现代操作系统应用开发实验报告

姓名：_____刘宇庭_____

学号：_____ 16340158_____

实验名称：_____UWP_实验三_____

一、参考资料

week-7 网络访问

1.HttpWebRequest

https://msdn.microsoft.com/zh-cn/library/system.net.httpwebrequest(v=vs.80).aspx

2.HttpClient :

https://msdn.microsoft.com/en-us/library/system.net.http.httpclient(v=vs.118).aspx

3.Json 解析

https://blog.csdn.net/coolszy/article/details/8606803

4.xml 解析

https://msdn.microsoft.com/zh-cn/library/windows/apps/windows.data.xml.dom.xml

document.aspx

week-8 音乐播放器

1. MediaElement

https://msdn.microsoft.com/zh-cn/library/windows/apps/mt187272.aspx

2.MediaPlayerElement

https://docs.microsoft.com/en-us/uwp/api/Windows.UI.Xaml.Controls.MediaPlayerElement

3.使用 MediaPlayer 播放音频和视频

https://docs.microsoft.com/zh-cn/windows/uwp/audio-video-camera/play-audio-and-video-with-mediaplayer

4. 图片旋转

https://blog.csdn.net/hzw2945/article/details/72467820

5. DispatcherTimer

https://msdn.microsoft.com/zh-cn/library/system.windows.threading.dispatchertimer.aspx

6. TimeSpan

https://blog.csdn.net/u010771437/article/details/40372631

7. 情节提要动画

https://docs.microsoft.com/zh-cn/windows/uwp/design/motion/storyboarded-animations

8. UWP AppButtonBar 样式

https://blog.csdn.net/lindexi_gd/article/details/49307913

二、实验步骤

① 网络服务：

要求：

1. 使用 HttpWebRequest 或 HttpClient 访问网络。
2. 输入城市查询天气， 快递查询等生活实用功能至少完成一种。

实验步骤：

1. 设计页面布局。

   使用 RadioButton 变成切换型的，本次做了天气查询和 Ip 归属地查询。

2. 寻找相对应的查询 Api
3. 对相对应的信息进行解析(xml 和 json)并且在页面上显示

② 音乐播放器

要求：

1. 使用 MediaElement 或 MediaPlayer 打造一个播放器，可播放视频和音乐
2. 实现暂停，播放，停止等操作
3. 实现进度条，实时显示媒体的播放速度;同时，拖动进度条，可以使媒体快速前进，后退到相应的位置
4. 播放视频时，可全屏显示、退出全屏

实验步骤：

1. 设计播放器的界面
   - 标题
   - 播放视频或者音乐
   - 功能区

2. 界面功能实现

- 视频进度条的实现。

  将播放的进度条与 MediaPlayer 组件的 MediaTimelineController 的 position 进行双向绑定，并且设计一个计时器，当点击开始播放以后，计时器开始计时，并且添加一个委托函数，每一秒调用一次，函数修改播放进度条的 value 属性，当其播放完时候，将 MediaTimelineController 的 position 置回 0，并且暂停，等待再次点击播放，同时在该函数也处理一些 Ui 的设计。

- 播放、暂停、停止实现

  当 MediaPlayer 组件设置过 MediaTimelineController 的时候，只能通过 MediaTimelineController 进行启动和暂。对于播放和暂停功能的实现，MediaTimelineController 有相对应的功能，对于停止，可以将 MediaTimelineController 的 positon 设置为 0，并且调用其暂停函数。

- 全屏的设置

```
private void FullPrint_Click(object sender, RoutedEventArgs e)
{
    ApplicationView view = ApplicationView.GetForCurrentView();
    if(view.IsFullScreenMode)
    {
        view.ExitFullScreenMode();
    }
    else
    {
        view.TryEnterFullScreenMode();
    }
}
```

- 初始化开始播放时候的音量和设置音量

  只需修改 MediaPlayer 的 Volume 属性

- 选择播放文件实现

```
private async void Add_Click(object sender, RoutedEventArgs e)
{
    var filePicker = new FileOpenPicker();

    filePicker.SuggestedStartLocation = PickerLocationId.VideosLibrary;
    filePicker.FileTypeFilter.Add(".wmv");
    filePicker.FileTypeFilter.Add(".mp4");
    filePicker.FileTypeFilter.Add(".mp3");
    filePicker.FileTypeFilter.Add(".wma");

    StorageFile file = await filePicker.PickSingleFileAsync();

    if (file != null)
    {
        int start = file.Path.LastIndexOf('\\') + 1;
        int end = file.Path.LastIndexOf('.');
        string FileName = file.Path.Substring(start, end - start);
        Debug.WriteLine(FileName);
        fileName.Text = FileName;
        var source = MediaSource.CreateFromStorageFile(file);
        source.OpenOperationCompleted += setTimeLine;
        mediaPlayer.Source = source;
        if (file.FileType == ".mp3" || file.FileType == ".wma")
        {
            LayoutRoot.Visibility = Visibility.Visible;
            myMediaElement.Visibility = Visibility.Collapsed;
            myMediaElement.SetMediaPlayer(mediaPlayer);
        }
        else
        {
            LayoutRoot.Visibility = Visibility.Collapsed;
            myMediaElement.Visibility = Visibility.Visible;
        }
        timeLine.Value = 0;
        Start.Visibility = Visibility.Visible;
        Pause.Visibility = Visibility.Collapsed;
        StopMedia(null, null);
    }
}
```

