**C# Příručka pro učitele**

**2. Lekce – Navigace v Universal Windows Platform**

Vytvořeno v rámci programu Microsoft Studentské Trenérské Centrum, dokončeno 18.11.2020

Autor: Václav Horčička

**Úvodní informace**

* V druhé lekci se podíváme na to, jak si vytvořit a upravit navigaci ve své aplikaci, přidání více stránek.

Přidání ikonek k tlačítkům a reakce na stisknutí tlačítka.

**Znalostní prerekvizity**

* Znát vše co se vyskytuje v 1. lekci

**Technické prerekvizity**

* Visual Studio s C# UWP

**Probírané funkce**

* Navigace v UWP
* Přidávání více stránek
* Styly v XAML, co je to ResourceDictionary?
* StackPanel a rozdíl mez Gridem.

**Očekávaný výstup**

* Schopnost vytvářet navigační menu
* A pomocí kódu se přesouvat z jedná stránky na druhou

**Odhadovaní časová dotace**

* 45minut

**Věková doporučení**

* 14 - 99let

Contents

[Vytvoření nové stránky 3](#_Toc56630840)

[Vytvoření a úprava navigace na stránce 3](#_Toc56630841)

[Využití navigace k přecházení mezi stránkami 5](#_Toc56630842)

[Vracení se zpět, pomocí Frame.Back() 8](#_Toc56630843)

[Reakce na tlačítka, ikony a jejich text 9](#_Toc56630844)

[StackPanel a rozdíl mezi Gridem 10](#_Toc56630845)

[XAML výchozí styly, Light/Dark Mode 11](#_Toc56630846)

[XAML ResourceDictionary 11](#_Toc56630847)

[Závěr vzorové hodiny 12](#_Toc56630848)

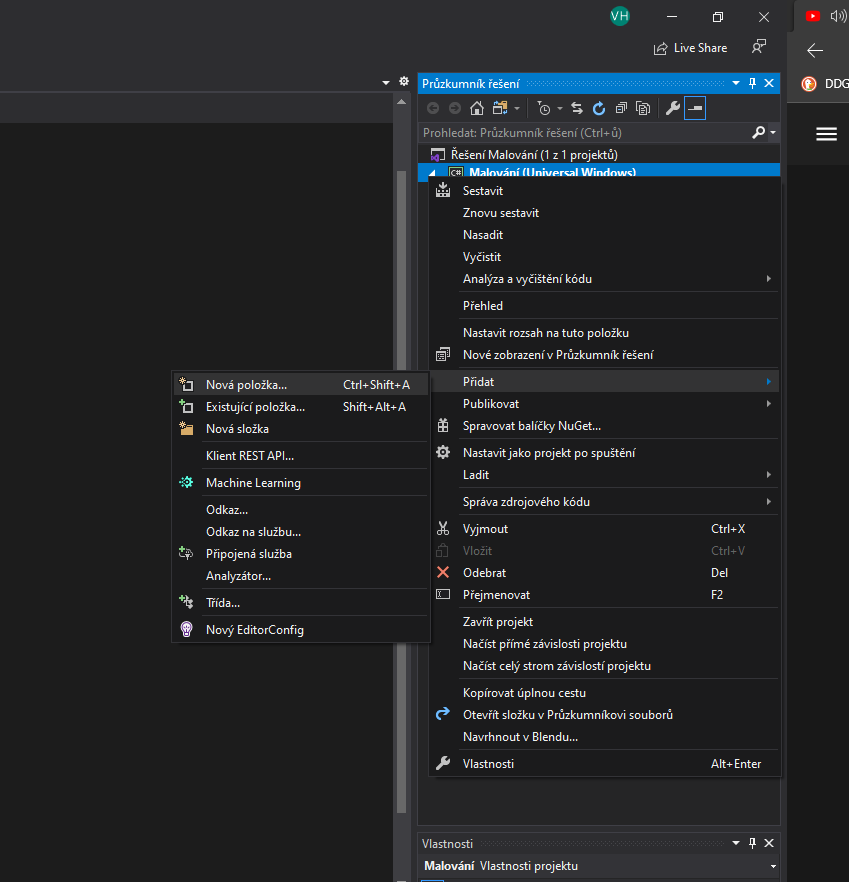
[Dokumentace Microsoft 12](#_Toc56630849)

