

现代操作系统应用开发实验报告

学号： 14970011

班级： 2015 级教务 2 班

姓名： 宋思亭

实验名称： homework7

一 . 参考资料

Windows UWP Namespaces : <https://docs.microsoft.com/en-us/uwp/api/>

课件、Demo

二 . 实验步骤

1. 阅读作业需求和课件 PPT , 阅读官方文档
2. 在MainPage中写主要界面 , 由封面图片、媒体、进度条和底部按钮构成 , 另外一个TextBlock显示视频总长度 , 底部按钮有播放、暂停、停止、全屏和打开本地图片 , 另有一个Slider调节音量。

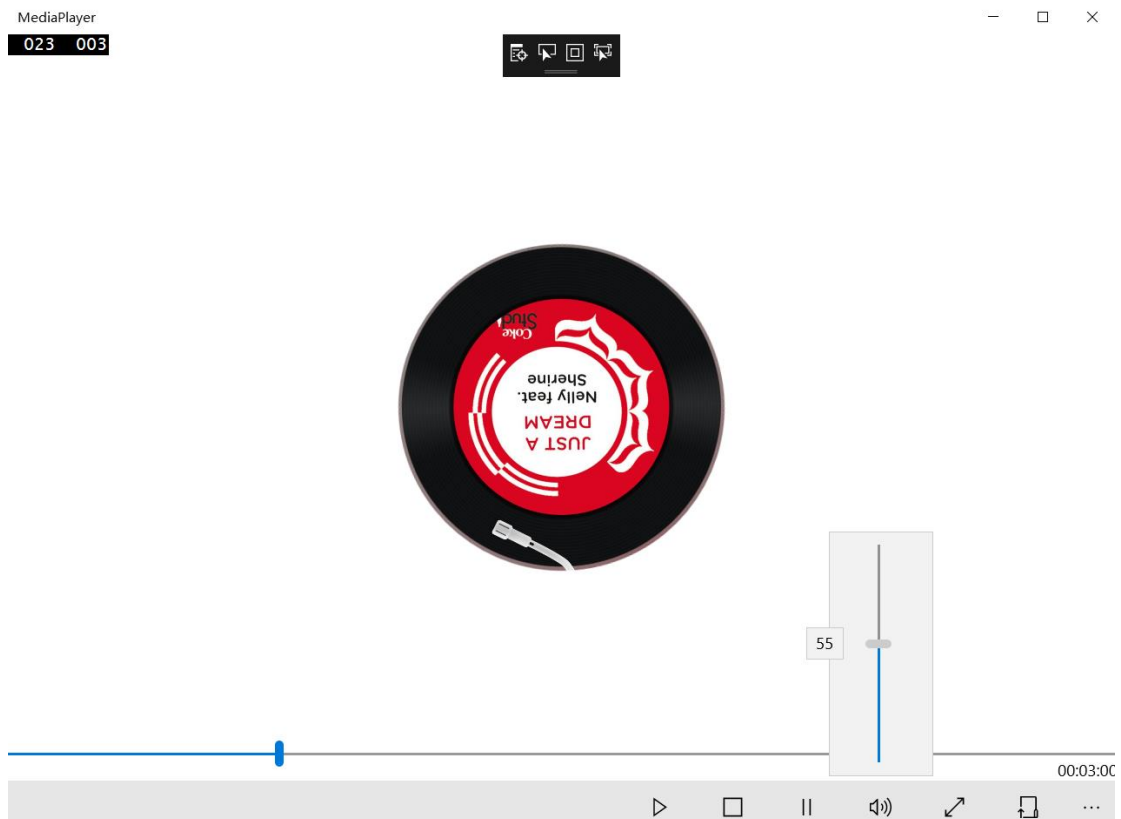
```

<Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
    <Ellipse Height="300" Width="300" RenderTransformOrigin="0.5,0.5">
        <Ellipse.Fill>
            <ImageBrush ImageSource="Assets/background.jpg"/></ImageBrush>
        </Ellipse.Fill>
        <Ellipse.RenderTransform>
            <RotateTransform x:Name="trans"/>
        </Ellipse.RenderTransform>
    </Ellipse>
    <MediaElement x:Name="mediaPlayer" MediaOpened="mediaOpened" Source="Assets/music/Paper Plane's Adventure.
    <TextBlock x:Name="leftTime" Text="00:00:00" VerticalAlignment="Bottom" HorizontalAlignment="Right"></Tex
    <Slider x:Name="mediaSlider" Value="{Binding ElementName=mediaPlayer,Path=Position,Converter={StaticResou
</Grid>

<Page.BottomAppBar>
    <CommandBar x:Name="bottomCommandBar">
        <AppBarButton x:Name="playButton" Icon="Play" Label="播放" Tapped="playButton_Tapped"/>
        <AppBarButton x:Name="stopButton" Icon="Stop" Label="停止" Tapped="stopButton_Tapped"/>
        <AppBarButton x:Name="pauseButton" Icon="Pause" Label="暂停" Tapped="pauseButton_Tapped"/>
        <AppBarButton Icon="Volume" Label="音量">
            <AppBarButton.Flyout>
                <Flyout>
                    <Slider x:Name="volumeSlider" Orientation="Vertical" Height="200" HorizontalAlignment="Ce
                </Flyout>
            </AppBarButton.Flyout>
        </AppBarButton>
        <AppBarButton x:Name="fullButton" Icon="FullScreen" Label="全屏" Tapped="fullButton_Tapped"/>
        <AppBarButton x:Name="localVideoButton" Icon="OpenLocal" Label="本地" Tapped="onTappedChangeVideoMeth
    </CommandBar>
</Page.BottomAppBar>

