现代操作系统应用开发实验报告

姓名：余漫霖

学号：16340280

实验名称：网络访问与播放器

一、参考资料

使用HttpWebRequest从API获取Json文件

使用HttpWebRequest从API获取Xml文件

在uwp应用中实现全屏/退出全屏

将缩略图StorageItemThumbnail赋值给Image控件的绑定源

uwp实现图片旋转动画

Xaml控件的Symbol枚举值

用slider控件来同步显示MediaElement播放进度

自定义定时器修改MediaElement的Position属性

二、实验步骤

网络访问

1. 编写相关**Xaml**

2. 从简繁转换**API**获取**Json**格式的数据并解析

```
String sim = simple.Text;
String appkey = "33227";
String sign = "1aad55b9adfb581a932497a612860997";
String format = "http://api.k780.com/?app=code.hanzi_fanjian&typeid=1&wd=" + sim +
  "&appkey=" + appkey + "&sign=" + sign + "&format=json";

//访问API
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(format);
//读取数据
HttpWebResponse response = (HttpWebResponse)(await request.GetResponseAsync());
Stream stream = response.GetResponseStream();
StreamReader streamReader = new StreamReader(stream);
string strResult = streamReader.ReadToEnd();
stream.Dispose();
streamReader.Dispose();

//解析Json
JsonObject rootObject = JsonObject.Parse(strResult);
if (rootObject["success"].GetString().Equals("1"))
  traditional.Text = rootObject["result"].GetObject()["text"].GetString();
else traditional.Text = "出现异常，转换失败！";
```

3. 从汇率查询API获取Xml格式的数据并解析

```
ComboBoxItem srcItem = ((ComboBoxItem)srcCurrency.SelectedItem);
String src = (String)srcItem.Content;
ComboBoxItem dstItem = ((ComboBoxItem)dstCurrency.SelectedItem);
String dst = (String)dstItem.Content;
src = src.Substring(src.Length - 3);
dst = dst.Substring(dst.Length - 3);

String appkey = "33227";
String sign = "1aad55b9adfb581a932497a612860997";
String format = "http://api.k780.com/?app=finance.rate&scur=" + src + "&tcur=" +
  dst + "&appkey=" + appkey + "&sign=" + sign + "&format=xml";

//访问API
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(format);
request.Method = "GET";
request.ContentType = "application/xml";
//读取数据
HttpWebResponse response = (HttpWebResponse)(await request.GetResponseAsync());
Stream stream = response.GetResponseStream();
StreamReader streamReader = new StreamReader(stream);
string xmlResult = streamReader.ReadToEnd();
stream.Dispose();
streamReader.Dispose();

//解析Xml
XmlDocument xmlDocument = new XmlDocument();
xmlDocument.LoadXml(xmlResult);
XmlNode root = xmlDocument.ChildNodes[1];
if(root.ChildNodes[0].InnerText == "0")
{
    rate.Text = "Error";
    return;
}
XmlNode result = root.ChildNodes[1];
rate.Text = result.ChildNodes[4].InnerText;
```

播放器

1. 编写相关Xaml

添加了 `MediaElement` （播放媒体文件）、 `Ellipse` （音乐专辑图封面旋转动画）、 `Slider` （进度条，用于同步媒体播放进度）、 `Page.BottomAppBar` （对媒体的播放设置等按钮）等元素。

2. 设置Page.BottomAppBar中的AppBarButton的相关方法

```
//打开媒体文件
private async void openMedia_Click(object sender, RoutedEventArgs e);

//播放/暂停按钮的操作
private void playControl_Click(object sender, RoutedEventArgs e);

//停止按钮的操作
private void stop_Click(object sender, RoutedEventArgs e);

//修改音量
private void volumn_Changed(object sender, RangeBaseValueChangedEventArgs e);

//全屏/退出全屏操作
private void scale_Click(object sender, RoutedEventArgs e);

//播放器进度条改变时的操作
private void mediaSlider_Changed(object sender, RangeBaseValueChangedEventArgs e);

//点击显示/隐藏进度条的操作
private void showSlider_Click(object sender, RoutedEventArgs e);
```

