# 一、项目经验

## <1>关于流媒体开发

### －需求分析

## <2>物体位置信息以及标牌避让

### －需求分析

|  |  |  |
| --- | --- | --- |
| **具体需求** | | **性能要求** |
| 600移动的物体及其轨迹相对于中心点的位置展示在坐标轴上 | | 不卡顿 |
| 与600移动的物体(可以看成一个移动的点)一一对应相连的600个标牌以及连线 | |
| 标牌与运动点之间交互关系的处理 | 双击标牌可以对运动点的信息进行显示 |
| 选中或者拖拽标牌（标牌与运动点的连线可以实时改变） |
| 右击标牌弹出菜单可以对运动点进行设置（比如让运动点进行居中显示） |
| 600个标牌需要进行避让 | |

### －重构原因

１：QWT的限制

２：新性能以及新需求的要求

### －QWT简介

基于qt库的一个开源的绘制2维的统计图的库

它继承自QFrame和QwtPlotDict，QFrame提供一个QWidget的框架，QwtPlotDict为QwtPlot管理在其中的plot items，就是绘制的项。在QwtPlot上我们可以绘制无限多个的plot items，这些plot items可以是曲线，标记，格子以及继承自QwtPlotItem的子类。一个QwtPlot可以有四个轴，每个plot item连接到x和y轴上。在轴上的比例变换可以使用QwtScaleDiv，对于plot items比例可以使用QwtScaleEngine来计算，在每个轴上，QwtScaleEngine可以被单独设置。

在QwtPlot中有两个枚举类型。

Axis，轴，5个值，一个QwtPlot除了x和y，还有top和bottom轴，第五个是axisCnt，轴数，枚举从0开始，第五个为4，说明一共四个轴。另一个是LegendPosition，图例的位置。

它有五个值，分别指定插入一个图例仔什么位置，四个都是和x和y轴的位置有关，最后一个是特殊的，它允许不在这个Plot中，就是外部的。

### －难点攻破

1. 性能不足

|  |  |  |
| --- | --- | --- |
| **渲染方法** | **优点** | **缺点** |
| QWT | 提供 |  |
| paintEvent |  |  |
| Label->setPixMap |  |  |
| 其他 |  |  |

1. 坐标轴转换

需要将移动的物体（以下称动点）的经度、纬度、高度转换为相对于中心点的相对位置

即动点的RH，所以该需求实际上将动点以及其轨迹相对于中心点的距离和高度描述在坐标轴上提供给用户，而附属在动点上的标牌则是提供给用户更多的信息，以及给用提供其它的操作，这个具体的公式参照百度（北偏东坐标系）。

需要将RH即坐标轴的点转换为实际渲染点，QT坐标系以对应窗口的左上角为坐标原点，但是用户希望看到的坐标轴是以左下角为坐标原点，所以需要将坐标轴对应的点转换为实际渲染到窗口的点；如下图需要实时记录该红点的对应的坐标轴的（x,y）,因为这是一个虚拟的坐标，所以将其称为（Vx，Vy）；对应的实际渲染到该窗口的坐标点称为（Ax，Ay）

我们需要实时的记录（Vx，Vy）,（Ax，Ay），（Vx，VyMax）（Ax，AyMax），（VxMax，Vy）（AxMax，Ay）当我们收到一个目标点，假如它对应的坐标轴为（Vx1，Vy1）

第一步我们需要知道判断以下是否在坐标的范围内（黄色方框内），即Vx< Vx1 < VxMax和Vy< Vy1 < VyMax;

第二步我们需要通过（Vx1，Vy1）来求它实际的渲染坐标（Ax1，Ay1）？

思路分析：我们需要求（Ax1，Ay1），已知（Ax，Ay），假如我们求得（Ax1，Ay1）到x轴和y轴的距离ADisX,ADisy，然后Ax1 = Ax + ADisX；Ay1= Ay - ADisy；因为方向是相反的所以用减号。

