

现代操作系统应用开发实验报告

学号： 15331046

班级： 软工一班

姓名： 陈志扬

实验名称： HW2-Todos

一．参考资料

请在这里列出对本实验有帮助你所参考的资料或者网站。

资料：2017week2 课件、2017-homework2.pptx 课件、2017-homework2-demo

网站：<http://stackoverflow.com/> <http://blog.csdn.net/> <https://www.baidu.com/>

<https://developer.microsoft.com/zh-cn/windows/apps/develop>

[https://docs.microsoft.com/zh-cn/windows/uwp/xaml-platform/xaml-syntax-guid](https://docs.microsoft.com/zh-cn/windows/uwp/xaml-platform/xaml-syntax-guide)

[e](https://docs.microsoft.com/zh-cn/windows/uwp/xaml-platform/)

<https://docs.microsoft.com/zh-cn/windows/uwp/xaml-platform/>

<https://social.msdn.microsoft.com/forums/zh-cn/home> 等等

二．实验步骤

请在这里简要写下你的实验过程。

首先，阅读相关课件，学习 xaml 语法，学习 Page Navigation；

然后，在 vs2015 新建项目，根据 demo 构建出界面，从而进一步理解 xaml 各种控件及属性的使用；

接着，在 App.xaml.cs 文件中编写 BackRequested 代码，在 MainPage.xaml.cs 文件中编写代码，使点击+号添加 Add 按钮可跳转到 NewPage，然后再 NewPage.xaml.cs 文件中编写代码使返回按钮 BackButton 可见，至此，作业要求前两个任务基本实现。

再者，完成 task3 简单逻辑。

对于 CheckBox，添加事件 CheckBoxClick，Line 的 visibility 设置为 Collapsed，在 MainPage.xaml.cs 中处理函数 CheckBoxClick 使之点击 CheckBox 后 Line 才出现。

对于 NewPage 点击 Create 按钮 先在 NewPage.xaml 中对 Create 按钮添加事件 Create，然后在 Title、Details 和 Due Date 分别添加 Name：Title、Details、DatePicker，然后在 NewPage.xaml.cs 中处理函数 Create 判断 Title、Details、DatePicker 是否合法。

对于 NewPage 点击 Cancel 按钮 先在 NewPage.xaml 中对 Cancel 按钮添加事件 Cancel，在 NewPage.xaml.cs 中处理函数 Cancel，将 Title、Details 置为空，Due Date 置为当前日期。

对于 Bonus：添加一个 AppBarButton 按钮，可从本地选择图片进行显示。用到了

```
var picker = new Windows.Storage.Pickers.FileOpenPicker();  
IRandomAccessStream fileStream  
= await file.OpenAsync(Windows.Storage.FileAccessMode.Read);  
详细代码见“实验过程遇到的问题”。
```

