现代操作系统应用开发实验报告

姓名: 徐伟元 学号: 16340261 Blog : My Blog

实验名称 : Lab3 Query & MediaPlayer

参考资料

- newtonsoft-json
- CSDN-xml
- Microsoft Documents-MediaElement
- MSDN-composite

基础实验步骤

Query

1. 使用 HttpClient 访问网络
2. 提供城市**天气查询**(json Phrase)

MediaPlayer

1. 利用自定义控件控制视频的播放，暂停，**快进，快退**
2. 实现视频播放的全屏与退出全屏
3. 自制 slider 实现视频的进度条

关键代码与截图

Query

1. 使用 HttpClient 访问网络

   下面参考 Microsoft UWP Document 提供了利用 HttpClient 通过 网址 url 进行网络访问的代码模板。我们要做的事就三个：

   1. 给函数提供访问的 url 所需的参数；
   2. 替换将要访问的 url
   3. 完成下面的 http response处理代码

   函数的返回值可以自行修改。

```
private async Task<string> WeatherApi(/*params*/) {
    HttpClient httpClient = new HttpClient();
    var headers = httpClient.DefaultRequestHeaders;

    // The safe way to add a header value is to use the TryParseAdd method
    // and verify the return value is true,
    // especially if the header value is coming from user input.
    string header = "ie";
    if (!headers.UserAgent.TryParseAdd(header)) {
        throw new Exception("Invalid header value: " + header);
    }

    header = "Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0)";
    if (!headers.UserAgent.TryParseAdd(header)) {
        throw new Exception("Invalid header value: " + header);
    }

    string uri = "url";
    Uri requestUri = new Uri(uri);
    //Send the GET request asynchronously and retrieve the response as a string.
    HttpResponseMessage httpResponse = new HttpResponseMessage();
    string httpResponseBody = "";

    try {
        //Send the GET request
        httpResponse = await httpClient.GetAsync(requestUri);
        httpResponse.EnsureSuccessStatusCode();
        // Set encoding to 'UTF-8'
        Byte[] getByte1 = await httpResponse.Content.ReadAsByteArrayAsync();
        Encoding code1 = Encoding.GetEncoding("UTF-8");
        string result1 = code1.GetString(getByte1, 0, getByte1.Length);
        // Pharse the json data
        JsonReader reader = new JsonTextReader(new StringReader(result1));
        while (reader.Read()) {
            // Do something here with the response data
        }
    } catch (Exception ex) {
        httpResponseBody = "Error: " + ex.HResult.ToString("X") + " Message: " + ex.Message;
    }
    return /*string*/;
}
```

2. 提供城市天气查询(json Phrase)

   下面给出有关于 json phrase 的代码。

   这里调用的是**中国气象局**的api，作为一个年迈的网站，它的查询 url 不支持城市中文名，而应使用*城市代码*。感谢blouc的贡献，我直接将它洗成 xml 来辅助使用了，具体见代码文件内的 weather.xml 文件。

   对于 json 数据的解析，使用 newton 库提供的 json 解析库。

```
/* Accomplish the url */
// Get the city code for the api
string code = "101010100";
XmlDocument cityCode = new XmlDocument();
```

```
cityCode.LoadXml(System.IO.File.ReadAllText("Weather.xml"));
foreach (IXmlNode node in cityCode.GetElementsByTagName("City")) {
    if ((string)node.Attributes[0].NodeValue == city) {
        code = (string)node.Attributes[2].NodeValue;
    }
}
string uri = "http://www.weather.com.cn/data/sk/" + code + ".html";

/* Json Phrase Code for httpResponse data */

//Send the GET request
httpResponse = await httpClient.GetAsync(requestUri);
httpResponse.EnsureSuccessStatusCode();
// Set encoding to 'UTF-8'
Byte[] getByte1 = await httpResponse.Content.ReadAsByteArrayAsync();
Encoding code1 = Encoding.GetEncoding("UTF-8");
string result1 = code1.GetString(getByte1, 0, getByte1.Length);
// Pharse the json data
JsonReader reader = new JsonTextReader(new StringReader(result1));
while (reader.Read()) {
    if ((String)reader.Value == "city") {
        weather += "城市: ";
        reader.Read();
        if (reader.Value != null)
            weather += reader.Value + "\n";
    } else if ((String)reader.Value == "temp") {
        weather += "温度: ";
        reader.Read();
        if (reader.Value != null)
            weather += reader.Value + "℃\n";
    } else if ((String)reader.Value == "WD") {
        weather += "风向: ";
        reader.Read();
        if (reader.Value != null)
            weather += reader.Value + "\n";
    } else if ((String)reader.Value == "WS") {
        weather += "风速: ";
        reader.Read();
        if (reader.Value != null)
            weather += reader.Value + "\n";
    } else if ((String)reader.Value == "SD") {
        weather += "相对湿度: ";
        reader.Read();
        if (reader.Value != null)
            weather += reader.Value + "\n";
        } else if ((String)reader.Value == "time") {
            weather += "时间: ";
            reader.Read();
            if (reader.Value != null)
                weather += reader.Value + "\n";
        }
    }
```

