TRUNG TÂM TIN HỌC
Trường Đại học Khoa Học Tự Nhiên - Đại học quốc gia TP.HCM
Công GMC.blogspot.com MediaFire Blogger

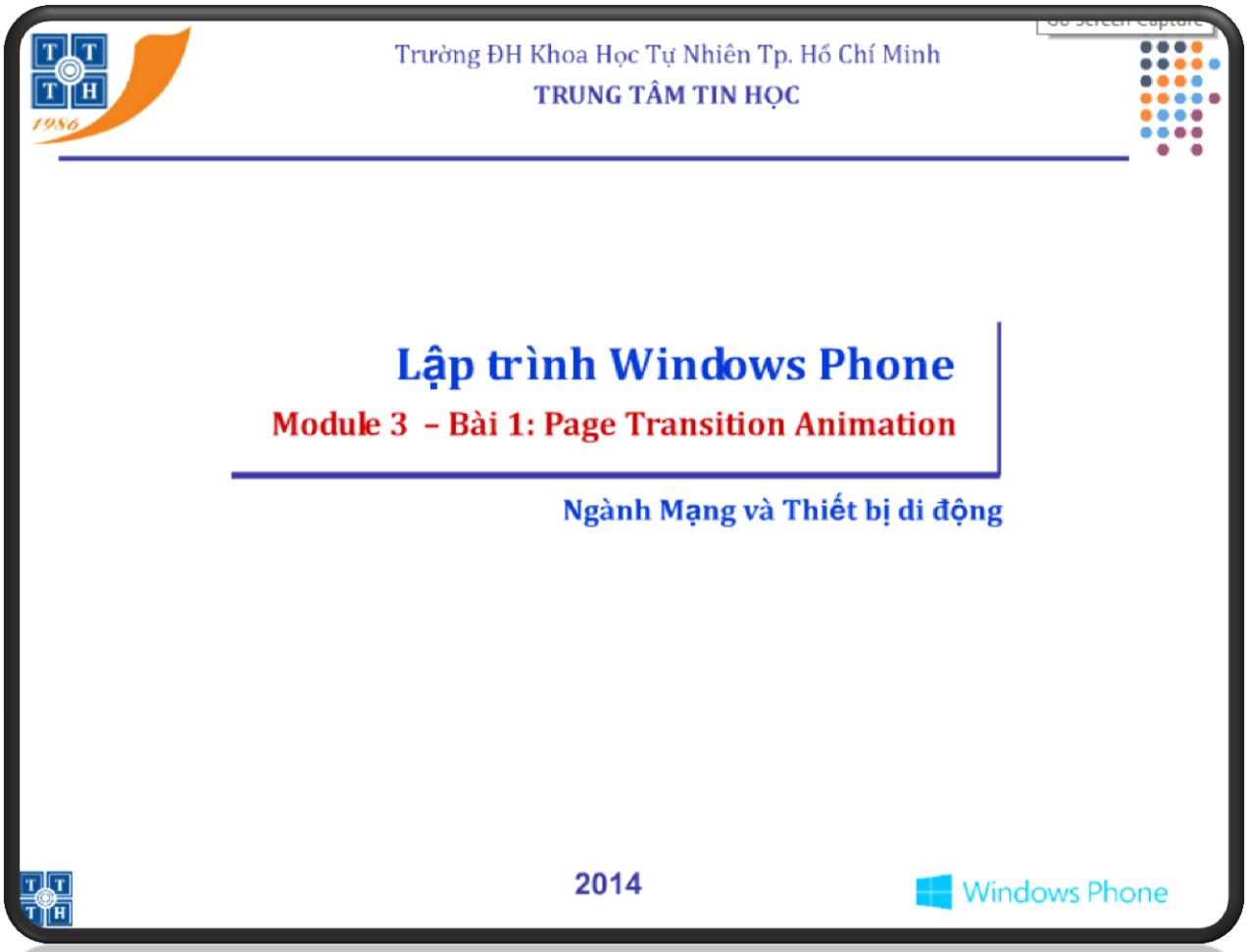
LẬP TRÌNH WINDOW PHONE

Module 3

*Click vào phụ lục để chuyển tới **bài cần đọc***

Phụ lục

Công GMC.blogspot.com
MediaFire Blogger


Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC
Lập trình Windows Phone
Module 3 – Bài 1: Page Transition Animation
Ngành Mạng và Thiết bị di động
2014
Windows Phone

Nội dung

1. Giới thiệu hiệu ứng chuyển trang
2. Cách sử dụng Windows Phone Toolkit
3. Các loại Page transition animation
4. Hiệu ứng cho các UIElement

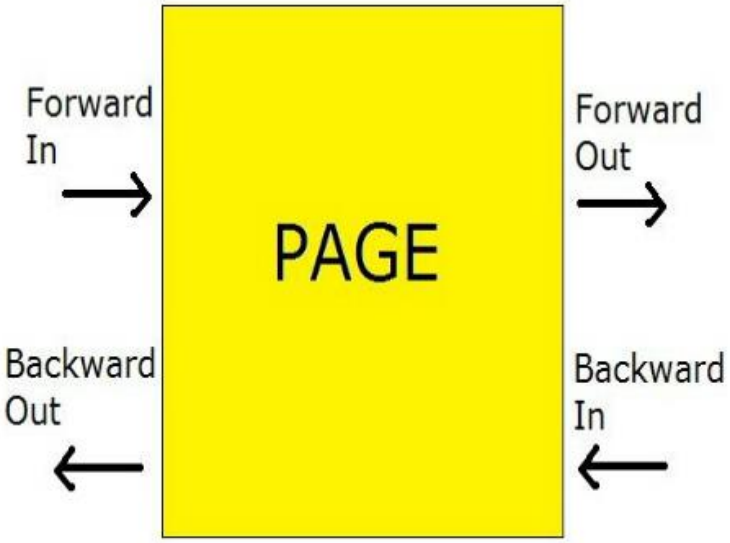
1. Giới thiệu hiệu ứng chuyển trang

Microsoft đã hỗ trợ hiệu ứng chuyển trang bắt mắt trong gói Microsoft Windows Phone Toolkit, một số hiệu ứng có thể kể đến là:

- ✓ Turnstile transition
- ✓ Swivel transition
- ✓ Slide transition
- ✓ Roll transition
- ✓ Rotate transition

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

1. Giới thiệu hiệu ứng chuyển trang



2. Cách sử dụng Windows Phone Toolkit

Để sử dụng được Page transition animation, ta phải dùng thư viện:

Microsoft.Phone.Controls.Toolkit.dll

Có 2 cách để tải thư viện này về:

- ✓ Tải từ codeplex: http://silverlight.codeplex.com/releases/view/55034
- ✓ Tải từ NuGet Packages

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2. Cách sử dụng Windows Phone Toolkit

Ta cần tạo namespace để sử dụng bộ toolkit này:

```
<phone:PhoneApplicationPage
    x:Class="LearnTransition.Page2"
    xmlns:toolkit="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone.Controls.Toolkit"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
```

Với **toolkit** là một prefix, ta có thể đặt bất kỳ tên gì, tuy nhiên nên đặt tên mang tính gợi nhớ.

Thông qua **toolkit** ta có thể truy suất tới các lớp thư viện nằm bên trong.

2. Cách sử dụng Windows Phone Toolkit

Ví dụ để tạo hiệu ứng chuyển trang trong XAML:

```
<toolkit:TransitionService.NavigationInTransition>
  <toolkit:NavigationInTransition>
     <toolkit:NavigationInTransition.Forward>
        <toolkit:RollTransition></toolkit:RollTransition>
     </toolkit:NavigationInTransition.Forward>
  </toolkit:NavigationInTransition>
</toolkit:TransitionService.NavigationInTransition>
<toolkit:TransitionService.NavigationOutTransition>
  <toolkit:NavigationOutTransition>
     <toolkit:NavigationOutTransition.Backward>
        <toolkit:TurnstileTransition Mode="BackwardOut"/>
     </toolkit:NavigationOutTransition.Backward>
  </toolkit:NavigationOutTransition>
</toolkit:TransitionService.NavigationOutTransition>
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2. Cách sử dụng Windows Phone Toolkit

Sau cùng để có thể kích hoạt được hiệu ứng chuyển trang ta thay đổi instance RootFrame như sau:

Trong hàm **InitializePhoneApplication**() của lớp **App** (App.xaml.cs):

Code cũ:

```
RootFrame = new PhoneApplicationFrame();
```

Đổi lại thành:

```
RootFrame = new TransitionFrame();
```


3. Các loại Page transition animation

Gói Windows Phone Toolkit hỗ trợ các hiệu ứng:

a. **Turnstile** transition
b. **Swivel** transition
c. **Slide** transition
d. **Roll** transition
e. **Rotate** transition

Ta sẽ lần lượt giới thiệu sơ lược từng loại, chi tiết học học xem trong giáo trình.

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

a. Turnstile transition

Hiệu ứng này cho phép quay trang giống như quay cửa sổ từ một góc, đây là một trong những hiệu ứng phổ biến và sử dụng đơn giản nhất. Hiệu ứng này có 4 kiểu:

- ✓ TurnstileTransitionMode. **ForwardIn**
- ✓ TurnstileTransitionMode. **ForwardOut**
- ✓ TurnstileTransitionMode. **BackwardIn**
- ✓ TurnstileTransitionMode. **BackwardOut**

a. Turnstile transition

BackwardIn – BackwardOut

ForwardIn - ForwardOut

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

a. Turnstile transition

```
<toolkit:TransitionService.NavigationInTransition>
    <toolkit:NavigationInTransition>
      <toolkit:NavigationInTransition.Backward>
        <toolkit:TurnstileTransition Mode="BackwardIn"/>
      </toolkit:NavigationInTransition.Backward>
      <toolkit:NavigationInTransition.Forward>
        <toolkit:TurnstileTransition Mode="ForwardIn"/>
      </toolkit:NavigationInTransition.Forward>
    </toolkit:NavigationInTransition>
  </toolkit:TransitionService.NavigationInTransition>
  <toolkit:TransitionService.NavigationOutTransition>
    <toolkit:NavigationOutTransition>
      <toolkit:NavigationOutTransition.Backward>
        <toolkit:TurnstileTransition Mode="BackwardOut"/>
      </toolkit:NavigationOutTransition.Backward>
      <toolkit:NavigationOutTransition.Forward>
        <toolkit:TurnstileTransition Mode="ForwardOut"/>
      </toolkit:NavigationOutTransition.Forward>
    </toolkit:NavigationOutTransition>

  </toolkit:TransitionService.NavigationOutTransition>
```

XAML code example

Lập trình Windows Phone – Page Transition Animation 12

a. Turnstile transition

```
protected override void OnNavigatedTo(NavigationEventArgs e)
    {
      base.OnNavigatedTo(e);
      TurnstileTransition turnstileTransition =
            new TurnstileTransition ();
      turnstileTransition.Mode =
            TurnstileTransitionMode .ForwardIn;
      ITransition transition =
            turnstileTransition.GetTransition( this);
      transition.Begin();
      transition.Completed += delegate
            { transition.Stop(); };
    }
```

Coding behind example

Lập trình Windows Phone – Page Transition Animation 13

b. Swivel transition

Hiệu ứng xoay (Swivel) tương tự như hiệu ứng quay cửa sổ (Turnstile), trong trường hợp này nó sẽ lấy trục quay là nằm ngang giữa trang (Height/2). Hiệu ứng này có các kiểu sau:

- ✓ SwivelTransitionMode.BackwardIn
- ✓ SwivelTransitionMode.BackwardOut
- ✓ SwivelTransitionMode.ForwardIn
- ✓ SwivelTransitionMode.ForwardOut
- ✓ SwivelTransitionMode.FullScreenIn
- ✓ SwivelTransitionMode.FullScreenOut

b. Swivel transition

BackwardIn - BackwardOut

ForwardIn - ForwardOut

FullScreenIn - FullScreenOut

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

b. Swivel transition

```
<toolkit:TransitionService.NavigationInTransition>
    <toolkit:NavigationInTransition>
      <toolkit:NavigationInTransition.Backward>
        <toolkit:SwivelTransition Mode="BackwardIn"/>
      </toolkit:NavigationInTransition.Backward>
      <toolkit:NavigationInTransition.Forward>
        <toolkit:SwivelTransition Mode="ForwardIn"/>
      </toolkit:NavigationInTransition.Forward>
    </toolkit:NavigationInTransition>
</toolkit:TransitionService.NavigationInTransition>
<toolkit:TransitionService.NavigationOutTransition>
    <toolkit:NavigationOutTransition>
      <toolkit:NavigationOutTransition.Backward>
        <toolkit:SwivelTransition Mode="BackwardOut"/>
      </toolkit:NavigationOutTransition.Backward>
      <toolkit:NavigationOutTransition.Forward>
        <toolkit:SwivelTransition Mode="ForwardOut"/>
      </toolkit:NavigationOutTransition.Forward>
    </toolkit:NavigationOutTransition>

</toolkit:TransitionService.NavigationOutTransition>
```

XAML code example

b. Swivel transition

```
protected override void OnNavigatedTo(NavigationEventArgs e)
    {
        base.OnNavigatedTo(e);
        SwivelTransition swivelTransition =
                new SwivelTransition();
        swivelTransition.Mode =
                SwivelTransitionMode.FullScreenIn;
        ITransition transition =
                swivelTransition.GetTransition(this);
        transition.Begin();
        transition.Completed += delegate
                { transition.Stop(); };

    }
```

Coding behind example

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

c. Slide transition

Hiệu ứng trượt trang cho phép trang di chuyển theo một số hướng nào đó: Từ dưới lên, từ trái qua, từ phải qua...Cụ thể ta có các kiểu hiểu ứng sau:

- ✓ **SlideTransitionMode** .SlideUpFadeIn
- ✓ **SlideTransitionMode** .SlideUpFadeOut
- ✓ **SlideTransitionMode** .SlideDownFadeIn
- ✓ **SlideTransitionMode** .SlideDownFadeOut
- ✓ **SlideTransitionMode** .SlideLeftFadeIn
- ✓ **SlideTransitionMode** .SlideLeftFadeOut
- ✓ **SlideTransitionMode** .SlideRightFadeIn
- ✓ **SlideTransitionMode** .SlideRightFadeOut

c. Slide transition

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

c. Slide transition

```
<toolkit:TransitionService.NavigationInTransition>
    <toolkit:NavigationInTransition>
      <toolkit:NavigationInTransition.Backward>
        <toolkit:SlideTransition  Mode="SlideDownFadeIn"/>
      </toolkit:NavigationInTransition.Backward>
      <toolkit:NavigationInTransition.Forward>
        <toolkit:SlideTransition  Mode="SlideRightFadeIn"/>
      </toolkit:NavigationInTransition.Forward>
    </toolkit:NavigationInTransition>
  </toolkit:TransitionService.NavigationInTransition>
  <toolkit:TransitionService.NavigationOutTransition>
    <toolkit:NavigationOutTransition>
      <toolkit:NavigationOutTransition.Backward>
        <toolkit:SlideTransition  Mode="SlideLeftFadeOut"/>
      </toolkit:NavigationOutTransition.Backward>
      <toolkit:NavigationOutTransition.Forward>
        <toolkit:SlideTransition  Mode="SlideUpFadeOut"/>
      </toolkit:NavigationOutTransition.Forward>
    </toolkit:NavigationOutTransition>

</toolkit:TransitionService.NavigationOutTransition>
```

XAML code example

Lập trình Windows Phone – Page Transition Animation 20

c. Slide transition

```
protected override void OnNavigatedTo(NavigationEventArgs e)
    {
      base.OnNavigatedTo(e);

      SlideTransition  slideTransition =
            new SlideTransition ();
      slideTransition.Mode =
            SlideTransitionMode .SlideRightFadeIn ;
      ITransition transition =
            slideTransition.GetTransition( this);
      transition.Begin();
      transition.Completed +=  delegate
            { transition.Stop(); };

    }
```

Coding behind example

Lập trình Windows Phone – Page Transition Animation 21

d. Roll transition

Hiệu ứng này dùng để quay tr òn trang, đây là hiệu ứng không có kiểu Mode nên cách sử dụng cũng khá đơn giản.

d. Roll transition

