Image Beautify 模块报告之View层

报告人：黄文璨

报告日期：2018年7月23日

# 一、MVVM模式中View层简介

MVVM架构使得View层与Model层完全实现解耦合，View层专门负责UI界面的显示和用户交互逻辑，可以是控件容器和集合。View与ViewModel之间通过Data Binding进行数据绑定，通过Commands绑定操作的调用，当属性改变或命令执行时ViewModel向View发出通知（Send Notifications）。

由此可以发现，View层的主要工作有以下三个部分。

1、设计用户界面并实现用户交互，界面显示所需的数据、用户交互发出的命令都与ViewModel层进行绑定。

2、实现属性通知的接收器，当ViewModel发出属性改变的通知时调用View层暴露的接口实时更新界面显示。

3、实现命令通知的接收器，当ViewModel发出命令执行情况的通知时调用View层暴露的接口实时进行反馈与更新界面显示。

# 二、本项目中的View层实现

## 2.1 View层概览

|  |  |
| --- | --- |
| View层文件结构 | 描述 |
| 屏幕快照%202018-07-24%20下午2.49.57.png | 在ImageBeautify项目中，View层有一个主窗口类mainview，另外还有许多子窗口类在dialog文件夹中，并且如上一章所述需要实现两个接收器(sink)，分别用于接收属性改变的通知(mainimagesink)和命令执行情况的通知(maincommandsink)。 |
| 屏幕快照%202018-07-24%20下午2.50.25.png | 左图为本项目中View层实现的部分子窗口类，都放在dialog文件夹中。 |

## 2.2 View与ViewModel间的数据绑定与命令绑定

|  |  |
| --- | --- |
| 属性绑定(data binding) | |
| View | ViewModel |
| View中的数据指针shared\_ptr | ViewModel中对应的数据指针 |
| 屏幕快照%202018-07-24%20下午3.01.03.png | 屏幕快照%202018-07-24%20下午3.03.34.png |
| View设置数据绑定入口函数 | ViewModel将属性暴露出来 |
| 屏幕快照%202018-07-24%20下午3.01.17.png | 屏幕快照%202018-07-24%20下午3.04.47.png |

|  |  |
| --- | --- |
| 命令绑定(commands binding) | |
| View | ViewModel |
| View中的命令指针（部分），类型为命令基类 | ViewModel中对应的数据指针（部分） |
| 屏幕快照%202018-07-24%20下午3.10.29.png | 屏幕快照%202018-07-24%20下午3.12.43.png |
| View设置命令绑定入口函数（部分） | ViewModel将属性（部分）暴露出来 |
| 屏幕快照%202018-07-24%20下午3.11.46.png | 屏幕快照%202018-07-24%20下午3.13.26.png |

## 2.3 View层属性通知接收器

View层的属性通知接收器负责响应ViewModel层发出的属性改变的通知并及时更新视图。接收器类与View通过对象指针的聚合实现相互调用。

MainImageSink需要实现基类中的OnPropertyChanged函数以响应属性的改变，具体实现时需要按属性类型分类调用View层的接口。

void MainImageSink::OnPropertyChanged(const propertyType pt)

{

if(pt == MAIN\_IMAGE){

mainview->update();

} else if(pt == SUB\_IMAGE){

mainview->updateSubImage();

} else if(pt == LOG){

mainview->updateLogManager();

}

}

class MainImageSink : public IPropertyNotification

{

public:

MainImageSink(MainView \*mainview);

virtual void OnPropertyChanged(const propertyType pt);

private:

MainView \* mainview;

};

View层通过将消息接收器暴露出去，并挂载到ViewModel的通知代理函数中，实现完整的通知与消息接收信号链。

|  |  |
| --- | --- |
| View | ViewModel |
| View中的私有信号接收器 | ViewModel继承了通知代理基类 |
| 屏幕快照%202018-07-24%20下午3.31.26.png | 屏幕快照%202018-07-24%20下午3.31.52.png |
| View暴露接收器的出口函数 | APP层中将接收器挂载到ViewModel的通知代理中 |
| 屏幕快照%202018-07-24%20下午3.31.40.png | 屏幕快照%202018-07-24%20下午3.32.04.png |

## 2.4 View层命令通知接收器

与属性通知接收器类似，View层的命令通知接收器负责响应ViewModel层发出的命令完成结果的通知并及时更新视图状态。接收器类与View通过对象指针的聚合实现相互调用。

class MainCommandSink : public ICommandNotification

{

public:

MainCommandSink(MainView \*mainview);

virtual void OnCommandComplete(const commandsType cmd, bool OK);

private:

MainView \* mainview;

};

MainCommandSink需要实现基类中的OnCommandComplete函数以响应命令执行完成的通知，具体实现时需要按命令类型分类调用View层的接口。

void MainCommandSink::OnCommandComplete(const commandsType cmd, bool OK){

switch (cmd) {

case OPEN\_FILE:

if(OK);

else mainview->HandleOpenFileException();

break;

(此处省略若干条命令类型)

default:

break;

}

}

View层通过将命令接收器暴露出去，并挂载到ViewModel的通知代理函数中，实现完整的通知与消息接收信号链。

|  |  |
| --- | --- |
| View | ViewModel |
| View中的私有信号接收器 | ViewModel继承了通知代理基类 |
| 屏幕快照%202018-07-24%20下午3.31.26.png | 屏幕快照%202018-07-24%20下午3.31.52.png |
| View暴露接收器的出口函数 | APP层中将接收器挂载到ViewModel的通知代理中 |
| 屏幕快照%202018-07-24%20下午3.31.40.png | 屏幕快照%202018-07-24%20下午3.32.04.png |