MediaPlayer

强烈推荐Microsoft Documents-MediaElement。这是第一次在官方文档找到如此棒的一个指引文档。
关于 MediaElement 的 xml 代码，可以仿照上面的文档进行设置，下面主要讲讲具体实现的基础功能按钮的代码。

1. 利用自定义控件控制视频的播放，暂停，快进，快退
   函数中的 `Cover` 相关代码，为播放音乐旋转封面的相关代码，基础部分无需实现。

```
// Play media
private void MdeiaPlay(object sender, RoutedEventArgs e) {
    // Reset the play rate to normal
    if (myMediaPlayer.DefaultPlaybackRate != 1) {
```

```
            myMediaPlayer.DefaultPlaybackRate = 1.0;
        }
        myMediaPlayer.Play();
        if (Cover.Visibility == Visibility.Visible) {
            RotateCover.Begin();
        }
    }

    // Pause media
    private void MediaPause(object sender, RoutedEventArgs e) {
        myMediaPlayer.Pause();
        if (Cover.Visibility == Visibility.Visible) {
            RotateCover.Pause();
        }
    }

    // Stop media
    private void MediaStop(object sender, RoutedEventArgs e) {
        myMediaPlayer.Stop();
        if (Cover.Visibility == Visibility.Visible) {
            RotateCover.Stop();
        }
    }

    // Set back playing rate
    private void MediaBack(object sender, RoutedEventArgs e) {
        myMediaPlayer.DefaultPlaybackRate = -2.0;
        myMediaPlayer.Play();
    }

    // Set forward playing rate
    private void MediaForward(object sender, RoutedEventArgs e) {
        myMediaPlayer.DefaultPlaybackRate = 2.0;
        myMediaPlayer.Play();
    }
```

2. 实现视频播放的全屏与退出全屏

   这里的全屏为应用全屏。

   为了实现退出全屏，需要存储全屏之前的 size，所以需要一个 previousSize 来存储。

   全屏退出采取按键响应。对应在 keyUp 函数内调用。

```
    // Set full screen or not
    private void FullScreenToggle() {
        this.IsFullScreen = !this.IsFullScreen;
        if (this.IsFullScreen) {
            TransportControlsPanel.Visibility = Visibility.Collapsed;
            previousSize.Width = videoContainer.ActualWidth;
            previousSize.Height = videoContainer.ActualHeight;

            videoContainer.Width = Window.Current.Bounds.Width;
            videoContainer.Height = Window.Current.Bounds.Height;
            myMediaPlayer.Width = Window.Current.Bounds.Width;
            myMediaPlayer.Height = Window.Current.Bounds.Height;
        } else {
            TransportControlsPanel.Visibility = Visibility.Visible;

            videoContainer.Width = previousSize.Width;
            videoContainer.Height = previousSize.Height;
            myMediaPlayer.Width = previousSize.Width;
            myMediaPlayer.Height = previousSize.Height;
        }
    }

    //Start full screen
    private void MediaFullScreen(object sender, RoutedEventArgs e) {
```

```
        FullScreenToggle();
    }

    // Listen for [ESC] to exit full screen
    if (IsFullScreen && e.Key == Windows.System.VirtualKey.Escape) {
        FullScreenToggle();
    }
```