- 旋转动画实现

```
<Grid x:Name="LayoutRoot" Background="{ThemeResource ApplicationPageBackgroundThemeBrush}" Grid.Row="1" Visibility="Visible">
    <!--Ellipse是绘制一个椭圆形-->
    <Ellipse x:Name="ellipse" Width="500" Height="500" RenderTransformOrigin="0.5,0.5" >
        <Ellipse.RenderTransform>
            <CompositeTransform/>
        </Ellipse.RenderTransform>
        <Ellipse.Resources>
            <!--Storyboard是一个动画容器-->
            <Storyboard x:Name="EllStoryboard" RepeatBehavior="Forever">
                <DoubleAnimation Duration="0:0:20" To="360" Storyboard.TargetProperty="(UIElement.RenderTransform).(CompositeTransform.Rotation)" Storyboard.TargetName="ellipse" d:IsOptimized="True"/>
            </Storyboard>
        </Ellipse.Resources>
        <!--这是用一张图片来填充这个椭圆形-->
        <Ellipse.Fill>
            <ImageBrush ImageSource="Assets\pic_1.jpg" />
        </Ellipse.Fill>
    </Ellipse>
</Grid>
```

Ellipse 是一个椭圆绘图工具，可以绘制圆形，它有一个 Ellipse.RenderTransform 标签可以定义动作变换； Storyboard 可以看作是一个动画的容器，DoubleAnimation 定义的动画嵌套在 StoryBoard 内，Storyboard.TargetName

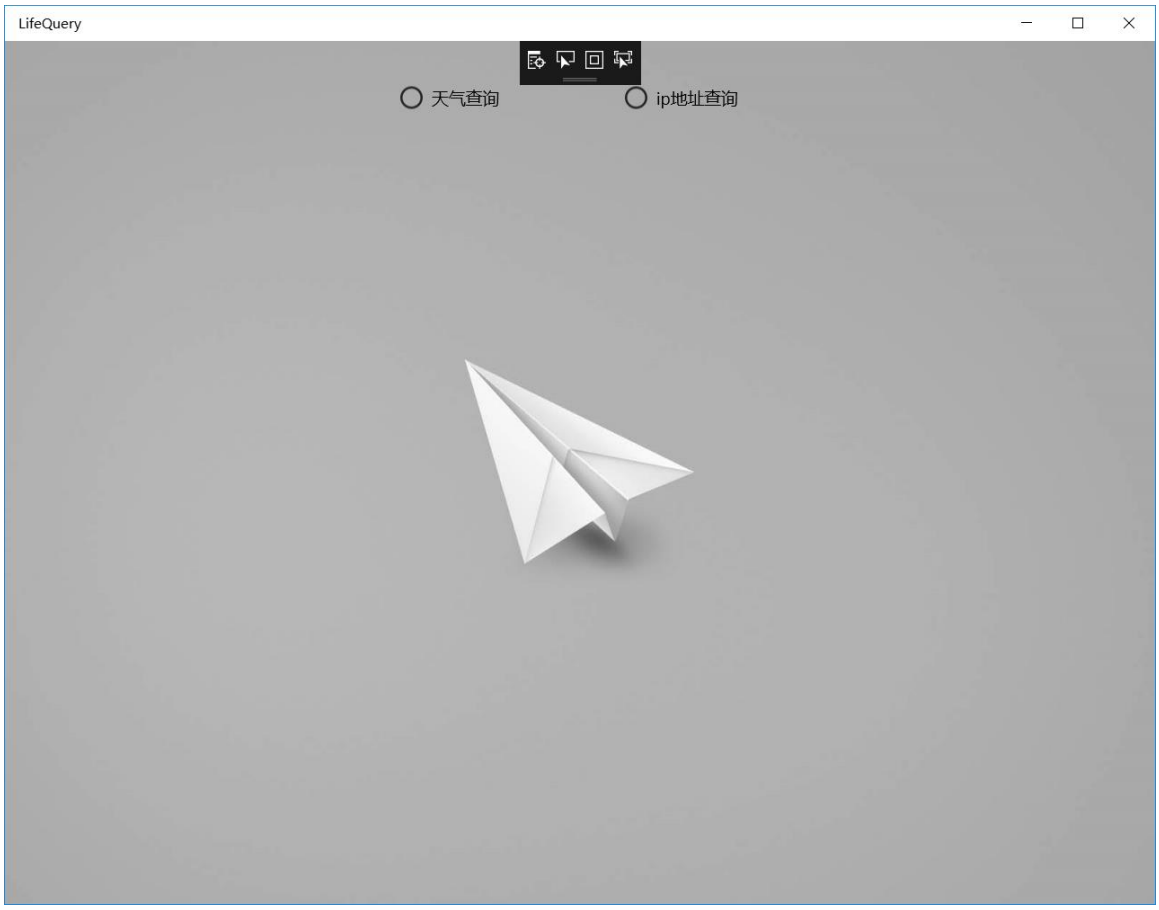
表示动画作用的控件， Storyboard.TargetProperty 表示动画作用的属性；