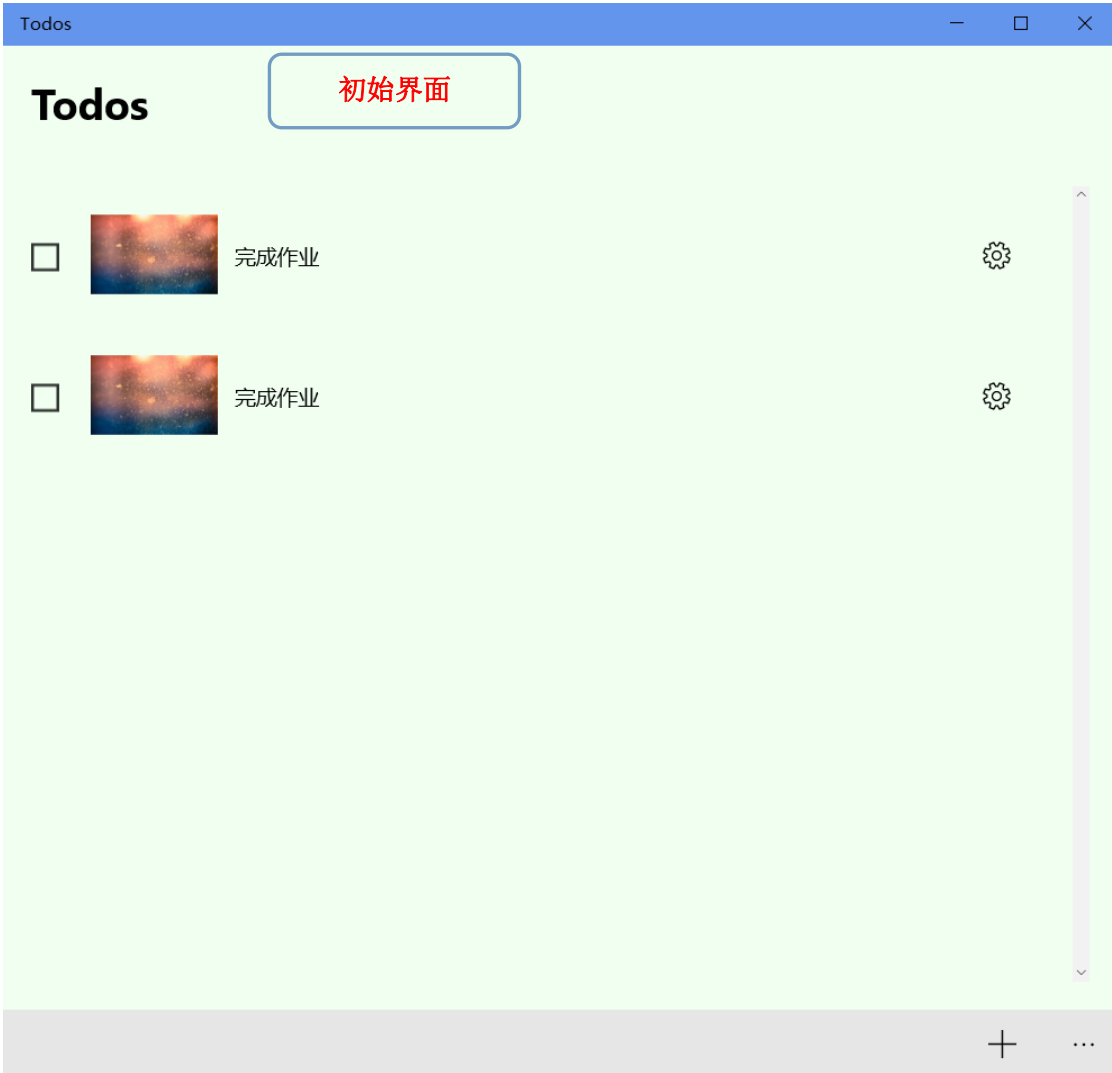
最后，我自己在 MainPage 里对 setting 按钮作了一些“很鸡肋”的操作，具体是

