

# 现代操作系统应用开发实验报告

姓名: 徐伟元 学号: 16340261 Blog: [My Blog](#)

## 实验名称: Lab1 MyList

### 参考资料

- [Microsoft Documents](#)
- CSDN
- MSDN
- Stack Overflow

### 基础实验步骤

#### • Week1

##### 1. checkBox 和 line 的绑定处理

当 CheckBox 被勾选时 line 出现, 取消勾选则line消失

##### 2. 新建Item的逻辑判断

新建一个 NewPage。点击 create 按钮时, 检查 Title、Description 是否为空, DueDate 是否正确(是否大于等于当前日期)。如果不正确, 弹出对话框, 显示错误信息。

#### • Week2

##### 1. 界面设计

完成 MainPage、NewPage 两个界面的设计

##### 2. Adaptive UI

- 当窗口宽度小于800时, 只显示原本在界面左侧的列表部分, 底部导航栏只显示 Add 图标
- 窗口宽度大于800时, 显示完整界面
- 窗口宽度小于600时, 列表项中的图片不显示

#### • Week3

##### 1. 数据绑定

##### 2. 剩余各页面内剩余逻辑和拓展

### 关键代码与截图

#### 1. checkBox 和 line 的绑定处理

```
<CheckBox IsChecked="{x:Bind IsCompleted, Mode=TwoWay}"/>

<Line Visibility="{x:Bind IsCompleted, Converter={StaticResource Converter}, Mode=OneWay}"/>
```

```
class CompletedConveter : IValueConverter {
    public object Convert(object value, Type targetType, object parameter, string language) {
        bool result = (bool)value;
        if (result) return Visibility.Visible;
        return Visibility.Collapsed;
    }

    public object ConvertBack(object value, Type targetType, object parameter, string language) {
        throw new NotImplementedException();
    }
}
```

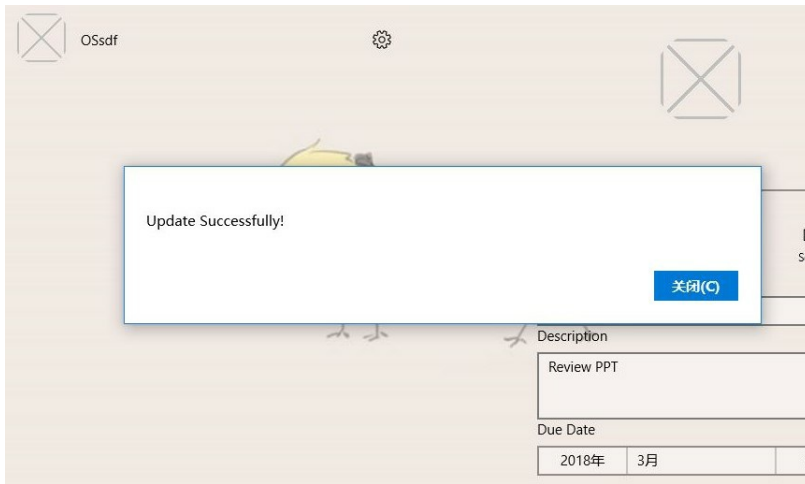


## 1. Adaptive UI

```
<!--Adaptive UI Settings-->
<VisualStateManager.VisualStateGroups>
  <VisualStateGroup>
    <VisualState>
      <VisualState.Setters>
        <Setter Target="Detail_Part.(UIElement.Visibility)" Value="Collapsed"/>
        <Setter Target="List_Part.(Grid.ColumnSpan)" Value="2"/>
      </VisualState.Setters>
      <VisualState.StateTriggers>
        <AdaptiveTrigger MinWindowWidth="1"/>
      </VisualState.StateTriggers>
    </VisualState>
    <VisualState x:Name="Width_800">
      <VisualState.StateTriggers>
        <AdaptiveTrigger MinWindowWidth="800"/>
      </VisualState.StateTriggers>
    </VisualState>
  </VisualStateGroup>
</VisualStateManager.VisualStateGroups>
```