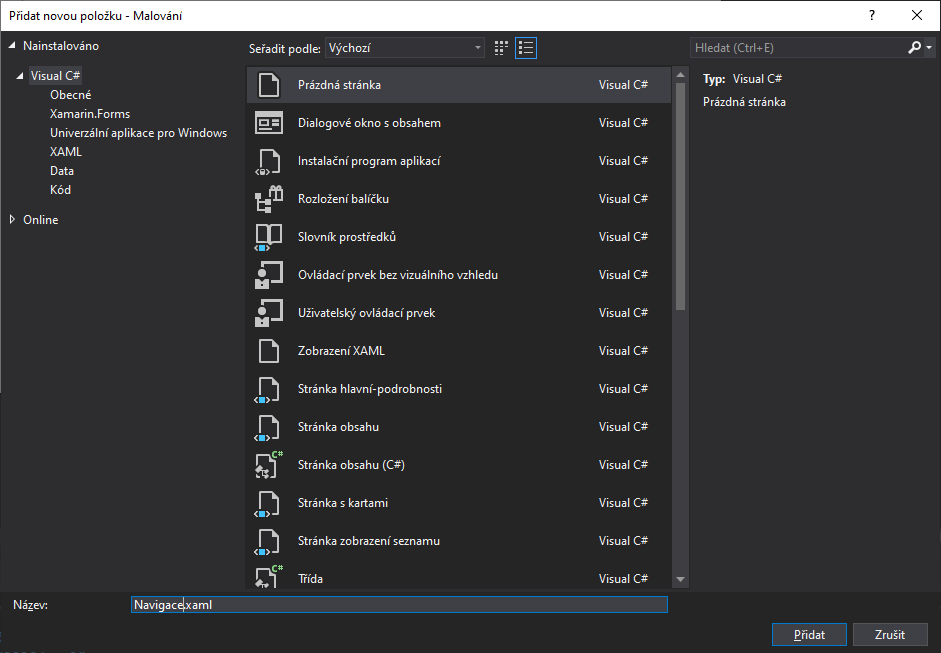
Legenda

* odkaz na dokumentaci
* zajímavost, tip či rada

## Vytvoření nové stránky

V průzkumníku řešení (nachází se na pravé straně) klikneme pravím tlačítkem myši a sjedeme ke kolonce přidat a klikneme na nová položka (klávesová zkratka: Ctrl + Shift + A) (viz obrázek 1,2)