```
<toolkit:TransitionService.NavigationInTransition>
    <toolkit:NavigationInTransition>
      <toolkit:NavigationInTransition.Forward>
        <toolkit:RollTransition></toolkit:RollTransition>
      </toolkit:NavigationInTransition.Forward>
    </toolkit:NavigationInTransition>
</toolkit:TransitionService.NavigationInTransition>
<toolkit:TransitionService.NavigationOutTransition>
    <toolkit:NavigationOutTransition>
      <toolkit:NavigationOutTransition.Backward>
        <toolkit:RollTransition></toolkit:RollTransition>
      </toolkit:NavigationOutTransition.Backward>
    </toolkit:NavigationOutTransition>

</toolkit:TransitionService.NavigationOutTransition>
```

XAML code example

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

d. Roll transition

```
protected override void OnNavigatedTo(NavigationEventArgs e)
    {
        base.OnNavigatedTo(e);
        RollTransition rollTransition =
                new RollTransition();
        ITransition transition =
                rollTransition.GetTransition(this);
        transition.Begin();
        transition.Completed += delegate
                { transition.Stop(); };

    }
```

Coding behind example

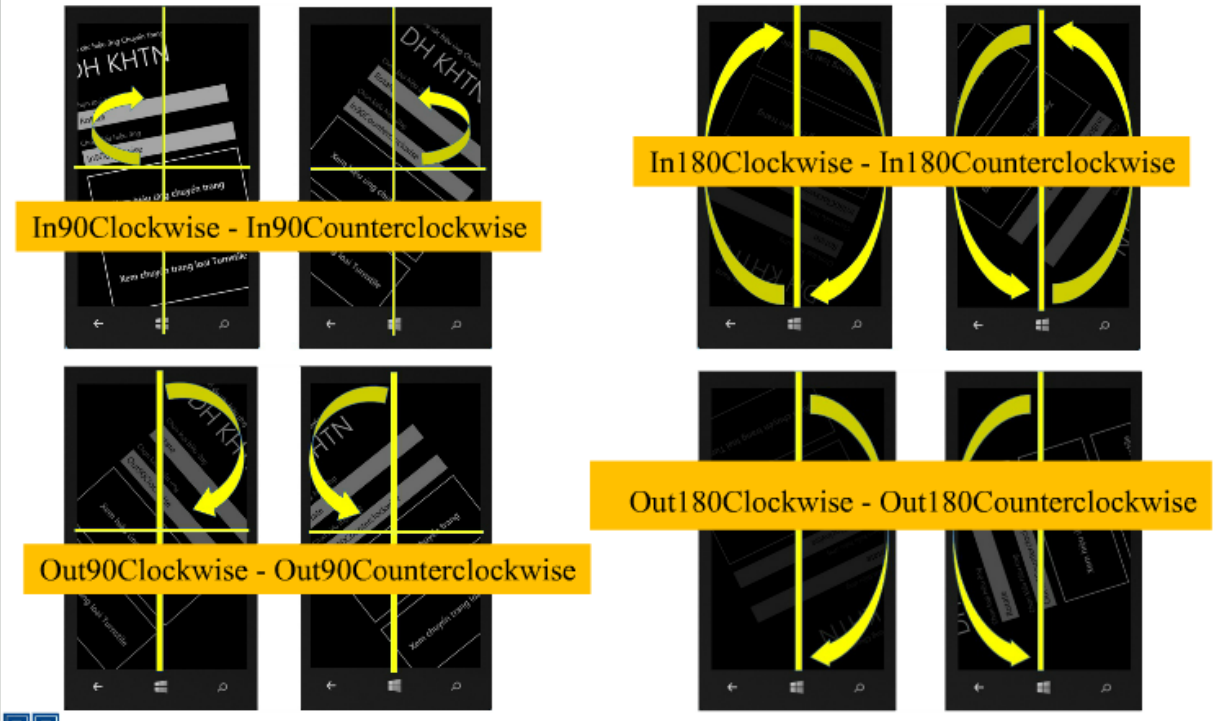
e. Rotate transition

Rotate chuyển trang quay theo một số góc độ cố định (có thể nói nó tương tự như Slide Transition):

- ✓ **RotateTransitionMode** .In90Clockwise
- ✓ **RotateTransitionMode** .In90Counterclockwise
- ✓ **RotateTransitionMode** .In180Clockwise
- ✓ **RotateTransitionMode** .In180Counterclockwise
- ✓ **RotateTransitionMode** .Out90Clockwise
- ✓ **RotateTransitionMode** .Out90Counterclockwise
- ✓ **RotateTransitionMode** .Out180Clockwise
- ✓ **RotateTransitionMode** .Out180Counterclockwise

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

e. Rotate transition



e. Rotate transition

```
<toolkit:TransitionService.NavigationInTransition>
    <toolkit:NavigationInTransition>
      <toolkit:NavigationInTransition.Backward>
        <toolkit:RotateTransition Mode="In90Clockwise" />
      </toolkit:NavigationInTransition.Backward>
      <toolkit:NavigationInTransition.Forward>
        <toolkit:RotateTransition Mode="In180Clockwise" />
      </toolkit:NavigationInTransition.Forward>
    </toolkit:NavigationInTransition>
  </toolkit:TransitionService.NavigationInTransition>
  <toolkit:TransitionService.NavigationOutTransition>
    <toolkit:NavigationOutTransition>
      <toolkit:NavigationOutTransition.Backward>
        <toolkit:RotateTransition Mode="Out180Counterclockwise" />
      </toolkit:NavigationOutTransition.Backward>
      <toolkit:NavigationOutTransition.Forward>
        <toolkit:RotateTransition Mode="Out90Counterclockwise" />
      </toolkit:NavigationOutTransition.Forward>
    </toolkit:NavigationOutTransition>
  </toolkit:TransitionService.NavigationOutTransition>
```

XAML code example

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

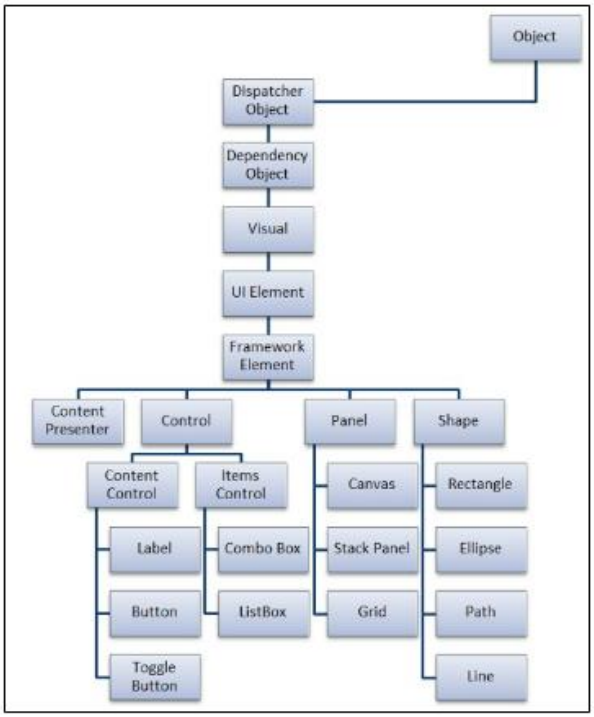
e. Rotate transition

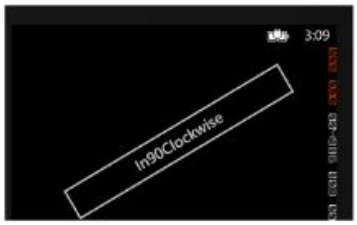
```
protected override void OnNavigatedTo(NavigationEventArgs e)
    {
      base.OnNavigatedTo(e);

      RotateTransition rotateTransition =
            new RotateTransition();
      rotateTransition.Mode =
            RotateTransitionMode.In180Clockwise;
      ITransition transition =
            rotateTransition.GetTransition(this);
      transition.Begin();
      transition.Completed += delegate
            { transition.Stop(); };

    }
```

Coding behind example

Lập trình Windows Phone – Page Transition Animation 28

4. Hiệu ứng cho các UIElement





Lập trình Windows Phone – Page Transition Animation 29

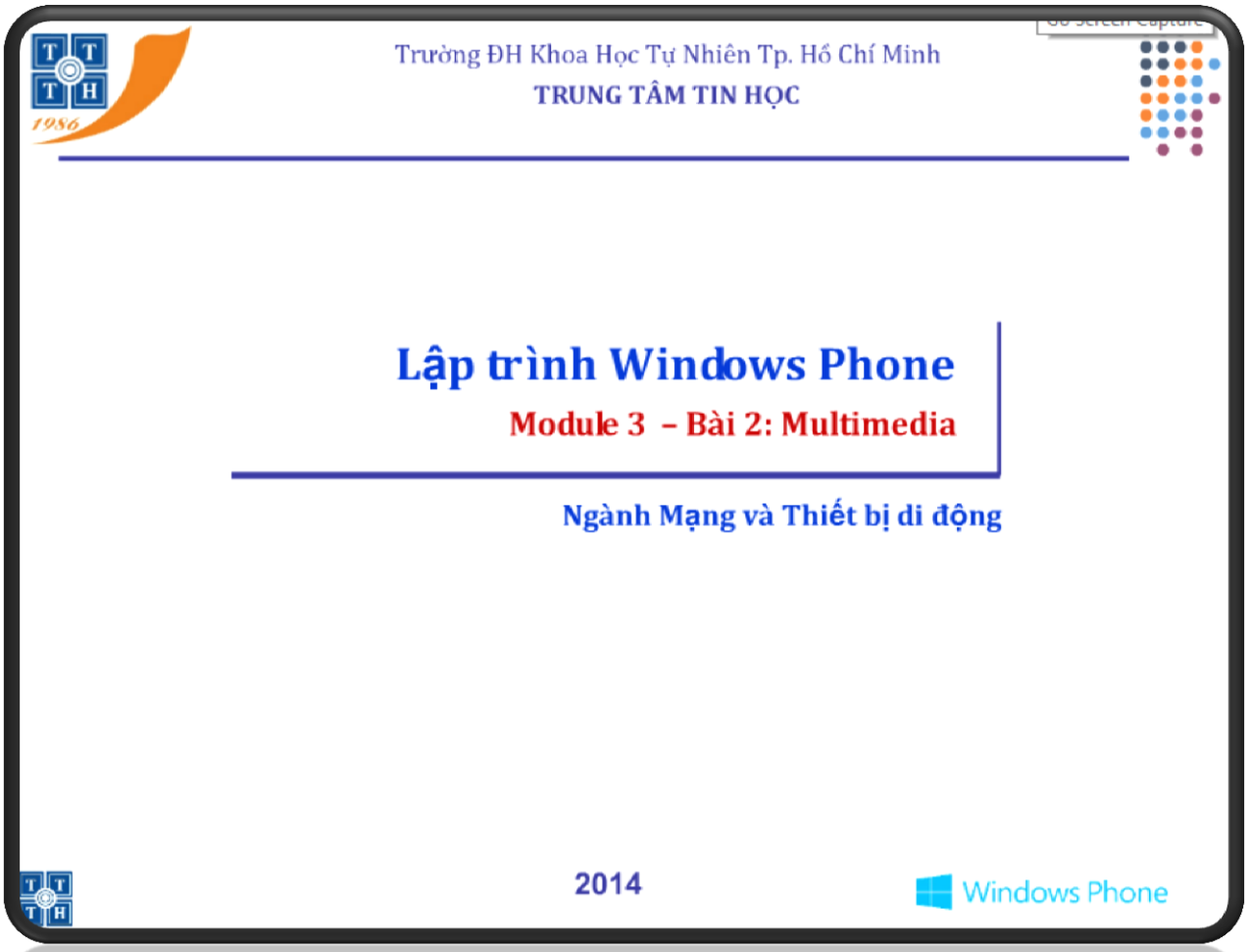
4. Hiệu ứng cho các UIElement

Theo như kiến trúc ở trên thì về cơ bản các Control (Button, Label, TextBox, TextBlock, Listbox...) các Panel (Canvas, StackPanel, Grid...) đều được dẫn xuất từ UIElement, do đó có thể nói rằng các đối tượng này là UIElement.

```
private void btnRotation _Click(object sender, RoutedEventArgs e)  {
        RotateTransition  rotate = new RotateTransition ();
        rotate.Mode =  RotateTransitionMode .In90Clockwise;
        ITransition transition =
                rotate.GetTransition(btnRotation );
        transition.Begin ();
        transition.Completed +=  delegate
                { transition.Stop(); };
    }
```


Thảo luận

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC


Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC
Lập trình Windows Phone
Module 3 – Bài 2: Multimedia
Ngành Mạng và Thiết bị di động
2014
Windows Phone

Nội dung

1. Giới thiệu Multimedia
2. Cách sử dụng Camera & Photo
3. Làm việc với Video
4. Làm việc với Audio

1. Giới thiệu Multimedia

- ✓ Windows Phone hỗ trợ đa phương tiện rất mạnh mẽ, lập trình viên có thể tương tác được với Camera, Pho, Video và Audio.
- ✓ Trong giới hạn của phần này người học sẽ được cung cấp các kiến thức nền tảng và nâng cao về cách tương tác tới Camera: Cách Preview, cách chụp, cách lưu trữ hình ảnh; Video: Cách play, pause, stop, mute, seek, ghi âm; Audio: Cách đưa Sound File vào ứng dụng cũng như cách thức lập trình cho Audio chạy background.
- ✓ Học viên sẽ được củng cố them kiến thức về MediaElement, MediaPlayerLauncher …

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2. Cách sử dụng Camera & Photo

- Chế độ Camera Preview
- Chụp hình từ Camera
- Sử dụng Camera với ViewModel
- Thiết lập hiệu ứng âm thanh cho Camera
- Tạo Camera Focus

2.1 Chế độ Camera Preview

Chế độ Camera Preview cho phép chúng ta xem trước khung cảnh, để có thể chụp được một bức ảnh thì ta phải khởi tạo PhotoCamera và cung cấp **ID_CAP_ISV_CAMERA** trong WMAppManifest.xml Capabilities.

- Sử dụng 2 lớp thư viện dưới đây:
  - ✓ PhotoCamera
  - ✓ VideoBrush

Hai lớp trên nằm trong:

```
using Microsoft.Devices;
using System.Windows.Media;
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2.1 Chế độ Camera Preview

```
private PhotoCamera _camera;
private VideoBrush _videoBrush;
protected override void
OnNavigatedTo(NavigationEventArgs e)
{
   base.OnNavigatedTo(e);
   _camera = new PhotoCamera(CameraType.Primary);
   _camera.Initialized += _camera_Initialized;
   _videoBrush = new VideoBrush();
   _videoBrush.SetSource(_camera);
}
```


2.1 Chế độ Camera Preview

```
void _camera_Initialized(object sender,
CameraOperationCompletedEventArgs e)      {
if (e.Succeeded) //kiểm tra kích hoạt thành công
 {
  var resolution = new Size(2048, 1536);
 if (_camera.AvailableResolutions.Contains(resolution))
            {
               _camera.Resolution = resolution;
            }
   Dispatcher.BeginInvoke(() => {
          this.LayoutRoot.Background = _videoBrush;
       });
 }
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2.2. Chụp hình từ Camera

Để có thể chụp hình được từ Camera ta cần **ID_CAP_MEDIALIB_PHOTO** capability và thêm 2 sự kiện sau vào quá trình khởi tạo:

- **Sự kiện Người sử dụng bấm nút chụp:**

```
CameraButtons.ShutterKeyPressed +=
        CameraButtons_ShutterKeyPressed;
```

- **Khi hình ảnh sẵn sàng được chụp:**

```
_camera.CaptureImageAvailable +=
        _camera_CaptureImageAvailable;
```


2.2. Chụp hình từ Camera

```
void CameraButtons_ShutterKeyPressed
      (object sender, EventArgs e) {
    try {
        _camera.CaptureImage();
      }
      catch (InvalidOperationException ex)
      {
        this.Dispatcher.BeginInvoke(() =>
        {
          MessageBox.Show(ex.Message);
        });
      }
  }
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2.2. Chụp hình từ Camera

```
void _camera_CaptureImageAvailable
        (object sender, ContentReadyEventArgs e)
    {
        string path = "picture_drthanh.jpg";
        MediaLibrary library = new MediaLibrary();
        library.SavePictureToCameraRoll
                (path, e.ImageStream);
    }
```


2.3. Sử dụng Camera với ViewModel

Để giảm thiểu thời gian coding, người ta sử dụng Camera ViewModel và binding tới trang ứng dụng.