3. 设置MediaElement和同步媒体播放进度的Slider的相关方法

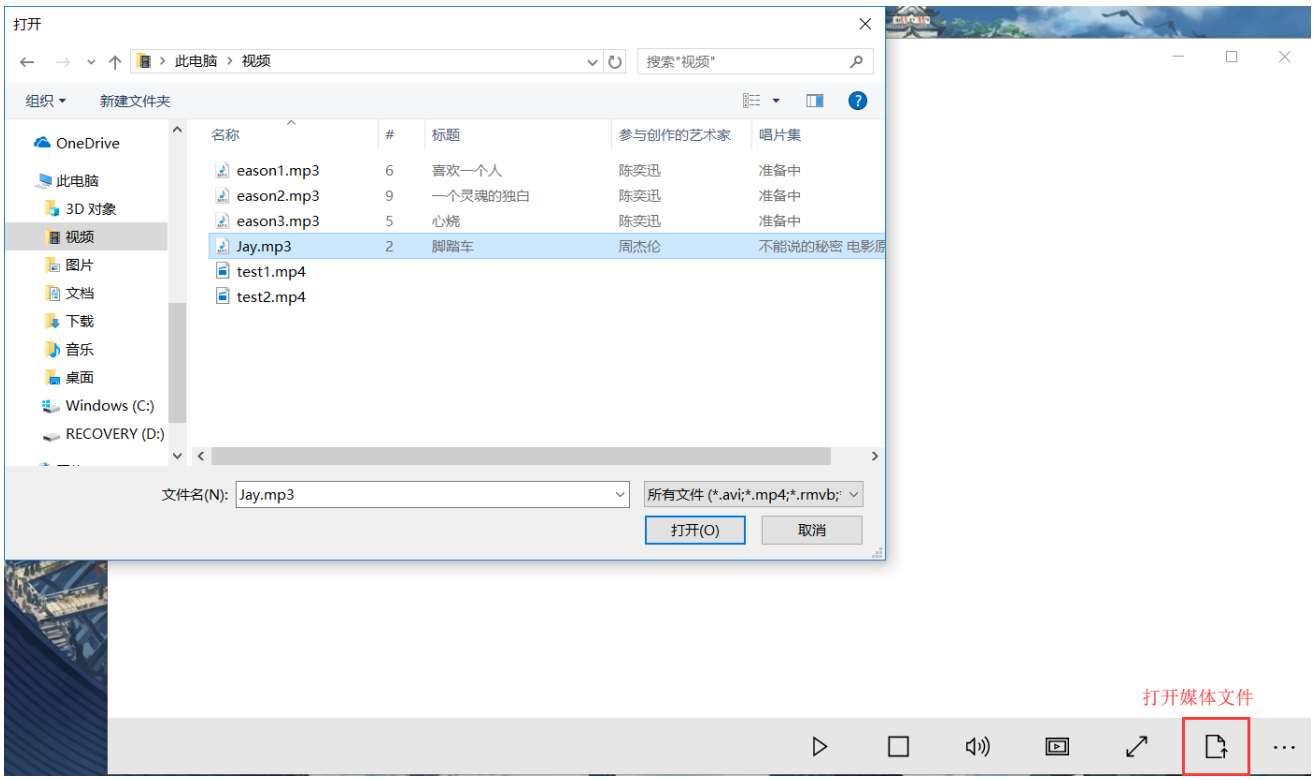
```
//mediaPlayer完成加载时的操作
private void mediaPlayer_Opened(object sender, RoutedEventArgs e);

//计时事件，控制进度条滑块的移动
private void timer_tick(object sender, object e);

//媒体结束播放时的操作
private void mediaPlayer_Ended(object sender, RoutedEventArgs e);
```

