

esc


F1


F2


F3

~
·

!
1

@
2

3

¥
4

→|

Q

W

E

•
中/英

A

S

D

↑

Z

X

fn

^
control

⇧
option

⌘
command

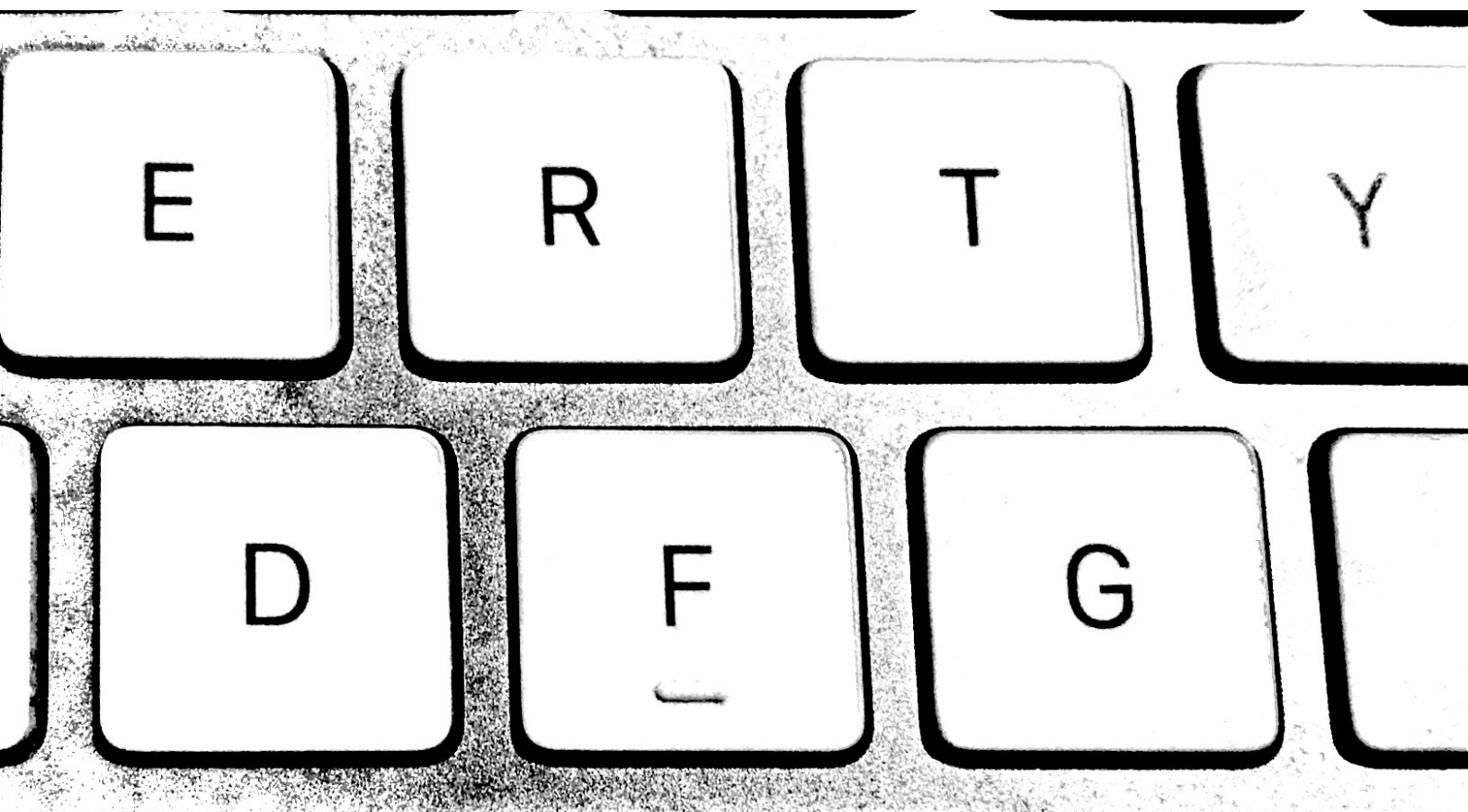
博扬诚信北京总

ASSURANCE. I WISH YOU
Genuine Guarantee

座82

理

ASSINSS



const QVector<QPointF> &data)