Các bước chính cần làm:

1. Tạo lớp **BaseViewModel** kế thừa từ **INotifyPropertyChanged**.
2. Tạo lớp **CameraViewModel** kế thừa từ **BaseViewModel** và **IDisposable**
3. Binding tới VideoBrush:

```
<Grid x:Name="LayoutRoot"
            Background="{Binding Preview}"/>
```

✓ Chi tiết xem giáo trình trang 6

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2.4. Thiết lập hiệu ứng âm thanh cho Camera

- Việc đưa âm thanh vào quá trình chụp hình trong Camera là khá quan trọng và hấp dẫn, mỗi khi người sử dụng bấm nút chụp thì chương trình sẽ phát ra những âm thanh để có thể biết được thao tác thành công hay thất bại.
- Ta sử dụng các thư viện sau:

```
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using System.Windows.Resources;
using System.Windows;
```


2.4. Thiết lập hiệu ứng âm thanh cho Camera

- Bước 1: Lấy Stream Resource sound file:

```
Uri uri=new Uri("Stop.wav", UriKind.Relative);
StreamResourceInfo resource =
        Application.GetResourceStream(uri);
```

- Bước 2: lấy đối tượng **SoundEffect:**

```
var effect = SoundEffect.FromStream(resource.Stream);
```

- Bước 3: Gọi hàm Update để gửi thông điệp tới XNA Framework:

```
FrameworkDispatcher.Update();
```

- Bước 4: Gọi phương thức Play để mở sound:

```
effect.Play();
```

✓ Chi tiết xem trang 11

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2.5. Tạo Camera Focus

- Thường thì Camera hỗ trợ chúng ta chức năng Focus vào một đối tượng hay một vùng nào đó trong quá trình chúng sử dụng.
- Ta sử dụng thuộc tính **IsFocusSupported**, **IsFocusAtPointSupported** của **PhotoCamera** để kiểm tra xem camera có hỗ trợ Focus hay không.
- Dùng phương thức **Focus()** và **FocusAtPoint()** để thiết lập focus. Khi quá trình Focus hoàn thành thì nó sẽ tự động xuất hiện sự kiện **AutoFocusCompleted**, dùng sự kiện này để ta xử lý nhiều kết quả.

2.5. Tạo Camera Focus

- Chú ý là ta phải dùng **Canvas** để có thể di chuyển được ô Focus tới vị trí bất kỳ trên màn hình điện thoại.

```
public void Focus()    {
 if ((_camera.IsFocusSupported) && (!_isCapturing))
      {
         Deployment.Current.Dispatcher.BeginInvoke(()
             => {
           this.IsFocusing = true;
         });
         _camera.Focus();
      }
   }
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2.5. Tạo Camera Focus

```
public void FocusAtPoint(double x, double y)
    {
      if ((_camera.IsFocusAtPointSupported) &&
            (!_isCapturing))       {
        Deployment.Current.Dispatcher.BeginInvoke(()
       => {
          this.IsFocusing = true;
        });
        _camera.FocusAtPoint(x, y);
      }
}
```

Chi tiết xem trang 16

3. Làm việc với Video

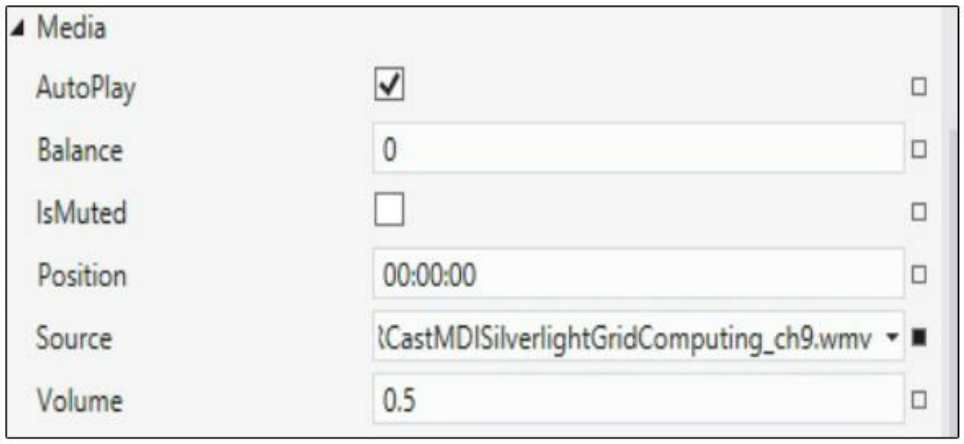
- MediaElement và MediaPlayerLauncher
- Các thao tác thường sử dụng:
  - ✓ Play
  - ✓ Stop
  - ✓ Pause
  - ✓ Mute
  - ✓ Seek
  - ✓ Record

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

3.1. MediaElement và MediaPlayerLauncher

- MediaElement

```
<MediaElement x:Name="mymedia"
              Source="film.wmv"/>
```



3.1. MediaElement và MediaPlayerLauncher

- MediaPlayerLauncher

```
MediaPlayerLauncher player =
                    new MediaPlayerLauncher();

player.Media = new Uri("film.wmv", UriKind.Relative);

player.Location = MediaLocationType.Install;

player.Show();
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

3.2. Các thao tác thường sử dụng

- ✓ Play
- ✓ Stop
- ✓ Pause
- ✓ Mute
- ✓ Seek
- ✓ Record

3.2. Các thao tác thường sử dụng

- ✓ **Play**

```
private void btnPlay_Click
                (object sender, RoutedEventArgs e)
    {
      if (this.CanPlay())
      {
        mediaPlayer.Play();
      }
    }
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

3.2. Các thao tác thường sử dụng

✓ **Stop**

```
private void btnStop_Click
            (object sender, RoutedEventArgs e)
    {
      mediaPlayer.Stop();
      mediaPlayer.Position =
     System.TimeSpan.FromSeconds(0);
      lblStatus.Text = "Stopped";
    }
```

3.2. Các thao tác thường sử dụng

✓ **Pause**

```
private void btnPause_Click
        (object sender, RoutedEventArgs e) {
        if (mediaPlayer.CanPause)
        {
           mediaPlayer.Pause();
           lblStatus.Text = "Paused";
        }
        else  {
           lblStatus.Text = "Can not be Paused!!";
        }
    }
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

3.2. Các thao tác thường sử dụng

✓ **Mute**

```
private void btnMute_Click
        (object sender, RoutedEventArgs e){
  if (lblSoundStatus.Text.Equals("Sound On",
      StringComparison.CurrentCultureIgnoreCase)) {
            lblSoundStatus.Text = "Sound Off";
            mediaPlayer.IsMuted = true;
   } else {
            lblSoundStatus.Text = "Sound On";
            mediaPlayer.IsMuted = false;}
```

3.2. Các thao tác thường sử dụng

✓ **Seek**

Thường ta dùng Slider để di chuyển

```
<Slider Height="84" HorizontalAlignment="Left"
Margin="13,423,0,0" Name="mediaTimeline"
VerticalAlignment="Top" Width="443"
ValueChanged="mediaTimeline_ValueChanged"
Maximum="1" LargeChange="0.1" />
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

3.2. Các thao tác thường sử dụng

✓ **Seek**

```
private void mediaTimeline_ValueChanged(object sender,
RoutedPropertyChangedEventArgs<double> e)  {
 if (mediaPlayer.CanSeek) {
   TimeSpan duration =
              mediaPlayer.NaturalDuration.TimeSpan;
    int newPosition = (int)(duration.TotalSeconds *
                        mediaTimeline.Value);
     mediaPlayer.Position =
          new TimeSpan(0, 0, newPosition);      }
}
```


3.2. Các thao tác thường sử dụng

✓ **Record**

Để quay Video ta c ần dùng các thư viện sau:

```
using System.Windows.Media;
using System.IO;
using System.IO.IsolatedStorage;
using Microsoft.Xna.Framework.Media;
```

Ta có 3 thao tác chính:

Preview – xem trước

Record – Quay

Playback – Phát lại

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

3.2. Các thao tác thường sử dụng

✓ **Preview**

```
private void Preview() {
        VideoPlayer.Stop();
        VideoPlayer.Source = null;
        _captureSource.Stop();
        _videoBrush.SetSource(_captureSource);
       recordvideo.Background = _videoBrush;
        _captureSource.Start();
 }
```

3.2. Các thao tác thường sử dụng

✓ **Record**

```
private void Record()
 {
     _captureSource.Stop();
     using (IsolatedStorageFile store =

IsolatedStorageFile.GetUserStoreForApplication())
     {
     store.CreateFile(path).Close();
     }
     _fileSink.CaptureSource = _captureSource;
     _captureSource.Start();
 }
```

3.2. Các thao tác thường sử dụng

✓ **Playback**

```
private void Playback() {
        _captureSource.Stop();
         IsolatedStorageFile store
=IsolatedStorageFile.GetUserStoreForApplication();
         if (store.FileExists(path)) {
          IsolatedStorageFileStream isoStream = new
IsolatedStorageFileStream (path,FileMode.Open, FileAccess.Read, store);
              VideoPlayer.SetSource(isoStream);
              VideoPlayer.MediaEnded += new
RoutedEventHandler(VideoPlayer_MediaEnded);
              VideoPlayer.Play();
          }
      }
```

4. Làm việc với Audio

- Thêm âm thanh vào ứng dụng
- Cách chạy Audio Background

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

4.1. Thêm âm thanh vào ứng dụng

- Giống như việc đưa Video vào ứng dụng

Ta cũng sử dụng các thư viện sau:

```
using Microsoft.Xna.Framework.Media;
using Microsoft.Xna.Framework;
```

4.1. Thêm âm thanh vào ứng dụng