3. 自制 slider 实现视频的进度条
   在完整实现了官方文档之后，我很是怀疑，这个进度条，真的需要这么复杂吗？
   先来讲讲官方文档的实现逻辑：

   - 实现一个 slider 控件作为进度条
   - 实现 slider 控件的拖动分辨率
   - 实现一个 timer 跟随 mediaElement 一起播放计时和暂停
   - 实现一个 pointerPressed handler 将 mediaElement 的 position 值和 slider 值通过 timer 进行绑定
   - 每次新载入一个 media 文件，重置 timer
   - 初始化整个 APP 时，为 slider 注册 handler 事件
     其中 slider 拖动分辨率可以调整，这里我将其进一步精化了，为了适应 media 文件的总时长。
     代码太长了，而且逻辑层叠太过复杂，具体实现看代码文件，下面附上函数名。
     关于对这个冗杂的实现的改进，在遇到的问题模块提出解决方案。

```
    // Get the frequency of the steps on the slider's scale
    private double SliderFrequency(TimeSpan time) {}

    // Initialize the timer
    private void SetupTimer() {}

    // Keep timer in sync with media
    private void TimerTick(object sender, object e) {}

    // Start timer to count
    private void StartTimer() {}

    // Stop timer to count
    private void StopTimer() {}

    // slider pressed
    void SliderPointerEntered(object sender, PointerRoutedEventArgs e) {}

    // slider lost pressed
    void SliderPointerCaptureLost(object sender, PointerRoutedEventArgs e) {}

    void TimelineSliderValueChanged(object sender, RangeBaseValueChangedEventArgs e) {}

    // Reset timer for slider
    private void MyMediaOpened(object sender, RoutedEventArgs e) {}

    // Listen for the media state and timer
    private void MyMediaCurrentStateChanged(object sender, RoutedEventArgs e) {}
```

亮点与改进

Query

1. 提供 IP 地址查询(xml Phrase)
   调用网络服务的代码与天气查询的是一致的，只是更换了 url 网址与下面的 httpResponse 处理代码。
   这里返回了 xml 数据，我们利用官方提供的 `XmlDocument` 库进行解析。

```
    Byte[] getByte = await httpResponse.Content.ReadAsByteArrayAsync();
    Encoding code = Encoding.GetEncoding("UTF-8");
    string result = code.GetString(getByte, 0, getByte.Length);
```

```
// Pharse the xml data
XmlDocument document = new XmlDocument();
document.LoadXml(result);

var province = document.GetElementsByTagName("province");
location += "省份: " + province[0].InnerText + "\n";

var city = document.GetElementsByTagName("city");
location += "城市: " + city[0].InnerText + "\n";

var rectangle = document.GetElementsByTagName("rectangle");
var temp = rectangle[0].InnerText.Split(';');
location += "区域:\n\t" + temp[0] + "\n\t" + temp[1] + "\n";
```

MediaPlayer

1. **本地选择**多媒体文件(视频，音乐)进行播放
   选择多媒体文件并不困难，使用 FileOpenPicker 就好了。
   需要注意的是，当读取的是音乐文件时，需要隐藏视频页面，显示 cover 页面(有点面包屑的意味，不需要跳转页面，省去开销)。

```
// Select media to play
private async void MediaSelect(object sender, RoutedEventArgs e) {
    FileOpenPicker picker = new FileOpenPicker();
    /// Initialize the picture file type to take
    picker.FileTypeFilter.Add(".mp4");
    picker.FileTypeFilter.Add(".wmv");
    picker.FileTypeFilter.Add(".wma");
    picker.FileTypeFilter.Add(".mp3");
    picker.SuggestedStartLocation = PickerLocationId.PicturesLibrary;

    StorageFile file = await picker.PickSingleFileAsync();

    if (file != null) {
        /// Load the selected picture
        IRandomAccessStream ir = await file.OpenAsync(FileAccessMode.Read);
        myMediaPlayer.SetSource(ir, file.ContentType);
        myMediaPlayer.Play();
        if (file.ContentType == "audio/mpeg") {
            Cover.Visibility = Visibility.Visible;
        } else {
            Cover.Visibility = Visibility.Collapsed;
        }
    }
}
```

2. 添加 slider 版本**音量控制**
   理解数据绑定就很容易了，就是 slider 值和 mediaElement 的音量值绑定。
   Attention! 音量值为0 - 1，分割值为0.1，这是 mediaElement 官方类中给出的说明。所以，设置一下 slider 的值是必须的。(在这里被坑了很久，以为实现不了 slider 音量控制)

```
<Slider x:Name="volumeslider" Maximum="1" Minimum="0" StepFrequency="0.1"
        Value="{x:Bind myMediaPlayer.Volume, Mode=TwoWay}">
</Slider>
```

