

现代操作系统应用开发实验报告

学号： 15331248

班级： 上午班

姓名： 潘承远

实验名称： Todos(HomeWork5)

一. 参考资料

- 《13-LiveTiles-new》: 了解磁贴大题概念;
- 《15-AppToAppCommunication》: 这个 PPT 把 “数据共享” 的知识讲得很全面, 就我而言, 我觉的作业第二部分 “数据共享” 要比作业第一部分 “磁贴通知” 简单很多;
- <https://docs.microsoft.com/zh-cn/windows/uwp/controls-and-patterns/tiles-badges-notifications>

这个网站教我学会如何 “本地通知”, 并提供了一些自适应磁贴的框架。最重要的是这个网站的末尾提供了 GitHub 链接, 链接是初级的 live tile 的 demo。

- <https://msdn.microsoft.com/zh-cn/library/windows/apps/xaml/hh868253.aspx>




这个网站是我的磁贴入门教程

- https://mva.microsoft.com/en-US/training-courses/windows-10-live-tiles-and-notifications-14585?l=XemBh4EsB_8105632527

这个网站是微软虚拟学院的网站, 上面提供了 live tile 的视频课程





实验步骤

- 仔细阅读课件与相关文档
- 下载可视化工具 “Notificatinos Visualizer” 并设计磁贴。这是我设计的磁贴类型



XML Payload

```
<image src =  
"http://pic.nipic.com/2007-11-15/2007111592158497_2.jpg"  
placement = "background"/>  
</binding>  
  
<binding template='TileMedium'>  
<text> Title</text>  
<text hint-style='subtitle'>Details</text>  
<image src =  
"http://pic.nipic.com/2007-11-15/2007111592158497_2.jpg"  
placement = "background"/>  
</binding>  
  
<binding template='TileWide'>  
<text hint-style='title'>Title</text>  
<text hint-style='subtitle'>Details</text>  
<image src =  
"http://pic.nipic.com/2007-11-15/2007111592158497_2.jpg"  
placement = "background"/>  
</binding>  
  
<binding template='TileLarge'>  
<text hint-style='title'>Title</text>  
<text hint-style='subtitle'>Details</text>  
<image src =  
"http://pic.nipic.com/2007-11-15/2007111592158497_2.jpg"  
placement = "background"/>  
</binding>  
</visual>  
</tile>
```