2

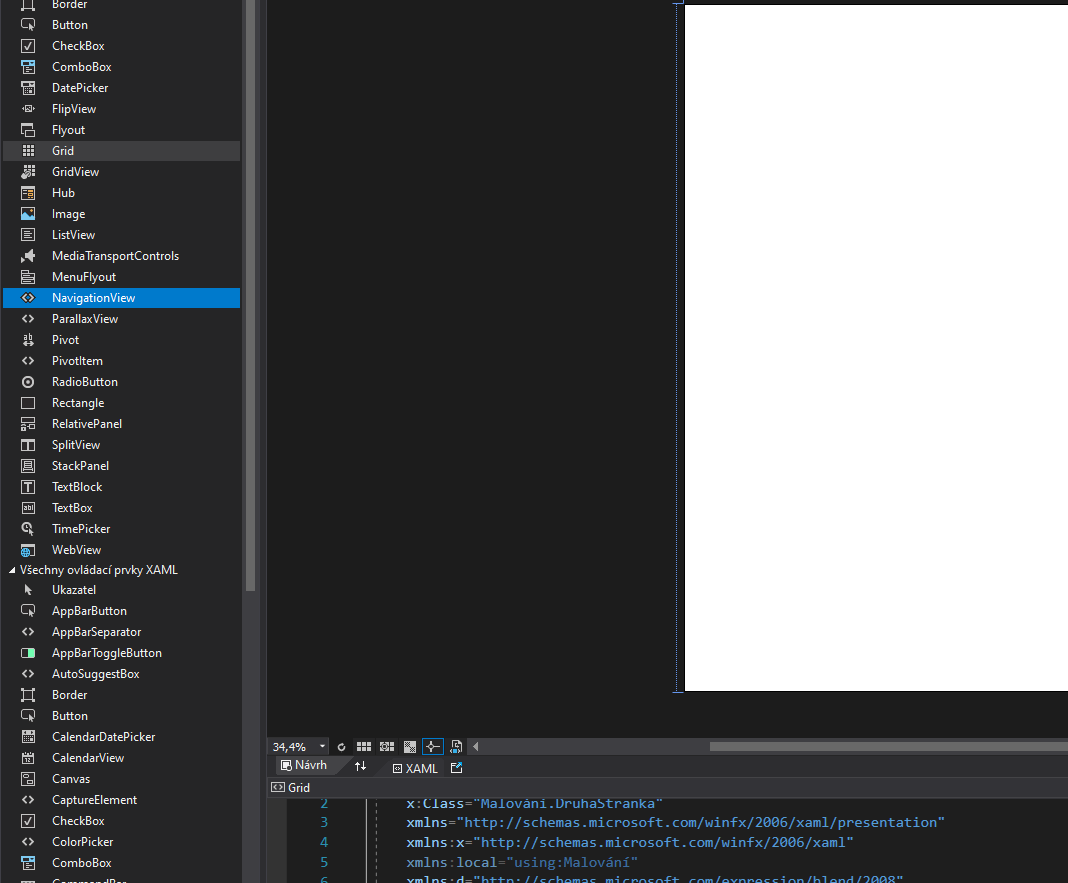
1

Novou stránku pojmenujeme „Navigace“, protože zde se bude nacházet hlavní navigační menu, zde na této stránce, také budeme zobrazovat ostatní stránky, tento postup je jednodušší než mít na každé stránce jedno menu, protože když bychom potřebovali udělat změnu, tak bychom ji museli udělav v každém okně samostatně. Také si vytvoříme stránku „DruhaStranka“ (budeme pracovat později)

Není potřeba vytvářet soubor s kódem, ten se automaticky vytvoří s novou stránkou.

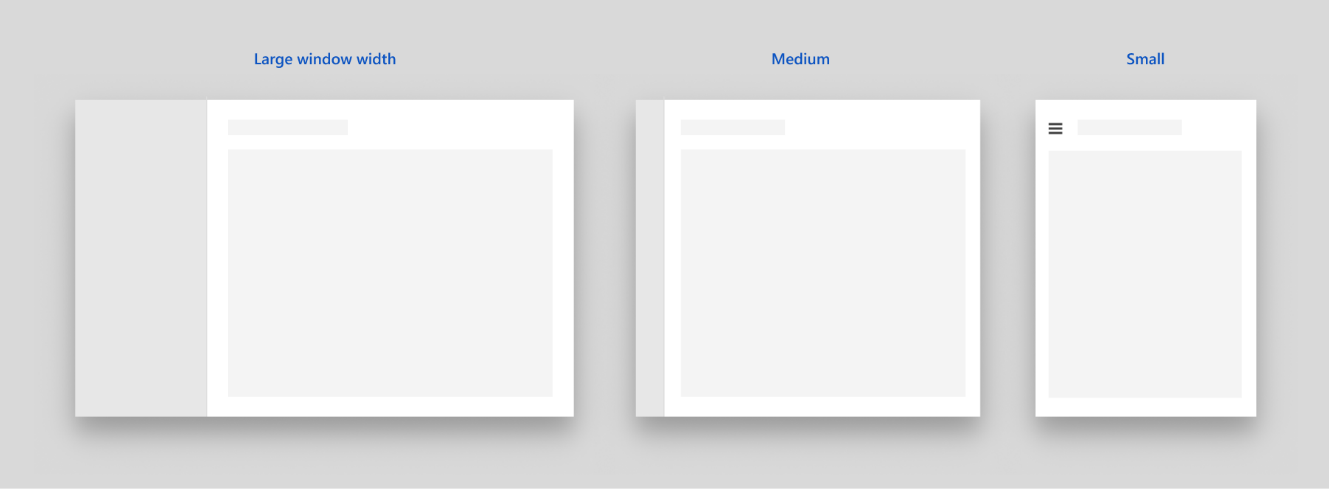
## Vytvoření a úprava navigace na stránce

O komponentech jsme se učili již v minulé hodině a nyní vložíme na stránku „Navigace“ komponentu, která se jmenuje „NavigationView“ (viz obrázek 3)

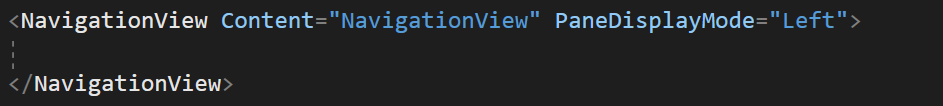


3

Nyní máme na stránce vložené navigační panel a ukážeme si, jak můžeme změnit jeho vzhled.  
Do <NavigationView Content="NavigationView">< /NavigationView> si napíšeme PaneDisplayMode=""  
mezi uvozovky můžeme napsat:

* PaneDisplayMode="Auto" Navigační liště se sama přizpůsobí velikosti okna. (viz obrázek 4)
* PaneDisplayMode="Left" Navigační lišta se bude nacházet na levé straně panelu
* PaneDisplayMode="LeftCompact" Užší navigační lišta, pokud odkazy mají ikony, ukazují se pouze ty.
* PaneDisplayMode="LeftMinimal" Úzká lišta bez ikonek.
* PaneDisplayMode="Top" Navigační lišta se nachází nahoře.