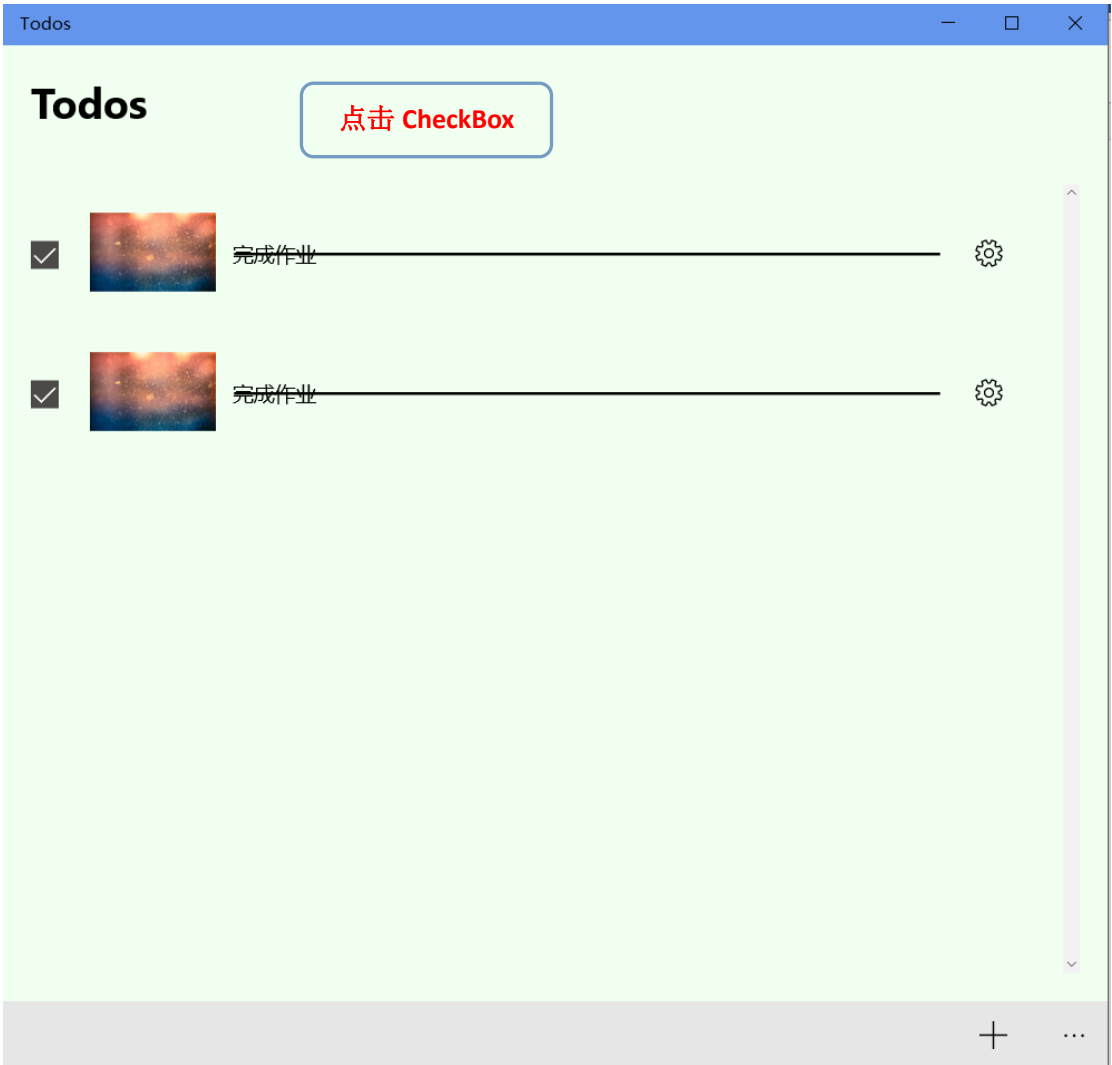
当用户点击 Delete 时，将 TextBlock 置空，并取消 CheckBox 的点击和 Line 的出现；

当用户点击 Edit 时，首先判断 TextBlock 是否为空，若为空，恢复“完成作业”四个字，若不为空，则仅取消 CheckBox 的点击和 Line 的出现，而不理 TextBlock 的内容。

三．实验结果截图

请在这里把实验所得的运行结果截图。





Todos

点击 Delete



完成作业



Edit Todo Item

NewPage 界面



Welcome!

关闭(C)

Details

Due Date

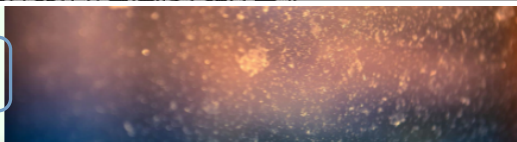
2017年	3月	8日
-------	----	----

Create

Cancel

...

Title 为空



The title can't be empty!

关闭(C)

Details

Due Date

2017年	3月	8日
-------	----	----

Create

Cancel

Details 为空

The details can't be empty!

关闭(C)

Details

Due Date

2017年	3月	8日
-------	----	----

Create

Cancel

日期错误

The date before current date is wrong!

关闭(C)

Details

Due Date

2017年	2月	8日
-------	----	----

Create

Cancel

You have created it!


关闭(C)

关闭(C)

创建成功

Title

222222222222222222222222222222222222
222222222222222222222222222222222222



Wrap 自动换行

Details

[illegible]

Due Date

2017年

3月

8日

Create

Cancel





四 . 实验过程遇到的问题

请在这里写下你在实验过程中遇到的问题以及解决方案。

- ① 对 xaml 各种控件的滥用乱用，以及对控件属性的不熟悉需要付出一定时间来学习，经过仔细对照 xaml 编写和设计器的效果逐渐客服该困难。
- ② 对于 task2 两个界面之间的导航 Navigation：

第一，数据无法保存，假如在 MainPage 点击了 CheckBox，那么将会有 Line 的出现，但是一旦点击了+号跳转到 NewPage 再回来 MainPage 又被“刷新”了，即又恢复到初始状态，在官网上查到这么一行代码即可解决该问题。