Ellipse.Fill 表示 Ellipse 所绘制图形中索要填充的内容。

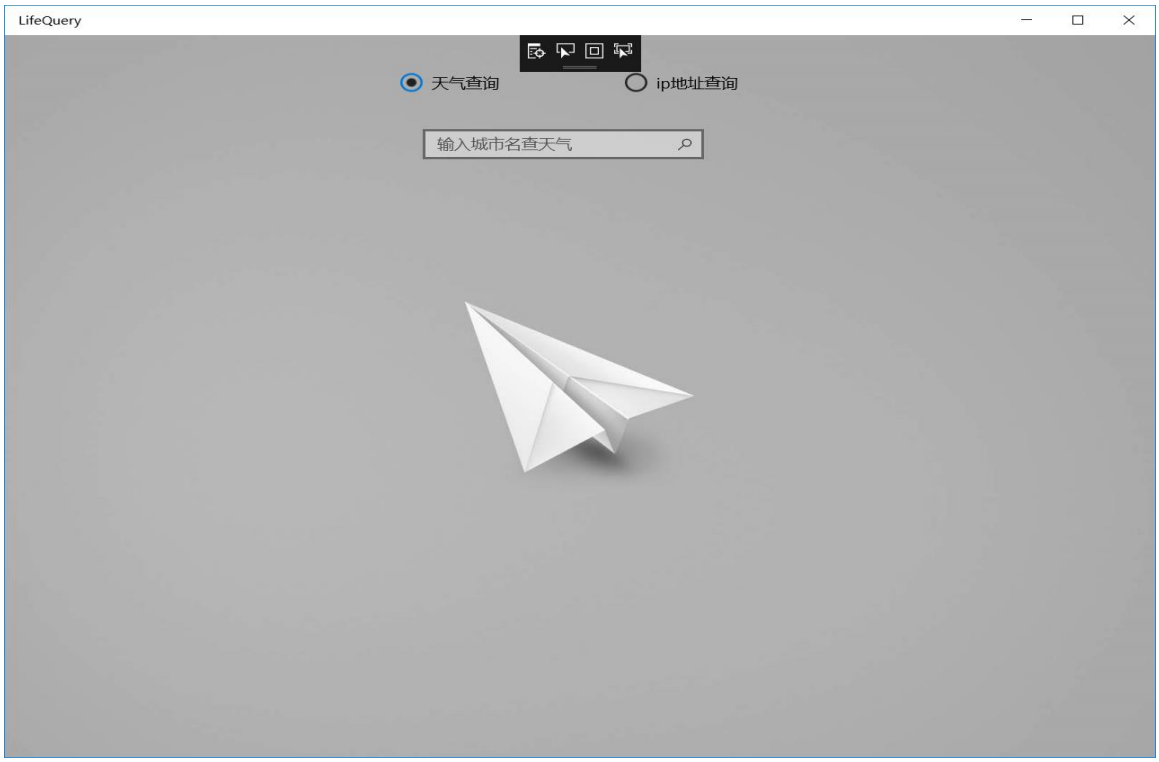
三、关键步骤截图

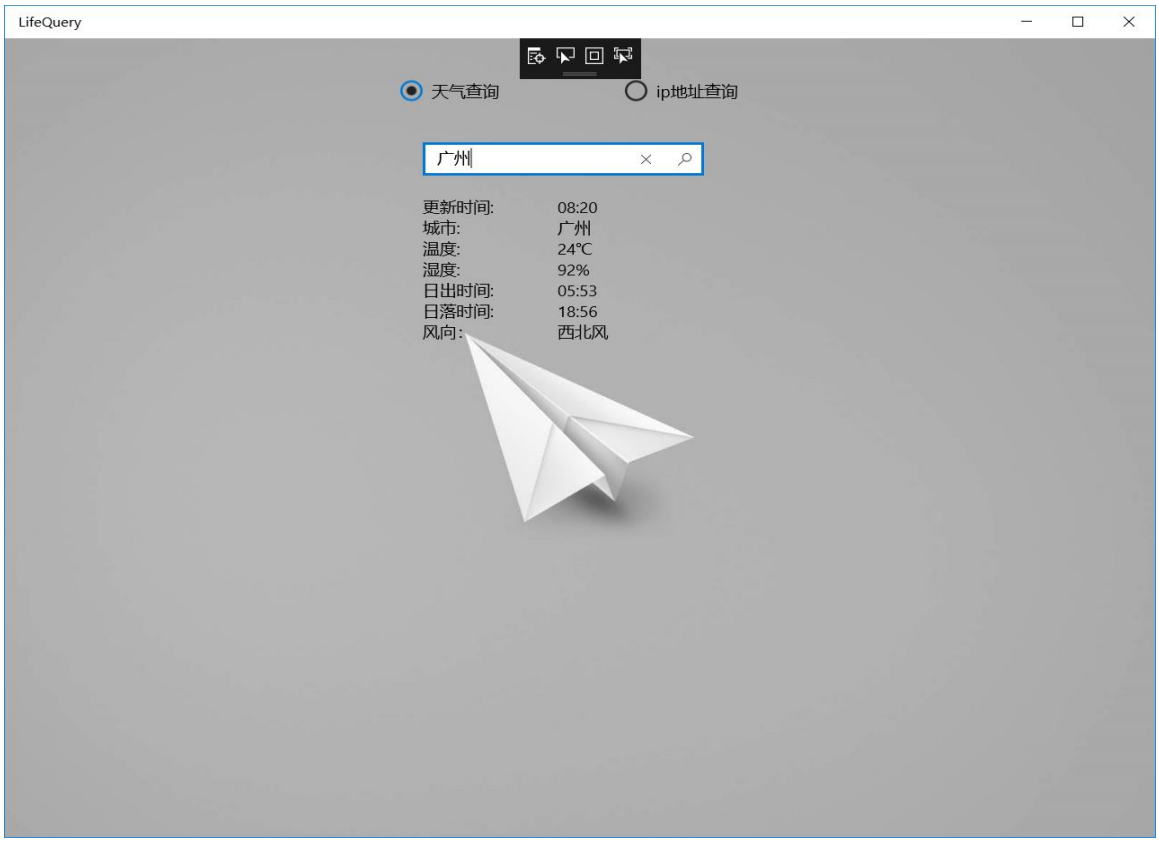