Pin tile to Start

Update most recent tile

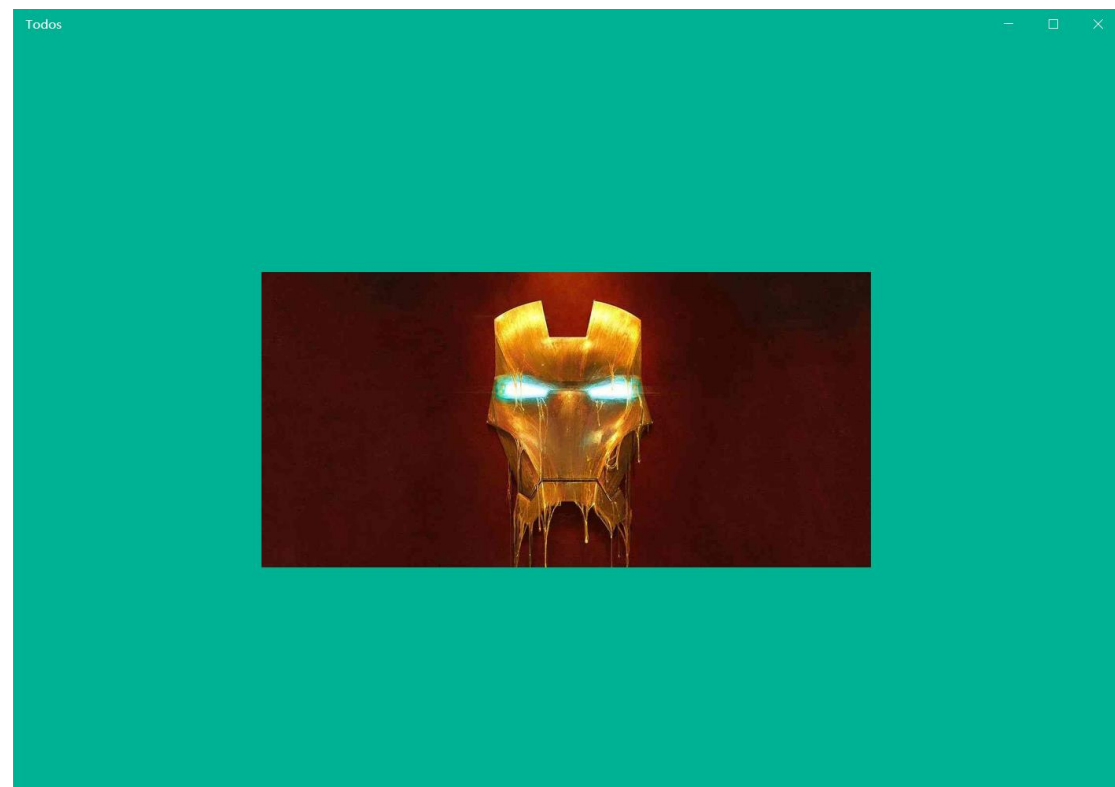
- 实现“磁贴更新”功能。我看了老师的 PPT，以及课件推荐的微软官网教程，并且仔细地看了一遍 Notificatinos Visualizer 的模板 xml 文件代码。然后自己便开始在 Notificatinos Visualizer 编写自己磁贴的 xml 代码。但是最后我选择将磁贴的 xml 代码翻译成 CSharp 代码加在 MainPage.xaml.cs 和 NewPage.xaml.cs 文件中。部分代码如下：

```
TileContent content = new TileContent()
{
    Visual = new TileVisual()
    {
        TileSmall = new TileBinding()
        {
            Content = new TileBindingContentAdaptive()
            {
                Children =
                {
                    new AdaptiveText()
                    {
                        Text = from
                    },
                    new AdaptiveText()
                    {
                        Text = subject,
                        HintStyle = AdaptiveTextStyle.CaptionSubtle
                    }
                },
                BackgroundImage = new TileBackgroundImage()
                {
                    Source = "Assets/taylor2.jpg",
                    HintOverlay = 20
                }
            }
        }
    }
}
```

