Refuse Classification

——啦啦啦队 qt 大作业报告

一．程序功能介绍

1.垃圾词条库

在词条库的基础上我们实现了对于某种垃圾所属类别的搜索：首先，用户应先选择所在城市，接着在界面最上方的搜索栏中输入想要知道所属类别的垃圾名称，若词条库中包含与输入相关或相对应的词条，则会将这些词条从词条库“挑选”出来并展示给用户。若不包含对应的词条，则会显示诸如“数据库里暂时没有收录这种垃圾”的字样。值得一提的是，当输入比较模糊的类别关键词（如纸）时，所有与之相关的垃圾词条都将被展示出来。

此外，我们也允许用户自定义添加词条，在界面中会有一个”添加垃圾”的按钮，选择这一按钮即可进入添加词条的界面。为了将词条加入词条库中，用户可以手动输入垃圾的名字和以及垃圾类别。删除和修改自定义垃圾则需要手动输入垃圾 id、垃圾名称、垃圾类别。

2.校园垃圾桶向导

当用户选择这一功能后，首先会呈现北京大学校园的平面图，我们会将平面图分为四个区域，用户可根据自己的实际情况选择具体的区域（如未名湖周边）。在用户选择以后会弹出对应区域的标注有垃圾桶具体位置的平面图。在区域图中，我们将用带有不同颜色的符号来区分不同的垃圾桶（如蓝色代表可回收垃圾桶，红色代表有害垃圾

等）。

3．问答挑战

这一功能允许用户进行垃圾分类相关常识的自我检测。我们准备了问答题库，进入这一功能后，用户首先应选择标准模式 or 限时模式，不论哪种模式，都会从题库中随机挑选若干道题。（标准模式下有 10 道题，限时模式不限）我们将采用每次顺序向用户展示一道题的形式实现这一功能，当前题目回答完毕以后选择“下一题”按钮即可跳转至下一题，进行下一题的回答，直至所有题目回答完毕以后可以选择交卷。（标准模式下还会向用户展示本题的正确答案和作答正误情况。）最后向用户展示此次作答答对题数，用户还可获得相应的“成就”，这在“解锁成就”一栏中可以看到。

二．项目各模块实现细节

1.主界面

mainwindow 将三个按钮分别与对应的 lambda 函数（作为槽函数）连接，如：

```
connect(button2,&QPushButton::clicked,this,[](){
        MapWidget*function2=new MapWidget;
        function2->show();
});
```

2.功能一

（1）makedic.hpp 实现基本逻辑

分别用 NameToId、GarbageidToTypeid、TypeidToTypename 这三

个 map 实现了从垃圾名称到垃圾 ID、从垃圾 ID 到垃圾类别 ID、从垃圾类别 ID 到垃圾类别名称的三个映射。

```
map<string,int> NameToId;//把垃圾的名字转成 id
map<int,int> GarbageidToTypeid[4];//把垃圾的 id 转成城市里面垃圾种类的 id
map<int,string> TypeidToTypename[4];//把城市里面垃圾种类的 id 转换成字符串
```

主要包括以下函数：

- work()

从 garbage.csv 文件中读取垃圾名称和分类信息，将其存入 NameToId 和 GarbageidToTypeid 中。初始化不同城市的垃圾类型名称映射 TypeidToTypename。

- search()

根据用户输入的城市和垃圾名称，搜索并输出垃圾在该城市的分类信息。

- findgarbage(string s, int cityid)

根据输入的垃圾名称和城市 ID，查找并返回垃圾的分类信息。

- add(string name, int sort),changesort(int id, int sort) 和 del(int id)

添加新的垃圾条目到系统中，并更新相关映射，修改指定垃圾条目的分类或删除垃圾条目。

- listcustom(int cityid)

返回特定城市中用户自定义的垃圾列表和其分类信息。

（2）widget.h、widget.cpp、widget.ui 实现功能一界面

发挥作用的槽函数主要有：

- void Widget::on_choosecity_currentIndexChanged(int index)