`this.NavigationCacheMode = Windows.UI.Xaml.Navigation.NavigationCacheMode.Enabled;`

第二：点击+号实现跳转到 NewPage，在 MainPage.xaml.cs 中添加以下代码，由于

在 App.xaml.cs 中 `rootFrame.Navigate(typeof(MainPage), e.Arguments)`, 这里并没有参数故删除 `e.Arguments`.

```
private void AddAppBarButton_Click(object sender, RoutedEventArgs e)
{
    // TODO
    Frame rootFrame = Window.Current.Content as Frame;
    rootFrame.Navigate(typeof(NewPage));
}
```

点击返回按钮←返回到 MainPage, 在 NewPage 设置←可见, 添加如下代码, 这里我犯了一个非常严重的错误, 一直以为有个 AppBarButton 控件可添加 Back 按钮, 因此浪费了很多时间, 最后在 msdn 上发现并没有该控件, 我就知道是直接把下面的代码添加到 NewPage.xaml.cs 中即可。

```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    //base.OnNavigatedTo(e);
    Frame rootFrame = Window.Current.Content as Frame;
    if (rootFrame.CanGoBack)
    {
        // Show UI in title bar if opted-in and in-app backstack is not empty.
        SystemNavigationManager.GetForCurrentView().AppViewBackButtonVisibility =
            AppViewBackButtonVisibility.Visible;
    }
    else
    {
        // Remove the UI from the title bar if in-app backstack is empty.
        SystemNavigationManager.GetForCurrentView().AppViewBackButtonVisibility =
            AppViewBackButtonVisibility.Collapsed;
    }
}
```

在 Navigation 上, 刚开始我删掉了 MainPage.xaml.cs 中上面这段代码, 这导致了在 MainPage 上也出现了返回按钮, 后来发现 MainPage 并不能继续返回, 即! CanGoBack, 随即添加上面代码到 MainPage.xaml.cs 中即可。

③对于 CheckBox, 添加 Name 和 Click 如下, 在 Line 中添加 Name 并将其 Visibility 设置为 Collapsed

```
<CheckBox x:Name="CheckBox" Click="CheckBoxClick" Grid.Column="0" VerticalAlignment="Center" Height="32" Width="32" />
```

```
<Line x:Name="Line" Visibility="Collapsed" Grid.Column="2" Stretch="Fill" Stroke="Black"
```

然后在 MainPage.xaml.cs 中添加处理函数 CheckBoxClick 如下, 在这里为什么需要判断 `task.Text != ""`, 因为我自己在设置 setting 中的 Edit 和 Delete 作了一点处理代码, 可执行相应的 Edit 和 Delete; CheckBox.IsChecked 判断是否点击了 CheckBox; 若两者为 True, 则将 Line 设置为 Visible, 反之设置为 Collapsed。

```

private void CheckBoxClick(object sender, RoutedEventArgs e)
{
    if (CheckBox.IsChecked == true && task.Text != "")
    {
        Line.Visibility = Visibility.Visible;
    }
    else
    {
        Line.Visibility = Visibility.Collapsed;
    }
    if (CheckBox1.IsChecked == true && task1.Text != "")
    {
        Line1.Visibility = Visibility.Visible;
    }
    else
    {
        Line1.Visibility = Visibility.Collapsed;
    }
}

```

④对于 Create 和 Cancel，在 NewPage.xaml 中对应的 Title、Details、DatePicker 作出如下处理

```

<TextBox x:Name="Title" Width="350" Header="Title" Margin="0,12,0,0" TextWrapping="Wrap"/>
<TextBox x:Name="Details" Width="350" Height="96" Margin="0,12,0,0" Header="Details" TextWrapping="Wrap"/>
<DatePicker x:Name="DatePicker" Width="350" Margin="0,12,0,0" Header="Due Date" HorizontalAlignment="Center" />
<RelativePanel Width="350" Margin="0,24,0,0">
    <Button x:Name="CreateButton" Content="Create" Click="Create"></Button>
    <Button x:Name="CancelButton" Content="Cancel" Click="Cancel" RelativePanel.AlignRightWithPanel="True"></Button>
</RelativePanel>