4

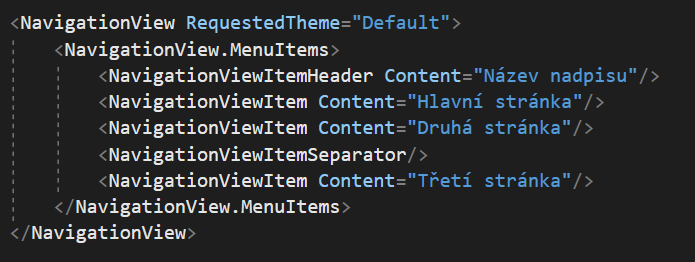
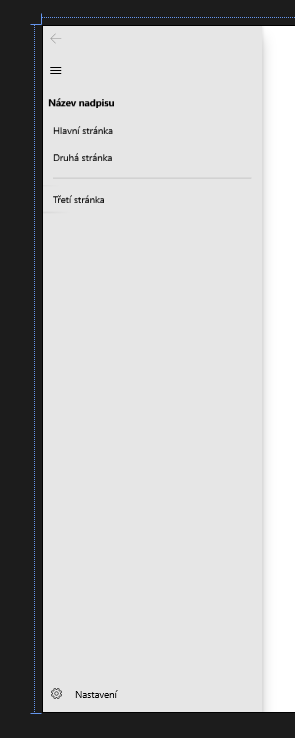
Do teď by jsme měli mít na stránce „Navigace“ toto (viz obrázek 5)

5

Abychom si mohl do navigace vložit také položky pro kliknutí, vložíme si dovnitř:

<NavigationView.MenuItems> </NavigationView.MenuItems>

Aby menu vypadalo zajímavěji můžeme si zvolit také nadpisy a máme možnost oddělní různých částí navigace.  
<NavigationViewItemHeader Content="Název nadpisu"/> Nadpisy  
<NavigationViewItem Content="Hlavní stránka"/> Zde budeme zapisovat jednotlivé odkazy v menu  
<NavigationViewItemSeparator/> Čára pro oddělení různých částí od sebe.

Zápis menu může vypadat například takto: (viz obrázky 6,7)

6

7

Abychom mohli využít zobrazení v režimu „LeftCompact“, tak si přidáme ikony.  
do <NavigationViewItem Content="Hlavní stránka"/> přidáme argument Icon="", po napsání se nám zobrazí nápověda, která nám pomůže vybrat si nespočet různých ikonek.

## Využití navigace k přecházení mezi stránkami

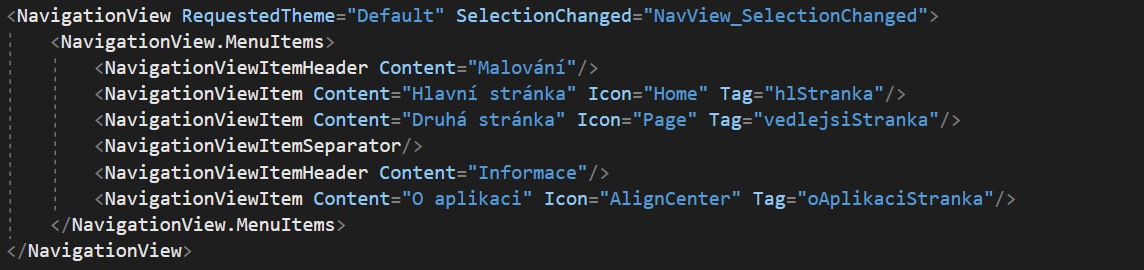
V této části se již vrhneme na trošku kódu, v první hodině jsme se seznámili se syntaxí jazyku C#  
tak teď toho využijeme.

Do NavigationView přidáme argument SelectionChanged="" po napsání zmáčkneme klávesu TAB a v kódu se nám automaticky vytvoří metoda, která se zavolá vždy, když dojde k vybrání tlačítka v navigaci.  
po správném zapsání vypadá řádek takto (viz obrázek 8)