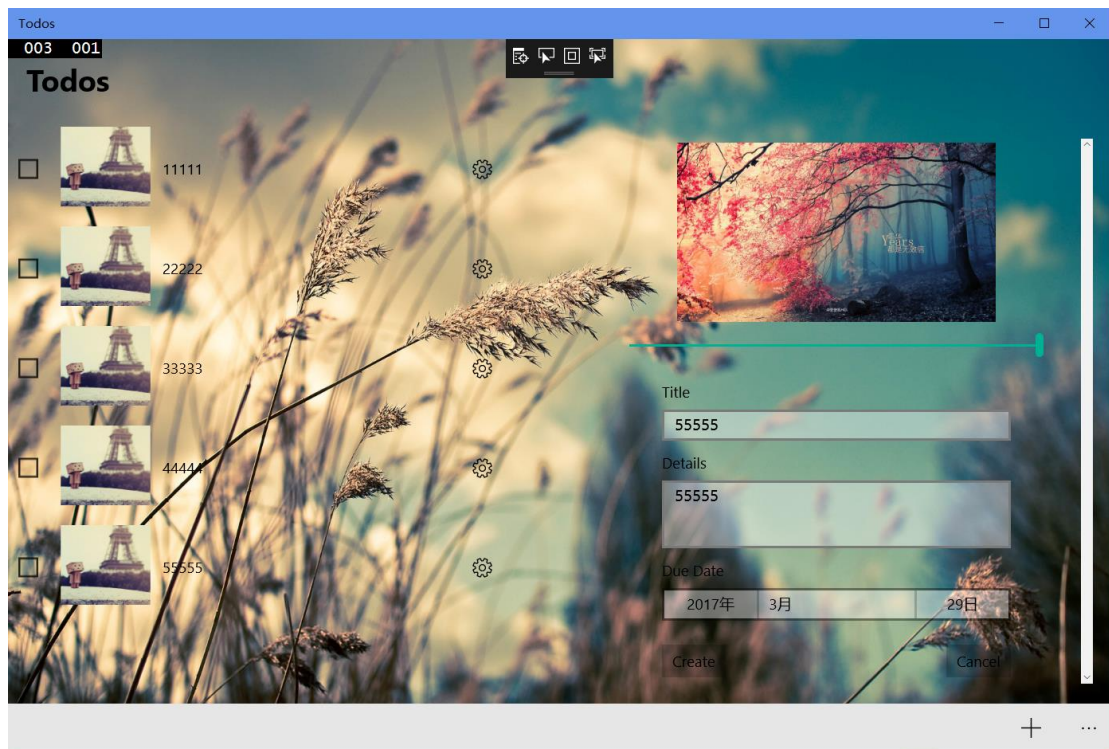
- 实现“文字共享功能”功能。
- 实现“图片共享功能”功能。

二．实验结果截图

1. 程序的初始屏幕



2. 输入 5 个待办事项



3. 这是开始界面的 Samll Adaptive Tile



4. 这是开始界面的 Medium Adaptive Tile