  当城市选择下拉框的选择项发生改变时触发。根据选择的城市设置 cityid，并根据城市是否合法来启用或禁用确认按钮和添加垃圾按钮。

- void Widget::on_confirm_clicked()

  当确认按钮被点击时触发。获取用户输入的垃圾名称(ui->entergarbage->toPlainText()) 和选择的城市 (cityid)。调用 findgarbage() 函数查找垃圾分类信息，并将结果显示在 ui->showgarbage 控件中。如果 findgarbage() 返回空结果，显示提示信息。

- void Widget::on_addgarbage_clicked()

  当添加垃圾按钮被点击时触发。创建并显示 customdic 对话框，用于用户手动添加垃圾信息的功能。

（3）customdic.h、customdic.cpp、customdic.ui 实现自定义垃圾界面

以修改自定义垃圾条目为例，添加与删除同理：

- void customdic::on_entername_textChanged()

当垃圾名称输入框内容发生变化时触发。如果输入框不为空，启用新增按钮 (ui->new_2)。

- void customdic::on_change_clicked()

获取用户输入的垃圾 ID 和分类。如果 ID 大于等于 begincustom（表示是自定义垃圾条目），调用 changesort(id, sort) 函数修改该条目的分类。调用 showlist(ui) 函数更新列表显示。清空输入框，并重置按钮和分类下拉框的状态。

3.功能二：

（1）MapGraphicsItem 类（实现美观）

MapGraphicsItem 类是 QGraphicsPixmapItem 的子类，处理以下事件：

- 悬停事件

当鼠标悬停在项目上时，项目会稍微放大，并应用一个蓝色阴影效果；当鼠标离开项目时，项目恢复原始大小，并移除阴影效果。

```
void MapGraphicsItem::hoverLeaveEvent(QGraphicsSceneHoverEvent *event)
{
    m_isHovered = false;
```

```
    setScale(1.0);
    setGraphicsEffect(nullptr);
    QGraphicsPixmapItem::hoverLeaveEvent(event);
}
```

- 鼠标按下事件

当点击项目时，检查点击的坐标。根据坐标，显示不同的区域窗口（AreaA、AreaB、AreaC、AreaD）。

```
void MapGraphicsItem::mousePressEvent(QGraphicsSceneMouseEvent *event) {
    QPointF scenePos = mapToScene(event->pos());
    if (scenePos.x()  >= 260 && scenePos.x()  <= 540 && scenePos.y()>= 5
&& scenePos.y()<= 130) {
        a->show();
//此处的示例代码仅以区域 a 的显示为例，省略了其他区域的显示
    QGraphicsPixmapItem::mousePressEvent(event);
}
```

（2）MapWidget 类

MapWidget 类初始化主要地图视图，并定义交互区域的行为：

- 初始化 QGraphicsScene 并将其设置为视图的场景。
- 加载主地图图像 (mapPixmap) 并将其添加到场景中。
- 创建交互区域 (MapGraphicsItem)，将它们定位在地图上，并添加标签。
- 初始化 AreaA、AreaB、AreaC 和 AreaD 对象，并将它们的 "back" 信号连接到 MapWidget 中的槽函数，以处理从详细视图返回的操作。

4.功能三：

（1）page1 实现功能三模式选择界面

（2）page2 实现标准模式

- init()函数

打开名为 QandA.csv 文件，并逐行读取其中的内容到 QStringList 变量 lines 中。使用 QTime 获取当前时间，结合 srand 和 rand 函数生成一个随机数 n，范围在 1 到 70 之间。根据随机数 n，从 lines 中选择特定索引的数据，分别赋值给 question、answerA、answerB、answerC、answerD、correctanswer 和 reason 变量。