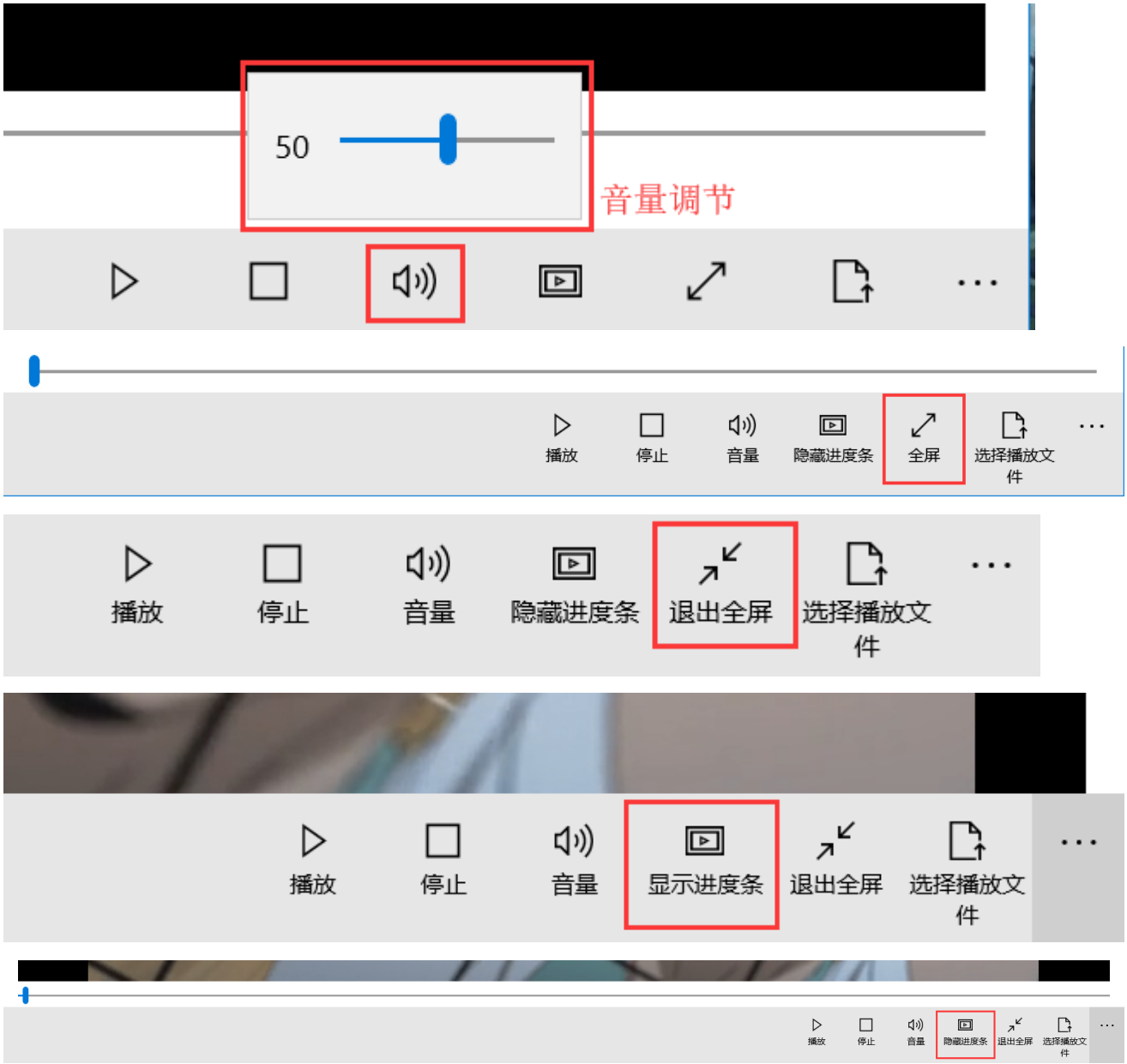
三、关键步骤截图

网络访问


Week7
简体字转繁体字
请输入要转换成繁体字的简体字:
忧郁的台湾乌龟
提交
转换结果:
憂鬱的颱灣烏龜
币种汇率查询
1
欧元EUR
兑换
0.881994
英镑GBP
查询

