<https://blog.csdn.net/qq_17813937/article/details/51370106>

**QGraphicsItem**

2016-05-11 02:49:53 [点兔酱](https://me.csdn.net/qq_17813937) 阅读数 6904更多

分类专栏： [QT](https://blog.csdn.net/qq_17813937/article/category/6217567)

**Public Types**

**enum QGraphicsItem::CacheMode**

QGraphicsItem::NoCache   
禁用缓存  
  
QGraphicsItem::ItemCoordinateCache

缓存能够用于元素的逻辑坐标系统,QGraphicsItem用一个可配置的大小/解决方案建立了屏幕以外的像素缓冲区，追求质量会典型地降低，取决于缓冲的分辨率和元素变换，可以调用setCacheMode()再次调整缓存的分辨率

QGraphicsItem::DeviceCoordinateCache   
缓存能够用于设备的逻辑坐标系统,这种模式可以移动元素,但不可以旋转,缩放或剪切，总是显示最好的画质

**enum QGraphicsItem::GraphicsItemChange**

QGraphicsItem::ItemEnabledChange   
项目启用状态更改，可以从itemChanged()返回新的状态   
  
QGraphicsItem::ItemEnabledHasChanged   
项目启用状态更改，itemChange将忽略该返回值   
  
QGraphicsItem::ItemMatrixChange  
项目矩阵旋转   
  
QGraphicsItem::ItemPositionChange  
项目位置变化，如果是ItemSendsGeometryChanges发送，可以调用pos()得到原来的位置,  
调用itemChange()后QGraphicsItem将会立即发送ItemPositionHasChanged，得到变化后的位置  
  
QGraphicsItem::ItemPositionHasChanged   
项目位置发生变化，pos()获得当前位置，QGraphicsItem忽略该消息的返回值  
  
QGraphicsItem::ItemTransformChange  
项目变换矩阵变化，如果是ItemSendsGeometryChanges发送，transform返回以前的矩阵，可以从ItemChange获得变换后的矩阵，如果更改了转换属性，该信号将不会发送  
  
QGraphicsItem::ItemTransformHasChanged  
项目变换矩阵变化，可能是setTransform或转换特性的改变，如果是ItemSendsGeometryChanges发送，transform和QGraphicsItem忽略该消息的返回值  
  
QGraphicsItem::ItemRotationChange  
项目旋转属性更改，如果是ItemSendsGeometryChanges发送，rotation返回以前的旋转，从itemChange得到新的矩阵  
  
QGraphicsItem::ItemRotationHasChanged  
项目旋转属性更改，如果是ItemSendsGeometryChanges发送，QGraphicsItem将忽略该消息的返回值  
  
QGraphicsItem::ItemScaleChange  
项目比例发生变化，scale获得以前的比例，itemChange返回当前尺寸  
  
QGraphicsItem::ItemScaleHasChanged  
项目比例发生变化，如果是ItemSendsGeometryChanges发送，QGraphicsItem将忽略该消息的返回值  
  
QGraphicsItem::ItemTransformOriginPointChange  
项目变换原点坐标改变，如果是ItemSendsGeometryChanges发送，transformOriginPoint获得以前的坐标，itemChange获得当前坐标  
  
QGraphicsItem::ItemTransformOriginPointHasChanged  
项目变换原点坐标改变，如果是ItemSendsGeometryChanges发送，QGraphicsItem将忽略该消息的返回值  
  
QGraphicsItem::ItemSelectedChange

项目选定状态改变，如果该项目当前是选定的，它将会选择。 itemChange返回新的选中状态

QGraphicsItem::ItemSelectedHasChanged  
项目选定状态改变，itemChange忽略返回值  
  
QGraphicsItem::ItemVisibleChange  
项目可见状态改变，itemChage返回新的可见状态  
  
  
QGraphicsItem::ItemVisibleHasChanged  
项目可见状态改变，itemChange忽略返回值  
  
  
QGraphicsItem::ItemParentChange  
项目父项改变，itemChage返回新的可见状态  
  
  
QGraphicsItem::ItemParentHasChanged  
项目父项改变，itemChange忽略返回值  
  
QGraphicsItem::ItemChildAddedChange  
添加了一个子项，返回值未使用，新的子项可能无法完全构造，纯虚函数会导致崩溃  
  
QGraphicsItem::ItemChildRemovedChange  
移除一个子项，返回值未使用  
  
QGraphicsItem::ItemSceneChange  
项目被移动到新的场景，当该项目被删除时scene是老场景或者一个空指针，itemChange返回新场景  
  
