

现代操作系统应用开发实验报告

学号：153311117

班级：教务 2 班

姓名：黄楠绚

实验名称：homework8

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。

<http://www.cnblogs.com/wpali/p/5657389.html>

<https://msdn.microsoft.com/zh-cn/library/windows/apps/xaml/hh465160.aspx>

<http://www.cnblogs.com/NEIL-X/p/5461934.html>

二. 实验步骤

请在这里简要写下你的实验过程。

先设计 xaml，基本模仿 demo 的形式，没什么创新。

动画 StoryBoard 一开始无法添加，后来看了一下官网发现它是 resource 的属性，因此把它和数据绑定的 convertor 放在一起：

```
<Page.Resources>
    <cvt:valueConvert x:Key="valueConvert"/>
    <cvt:volumeConvert x:Key="volumeConvert"/>
    <Storyboard x:Name="Storyborad" RepeatBehavior="Forever" >
        <DoubleAnimation Duration="0:0:30" To="360" Storyboard.TargetProperty="(UIElement.RenderTransform).(CompositeTransform.Rotation)" Storyboard.
    </Storyboard>
</Page.Resources>
```

播放音频时的旋转图片：

```
<Ellipse x:Name="BackgroundEllipse" Width="250" Height="250" VerticalAlignment="Center" HorizontalAlignment="Center" RenderTrans
    <Ellipse.RenderTransform>
        <CompositeTransform/>
    </Ellipse.RenderTransform>
    <Ellipse.Fill>
        <ImageBrush ImageSource="Assets\background.jpg" Stretch="Uniform"/>
    </Ellipse.Fill>
</Ellipse>
```

```
<MediaElement x:Name="media" Source="Assets\霹雳英雄 - 涅槃.mp3" MediaOpened="media_MediaOpened" MediaFailed="media_MediaFailed" MediaEnded="media_Me
```

两个滑动条的绑定：一个是绑定播放进度，一个是绑定音量

```
<Slider x:Name="mediaSlider" Value="{Binding ElementName=media, Path=Position, Converter={StaticResource valueConvert}, Mode=TwoWay}" HorizontalAlignm
<TextBlock Text="Volume" HorizontalAlignment="Right" VerticalAlignment="Bottom" Margin="0,10,117,64" Grid.Column="1" />
<Slider x:Name="volumeSlider" Value="{Binding ElementName=media, Path=Volume, Converter={StaticResource volumeConvert}, Mode=TwoWay}" Maximum="100" Ho
```

播放进度与进度条的转换：

```
class valueConvert : IValueConverter
{
    public object Convert(object value, Type targetType, object parameter, string language)
    {
        return ((TimeSpan)value).TotalSeconds;
    }

    public object ConvertBack(object value, Type targetType, object parameter, string language)
    {
        return TimeSpan.FromSeconds((double)value);
    }
}
```

音量与进度条的转换：

```
class volumeConvert : IValueConverter
{
    public object Convert(object value, Type targetType, object parameter, string language)
    {
        return (double)value * 100;
    }

    public object ConvertBack(object value, Type targetType, object parameter, string language)
    {
        return (double)value / 100;
    }
}
```

程序刚启动，数据的初始化：

```
private bool isVedio = false; // 判断正在播放的文件的类型
public MainPage()
{
    this.InitializeComponent();
    media.AutoPlay = false; // 程序刚启动不能自动播放
    isVedio = false; // 默认播放Assets中的音频文件
    CommandBar.ClosedDisplayMode = AppBarClosedDisplayMode.Compact; // 显示控制栏的按钮
    media.Volume = 0.2; // 默认音量
}
```

点击播放媒体文件，判断是否是视频文件，如果是视频文件，则停止播放动画，避免不必要的 cpu 消耗

```
private void Play_Click(object sender, RoutedEventArgs e)
{
    media.Play();
    if (!isVedio)
    {
        Storyborad.Begin();
    }
    else // 如果是视频文件，则停止播放动画，避免不必要的cpu消耗
    {
        Storyborad.Stop();
    }
}
```

点击暂停和停止:

```
private void Stop_Click(object sender, RoutedEventArgs e)
{
    media.Stop();
    Storyborad.Stop();
}

private void Pause_Click(object sender, RoutedEventArgs e)
{
    media.Pause();
    if (!isVedio)
    {
        Storyborad.Pause();
    }
}
```

