**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\_\_\_\_\_\_\_\_

1. 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.xmldocument.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

1. 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。

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

1. 界面功能实现

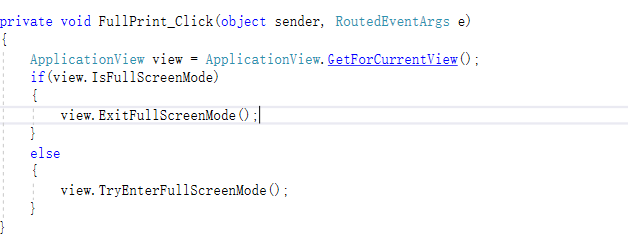
* 视频进度条的实现。

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。

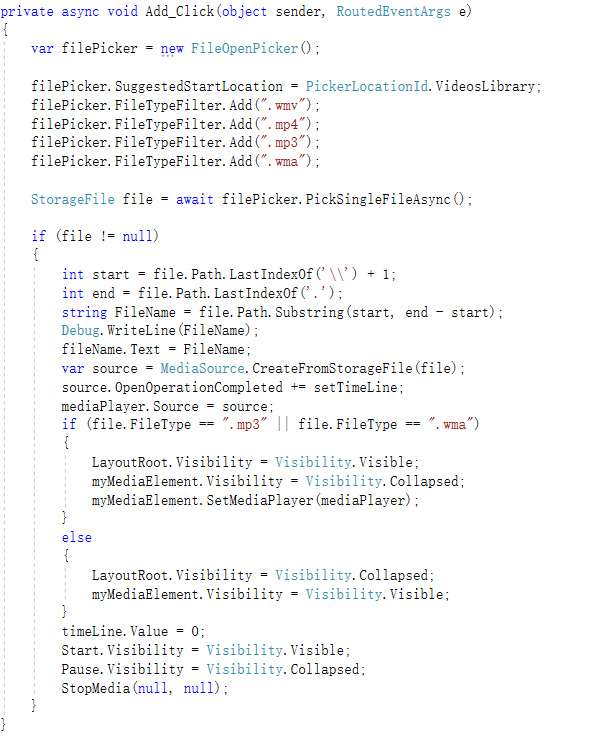
* 全屏的设置



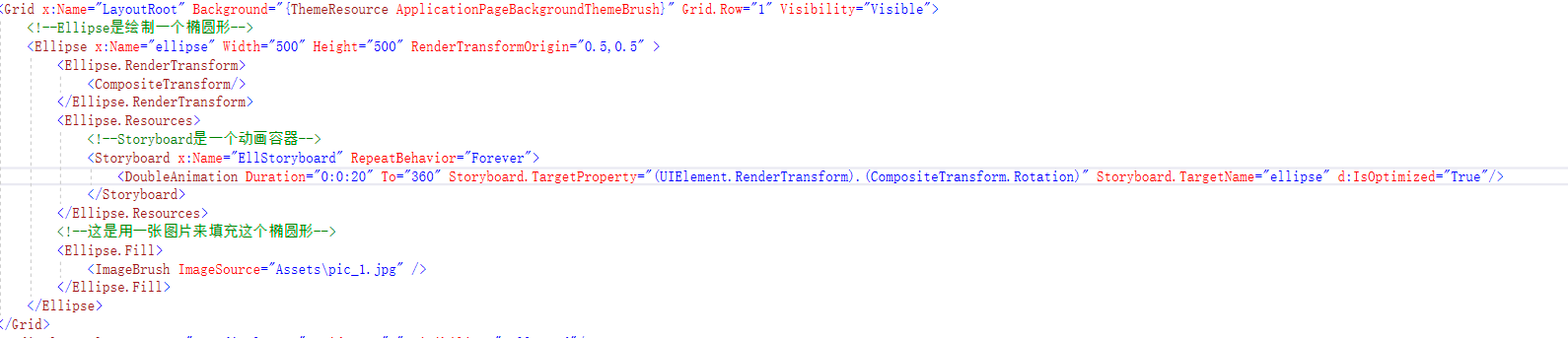
* 初始化开始播放时候的音量和设置音量

只需修改MediaPlayer的Volume属性

* 选择播放文件实现



* 旋转动画实现

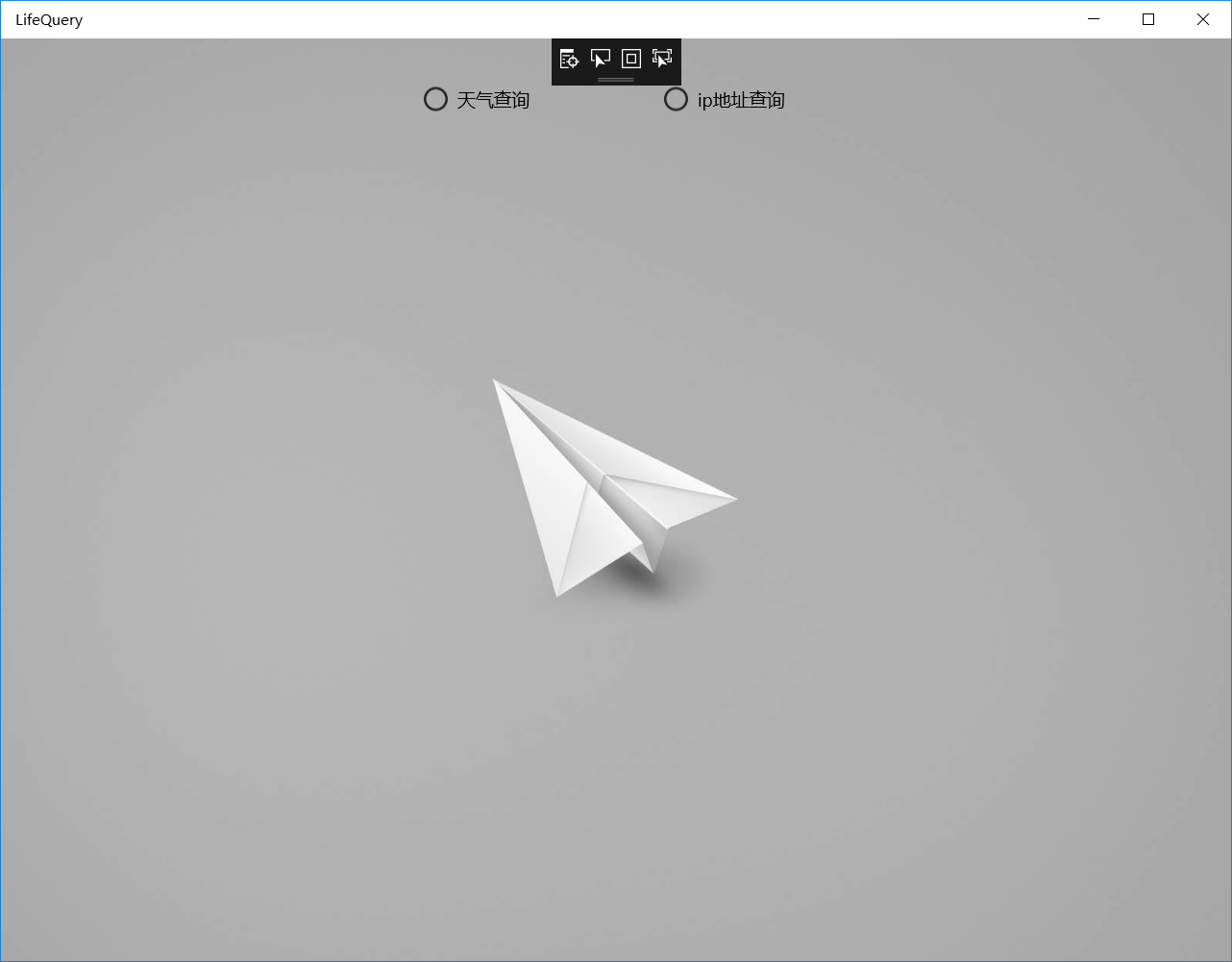


Ellipse是一个椭圆绘图工具，可以绘制圆形，它有一个Ellipse.RenderTransform标签可以定义动作变换；Storyboard可以看作是一个动画的容器，DoubleAnimation定义的动画嵌套在StoryBoard内，Storyboard.TargetName表示动画作用的控件，Storyboard.TargetProperty表示动画作用的属性；Ellipse.Fill表示Ellipse所绘制图形中索要填充的内容。

