Awaiting event handlers in Windows Phone 8

This article explains how to await event handlers in an async function.

Introduction

With the introduction of `async`/`await` keywords, organizing code is much easier. When we have to work with libraries that don't offer awaitable functions, maintaining code that can be overviewed easily can be painful especially when it would be more comfortable to wait for an event in an async function. (Please note: what I'm writing might be wrong - I'm just sharing my personal experience, it can be wrong)

The idea

Let's suppose that we want to use a Google Translate class and need to await the result in a function. (I chose that because I already wrote a class for it Google Translate class)

To use it in the usual way, we need to do the following:

```
class TestClass
{
 private Translator translator;

 void TestFunction()
 {
  translator = new Translator();
  translator.TranslatedString += t_TranstlatedString;
  translator.TranslateString("de","How are you?");
 }

 void t_TranstlatedString(TranslatedStringEventArgs e)
 {
  if (e.Error == false)
  {
                MessageBox.Show(e.translatedString);
  }
  else
  {
   MessageBox.Show(e.ErrorMessage, ":(", MessageBoxButton.OK);
  }
 }
}
```

If we'd like to have an async function that awaits the event handler we need to do some tricks:

```
class TestClass
{
 private Translator translator;

 private SemaphoreSlim signal;
 private TranslatedStringEventArgs eventArgs;

 async void TestFunction()
 {
  signal = new SemaphoreSlim(0,1); //0 current requests, 1 maximum request.

  translator = new Translator();
```


(C) Copyright Nokia 2014. All rights reserved.

```
  translator.TranslatedString += t_TranstlatedString;
  translator.TranslateString("de","How are you?");

  await signal.WaitAsync(5000); // it's recommended to set a timeout value, for example
5 seconds
  if (eventArgs.Error == false)
  {
                Dispatcher.BeginInvoke(()=>
MessageBox.Show(eventArgs.translatedString));
  }
  else
  {
    Dispatcher.BeginInvoke(()=> MessageBox.Show(eventArgs.ErrorMessage, ":(",
MessageBoxButton.OK));
  }
 }

 void t_TranstlatedString(TranslatedStringEventArgs e)
 {
  eventArgs = e;
  signal.Release();
 }
}
```

I would like to say that it's just my experience and possibly not the only solution or not the best solution.

(C) Copyright Nokia 2014. All rights reserved.