那么我们需要求ADisX,ADisy？

现在我们分析以下已知条件

（Vx，Vy）,（Ax，Ay）：红色点对应的坐标轴的坐标，以及对应的相对于QT坐标系的坐标。

（Vx，VyMax）（Ax，AyMax）：y轴最大值那个点对应的坐标轴的坐标，以及对应的相对于QT坐标系的坐标。

（VxMax，Vy）（AxMax，Ay）：x轴最大值那个点对应的坐标轴的坐标，以及对应的相对于QT坐标系的坐标。

（Vx1，Vy1）：目标点

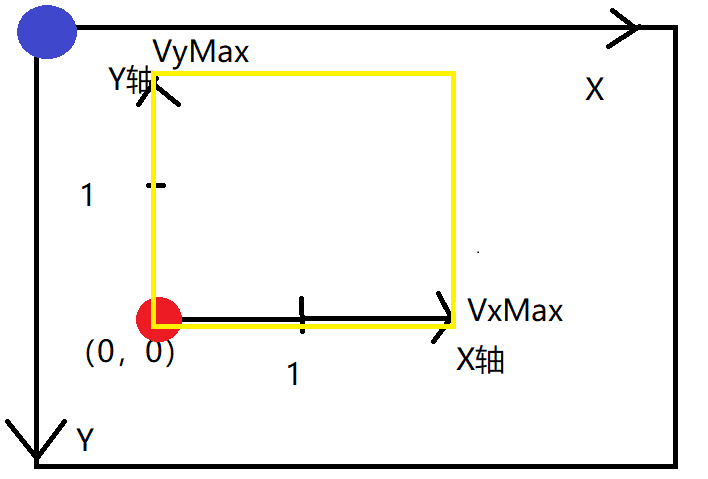
y轴最大值那个点称为yP点，x轴最大值那个点称为xP点，需要求的点为sP点.

sP到x轴实际的距离/sP到x轴坐标轴的距离 = yP到x轴实际的距离/yP到x轴坐标轴的距离

sP到y轴实际的距离/sP到y轴坐标轴的距离 = yP到y轴实际的距离/yP到y轴坐标轴的距离

sP到x轴实际的距离为ADisX，ADisX = （yP到x轴实际的距离/yP到x轴坐标轴的距离）\* sP到x轴坐标轴的距离 = （AyMax/VyMax）\* Vy1

sP到y轴实际的距离为ADisY，ADisY= （yP到y轴实际的距离/yP到y轴坐标轴的距离）\* sP到y轴坐标轴的距离 = （AxMax/VxMax）\* Vx1



1. painEvent放弃

painEvent()缺点

4.8调用update()函数效率比较低当拖拽鼠标或着鼠标滚轮且渲染内容较多的时候主动调用update()函数时，无法满足实时性（有卡顿）。提供的解决方法：开启timer（可以做一个线程的timer），加载绘图设备（QPixmap）在QPixmap上绘图，调用QLable加载QPixmap

1. 拖拽以及放大缩小等操作

拖拽坐标轴，放大缩小坐标轴。

拖拽标牌，标牌右击标牌弹出菜单，双击显示详标牌

详标牌内容动态的显示或者隐藏

详标牌内容扩展（不修改代码）通过xml

用户可以通过修改配置文件(xml)来更改物体轨迹的颜色，标牌字体的配色标牌和物体连线的配色等等

4:标牌避让

### －收获

## <３>框架搭建

功能：

## <４>Ｃ＋＋编码规范

### －变量初始化

一定要注意变量初始化，否则容易出现一些意想不到的问题。

指针初始化未指空或者delete(free)为指空，导致野指针的问题。

用于控制代码流向的变量未初始化导致，代码流向不确定。

int flag;

if( flag >= 1 )

{

//Do something;

}

else if

{

//Do something;

}

其中vs的debug会给代码初始化，而release则不会容易导致问题出现。

为避免变量未初始化一定要主意以下地方

**普通变量声明的时候便初始化**

**成员变量一定要在构造函数将其初始化**

**结构体要带有默认的构造函数**

### －容器越界

使用容器一定要主意保护边界，否则很容易造成崩溃。

访问map之前一定要判断其是否有该key值

访问list或者vector一定要其边界，是否为空？是否访问了容器最大值之外的值。

如果容器可能会因为程序运行变得越来越大，一定要设置极限值，当超越该值的时候及时出该容器。

### －代码冗余

无效代码，伴随者程序开发一定会出现一些无效的变量或者无效的逻辑代码，一定要及时**注释甚至删除**，否则会导致后续代码不易维护，以及增加二次开发时，代码解读的难度。

重复代码，相同的代码逻辑可以将其抽成函数，结构一样的代码可以用模板。

### －代码重构

## <５>**QT常见的误区**

### －信号发出槽函数未响应的原因

未添加关键字Q\_OBJECT，没有Q\_OBJECT，类中的slots,signal,emit等都不会被解析。

处理函数没有定义为槽函数。

检查connect()函数是否连接成功，连接过程中，只需要将参数的类型加上即可，不能加变量。

检查信号和槽函数中的类型QT是否识别，如果不识别**需要注册一下**，注意这个时候connect()函数会返回**true**,但是信号发送时槽函数没有响应。

### －QT的API使用技巧

QT的帮助文档提供了很多的便利，要善于利用阅读qt的帮助文档来解决问题，例如要使用QT的QTreeWidget，一定要通读QT和QTreeWidget相关的内容。

## **<6>**Ｃ＋＋的bug调试经验

# 面试技巧

## <1>关于离职原因

### －不得体答案

### －备选答案

## <２>薪资要求

## <３>应聘者提问