaleChanged	39
6.4.3.22 sendOutputValue	39
6.4.3.23 sendToBack	40
6.4.3.24 setGetValueWindow	40
6.4.3.25 textInserted	40
6.4.4 结构体成员变量说明	40
6.4.4.1 aboutAction	40
6.4.4.2 AboutMenu	41
6.4.4.3 addAction	41
6.4.4.4 boldAction	41
6.4.4.5 buttonGroup	41
6.4.4.6 colorToolBar	41
6.4.4.7 deleteAction	41
6.4.4.8 editToolBar	42
6.4.4.9 exitAction	42
6.4.4.10 fileMenu	42
6.4.4.11 fillAction	42
6.4.4.12 fontCombo	42
6.4.4.13 fontSizeCombo	42
6.4.4.14 italicAction	43
6.4.4.15 itemColorCombo	43
6.4.4.16 itemMenu	43
6.4.4.17 lineAction	43
6.4.4.18 pointerToolBar	43
6.4.4.19 pointerTypeGroup	43
6.4.4.20 scene	44
6.4.4.21 sceneScaleCombo	44
6.4.4.22 sendBackAction	44

6.4.4.23 setGetAction	44
6.4.4.24 setGetWindow	44
6.4.4.25 textAction	44
6.4.4.26 textColorCombo	45
6.4.4.27 textToolBar	45
6.4.4.28 toFrontAction	45
6.4.4.29 toolBox	45
6.4.4.30 underlineAction	45
6.4.4.31 view	45
6.5 SimulationScene类 参考	46
6.5.1 详细描述	47
6.5.2 成员枚举类型说明	47
6.5.2.1 Mode	47
6.5.3 构造及析构函数说明	48
6.5.3.1 SimulationScene()	48
6.5.4 成员函数说明	48
6.5.4.1 editorLostFocus	48
6.5.4.2 font()	49
6.5.4.3 isItemChange()	49
6.5.4.4 itemInserted	50
6.5.4.5 itemSelected	50
6.5.4.6 mouseMoveEvent()	50
6.5.4.7 mousePressEvent()	51
6.5.4.8 mouseReleaseEvent()	51
6.5.4.9 setFont()	52
6.5.4.10 setItemType	53
6.5.4.11 setMode	53
6.5.4.12 textInserted	53
6.5.5 结构体成员变量说明	54
6.5.5.1 leftButtonDown	54
6.5.5.2 line	54
6.5.5.3 myFont	54
6.5.5.4 myItemColor	54
6.5.5.5 myItemMenu	54
6.5.5.6 myItemType	55
6.5.5.7 myLineColor	55
6.5.5.8 myMode	55
6.5.5.9 myTextColor	55
6.5.5.10 startPoint	55
6.5.5.11 textItem	55
6.6 TextItem类 参考	56
6.6.1 详细描述	57

6.6.2 成员枚举类型说明	57
6.6.2.1 anonymous enum	57
6.6.3 构造及析构函数说明	57
6.6.3.1 TextItem()	57
6.6.4 成员函数说明	57
6.6.4.1 focusOutEvent()	57
6.6.4.2 itemChange()	58
6.6.4.3 lostFocus	58
6.6.4.4 mouseDoubleClickEvent()	59
6.6.4.5 selectedChange	60
6.6.4.6 type()	60
7 文件说明	61
7.1 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/arrow.cpp 文件参考	61
7.1.1 详细描述	61
7.2 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/arrow.h 文件参考	62
7.2.1 详细描述	63
7.3 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/ioset.cpp 文件参考	64
7.3.1 详细描述	65
7.4 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/ioset.h 文件参考	65
7.4.1 详细描述	67
7.5 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/items.cpp 文件参考	67
7.5.1 详细描述	68
7.6 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/items.h 文件参考	69
7.6.1 详细描述	69
7.7 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/main.cpp 文件参考	70
7.7.1 详细描述	71
7.7.2 函数说明	72
7.7.2.1 main()	72
7.8 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/mainwindow.cpp 文件参考	72
7.8.1 详细描述	73
7.8.2 变量说明	73
7.8.2.1 InsertTextButton	73
7.9 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/mainwindow.h 文件参考	74
7.9.1 详细描述	75
7.10 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/simulationScene.cpp 文件参考	75
7.10.1 详细描述	76
7.11 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/simulationScene.h 文件参考	77
7.11.1 详细描述	77
7.12 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/textItem.cpp 文件参考	78
7.12.1 详细描述	79
7.13 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/textItem.h 文件参考	80

7.13.1 详细描述	81
Index	83

Chapter 1

命名空间索引

1.1 命名空间列表

这里列出了所有命名空间定义,附带简要说明:

Ui	9
----------	---

Chapter 2

继承关系索引

2.1 类继承关系

此继承关系列表按字典顺序粗略的排序:

QDialog	
IOset	17
QGraphicsLineItem	
Arrow	11
QGraphicsPolygonItem	
Items	20
QGraphicsScene	
SimulationScene	46
QGraphicsTextItem	
TextItem	56
QMainWindow	
MainWindow	31

Chapter 3

结构体索引

3.1 结构体

这里列出了所有结构体，并附带简要说明:

Arrow	11
IOset	17
Items	20
MainWindow	31
SimulationScene	46
TextItem	56

Chapter 4

文件索引

4.1 文件列表

这里列出了所有文件，并附带简要说明:

C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/arrow.cpp	
连接线函数文件	61
C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/arrow.h	
项目连接线函数头文件	62
C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/ioset.cpp	
设置对应项的输入及显示输出的功能文件	64
C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/ioset.h	
设置对应项的输入及显示输出的功能文件	65
C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/items.cpp	
功能项函数文件	67
C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/items.h	
项目功能项函数声明头文件	69
C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/main.cpp	
QT 程序运行主函数	70
C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/mainwindow.cpp	
项目主函数文件	72
C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/mainwindow.h	
项目主函数头文件	74
C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/simulationScene.cpp	
仿真场景实现函数文件	75
C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/simulationScene.h	
项目场景函数头文件	77
C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/textItem.cpp	
文本编辑框函数文件	78
C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/textItem.h	
可编辑文本项类声明头文件	80

Chapter 5

命名空间文档

5.1 Ui 命名空间参考

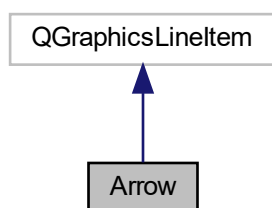
Chapter 6

结构体说明

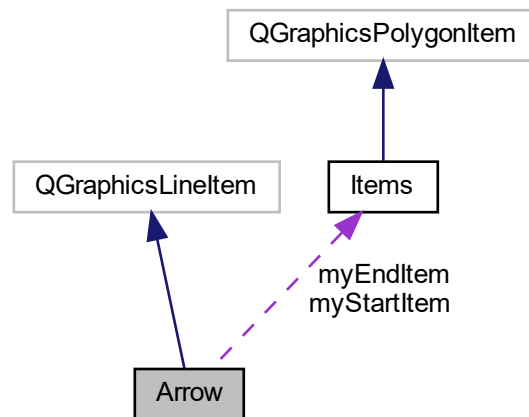
6.1 Arrow类 参考

```
#include <arrow.h>
```

类 Arrow 继承关系图:



Arrow 的协作图:



Public 类型

- enum { **Type** = UserType + 4 }

Public 成员函数

- **Arrow** (Items *startItem, Items *endItem, QGraphicsItem *parent=0)
- int **type** () const override
- QRectF **boundingRect** () const override
Arrow::boundingRect 重写 *QGraphicsLineItem* 函数以检测碰撞 我们需要重新实现这个函数，因为 *Arrow* 比 *QGraphicsLineItem* 的边界矩形大。图形场景使用边界矩形来知道场景的哪些区域需要更新。
- QPainterPath **shape** () const override
Arrow::shape 重写 *QGraphicsLineItem* 函数以检测是否选择
- Items * **startItem** () const
- Items * **endItem** () const
- void **updatePosition** ()
Arrow::updatePosition 在起止项之间设立连接线

Protected 成员函数

- void **paint** (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget=0) override
Arrow::paint 连接线绘制函数

Private 属性

- Items * **myStartItem**
- Items * **myEndItem**
- QPolygonF **arrowHead**
- QColor **myColor**

6.1.1 详细描述

在文件 `arrow.h` 第 36 行定义.

6.1.2 成员枚举类型说明

6.1.2.1 anonymous enum

`anonymous enum`

枚举值

Type	
------	--

在文件 `arrow.h` 第 39 行定义.

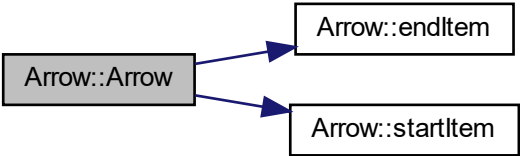
6.1.3 构造及析构函数说明

6.1.3.1 Arrow()

```
Arrow::Arrow (  
    Items * startItem,  
    Items * endItem,  
    QGraphicsItem * parent = 0 )
```

在文件 `arrow.cpp` 第 30 行定义.

函数调用图:



6.1.4 成员函数说明

6.1.4.1 boundingRect()

```
QRectF Arrow::boundingRect ( ) const [override]
```

`Arrow::boundingRect` 重写 `QGraphicsLineItem` 函数以检测碰撞 我们需要重新实现这个函数，因为 `Arrow` 比 `QGraphicsLineItem` 的边界矩形大。图形场景使用边界矩形来知道场景的哪些区域需要更新。

返回

返回重写后的矩形 `QRectF`

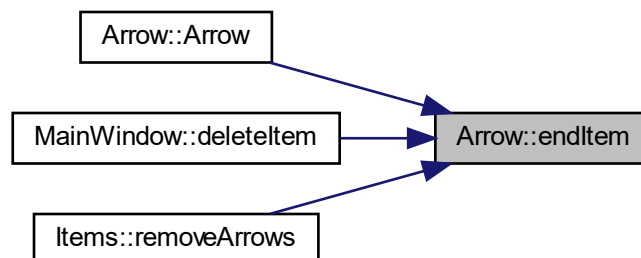
在文件 `arrow.cpp` 第 46 行定义.

6.1.4.2 endItem()

```
Items* Arrow::endItem ( ) const [inline]
```

在文件 `arrow.h` 第 47 行定义.

这是这个函数的调用关系图:



6.1.4.3 paint()

```
void Arrow::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget = 0 ) [override], [protected]
```

`Arrow::paint` 连接线绘制函数

参数

<i>painter</i>	QPainter
----------------	----------

在文件 `arrow.cpp` 第 76 行定义.

函数调用图:



6.1.4.4 shape()

```
QPainterPath Arrow::shape ( ) const [override]
```

`Arrow::shape` 重写 `QGraphicsLineItem` 函数以检测是否选择

返回

返回一个带有箭头和路径的连接线 `QPainterPath`

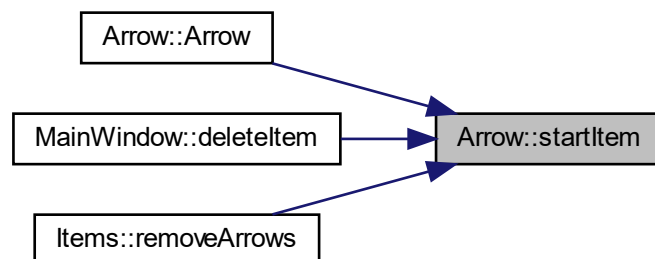
在文件 `arrow.cpp` 第 56 行定义.

6.1.4.5 startItem()

```
Items* Arrow::startItem ( ) const [inline]
```

在文件 `arrow.h` 第 46 行定义.

这是这个函数的调用关系图:



6.1.4.6 type()

```
int Arrow::type ( ) const [inline], [override]
```

在文件 `arrow.h` 第 43 行定义.

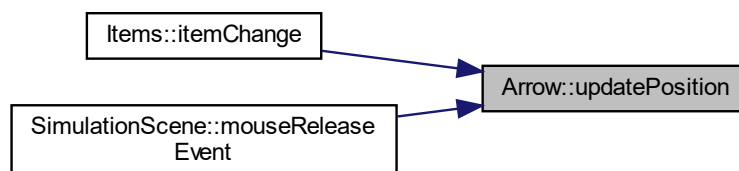
6.1.4.7 updatePosition()

```
void Arrow::updatePosition ( )
```

[Arrow::updatePosition](#) 在起止项之间设立连接线

在文件 `arrow.cpp` 第 66 行定义.