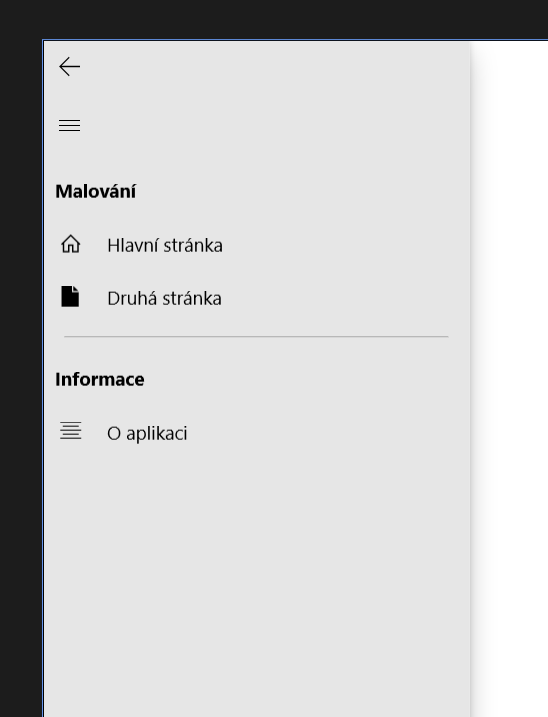
8

8

Před chviličkou jsme přidávali ikony k naším tlačítkům v navigaci, ale je také potřeba přidat Tag="",  
pomocí tagu budeme určovat na co uživatel klikl. Název pro tag použijeme krátký, výstižný a bez mezer.

Pro lepší orientaci může kód našeho menu vypadat takto: (viz obrázek 9)

9

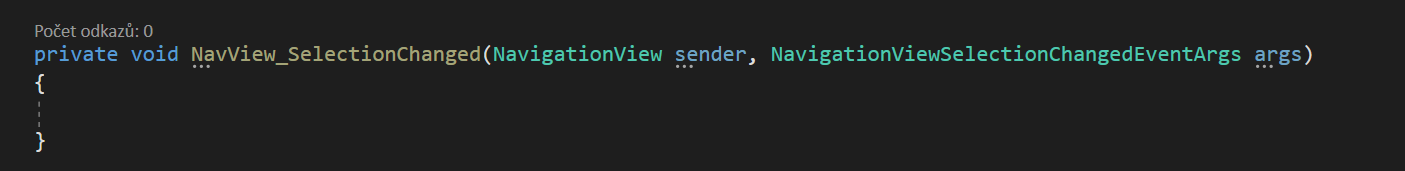


Kód výše vytvoří toto menu (viz obrázek 10)

10

Již máme vše potřebné v XAML hotovo. Nyní klikneme pravím tlačítkem myši, tam kde zapisujeme XAML kód a dáme „Zobrati kód“ (klávesová zkatka F7)

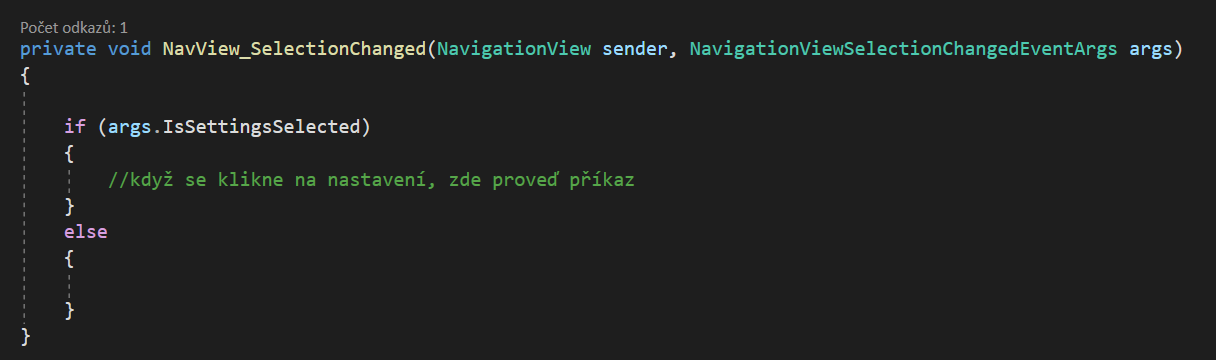
Nyní v kódu vidíme automaticky vytvořenou metodu, které se vytvořila po zadání SelectionChanged="…"  
(tato metoda se vždy zavolá, pokud uživatel vybere jakoukoliv položku v navigaci, zmíněno na začátku)  
(viz obrázek 11)



111

Pokud si přepneme zpět na Návrhář Zobrazení (klávesová zkratka Shift + F7)  
Můžeme spatřit, že dole na navigační liště se nachází automaticky tlačítko nastavení a nyní je potřeba přidat podmínku, která bude kontrolovat, zda uživatel neklikl na nastavení.

