

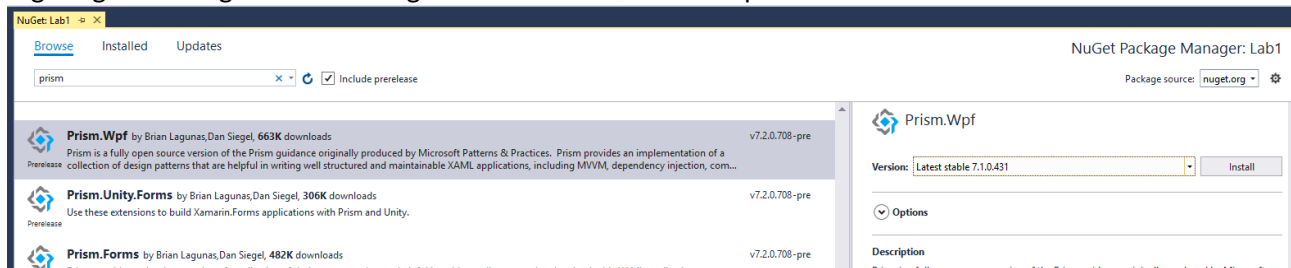
GUI Lab: Agent Assignment 2

Kommentarer til løsningsforslag

Lab1

Menu og toolbar laves i Xaml og bør ikke volde de store problemer. Den store udfordring i denne opgave er anvendelsen af DelegateCommands.

Jeg bruger "Manage NuGet Packages" til at installere Prism.Wpf



Følger eksempler vist på slide + demoen . Jeg tager udgangspunkt i eksemplet UseOfCanExecute som findes i CommandDemos.

Tilføjer f.eks. en property som returnerer en instans af CommandCommand-klassen, hvor Execute og CanExecute peger på de tilhørende handlers, f.eks.:

```

ICommand _PreviousCommand;
public ICommand PreviousCommand
{
    get
    {
        return _PreviousCommand ??
            (_PreviousCommand = new DelegateCommand(
                PreviousCommandExecute, PreviousCommandCanExecute)
                .ObservesProperty(() => CurrentIndex));
    }
}

private void PreviousCommandExecute()
{
    --CurrentIndex;
}

private bool PreviousCommandCanExecute()
{
    return CurrentIndex > 0;
}

```

Og binder til den i XAML:

```

<MenuItem Header="_Previous" Command="{Binding PreviousCommand}"/>
<Button Height="auto"
        ToolTip="Move to previous agent"
        Content="<"
        Command="{Binding Path=PreviousCommand}"
        />

```

Lab2

Tilføjer de klassiske film-menu-valg (new, open, save og saveAs). Det vil være naturligt at bruge de indbyggede fildialoger her, men da dette ikke har været pensum endnu, så bruger jeg en textBox på toolbaren til at angive filnavnet i. Selve persisteringen af data er efter "bogen" (find de relevante slides eller google C# xmlserializer).

Bemærk hvordan jeg i MainViewViewModel-konstruktoren kun indsætter data, hvis der bygges til DEBUG mode ved brug af kompilerdirektivet #if.

Jeg kunne også have valgt kun at indsætte data i design-mode:

```
public MainWindowViewModel()
{
    if ((bool)(DesignerProperties.IsInDesignModeProperty.GetMetadata(
        typeof(DependencyObject)).DefaultValue))
    {
        // In Design mode
        Agents.Add(new Agent("001", "Nina", "Assassination", "UpperVolta"));
        Agents.Add(new Agent("007", "James Bond", "Martinis", "North Korea"));
    }
}
```

Lab3

At tilføje en statusbar få et statusBarItem til at vise antallet ved at binde til Agents.Count er en let gjort i Xaml.

Afhængig af hvordan din kode er sat sammen kan der dog være problemer med at Count ikke bliver opdateret automatisk. Hvis du oplever dette, så kan problemet løses ved at kalde NotifyPropertyChanged("Count"); når antallet ændres. Dvs. i Add og Delete command-handlerne.

Uret i højre hjørne laves ved at tilføje en DispatcherTimer og initiere denne i MainViewModels konstruktør. Selve opdateringen sker i timerens tick-eventhandler. Jeg har lavet en Clock-hjælpeklasse som Xaml-koden binder til, og som opdateres fra eventhandleren.