

HoGent

INLEIDING

- Key Windows Phone goal
 - Snelheid en responsiveness
 - Background proces mag geen impact hebben op foreground proces en batterij
- Vandaar : enkel 1 app running at a time
 - WP7 past hiervoor tombstoning toe voor non-foreground apps
 - Dormant state : bewaart app image in memory
 - = FAST RESUME (FAS)
 - ↓ Out of memory
 - Tombstoned state : WP7 slaat de application state dictionaries op voor latere restore
 - = SLOW RESUME

2

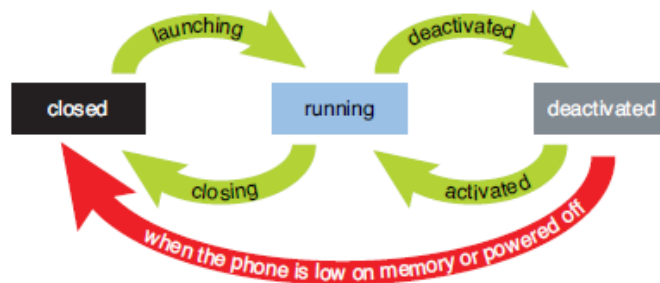
INLEIDING

- App verlaten kan op 2 manieren
 - Close :
 - Door klikken van Back button op *home* page
 - Deactivate (*dormant* -> *tombstoned*) : “pushed to the background”
 - Start button
 - Screen locking
 - Lanceren van andere app : phone call, ..
 - App lanceert zelf een andere app (launchers, choosers,..)

3

APPLICATION LIFECYCLE

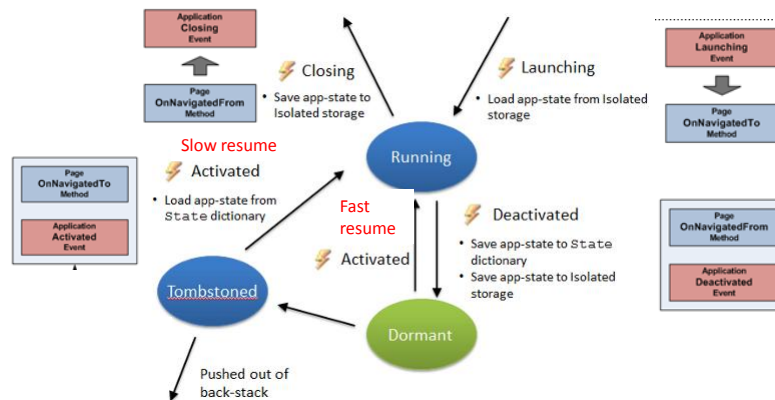
- States van app



- **PhoneApplicationService** class : biedt toegang tot de verschillende states van de application's lifetime. Dit omvat management van app idle behavior en state wanneer app ge(de)activeerd wordt

4

APPLICATION LIFECYCLE



5

APPLICATION LIFECYCLE

- States en events

- App bevatten meestal data die behouden moet blijven als app gedeactiveerd of afgesloten wordt, en gerestored moet worden bij reactivatie of herstarten.
- 2 soorten state
 - Page state**: is the state specific to a particular page, usually referring to the visual appearance (content of controls, position of scroll bars, etc.).
 - Application state**: is the state common to the entire application and not specific to a particular page.

6

APPLICATION LIFECYCLE

States en events

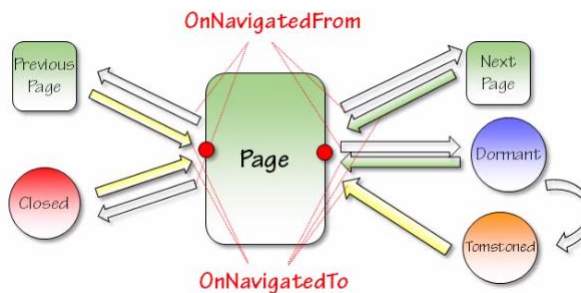
- **Page state** afhandelen in Page event handlers:
 - **OnNavigatedTo**: called when a page becomes the active page in a frame
 - **OnNavigatedFrom**: called when a page is no longer the active page in a frame
- **Application state** afhandelen in PhoneApplicationService eventhandlers
 - **Launching**: occurs when the application is being launched.
 - **Closing**: occurs when the application is terminated as a result of pressing the Back button on the first page.
 - **Activated**: occurs when the application is made active after being previously tombstoned, as a result of returning to the application with the Back button or a Launcher/Chooser finishing work.
 - **Deactivated**: occurs when the application is tombstoned as a result of pressing the Start button or a Launcher/Chooser activating.