   其中，UI 页面提供了音量按钮，所以，点击按钮直接静音，然后需要储存静音前的音量。逻辑与全屏类似，不赘述，实现代码见代码文件。

3. 实现**封面旋转**
   - 播放视频时，隐藏封面页面(在读取文件中已实现)
   - 播放音乐时，封面开始旋转
   - 暂停音乐时，封面旋转暂停

- 重置音乐或结束时，封面复位
  xml 中实现动画代码。

```
<Ellipse x:Name="Cover" RenderTransformOrigin="0.5,0.5">
    <Ellipse.RenderTransform>
        <CompositeTransform></CompositeTransform>
    </Ellipse.RenderTransform>
    <Ellipse.Resources>
        <Storyboard x:Name="RotateCover" RepeatBehavior="Forever">
            <DoubleAnimation Duration="0:0:20" To="360" Storyboard.TargetName="Cover"
                             Storyboard.TargetProperty="(UIElement.RenderTransform).(CompositeTransform.Rota
            </DoubleAnimation>
        </Storyboard>
    </Ellipse.Resources>
    <Ellipse.Fill>
        <ImageBrush x:Name="picture" ImageSource="Assets/cover.jpg"></ImageBrush>
    </Ellipse.Fill>
</Ellipse>
```

在需要动画开始，暂停，复位的地方，调用其对应函数即可。如，音乐暂停时，动画暂停:

```
// Pause media
private void MediaPause(object sender, RoutedEventArgs e) {
    myMediaPlayer.Pause();
    if (Cover.Visibility == Visibility.Visible) {
        RotateCover.Pause();
    }
}
```

4. 添加**键盘按键响应**
   作为一个多媒体播放应用，空格键播放暂停，上下方向按键调节音量是基础功能，所以添加了这几个基础按键响应。
   mediaElement 提供了 keyUp 和 keyDown 两个接口，前者响应一次按键，后者响应持续按键。

```
// Keyboard listener for one press
private void VideoContainerKeyUp(object sender, KeyRoutedEventArgs e) {
    // Listen for [Space] to stop or play media
    if (e.Key == Windows.System.VirtualKey.Space) {
        if (myMediaPlayer.CurrentState == MediaElementState.Playing) {
            myMediaPlayer.Pause();
        } else {
            myMediaPlayer.Play();
        }
    }

    // Listen for [Up] and [Down] to control volume
    if (e.Key == Windows.System.VirtualKey.Up) {
        if (myMediaPlayer.Volume < 1) {
            myMediaPlayer.Volume += 0.1;
        }
    } else if (e.Key == Windows.System.VirtualKey.Down) {
        if (myMediaPlayer.Volume > 0) {
            myMediaPlayer.Volume -= 0.1;
        }
    }

    e.Handled = true;
}

// Keyboard listener for holding press
private void VideoContainerKeyDown(object sender, KeyRoutedEventArgs e) {
    // Listen for [Up] and [Down] to control volume
    if (e.Key == Windows.System.VirtualKey.Up) {
        if (myMediaPlayer.Volume < 1) {
            myMediaPlayer.Volume += 0.1;
        }
```

```
        } else if (e.Key == Windows.System.VirtualKey.Down) {
            if (myMediaPlayer.Volume > 0) {
                myMediaPlayer.Volume -= 0.1;
            }
        }
    }
```

遇到的问题

xml 解析

神奇的事情，使用 `System.Xml` 进行解析，VS经常报错说无法找到类或无法实例化等奇怪的报错，无奈之下，换用 `XmlDocument`

slider 实现进度条

是的，像前面说的，官网文档提供的实现方法，太过复杂。有什么改进方法吗?
考虑值绑定。我们将 slider 的值和 mediaElement 的 position 值绑定在一起，就实现了这个进度条。
类似之前我们实现的 MyList 中 checkbox 和 line 的绑定，我们写一个 converter 将 position 这个 TimeSpan 类型的值 转换为 double 类型，就可以与 slider 值绑定了。

思考与总结

这两次作业总体来说，比之前要简单，因为资料会多一些。不过，依旧有一些需要自己解决的问题，所以对于开发过程中遇到的问题，有感于依赖官网和技术博客，因此，个人的感悟和心路历程以博客的形式记录了下来，起备忘和给他人引用来使用。报告也就从博客上应运而生了。建议前往博客看看，报告使用markdown在线网站编写，附图不便，博客中有贴图。