## 1. Data Binding

```
<ListView ItemsSource="{x:Bind ViewModel.AllItems}">
  <ListView.ItemTemplate>
    <DataTemplate x:DataType="md:ListItem">
      <UserControl>
        <Grid>
          <CheckBox ... IsChecked="{x:Bind IsCompleted, Mode=TwoWay}"/>
          <Rectangle ... Fill="{x:Bind Image, Mode=OneWay}"/>
          <TextBlock ... Text="{x:Bind Title, Mode=OneWay}"/>
          <Line Visibility="{x:Bind IsCompleted, Converter={StaticResource Converter},
Mode=OneWay}"/>
        </Grid>
      </UserControl>
    </DataTemplate>
  </ListView.ItemTemplate>
</ListView>
```

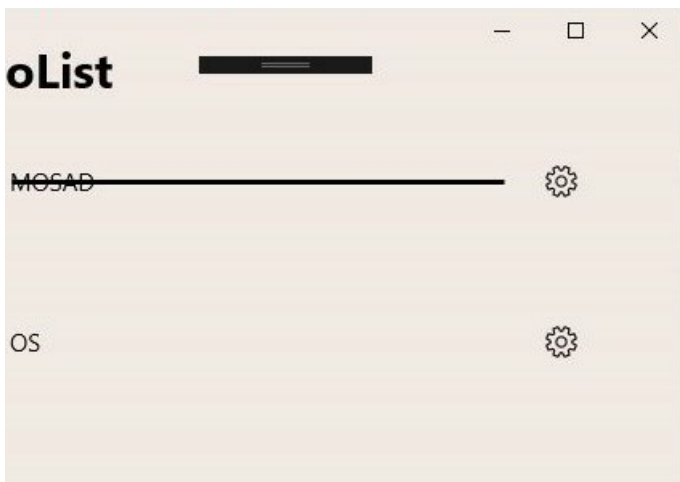


## 亮点与改进

此部分仅附部分代码和效果图

### 1. 标题栏透明化

```
// Set the titleBar to be transparent
CoreApplication.GetCurrentView().TitleBar.ExtendViewIntoTitleBar = true;
// Set the color style for the command button of titleBar
ApplicationView.GetForCurrentView().TitleBar.ButtonBackgroundColor = Color.FromArgb(0, 0, 0, 0);
ApplicationView.GetForCurrentView().TitleBar.ButtonForegroundColor = Colors.Black;
```



### 1. 图片本地选取

```
private async void Select_Photo(object sender, RoutedEventArgs e) {
    FileOpenPicker picker = new FileOpenPicker();

    // Initialize the picture file type to take
    picker.FileTypeFilter.Add(".jpg");
    picker.FileTypeFilter.Add(".jpeg");
    picker.FileTypeFilter.Add(".png");
    picker.FileTypeFilter.Add(".bmp");

    picker.SuggestedStartLocation = PickerLocationId.PicturesLibrary;

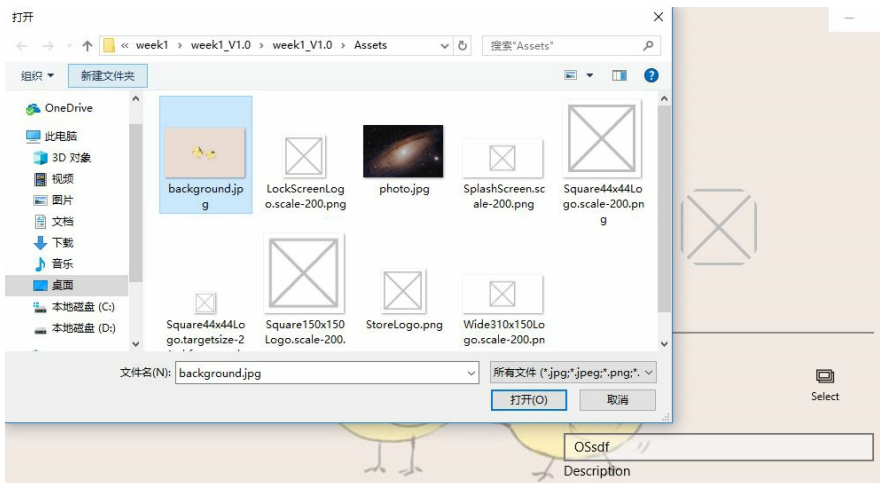
    StorageFile file = await picker.PickSingleFileAsync();
}
```

```

if (file != null) {

    // Load the selected picture
    imagePath = file.Path;
    IRandomAccessStream ir = await file.OpenAsync(FileAccessMode.Read);
    BitmapImage bi = new BitmapImage();
    await bi.SetSourceAsync(ir);
    var brush = new ImageBrush {
        ImageSource = bi
    };
    Photo.Fill = brush;
}
}