点击全屏:

```
private void FullScreen_Click(object sender, RoutedEventArgs e)
{
    media.IsFullWindow = !media.IsFullWindow;
    // 播放视频且全屏，最小化控制面板的按钮，避免挡住视频影响用户体验
    if (media.IsFullWindow && isVedio)
    {
        CommandBar.ClosedDisplayMode = AppBarClosedDisplayMode.Minimal;
    }
    else
    {
        CommandBar.ClosedDisplayMode = AppBarClosedDisplayMode.Compact;
    }
}
```

选择播放文件

```
private async void SelectFile_Click(object sender, RoutedEventArgs e)
{
    FileOpenPicker picker = new FileOpenPicker();
    picker.ViewMode = PickerViewMode.Thumbnail;
    picker.SuggestedStartLocation = PickerLocationId.PicturesLibrary;
    addPickerType(picker);

    StorageFile file = await picker.PickSingleFileAsync();

    if (file != null)
    {
        changeMediaFile(file);
    }
}

private void addPickerType(FileOpenPicker picker)
{
    picker.FileTypeFilter.Add(".wmv");
    picker.FileTypeFilter.Add(".mp3");
    picker.FileTypeFilter.Add(".mp4");
    picker.FileTypeFilter.Add(".wma");
    picker.FileTypeFilter.Add(".mkv");
}
```



```
private async void changeMediaFile(StorageFile file)
{
    var fileStream = await file.OpenAsync(FileAccessMode.Read);
    media.SetSource(fileStream, file.ContentType);
    media.AutoPlay = true; // 用户选择一个文件后即可自动播放，无需点击Play
    // 如果媒体文件为视频文件
    if (isVedioFile(file.FileType))
    {
        isVedio = true;
        // 如果是播放视频且全屏播放，则最小化控制面板的按钮，控制面板按钮避免挡住视频影响用户体验
        if (media.IsFullWindow)
        {
            CommandBar.ClosedDisplayMode = AppBarClosedDisplayMode.Minimal;
        }
    }
    else
    {
        isVedio = false;
        CommandBar.ClosedDisplayMode = AppBarClosedDisplayMode.Compact;
    }
}
}
```

```
// 根据文件后缀名判断文件是否是视频文件
private bool isVedioFile(string fileType)
{
    return fileType == ".mp4" || fileType == ".wmv" || fileType == ".mkv";
}
```