曲线 ID 中的数据。那些数据

函数设置了用于给定曲面的新的曲线数据替换。在 `QVector<QPoint>` 中，`QVector` 是 `QVector` 的别名。

`setCurveData()` 那么将会用 `QMap`

清除给定的曲线。

```

    {
        curveMap.remove();
        refreshPixmap();
    }
}

// 中移除一条
const
// 中移除一条

```

`Curve()` 函数可以以从 `curve`

QSize plotter.
return QSize(6 * margin, sizeHint()) 函数相似。空一个窗口

imumSizeHint()函数与
imumSizeHint()可以指定
小提示还要小的大小
(50)

理想大小一样, minimum
修改为比它的最小大小, 因为 Margin = 200
200 × 200 (因为这个大小, 我们

这个值是 300。如果小于这个值，那么...



OUTA
欧泰卫
用心服务



菊花决明子茶

精选六种草本食料，沸水中泡，
汤色清亮，经典古方，科学配比。

净含量：120克（4包×30）

NG Technology Co., Ltd.

并且显示出来,而通过调用 `refreshPixmap()` 可以更新当前的显示结果。

```
void Plotter::zoomIn()
{
    if (curZoom < zoomStack.count() - 1) {
        ++curZoom;
        zoomInButton->setEnabled(curZoom < zoomStack.count() - 1);
        zoomOutButton->setEnabled(true);
        zoomOutButton->show();
        refreshPixmap();
    }
}
```

如果用户在此之前已经放大过图形并且又缩小了图形,那么用于下一缩放级数的 `PlotSettings` 将会放在这个缩放堆栈中,因而就可以放大图形。(此外,还有可能通过橡皮筋选择框来放大图形。)

这个槽可以把 `curZoom` 移动到缩放堆栈中更深的位置上,它会根据是否允许把这个图形放得更大来决定 `Zoom In` 按钮是生效还是无效,并且会启用和显示 `Zoom Out` 按钮。此外,会再次调用 `refreshPixmap()`,以便让这个绘图区能够使用到最新的缩放设置值。

```
void Plotter::setCurveData(int id, const QVector<QPointF> &data)
{
    curveMap[id] = data;
    refreshPixmap();
}
```

`setCurveData()` 函数设置了用于给定曲线 ID 中的数据。如果一条曲线的 ID 与 `curveMap` 中已经存在的 ID 相同,那么将会用新的曲线数据替换原有的那些数据;否则,只是简单地插入新的曲线。`curveMap` 成员变量的类型为 `QMap<int, QVector<QPointF>>`。

```
void Plotter::clearCurve(int id)
{
    curveMap.remove(id);
    refreshPixmap();
}
```

`clearCurve()` 函数可以从 `curveMap` 中移除一条给定的曲线。

```
QSize Plotter::minimumSizeHint() const
{
    return QSize(6 * Margin, 4 * Margin);
}
```

`minimumSizeHint()` 函数与 `sizeHint()` 函数相似。就像 `sizeHint()` 函数可以指定一个窗口部件的理想大小一样, `minimumSizeHint()` 可以指定一个窗口部件理想的最小大小。布局不会把一个窗口部件的大小修改为比它的最小大小提示还要小的大小。

我们返回的这个值是 300×200 (因为 `Margin = 50`),这可以在 4 条边上留出一些空白区域,也可以为图形本身留出一些空间。如果小于这个大小,那么该绘图区就会显得太小了,也就没有什么用处了。

```
QSize Plotter::sizeHint() const
{
    return QSize(12 * Margin, 8 * Margin);
}
```

在 `sizeHint()` 中,我们返回一个和边白常量 `Margin` 成比例的“理想”大小,它们会形成合适的 3:2 比例,就像 `minimumSizeHint()` 中用到的比例一样。

至此,就完成了对 `Plotter` 公有函数和公有槽的回顾。现在,来看一下处于 `protected` 段中的事件处理器。