Week-7：

主页面：

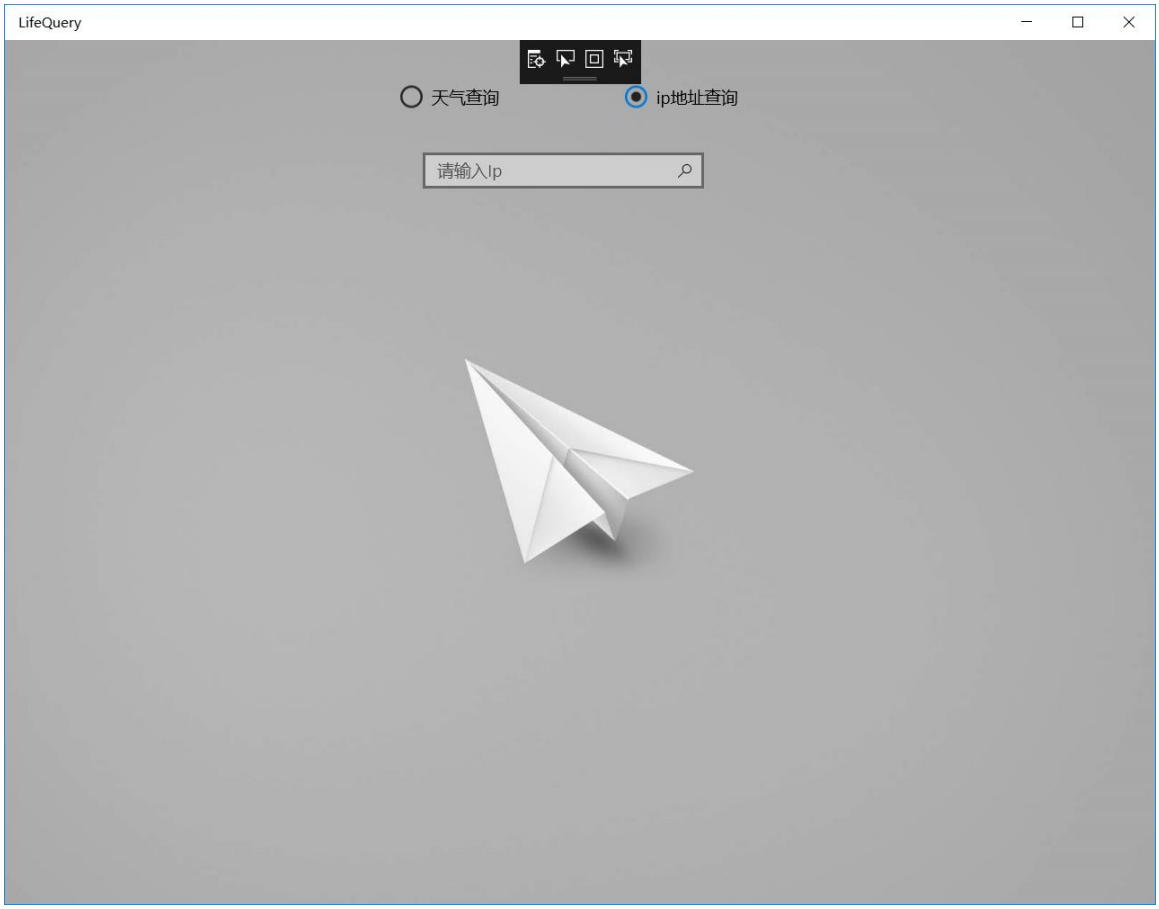


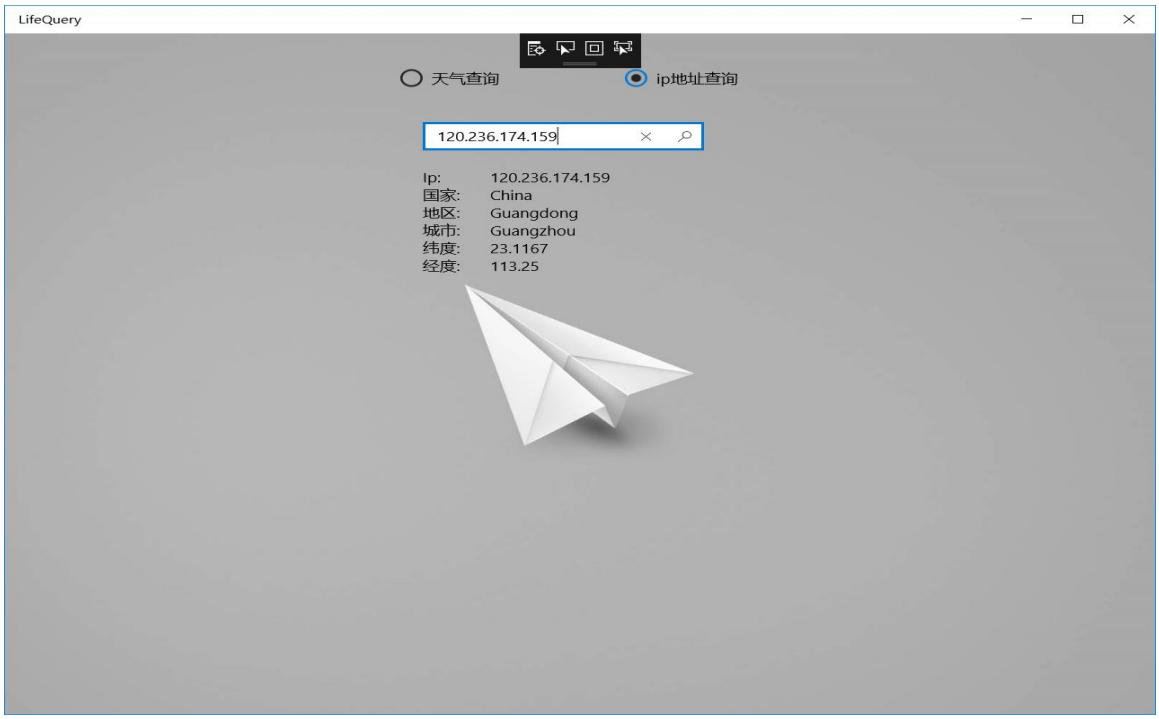
天气查询：


LifeQuery
天气查询
ip地址查询
输入城市名查天气


LifeQuery
天气查询
ip地址查询
广州
更新时间: 08:20
城市: 广州
温度: 24℃
湿度: 92%
日出时间: 05:53
日落时间: 18:56
风向: 西北风