7

PAGE STATE

Page State

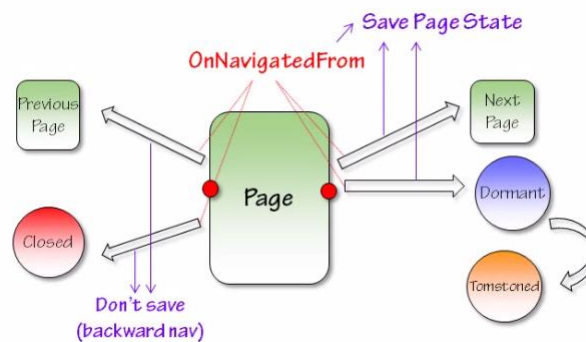
- **Page state** afhandelen in Page event handlers:
 - **OnNavigatedTo**: called when a page becomes the active page in a frame
 - **OnNavigatedFrom**: called when a page is no longer the active page in a frame



8

PAGE STATE

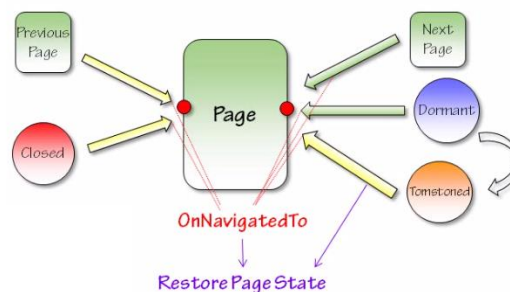
- Wanneer page state opslaan?
 - Altijd, behalve bij backward navigation.
 - Back navigation kan je controleren :
 - `NavigationEventArgs.NavigationMode != NavigationMode.Back`



9

PAGE STATE

- Wanneer page state restoren?
 - Forward navigation => nieuwe instantie
 - Retour uit dormant staat => bestaande instantie (page state is bewaard)
 - Retour uit thombst.oned state => nieuwe instantie
- => Dus enkel in laatste geval Page State restoren. Om te weten of het om een nieuwe instantie gaat => variabele aanmaken



10

PAGE STATE

- Waar page state opslaan?
 - Opslaan van de page state : PhoneApplicationService.State
 - Dictionary van key/waarde paren
 - Kan enkel data bevatten die geserialiseerd kan worden
 - Inhoud wordt automatisch gepersisteerd (to disk) als app gedeactiveerd wordt gerestored bij activatie app
 - Wordt niet gepersisteerd als app afgesloten wordt
 - Wordt geïnitieerd als app gestart wordt

11

PAGE STATE

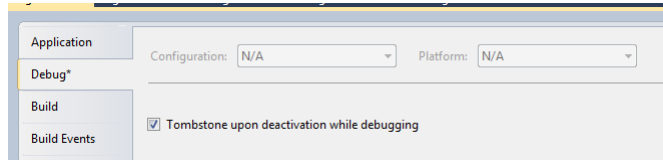
- Voorbeeld
 - App bevat
 - MainPage met hyperlinkbutton naar PageStatePage
 - PageStatePage bevat een textbox. Di de Page State + hyperlinkButton om naar AnotherPage te navigeren
 - AnotherPage
 - Run de app
 - Ga naar PageStatePage, vul een naam in
 - Druk op de Start knop om app te deactiveren
 - Druk op Back knop om app terug te activeren
 - D.i. FAS : de state blijft bewaard

12

PAGE STATE

o Voorbeeld

- Om tombstoning te simuleren
 - o Ga naar Project Properties > Debug > vink tombstoning aan