```
private void btnPlay_Click
            (object sender, RoutedEventArgs e)
 {
      if (CanPlay())        {
         mymedia.Stop();
         mymedia.Source = new
System.Uri("sound.wma", System.UriKind.Relative);
         mymedia.Play();
      }
 }
```

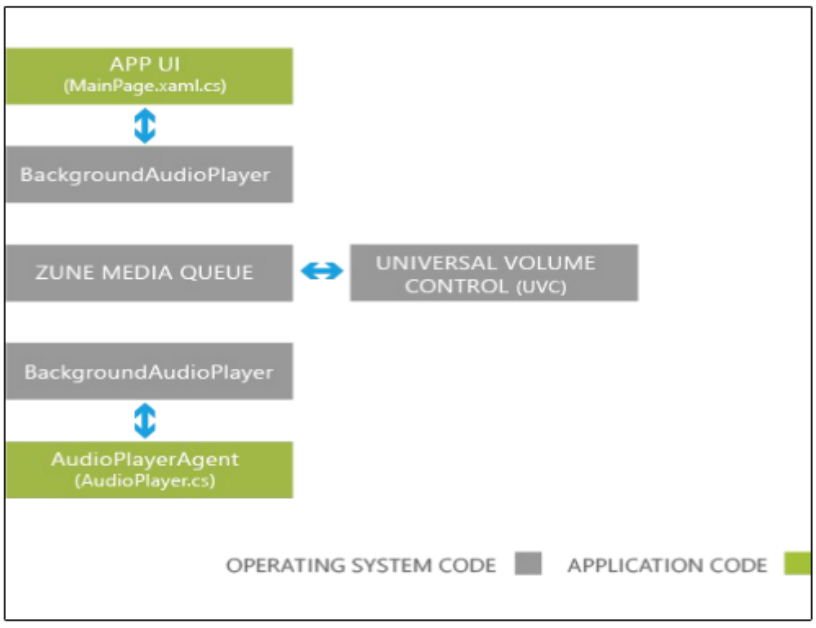
Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

4.2. Cách chạy Audio Background

- Có 2 cách chạy nền:
- ✓ **AudioPlayerAgent,**
- ✓ **AudioStreamingAgent**

4.2. Cách chạy Audio Background

- ✓ **AudioPlayerAgent**



Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

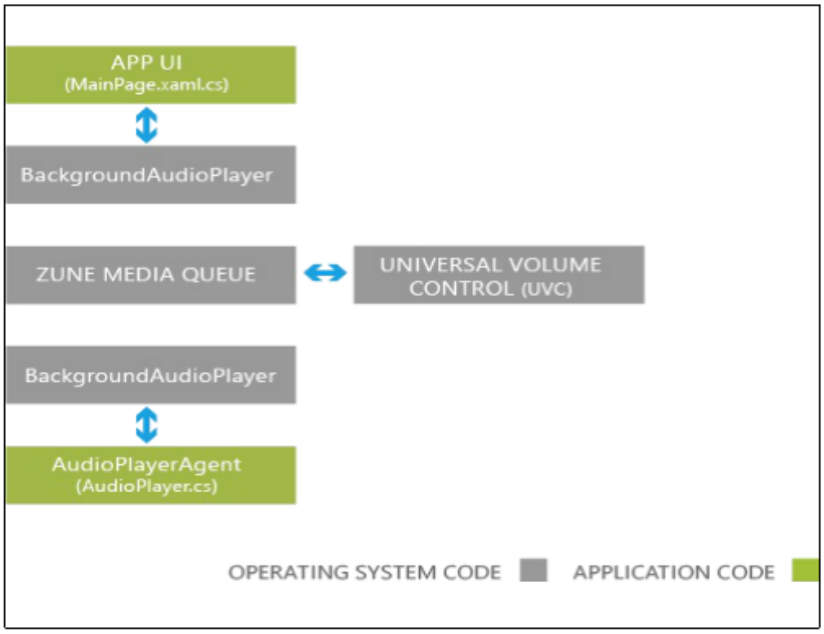
4.2. Cách chạy Audio Background

✓ **AudioPlayerAgent**

1. **OnPlayStateChanged** : Hàm này để tự động kiểm tra sự thay đổi trạng thái play, bao gồm: buffering, playing, track ready, track ended, v à shutdown.
2. **OnUserAction**: Hàm này sẽ được triệu gọi khi người sử dụng tiến hành những thao tác nào đó trong Application's audio playlist ở vòng đời Foreground (dùng controls provider ho ặc Universal Volume Control)
3. **OnError**: Hàm này sẽ được triệu gọi khi có lỗi sảy ra trong quá trình audio playback.

4.2. Cách chạy Audio Background

✓ **AudioStreamingAgent**



Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

4.2. Cách chạy Audio Background

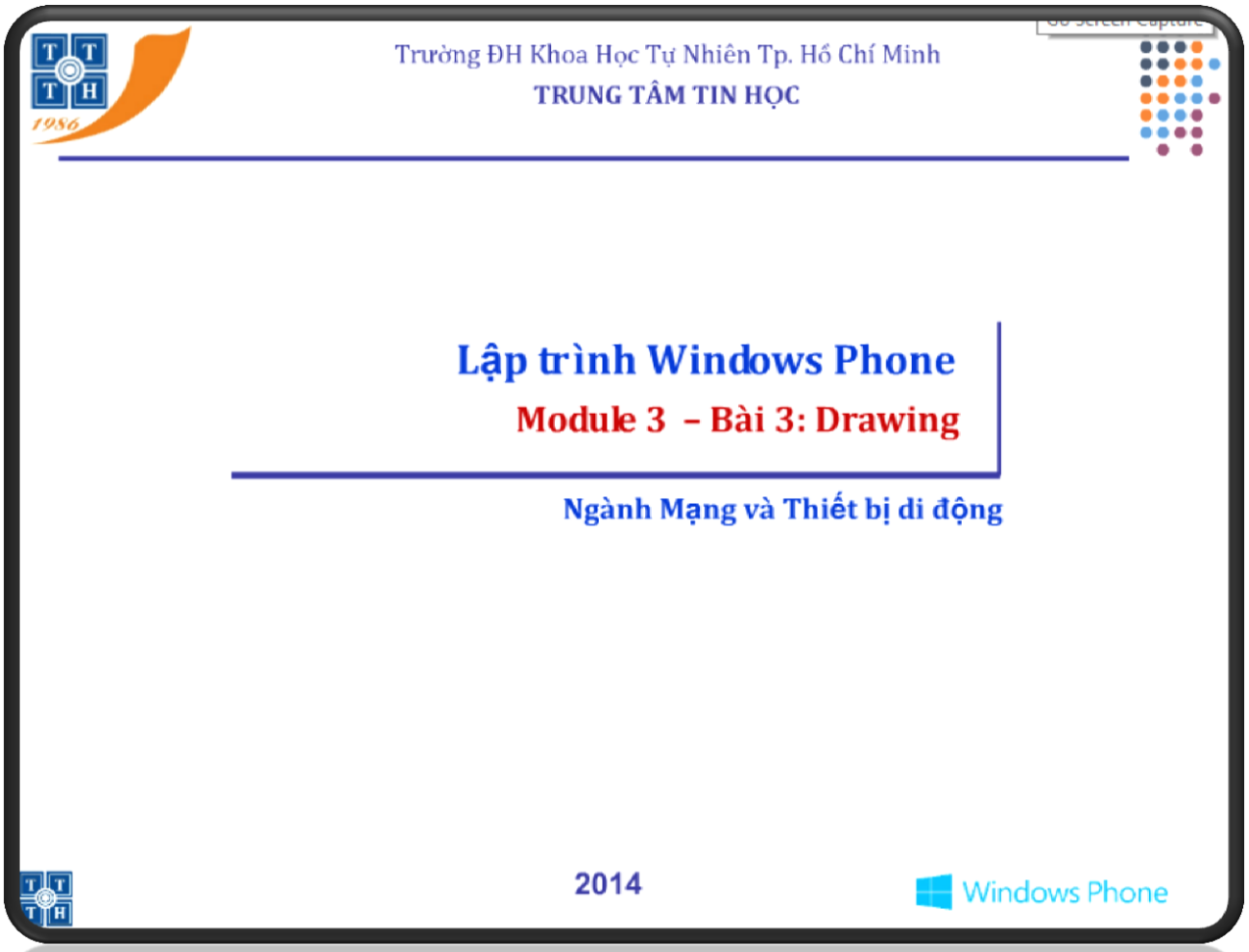
✓ **AudioStreamingAgent**

1. **OnBeginStreaming:** Hàm này sẽ được triệu gọi khi ta bắt đầu Audio Streaming
2. **OnCancel:** Hàm này sẽ được triệu gọi khi Audio Stream bị hủy.

Chi tiết cách chạy nền xem trang 31

Thảo luận

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

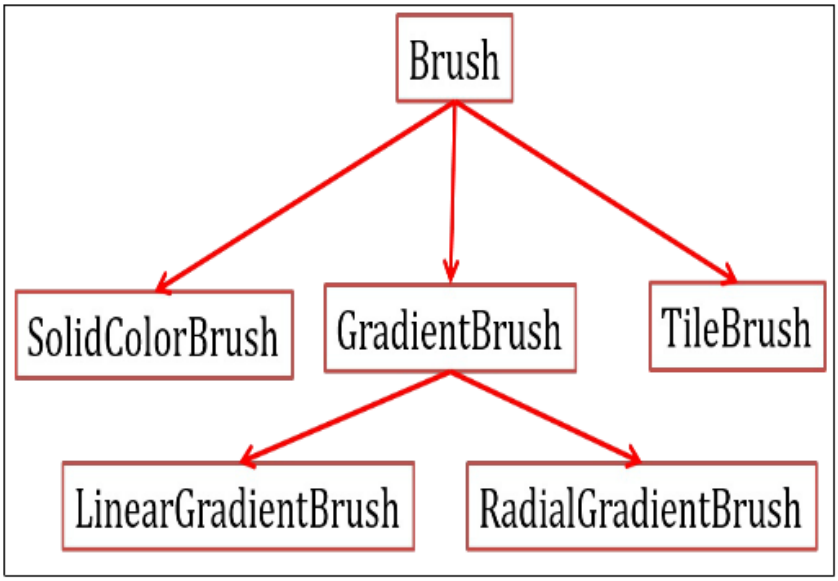

Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC
Lập trình Windows Phone
Module 3 – Bài 3: Drawing
Ngành Mạng và Thiết bị di động
2014
Windows Phone

Nội dung

1. Các đối tượng vẽ màu phổ biến (Brush)
2. Canvas
3. Các loại đối tượng Shapes

1. Các đối tượng vẽ màu phổ biến (Brush)

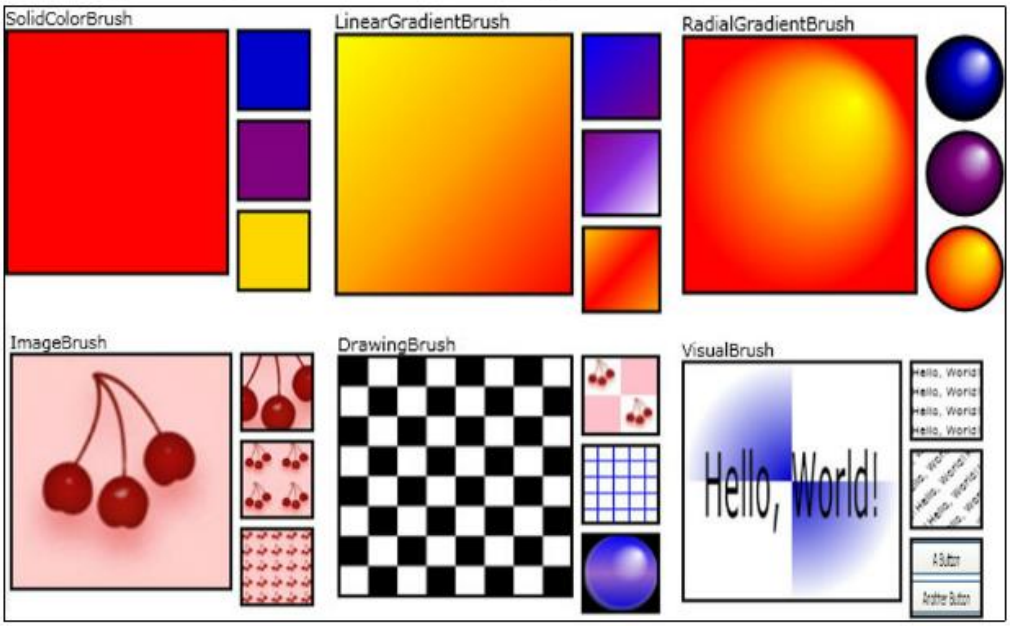
✓ Windows Phone hỗ trợ hàng loại các Brush:



Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

1. Các đối tượng vẽ màu phổ biến (Brush)

- ✓ Ví dụ bảng màu theo từng Brush:



1. Các đối tượng vẽ màu phổ biến (Brush)

- ✓ SolidColorBrush
- ✓ LinearGradientBrush
- ✓ RadialGradientBrush
- ✓ ImageBrush

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

1.1. SolidColorBrush

✓ Tô màu đồng nhất

Dùng XAML:

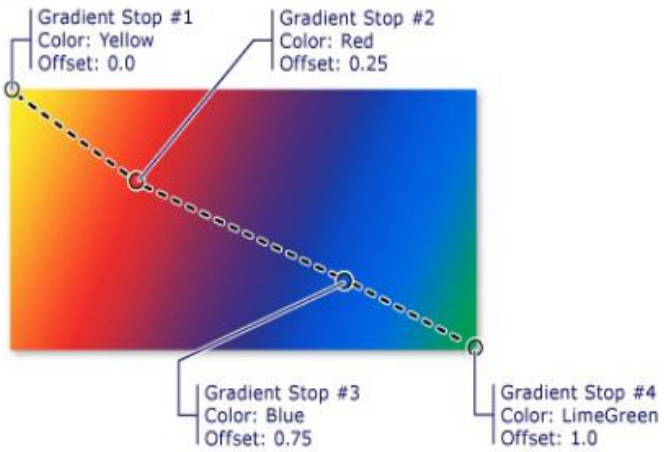
```
<Rectangle Width="50" Height="50" Fill="Blue" />
```

Dùng Coding behind:

```
Rectangle myPredefinedBrushRectangle =
        new Rectangle();
myPredefinedBrushRectangle.Width = 50;
myPredefinedBrushRectangle.Height = 50;
myPredefinedBrushRectangle.Fill = Brushes.Blue;
```


1.2. LinearGradientBrush

✓ Cho phép phối màu tuyến tính



Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

1.2. LinearGradientBrush

```
<Rectangle Width="200" Height="100">
  <Rectangle.Fill>
    <LinearGradientBrush
        StartPoint="0,0"
        EndPoint="1,1">
      <GradientStop Color="Yellow" Offset="0.0" />
      <GradientStop Color="Red" Offset="0.25" />
      <GradientStop Color="Blue" Offset="0.75" />
      <GradientStop Color="LimeGreen" Offset="1.0" />
    </LinearGradientBrush>
  </Rectangle.Fill>
</Rectangle>
```

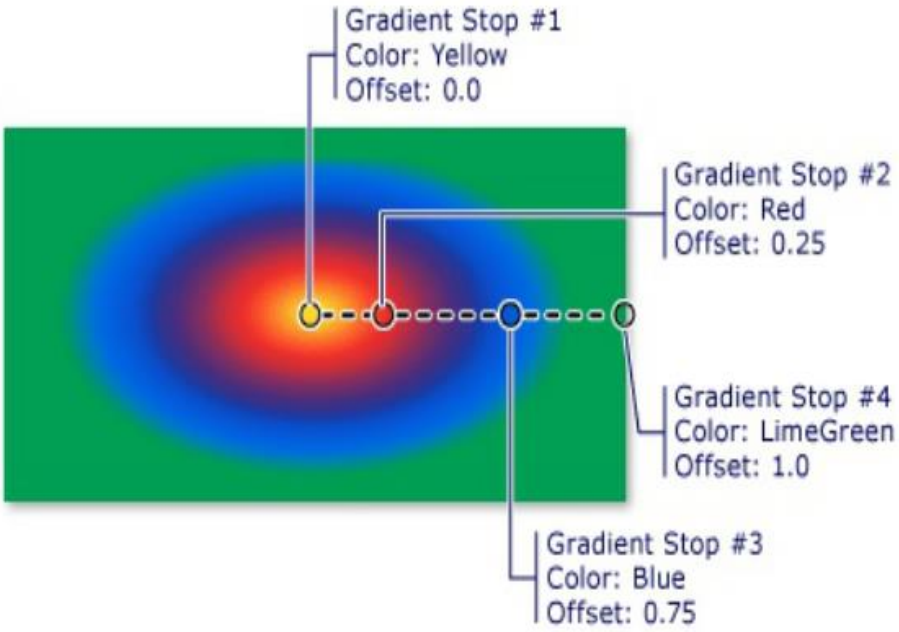

1.2. LinearGradientBrush

```
Rectangle diagonalFillRectangle = new Rectangle();
diagonalFillRectangle.Width = 200;
diagonalFillRectangle.Height = 100;
// Create a diagonal linear gradient with four stops.
LinearGradientBrush myLinearGradientBrush =
    new LinearGradientBrush();
myLinearGradientBrush.StartPoint = new Point(0,0);
myLinearGradientBrush.EndPoint = new Point(1,1);
myLinearGradientBrush.GradientStops.Add(
    new GradientStop(Colors.Yellow, 0.0));
myLinearGradientBrush.GradientStops.Add(
    new GradientStop(Colors.Red, 0.25));
myLinearGradientBrush.GradientStops.Add(
    new GradientStop(Colors.Blue, 0.75));
myLinearGradientBrush.GradientStops.Add(
    new GradientStop(Colors.LimeGreen, 1.0));
diagonalFillRectangle.Fill = myLinearGradientBrush;
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

1.3. RadialGradientBrush

✓ Phối màu theo hình cầu, có tiêu điểm



1.3. RadialGradientBrush

```
<Rectangle Width="200" Height="100">
  <Rectangle.Fill>
   <RadialGradientBrush
    GradientOrigin="0.5,0.5"
    Center="0.5,0.5" RadiusX="0.5" RadiusY="0.5">
    <RadialGradientBrush.GradientStops>
     <GradientStop Color="Yellow" Offset="0" />
     <GradientStop Color="Red" Offset="0.25" />
     <GradientStop Color="Blue" Offset="0.75" />
     <GradientStop Color="LimeGreen" Offset="1" />
    </RadialGradientBrush.GradientStops>
   </RadialGradientBrush>
  </Rectangle.Fill>
</Rectangle>
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

1.3. RadialGradientBrush

```
RadialGradientBrush radialGradient = new
RadialGradientBrush();

// Set the GradientOrigin to the center of the area being
painted.
radialGradient.GradientOrigin = new Point(0.5, 0.5);

// Set the gradient center to the center of the area being painted.
radialGradient.Center = new Point(0.5, 0.5);

// Set the radius of the gradient circle so that it extends to
// the edges of the area being painted.
radialGradient.RadiusX = 0.5;
radialGradient.RadiusY = 0.5;
```

1.3. RadialGradientBrush

```
// Create four gradient stops.
radialGradient.GradientStops.Add(new
GradientStop(Colors.Yellow, 0.0));
radialGradient.GradientStops.Add(new
GradientStop(Colors.Red, 0.25));
radialGradient.GradientStops.Add(new
GradientStop(Colors.Blue, 0.75));
radialGradient.GradientStops.Add(new
GradientStop(Colors.LimeGreen, 1.0));

// Freeze the brush (make it unmodifiable) for performance
benefits.
radialGradient.Freeze();
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

1.3. RadialGradientBrush

```
// Create a rectangle and paint it with the
// RadialGradientBrush.
Rectangle aRectangle = new Rectangle();
aRectangle.Width = 200;
aRectangle.Height = 100;
aRectangle.Fill = radialGradient;
```


1.4. ImageBrush

- ✓ Brush cho phép hiển thị hình ảnh

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

1.4. ImageBrush

```
<Button
    Foreground="White" FontWeight="Bold"
    FontSize="16pt" FontFamily="Verdana"
    Content="Berries"
    Padding="20"
    HorizontalAlignment="Left">
     <Button.Background>
        <ImageBrush
        Viewport="0,0,0.5,0.5"
        TileMode="FlipXY"
        ImageSource="sampleImages\berries.jpg" />
     </Button.Background>
</Button>
```


2. Canvas

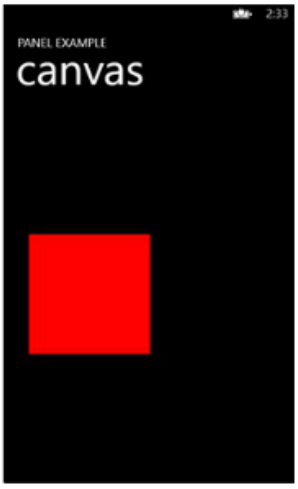
- ✓ **Đặc tính của Canvas**
- ✓ **Kết xuất hình ảnh từ Canvas**

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2.1. Đặc tính của Canvas

Canvas là một loại panel cho phép ta định vị các đối tượng trong nó theo tọa độ tùy thích.

```
<Canvas Background="Transparent">
   <Rectangle
         Canvas.Left="30"
         Canvas.Top="200"
         Fill="red"
         Width="200"
         Height="200" />
</Canvas>
```



2.1 Đặc tính của Canvas

```
myCanvas1 = new Canvas();
myCanvas1.Background = Brushes.Red;
myCanvas1.Height = 200;
myCanvas1.Width = 200;
Canvas.SetTop(myCanvas1, 200);
Canvas.SetLeft(myCanvas1, 30);
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2.2. Kết xuất hình ảnh từ Canvas

Canvas cho ta kết xuất thành định dạng hình ảnh và cho phép lưu vào điện thoại, để sử dụng được tính năng này thì ta phải:

```
using Microsoft.Xna.Framework.Media;
using System.Windows.Media.Imaging;
using System.IO;
using System.Windows.Media;
```

2.2. Kết xuất hình ảnh từ Canvas

```
private void Save_Click(object sender,
RoutedEventArgs e)
    {
        MediaLibrary library = new MediaLibrary();
        WriteableBitmap bitMap = new
            WriteableBitmap(drawCanvas, null);
        MemoryStream ms = new MemoryStream();
        Extensions.SaveJpeg(bitMap, ms,
  bitMap.PixelWidth,bitMap.PixelHeight, 0, 100);
        ms.Seek(0, SeekOrigin.Begin);
library.SavePicture(string.Format("Images\\{0}.jpg",
                        Guid.NewGuid()), ms);
    }
```

3. Các loại đối tượng Shapes

- ✓ Ellipse
- ✓ Rectangle
- ✓ Line
- ✓ Polyline
- ✓ Polygon
- ✓ Path

3.1. Ellipse

- ✓ Để tạo Ellipse ta làm như sau:

```
<Ellipse Fill="Blue"
    Height="300"
    Width="300"/>
```


```
Ellipse ellipse = new Ellipse();
ellipse.Fill = new SolidColorBrush(Colors.Blue);
ellipse.Width = 300;
ellipse.Height = 300;
mycanvas.Children.Add(ellipse);
```

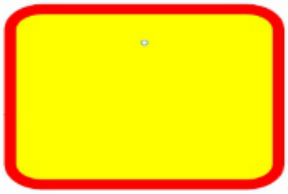
Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

3.2. Rectangle

✓ Rectangle cho phép vẽ khung hình chữ nhật với các góc cạnh, bo góc, màu đường viền và độ dày của đường viền, màu nền:

```
Rectangle rect = new Rectangle();
rect.Fill=new
        SolidColorBrush(Colors.Yellow);
rect.Width=350;
rect.Height=275;
rect.Stroke=new
        SolidColorBrush(Colors.Red);
rect.StrokeThickness=15;
rect.RadiusX=40.0;
rect.RadiusY=30.0;
```

```
<Rectangle Fill="Yellow"
  Height="275"
  Width="350"
  Stroke="Red"
  StrokeThickness="15"
  RadiusX="40"
  RadiusY="30"/>
```



3.3. Line

✓ Line cho phép vẽ một đường thẳng theo tọa độ X1,Y1 nối với X2,Y2:

```
<Line Fill="Yellow"
  Stroke="Red"
  StrokeThickness="5"
  X1="0"  Y1="10"
  X2="100"  Y2="200"
  />
```

```
Line line = new Line();
line.Fill=new
SolidColorBrush(Colors.Yellow);
line.Stroke=new
SolidColorBrush(Colors.Red);
line.StrokeThickness=5;
line.X1=0.0;
line.Y1=10.0;
line.X2=100.0;
line.Y2=200.0;
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

3.4. Polyline

✓ Polyline cho phép vẽ các đường Line liên tiếp nhau nhưng không tạo thành một vòng khép kín.

```
<Canvas>
 <Polyline Points="50,25 0,100 100,100 50,25"
  Stroke="Red" StrokeThickness="10"
  Fill="Yellow"
  Canvas.Left="75" Canvas.Top="50" />
</Canvas>
```

3.4. Polyline

```
Polyline polyline=new Polyline();
polyline.Points.Add(new Point(50.0,25.0));
polyline.Points.Add(new Point(0.0,100.0));
polyline.Points.Add(new Point(100.0,100.0));
polyline.Points.Add(new Point(50.0,25.0));
polyline.Stroke=new SolidColorBrush(Colors.Red);
polyline.StrokeThickness=10.0;
polyline.Fill = new SolidColorBrush(Colors.Yellow);
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

3.5. Polygon

Polyline cho phép vẽ các đường Line liên tiếp nhau và tạo thành một vòng khép kín.

```
<Polygon
 Points="50,25 0,100 100,100 50,25"
  Stroke="Red"
  StrokeThickness="10"
  Fill="Yellow"
  Canvas.Left="75"
  Canvas.Top="50" />
```

3.5. Polygon

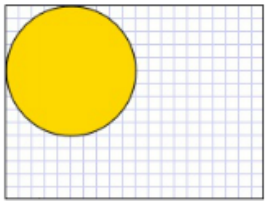
```
Polygon polygon = new Polygon();
polygon.Points.Add(new Point(50.0, 25.0));
polygon.Points.Add(new Point(0.0, 100.0));
polygon.Points.Add(new Point(100.0, 100.0));
polygon.Points.Add(new Point(50.0, 25.0));
polygon.Stroke = new SolidColorBrush(Colors.Red);
polygon.StrokeThickness = 10.0;
polygon.Fill = new SolidColorBrush(Colors.Yellow);
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

3.6. Path

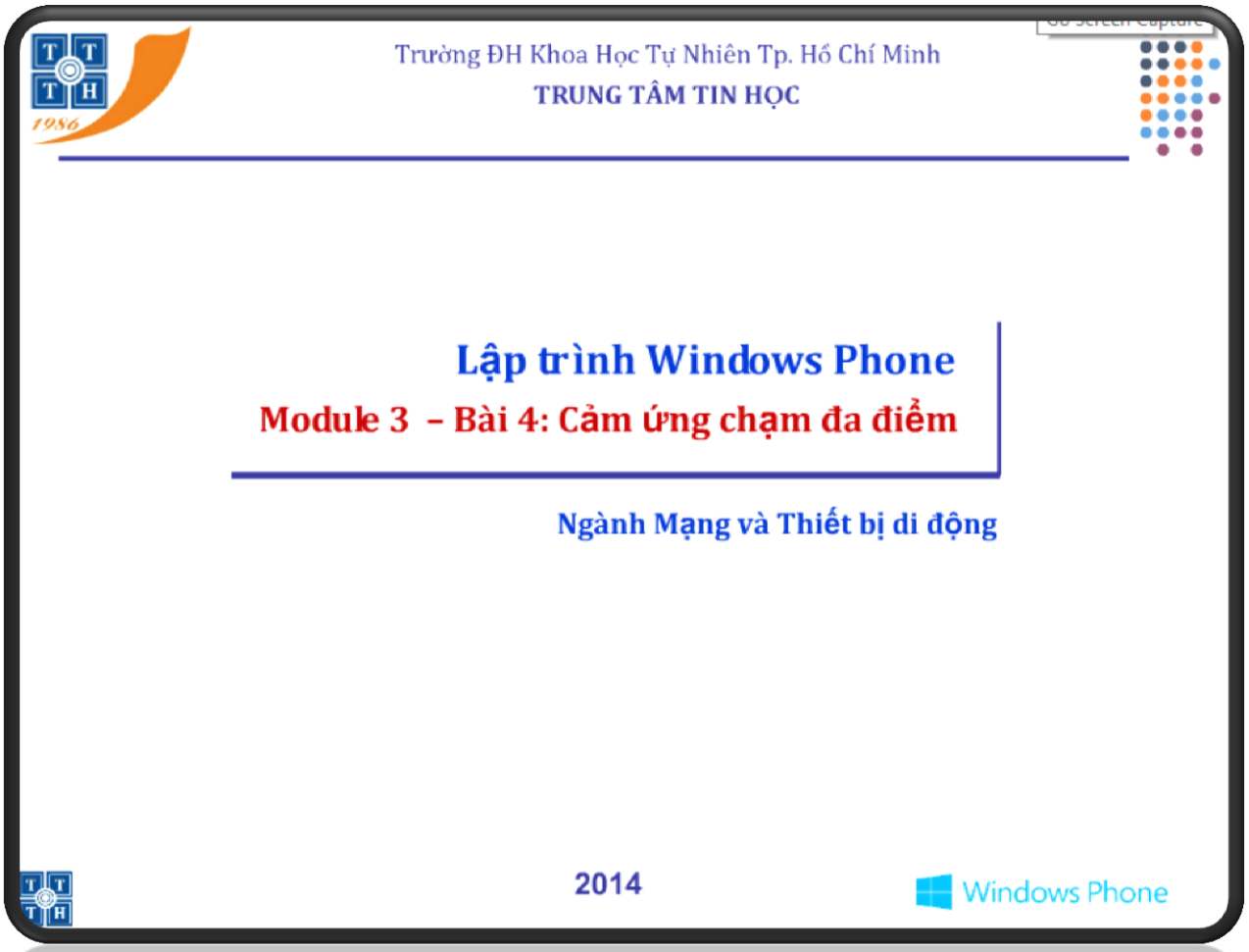
✓ Path cho phép vẽ các đối tượng Shape một cách linh động, dưới đây là một số ví dụ về cách sử dụng Path trong XAML code.

```
<Canvas>
  <Path Fill="Gold" Stroke="Black"
StrokeThickness="1">
    <Path.Data>
      <EllipseGeometry Center="50,50"
RadiusX="50" RadiusY="50" />
    </Path.Data>
  </Path>
</Canvas>
```



Thảo luận

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC


Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC
Lập trình Windows Phone
Module 3 – Bài 4: Cảm ứng chạm đa điểm
Ngành Mạng và Thiết bị di động
2014
Windows Phone

Nội dung

1. Khái niệm Gestures
2. Tap, Double – Tap, Hold event
3. Pan
4. Flicking
5. Pinch và Stretch

1. Khái niệm Gestures

Đơn điểm
- Tap
- Double Tap
- Touch and Hold
- Pan
- Flick

Đa điểm
- Pinch
- Stretch

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2. Tap, Double – Tap, Hold event

Tap

Hành động

- Dùng ngón tay chạm lên đối tượng rồi nhả ra

Sử dụng:

- MouseLeftButton Up/Down
- Button Click
- Toolkit Tap Event

2. Tap, Double – Tap, Hold event

Double Tap

Hành động:

- Nhấn 2 lần liên tục vào đối tượng để phóng to hay thu nhỏ

Sử dụng:

- Toolkit DoubleTap Event

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2. Tap, Double – Tap, Hold event

Touch and Hold

Hành động:

- Nhấn và giữ lâu ngón tay trên đối tượng → hiển thị Context Menu

Sử dụng:

- Toolkit Hold Event
- Custom Timer
  - Touch FrameReported
  - MouseLeftButton Up/Down

3. Pan

Pan

Hành động:

- Dùng ngón tay chạm đối tượng rồi quét trên màn hình theo hướng nào đó.

Sử dụng:

- Thumb DragDelta
- ManipulationDelta Translate
- Touch FrameReported TouchPoint.Position
- Toolkit DragDelta
- VisualStateManager
- FluidMove Behaviors

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

4. Flicking

Flick

Hành động:

- Đổi trang, đổi nội dung
- Thường đi sau hành động Pan

Sử dụng:

- ScrollViewer, ListBox, Panorama, Pivot
- ManipulationDelta Delta Translate
- Toolkit Flick
- Toolkit PageTransition
- VisualStateManager

5. Pinch và Stretch

Pinch và Stretch

Hành động:

- Hai ngón tay chạm trên màn hình và tiến hành giãn ra hoặc sát lại gần nhau.

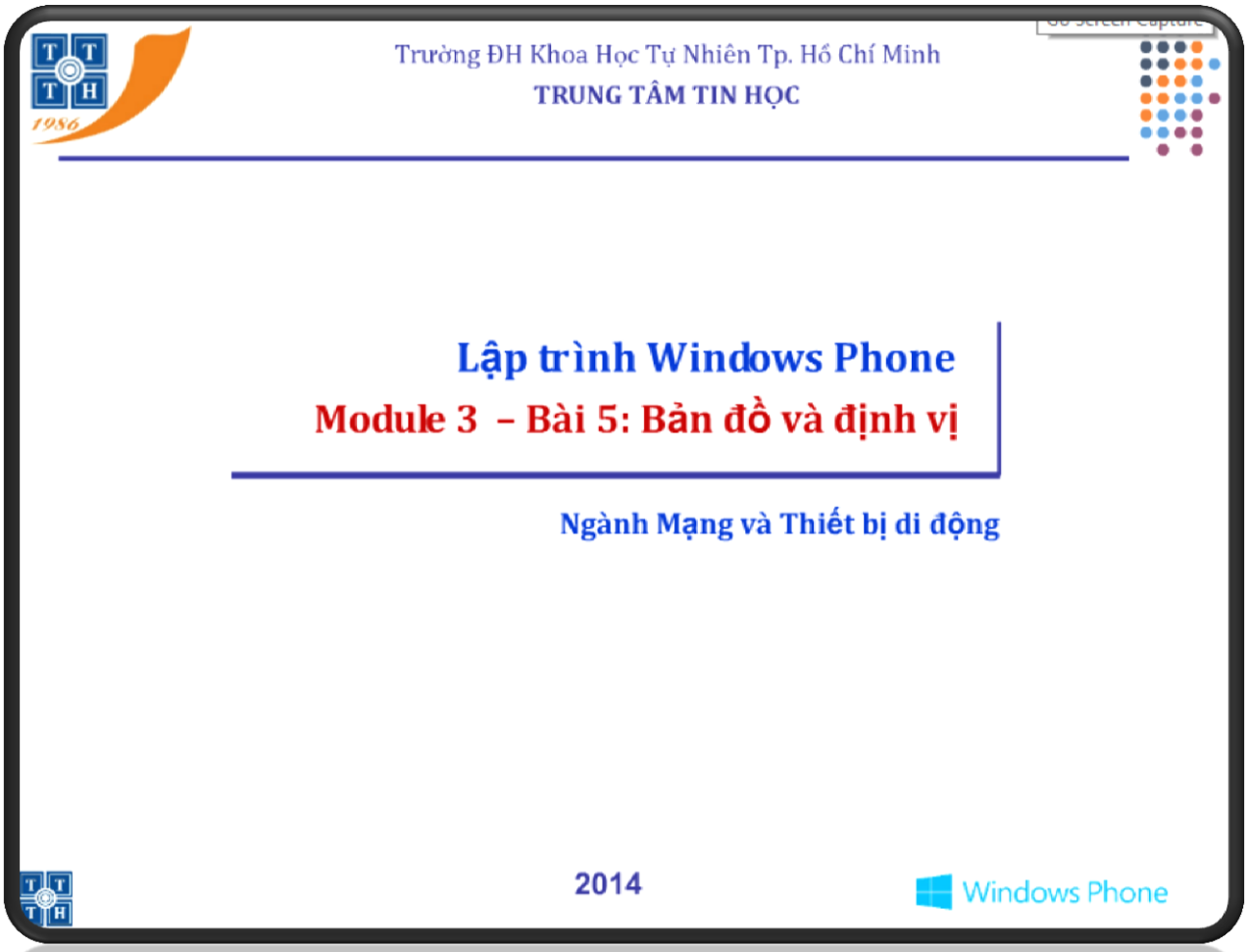
Sử dụng:

- ManipulationDelta Scale
  - No Rotate
- Toolkit Pinch

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

Thảo luận

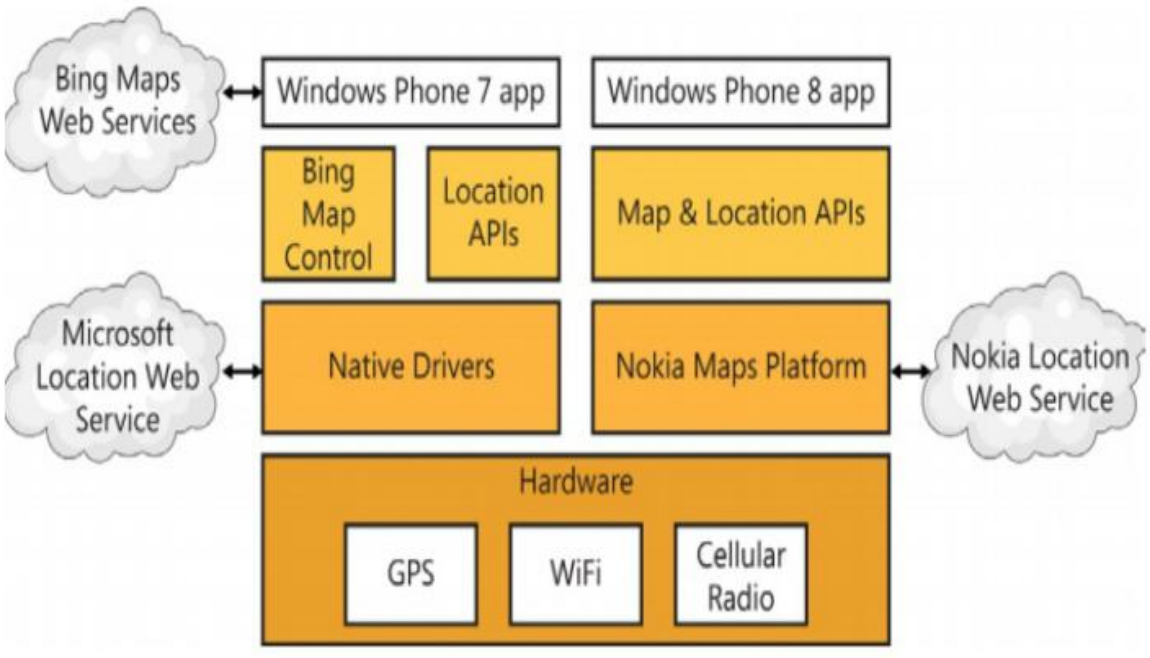
Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC


Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC
Lập trình Windows Phone
Module 3 – Bài 5: Bản đồ và định vị
Ngành Mạng và Thiết bị di động
2014
Windows Phone

Nội dung

1. Kiến trúc bản đồ và định vị
2. Maps Task Launchers
3. Cách sử dụng Map Control
4. Cách tạo Map overlay
5. Mở rộng cách tạo đường đi và chỉ dẫn

1. Kiến trúc bản đồ và định vị



Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2. Maps Task Launchers

1. MapsTask
2. MapsDirectionsTask
3. MapDownloaderTask v à MapUpdaterTask
4. Xác định vị trí hiện tại trên bản đồ

2.1 MapsTask

Thuộc tính	Mô tả
Center	Công cụ được built – in Windows
SearchTerm	Phone giúp ta tìm ki ế m địa điểm một
ZoomLevel	cách nhanhchóng.