1. 关键步骤截图

Week-7：

主页面：

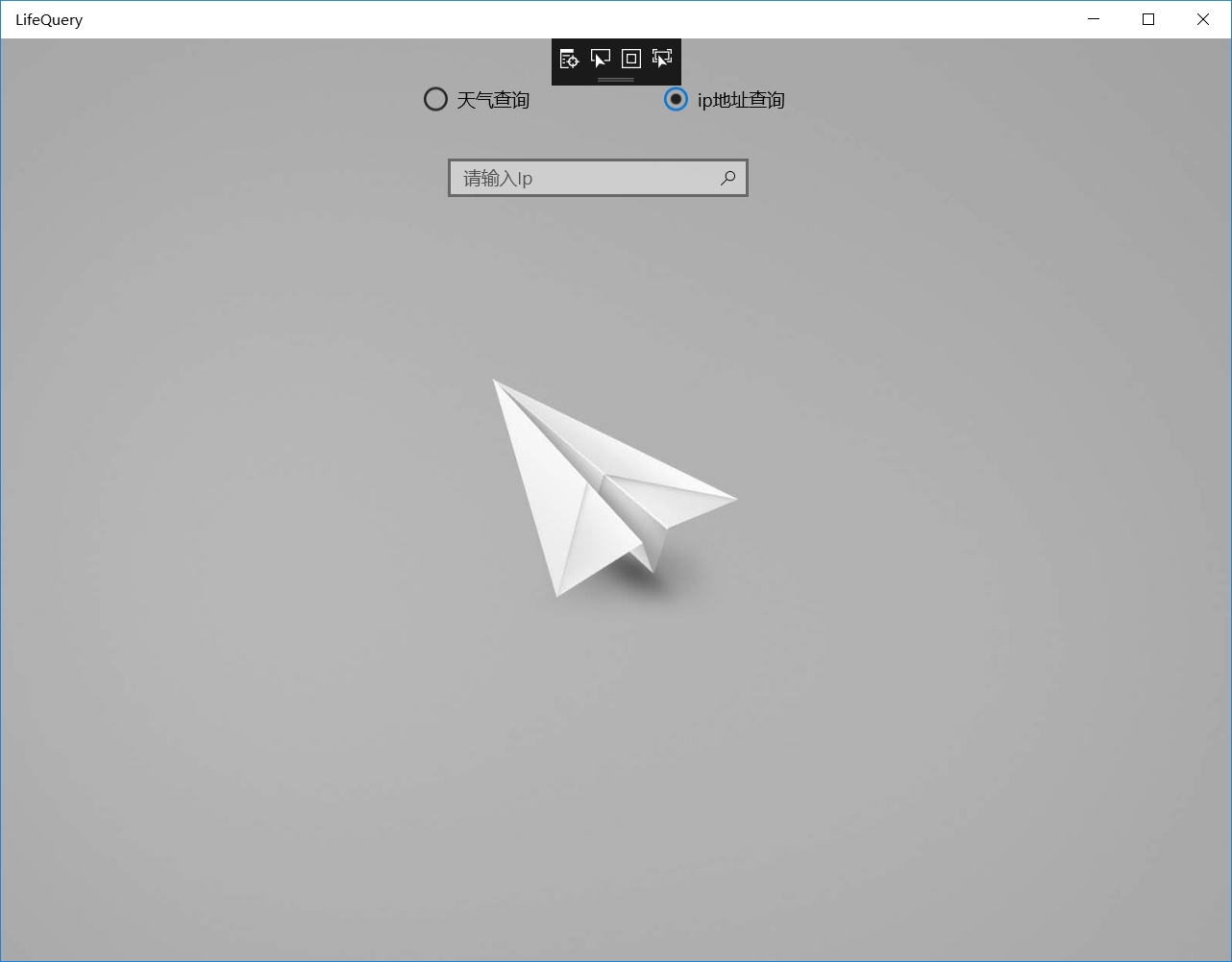


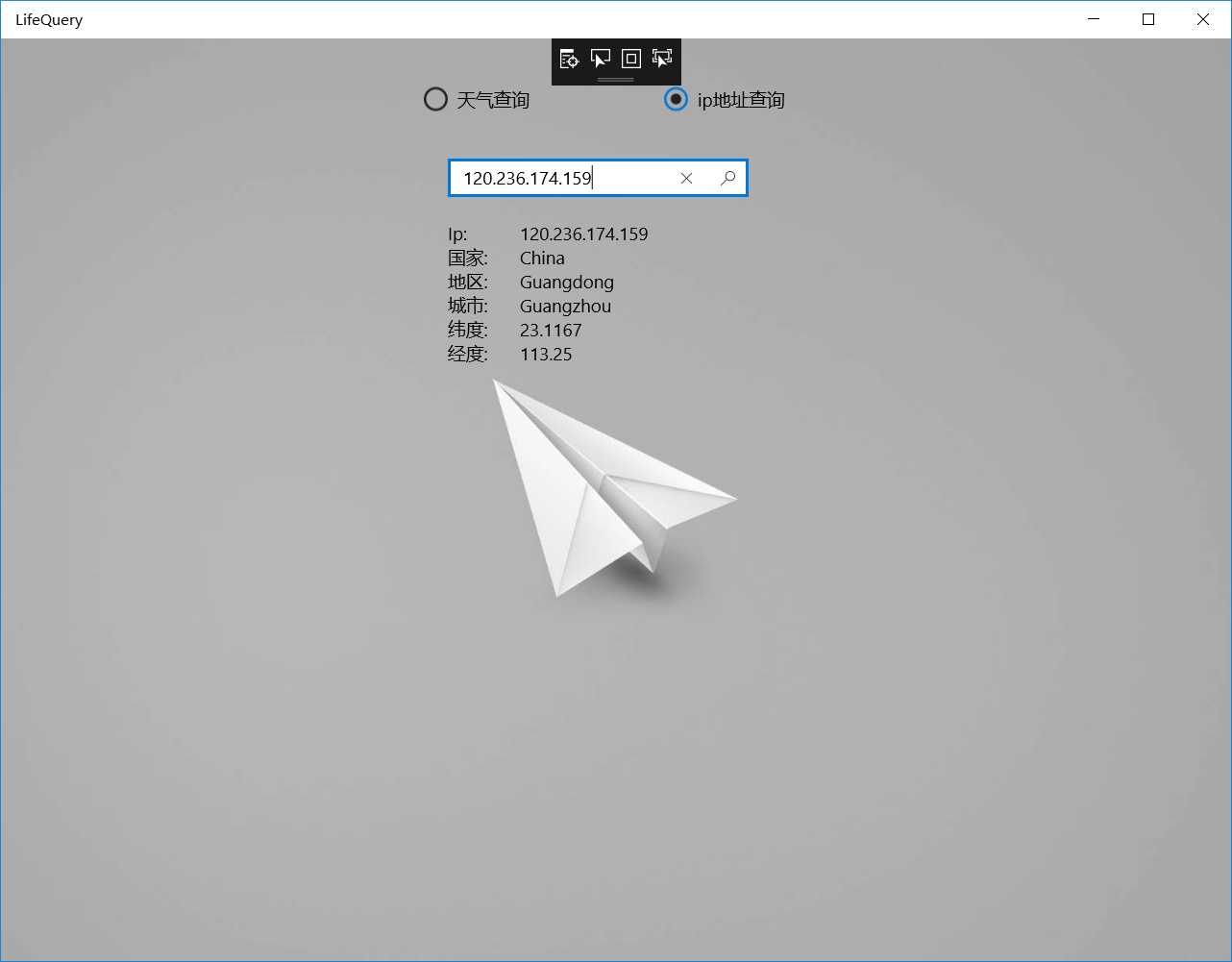
天气查询：





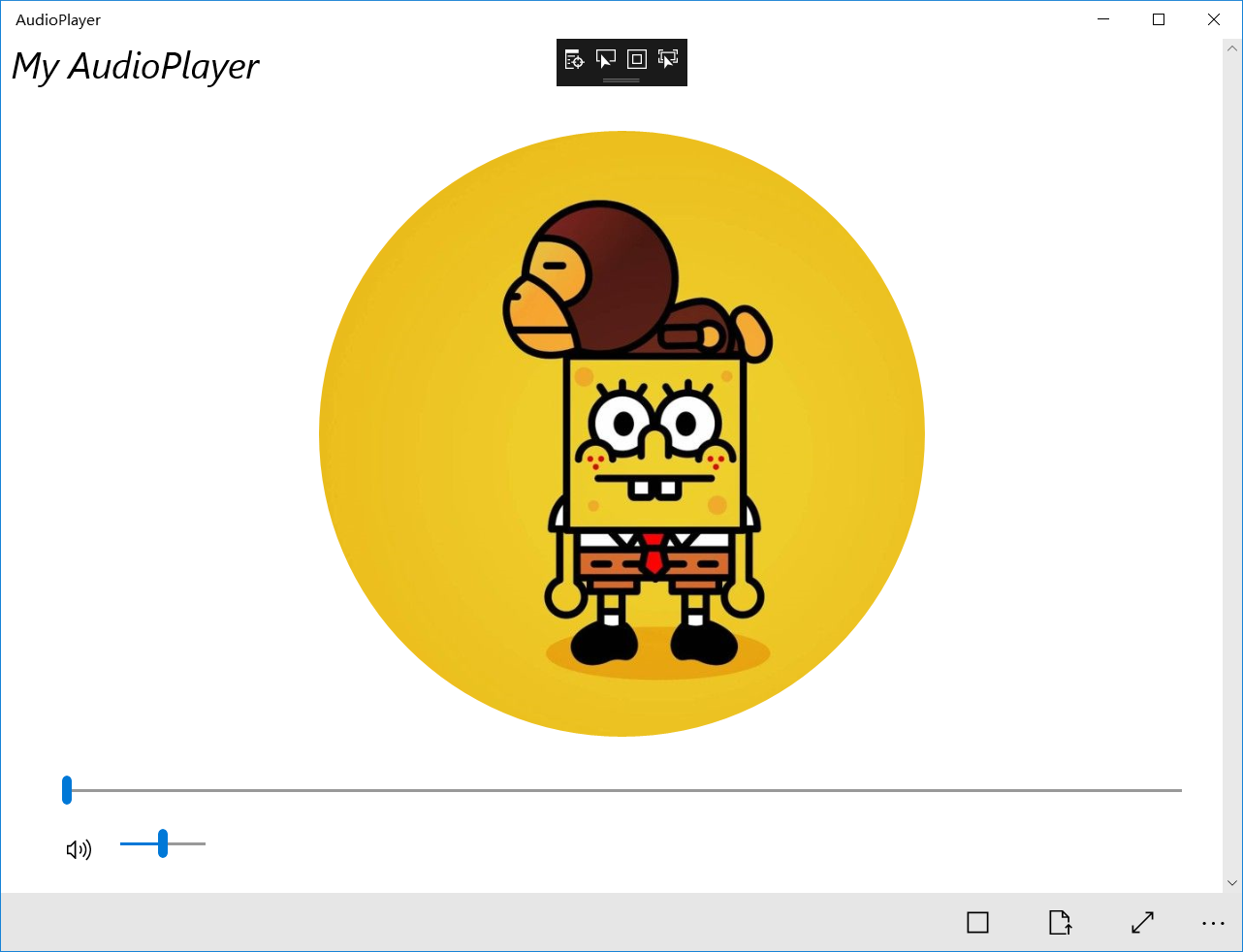