```

在 NewPage.xaml.cs 中添加 Create 和 Cancel 的函数

对于 Create 函数，给出不同判断提示用户是哪输入不合法，这样设计对用户是友好的，但由于 Create 成功不作要求，故只是弹出信息提示创建成功。

这个函数难点主要涉及日期的判断，因为 DateTime.Now 是精确到秒的，故刚开始我只写成 DatePicker.Date < Date.Now，在当前日期无法创建成功，所以要在 DateTime.Now 后面加上 Date 使之变成 DateTime.Now.Date 用以获得当前日期。

```

private void Create(object sender, RoutedEventArgs e)
{
    if (Title.Text == "")
    {
        var i = new MessageBox("The title can't be empty!").ShowAsync();
    }
    else if (Details.Text == "")
    {
        var i = new MessageBox("The details can't be empty!").ShowAsync();
    }
    else if (DatePicker.Date < DateTime.Now.Date)
    {
        var i = new MessageBox("The date before current date is wrong!").ShowAsync();
    }
    else
    {
        var i = new MessageBox("You have created it!").ShowAsync();
    }
}

```

对于 Cancel 函数，只需将 Title、Details 设置为空，日期改成当前日期即可，当前日期可由上面讨论的 DateTime.Now.Date 获得。

```

private void Cancel(object sender, RoutedEventArgs e)
{
    Title.Text = "";
    Details.Text = "";
    DatePicker.Date = DateTime.Now.Date;
}

```

⑤至此，还有一个问题，当我尝试在 Title 和 Details 无限输入时，发现 Title 并不会自动换行，我觉得不太友好，故我在控件 Title 和 Details 添加如下的属性，这样即可实现。

```

<TextBox x:Name="Title" Width="350" Header="Title" Margin="0,12,0,0" TextWrapping="Wrap"/>
<TextBox x:Name="Details" Width="350" Height="96" Margin="0,12,0,0" Header="Details" TextWrapping="Wrap"/>

```

⑥对 Bonus 的处理，是真的浪费了很多查找资料的时间，因为一直找到各种 OpenFileDialog 之类的东西，但是在 uwp 好像又用不了。最后，经过群里大佬的提示 FileOpenPicker 才可以实现打开本地文件窗口，再经过下面随机输入输出流和 BitmapImage 才实现 Bonus 功能。代码如下：async 也是经查资料才得知的。

```

private async void SelectPicture(object sender, RoutedEventArgs e)
{
    var picker = new Windows.Storage.Pickers.FileOpenPicker();
    picker.ViewMode = Windows.Storage.Pickers.PickerViewMode.Thumbnail;
    picker.SuggestedStartLocation =
        Windows.Storage.Pickers.PickerLocationId.PicturesLibrary;
    picker.FileTypeFilter.Add(".jpg");
    picker.FileTypeFilter.Add(".jpeg");
    picker.FileTypeFilter.Add(".png");

    Windows.Storage.StorageFile file = await picker.PickSingleFileAsync();
    if (file != null)
    {
        IRandomAccessStream fileStream = await file.OpenAsync(Windows.Storage.FileAccessMode.Read);
        BitmapImage bitmapImage = new BitmapImage();
        bitmapImage.DecodePixelWidth = 350;
        bitmapImage.DecodePixelHeight = 180;
        await bitmapImage.SetSourceAsync(fileStream);
        background.Source = bitmapImage;
    }
}

```

⑦对于 Edit 和 Delete, “鸡肋”的操作代码如下:

```
private void Edit(object sender, RoutedEventArgs e)
{
    if (task.Text == "") task.Text = "完成作业";
    Line.Visibility = Visibility.Collapsed;
    CheckBox.IsChecked = false;
}
private void Delete(object sender, RoutedEventArgs e)
{
    task.Text = "";
    Line.Visibility = Visibility.Collapsed;
    CheckBox.IsChecked = false;
}
```

五. 思考与总结

请在这里写下你本次试验的心得体会以及所思所想。

目前我们主要在学习 xaml, 真正用 C#实现逻辑处理还非常少, 至少对于本次实验如果不考虑后面 bonus 和实现 Edit、Delete 来说。总感觉课上老师并没有讲多少, 许多知识都得靠我们自己找来学习弄懂, 这无疑让我们付出了很大的时间成本, 总的来说, 我觉得真的是培养自主学习能力, 不断学习的能力是非常重要的, 但是自学能力是必不可少的。

1. 实验报告提交格式为 pdf。
2. 实验内容不允许抄袭, 我们要进行代码相似度对比。如发现抄袭, 按 0 分处理。