- o Run app opnieuw
 - o Herhaal handelingen
 - o Page state is verdwenen
- => Code aanpassen

13

```

public partial class Page1 : PhoneApplicationPage
{
    private bool isNewInstance;
    public Page1()
    {
        InitializeComponent();
        isNewInstance = true;
    }

    protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
    {
        base.OnNavigatedTo(e);
        if (isNewInstance)
        {
            if (State.ContainsKey("name"))
                textBoxName.Text = State["name"].ToString();
            isNewInstance = false;
        }

        protected override void OnNavigatedFrom(System.Windows.Navigation.NavigationEventArgs e)
        {
            base.OnNavigatedFrom(e);
            if (e.NavigationMode != System.Windows.Navigation.NavigationMode.Back)
                State["name"] = textBoxName.Text;
        }
    }
}

```

Restoren state als nieuwe instantie en page state werd reeds bijgehouden (di igv tombstoning)

Opslaan state, niet igv Back button

4

APPLICATION STATE

- Application level data
 - Afkomstige van een netwerk service of phone storage
- Return van dormant state => geen restore nodig
 - Testen via `ActivatedEventArgs.IsApplicationInstancePreserved`
- Return van tombstoned state => restore
- App lifecycle events in app.xaml
 - Launching, Deactivated, Activated en Closing



15

APPLICATION STATE

- Voorbeeld
 - Stel Application State is een product (later is dit een VM)
 - Zie app.xaml.cs

```

private static Product theProduct;

public static Product TheProduct
{
    get
    {
        if (theProduct == null)
        {
            theProduct = new Product { ID = 12, Name = "LG Quantum", Description = "Windows Phone 7" };
            return theProduct;
        }
        set { theProduct = value; }
    }
}

```

16

APPLICATION STATE

• Voorbeeld

- In AppStatePage.xaml.cs

```
protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
{
    base.OnNavigatedTo(e);
    if (isNewInstance)
    {
        DataContext = App.TheProduct;
    }

    if (State.ContainsKey("NotBound"))
        textBoxNotBound.Text = State["NotBound"].ToString();
    isNewInstance = false;
}
```

17

APPLICATION STATE

• Voorbeeld

- Opslag en restore van App state

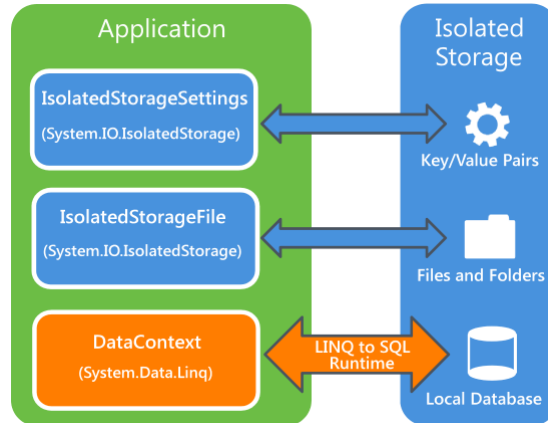
```
private void Application_Activated(object sender, ActivatedEventArgs e)
{
    if (!e.IsApplicationInstancePreserved)
    {
        var appState = PhoneApplicationService.Current.State;
        if (appState.ContainsKey("Product"))
        {
            TheProduct = appState["Product"] as Product;
        }
    }
}

// Code to execute when the application is deactivated (sent to background)
// This code will not execute when the application is closing
private void Application_Deactivated(object sender, DeactivatedEventArgs e)
{
    PhoneApplicationService.Current.State["Product"] = TheProduct;
}
```

18

APPLICATION LIFECYCLE

- Persisteren van data over verschillende instanties heen (zie later)



19

REFERENTIES

- [http://msdn.microsoft.com/en-us/library/ff817008\(v=vs.92\)](http://msdn.microsoft.com/en-us/library/ff817008(v=vs.92))
- http://www.codeguru.com/csharp/.net/net_silverlight/article.php/c18397/Windows-Phone-7-Quick-Tutorials-Part-4--Tombstoning-and-Data-Persistence.htm
- Pluralsight : Windows Phone 7

20