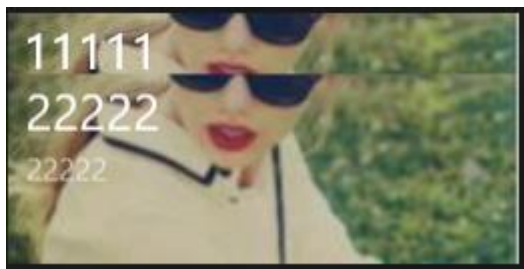
5. 这是开始界面的 Medium Adaptive Tile

从照片可以看到内容在循环展示（正在从“44444”更新到“55555”）



6. 这是开始界面的 Wide Adaptive Tile

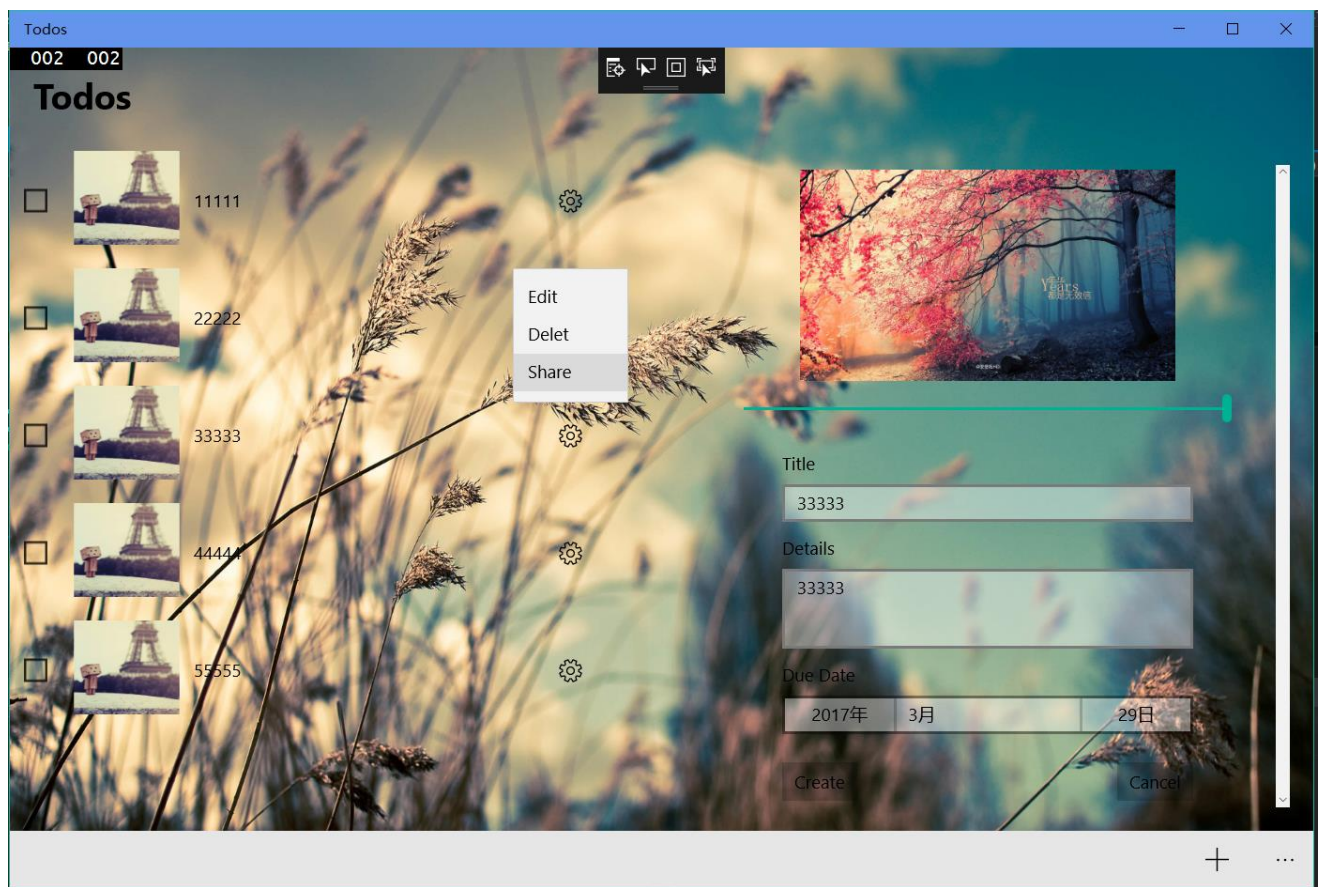
从照片可以看到内容在循环展示（正在从“11111”更新到“22222”）



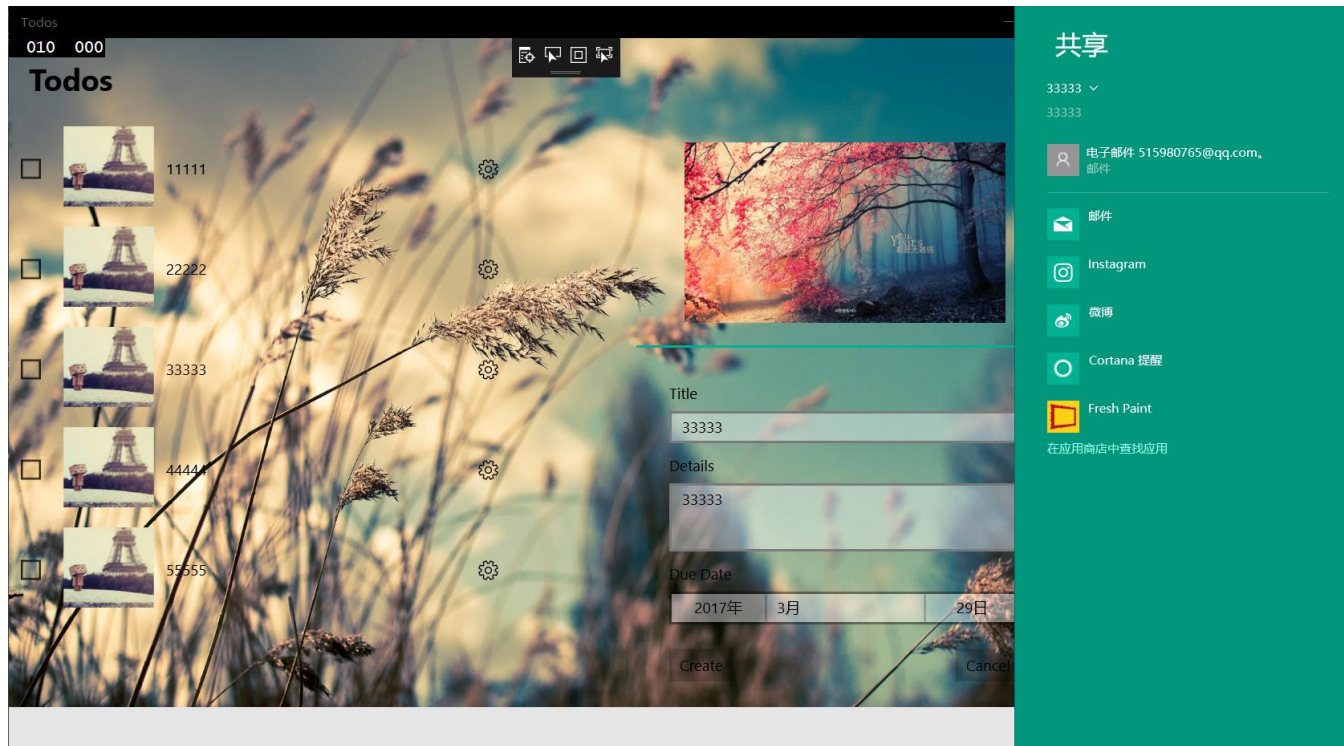
7. 当我添加“66666”的时候磁贴能进行更新，并且更新的内容循环展示（正在从“55555”更新到“66666”）



