**Task 0: Adding a boolean delete\_mode\_ field to your project**   
-----------------------------------------------  
1) Add a boolean delete\_mode\_ field to your PlotWindow class. Don't forget to initialize this field in your PlotWindow constructor file and add any appropriate getters/setters! Did you make this field static or non-static?

2) Attach the "Delete Points" button's clicked signal to a slot that changes the value of the PlotWindow's delete\_mode\_ field to the opposite of the value that it previously had. Use the ui->centralWidget->setCursor() method ([http://doc.qt.io/qt-5/qt.html#CursorShape-enumLinks to an external site.](http://doc.qt.io/qt-5/qt.html#CursorShape-enum)) to change the appearance of the cursor to something else when the user is in delete mode. Make sure to change it back when the user is no longer in delete mode!

#### **Task 1: Emitting a DeletePoint signal from Point** **------------------------------------------------**

1) When a Point is clicked while the PlotWindow is in delete\_mode\_, emit a new signal with the following signature:  
void DeletePoint(Point \* p)  
Remember that you'll need to declare this signal in the Point's header, then you can emit it in Point::mousePressEvent.

Copy + paste the code below.

2) Create a corresponding slot in PlotWindow. Make sure to connect each Point's DeletePoint signal to the slot that you create in PlotWindow. In this slot, we will need to call the scene->removeItem(QGraphicsItem \* item) method to delete the Point. Do this.

Copy + paste your slot's code here.

#### **Task 2: Make something new happen when a Point is shift-clicked!** **-------------------------------------------------**

1) In the mousePressEvent in Point, add a conditional that something of your choice when the user shift-clicks a point. You can access the modifiers of the event that triggered this slot by using event->modifiers() method. To see if a certain modifier was used, you can test if the flags return == Qt::ModifierYouAreLookingFor.

You may find looking at [https://doc.qt.io/qt-5/qt.html#KeyboardModifier-enumLinks to an external site.](https://doc.qt.io/qt-5/qt.html#KeyboardModifier-enum) useful.

Take a look at the documentation for the modifiers() method--what is one way that you can break it?

Copy + paste the corresponding code here.

#### **Task 3: Drawing lines between two points** **----------------------------------------**

Our next goal is so that when you shift-click two points in a row, a line gets drawn between them. When you shift click a 3rd point, a line should be drawn between the 2nd and 3rd points. Implement this. Feel free to add field(s) as necessary to the PlotWindow to help keep track of the previous point(s) that have been clicked.

Use QGraphicsScene's addLine method to achieve this. If you were to delete one (or more) of the points involved in your lines, the lines will still be there. That is okay.

Copy + paste your slot that draws the line here.

#### **Task 4: Improve the UI** **----------------------**

Do one thing to improve the UI. For instance, make it more obvious when you are in delete mode. Make it more obvious how to interact with the interface. Make the interface layout more intuitive, etc. Make it more obvious which point you have previously shift-clicked on. etc.

Copy + paste the relevant code here.