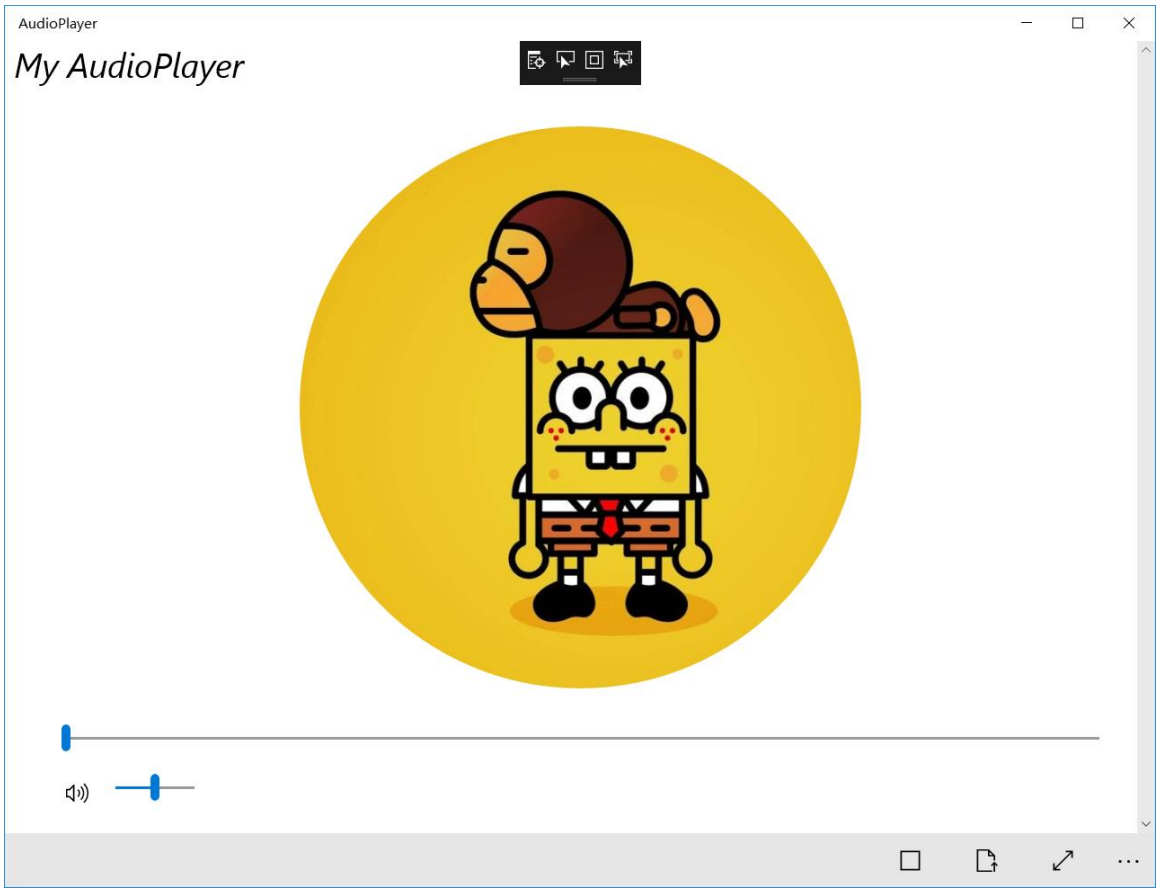
Ip 归属查询：


LifeQuery
天气查询
ip地址查询
请输入Ip

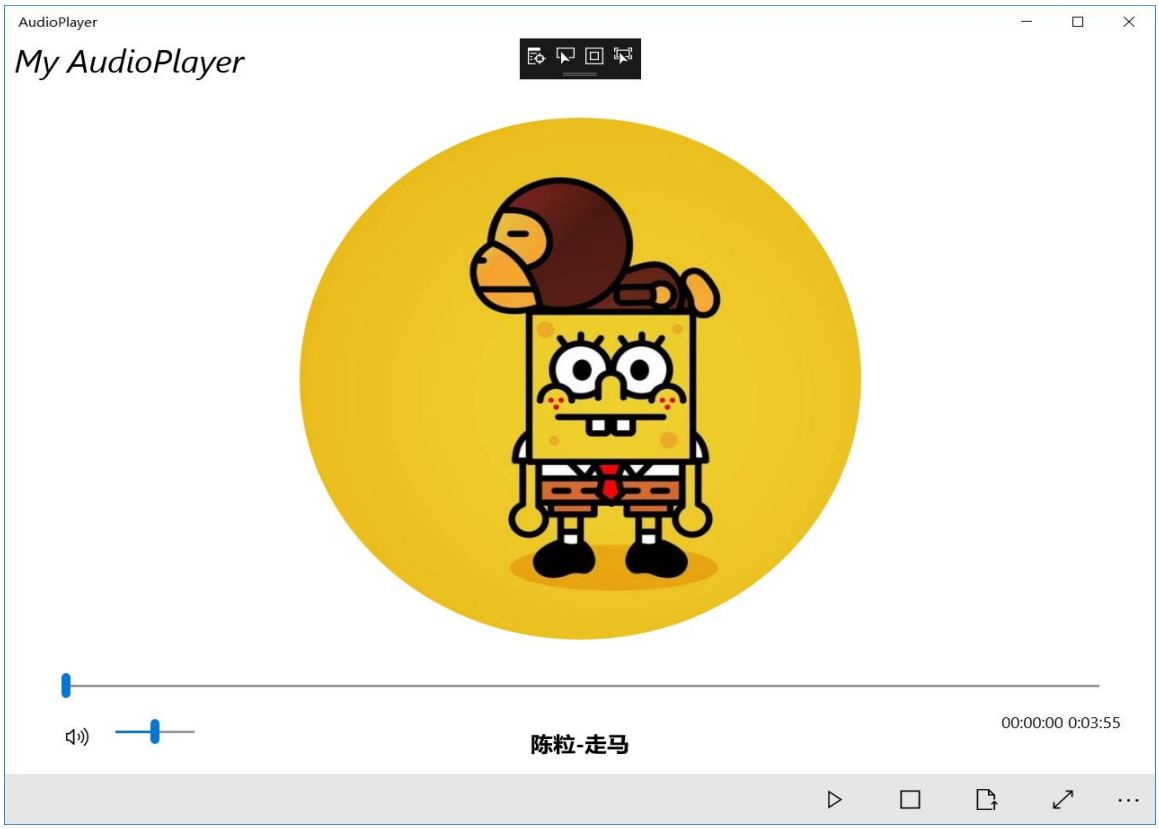

LifeQuery
天气查询
ip地址查询
120.236.174.159
Ip: 120.236.174.159
国家: China
地区: Guangdong
城市: Guangzhou
纬度: 23.1167
经度: 113.25