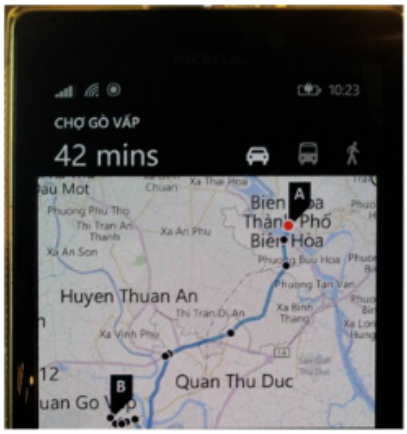
```
MapsTask mapTask = new MapsTask();
mapTask.ZoomLevel = 15;
mapTask.SearchTerm = searchTerm;
mapTask.Show();
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2.2. MapsDirectionsTask

Công cụ giúp chúng ta tìm đường đi và chỉ dẫn

```
MapsDirectionsTask mapsDic = new MapsDirectionsTask();
LabeledMapLocation start = new LabeledMapLocation();
start.Label = txtStart.Text;
LabeledMapLocation end=new LabeledMapLocation();
end.Label = txtEnd.Text;
mapsDic.Start = start; // điểm bắt đầu
mapsDic.End = end; // điểm k ế t thúc
mapsDic.Show();
```



2.3. MapDownloaderTask v à MapUpdaterTask

Windows Phone cung cấp 2 Task đơn giản dễ sử dụng để giúp người dùng có thể tải bản đồ cũng như cập nhập mới.

```
MapDownloaderTask mapDownloaderTask = new
MapDownloaderTask();
mapDownloaderTask.Show();
```

```
MapUpdaterTask mapUpdaterTask = new
MapUpdaterTask();
mapUpdaterTask.Show();
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2.4. Xác định vị trí hiện tại trên bản đồ

```
private async Task<GeoCoordinate> GetCurrentCoordinate() {
        Geolocator locator = new Geolocator();
        locator.DesiredAccuracy = PositionAccuracy.High;
        Geoposition position = await locator.GetGeopositionAsync();
         GeoCoordinate coordinate =
        new GeoCoordinate(position.Coordinate.Latitude,
                          position.Coordinate.Longitude);
        return coordinate;
    }
```


3. Cách sử dụng Map Control

Map Control giống như 1 component, nó nằm trong thư viện **Microsoft.Phone.Maps.Controls**, thông qua control này ta có thể đính kèm nó một phần trong ứng dụng của mình, nó khác với MapTask launcher. Sử dụng Map control rất đơn giản và vô cùng tiện lợi.

4. Cách tạo Map overlay

4. Cách tạo Map overlay

```
MapOverlay overlay = new MapOverlay();
overlay.Content = image;
overlay.GeoCoordinate = coordinate;

layer = new MapLayer();
theMap.Layers.Add(layer);//đưa Layer vào bản đồ trước

layer.Add(overlay);// đưa overlay vào Layer sau.
```

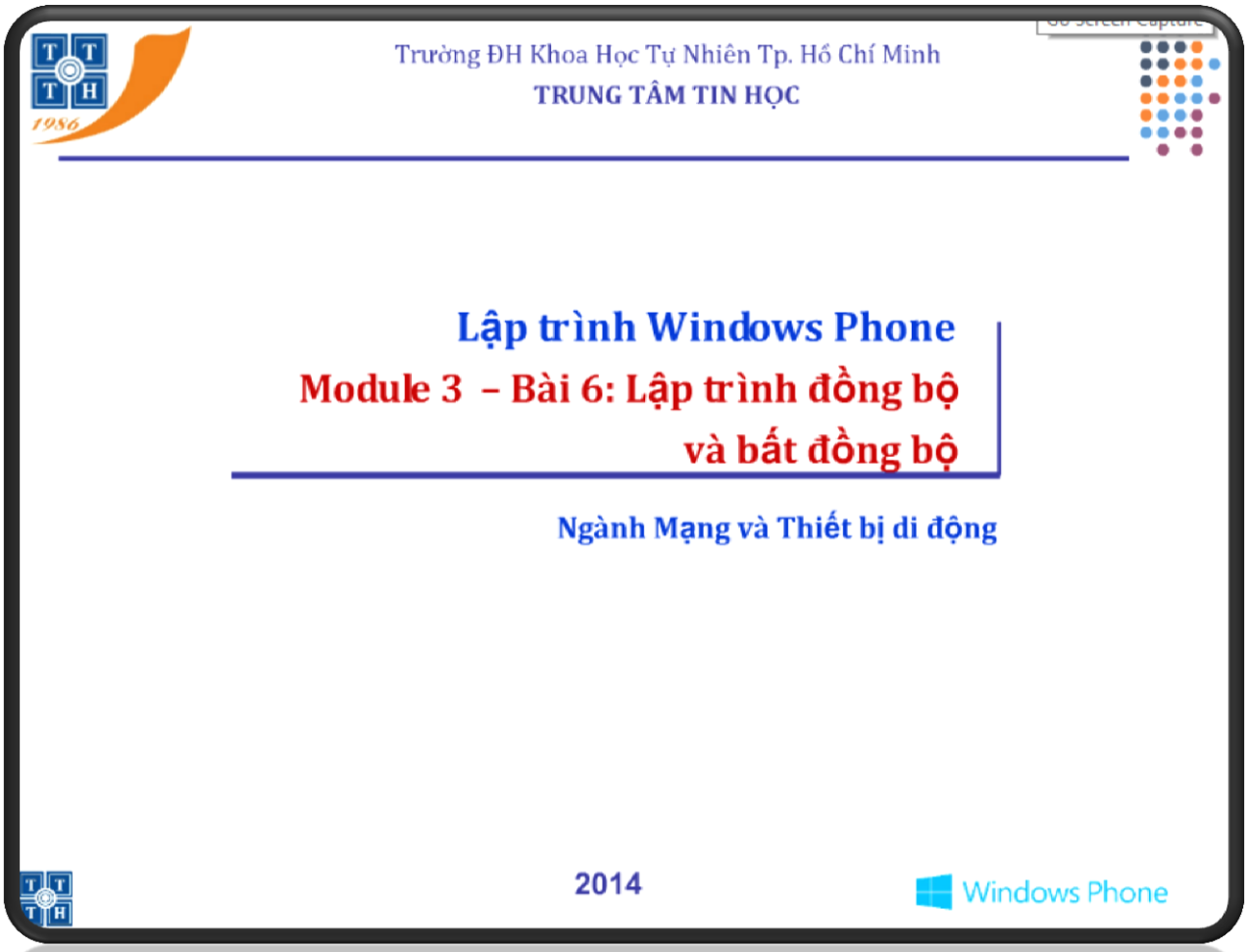

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

5. Mở rộng cách tạo đường đi và chỉ dẫn

1. GeocodeQuery
2. MapAddress
3. RouteQuery
4. RouteLeg
5. RouteManeuver

Thảo luận

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC


Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC
Lập trình Windows Phone
Module 3 – Bài 6: Lập trình đồng bộ và bất đồng bộ
Ngành Mạng và Thiết bị di động
2014
Windows Phone

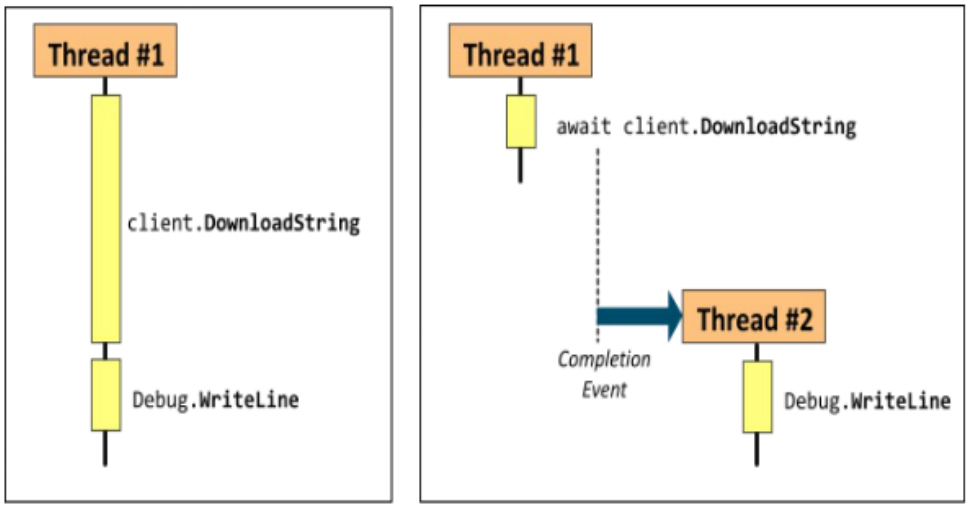
Nội dung

1. Khái niệm đồng bộ và bất đồng bộ
2. Async method
3. BackgroundWorker
4. Task

Khái niệm đồng bộ và bất đồng bộ

Synchronous (đồng bộ)

Asynchronous (bất đồng bộ)



Synchronous execution *Execution with async and await*

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

Khái niệm đồng bộ và bất đồng bộ

Synchronous method

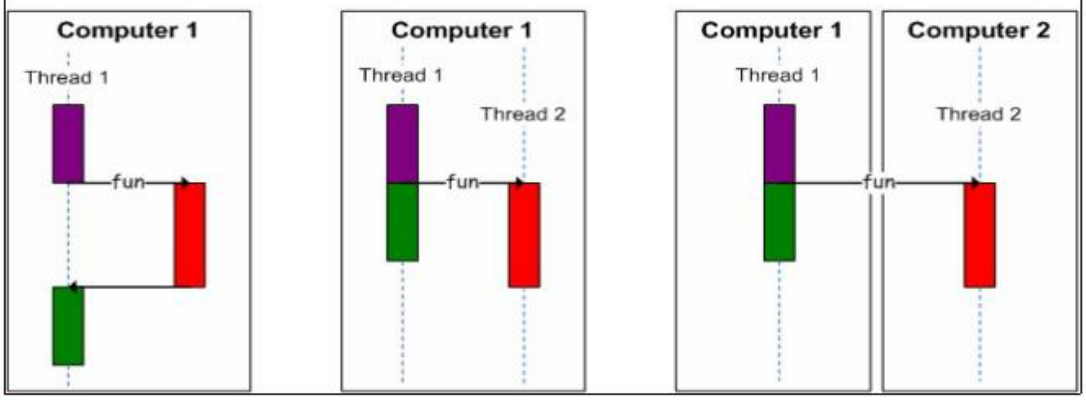
```
void fun() {
  // Business logic 1
}
...
void main() {
  // Business logic 2
  fun();
  // Business logic 3
}
```

Asynchronous method

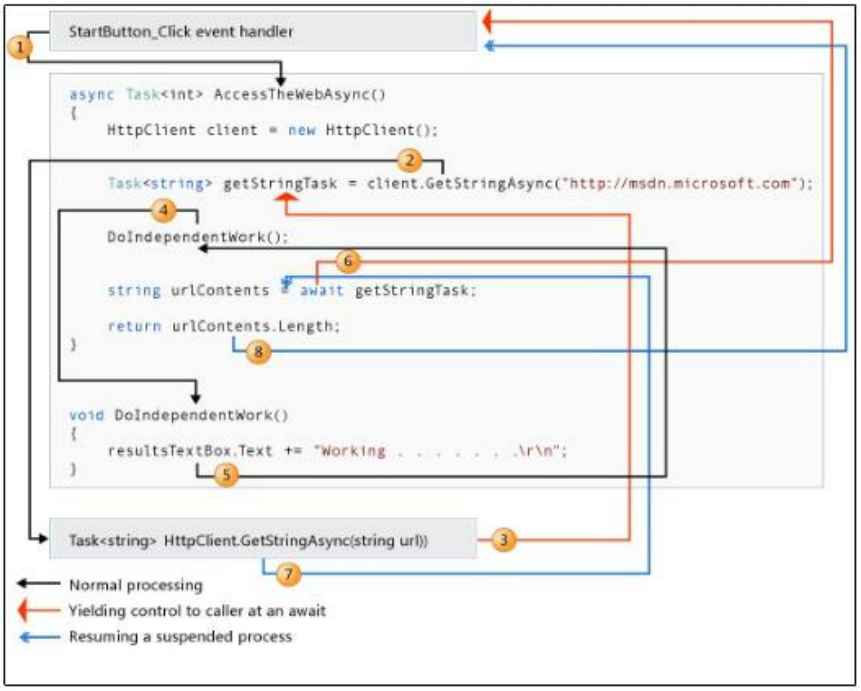
```
async fun() {
  // Business logic 1
}
...
void main() {
  // Business logic 2
  fun();
  // Business logic 3
}
```

Movable method

```
movable fun() {
  // Business logic 1
}
...
void main() {
  // Business logic 2
  fun();
  // Business logic 3
}
```



Async method



Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

Async method

```
private async void startButton_Click(object sender,
RoutedEventArgs e)
  {
    // ONE
    Task<int> getLengthTask = AccessTheWebAsync();
    // FOUR
    int contentLength = await getLengthTask;
    // SIX
    resultsTextBox.Text +=
      String.Format("\r\nLength of the downloaded string:
{0}.\r\n", contentLength);
  }
```

Async method

```
async Task<int> AccessTheWebAsync()
  {
    // TWO
    HttpClient client = new HttpClient();
    Task<string> getStringTask =
      client.GetStringAsync("http://msdn.microsoft.com");
    // THREE
    string urlContents = await getStringTask;
    // FIVE
    return urlContents.Length;
  }
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

BackgroundWorker

BackgroundWorker là một lớp cho phép ta viết đa tiến trình để cập nhật giao diện ở thời gian thực, nằm trong thư viện **System.ComponentModel**

```
BackgroundWorker bw = new BackgroundWorker();
//hỗ trợ cho phép hủy
bw.WorkerSupportsCancellation = true;
//hỗ trợ cho phép cập nhật
bw.WorkerReportsProgress = true;
//sự kiện để kích hoạt băt đầu chạy BackgroundWorker
bw.DoWork += bw_DoWork;
//Sự kiện để cập nhật
bw.ProgressChanged += bw_ProgressChanged;
//sự kiện đánh dấu là đã xong
bw.RunWorkerCompleted += bw_RunWorkerCompleted;
```


BackgroundWorker

```
bw.RunWorkerAsync();
private void bw_DoWork(object sender,
        DoWorkEventArgs e) {
        bw.ReportProgress(i * 10); }
private void bw_ProgressChanged(object sender,
ProgressChangedEventArgs e)      {
...
}
private void bw_RunWorkerCompleted(object sender,
RunWorkerCompletedEventArgs e)      {
...
    }
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

Task

1. **Run→chạy bằng hàm Run**
2. **Start → chạy bằng hàm Start**
3. StartNew → chạy bằng hàm StartNew
4. ContinueWith→Chạy Chuỗi tác vụ
5. Result→lấy kết qủa trả về
6. Wait & WaitAll→chờ 1 tác vụ, nhiều tác vụ

Run

```
public int Fib(int n){
        if (n <= 2) return 1;
        return Fib(n - 2) + Fib(n - 1);
}
var task = Task.Run(()=>Fib(10));
MessageBox.Show("Fib="+task.Result);
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

Start