播放器


打开
此电脑 › 视频
搜索"视频"
组织
新建文件夹
OneDrive
此电脑
3D 对象
视频
图片
文档
下载
音乐
桌面
Windows (C:)
RECOVERY (D:)
名称
#
标题
参与创作的艺术家
唱片集
eason1.mp3 6 喜欢一个人 陈奕迅 准备中
eason2.mp3 9 一个灵魂的独白 陈奕迅 准备中
eason3.mp3 5 心烧 陈奕迅 准备中
Jay.mp3 2 脚踏车 周杰伦 不能说的秘密 电影原
test1.mp4
test2.mp4
文件名(N): Jay.mp3
所有文件 (*.avi;*.mp4;*.rmvb;
打开(O)
取消
打开媒体文件



四、亮点与改进

网络访问

分别使用了Json格式和Xml格式来完成不同的API。详见【二、实验步骤】。

播放器

1. 手动选择本地的媒体资源进行播放

添加了打开文件的按钮。点击后可触发以下操作：

```
private async void openMedia_Click(object sender, RoutedEventArgs e);
```

2. 实现封面旋转

```
        <Ellipse Grid.Row="0" x:Name="ellipse" Visibility="Collapsed" VerticalAlignment="Center"
HorizontalAlignment="Center" Width="300" Height="300" RenderTransformOrigin="0.5,0.5">
            <Ellipse.RenderTransform>
                <CompositeTransform />
            </Ellipse.RenderTransform>
            <Ellipse.Fill>
                <ImageBrush x:Name="imageBrush" ImageSource="Assets/timg.jpg"
Stretch="UniformToFill"></ImageBrush>
            </Ellipse.Fill>
            <Ellipse.Resources>
                <Storyboard x:Name="storyboard" RepeatBehavior="Forever">
                    <DoubleAnimation Duration="0:0:20" To="360" Storyboard.TargetProperty="
(UIElement.RenderTransform).(CompositeTransform.Rotation)" Storyboard.TargetName="ellipse"/>
                </Storyboard>
            </Ellipse.Resources>
        </Ellipse>
```

3. 一些微小的修改（相对于Demo而言）

- 将播放按钮与暂停按钮合并，使之更符合实际中的播放器。（同一按钮，可视情况而显示为播放或暂停的符号。）

- 全屏按钮可视情况而切换为退出全屏按钮。
- 添加了显示/隐藏进度条的按钮，使得在全屏状态下也可以调出进度条。（不会做鼠标移过而自动浮出进度条的效果，只能这样做了。）