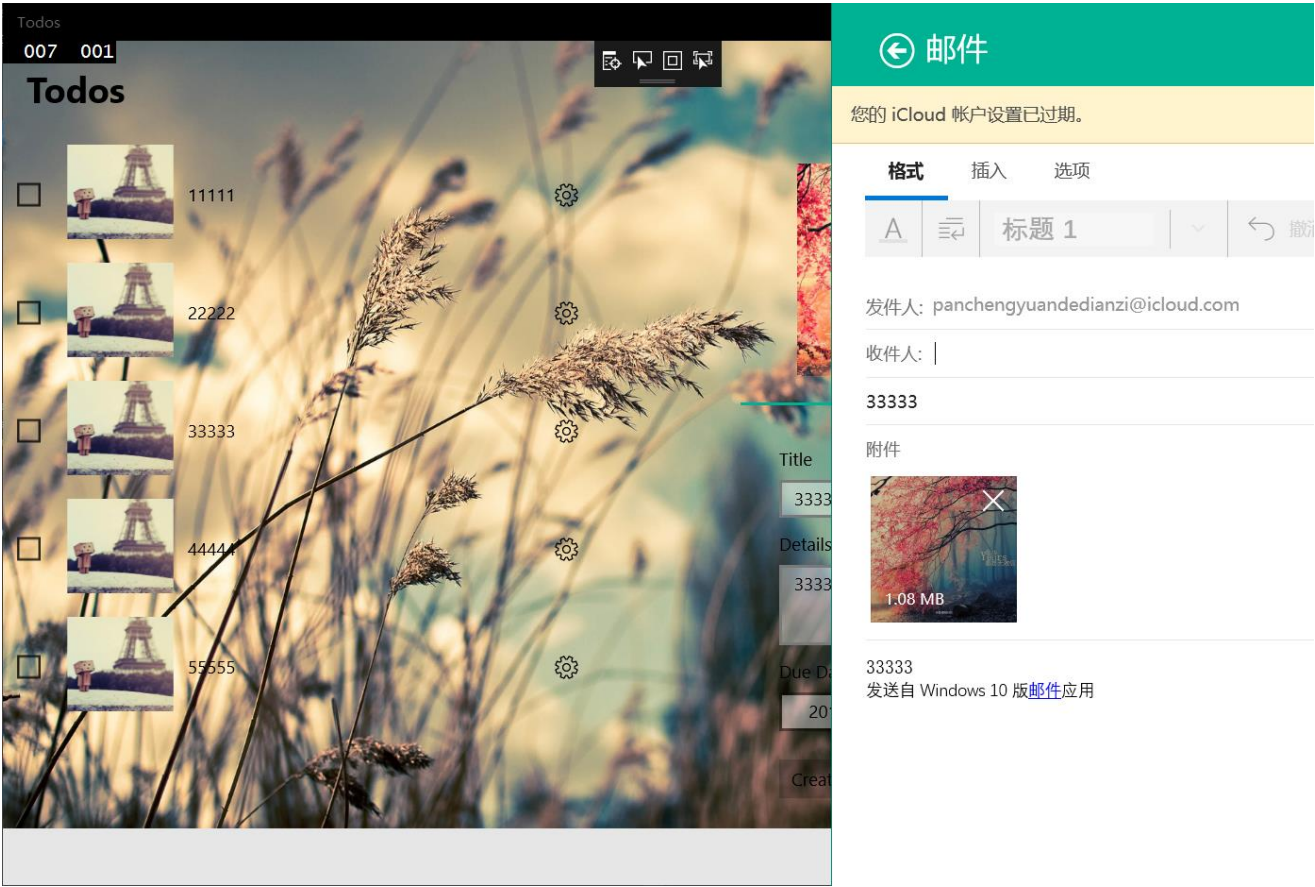
8. 分享待办事项 “33333”