Vytvoříme podmínku if (args.IsSettingsSelected) podmínka by měla vypadat, takto (viz obrázek 12)



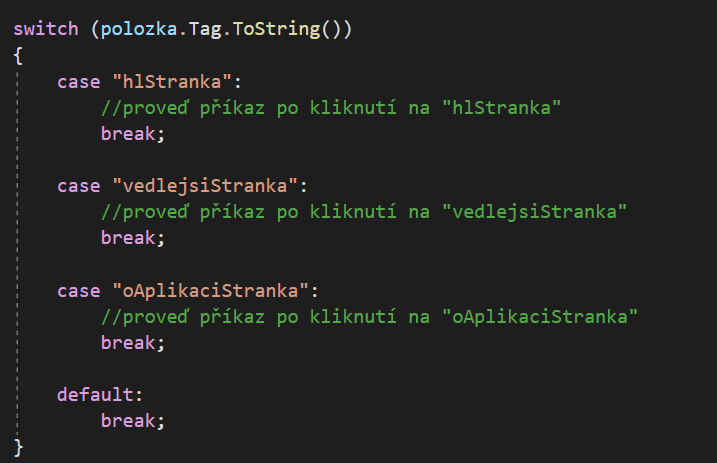
12

Pro další určení našeho navigačního menu, použijeme třídu NavigationViewItem  
která nám pomůže zjistit na co to uživatel klikl.

Do podmínky else vložíme:  
NavigationViewItem "název proměné" = args.SelectedItem as NavigationViewItem;

Nyní máme v proměnné uložené potřebné parametry, ze kterých určíme, na co uživatel klikl, k efektivnímu roztřídění použijeme podmínku typu Switch (viz obrázek 13)  
  
V našem případě jsem si proměnou pojmenovali polozka a v podmínce typu Switch nás z proměnné polozka zajímá její Tag, jestli si dobře pamatujeme, tak tag jsme právě zadávali v XAML, když jsme vytvářeli naše menu. Tag převedeme na čistý textový řetězec pomocí metody ToString()  
a pak již porovnáváme čemu se daný textový řetězec rovná.

Jak vidíme (viz obrázek 11), tak kde máme komentář typu //proveď… zde napíšeme příkaz, pro změnu stránky, jenže nemůžeme jen tak přejít na jinou stránku, protože naše navigace se nachází pouze na stránce  
"Navigace", takže po změně stránky bychom byli bez navigace.

 Naštěstí toto není problém a my každou jednotlivou stránku do naší stránky navigace „vložíme“.

13

**Jak tedy můžeme obsah stránky vložit do stránky?**

Opět si přepneme do Návrháře Zobrazení (klávesová zkratka Shift + F7)  
a do XAML kódu vložíme pod </NavigationView.MenuItems> **musí být vloženo v těle <NavigationView>**

<Frame x:Name="ObsahStranky"> pojmenujeme si, podle názvu budeme k Frame přistupovat v kódu

<Frame.ContentTransitions>

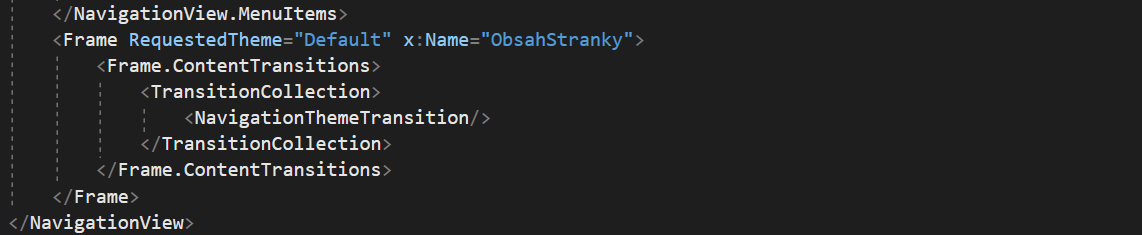
<TransitionCollection>

<NavigationThemeTransition/> umožnuje animované přechody

</TransitionCollection>

</Frame.ContentTransitions>  
</Frame>

Frame zobrazuje jednotlivé instance stránek, díky tomu že je vložen v těle NavigationView, budeme vidět navigaci na všech stránkách. XAML bude vypadat následovně: (viz obrázek 14)

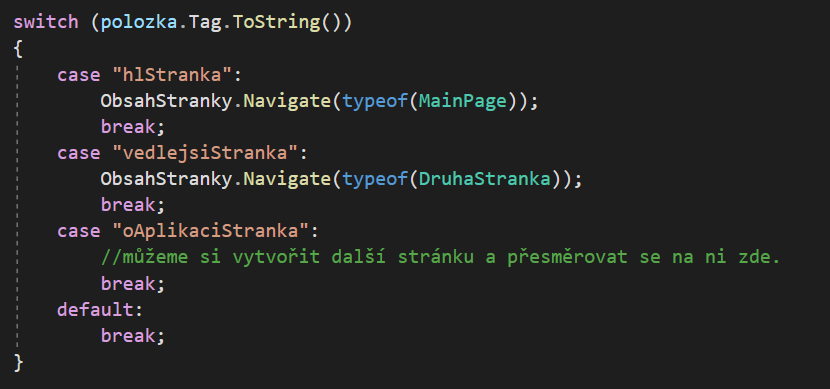


14

Znovu se vrátíme ke kódu (klávesová zkratka F7) k části, kde se nachází Switch podmínka  
do okomentovaných řádků (viz obrázek 11) vložíme:

ObsahStranky.Navigate(typeof(MainPage));

Kde ObsahStranky odpovídá názvu x:Name v XAML a MainPage odpovídá nazvu dané stránky, kterou chceme zobrazit. Kód bude nyní vypadat takto: (viz obrázek 15)



15

Nyní můžeme aplikaci spustit a vyzkoušet navigaci, jak funguje.