这是这个函数的调用关系图:



6.1.5 结构体成员变量说明

6.1.5.1 arrowHead

```
QPolygonF Arrow::arrowHead [private]
```

在文件 `arrow.h` 第 57 行定义.

6.1.5.2 myColor

```
QColor Arrow::myColor [private]
```

在文件 `arrow.h` 第 58 行定义.

6.1.5.3 myEndItem

```
Items* Arrow::myEndItem [private]
```

在文件 arrow.h 第 56 行定义.

6.1.5.4 myStartItem

```
Items* Arrow::myStartItem [private]
```

在文件 arrow.h 第 55 行定义.

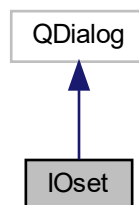
该类的文档由以下文件生成:

- C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/arrow.h
- C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/arrow.cpp

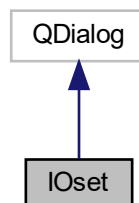
6.2 IOset类 参考

```
#include <ioset.h>
```

类 IOset 继承关系图:



IOset 的协作图:



信号

- void `sendInputValue` (double inputNum)

Public 成员函数

- `IOset` (QWidget *parent=nullptr)
`IOset::IOset` ui类对应构造函数
- `~IOset` ()
`IOset::~~IOset` ui类对应析构函数

Private 槽

- void `on_SetInput_clicked` ()
`IOset::on_SetInput_clicked` Set Input 按钮点击事件，将inputValue更新到item的inputNum中
- void `getOutputValue` (double outputNum)
`IOset::getOutputValue` 获取主窗口对应项的输出值，并显示在文本框中

Private 属性

- Ui::IOset * `ui`

6.2.1 详细描述

在文件 `ioset.h` 第 31 行定义.

6.2.2 构造及析构函数说明

6.2.2.1 IOset()

```
IOset::IOset (
    QWidget * parent = nullptr ) [explicit]
```

`IOset::IOset` ui类对应构造函数

参数

<code>parent</code>	QWidget
---------------------	---------

在文件 `ioset.cpp` 第 30 行定义.

6.2.2.2 ~IOset()

IOset::~~IOset ()

IOset::~~IOset ui类对应析构函数

在文件 ioaset.cpp 第 40 行定义.

6.2.3 成员函数说明

6.2.3.1 getOutputValue

void IOset::getOutputValue (
double outputNum) [private], [slot]

IOset::getOutputValue 获取主窗口对应项的输出值，并显示在文本框中

参数

<i>outputNum</i>	输出值
------------------	-----

在文件 ioaset.cpp 第 49 行定义.

6.2.3.2 on.SetInput_clicked

void IOset::on.SetInput_clicked () [private], [slot]

IOset::on.SetInput_clicked Set Input 按钮点击事件，将inputValue更新到item的inputNum中

参数

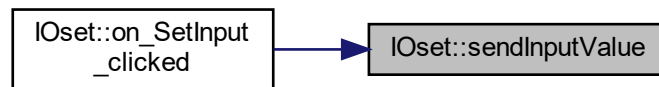
<i>item</i>	需要更新的item
-------------	-----------

在文件 ioaset.cpp 第 60 行定义.

6.2.3.3 sendInputValue

void IOset::sendInputValue (
double inputNum) [signal]

这是这个函数的调用关系图:



6.2.4 结构体成员变量说明

6.2.4.1 ui

```
Ui::IOset* IOset::ui [private]
```

在文件 `ioset.h` 第 48 行定义.

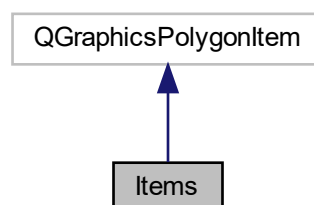
该类的文档由以下文件生成:

- [C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/ioset.h](#)
- [C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/ioset.cpp](#)

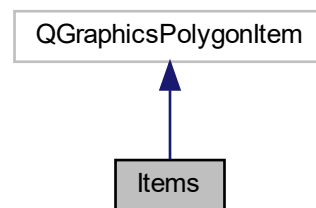
6.3 Items类 参考

```
#include <items.h>
```

类 `Items` 继承关系图:



Items 的协作图:



Public 类型

- enum { `Type` = UserType + 15 }
- enum `ItemType` { `test` }

Public 成员函数

- `Items (ItemType itemType, QMenu *contextMenu, QGraphicsItem *parent=0)`
- void `removeArrow (Arrow *arrow)`
Items::removeArrow 当与连接线相关的项被删除时，删除该连接线
- void `removeArrows ()`
Items::removeArrows 当一个项被删除时，删除与这个项有关的所有连接线
- `ItemType itemType () const`
- `QPolygonF polygon () const`
- void `addArrow (Arrow *arrow)`
Items::addArrow 将箭头添加到箭头列表
- `QPixmap image () const`
Items::image 为工具箱中的工具按钮创建图标
- int `type () const override`
- void `setInputNum (double inputNum)`
Items::setInputNum 设置该Item的InputNum
- `ItemType getItemType ()`
Items::getItemType 获取对应项的种类
- double `getOutputNum ()`
Items::getOutputNum 输出该Item的OutputNum
- double `getInputNum ()`
Items::getInputNum 输出该Item的inputNum
- double `calculateResult (ItemType myItemType, double inputNum)`
Items::calculateResult 根据不同的项进行不同的计算

Protected 成员函数

- void `contextMenuEvent (QGraphicsSceneContextMenuEvent *event) override`
Items::contextMenuEvent 鼠标右键时显示菜单，默认情况下不选择项目
- `QVariant itemChange (GraphicsItemChange change, const QVariant &value) override`
Items::itemChange 如果项移动，更新箭头连接的位置

Private 属性

- double `inputNum` = 3.0
- double `outputNum` = 0
- `ItemType` `myItemType`
- `QPolygonF` `myPolygon`
- `QMenu *` `myContextMenu`
- `QList< Arrow * >` `arrows`

6.3.1 详细描述

在文件 `items.h` 第 46 行定义.

6.3.2 成员枚举类型说明

6.3.2.1 anonymous enum

anonymous enum

枚举值

Type	
------	--

在文件 `items.h` 第 50 行定义.

6.3.2.2 ItemType

enum `Items::ItemType`

枚举值

test	
------	--

在文件 `items.h` 第 51 行定义.

6.3.3 构造及析构函数说明

6.3.3.1 Items()

```
Items::Items (
    ItemType itemType,
    QMenu * contextMenu,
    QGraphicsItem * parent = 0 )
```

在文件 items.cpp 第 34 行定义.

函数调用图:



6.3.4 成员函数说明

6.3.4.1 addArrow()

```
void Items::addArrow (
    Arrow * arrow )
```

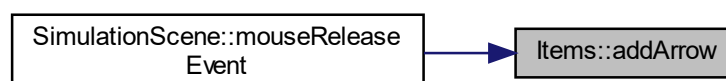
`Items::addArrow` 将箭头添加到箭头列表

参数

<i>arrow</i>	要添加的箭头Arrow
--------------	-------------

在文件 items.cpp 第 89 行定义.

这是这个函数的调用关系图:



6.3.4.2 calculateResult()

```
double Items::calculateResult (
    ItemType myItemType,
    double inputNum )
```

Items::calculateResult 根据不同的项进行不同的计算

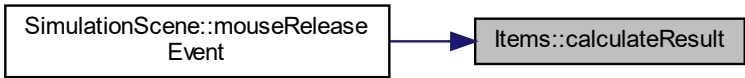
参数

<i>myItemType</i>	ItemType
<i>inputNum</i>	输入值double

返回

在文件 items.cpp 第 145 行定义.

这是这个函数的调用关系图:



6.3.4.3 contextMenuEvent()

```
void Items::contextMenuEvent (
    QGraphicsSceneContextMenuEvent * event ) [override], [protected]
```

Items::contextMenuEvent 鼠标右键时显示菜单，默认情况下不选择项目

参数

<i>event</i>	QGraphicsSceneContextMenuEvent
--------------	--------------------------------

在文件 items.cpp 第 114 行定义.

6.3.4.4 getInputNum()

```
double Items::getInputNum ( )
```

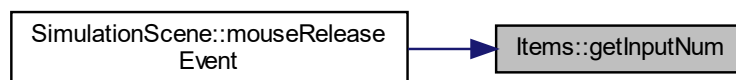

`Items::getInputNum` 输出该Item的inputNum

返回

该Item的inputNum

在文件 `items.cpp` 第 180 行定义.

这是这个函数的调用关系图:



6.3.4.5 getItemType()

`Items::ItemType` `Items::getItemType ()`

`Items::getItemType` 获取对应项的种类

返回

返回枚举类型的对应项种类

在文件 `items.cpp` 第 189 行定义.

函数调用图:



6.3.4.6 getOutputNum()

```
double Items::getOutputNum ( )
```

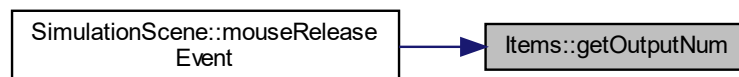
`Items::getOutputNum` 输出该Item的OutputNum

返回

该Item的OutputNum

在文件 `items.cpp` 第 171 行定义.

这是这个函数的调用关系图:



6.3.4.7 image()

```
QPixmap Items::image ( ) const
```

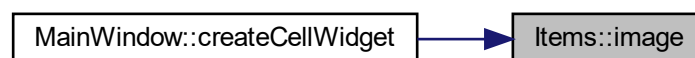
`Items::image` 为工具箱中的工具按钮创建图标

返回

QPixmap

在文件 `items.cpp` 第 98 行定义.

这是这个函数的调用关系图:



6.3.4.8 itemChange()

```
QVariant Items::itemChange (
    GraphicsItemChange change,
    const QVariant & value ) [override], [protected]
```

`Items::itemChange` 如果项移动, 更新箭头连接的位置

参数

<i>change</i>	项移动事件GraphicsItemChange
<i>value</i>	QVariant

返回

箭头连接线更新结果

在文件 items.cpp 第 127 行定义.

函数调用图:

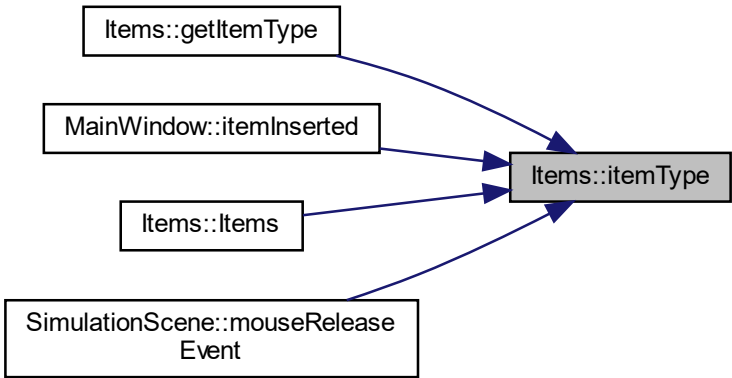


6.3.4.9 itemType()

```
ItemType Items::itemType ( ) const [inline]
```

在文件 items.h 第 57 行定义.

这是这个函数的调用关系图:



6.3.4.10 polygon()

```
QPolygonF Items::polygon ( ) const [inline]
```

在文件 items.h 第 58 行定义.

这是这个函数的调用关系图:



6.3.4.11 removeArrow()

```
void Items::removeArrow (
    Arrow * arrow )
```

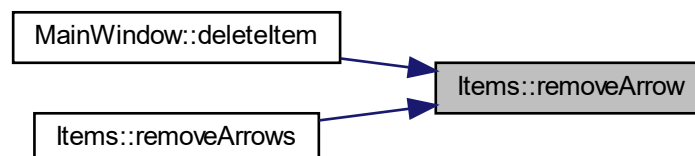
`Items::removeArrow` 当与连接线相关的项被删除时，删除该连接线

参数

<code>arrow</code>	<code>Arrow</code>
--------------------	--------------------

在文件 items.cpp 第 64 行定义.