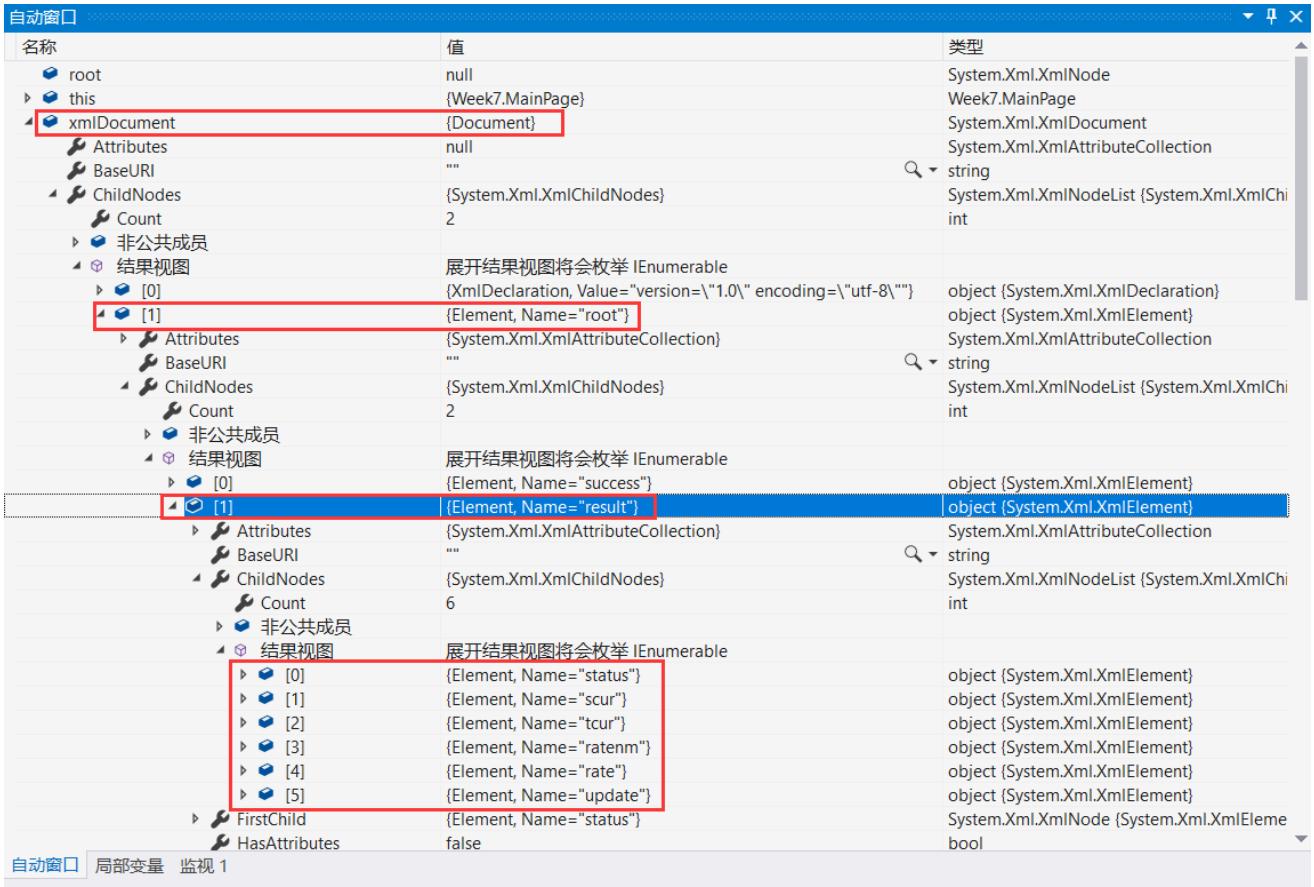
五、遇到的问题

网络访问

1. 接口网站未给出Xml文件的结构，不知道怎么取到所需的部分数据

解决：在语句 `XmlNode root = xmlDocument.ChildNodes[1];` 处添加断点，进行调试，然后在自动窗口里查看 `XmlDocument` 类型的变量 `xmlDocument` 的值。可以看到元素的嵌套结构和序号、名称等等。

`Name=";rate";` 对应的元素即为所需的数据（汇率）。



播放器

1. 使用数据绑定进行进度条的**Value**与**MediaElement**的**Position**属性的双向绑定，但播放后进度条滑块并不滚动。

分析：查了资料，有的人说和循环赋值有关，有的人说（WPF中的）MediaElement的Position属性不是依赖属性，不能进行双向绑定。

解决：最后，改成使用DispatcherTimer来更新进度条的滑块：

```
//mediaPlayer完成加载时的操作
private void mediaPlayer_Opened(object sender, RoutedEventArgs e)
{
    //读取媒体时间长度
    mediaSlider.Maximum = mediaPlayer.NaturalDuration.TimeSpan.TotalSeconds;

    //新的计时器
    timer = new DispatcherTimer();
    timer.Interval = TimeSpan.FromSeconds(0.1);
    timer.Tick += new EventHandler<object>(timer_tick);
    timer.Start();
}

//计时事件，控制进度条滑块的移动
private void timer_tick(object sender, object e)
{
    mediaSlider.Value = mediaPlayer.Position.TotalSeconds;
}
```