QGraphicsItem::ItemSceneHasChanged  
项目被移动到新的场景，返回值被忽略  
  
QGraphicsItem::ItemCursorChange  
项目光标改变，itemChange返回新光标  
  
QGraphicsItem::ItemCursorHasChanged  
项目光标改变，返回值被忽略  
  
QGraphicsItem::ItemToolTipChange  
项目提示改变，itemChange返回新提示  
  
QGraphicsItem::ItemToolTipHasChanged  
项目提示改变，返回值被忽略  
  
QGraphicsItem::ItemFlagsChange  
项目焦点改变，itemChange返回新焦点  
  
QGraphicsItem::ItemFlagsHaveChanged  
项目焦点改变，返回值被忽略  
  
QGraphicsItem::ItemZValueChange  
项目Z值改变，itemChange返回新Z值  
  
QGraphicsItem::ItemZValueHasChanged  
项目Z值改变，返回值被忽略  
  
QGraphicsItem::ItemOpacityChange  
项目透明度改变，itemChange返回透明度  
  
QGraphicsItem::ItemOpacityHasChanged  
项目透明度改变，返回值被忽略  
  
QGraphicsItem::ItemScenePositionHasChanged  
项目场景位置发生变化，如果是ItemSendsScenePositionChanges发出，QGraphicsItem忽略该消息的返回值

**enum QGraphicsItem::GraphicsItemFlag**

QGraphicsItem::ItemIsMovable  
项目可以通过点击和拖动移动（递归子项）  
  
QGraphicsItem::ItemIsSelectable  
项目可以通过点击，框选拖动或者在QGraphicsScene::setSelectionArea()调用所影响的区域内被选中  
  
QGraphicsItem::ItemIsFocusable  
项目支持键盘按下事件  
  
QGraphicsItem::ItemClipsToShape  
按自己的形状剪切，不管如何绘制，也不能接收在它形状之外的事件  
  
QGraphicsItem::ItemClipsChildrenToShape  
项目按照自己的形状剪切它的所有子项，【递归】  
  
QGraphicsItem::ItemIgnoresTransformations  
忽略应用到它的父项的变换（位置与父项关联）  
  
QGraphicsItem::ItemIgnoresParentOpacity  
忽略父项透明度  
  
QGraphicsItem::ItemDoesntPropagateOpacityToChildren  
透明度不影响子项  
  
QGraphicsItem::ItemStacksBehindParent  
堆叠到父项后面  
  
QGraphicsItem::ItemUsesExtendedStyleOption  
为项提供额外的QStyleOptionGraphicsItem属性  
  
QGraphicsItem::ItemHasNoContents  
项目不绘制任何内容  
  
QGraphicsItem::ItemSendsGeometryChanges  
项目位置或变换发生改变时调用itemChange  
  
QGraphicsItem::ItemAcceptsInputMethod  
项目接受输入法  
  
QGraphicsItem::ItemNegativeZStacksBehindParent  
如果Z值是负的，自动放在父项后方  
  
QGraphicsItem::ItemIsPanel  
该项目是一个面板，提供了激活和焦点处理，只有一个面板可以处于活动状态  
  
QGraphicsItem::ItemSendsScenePositionChanges  
项目在位置变化发生时调用itemChange()  
  
QGraphicsItem::ItemContainsChildrenInShape  
显示所有项目的直接或间接的子项只在项目的形状中绘制，默认情况下禁用

**enum QGraphicsItem::PanelModality**

QGraphicsItem::NonModal  
该面板是不是模态，并不会阻止输入到其他面板。这是面板的默认值。  
  
QGraphicsItem::PanelModal  
面板模型？  
  
QGraphicsItem::SceneModal

窗口是模态的，整个场景和块输入到所有面板。

**public Functions**

**boundingRect**() const = 0

这是个纯虚函数，定义了边界矩形尺寸

**paint**(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget = Q\_NULLPTR) = 0  
这是个纯虚函数，指定了在边界矩形内如何绘画

1. class SimpleItem : public QGraphicsItem
2. {
3. public:
4. QRectF boundingRect() const
5. {
6. qreal penWidth = 1;
7. return QRectF(-10 - penWidth / 2, -10 - penWidth / 2,
8. 20 + penWidth, 20 + penWidth);
9. }
11. void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option,
12. QWidget \*widget)
13. {
14. painter->drawRoundedRect(-10, -10, 20, 20, 5, 5);
15. }
16. };