这是这个函数的调用关系图:



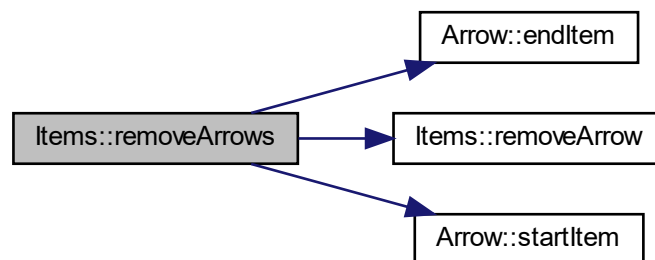
6.3.4.12 removeArrows()

```
void Items::removeArrows ( )
```

[Items::removeArrows](#) 当一个项被删除时，删除与这个项有关的所有连接线

在文件 items.cpp 第 74 行定义.

函数调用图:



6.3.4.13 setInputNum()

```
void Items::setInputNum (
    double inputNum )
```

[Items::setInputNum](#) 设置该Item的InputNum

参数

<i>inputNum</i>	double类型
-----------------	----------

在文件 items.cpp 第 162 行定义.

6.3.4.14 type()

```
int Items::type ( ) const [inline], [override]
```

在文件 items.h 第 61 行定义.

6.3.5 结构体成员变量说明

6.3.5.1 arrows

```
QList<Arrow *> Items::arrows [private]
```

在文件 items.h 第 81 行定义.

6.3.5.2 inputNum

```
double Items::inputNum = 3.0 [private]
```

在文件 items.h 第 76 行定义.

6.3.5.3 myContextMenu

```
QMenu* Items::myContextMenu [private]
```

在文件 items.h 第 80 行定义.

6.3.5.4 myItemType

```
ItemType Items::myItemType [private]
```

在文件 items.h 第 78 行定义.

6.3.5.5 myPolygon

```
QPolygonF Items::myPolygon [private]
```

在文件 items.h 第 79 行定义.

6.3.5.6 outputNum

```
double Items::outputNum = 0 [private]
```

在文件 `items.h` 第 77 行定义.

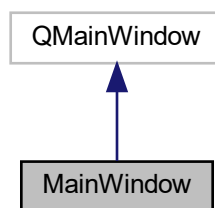
该类的文档由以下文件生成:

- [C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/items.h](#)
- [C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/items.cpp](#)

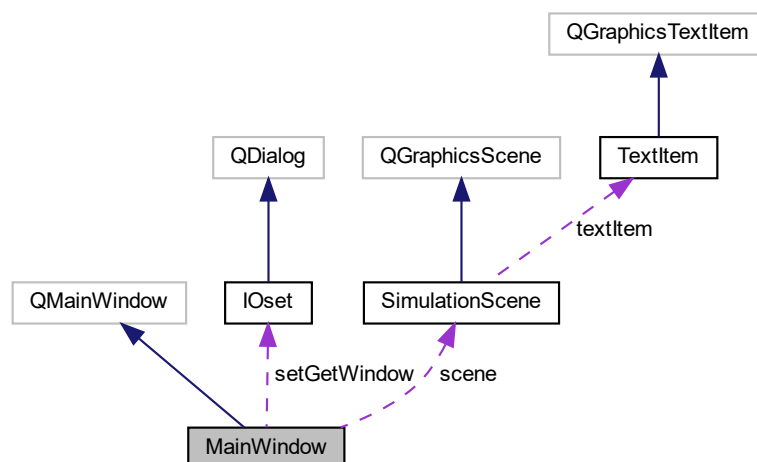
6.4 MainWindow类 参考

```
#include <mainwindow.h>
```

类 `MainWindow` 继承关系图:



`MainWindow` 的协作图:



信号

- void `sendOutputValue` (double outputNum)

Public 成员函数

- `MainWindow` ()

Private 槽

- void `buttonGroupClicked` (int id)
MainWindow::buttonGroupClicked 根据不同的图标切换不同的运行模式
- void `deleteItem` ()
MainWindow::deleteItem() 删除对应的项
- void `pointerGroupClicked` (int id)
MainWindow::pointerGroupClicked 决定场景是否是 *ItemMove*或*InsertLine*模式
- void `bringToFront` ()
MainWindow::bringToFront() 将页面下方的项移到表面
- void `sendToBack` ()
MainWindow::sendToBack() 将页面表面的项移到下面
- void `itemInserted` (Items *item)
MainWindow::itemInserted 插入新的项
- void `textInserted` (QGraphicsTextItem *item)
MainWindow::textInserted 插入文本框
- void `currentFontChanged` (const QFont &font)
MainWindow::currentFontChanged 修改文本字体
- void `fontSizeChanged` (const QString &size)
MainWindow::fontSizeChanged 修改文本字体大小-绑定事件
- void `sceneScaleChanged` (const QString &scale)
MainWindow::sceneScaleChanged 更改界面显示比例
- void `handleFontChange` ()
MainWindow::handleFontChange 修改字体形态
- void `itemSelected` (QGraphicsItem *item)
MainWindow::itemSelected 选择指定项
- void `about` ()
MainWindow::about 简介页面
- void `setGetValueWindow` ()
MainWindow::setGetValueWindow 唤醒子窗口
- void `getInputValue` (double inputNum)
MainWindow::getInputValue 获得输入值并根据项的种类更新输出值

Private 成员函数

- void `createToolBox` ()
[`MainWindow::createToolBox`](#) 创建工具箱
- void `createActions` ()
[`MainWindow::createActions`](#) 创建顶部操作栏
- void `createMenus` ()
[`MainWindow::createMenus`](#) 创建菜单栏
- void `createToolbars` ()
[`MainWindow::createToolbars`](#) 创建工作台
- QWidget * `createBackgroundCellWidget` (const QString &text, const QString &image)
- QWidget * `createCellWidget` (const QString &text, [`Items::ItemType`](#) type)
[`MainWindow::createCellWidget`](#) 创建单元部件
- QMenu * `createColorMenu` (const char *slot, QColor defaultColor)
- QIcon `createColorToolButtonIcon` (const QString &image, QColor color)
- QIcon `createColorIcon` (QColor color)

Private 属性

- [`SimulationScene`](#) * `scene`
- QGraphicsView * `view`
- QAction * `exitAction`
- QAction * `addAction`
- QAction * `deleteAction`
- QAction * `setGetAction`
- QAction * `toFrontAction`
- QAction * `sendBackAction`
- QAction * `aboutAction`
- QAction * `boldAction`
- QAction * `underlineAction`
- QAction * `italicAction`
- QAction * `textAction`
- QAction * `fillAction`
- QAction * `lineAction`
- QMenu * `fileMenu`
- QMenu * `itemMenu`
- QMenu * `AboutMenu`
- QToolBar * `textToolBar`
- QToolBar * `editToolBar`
- QToolBar * `colorToolBar`
- QToolBar * `pointerToolBar`
- QComboBox * `sceneScaleCombo`
- QComboBox * `fontSizeCombo`
- QFontComboBox * `fontCombo`
- QComboBox * `itemColorCombo`
- QComboBox * `textColorCombo`
- QToolBox * `toolBox`
- QButtonGroup * `buttonGroup`
- QButtonGroup * `pointerTypeGroup`
- IOset * `setGetWindow`

6.4.1 详细描述

在文件 `mainwindow.h` 第 48 行定义.

6.4.2 构造及析构函数说明

6.4.2.1 MainWindow()

```
MainWindow::MainWindow ( )
```

在文件 `mainwindow.cpp` 第 39 行定义.

6.4.3 成员函数说明

6.4.3.1 about

```
void MainWindow::about ( ) [private], [slot]
```

[MainWindow::about](#) 简介页面

在文件 `mainwindow.cpp` 第 256 行定义.

6.4.3.2 bringToFront

```
void MainWindow::bringToFront ( ) [private], [slot]
```

[MainWindow::bringToFront\(\)](#) 将页面下方的项移到表面

在文件 `mainwindow.cpp` 第 131 行定义.

6.4.3.3 buttonGroupClicked

```
void MainWindow::buttonGroupClicked (
    int id ) [private], [slot]
```

[MainWindow::buttonGroupClicked](#) 根据不同的图标切换不同的运行模式

参数

<i>id</i>	点击的按键对应的id
-----------	------------

在文件 `mainwindow.cpp` 第 74 行定义.

6.4.3.4 createActions()

```
void MainWindow::createActions ( ) [private]
```

[MainWindow::createActions](#) 创建顶部操作栏

在文件 `mainwindow.cpp` 第 306 行定义.

6.4.3.5 createBackgroundCellWidget()

```
QWidget* MainWindow::createBackgroundCellWidget (
    const QString & text,
    const QString & image ) [private]
```

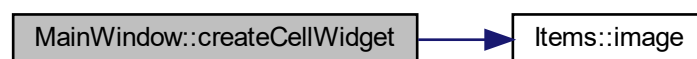
6.4.3.6 createCellWidget()

```
QWidget * MainWindow::createCellWidget (
    const QString & text,
    Items::ItemType type ) [private]
```

[MainWindow::createCellWidget](#) 创建单元部件

在文件 `mainwindow.cpp` 第 441 行定义.

函数调用图:



6.4.3.7 createColorIcon()

```
QIcon MainWindow::createColorIcon (
    QColor color ) [private]
```

6.4.3.8 createColorMenu()

```
QMenu* MainWindow::createColorMenu (
    const char * slot,
    QColor defaultColor ) [private]
```

6.4.3.9 createColorToolButtonIcon()

```
QIcon MainWindow::createColorToolButtonIcon (
    const QString & image,
    QColor color ) [private]
```

6.4.3.10 createMenus()

```
void MainWindow::createMenus ( ) [private]
```

[MainWindow::createMenus](#) 创建菜单栏

在文件 `mainwindow.cpp` 第 361 行定义.

6.4.3.11 createToolbars()

```
void MainWindow::createToolbars ( ) [private]
```

[MainWindow::createToolbars](#) 创建工作台

在文件 `mainwindow.cpp` 第 380 行定义.

6.4.3.12 createToolBox()

```
void MainWindow::createToolBox ( ) [private]
```

[MainWindow::createToolBox](#) 创建工具箱

在文件 `mainwindow.cpp` 第 264 行定义.

6.4.3.13 currentFontChanged

```
void MainWindow::currentFontChanged (
    const QFont & font ) [private], [slot]
```

[MainWindow::currentFontChanged](#) 修改文本字体

参数

<code>QFont</code>	Qt控件文字属性
--------------------	----------

在文件 `mainwindow.cpp` 第 196 行定义.

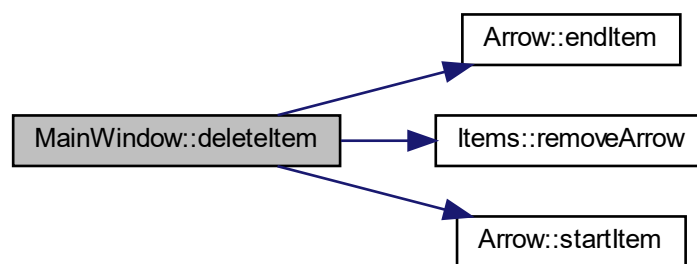
6.4.3.14 deleteItem

```
void MainWindow::deleteItem ( ) [private], [slot]
```

[MainWindow::deleteItem\(\)](#) 删除对应的项

在文件 `mainwindow.cpp` 第 97 行定义.

函数调用图:



6.4.3.15 fontSizeChanged

```
void MainWindow::fontSizeChanged (
    const QString & size ) [private], [slot]
```

[MainWindow::fontSizeChanged](#) 修改文本字体大小-绑定事件

参数

<code>QString</code>	字符串类型, 表示大小
----------------------	-------------

在文件 `mainwindow.cpp` 第 205 行定义.

6.4.3.16 getInputValue

```
void MainWindow::getInputValue (
    double inputNum ) [private], [slot]
```

[MainWindow::getInputValue](#) 获得输入值并根据项的种类更新输出值

参数

<i>inputNum</i>	子窗口设定的输入值
-----------------	-----------

在文件 `mainwindow.cpp` 第 490 行定义.

