我其實只大概知道要怎麼用，不過我會把我的思路告訴給你XD

我一開始查到**collidingitems()**這個語法後

去研究一下這個碰撞判斷的函式

發現他的用途是

根據呼叫函式的**Qgraphicsitems**物件

來計算出

呼叫函式的那一刻，這個物件**總共**有碰到哪些**Qgraphicsitem**物件

並且將那些有碰撞到的**Qgraphicsitem**整理成一份**QList**後

函式最後會回傳那個**QList的位址**

譬如說 我有一個物件叫**player**

我使用**player**-> **collidingitems()**後 此函式會回傳一個**QList位址**

因此我前面還要有一個QList來接收這個位址，

所以前面會多一個**QList<QGraphicsItem\*> tmp**

這一行中，tmp是變數的名字 QList則是型態

後面的**<QGraphicsItem\*>** 則是QList裡所存的資料型態

### 因為官網是寫[QList](https://doc.qt.io/archives/qt-4.8/qlist.html)<[QGraphicsItem](https://doc.qt.io/archives/qt-4.8/qgraphicsitem.html#QGraphicsItem) \*> 變數名稱

### 所以我才會一樣設成QList<QGraphicsItem\*> tmp

### 因此我現在的程式碼可能是

**QList<QGraphicsItem\*> tmp** = player-> **collidingitems() ;**

接下來我只要**對tmp進行操作**就好，

因為我想知道**碰到player**的**物件有哪些**，以及分別**是甚麼**

至於**QList**的物件**tmp**要怎麼操作呢?

其實你可以把它當陣列來取資料

也就是**tmp[0]**是操作在第一個物件

**tmp[1]**是操作在第二個物件

所以譬如說我現在只有一個物件碰到**player**，然後存在**tmp**裡了

那我就可以把**tmp[0]**當作是那個物件

所以你打**tmp[0]**->setPos(0,0)

那個碰到的物件就會被傳到(0,0)的位置

或是你打scene->removeItems(**tmp[0]**)

那個碰到的物件就會從場景中被移除

對了，剛剛說**QList**是找到那些碰到的**Qgraphicsitem**物件

那麼**凡是有繼承Qgraphicsitem**的類別都能被放進**QList**裡操作

所以**QgraphicsPixmap** 之類的，或是你**自己寫的類別**也都可以!

範例程式碼大致如下!