## Vracení se zpět, pomocí Frame.Back()

Nyní máme funkční navigaci, ale určitě jste si také všimli v levém horním rohu navigace je tlačítko zpět, takže teď si ukážeme, jak ho zprovoznit.

16Opět si zajdeme do Návrháře Zobrazení (klávesová zkratka F7)  
Do našeho NavigationView si vložíme IsBackEnabled="True" BackRequested="NavView\_BackRequested"  
řádek by měl vypadat takto: (viz obrázek 16)

16

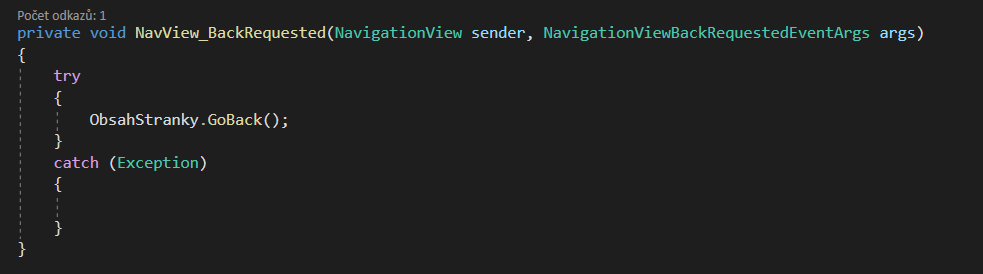
Poté co se vrátíme do C# kódu (klávesová zkratka F7), tak si můžeme všimnou, že se opět vytvořila metoda,  
ta se zavolá vždy, když na tlačítko zpět uživatel klikne.

Zde ale již musíme ošetřit vstup uživatele, protože co když uživatel klikne na tlačítko zpět, ihned po spuštění programu? V paměti není uložena žádná předchozí stránka, což **by vedlo k pádu programu.**

Proto uděláme jednoduché ošetření pomoci Try a Catch, Try se pokusí příkaz provést, ale pokud by to vedlo k chybě / pádu, tak se příkaz přeskočí a spustí se Catch.

Do Try napíšeme ObsahStranky.GoBack(), kde ObsahStranky, odpovída našemu x:Name="ObsahStranky", který jsme zadávali, když jsem v XAML přidávali <Frame>

Kód může vypadat následovně: (viz obrázek 17)