- addlayout()函数

将各个控件按照特定的布局方式进行排列，并将它们添加到 page2 窗口中显示。

- 页面切换

通过 nextpage 按钮的点击来实现页面切换逻辑。当用户点击 nextpage 按钮时，会触发 pagechanged 信号，并传递相关参数给连接的槽函数 switchpage。

（3）page3 实现限时模式

与标准模式基本相同，在此基础上增加了计时器及计时器闪烁这一机制，实现细节如下：

- connect 函数将 timer 对象的 timeout 信号与 page3 类的 updatetime 槽函数关联起来。每当计时器超时（即每秒），都会

调用 updatetime 函数来更新剩余时间的显示。

```
connect(timer, &QTimer::timeout, this, &page3::updatetime);
timer->start(1000);
```

- updatetime 函数是处理计时器事件的关键部分。它首先计算剩余的秒数 secondsLeft，如果剩余时间为 0，就停止计时器、停止闪烁定时器、关闭窗口，并发射 timeup 信号。

- handleflash 函数用于处理剩余时间为 5 秒时的闪烁效果。它通过修改 QLCDNumber 控件 Time 的文本颜色来实现闪烁效果，通过定时器 flash 来控制闪烁的频率和可见性的切换。

```
    QPalette lcdpat = Time->palette();
    if (visible) {
        lcdpat.setColor(QPalette::Normal,QPalette::WindowText,
Qt::blue);
    } else {
        lcdpat.setColor(QPalette::Normal,QPalette::WindowText,
Qt::red);
    }
    Time->setPalette(lcdpat);
    visible = !visible; // 切换可见性
```

（4）page4 实现解锁成就

通过读写 achievement.csv 文件实现成就的存储。在 QListWidgetItem 中可为每个 item 设置 icon，当读取当前 item 对应为 False 时，设为禁用状态的 icon；读取当前 item 为 True 时，设为标准 icon，从而实现 icon 的点亮。

三．小组分工情况

1.资料准备与实地考察阶段

（1）为确定校园垃圾桶分布，将校园划分为三个部分，三个同学分别负责其中一部分。

（2）另外，罗玥萦查找并搜集了关于垃圾分类的问答题，建立了问答题库；栾上博搜集了常见垃圾名称及类别，建立了垃圾库。

2.功能实现阶段

（1）功能一：栾上博　　功能二：罗玥萦　　功能三：梁书怡

（2）之后，梁书怡合并了三个功能并实现了 mainwindow 及其美观。

3.完善阶段

建立了材料库，三人继续实现各自所负责的功能部分的美观。最终，经过大家的进一步细化与完善实现了最终版。

四．项目总结与反思

1.组内合作的重要性

要想做好 Qt 大作业，组内成员的分工合作是至关重要的。在分工之前，任务量看上去很多，实现起来好像也比较困难，但在分工落实到组内的每个成员之后，每个人需要承担的任务量就很少，实现起来也变得容易。正是由于我们队伍分工明确，决策透明开放，团队沟通有效，才成就了这个属于我们三个人的小程序。

2.拒绝拖延，迎接高效

第一次线下见面确定主题和第一次分工内容，我们队伍就非常高

效，在每个人发表了自己的看法后，确定了此次 qt 大作业的主题与要实现的基本功能，并确定了第一阶段的分工。五一假期期间，大家也是非常负责地完成了自己的部分。接着迎来第二次线下分工，这次分工更是迅速，仅仅用了半个小时左右。回顾整个过程，我们队伍之所以能圆满完成此次大作业，起到关键作用的是队伍成员之间的高效沟通与 ddl 的设置。在 ddl 的监督之下，每个成员都高效且高质量地完成了自己负责的部分。

3.在一些细节上做得还不到位

与参与路演的其他队伍相比较，我们队伍还存在一些不足：虽然我们的立意很好，但是具体实现的功能显得有些单一；三个功能的实现仅仅是依赖于文件的读写，有待改进；与其他优秀队伍相比，程序能发挥的实际作用可能比较小等等。我们将汲取此次大作业的经验，后续进行进一步的开发与完善。