6.4.3.17 handleFontChange

```
void MainWindow::handleFontChange ( ) [private], [slot]
```

[MainWindow::handleFontChange](#) 修改字体形态

在文件 `mainwindow.cpp` 第 226 行定义.

6.4.3.18 itemInserted

```
void MainWindow::itemInserted (
    Items * item ) [private], [slot]
```

[MainWindow::itemInserted](#) 插入新的项

参数

<i>item</i>	所需要插入的项
-------------	---------

在文件 `mainwindow.cpp` 第 175 行定义.

函数调用图:



6.4.3.19 itemSelected

```
void MainWindow::itemSelected (
    QGraphicsItem * item ) [private], [slot]
```

[MainWindow::itemSelected](#) 选择指定项

参数

<i>item</i>	QGraphicsItem类型，表示所选择的项
-------------	-------------------------

在文件 `mainwindow.cpp` 第 242 行定义.

6.4.3.20 pointerGroupClicked

```
void MainWindow::pointerGroupClicked (
    int id ) [private], [slot]
```

[MainWindow::pointerGroupClicked](#) 决定场景是否是ItemMove或InsertLine模式

在文件 `mainwindow.cpp` 第 123 行定义.

6.4.3.21 sceneScaleChanged

```
void MainWindow::sceneScaleChanged (
    const QString & scale ) [private], [slot]
```

[MainWindow::sceneScaleChanged](#) 更改界面显示比例

参数

<i>scale</i>	表征页面显示比例的字符串
--------------	--------------

在文件 `mainwindow.cpp` 第 214 行定义.

6.4.3.22 sendOutputValue

```
void MainWindow::sendOutputValue (
    double outputNum ) [signal]
```

6.4.3.23 sendToBack

```
void MainWindow::sendToBack ( ) [private], [slot]
```

[MainWindow::sendToBack\(\)](#) 将页面表面的项移到下面

在文件 `mainwindow.cpp` 第 152 行定义.

6.4.3.24 setGetValueWindow

```
void MainWindow::setGetValueWindow ( ) [private], [slot]
```

[MainWindow::setGetValueWindow](#) 唤醒子窗口

在文件 `mainwindow.cpp` 第 466 行定义.

6.4.3.25 textInserted

```
void MainWindow::textInserted (
    QGraphicsTextItem * item ) [private], [slot]
```

[MainWindow::textInserted](#) 插入文本框

参数

<code>QGraphicsTextItem</code>	格式化的文本项
--------------------------------	---------

在文件 `mainwindow.cpp` 第 186 行定义.

6.4.4 结构体成员变量说明

6.4.4.1 aboutAction

```
QAction* MainWindow::aboutAction [private]
```

在文件 `mainwindow.h` 第 99 行定义.

6.4.4.2 AboutMenu

```
QMenu* MainWindow::AboutMenu [private]
```

在文件 mainwindow.h 第 111 行定义.

6.4.4.3 addAction

```
QAction* MainWindow::addAction [private]
```

在文件 mainwindow.h 第 93 行定义.

6.4.4.4 boldAction

```
QAction* MainWindow::boldAction [private]
```

在文件 mainwindow.h 第 101 行定义.

6.4.4.5 buttonGroup

```
QButtonGroup* MainWindow::buttonGroup [private]
```

在文件 mainwindow.h 第 127 行定义.

6.4.4.6 colorToolBar

```
QToolBar* MainWindow::colorToolBar [private]
```

在文件 mainwindow.h 第 116 行定义.

6.4.4.7 deleteAction

```
QAction* MainWindow::deleteAction [private]
```

在文件 mainwindow.h 第 94 行定义.

6.4.4.8 editToolBar

```
QToolBar* MainWindow::editToolBar [private]
```

在文件 mainwindow.h 第 115 行定义.

6.4.4.9 exitAction

```
QAction* MainWindow::exitAction [private]
```

在文件 mainwindow.h 第 92 行定义.

6.4.4.10 fileMenu

```
QMenu* MainWindow::fileMenu [private]
```

在文件 mainwindow.h 第 109 行定义.

6.4.4.11 fillAction

```
QAction* MainWindow::fillAction [private]
```

在文件 mainwindow.h 第 105 行定义.

6.4.4.12 fontCombo

```
QFontComboBox* MainWindow::fontCombo [private]
```

在文件 mainwindow.h 第 122 行定义.

6.4.4.13 fontSizeCombo

```
QComboBox* MainWindow::fontSizeCombo [private]
```

在文件 mainwindow.h 第 121 行定义.

6.4.4.14 italicAction

```
QAction* MainWindow::italicAction [private]
```

在文件 mainwindow.h 第 103 行定义.

6.4.4.15 itemColorCombo

```
QComboBox* MainWindow::itemColorCombo [private]
```

在文件 mainwindow.h 第 123 行定义.

6.4.4.16 itemMenu

```
QMenu* MainWindow::itemMenu [private]
```

在文件 mainwindow.h 第 110 行定义.

6.4.4.17 lineAction

```
QAction* MainWindow::lineAction [private]
```

在文件 mainwindow.h 第 106 行定义.

6.4.4.18 pointerToolBar

```
QToolBar* MainWindow::pointerToolBar [private]
```

在文件 mainwindow.h 第 117 行定义.

6.4.4.19 pointerTypeGroup

```
QButtonGroup* MainWindow::pointerTypeGroup [private]
```

在文件 mainwindow.h 第 128 行定义.

6.4.4.20 scene

```
SimulationScene* MainWindow::scene [private]
```

在文件 mainwindow.h 第 88 行定义.

6.4.4.21 sceneScaleCombo

```
QComboBox* MainWindow::sceneScaleCombo [private]
```

在文件 mainwindow.h 第 120 行定义.

6.4.4.22 sendBackAction

```
QAction* MainWindow::sendBackAction [private]
```

在文件 mainwindow.h 第 98 行定义.

6.4.4.23 setGetAction

```
QAction* MainWindow::setGetAction [private]
```

在文件 mainwindow.h 第 95 行定义.

6.4.4.24 setGetWindow

```
IOset* MainWindow::setGetWindow [private]
```

在文件 mainwindow.h 第 131 行定义.

6.4.4.25 textAction

```
QAction* MainWindow::textAction [private]
```

在文件 mainwindow.h 第 104 行定义.

6.4.4.26 textColorCombo

```
QComboBox* MainWindow::textColorCombo [private]
```

在文件 mainwindow.h 第 124 行定义.

6.4.4.27 textToolBar

```
QToolBar* MainWindow::textToolBar [private]
```

在文件 mainwindow.h 第 114 行定义.

6.4.4.28 toFrontAction

```
QAction* MainWindow::toFrontAction [private]
```

在文件 mainwindow.h 第 97 行定义.

6.4.4.29 toolBox

```
QToolBox* MainWindow::toolBox [private]
```

在文件 mainwindow.h 第 126 行定义.

6.4.4.30 underlineAction

```
QAction* MainWindow::underlineAction [private]
```

在文件 mainwindow.h 第 102 行定义.

6.4.4.31 view

```
QGraphicsView* MainWindow::view [private]
```

在文件 mainwindow.h 第 89 行定义.

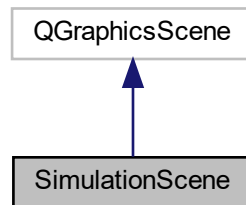
该类的文档由以下文件生成:

- C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/[mainwindow.h](#)
- C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/[mainwindow.cpp](#)

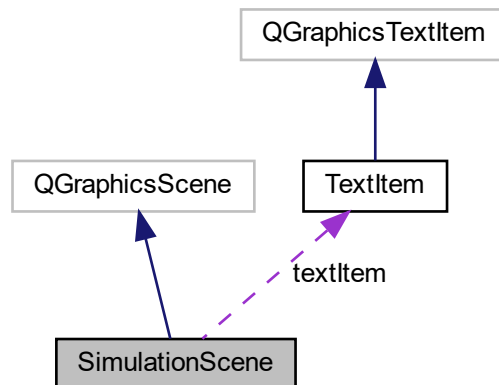
6.5 SimulationScene类 参考

```
#include <simulationScene.h>
```

类 SimulationScene 继承关系图:



SimulationScene 的协作图:



Public 类型

- enum `Mode` { `insertItem` , `insertLine` , `insertText` , `moveItem` }

Public 槽

- void `setMode` (`SimulationScene::Mode` mode)
`SimulationScene::setMode` 设置操作模式
- void `setItemType` (`Items::ItemType` type)
`SimulationScene::setItemType` 设置可选项的种类
- void `editorLostFocus` (`TextItem` *item)
`SimulationScene::editorLostFocus` 文本框失去焦点时, 如果项目没有文本, 则删除它

信号

- void `itemInserted` (`Items *item`)
- void `textInserted` (`QGraphicsTextItem *item`)
- void `itemSelected` (`QGraphicsItem *item`)

Public 成员函数

- `SimulationScene` (`QMenu *itemMenu`, `QObject *parent=0`)
- `QFont font` () const
- void `setFont` (const `QFont &font`)

SimulationScene::setFont 设置字体

Protected 成员函数

- void `mousePressEvent` (`QGraphicsSceneMouseEvent *mouseEvent`) override
SimulationScene::mousePressEvent 处理鼠标点击事件
- void `mouseMoveEvent` (`QGraphicsSceneMouseEvent *mouseEvent`) override
SimulationScene::mouseMoveEvent 调用 *QGraphicsScene::mouseMoveEvent* 处理项的移动
- void `mouseReleaseEvent` (`QGraphicsSceneMouseEvent *mouseEvent`) override
SimulationScene::mouseReleaseEvent 处理鼠标松开后的事件

Private 成员函数

- bool `isItemChange` (int type)
SimulationScene::isItemChange 检查所选项是否存在并且是否是指定类型

Private 属性

- `Items::ItemType myItemType`
- `QMenu * myItemMenu`
- `Mode myMode`
- bool `leftButtonDown`
- `QPointF startPoint`
- `QGraphicsLineItem * line`
- `QFont myFont`
- `QColor myItemColor`
- `QColor myTextColor`
- `QColor myLineColor`
- `TextItem * textItem`

6.5.1 详细描述

在文件 `simulationScene.h` 第 36 行定义.

6.5.2 成员枚举类型说明

6.5.2.1 Mode

```
enum SimulationScene::Mode
```

枚举值

insertItem	
insertLine	
insertText	
moveItem	

在文件 `simulationScene.h` 第 41 行定义.

6.5.3 构造及析构造函数说明

6.5.3.1 SimulationScene()

```
SimulationScene::SimulationScene (
    QMenu * itemMenu,
    QObject * parent = 0 ) [explicit]
```

在文件 `simulationScene.cpp` 第 29 行定义.

6.5.4 成员函数说明

6.5.4.1 editorLostFocus

```
void SimulationScene::editorLostFocus (
    QTableWidgetItem * item ) [slot]
```

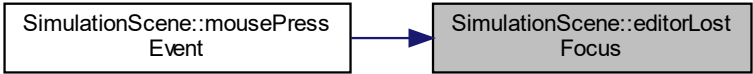
`SimulationScene::editorLostFocus` 文本框失去焦点时，如果项目没有文本，则删除它

参数

<i>item</i>	<code>QTableWidgetItem</code>
-------------	-------------------------------

在文件 `simulationScene.cpp` 第 79 行定义.

这是这个函数的调用关系图:

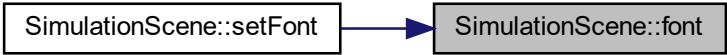


6.5.4.2 font()

```
QFont SimulationScene::font ( ) const [inline]
```

在文件 `simulationScene.h` 第 44 行定义.

这是这个函数的调用关系图:



6.5.4.3 isItemChange()

```
bool SimulationScene::isItemChange (
    int type ) [private]
```

`SimulationScene::isItemChange` 检查所选项是否存在并且是否是指定类型

参数

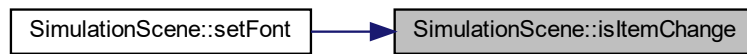
<i>type</i>	int类型，与enum中的列表对应
-------------	-------------------

返回

若对应，返回true；否则返回false

在文件 `simulationScene.cpp` 第 199 行定义.