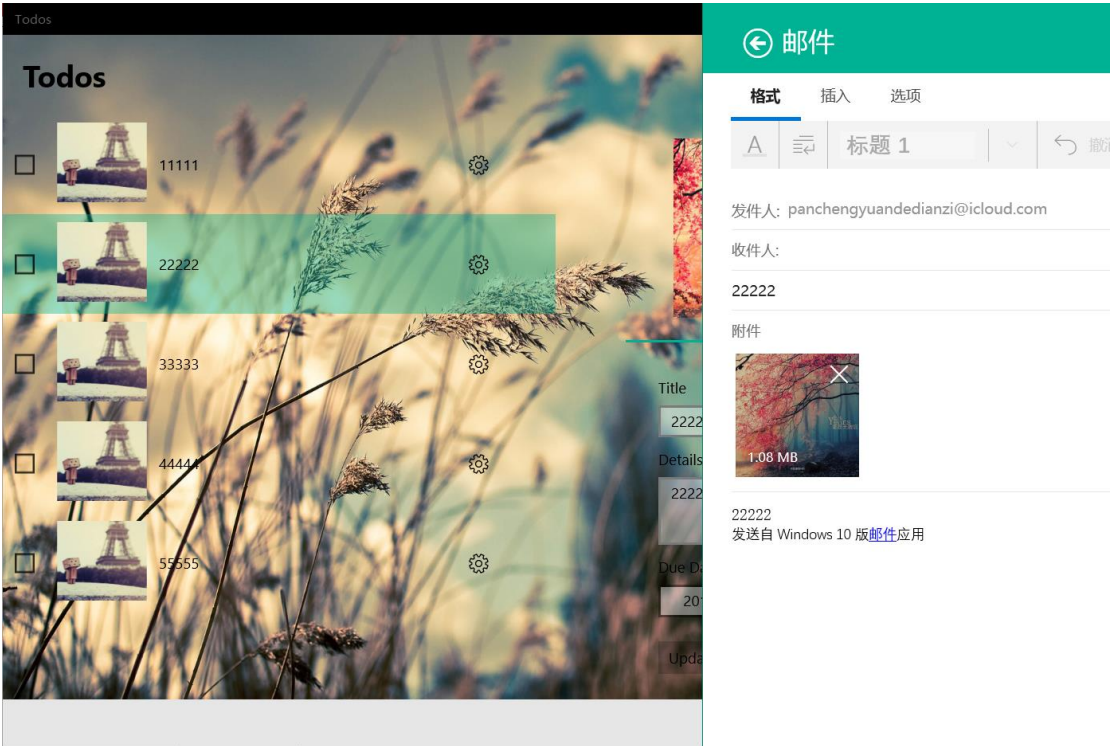
9. 显示共享 UI



10. 从邮件内容来看分享的 Title 和 Details 都正确（经过多次测试）
并且能够实现图片共享



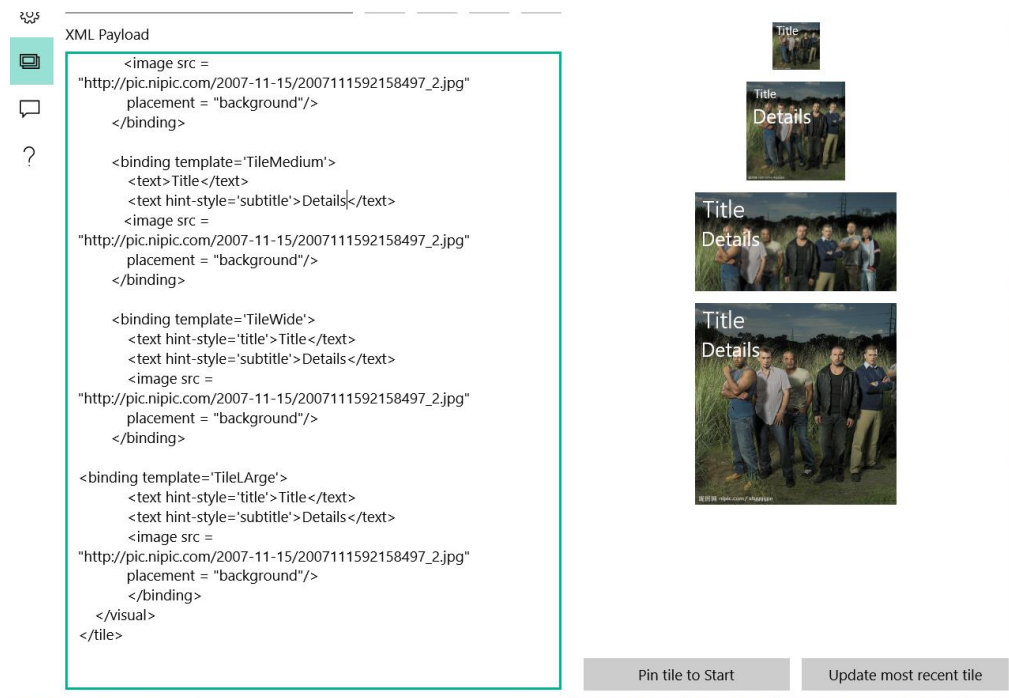
11. 用“22222”来分享，从邮件内容来看分享的 Title 和 Details 都正确（经过多次测试）
并且能够实现图片共享