```
int[] nums = { 2,19, 17, 21, 4, 13, 8, 12, 7, 3, 5 };
var f0 = new Task<double>(() => nums.Average());
f0.Start();
```

StartNew

```
int n = 5;
var task = Task.Factory.StartNew(() => Fib (n));
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

ContinueWith

```
var t1 = new Task(DoOnFirst);
var t2 = t1.ContinueWith(DoOnSecond);
t1.Start();
```

Wait & WaitAll

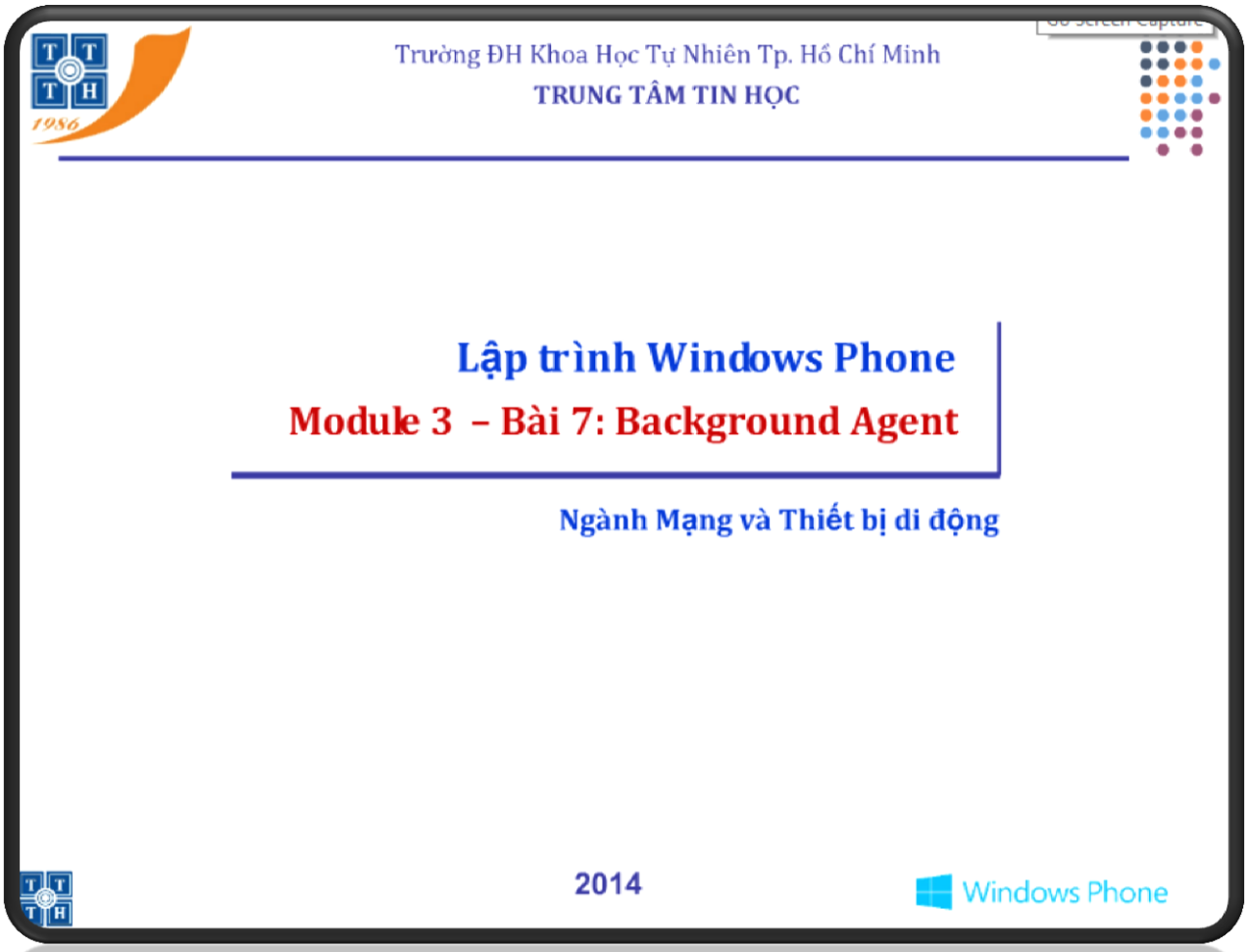
```
var task = Task.Factory.StartNew(() =>
        {
            int n = 5;
            int s = 0;
            for (int i = 1; i <= n; i++)
                s += i;
            return s;
        });
task.Wait();
MessageBox.Show("s="+task.Result);
```

Wait & WaitAll

```
public int Func1(int x) {return x * 2;}
public void Func2()
{....
}

var task1 =  Task.Factory.StartNew(() => Func1(6));
var task2 = Task.Factory.StartNew(() => Func2());

Task.WaitAll(task1, task2);
MessageBox.Show("x của task1="+task1.Result);
```


Thảo luận

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC


Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC
Lập trình Windows Phone
Module 3 – Bài 7: Background Agent
Ngành Mạng và Thiết bị di động
2014
Windows Phone

Nội dung

1. Alarm
2. Reminder
3. Background Transfer Service
4. Generic Background Agents
5. Background audio

1. Alarm

Cho phép gọi chương trình 1 lần hoặc theo định kỳ mà không lệ thuộc vào ứng dụng mà người sử dụng đang thao tác.

using Microsoft.Phone.Scheduler

Bước 1: Tạo một đối tượng **Alarm**

Bước 2: Thiết lập các thuộc tính cho **Alarm**

Bước 3: Đưa Alarm vào **ScheduledActionService**

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

1. Alarm

```
Alarm alarm = new Alarm("Cafe");
alarm.Content = "Nghỉ trưa rồi! đi cà phê nhé bạn!";
alarm.BeginTime = DateTime.Now.AddSeconds(5);
alarm.ExpirationTime =
DateTime.Now.AddSeconds(10);
alarm.Sound = new Uri("/Assets/ringout.wav",
                      UriKind.Relative);
ScheduledActionService.Add(alarm);
```

1. Alarm

```
IEnumerable<Alarm> oldAlarms =
ScheduledActionService.GetActions<Alarm>();
foreach (Alarm alarm in oldAlarms)
{
ScheduledActionService.Remove(alarm.Name);
}
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2. Reminder

Cho phép gọi chương trình 1 lần hoặc theo định kỳ mà không lệ thuộc vào ứng dụng mà người sử dụng đang thao tác.

```
using Microsoft.Phone.Scheduler
```

Bước 1: Tạo một đối tượng **Reminder**

Bước 2: Thiết lập các thuộc tính cho **Reminder**

Bước 3: Đưa Alarm vào **ScheduledActionService**

2. Reminder

```
var uniqueName = Guid.NewGuid().ToString();
Reminder reminder = new Reminder(uniqueName);
reminder.BeginTime = DateTime.Now.AddSeconds(3);
reminder.ExpirationTime =
DateTime.Now.AddSeconds(10);
reminder.Title = "Mở trang mới không?";
reminder.Content = "Chạm vào đây để mở trang mới";
reminder.NavigationUri = new Uri("/newpage.xaml",
                UriKind.Relative);
ScheduledActionService.Add(reminder);
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2. Reminder

```
var oldReminders =
ScheduledActionService.GetActions<Reminder>();
foreach (Reminder r in oldReminders)
        {
            ScheduledActionService.Remove(r.Name);
        }
```

3. Background Transfer Service

Khi tải hay upload dữ liệu từ website xuống ta thường dùng **HttpWebRequest** hay **WebClient**, nếu như nội dung quá lớn (hình ảnh, video có dung lượng lớn chẳng hạn) thì nó sẽ chiếm rất nhiều thời gian làm cho ứng dụng bị treo, để giải quyết vấn đề này thì có có thể sử dụng **Background Transfer Service**

using Microsoft.Phone.BackgroundTransfer

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

3. Background Transfer Service

```
BackgroundTransferRequest btr;
btr = new BackgroundTransferRequest
        (remoteFileUri, localFileUri);

BackgroundTransferService.Add(btr );

btr.TransferProgressChanged +=
btr_TransferProgressChanged;
btr.TransferStatusChanged +=
btr_TransferStatusChanged;
```

Background Agent 10

3. Background Transfer Service

```
private void btr_TransferProgressChanged(
object sender, BackgroundTransferEventArgs e)
{
Dispatcher.BeginInvoke(() =>
        {
progressBar.Maximum = btr.TotalBytesToReceive;
progressBar.Value = btr.BytesReceived;
        });
    }
```

Background Agent 11

3. Background Transfer Service

```
private void btr_TransferStatusChanged(object
sender, BackgroundTransferEventArgs e)
{
        //Đã tải hoàn thành
if (btr.TransferStatus == TransferStatus.Completed)
        {
            //xử lý sau khi tải xong
         }

 }
foreach(BackgroundTransferRequest res in
        BackgroundTransferService.Requests)
{
}
```

Background Agent 12

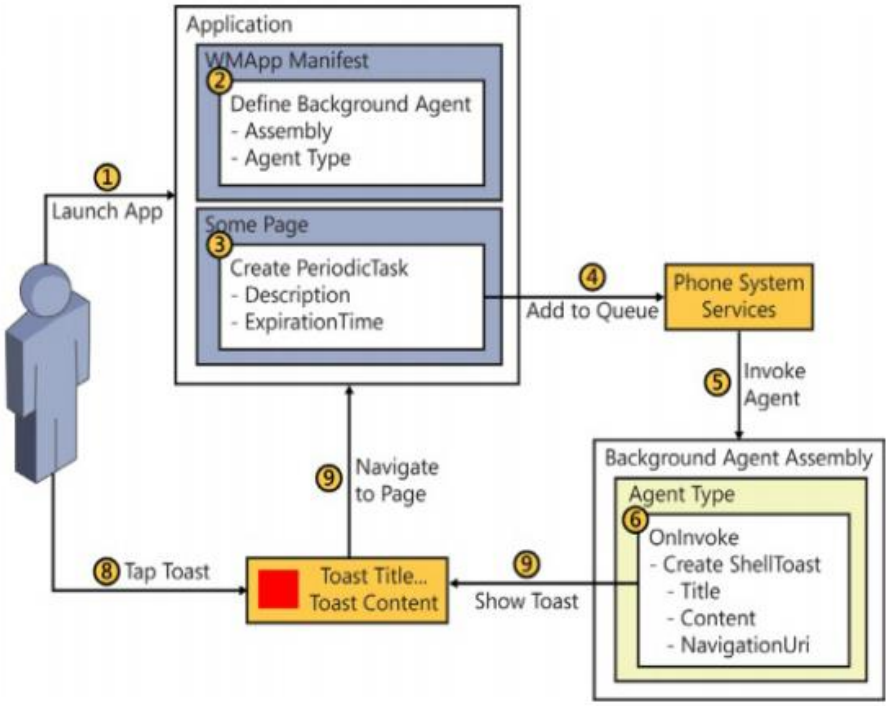
4. Generic Background Agents

Generic Background Agents (**GBA**) Cho phép chạy Background khi ứng dụng không còn trong trạng thái Foreground lifetime

- ✓ **Periodic (PeriodicTask):**
- ✓ **Resource-intensive(ResourceIntensiveTask):**

Background Agent 13

4. Generic Background Agents



4. Generic Background Agents

❖ **Các bước tạo GBA**

1. Tạo Main App Project
2. Thêm 1 Project Windows Phone Scheduled Task Agent vào solution,
3. Từ Main App Project vừa tạo ở bước 1 ta thêm chiếu tới Agent project ở bước 2.
4. Cập nhật lại Main App Manifest : Thêm **ExtendedTaskentry** để tạo background agent.

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

4. Generic Background Agents

5. Trong Main Project, gọi hàm **ScheduledActionService.Add** để đăng ký Agent với hệ thống.

6. Gọi **ScheduledActionService.LaunchForTest** để kích hoạt chương trình test khả năng chạy Agent (có thể bỏ qua bước này).

7.override hàm OnInvoke trong và sau cùng là **gọi NotifyComplete** trong lớp ScheduledAgent

Background Agent 16

5. Background audio

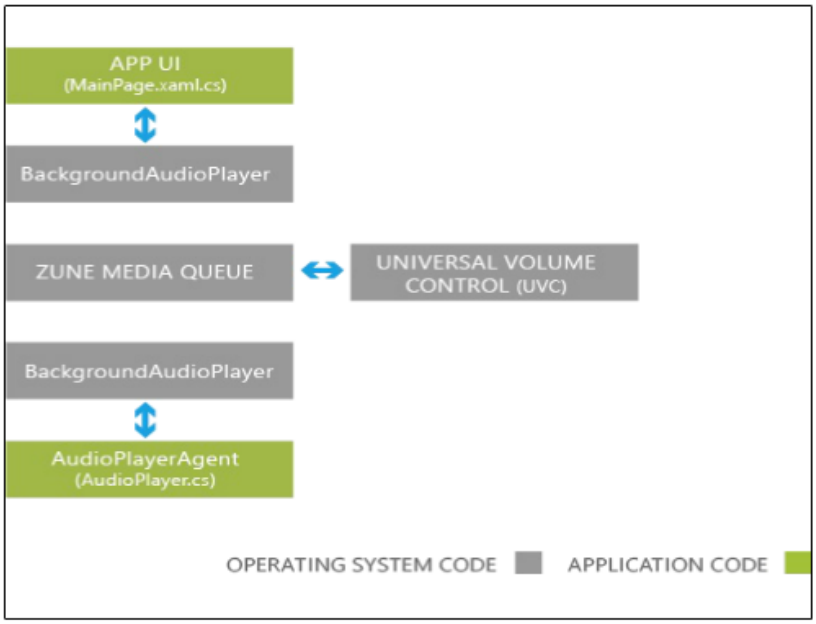
```
using Microsoft.Xna.Framework.Media;
using Microsoft.Xna.Framework;
```

- Có 2 cách chạy nền:
- ✓ **AudioPlayerAgent,**
- ✓ **AudioStreamingAgent**

Background Agent 17

5. Background audio

✓ **AudioPlayerAgent**



5. Background audio

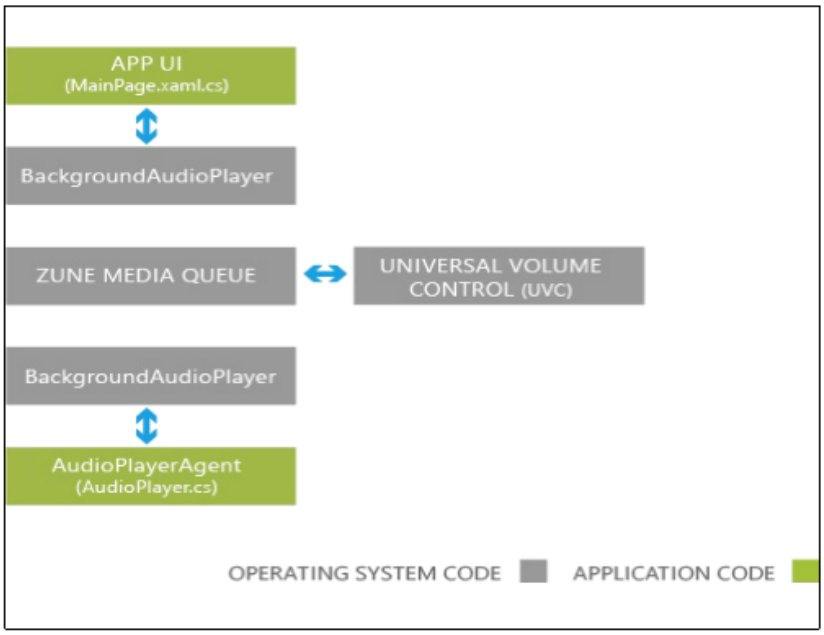
✓ **AudioPlayerAgent**

1. **OnPlayStateChanged** : Hàm này để tự động kiểm tra sự thay đổi trạng thái play, bao gồm: buffering, playing, track ready, track ended, v à shutdown.
2. **OnUserAction**: Hàm này sẽ được triệu gọi khi người sử dụng tiến hành những thao tác nào đó trong Application's audio playlist ở vòng đời Foreground (dùng controls provider ho ặc Universal Volume Control)
3. **OnError**: Hàm này sẽ được triệu gọi khi có lỗi sảy ra trong quá trình audio playback.