**boundingRegion(const QTransform &itemToDeviceTransform) const  
返回该item的轮廓区域，返回的区域的坐标系统依赖于参数itemToDeviceTransform  
如果你传递一个Qtransform对象作为参数，那么函数将返回本地坐标系统区域  
返回的区域是item内容可见的一个大概的轮廓，尽管计算起来很浪费空间和时间  
但是比boundingRect（）更精准，而且当重绘时候，它还能避免不必要的重绘  
对像线或者简单的多边形来说非常有效**

**childItems() const  
返回孩子item的一个链表，这些items按照栈顺序排列，考虑了item的插入顺序和Z-values  
  
childrenBoundingRect() const  
返回这个item的所有子孙items的本地坐标系统轮廓矩形。这个矩形包括了这个item所有的子孙item，如果这个item不含有子孙item，这个函数将返回一个空的矩形。  
  
clearFocus()  
去除item的光标，如果原来有光标焦点，当焦点失去时候，事件focus out event会发送给这个item，提示他将会失去焦点。  
  
clipPath() const  
返回此项目的裁剪路径  
  
collidesWithItem(const QGraphicsItem \*other, Qt::ItemSelectionMode mode = Qt::IntersectsItemShape) const  
如果这个item和另一个item碰撞就返回真，mode是应用于其他item的，默认是Qt::IntersectsItemShape，如果其他的item和这个item相互影响或者包含，被包含于这个item的形状（详见Qt::ItemSelectionMode）  
默认的函数是基于外形相互影响，它调用两个item的shape()函数，这将会花费很大时间，你可以在QgraphicsItem子类中重写这个函数，提供一个简单的算法。这样你可以使用自己item的一些参数，这样可以提高碰撞侦测的效果。例如，两个完全没有转换的圆item的碰撞可以通过对比他们的圆心位置和半径  
  
collidesWithPath(const QPainterPath &path, Qt::ItemSelectionMode mode = Qt::IntersectsItemShape) const  
如果item按照path碰撞就返回真  
  
collidingItems(Qt::ItemSelectionMode mode = Qt::IntersectsItemShape) const  
返回碰撞列表，碰撞模式取决于mode  
  
commonAncestorItem(const QGraphicsItem \*other) const  
返回item最近的祖先item，如果other为0，或者没有祖先item，就返回0  
  
contains(const QPointF &point) const  
如果item包含了点point就返回TRUE，否则返回FALSE，最常用的是在QgraphicsView中调用，来判断这个item是否在光标下面，如果要重写这个函数，那就尽可能的简单。默认的这个函数调用的shape()函数  
  
deviceTransform(const QTransform &viewportTransform) const  
返回item设备的转换矩阵，使用viewportTransform来从场景到设备影射坐标，这个矩阵可以用来从这个item本地坐标到视图口坐标系统映射坐标或者几何图形，如果要映射视图口坐标到本地坐标，首先应该转换返回的这个矩阵。  
  
effectiveOpacity() const  
返回item的有效透明度，这个数值在0.0~~1.0之间  
  
ensureVisible(const QRectF &rect = QRectF(), int xmargin = 50, int ymargin = 50)  
如果项目是场景，项目将会检查鼠标是否处在可见区域的外面并且根据检查的结果移动滚动文本区域  
  
focusItem() const  
如果子项具有获得焦点，将返回该项指针，否则返回0  
  
grabKeyboard()  
项目接收键盘事件，直到该项目不可见，从场景移除或删除该项  
  
grabMouse()  
项目接收鼠标事件，直到该项目不可见，从场景移除或删除该项  
  
hasCursor() const  
如果该项有一个光标设置返回真，否则为假，默认为假，cursor()返回标准的箭头光标  
  
hasFocus() const  
项目是活动的，返回真，否则为假  
  
hide()  
隐藏项目  
  
installSceneEventFilter(QGraphicsItem \*filterItem)  
安装场景事件过滤器  
  
isAncestorOf(const QGraphicsItem \*child) const  
如果项目有子项，返回真  
  
isBlockedByModalPanel(QGraphicsItem \*\*blockingPanel = Q\_NULLPTR) const  
如果项目被模态窗口阻挡，返回真  
  
isClipped() const  
如果项目被裁剪，返回真  
  
isObscured(const QRectF &rect = QRectF()) const  
如果被任何碰撞项上面的不透明的形状完全遮蔽该项，返回true  
  
isObscuredBy(const QGraphicsItem \*item) const  
如果这个项目的包围矩形被项目的不透明的形状完全遮蔽，返回true  
  