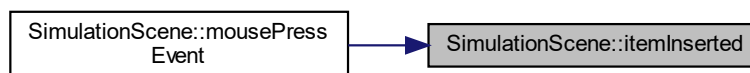
这是这个函数的调用关系图:



6.5.4.4 itemInserted

```
void SimulationScene::itemInserted (  
    Items * item ) [signal]
```

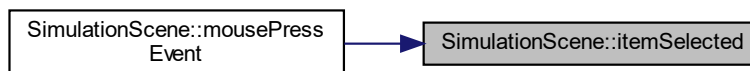
这是这个函数的调用关系图:



6.5.4.5 itemSelected

```
void SimulationScene::itemSelected (  
    QGraphicsItem * item ) [signal]
```

这是这个函数的调用关系图:



6.5.4.6 mouseMoveEvent()

```
void SimulationScene::mouseMoveEvent (  
    QGraphicsSceneMouseEvent * mouseEvent ) [override], [protected]
```

[SimulationScene::mouseMoveEvent](#) 调用 [QGraphicsScene::mouseMoveEvent](#) 处理项的移动

参数

<i>mouseEvent</i>	QGraphicsSceneMouseEvent
-------------------	--------------------------

在文件 simulationScene.cpp 第 142 行定义.

6.5.4.7 mousePressEvent()

```
void SimulationScene::mousePressEvent (
    QGraphicsSceneMouseEvent * mouseEvent ) [override], [protected]
```

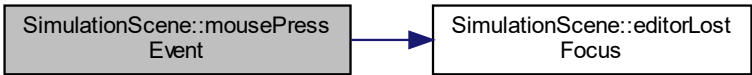
[SimulationScene::mousePressEvent](#) 处理鼠标点击事件

参数

<i>mouseEvent</i>	QGraphicsSceneMouseEvent
-------------------	--------------------------

在文件 simulationScene.cpp 第 95 行定义.

函数调用图:



6.5.4.8 mouseReleaseEvent()

```
void SimulationScene::mouseReleaseEvent (
    QGraphicsSceneMouseEvent * mouseEvent ) [override], [protected]
```

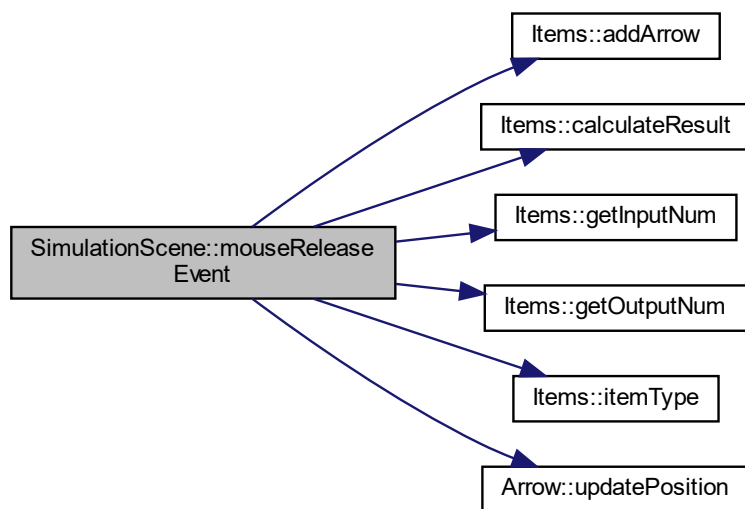
[SimulationScene::mouseReleaseEvent](#) 处理鼠标松开后的事件

参数

<i>mouseEvent</i>	QGraphicsSceneMouseEvent
-------------------	--------------------------

在文件 simulationScene.cpp 第 157 行定义.

函数调用图:



6.5.4.9 setFont()

```
void SimulationScene::setFont (
    const QFont & font )
```

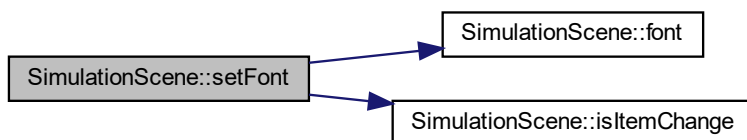
[SimulationScene::setFont](#) 设置字体

参数

<i>font</i>	QFont字体
-------------	---------

在文件 `simulationScene.cpp` 第 46 行定义.

函数调用图:



6.5.4.10 setItemType

```
void SimulationScene::setItemType (
    Items::ItemType type ) [slot]
```

[SimulationScene::setItemType](#) 设置可选项的种类

参数

<i>type</i>	项的种类ItemType
-------------	--------------

在文件 `simulationScene.cpp` 第 70 行定义.

6.5.4.11 setMode

```
void SimulationScene::setMode (
    SimulationScene::Mode mode ) [slot]
```

[SimulationScene::setMode](#) 设置操作模式

参数

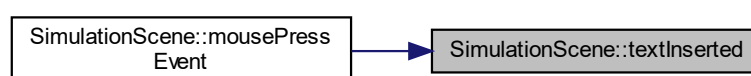
<i>mode</i>	Mode操作模式类型
-------------	------------

在文件 `simulationScene.cpp` 第 61 行定义.

6.5.4.12 textInserted

```
void SimulationScene::textInserted (
    QGraphicsTextItem * item ) [signal]
```

这是这个函数的调用关系图:



6.5.5 结构体成员变量说明

6.5.5.1 leftButtonDown

```
bool SimulationScene::leftButtonDown [private]
```

在文件 `simulationScene.h` 第 68 行定义.

6.5.5.2 line

```
QGraphicsLineItem* SimulationScene::line [private]
```

在文件 `simulationScene.h` 第 70 行定义.

6.5.5.3 myFont

```
QFont SimulationScene::myFont [private]
```

在文件 `simulationScene.h` 第 71 行定义.

6.5.5.4 myItemColor

```
QColor SimulationScene::myItemColor [private]
```

在文件 `simulationScene.h` 第 72 行定义.

6.5.5.5 myItemMenu

```
QMenu* SimulationScene::myItemMenu [private]
```

在文件 `simulationScene.h` 第 66 行定义.

6.5.5.6 myItemType

```
Items::ItemType SimulationScene::myItemType [private]
```

在文件 simulationScene.h 第 65 行定义.

6.5.5.7 myLineColor

```
QColor SimulationScene::myLineColor [private]
```

在文件 simulationScene.h 第 74 行定义.

6.5.5.8 myMode

```
Mode SimulationScene::myMode [private]
```

在文件 simulationScene.h 第 67 行定义.

6.5.5.9 myTextColor

```
QColor SimulationScene::myTextColor [private]
```

在文件 simulationScene.h 第 73 行定义.

6.5.5.10 startPoint

```
QPointF SimulationScene::startPoint [private]
```

在文件 simulationScene.h 第 69 行定义.

6.5.5.11 textItem

```
TextItem* SimulationScene::textItem [private]
```

在文件 simulationScene.h 第 75 行定义.

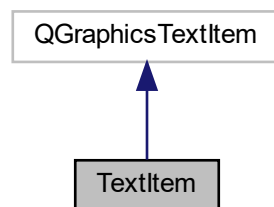
该类的文档由以下文件生成:

- C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/[simulationScene.h](#)
- C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/[simulationScene.cpp](#)

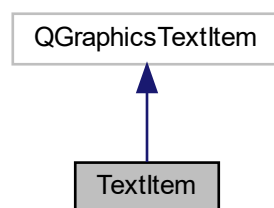
6.6 TextItem类 参考

```
#include <textItem.h>
```

类 TextItem 继承关系图:



TextItem 的协作图:



Public 类型

- enum { **Type** = UserType + 3 }

信号

- void **lostFocus** (TextItem *item)
- void **selectedChange** (QGraphicsItem *item)

Public 成员函数

- **TextItem** (QGraphicsItem *parent=0)
- int **type** () const override

Protected 成员函数

- QVariant [itemChange](#) (GraphicsItemChange change, const QVariant &value) override
[TextItem::itemChange](#) 当项目被选中时发出[selectedChanged](#)信号。主窗口使用这个信号来更新小部件，这些小部件将字体属性显示为选定文本项的字体
- void [focusOutEvent](#) (QFocusEvent *event) override
[TextItem::focusOutEvent](#) 若文本框为空，则在失去文本项焦点的时候删除该文本项
- void [mouseDoubleClickEvent](#) (QGraphicsSceneMouseEvent *event) override
[TextItem::mouseDoubleClickEvent](#) 当发生双击事件后，调用[QGraphicsTextItem](#)使项可编辑

6.6.1 详细描述

在文件 `textItem.h` 第 34 行定义.

6.6.2 成员枚举类型说明

6.6.2.1 anonymous enum

anonymous enum

枚举值

Type	
------	--

在文件 `textItem.h` 第 39 行定义.

6.6.3 构造及析构函数说明

6.6.3.1 TextItem()

```
TextItem::TextItem (
    QGraphicsItem * parent = 0 )
```

在文件 `textItem.cpp` 第 27 行定义.

6.6.4 成员函数说明

6.6.4.1 focusOutEvent()

```
void TextItem::focusOutEvent (
    QFocusEvent * event ) [override], [protected]
```

[TextItem::focusOutEvent](#) 若文本框为空，则在失去文本项焦点的时候删除该文本项

参数

<i>event</i>	QFocusEvent
--------------	-------------

在文件 `textItem.cpp` 第 52 行定义.

6.6.4.2 itemChange()

```
QVariant TextItem::itemChange (  
    GraphicsItemChange change,  
    const QVariant & value ) [override], [protected]
```

`TextItem::itemChange` 当项目被选中时发出selectedChanged信号。主窗口使用这个信号来更新小部件，这些小部件将字体属性显示为选定文本项的字体

参数

<i>change</i>	GraphicsItemChange
<i>value</i>	QVariant

返回

selectedChanged信号

在文件 `textItem.cpp` 第 41 行定义.

6.6.4.3 lostFocus

```
void TextItem::lostFocus (  
    TextItem * item ) [signal]
```

这是这个函数的调用关系图:



6.6.4.4 mouseDoubleClickEvent()

```
void TextItem::mouseDoubleClickEvent (
    QGraphicsSceneMouseEvent * event ) [override], [protected]
```

[TextItem::mouseDoubleClickEvent](#) 当发生双击事件后，调用QGraphicsTextItem使项可编辑

参数

<i>event</i>	鼠标双击事件
--------------	--------

在文件 `textItem.cpp` 第 63 行定义.

6.6.4.5 selectedChange

```
void TextItem::selectedChange (
    QGraphicsItem * item ) [signal]
```

这是这个函数的调用关系图:



6.6.4.6 type()

```
int TextItem::type ( ) const [inline], [override]
```

在文件 `textItem.h` 第 43 行定义.