对 MediaElement 的三种情况进行处理：

```
private void media_MediaOpened(object sender, RoutedEventArgs e)
{
    // 设置进度条长度与媒体文件播放长度一致
    mediaSlider.Maximum = media.NaturalDuration.TimeSpan.TotalSeconds;
    // 设置进度条的值，从0开始
    mediaSlider.Value = 0;
}

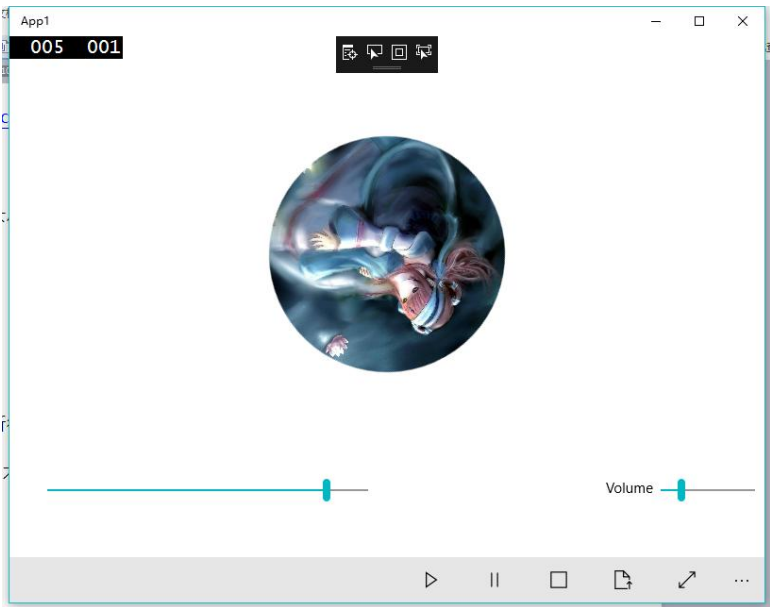
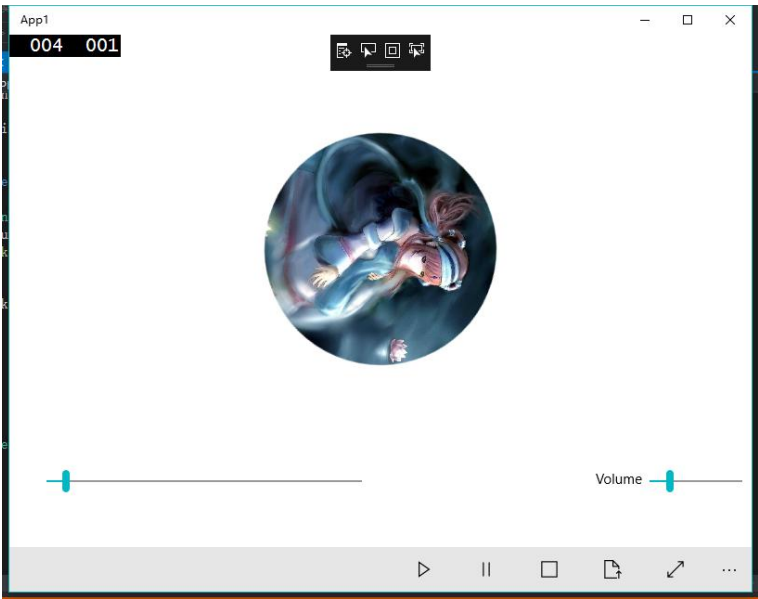
private void media_MediaEnded(object sender, RoutedEventArgs e)
{
    // 文件播放结束后，可以自动重新播放，即单曲循环模式
    media.Play();
}

// 播放媒体文件失败，弹出消息框提示用户
private void media_MediaFailed(object sender, ExceptionRoutedEventArgs e)
{
    var i = new MessageDialog("Sorry, unable to play!").ShowAsync();
}
```