Ip归属查询：



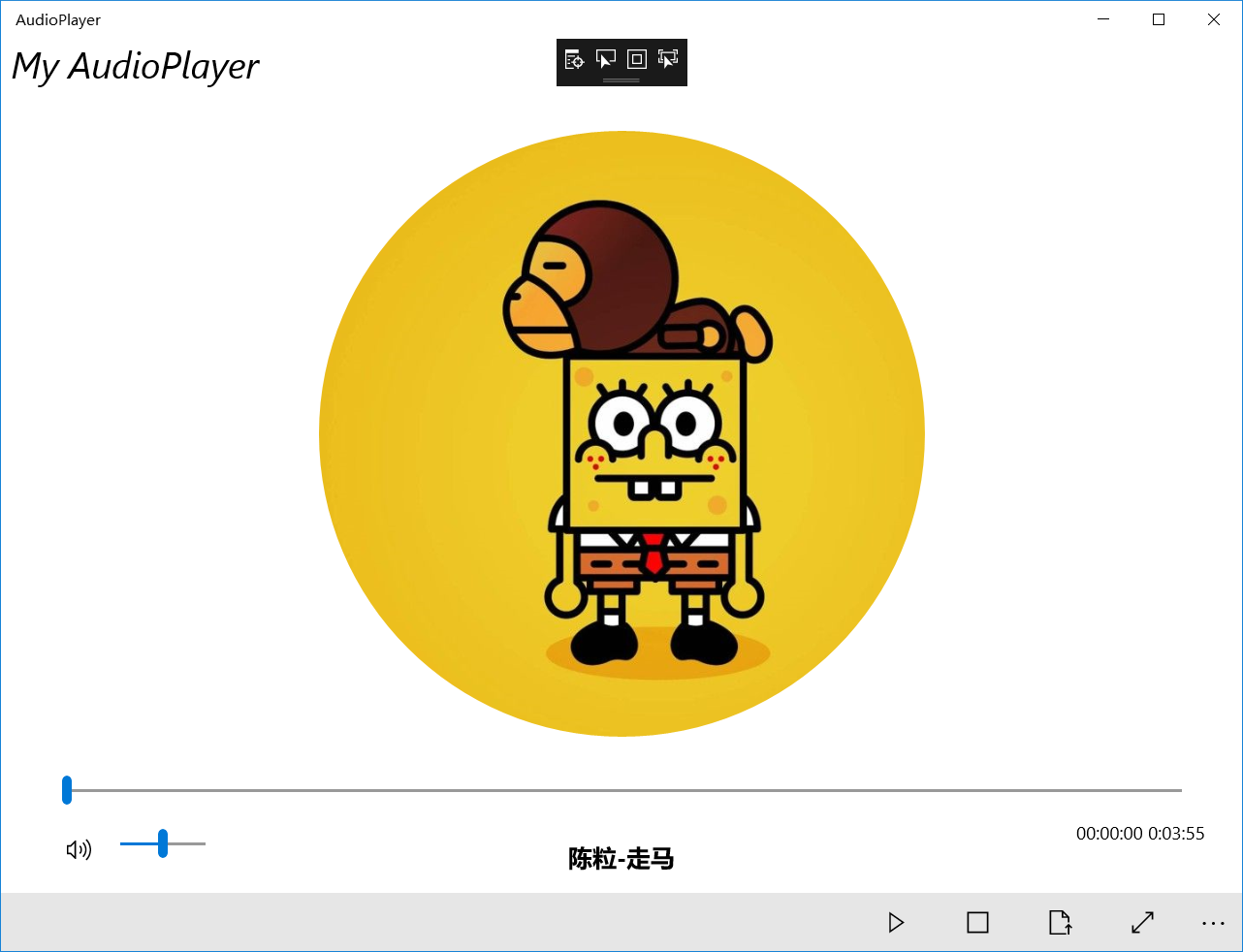


Week-8

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：