```

最终效果如下图：



3. 写播放、停止和暂停功能

```

private void stopButton_Tapped(object sender, TappedRoutedEventArgs e)
{
    mediaPlayer.Stop();
    storyboard1.Stop();
}

private async void playButton_Tapped(object sender, TappedRoutedEventArgs e)
{
    try
    {
        mediaPlayer.Play();
    }
    catch (Exception)
    {
        await new MessageDialog("播放失败，请尝试打开其他文件！").ShowAsync();
    }
    storyboard1.Begin();
}

private void pauseButton_Tapped(object sender, TappedRoutedEventArgs e)
{
    mediaPlayer.Pause();
    storyboard1.Pause();
}

```

4. 实现进度条实时显示媒体的播放进度

新建一个MusicConverter类做类型转换：

```

class MusicConverter : 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);
    }
}

```

设置<Slider x:Name="mediaSlider" Value="{Binding ElementName=mediaPlayer, Path=Position, Converter={StaticResource mc}, Mode=TwoWay}" ValueChanged="ChangeMediaSpeedRatio" VerticalAlignment="Bottom"></Slider>绑定实现显示播放进度。

通过函数ChangeMediaSpeedRatio实现拖动进度条，也可以使媒体快速前进/后退到相应的位置。

```

private void ChangeMediaSpeedRatio(object sender, RoutedEventArgs args)
{
    mediaPlayer.Position = TimeSpan.FromSeconds(mediaSlider.Value);
}

```

5. 实现播放视频全屏显示/退出全屏显示

```
private void fullButton_Tapped(object sender, TappedRoutedEventArgs e)
{
    if (mediaPlayer.IsFullWindow == false)
    {
        mediaPlayer.IsFullWindow = true;
    }
    else
    {
        mediaPlayer.IsFullWindow = false;
    }
}
```

6. 写调节音量的进度条

```
<AppBarButton Icon="Volume" Label="音量">
    <AppBarButton.Flyout>
        <Flyout>
            <Slider x:Name="volumeSlider" Orientation="Vertical" Height="200" HorizontalAlignment="Center">
            </Flyout>
        </AppBarButton.Flyout>
    </AppBarButton>
```