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

5. Background audio

✓ **AudioStreamingAgent**



5. Background audio

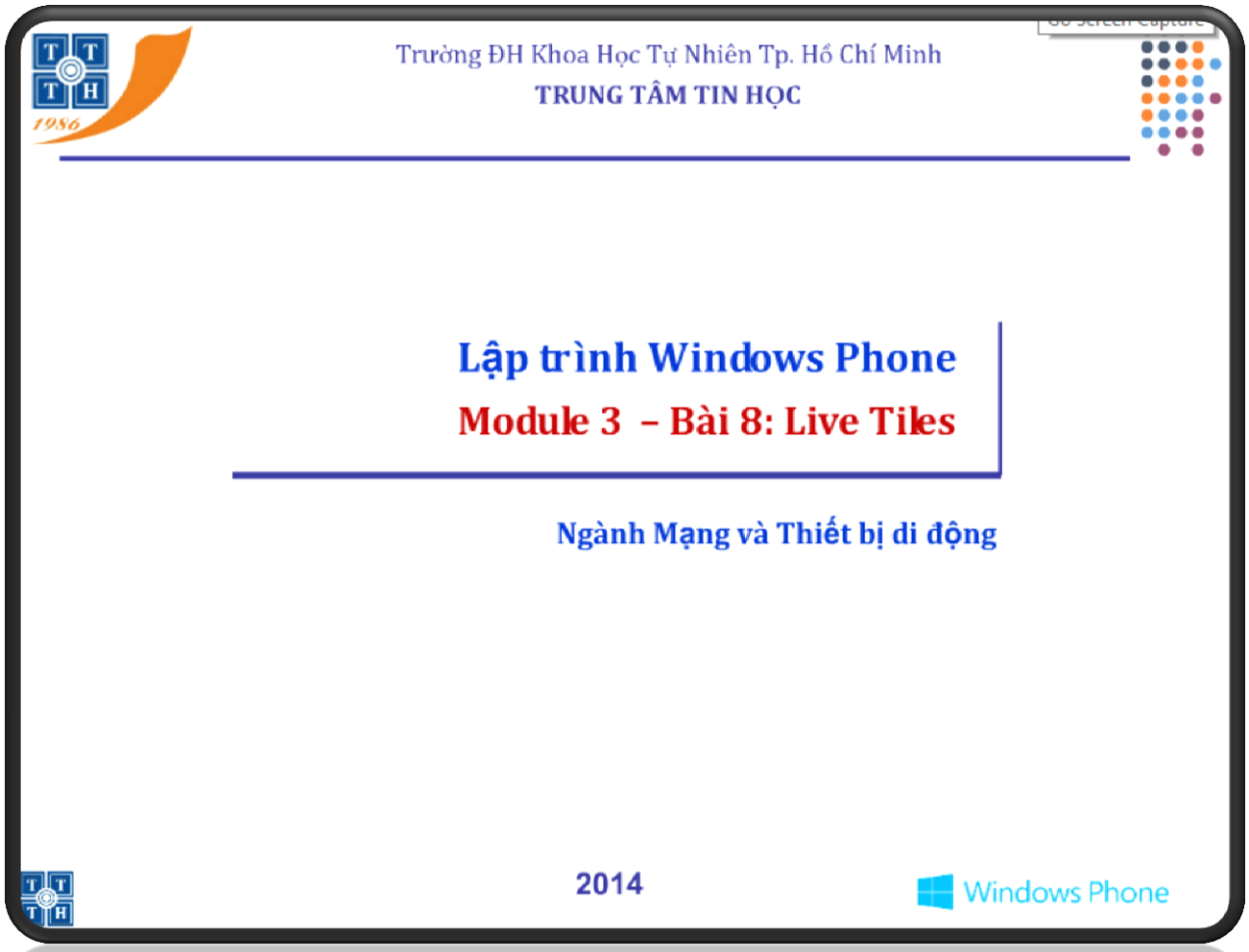
✓ **AudioStreamingAgent**

1. **OnBeginStreaming:** Hàm này sẽ được triệu gọi khi ta bắt đầu Audio Streaming
2. **OnCancel:** Hàm này sẽ được triệu gọi khi Audio Stream bị hủy.

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

Thảo luận

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC


Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC
Lập trình Windows Phone
Module 3 – Bài 8: Live Tiles
Ngành Mạng và Thiết bị di động
2014
Windows Phone

Nội dung

1. Giới thiệu Live Tiles
2. Cách tạo Live Tiles
3. Iconic
4. Flip
5. Cyclic
6. Cách update Tiles

1. Giới thiệu Live Tiles

Template	Tile size	Image size	Usable area
Flip	Small	159x159	121x121
	Medium	336x336	298x298
	Large	691x336	653x298
Cycle	Small	159x159	159x159
	Medium	336x336	336x336
	Large	691x336	691x336
Iconic	Small	71x110	71x110
	Medium	134x202	134x202

```
using Microsoft.Phone.Shell;
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2. Cách tạo Live Tiles

Ta có thể tạo Live Tiles bằng 2 cách:

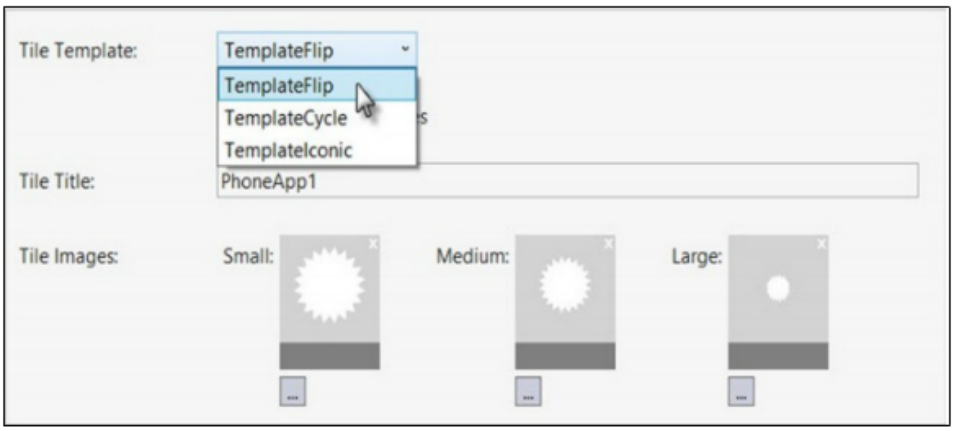
- XAML
- Coding behind

2. Cách tạo Live Tiles

- XAML

Chỉnh trong WMAppManifest.xml (GUI và XAML)

GUI:



Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2. Cách tạo Live Tiles

- XAML

```
<TemplateFlip>
    <SmallImageURI IsRelative="true"
    IsResource="false">
    Assets\Tiles\FlipCycleTileSmall.png
    </SmallImageURI>
    <Count>0</Count>
    <Title>My Title</Title>
</TemplateFlip>
```

2. Cách tạo Live Tiles

- Coding behind

```
var tileData = new IconicTileData()
        {
           Title = "Surf Sites",
           SmallIconImage=,
           IconImage =,
           Count = 6
        };
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2. Cách tạo Live Tiles

- Coding behind

```
Uri uri = new Uri("/SecondPage.xaml", UriKind.Relative);
var tile = ShellTile.ActiveTiles.FirstOrDefault(t =>
t.NavigationUri.Equals(uri));
if (tile == null)
{ShellTile.Create(uri, tileData, true);}
else
{tile.Update(tileData);};
```

Live Tiles 8

3. Iconic

Live Tiles 9

3. Iconic

```
<TemplateIconic>
    <SmallImageURI …/>
    <Count>6</Count>
    <IconImageURI IsRelative…/>
    <Title …/>
    <Message …/>
    <BackgroundColor …/>
    <LargeContent1 …/>
</TemplateIconic>
```

3. Iconic

```
Uri uri = new Uri(…);
var tileData = new IconicTileData() {
        Title = "Surf Sites",
        SmallIconImage =,
        Count = 6,
        WideContent1 = "LargeContent1",
    };
ShellTile.Create(uri, tileData, true);
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

4. Flip

4. Flip

```
<TemplateFlip>
    <SmallImageURI …/>
    <Count>6</Count>
    <BackgroundImageURI …/>
    <Title>Surf Sites</Title>
    <BackContent …/>
    <LargeBackBackgroundImageURI …/>
</TemplateFlip>
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

4. Flip

```
Uri uri = new Uri("/SecondPage.xaml",
UriKind.Relative);
 var tileData = new FlipTileData()
 {
    Title = "Surf Sites",
    SmallBackgroundImage =     new Uri(…),
    BackgroundImage =,
    Count = 6,
};
```

4. Flip

```
ShellTile oldTile =
ShellTile.ActiveTiles.FirstOrDefault(x =>
        x.NavigationUri == uri);
 if (oldTile == null)  {
    ShellTile.Create(uri, tileData, true);
 }  else  {
    oldTile.Update(tileData);
 }
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

5. Cyclic

Template cho phép hiển thị 9 hình ảnh liên tiếp theo chu kỳ

5. Cyclic

```
<TemplateCycle>
    <SmallImageURI …/>
    <Title>Surf Sites</Title>
    <Photo01ImageURI …/>
    <Photo02ImageURI …/>
    <Count>6</Count>
</TemplateCycle>
```


Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

5. Cyclic

```
Uri uri = new Uri(…);
 var tileData = new CycleTileData()  {
    Title = "Surf Sites",
   Count = 6,
    CycleImages = new List<Uri>()        {
          new Uri(…),
          new Uri(…),        },
};
```

5. Cyclic

```
ShellTile oldTile =
ShellTile.ActiveTiles.FirstOrDefault(x =>
                x.NavigationUri == uri);
    if (oldTile == null)     {
       ShellTile.Create(uri, tileData, true);
    }    else     {
       oldTile.Update(tileData);
    }
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

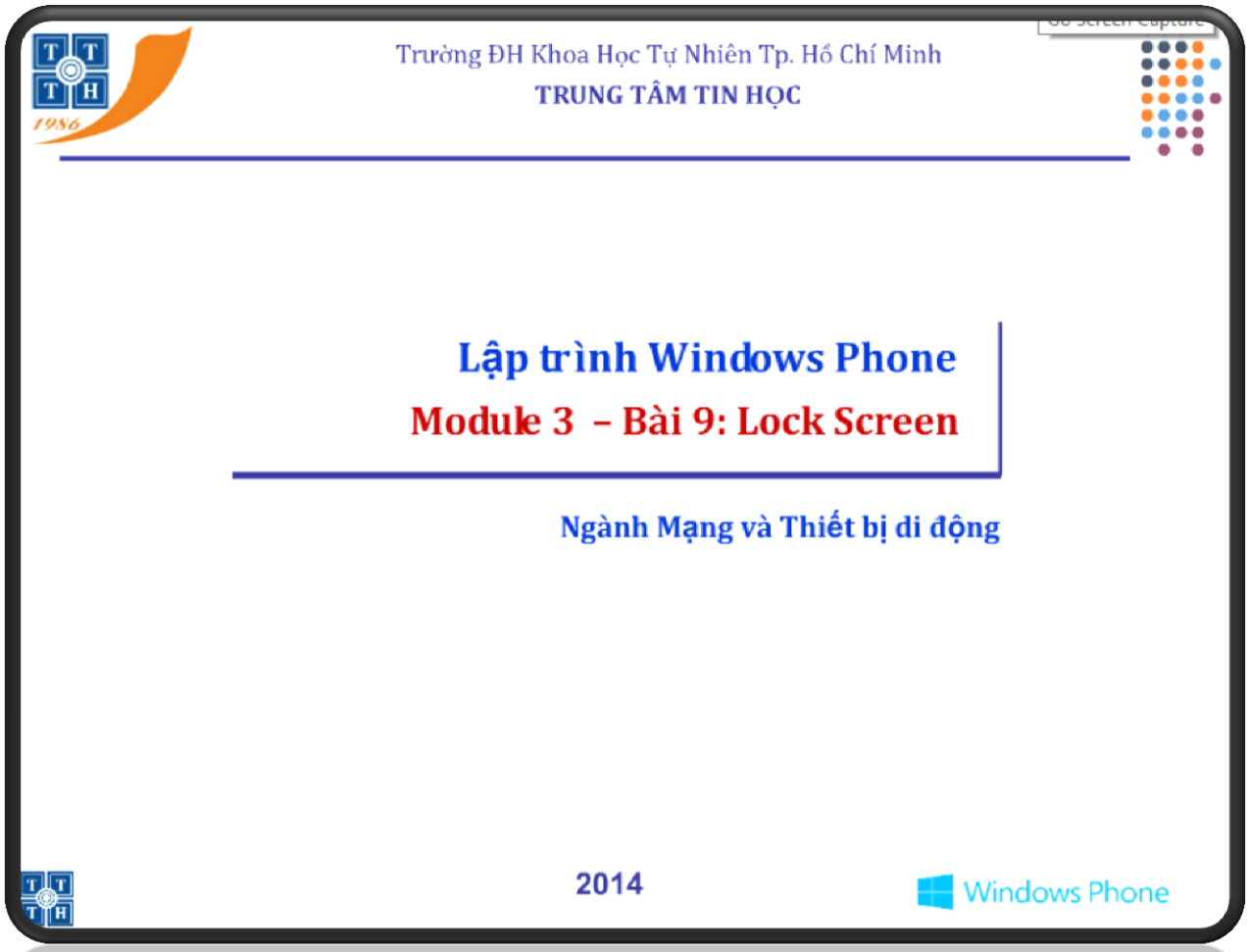
6. Cách update Tiles

Ta dùng **ShellTileSchedule** trong namespace **Microsoft.Phone.Shell** để cập nhập Tiles.

```
var tileSchedule = new ShellTileSchedule();
tileSchedule.StartTime = DateTime.Now;
tileSchedule.Recurrence = UpdateRecurrence.Interval;
tileSchedule.Interval = UpdateInterval.EveryHour;
Uri remoteUri = new Uri(@"http://drthanh.com/h1.jpg");
tileSchedule.RemoteImageUri = remoteUri;
tileSchedule.Start();
```


Thảo luận

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC


Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC
Lập trình Windows Phone
Module 3 – Bài 9: Lock Screen
Ngành Mạng và Thiết bị di động
2014
Windows Phone

Nội dung

1. Lock screen background
2. Lock screen notifications
3. Dynamic Lock screen

1. Lock screen background

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

1. Lock screen background

- Cấu hình Manifest

```
<Extensions>
    <Extension
ExtensionName="LockScreen_Background"
ConsumerID="{111DFF24-AA15-4A96-8006-
2BFF8122084F}" TaskID="_default" />
</Extensions>
```

1. Lock screen background

- Thiết lập Lock Screen

```
 var schema = isAppResource ? "ms-appx:///" : "ms-
appdata:///Local/";

var uri = new Uri(schema + filePathOfTheImage,
UriKind.Absolute);

// Set the lock screen background image:
Windows.Phone.System.UserProfile.
        LockScreen.SetImageUri(uri);
```

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

1. Lock screen background

- Setting Lock Screen


```
var op = await
Windows.System.Launcher.LaunchUriAsync(new
Uri("ms-settings-lock:"));
```


2. Lock screen Notification

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

2. Lock screen Notification

```
<DeviceLockImageURI IsRelative="true"
IsResource="false"> Assets\LockImage.png
</DeviceLockImageURI>
-----------------------------------------------------------
<Extensions>
        <Extension
    ExtensionName="LockScreen_Notification_IconCount"
    ConsumerID="{111DFF24-AA15-4A96-8006-
    2BFF8122084F}" TaskID="_default" />
        <Extension
    ExtensionName="LockScreen_Notification_TextField"
    ConsumerID="{111DFF24-AA15-4A96-8006-
    2BFF8122084F}" TaskID="_default" />
</Extensions>
```


3. Dynamic Lock screen

Step 1: Có thể thay đổi hình nền từ thư mục App install hoặc tải từ internet, RSS.

Step 2: Tải hình ảnh từ mạng và dùng **IsolatedStorageFile** để lưu trữ.

Step 3: Sử dụng **ScheduledTask** để tạo chu kỳ thay đổi hình nền Lock Screen

(Chi tiết trong giáo trình)

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC

Thảo luận

Bản quyền © Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên | Xuất bản bởi ®Công GMC