Mostrando / Escondendo conteúdo do mapa com Windows Phone API Maps

O exemplo completo para este código pode ser encontrado em Windows Phone 8 Maps examples project, o código utilizado aqui é implementado em exemplo simples no conteúdo dentro desse projeto.

Os códigos WP7 discutidos aqui são implementados em exemplos feitos para WP7, e pode ser encontrado em Maps Examples for Windows phone na página wiki.

Com WP7 Bing Maps API, você pode simplesmente usar a variável Visibility da classe de objeto mapa para defini-lo como visível ou oculto, como mostrado com o seguinte código:

```
if (poly.Visibility == System.Windows.Visibility.Visible){
    poly.Visibility = System.Windows.Visibility.Collapsed;
}
else
{
    poly.Visibility = System.Windows.Visibility.Visible;
}
```

No entanto, com WP8 API Maps isso geralmente não pode ser feito, única exceção dos marcadores com WP8 são apenas objetos de MapOverlay dentro de MapLayer. Com eles, você pode usar a variável Visibility, desde que ele esteja disponível para o conjunto de objetos como conteúdo para a MapOverlay.

O exemplo seguinte mostra como isso poderia ser feito com objetos elipse usado como conteúdo para MapOverlay.

```
if (markerLayer != null)
{
    for (var i = 0; i < markerLayer.Count(); i++)
    {
        Ellipse markker = (markerLayer[i].Content as Ellipse);
        if (markker != null)
        {
            if (markker.Visibility == System.Windows.Visibility.Visible)
            {
                markker.Visibility = System.Windows.Visibility.Collapsed;
            }
            else
            {
                markker.Visibility = System.Windows.Visibility.Visible;
            }
        }
    }
}
```

Com polígono e polilinha você ainda teria uma opção, e isso seria usar o canal alfa para definir o fator de transparência para a cor usada com o objeto mapa. Observe que, com polígono, você precisa definir a transparência tanto para o enchimento como para a linha, como mostrado aqui:

```
if (poly.StrokeColor == Color.FromArgb(0xFF, 0x00, 0x00, 0xFF))
{
    poly.FillColor = Color.FromArgb(0x00, 0x00, 0xFF, 0x00);
    poly.StrokeColor = Color.FromArgb(0x00, 0x00, 0x00, 0xFF);
}
else
{
    poly.FillColor = Color.FromArgb(0x55, 0x00, 0xFF, 0x00);
    poly.StrokeColor = Color.FromArgb(0xFF, 0x00, 0x00, 0xFF);
}
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.