在MediaElement中绑定

```
Volume="{Binding ElementName=volumeSlider,Path=Value}"
```

7. 写从本地打开媒体文件并播放的功能

FileOpenPicker打开本地文件,选择mp3或mp4类型,添加到MediaElement

的Source中播放,若播放失败弹出错误提示

```
private async void onTappedChangeVideoMethodButton(object sender, TappedRoutedEventArgs e)
{
    Button currentTappedBtn = sender as Button;
    mediaPlayer.Pause();
    FileOpenPicker newmedia = new FileOpenPicker();
    newmedia.FileTypeFilter.Add(".mp3");
    newmedia.FileTypeFilter.Add(".mp4");
    newmedia.SuggestedStartLocation = PickerLocationId.VideosLibrary;
    StorageFile file = await newmedia.PickSingleFileAsync();
    if (file != null)
    {
        try
        {
            var stream = await file.OpenAsync(FileAccessMode.Read);
            mediaPlayer.SetSource(stream, file.ContentType);
        }
        catch (Exception)
        {
            await new MessageDialog("打开文件失败, 请重试! ").ShowAsync();
        }
    }
}
```

8. 写圆形封面及播放时旋转效果

```

        <Storyboard x:Name="storyboard1">
            <DoubleAnimation x:Name="position" Storyboard.TargetName="trans" Storyboard.TargetProperty="Angle"
            </Storyboard>
        </Page.Resources>

        <Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
            <Ellipse Height="300" Width="300" RenderTransformOrigin="0.5,0.5">
                <Ellipse.Fill>
                    <ImageBrush ImageSource="Assets/background.jpg"></ImageBrush>
                </Ellipse.Fill>
                <Ellipse.RenderTransform>
                    <RotateTransform x:Name="trans"/>
                </Ellipse.RenderTransform>
            </Ellipse>
        </Grid>
    
```

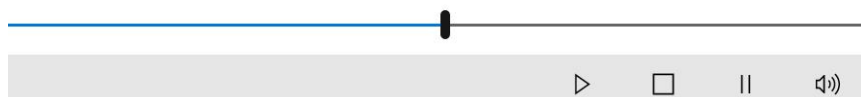
用Ellipse将图片绘制成圆形，Storyboard设置旋转动画，在播放、停止、暂停时分别设置动画播放、停止、暂停。