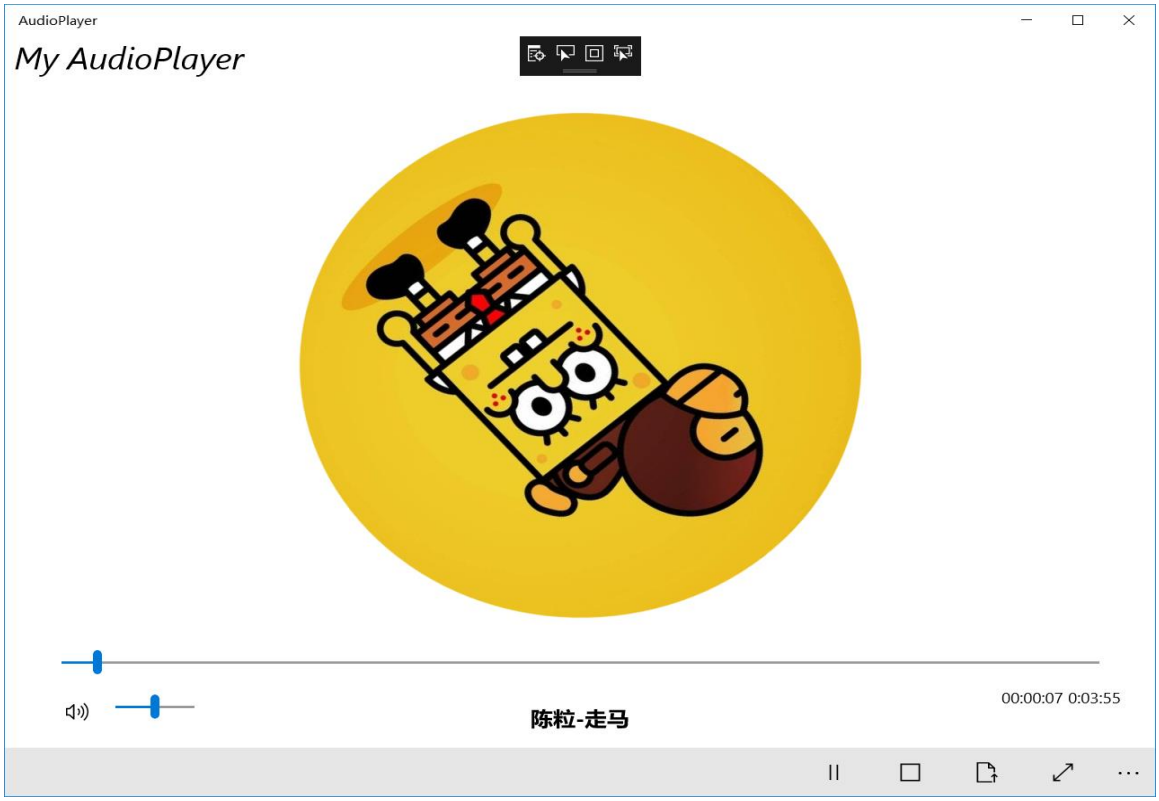
Week-8

主页面：（当选择了文件后会出现播放按钮）



选择一个音频文件并且播放：





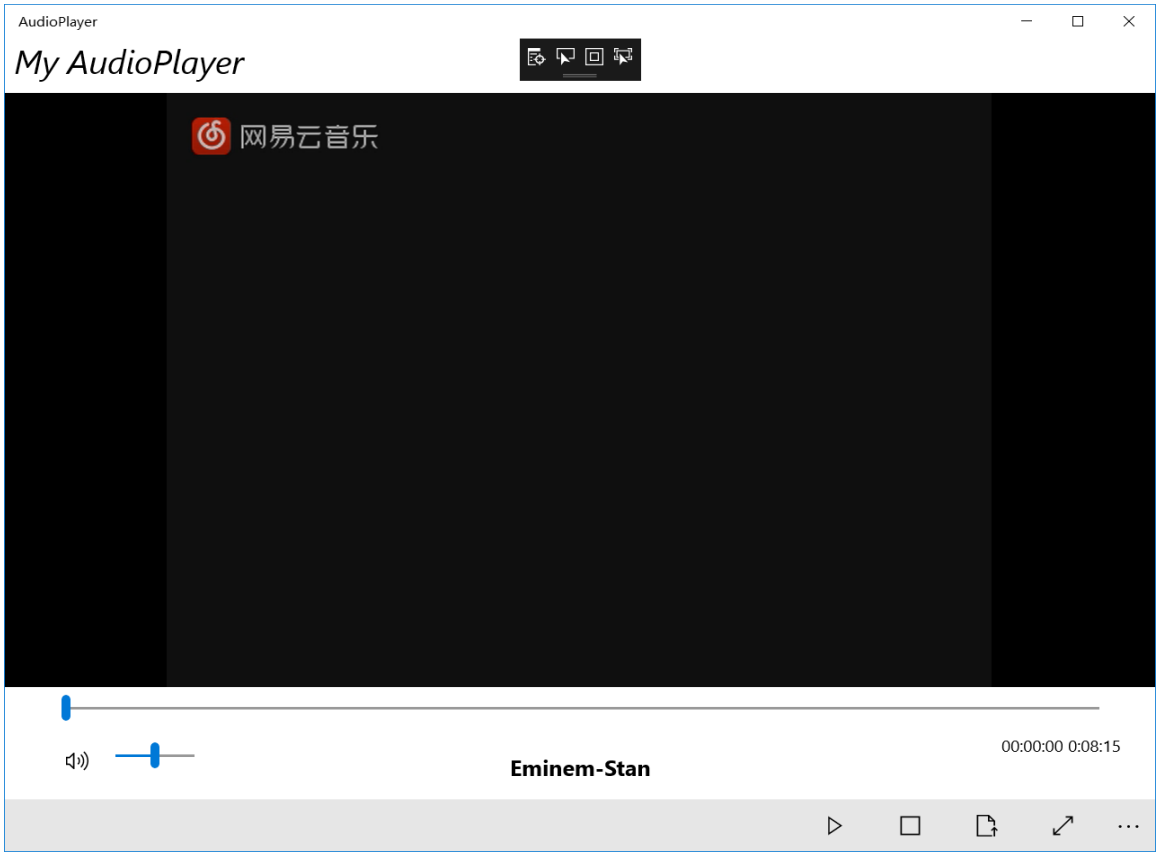
暂停：


AudioPlayer
My AudioPlayer
00:00:16 0:03:55
陈粒-走马

停止：


AudioPlayer
My AudioPlayer
00:00:00 0:03:55
陈粒-走马

选择一个视频并且播放：


AudioPlayer
My AudioPlayer
网易云音乐
00:00:00 0:08:15
Eminem-Stan


AudioPlayer
My AudioPlayer
网易云音乐
00:00:04 0:08:15
Eminem-Stan

全屏：

四、亮点与改进

亮点：

1. 可以自己选择播放的文件
2. 旋转图片
3. 播放时间与总时间
4. 播放文件名字
5. 音量调节

五、遇到的问题

1. 播放的时候，有时候选择视频，只有声音没有图像，但是选择个音频文件后再次选择又可以正常播放，部分 MP4 文件无法播放
2. 如何将播放的文件与进度条进行绑定起来。

解决方法：

设计一个转换器 converter，将播放进度条的 value 转化成 TimeSpan 类型然后返回给 MediaPlayer 里的 MediaTimelineController 的 position，同时有将 MediaTimelineController 的 position 转化成 double 类型返回给播放进度条。

```
using System.Text;
using System.Threading.Tasks;
using Windows.UI.Xaml.Data;

namespace AudioPlayer.Converter
{
    class TimeConverter : 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);
        }
    }
}
```

3. 播放长视频的时候，播放的进度条拖动后有时候会出现来回跳，双向绑定感觉并没有及时同步。

六、思考与总结

本次主要学习了网络访问以及播放器制作的相关知识，主要还是学习调用相关的 api 进行实现。对于网络访问部分，主要的问题在于对得到的 json 和 xml 信息进行解析和展示（我感觉找一个免费的查询 api 也挺难的），主要还是通过相对应的 parse 函数进行解析然后获取信息。对于音乐播放器，主要问题在于播放文件和时间进度条的绑定，最后通过完成一个转换器进行双向绑定。本次 uwp 的学习也结束了，但是真正的学习也只是刚刚开始吧，接下来还要自己继续学习吧。