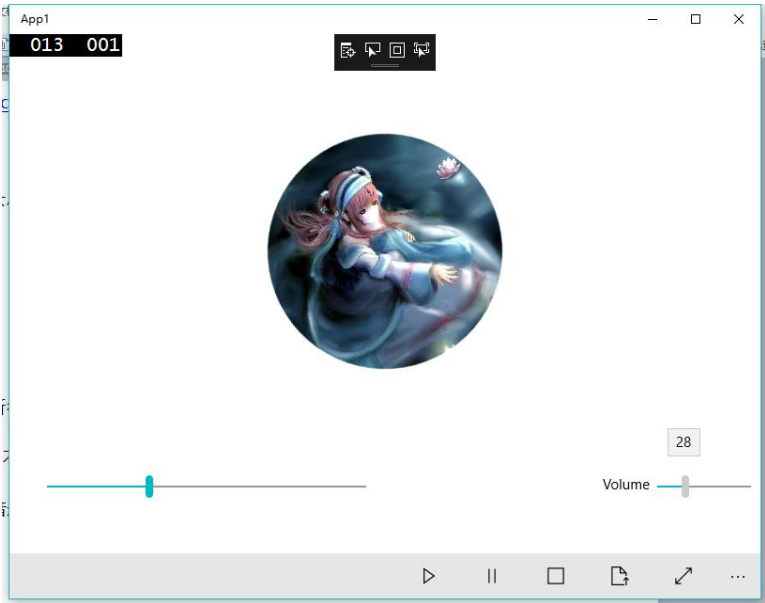
三．实验结果截图

启动程序，设置了不能自动播放，要点击 play 键才开始播放 Assets 中的默认歌曲，程序启动初始的音量

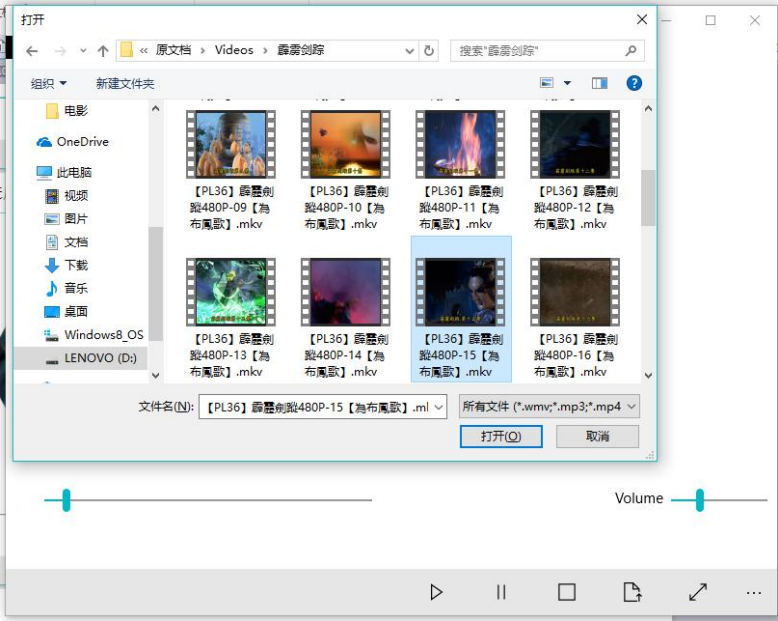
值为 20 (最高 100) , 播放过程中图片可以旋转 , 默认播放一个文件结束后会自动重新播放 , 单曲循环



可以调节音量和播放进度条:



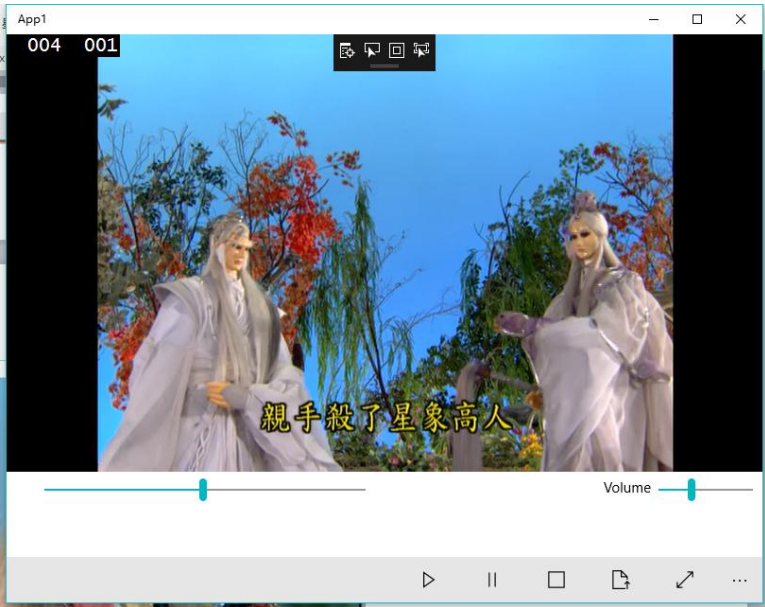
点击选择文件：



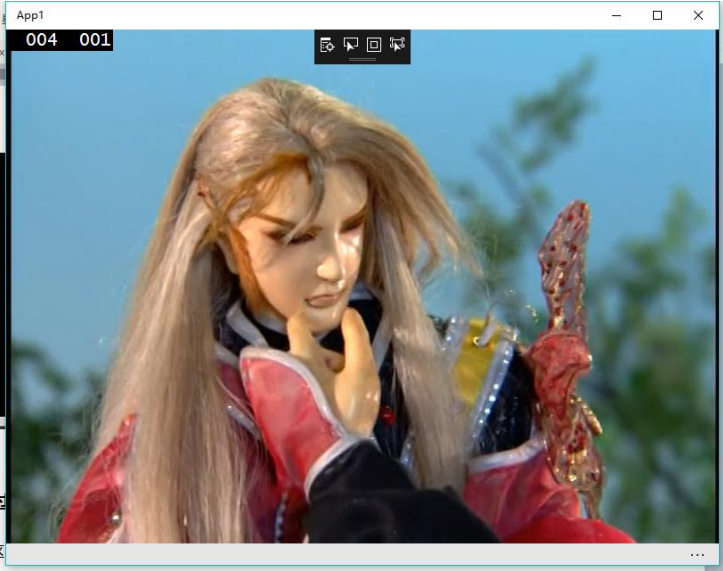
打开一个视频文件后，自动播放：



调整进度条，视频进度随着改变：



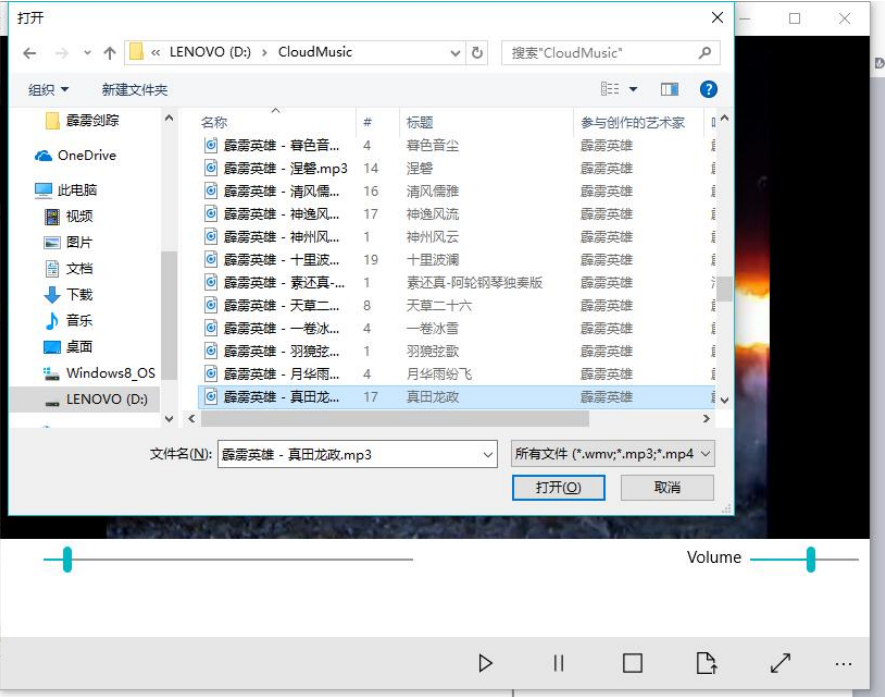
点击全屏，可以看到右下角的控制面板按钮消失：



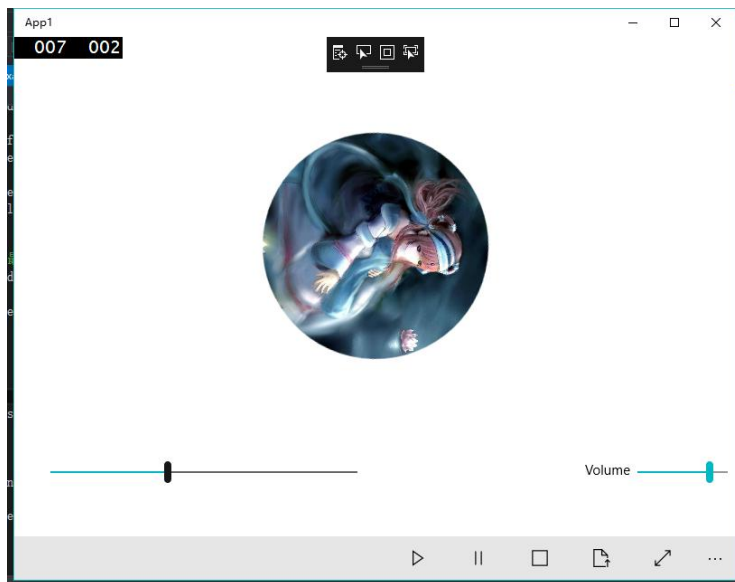
点击窗口最大化，嗯，效果好像挺不错的~：



再选择另一个音频：



正常播放，音量可以调节，而且图片也正常旋转~



测试完毕!!

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。

动画 StoryBoard 一开始无法添加，后来看了一下官网发现它是 resource 的属性，放进 Page.resource 就好了！

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。

这次的作业是我最喜欢的一次现操作业，可以用自己写的 app 播放自己的媒体文件，很开心，这次作业也写的比较快，所以是兴趣催生学习的动力吧~~

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。