9. 调试项目

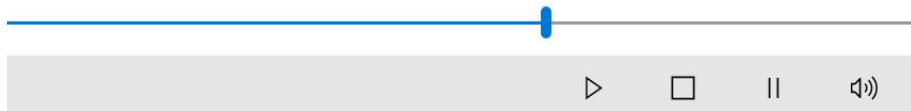
三 . 实验结果截图

1. 播放、停止、暂停和拖动进度条改变播放位置，同时展示播放音乐时，封面绕自身中心不断旋转，直至播放结束，封面复位。

播放：



暂停：



停止：

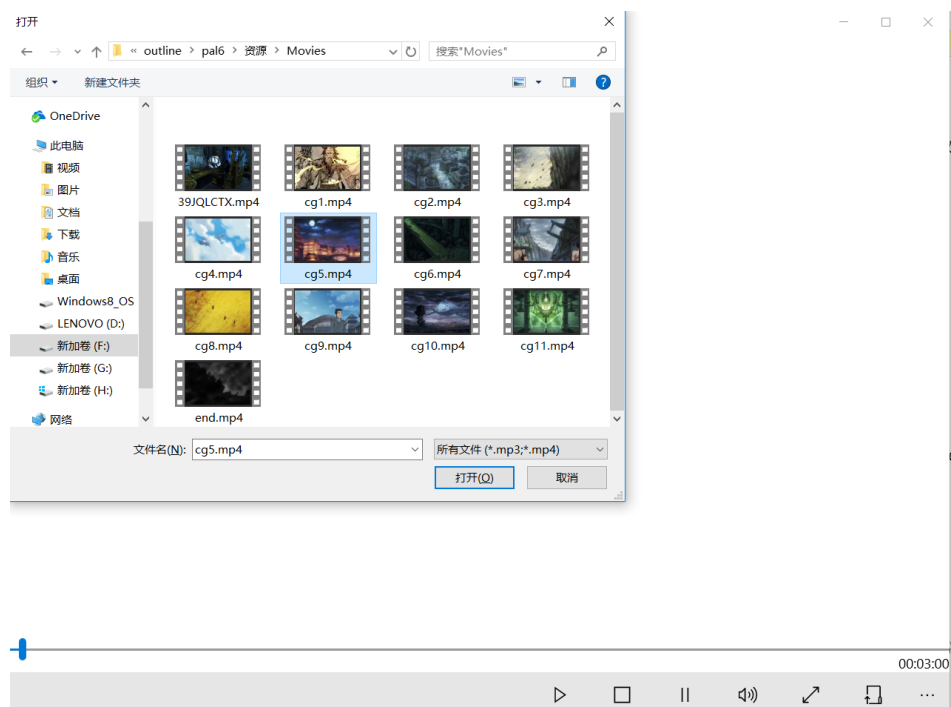


拖动进度条：

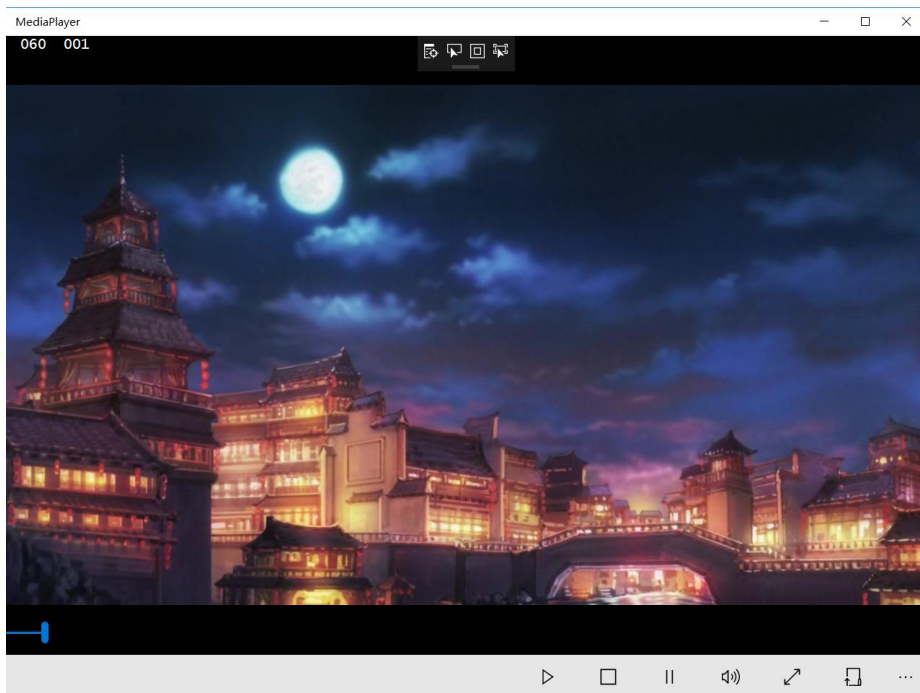


2. 打开本地媒体并播放

打开文件：



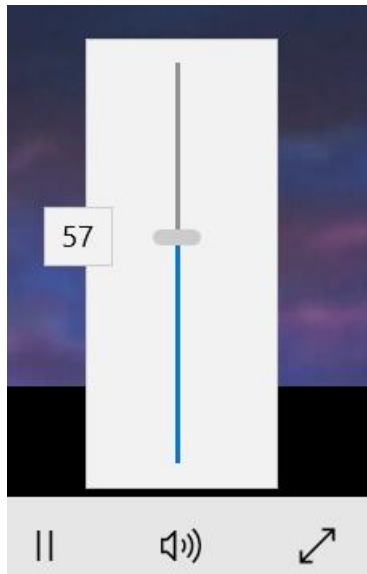
播放：



全屏：



3. 调节音量



四．实验过程遇到的问题

1. 打开本地文件时 `MediaElement.Source=file.path` 设置总是出错，查询后知道应该转为 `stream` 并用 `SetSource` 赋值：

```
var stream = await file.OpenAsync(FileAccessMode.Read);
```

```
mediaPlayer.SetSource(stream, file.ContentType);
```

2. 设置封面转动时，暂停后再点开始，封面总是从 0 开始旋转而不是之前暂停的设置，将 Storyboard 中 `DubleAnimation` 从 `From/To` 改为 `By` 赋值后解决

五．思考与总结

1. UWP 为我们提供的 `MdiaElement` 可以很方便的播放流媒体文件，自带的功能几乎可以满足日常需求