该类的文档由以下文件生成:

- C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/[textItem.h](#)
- C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/[textItem.cpp](#)

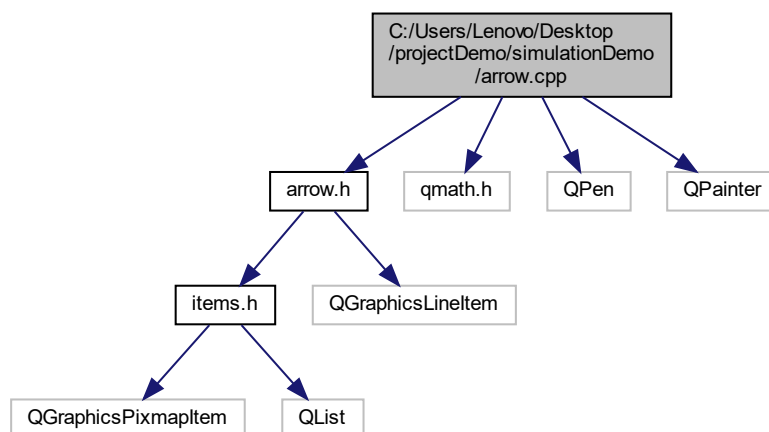
Chapter 7

文件说明

7.1 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/arrow.cpp 文件参考

连接线函数文件

```
#include "arrow.h"  
#include <qmath.h>  
#include <QPen>  
#include <QPainter>  
arrow.cpp 的引用(Include)关系图:
```



7.1.1 详细描述

连接线函数文件

主要包含两个项之间连接箭头的实现 **Arrow**类是一个连接两个**DiagramItems**的图形项。它在其中一个项目上画一个箭头。项目需要绘制自己，重新实现图形场景使用的方法，以检查碰撞和选择。该类继承**QGraphicsLine**项，并绘制箭头，并使用它所连接的项移动。

作者

LIN Guocheng

日期

2021-8-13

版本

1.0.0

注意

QT版本: 5.12.11

Date	Version	Author	Description
2021/08/06	0.0.1	LIN Guocheng	创建初始版本
2021/08/13	1.0.0	LIN Guocheng	完成第一代基础版本的适配

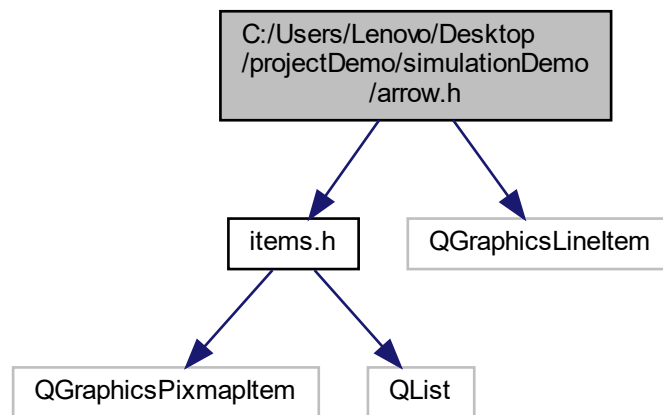
修改日志:

7.2 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/arrow.h 文件参考

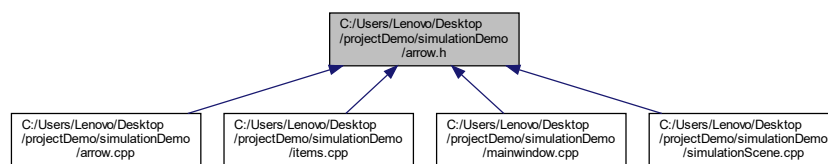
项目连接线函数头文件

```
#include "items.h"  
#include <QGraphicsLineItem>
```

arrow.h 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



结构体

- class [Arrow](#)

7.2.1 详细描述

项目连接线函数头文件

主要包含连接线功能的函数声明

作者

LIN Guocheng

日期

2021-8-13

版本

1.0.0

注意

QT版本: 5.12.11

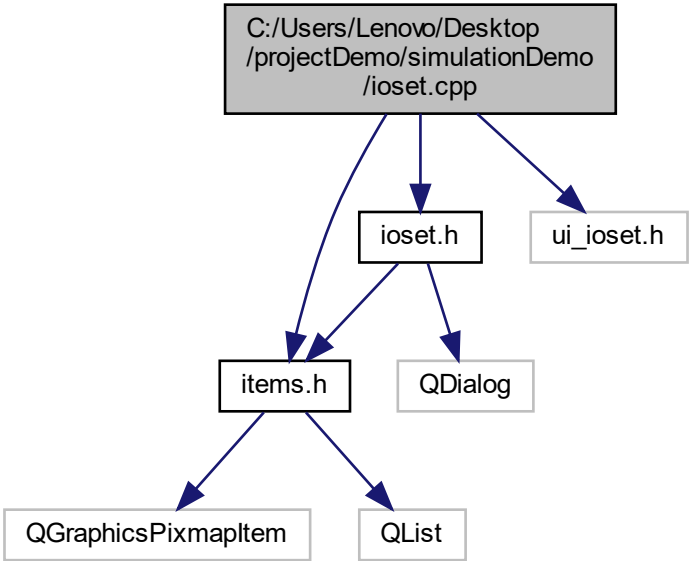
Date	Version	Author	Description
2021/08/06	0.0.1	LIN Guocheng	创建初始版本
2021/08/13	1.0.0	LIN Guocheng	完成第一代基础版本的适配

修改日志:

7.3 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/ioset.cpp 文件参考

设置对应项的输入及显示输出的功能文件

```
#include "ioset.h"
#include "items.h"
#include "ui_ioset.h"
ioset.cpp 的引用(Include)关系图:
```



7.3.1 详细描述

设置对应项的输入及显示输出的功能文件

主要包含连接线功能的函数实现

作者

LIN Guocheng

日期

2021-8-13

版本

1.0.0

注意

QT版本: 5.12.11

Date	Version	Author	Description
2021/08/06	0.0.1	LIN Guocheng	创建初始版本
2021/08/12	0.0.2	LIN Guocheng	增加与主页面交互的输入输出值窗口
2021/08/13	1.0.0	LIN Guocheng	完成第一代基础版本的适配

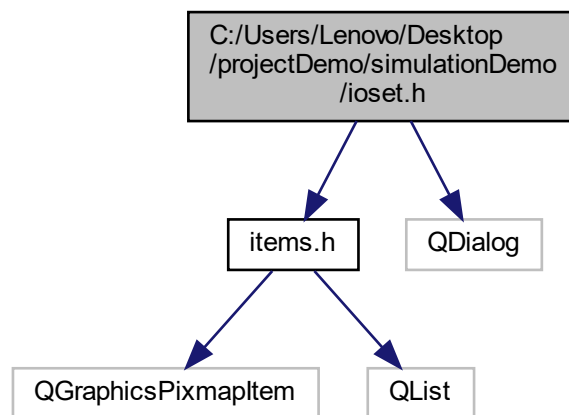
修改日志:

7.4 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/ioset.h 文件参考

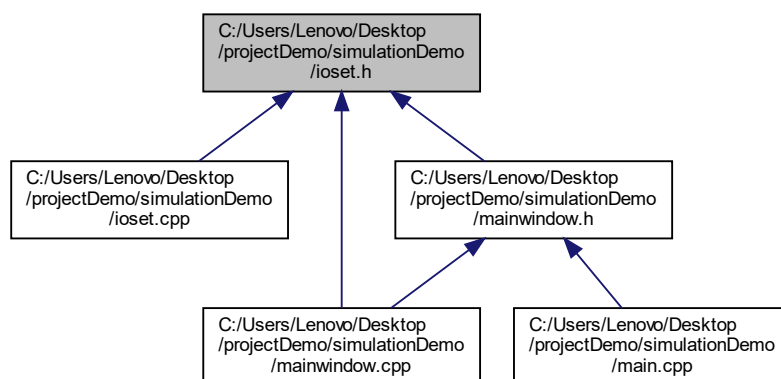
设置对应项的输入及显示输出的功能文件

```
#include "items.h"  
#include <QDialog>
```

ioset.h 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



结构体

- class **IOset**

命名空间

- **Ui**

7.4.1 详细描述

设置对应项的输入及显示输出的功能文件

主要包含连接线功能的函数声明

作者
LIN Guocheng

日期
2021-8-13

版本
1.0.0

注意
QT版本: 5.12.11

Date	Version	Author	Description
2021/08/06	0.0.1	LIN Guocheng	创建初始版本
2021/08/12	0.0.2	LIN Guocheng	增加与主页面交互的输入输出值窗口
2021/08/13	1.0.0	LIN Guocheng	完成第一代基础版本的适配

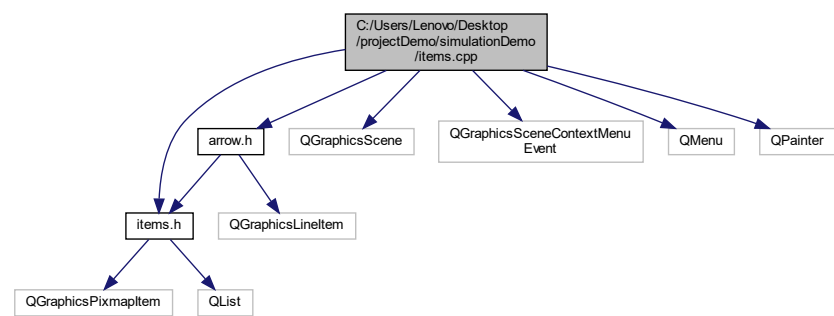
修改日志:

7.5 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/items.cpp
文件参考

功能项函数文件

```
#include "items.h"  
#include "arrow.h"  
#include <QGraphicsScene>  
#include <QGraphicsSceneContextMenuEvent>  
#include <QMenu>  
#include <QPainter>
```

items.cpp 的引用(Include)关系图:



7.5.1 详细描述

功能项函数文件

主要包含功能项类的实现 items表示SimulationScene中的功能项形状。它继承了QGraphicsPolygonItem，并为每个形状提供了一个多边形。枚举ItemType对每个流程图形状都有一个值。

作者

LIN Guocheng

日期

2021-8-13

版本

1.0.0

注意

QT版本: 5.12.11

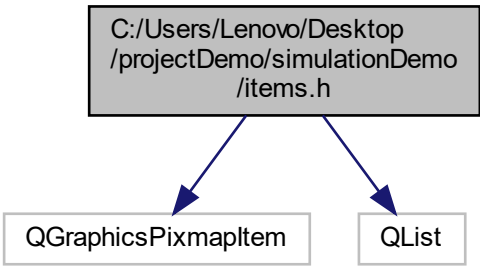
Date	Version	Author	Description
2021/08/06	0.0.1	LIN Guocheng	创建初始版本
2021/08/09	0.0.2	LIN Guocheng	增加器件计算函数
2021/08/10	0.0.3	LIN Guocheng	增加器件设置输入和显示输出函数
2021/08/12	0.0.4	LIN Guocheng	增加获取当前项种类的函数
2021/08/13	1.0.0	LIN Guocheng	完成第一代基础版本的适配

修改日志:

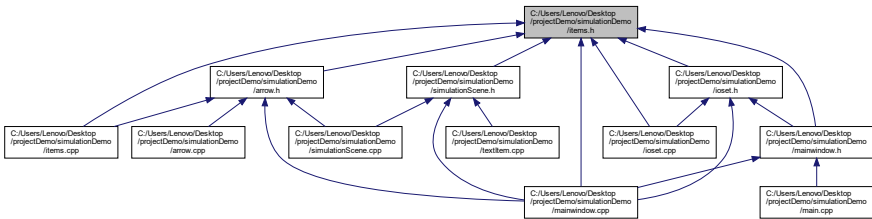
7.6 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/items.h 文件参考

项目功能项函数声明头文件

```
#include <QGraphicsPixmapItem>
#include <QList>
items.h 的引用(Include)关系图:
```



此图展示该文件直接或间接的被哪些文件引用了:



结构体

- class [Items](#)

7.6.1 详细描述

项目功能项函数声明头文件

主要包含功能项具体的函数声明

作者

LIN Guocheng

日期

2021-8-13

版本

1.0.0

注意

QT版本: 5.12.11

Date	Version	Author	Description
2021/08/06	0.0.1	LIN Guocheng	创建初始版本
2021/08/09	0.0.2	LIN Guocheng	增加器件计算函数
2021/08/10	0.0.3	LIN Guocheng	将器件输入输出值改为私有属性，并增加get/set函数
2021/08/12	0.0.4	LIN Guocheng	增加获取当前项种类的函数
2021/08/13	1.0.0	LIN Guocheng	完成第一代基础版本的适配

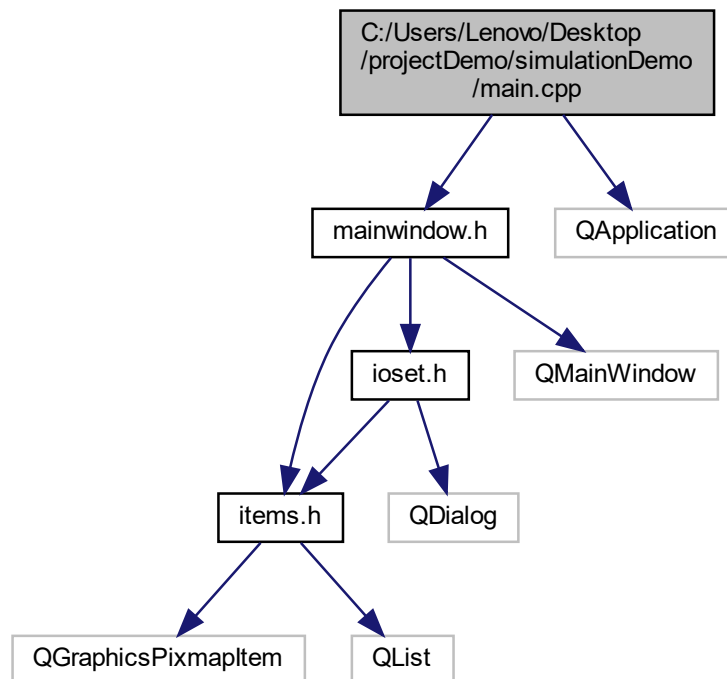
修改日志:

7.7 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/main.cpp 文件参考

QT 程序运行主函数

```
#include "mainwindow.h"  
#include <QApplication>
```

main.cpp 的引用(Include)关系图:



函数

- `int main (int argv, char *args[])`

7.7.1 详细描述

QT 程序运行主函数

主要包含主窗口界面的展示函数

作者

LIN Guocheng

日期

2021-8-13

版本

V0.0.1

注意

QT版本: 5.12.11

Date	Version	Author	Description
2021/08/06	0.0.1	LIN Guocheng	创建初始版本
2021/08/13	1.0.0	LIN Guocheng	完成第一代基础版本的适配

修改日志:

7.7.2 函数说明

7.7.2.1 main()

```
int main (
    int argv,
    char * args[] )
```

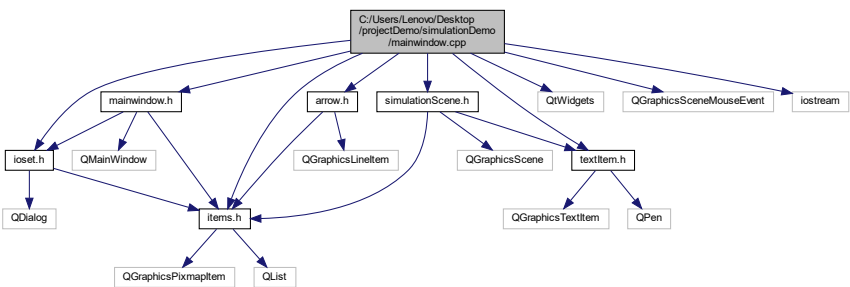
在文件 main.cpp 第 25 行定义.

7.8 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/mainwindow.cpp 文件参考

项目主函数文件

```
#include "mainwindow.h"
#include "arrow.h"
#include "items.h"
#include "simulationScene.h"
#include "textItem.h"
#include "ioset.h"
#include <QtWidgets>
#include <QGraphicsSceneMouseEvent>
#include <iostream>
```

mainwindow.cpp 的引用(Include)关系图:



变量

- `const int InsertTextButton = 10`

7.8.1 详细描述

项目主函数文件

主要包含主窗口界面的函数实现，QT 的 `MainWindow` 类，代表了展示给用户的 GUI 界面。`MainWindow` 类在 `QMainWindow` 中创建和布局小部件。类将输入从小部件转发到 `DiagramScene`。当图场景的文本项更改，或者一个图项或一个图文本项被插入到场景中时，它也会更新它的小部件。这个类还从场景中删除项目，这决定了项目在相互重叠时的绘制顺序。

作者

LIN Guocheng

日期

2021-8-13

版本

V1.0.0

注意

QT版本: 5.12.11

Date	Version	Author	Description
2021/08/06	0.0.1	LIN Guocheng	创建初始版本
2021/08/12	0.0.2	LIN Guocheng	增加与子窗口通信的信号和槽函数，用于传递输入输出值
2021/08/13	1.0.0	LIN Guocheng	完成第一代基础版本的适配

修改日志:

7.8.2 变量说明

7.8.2.1 InsertTextButton

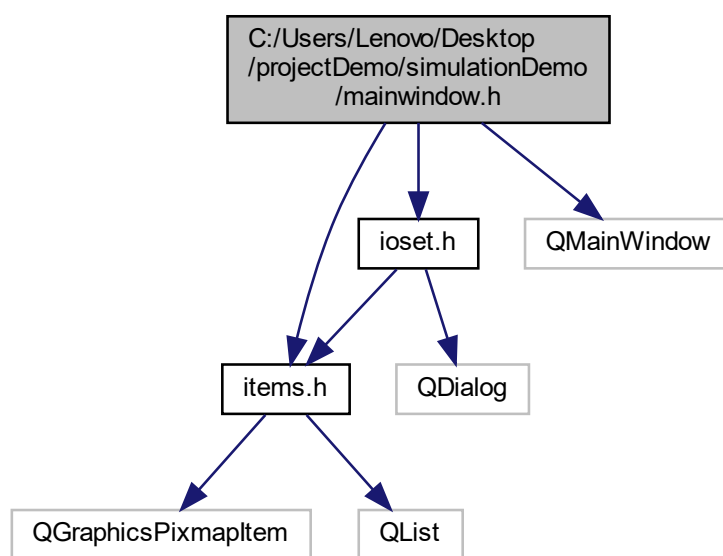
```
const int InsertTextButton = 10
```

在文件 `mainwindow.cpp` 第 37 行定义.

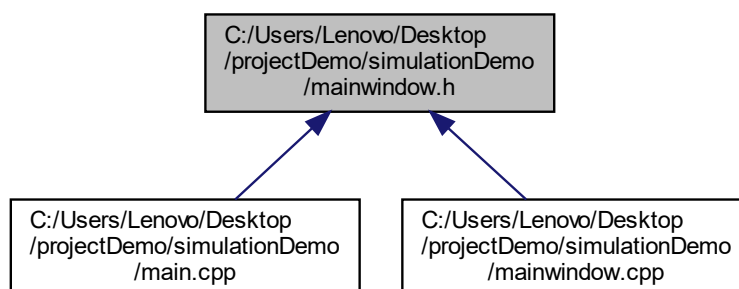
7.9 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/mainwindow.h 文件参考

项目主函数头文件

```
#include "items.h"  
#include "ioaset.h"  
#include <QMainWindow>  
mainwindow.h 的引用(Include)关系图:
```



此图展示该文件直接或间接的被哪些文件引用了:



结构体

- class [MainWindow](#)

7.9.1 详细描述

项目主函数头文件

主要包含主窗口界面的函数声明

作者

LIN Guocheng

日期

2021-8-13

版本

1.0.0

注意

QT版本: 5.12.11

Date	Version	Author	Description
2021/08/06	0.0.1	LIN Guocheng	创建初始版本
2021/08/12	0.0.2	LIN Guocheng	增加与子窗口通信的信号和槽函数，用于传递输入输出值
2021/08/13	1.0.0	LIN Guocheng	完成第一代基础版本的适配

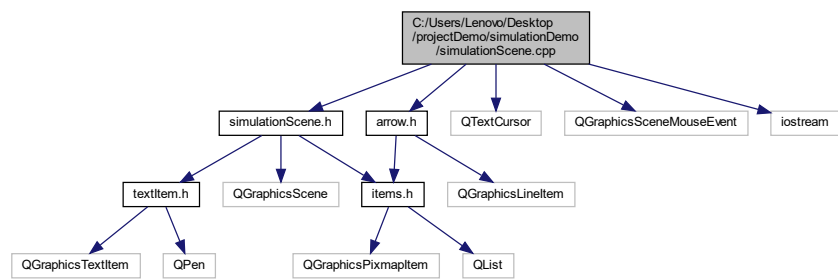
修改日志:

7.10 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/simulationScene.cpp 文件参考

仿真场景实现函数文件

```
#include "simulationScene.h"
#include "arrow.h"
#include <QTextCursor>
#include <QGraphicsSceneMouseEvent>
```

#include <iostream>
simulationScene.cpp 的引用(Include)关系图:



7.10.1 详细描述

仿真场景实现函数文件

主要包含仿真场景功能的实现 `SimulationScene`类继承了`QGraphicsScene`，并添加了处理`Items`、`arrow`和`TextItems`的功能，以及由它的超类处理的项目。

作者
LIN Guocheng

日期
2021-8-13

版本
1.0.0

注意
QT版本: 5.12.11

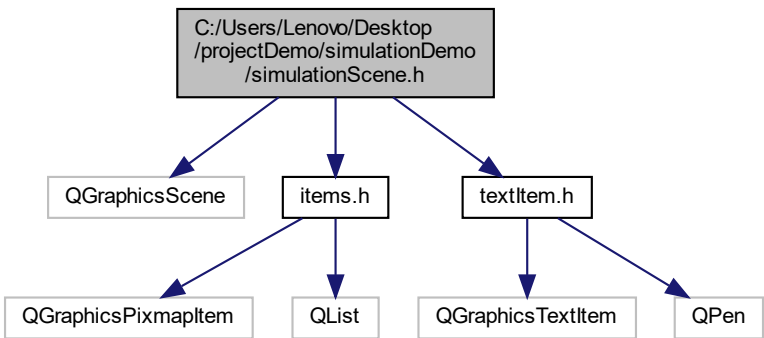
Date	Version	Author	Description
2021/08/06	0.0.1	LIN Guocheng	创建初始版本
2021/08/09	0.0.2	LIN Guocheng	mousePressEvent中增加了对于item的输入输出数处理结果
2021/08/13	1.0.0	LIN Guocheng	完成第一代基础版本的适配

修改日志:

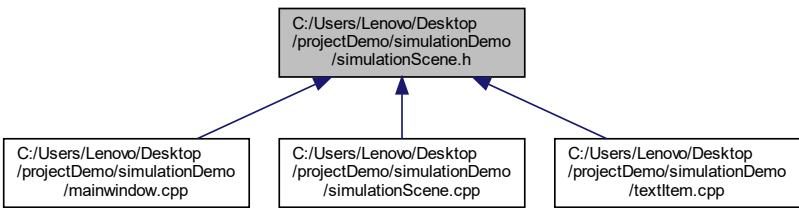
7.11 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/simulationScene.h 文件参考

项目场景函数头文件

```
#include <QGraphicsScene>
#include "items.h"
#include "textItem.h"
simulationScene.h 的引用(Include)关系图:
```



此图展示该文件直接或间接的被哪些文件引用了:



结构体

- class [SimulationScene](#)

7.11.1 详细描述

项目场景函数头文件

主要包含SimulationScene类的声明

作者

LIN Guocheng

日期

2021-8-13

版本

1.0.0

注意

QT版本: 5.12.11

Date	Version	Author	Description
2021/08/06	0.0.1	LIN Guocheng	创建初始版本
2021/08/13	1.0.0	LIN Guocheng	完成第一代基础版本的适配

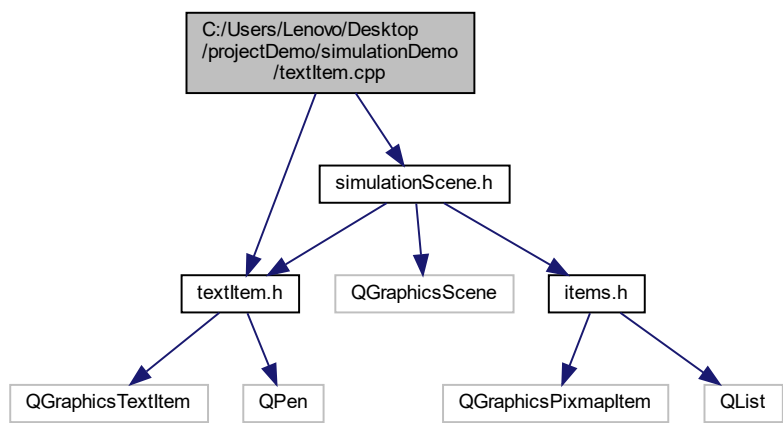
修改日志:

7.12 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/textItem.cpp 文件参考

文本编辑框函数文件

```
#include "textItem.h"
#include "simulationScene.h"
```

textItem.cpp 的引用(Include)关系图:



7.12.1 详细描述

文本编辑框函数文件

主要包含文本编辑框功能的实现 `textItem`类继承了`QGraphicsTextItem`，并添加了移动可编辑文本项的可能性。可编辑的`QGraphicsTextItems`被设计成在适当的位置进行固定，当用户单击项目时开始编辑。使用`textItem`，编辑以双击开始，留下可进行交互和移动的单单击。

作者
LIN Guocheng

日期
2021-8-13

版本
1.0.0

注意
QT版本: 5.12.11

Date	Version	Author	Description
2021/08/06	0.0.1	LIN Guocheng	创建初始版本
2021/08/13	1.0.0	LIN Guocheng	完成第一代基础版本的适配