isPanel() const  
如果项目是一个面板，返回真  
  
isUnderMouse() const  
如果光标在项目上，返回真  
  
isVisibleTo(const QGraphicsItem \*parent) const  
如果项目是可见的，返回真  
  
isWidget() const  
如果项目是控件，返回真  
  
isWindow() const  
如果项目是QGraphicsWidget，返回真  
  
itemTransform(const QGraphicsItem \*other, bool \*ok = Q\_NULLPTR) const  
返回QTarnsForm式地图坐标  
  
mapFromItem(const QGraphicsItem \*item, qreal x, qreal y, qreal w, qreal h) const  
项目坐标转换，返回映射的坐标，如果item是0，那么返回和mapFromScene()一样  
  
mapFromParent(qreal x, qreal y, qreal w, qreal h) const  
这是个父坐标系统转换，返回映射的坐标  
  
mapFromScene(qreal x, qreal y, qreal w, qreal h) const  
转换为场景坐标系统，返回映射的坐标  
  
mapRectFromItem(const QGraphicsItem \*item, qreal x, qreal y, qreal w, qreal h) const  
转换为项目矩形坐标，返回映射的矩形  
  
mapRectFromParent(qreal x, qreal y, qreal w, qreal h) const  
转换为父项目坐标系统，返回映射的矩形  
  
mapRectFromScene(qreal x, qreal y, qreal w, qreal h) const  
转换为场景坐标，返回映射的矩形  
  
mapRectToItem(const QGraphicsItem \*item, qreal x, qreal y, qreal w, qreal h) const  
转换为项目矩形坐标，返回映射的矩形，如果项目是0，这个函数返回的是mapFromScene()相同  
  
mapRectToParent(qreal x, qreal y, qreal w, qreal h) const  
转换为父项目坐标系统，返回映射的矩形  
  
mapRectToScene(qreal x, qreal y, qreal w, qreal h) const  
转换为场景坐标，返回映射的矩形  
  
mapToItem(const QGraphicsItem \*item, qreal x, qreal y, qreal w, qreal h) const  
转换为项目坐标，返回映射的坐标  
  
mapToParent(qreal x, qreal y, qreal w, qreal h) const  
转换为父项坐标，返回映射的坐标  
  
mapToScene(qreal x, qreal y, qreal w, qreal h) const  
转换为场景坐标，返回映射的坐标  
  
moveBy(qreal dx, qreal dy)  
相对移动  
  
opaqueArea() const  
返回不透明形状区域，默认将返回一个空QPainterPath  
  
panel() const  
如果该项是面板，返回自身，否则如果父项是面板将返回父项，否则返回0  
  
parentObject() const  
如果父项不是QGraphicsObject，返回0，否则返回父项  
  
parentWidget() const  
返回父项控件  
  
removeSceneEventFilter(QGraphicsItem \*filterItem)  
移除场景事件过滤器  
  
resetTransform()  
重置变换矩阵  
  
rotation() const  
返回顺时针旋转角度  
  
scene() const  
返回项目的当前场景  
  
sceneBoundingRect() const  
返回项目的场景的矩形  
  
scenePos() const  
返回项目在场景的位置  
  
sceneTransform() const  
返回这个项目的场景转换矩阵  
  
scroll(qreal dx, qreal dy, const QRectF &rect = QRectF())  
滚动项目，滚动比重绘快  
  
setAcceptDrops(bool on)  
设置接受拖放  
  
setAcceptHoverEvents(bool enabled)  
设置接受悬停事件  
  
setAcceptTouchEvents(bool enabled)  
设置接受触摸事件  
  
setAcceptedMouseButtons(Qt::MouseButtons buttons)  
设置该项目接受鼠标事件的鼠标按钮  
  
setActive(bool active)  
设置项目是否活动  
  
setBoundingRegionGranularity(qreal granularity)  
调节轮廓区域的粒度，默认的粒度是0，这时候item的区域和轮廓矩形一样的  
  