2. 专辑封面图不清晰。

解决：增大 `Storage.GetScaledImageAsThumbnailAsync(ThumbnailMode mode, uint requestedSize)` 中的参数 `requestedSize` 的值。

3. 播放mp3文件时，暂停后再播放，封面图旋转变慢；会在某一刻突然旋转了几十度，才继续平滑旋转。

分析：对于 `storyboard`，`Begin()` 与 `Stop()` 对应，`Resume()` 与 `Pause()` 对应。而在原来的代码中，在播放时直接调用了 `Begin()`，而非 `Resume()`。

我的理解是，暂停后再播放，若使用 `Begin()`，就会使动画把当前所停止的点作为一个新的起点（此时旋转角度为x°），产生一个由该起点到原有终点（旋转角度为360°）的新动画。但总用时不变（20秒），而一个周期中要旋转的角度变少了（从360°减少到(360 - x)°），看起来旋转就会变慢了；至于“在某一刻突然旋转了几十度，才继续平滑旋转”，是因为从一个周期的结束点（360°的旋转角）跳到了新周期的起点（x°的旋转角）。

解决：在 `MainPage` 中增加一个变量 `bool has_begun`，用于判断当前媒体是还未开始播放还是已经开始播放。在播放的时候，通过 `has_begun` 的值来判断对于动画是使用 `Begin()` 还是 `Resume()`。

4. 在同样的代码下，运行后打开的第一个媒体文件：

- mp3文件：有的可以播放；有的不可以播放，但是暂停再播放后就正常播放了。
- 视频文件：都可以正常播放。

分析：以上是初步观察结果。对同一个mp3文件继续进行测试，发现如果打开后立即点击播放按钮，则不可以播放，即便已调用 `MediaElement.Play()`；而如果稍等一下再点击播放按钮，则可以播放。再对视频文件进行相同的操作，也有类似的现象。

使用断点调试，发现在立即点击播放时，先调用 `MediaElement.Play()`，再进入 `MediaElement.MediaOpened` 事件（在 `MediaElement` 加载完成时执行），之后 `MediaElement.Position` 一直为0，进度条也因此一直停滞在开头；在稍等一下、还未点击播放按钮时，调用了 `MediaElement.MediaOpened` 事件，之后再点击播放按钮，就可以正常播放了。

因此，推断这个奇怪的bug是源于 `MediaElement` 的加载的时滞，如果打开媒体文件后过早点击播放按钮，这时 `MediaElement` 还未加载完毕，就会无法正常播放。

解决：需要保证 `MediaElement.Play()` 在 `MediaElement.MediaOpened` 完成之后才会启用。

在 `MainPage` 中增加了一个变量 `bool had_loaded` ，指示当前是否有已经加载的某个媒体文件。只有在自定义的 `MediaElement.MediaOpened` 事件中，它才会置为 `true` 。在打开新的媒体文件时，它会置为 `false` 。在播放媒体对应的方法中，要先判断 `had_loaded` 的值再决定是否播放。如果 `MediaElement` 没有加载完毕，点击播放按钮不会有反应。

六、思考与总结

知识点：

- 学到了如何使用HttpWebRequest来访问API并获取数据。
- 学到了如何解析Json格式和Xml格式的数据。
- 学到了如何使用MediaElement、Slider和Button来实现一个简单的播放器。
- 值得一提的是，原来AppBarButton的Flyout可以像一个xaml文件一样来使用。利用这个控件添加了TextBlock和Slider，实现了音量的调节。
- 学到了如何定义和使用一个简单的动画。

其他感想：

- 上面所报告的播放器bug是我遇到过的最神奇的bug之一，只因为点击的快慢这种很容易忽视的细节，就可以在相同的代码下产生不同的运行结果，这让我进一步见识到了编程和调试的复杂性。
- 以上的两个播放器bug【五-播放器-3/4】，让我了解到要对程序执行的流程有一个充分的认识，错误的执行流程可能导致不同的结果。
- IDE的调试工具很好用，可以使用调试工具来观察局部变量的值，以及观察方法执行的程序。