```

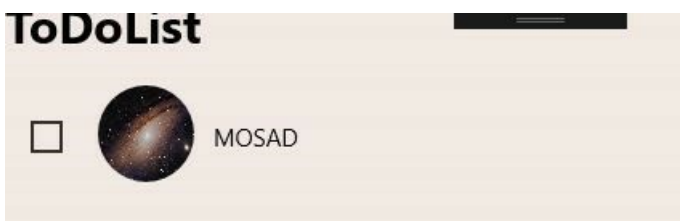


#### 1. 圆形图片

```

<Rectangle Height="60" Width="60" RadiusX="60" RadiusY="60">
    <Rectangle.Fill>
        <ImageBrush Source="...">
    </Rectangle.Fill>
</Rectangle>

```



#### 1. 图片绑定

```

<Rectangle Fill="{x:Bind Image, Mode=OneWay}"/>

```

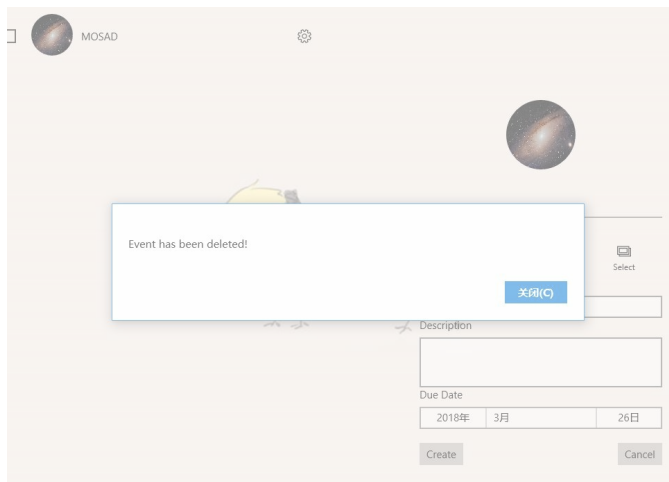
#### 1. 鲁棒性

```

//Check if selectedItem is deleted
if (ViewModel.SelectedItem != null) {
    errMsg += "Update Successfully!\n";
    ViewModel.UpdateItem(ViewModel.SelectedItem.id, Title.Text, Description.Text, Date.Date,
    Photo.Fill, imagePath);
} else {

```

```
errMsg += "Event has been deleted!\n";
Title.Text = "";
Description.Text = "";
Date.Date = DateTimeOffset.Now;
Create.Content = "Create";
ViewModel.SelectedItem = null;
}
```



## 遇到的问题

### 1. 标题栏

标题栏的处理过程经历了两个阶段:背景透明化按钮背景透明化。

### 2. 图片路径获取

因为采用了 `Rectangle` 来实现圆形图片，所以在已有图片的路径获取出现了曲折的道路。

## 思考与总结

UWP 开发不是一件简单的事情。网上的资料比较稀少，对于现有需求通常需要对官方文档进行反复的查阅。这对于信息的检索能力和获取能力有着极大的帮助。每周任务完成，会适当拓展一些功能，并且会以[博客](#)的形式记录下来。个人感觉对学习会有帮助。