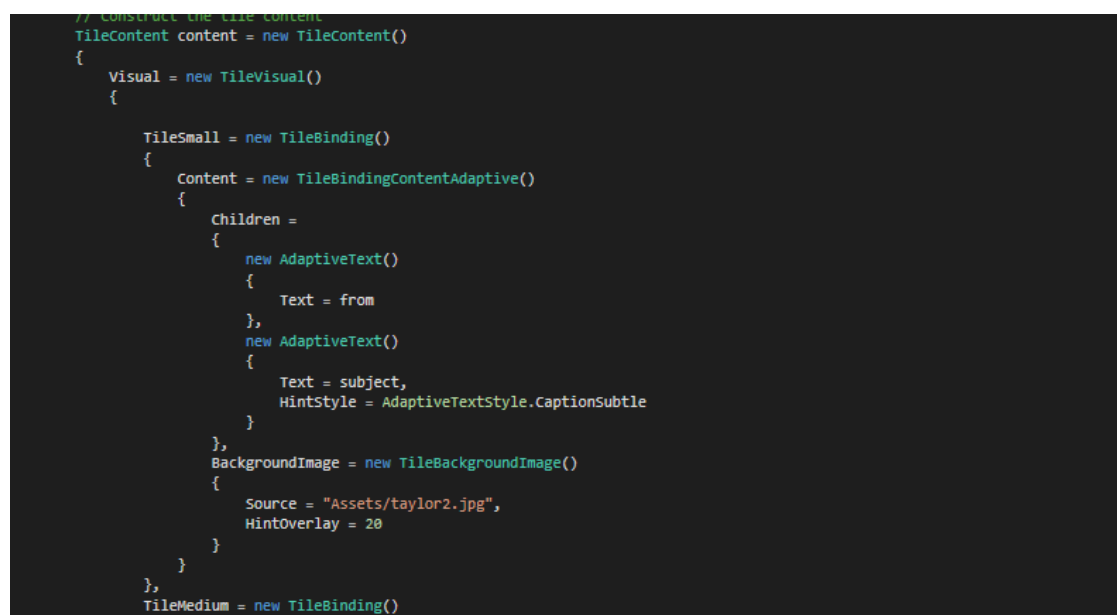
三 . 实验过程遇到的问题

- 在完成第一部分“磁贴的更新”的时候。我遇到的最大问题就是不明白如何使用在 Notifications Visualizer 上面已经写好的 xml 代码。PPT 以及 PPT 给的网址都没有点明如何“解析 xml”。最后自己采用的方法是用 CSharp 来完成作业要求，我将自己在 Notifications Visualizer 上面设计好的代码翻译成 CSharp 语言，然后将它们添加到 MainPage.xaml.cs 和 NewPage.xaml.cs 文件中。这样完美完成 part1 的要求。

这是 XML 代码



这是对应的 CSharp 代码（部分）



●在完成第二部分“App to App Communication”的时候，个人感觉比较简单。最开始我不知道如何点击 AaaBarButton 来获取当前点击的 Item。经过查询，我找到了解决方案，代码如下

```
private void Share_Click(object sender, RoutedEventArgs e)
{
    var dc = (sender as FrameworkElement).DataContext;
    var item = ToDoListView.ContainerFromItem(dc) as ListViewItem;
    ViewModel.SelectedItem = (Models.TODOItem)(item.Content);
    title_MainPage.Text = ViewModel.SelectedItem.title;
    description_MainPage.Text = ViewModel.SelectedItem.description;
    DataTransferManager.ShowShareUI();
}
```

代码的第 1、2、3 行就是获取当前点击的 Item (ViewModel.SelectedItem)，然后更新两个 TextBox 的值，这是因为要保证分享的 Title 和 Details 的内容正确（是当前的 Item 内容）。下图代码将解释原因

```
async void OnShareDataRequested(DataTransferManager sender, DataRequestedEventArgs args)
{
    var dp = args.Request.Data;
    var deferral = args.Request.GetDeferral();
    var photoFile = await StorageFile.GetFileFromApplicationUriAsync(new Uri("ms-appx:///Assets/111.jpg"));
    dp.Properties.Title = title_MainPage.Text;
    dp.SetText(description_MainPage.Text.Trim());
    dp.Properties.Description = description_MainPage.Text;
    dp.SetStorageItems(new List<StorageFile> { photoFile });
    deferral.Complete();
}
```

四 . 思考与总结

在做题目的过程中学到了很多。这次实验总的来说要比前几次简单，但是我想向老师和 TA 提一下建议：就是 XML 文件的解析这块知识没有比较完整地向学生提供，应该提供给我们对应的解析 XML 文件的文档或网页链接，就我本人来说，这次项目的一半以上时间都在网上寻找如何解析 XML 文件的知识。但是最后我选择将自己在 Notifications Visualizer 编写好的 XML 代码翻译成 CSharp 语言，这样才得以完成任务。所以希望下几届师弟师妹们能够有“如何解析 XML 文件”的文档或网站链接来学习，减少困惑。

所以说我的可视化开发流程就是：设计磁贴 UI→Notifications Visualizer 编写好 XML 代码→根据可视化软件修改 XML 代码→翻译成对应的 CSharp 语言