setCacheMode(CacheMode mode, const QSize &logicalCacheSize = QSize())  
设置缓存模式  
  
setCursor(const QCursor &cursor)  
设置光标形状  
  
setData(int key, const QVariant &value)  
设置自定义数据  
  
setEnabled(bool enabled)  
设置启用项目  
  
setFiltersChildEvents(bool enabled)  
设置过滤子项事件  
  
setFlag(GraphicsItemFlag flag, bool enabled = true)  
设置标志  
  
setFlags(GraphicsItemFlags flags)  
设置所有标志  
  
setFocus(Qt::FocusReason focusReason = Qt::OtherFocusReason)  
设置焦点  
  
setFocusProxy(QGraphicsItem \*item)  
将该项目设置为焦点代理  
  
setGraphicsEffect(QGraphicsEffect \*effect)  
设置效果  
  
setGroup(QGraphicsItemGroup \*group)  
设置项目组  
  
setInputMethodHints(Qt::InputMethodHints hints)  
设置此项当前输入方法提示  
  
setOpacity(qreal opacity)  
设置不透明度  
  
setPanelModality(PanelModality panelModality)  
设置面板模式  
  
setParentItem(QGraphicsItem \*newParent)  
设置父项  
  
setPos(const QPointF &pos)  
设置位置  
  
setRotation(qreal angle)  
设置旋转角度  
  
setScale(qreal factor)  
设置场景  
  
setSelected(bool selected)  
设置选择  
  
setToolTip(const QString &toolTip)  
设置提示  
  
setTransform(const QTransform &matrix, bool combine = false)  
设置变换  
  
setTransformOriginPoint(qreal x, qreal y)  
设置变换原点  
  
setTransformations(const QList<QGraphicsTransform \*> &transformations)  
设置旋转列表  
  
setVisible(bool visible)  
设置可见  
  
setX(qreal x)  
setY(qreal y)  
设置坐标  
  
setZValue(qreal z)  
设置Z值  
  
shape() const  
返回形状，用于碰撞检测，命中测试等，默认返回矩形  
  
show()  
显示  
  
stackBefore(const QGraphicsItem \*sibling)  
堆叠兄弟项目，项目必须是兄弟项  
  
toGraphicsObject() const  
返回转换为QGraphicsObject，如果本来是图像对象，返回0**

**topLevelItem() const  
返回顶级项目，如果没有父项，返回自己  
  
topLevelWidget() const  
返回一个指向该项目的顶级控件，如果祖先不是控件，返回0，如果没有顶级控件，返回自己  
  
type() const  
返回类型  
  
ungrabKeyboard()  
释放键盘事件  
  
ungrabMouse()  
释放鼠标事件  
  
unsetCursor()  
从项目中清楚光标  
  
update(const QRectF &rect = QRectF())  
更新  
  
window() const  
返回项目的窗口，或0**

**Static Public Members**

**Type  
重载设置项目类型  
  
UserType**

**最低项目类型值**

**Protected Functions**

**contextMenuEvent(QGraphicsSceneContextMenuEvent \*event)  
上下文菜单事件  
  
dragEnterEvent(QGraphicsSceneDragDropEvent \*event)  
拖拽输入事件  
  
dragLeaveEvent(QGraphicsSceneDragDropEvent \*event)  
拖拽离开事件  
  
  
dragMoveEvent(QGraphicsSceneDragDropEvent \*event)  
拖拽移动事件  
  
dropEvent(QGraphicsSceneDragDropEvent \*event)  
拖拽事件  
  
focusInEvent(QFocusEvent \*event)  
焦点事件  
  
focusOutEvent(QFocusEvent \*event)  
取消焦点事件  
  
hoverEnterEvent(QGraphicsSceneHoverEvent \*event)  
悬停进入事件  
  
hoverLeaveEvent(QGraphicsSceneHoverEvent \*event)  
悬停离开事件  
  
hoverMoveEvent(QGraphicsSceneHoverEvent \*event)  
悬停移动事件  
  
inputMethodEvent(QInputMethodEvent \*event)  
输入法事件  
  
inputMethodQuery(Qt::InputMethodQuery query) const  
输入法查询事件  
  
itemChange(GraphicsItemChange change, const QVariant &value)  
项目改变  
  
keyPressEvent(QKeyEvent \*event)  
按下按键  
  
keyReleaseEvent(QKeyEvent \*event)  
释放按键  
  
mouseDoubleClickEvent(QGraphicsSceneMouseEvent \*event)  
双击事件  
  
mouseMoveEvent(QGraphicsSceneMouseEvent \*event)  
鼠标移动事件**

**mousePressEvent(QGraphicsSceneMouseEvent \*event)  
鼠标单击事件**

**mouseReleaseEvent(QGraphicsSceneMouseEvent \*event)  
鼠标是否事件  
  
prepareGeometryChange()  
改变包围矩形  
  
sceneEvent(QEvent \*event)  
场景事件  
  
sceneEventFilter(QGraphicsItem \*watched, QEvent \*event)  
场景事件过滤  
  
updateMicroFocus()  
焦点更新  
  
wheelEvent(QGraphicsSceneWheelEvent \*event)  
滚轮事件  
  
qgraphicsitem\_cast(QGraphicsItem \*item)  
返回给定项目类型**