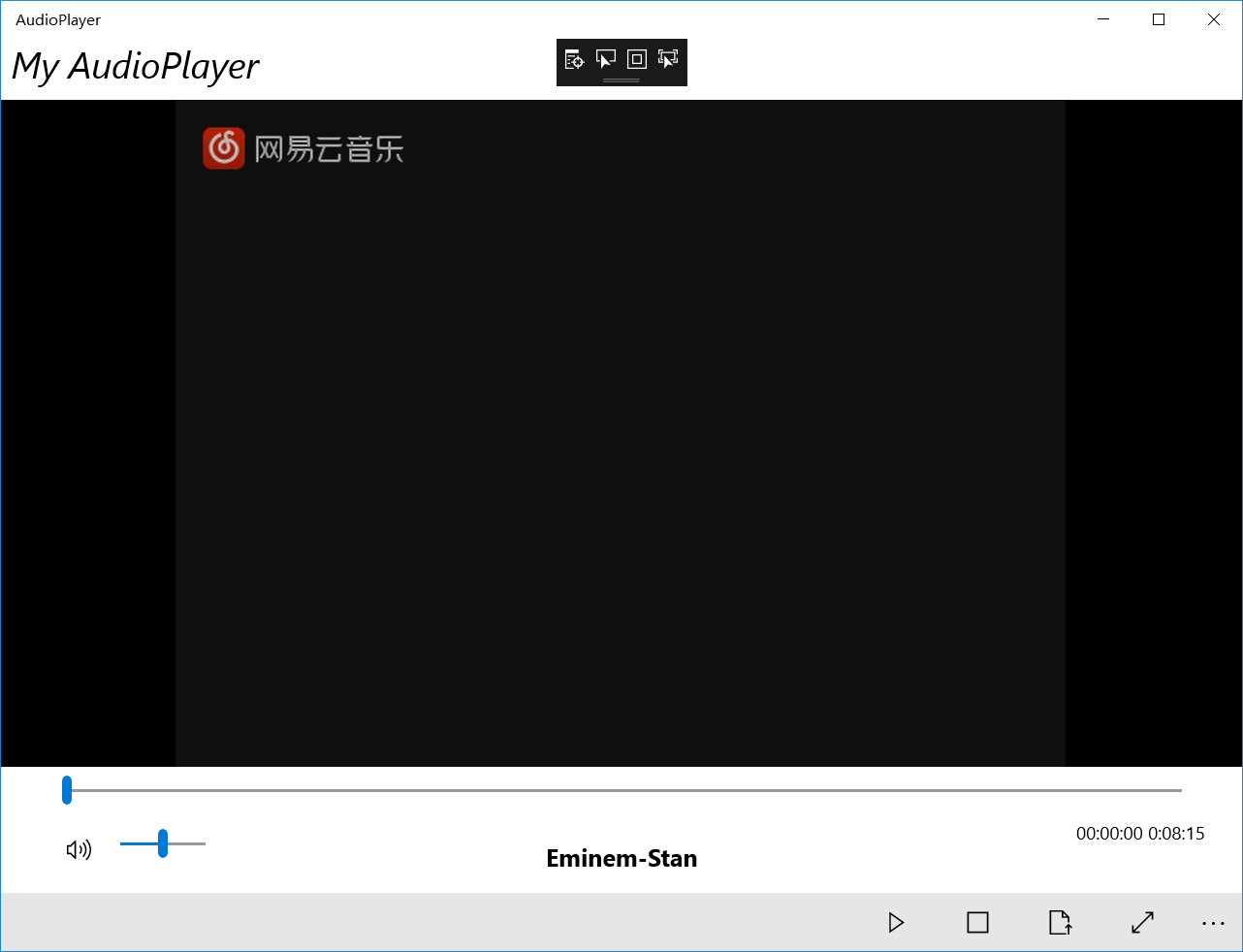
暂停：



停止：



选择一个视频并且播放：





全屏：



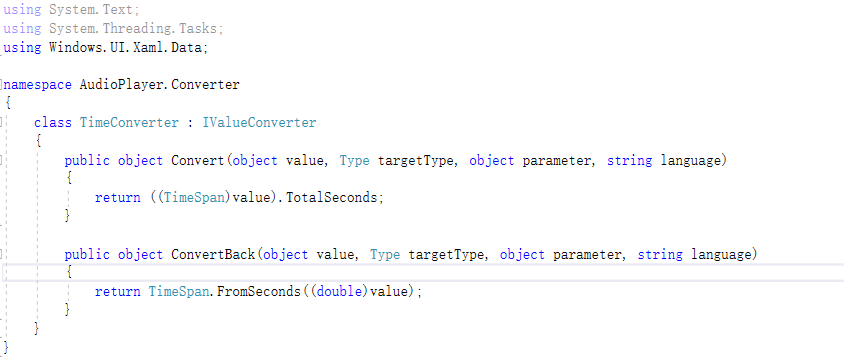
1. 亮点与改进

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
6. 遇到的问题
7. 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
8. 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。



1. 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。
2. 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。