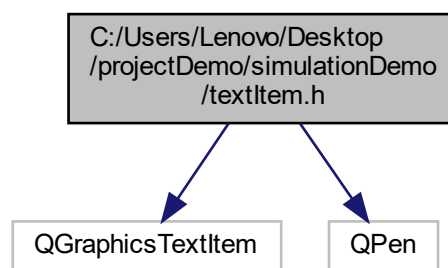
修改日志:

7.13 C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/textItem.h 文件参考

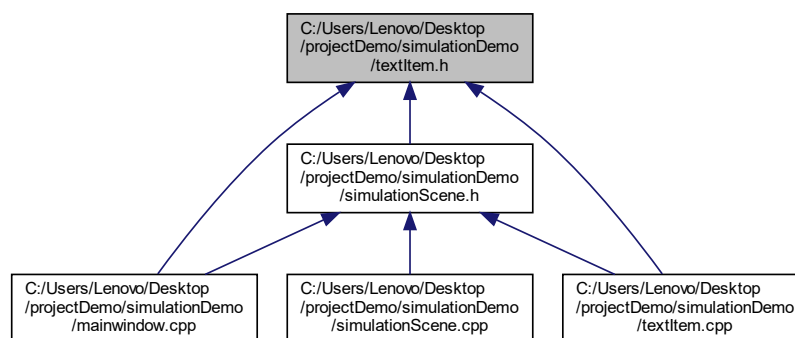
可编辑文本项类声明头文件

```
#include <QGraphicsTextItem>
#include <QPen>
```

textItem.h 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



结构体

- class [TextItem](#)

7.13.1 详细描述

可编辑文本项类声明头文件

主要包含可编辑文本项类声明

作者

LIN Guocheng

日期

2021-8-13

版本

1.0.0

注意

QT版本: 5.12.11

Date	Version	Author	Description
2021/08/06	0.0.1	LIN Guocheng	创建初始版本
2021/08/13	1.0.0	LIN Guocheng	完成第一代基础版本的适配

修改日志:

Index

~IOset
 IOset, [18](#)

about
 MainWindow, [34](#)

aboutAction
 MainWindow, [40](#)

AboutMenu
 MainWindow, [40](#)

addAction
 MainWindow, [41](#)

addArrow
 Items, [23](#)

Arrow, [11](#)
 Arrow, [13](#)
 arrowHead, [16](#)
 boundingRect, [14](#)
 endItem, [14](#)
 myColor, [16](#)
 myEndItem, [16](#)
 myStartItem, [17](#)
 paint, [14](#)
 shape, [15](#)
 startItem, [15](#)
 Type, [13](#)
 type, [15](#)
 updatePosition, [16](#)

arrowHead
 Arrow, [16](#)

arrows
 Items, [30](#)

boldAction
 MainWindow, [41](#)

boundingRect
 Arrow, [14](#)

bringToFront
 MainWindow, [34](#)

buttonGroup
 MainWindow, [41](#)

buttonGroupClicked
 MainWindow, [34](#)

C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/arrow.cpp,
 [61](#)

C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/arrow.h,
 [62](#)

C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/ioset.cpp,
 [64](#)

C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/ioset.h,
 [65](#)

C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/items.cpp,
 [67](#)

C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/items.h,
 [69](#)

C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/main.cpp,
 [70](#)

C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/mainwindow.cpp,
 [72](#)

C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/mainwindow.h,
 [74](#)

C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/simulationScene.
 [75](#)

C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/simulationScene.
 [77](#)

C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/textItem.cpp,
 [78](#)

C:/Users/Lenovo/Desktop/projectDemo/simulationDemo/textItem.h,
 [80](#)

calculateResult
 Items, [23](#)

colorToolBar
 MainWindow, [41](#)

contextMenuEvent
 Items, [24](#)

createActions
 MainWindow, [35](#)

createBackgroundCellWidget
 MainWindow, [35](#)

createCellWidget
 MainWindow, [35](#)

createColorIcon
 MainWindow, [35](#)

createColorMenu
 MainWindow, [36](#)

createColorToolButtonIcon
 MainWindow, [36](#)

createMenus
 MainWindow, [36](#)

createToolbars
 MainWindow, [36](#)

createToolBox
 MainWindow, [36](#)

currentFontChanged
 MainWindow, [36](#)

deleteAction
 MainWindow, [41](#)

deleteItem

- MainWindow, 37
- editorLostFocus
 - SimulationScene, 48
- editToolBar
 - MainWindow, 41
- endItem
 - Arrow, 14
- exitAction
 - MainWindow, 42
- fileMenu
 - MainWindow, 42
- fillAction
 - MainWindow, 42
- focusOutEvent
 - TextItem, 57
- font
 - SimulationScene, 49
- fontCombo
 - MainWindow, 42
- fontSizeChanged
 - MainWindow, 37
- fontSizeCombo
 - MainWindow, 42
- getInputNum
 - Items, 24
- getInputValue
 - MainWindow, 37
- getItemType
 - Items, 25
- getOutputNum
 - Items, 25
- getOutputValue
 - IOset, 19
- handleFontChange
 - MainWindow, 38
- image
 - Items, 26
- inputNum
 - Items, 30
- insertItem
 - SimulationScene, 48
- insertLine
 - SimulationScene, 48
- insertText
 - SimulationScene, 48
- InsertTextButton
 - mainwindow.cpp, 73
- IOset, 17
 - ~IOset, 18
 - getOutputValue, 19
 - IOset, 18
 - on_SetInput_clicked, 19
 - sendInputValue, 19
 - ui, 20
- isItemChange
 - SimulationScene, 49
- italicAction
 - MainWindow, 42
- itemChange
 - Items, 26
 - TextItem, 58
- itemColorCombo
 - MainWindow, 43
- itemInserted
 - MainWindow, 38
 - SimulationScene, 50
- itemMenu
 - MainWindow, 43
- Items, 20
 - addArrow, 23
 - arrows, 30
 - calculateResult, 23
 - contextMenuEvent, 24
 - getInputNum, 24
 - getItemType, 25
 - getOutputNum, 25
 - image, 26
 - inputNum, 30
 - itemChange, 26
 - Items, 22
 - ItemType, 22
 - ItemType, 27
 - myContextMenu, 30
 - myItemType, 30
 - myPolygon, 30
 - outputNum, 30
 - polygon, 27
 - removeArrow, 28
 - removeArrows, 28
 - setInputNum, 29
 - test, 22
 - Type, 22
 - type, 29
- itemSelected
 - MainWindow, 38
 - SimulationScene, 50
- ItemType
 - Items, 22
- itemType
 - Items, 27
- leftButtonDown
 - SimulationScene, 54
- line
 - SimulationScene, 54
- lineAction
 - MainWindow, 43
- lostFocus
 - TextItem, 58
- main
 - main.cpp, 72
- main.cpp

- main, 72
- MainWindow, 31
 - about, 34
 - aboutAction, 40
 - AboutMenu, 40
 - addAction, 41
 - boldAction, 41
 - bringToFront, 34
 - buttonGroup, 41
 - buttonGroupClicked, 34
 - colorToolBar, 41
 - createActions, 35
 - createBackgroundCellWidget, 35
 - createCellWidget, 35
 - createColorIcon, 35
 - createColorMenu, 36
 - createColorToolButtonIcon, 36
 - createMenus, 36
 - createToolbars, 36
 - createToolBox, 36
 - currentFontChanged, 36
 - deleteAction, 41
 - deleteItem, 37
 - editToolBar, 41
 - exitAction, 42
 - fileMenu, 42
 - fillAction, 42
 - fontCombo, 42
 - fontSizeChanged, 37
 - fontSizeCombo, 42
 - getInputValue, 37
 - handleFontChange, 38
 - italicAction, 42
 - itemColorCombo, 43
 - itemInserted, 38
 - itemMenu, 43
 - itemSelected, 38
 - lineAction, 43
 - MainWindow, 34
 - pointerGroupClicked, 39
 - pointerToolBar, 43
 - pointerTypeGroup, 43
 - scene, 43
 - sceneScaleChanged, 39
 - sceneScaleCombo, 44
 - sendBackAction, 44
 - sendOutputValue, 39
 - sendToBack, 39
 - setGetAction, 44
 - setGetValueWindow, 40
 - setGetWindow, 44
 - textAction, 44
 - textColorCombo, 44
 - textInserted, 40
 - textToolBar, 45
 - toFrontAction, 45
 - toolBox, 45
 - underlineAction, 45
 - view, 45
- mainwindow.cpp
 - InsertTextButton, 73
- Mode
 - SimulationScene, 47
- mouseDoubleClickEvent
 - TextItem, 58
- mouseMoveEvent
 - SimulationScene, 50
- mousePressEvent
 - SimulationScene, 51
- mouseReleaseEvent
 - SimulationScene, 51
- moveItem
 - SimulationScene, 48
- myColor
 - Arrow, 16
- myContextMenu
 - Items, 30
- myEndItem
 - Arrow, 16
- myFont
 - SimulationScene, 54
- myItemColor
 - SimulationScene, 54
- myItemMenu
 - SimulationScene, 54
- myItemType
 - Items, 30
 - SimulationScene, 54
- myLineColor
 - SimulationScene, 55
- myMode
 - SimulationScene, 55
- myPolygon
 - Items, 30
- myStartItem
 - Arrow, 17
- myTextColor
 - SimulationScene, 55
- on_SetInput_clicked
 - IOset, 19
- outputNum
 - Items, 30
- paint
 - Arrow, 14
- pointerGroupClicked
 - MainWindow, 39
- pointerToolBar
 - MainWindow, 43
- pointerTypeGroup
 - MainWindow, 43
- polygon
 - Items, 27
- removeArrow
 - Items, 28

- removeArrows
 - Items, 28
- scene
 - MainWindow, 43
- sceneScaleChanged
 - MainWindow, 39
- sceneScaleCombo
 - MainWindow, 44
- selectedChange
 - TextItem, 60
- sendBackAction
 - MainWindow, 44
- sendInputValue
 - IOset, 19
- sendOutputValue
 - MainWindow, 39
- sendToBack
 - MainWindow, 39
- setFont
 - SimulationScene, 52
- setGetAction
 - MainWindow, 44
- setGetValueWindow
 - MainWindow, 40
- setGetWindow
 - MainWindow, 44
- setInputNum
 - Items, 29
- setItemType
 - SimulationScene, 53
- setMode
 - SimulationScene, 53
- shape
 - Arrow, 15
- SimulationScene, 46
 - editorLostFocus, 48
 - font, 49
 - insertItem, 48
 - insertLine, 48
 - insertText, 48
 - isItemChange, 49
 - itemInserted, 50
 - itemSelected, 50
 - leftButtonDown, 54
 - line, 54
 - Mode, 47
 - mouseMoveEvent, 50
 - mousePressEvent, 51
 - mouseReleaseEvent, 51
 - moveItem, 48
 - myFont, 54
 - myItemColor, 54
 - myItemMenu, 54
 - myItemType, 54
 - myLineColor, 55
 - myMode, 55
 - myTextColor, 55
 - setFont, 52
 - setItemType, 53
 - setMode, 53
 - SimulationScene, 48
 - startPoint, 55
 - textInserted, 53
 - textItem, 55
- startItem
 - Arrow, 15
- startPoint
 - SimulationScene, 55
- test
 - Items, 22
- textAction
 - MainWindow, 44
- textColorCombo
 - MainWindow, 44
- textInserted
 - MainWindow, 40
 - SimulationScene, 53
- TextItem, 56
 - focusOutEvent, 57
 - itemChange, 58
 - lostFocus, 58
 - mouseDoubleClickEvent, 58
 - selectedChange, 60
 - TextItem, 57
 - Type, 57
 - type, 60
- textItem
 - SimulationScene, 55
- textToolBar
 - MainWindow, 45
- toFrontAction
 - MainWindow, 45
- toolBox
 - MainWindow, 45
- Type
 - Arrow, 13
 - Items, 22
 - TextItem, 57
- type
 - Arrow, 15
 - Items, 29
 - TextItem, 60
- Ui, 9
- ui
 - IOset, 20
- underlineAction
 - MainWindow, 45
- updatePosition
 - Arrow, 16
- view
 - MainWindow, 45