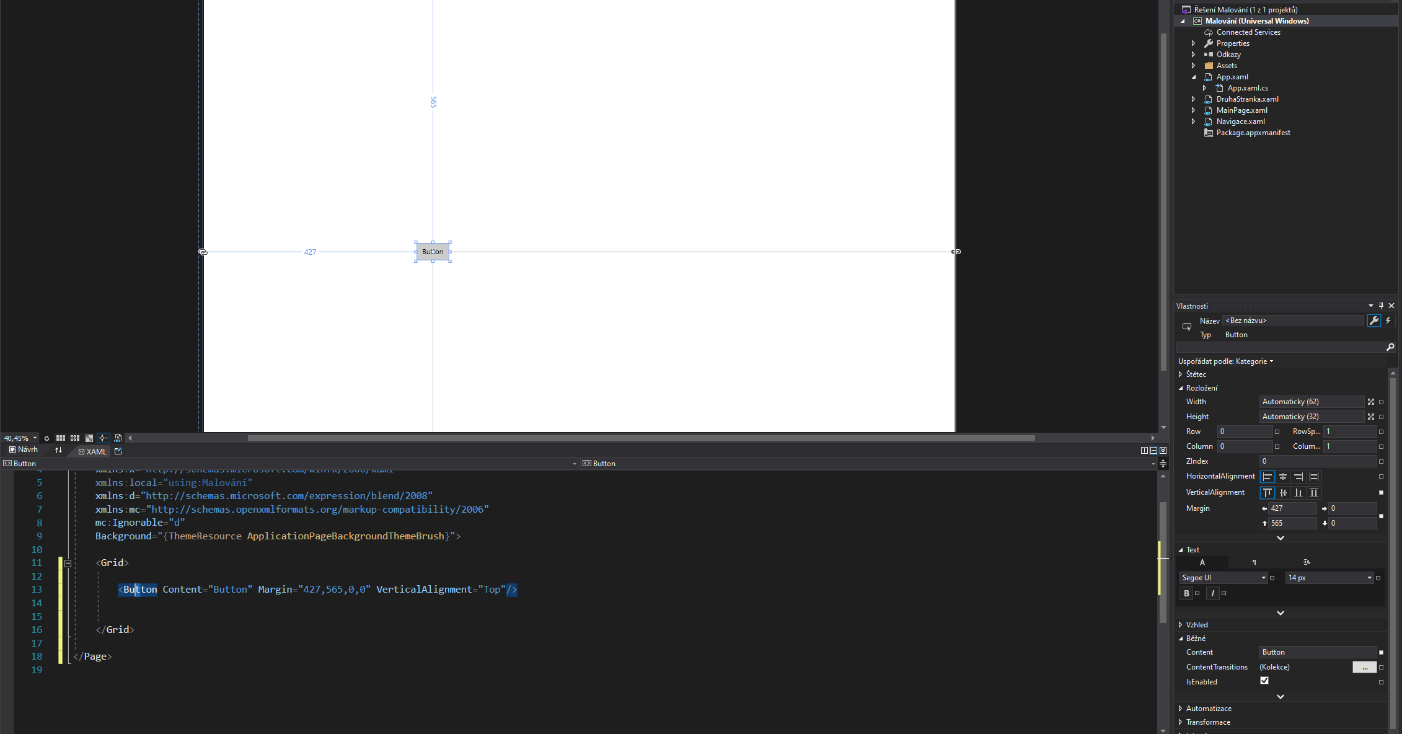
17

Opět program spustíme a jak můžeme vidět tlačítko zpět funguje.

## Reakce na tlačítka, ikony a jejich text

Před chvílí jsme zprovoznili navigační menu, ale na stránce „DruhaStranka“ nemáme nic, tak si zde můžeme ukázat, jak pomocí tlačítek volat metody a trošku si s nimi pohrát.

Zajdeme si na „DruhaStranka“ a otevřeme si Návrhář Zobrazení (klávesová zkratka F7)

V levé části si najdeme tlačítko a přetáhneme ho na obrazovku designéru

18

V XAML vidíme pozici od os X,Y a také nadpis tlačítka. (viz. obrázek 18)

V pravé části vidíme vlastnosti, které můžeme upravovat, od pozice tlačítka, po jeho vzhled, velikost písma.  
Upravíme si podle libosti, a přidáme Click="" opět můžeme dokončit klávesou TAB, sám se zvolí název metody, které se bude volat.  
Tlačítko, může vypadat takto: (viz obrázek 19)

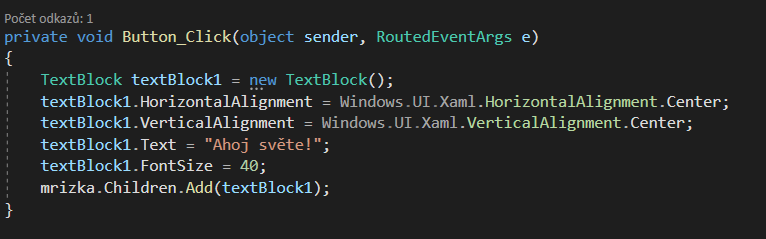
19

19

20Přidáme také jméno ke Gridu, abychom na něm mohli generovat komponenty (viz obrázek 20)

20

Níže můžete vidět jednoduchý kód, který se spustí vždy po stisknutí tlačítka a na středu obrazovky vygeneruje TextBlock s nápisem Ahoj Světe! (viz obrázek 21)



21

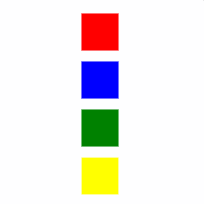
1. řádek si vytvoříme novou instanci třídy TextBlock pojmenujeme ji třeba textBlock1
2. Vycentrujeme v horizontální ose (vodorovně)
3. Vycentrujeme ve vertikální ose (svisle)
4. Nastavíme text
5. Nastavíme velikost fontu
6. Přidáme instanci do Gridu, který jsme si pojmenovali x:Name="mrizka"

Opět spustíme a vidíme že po stisknutí tlačítka se nám vypíše text.

## StackPanel a rozdíl mezi Gridem

V první hodině jsme se seznámili s tím co je to Grid a jak jednoduše rozdělit stránku na bloky.  
Nyní se podíváme na StackPanel.

Jak Grid, tak také StackPanel slouží jako kontejnery, kam vkládáme všechny jednotlivé komponenty na obrazovce. Od tlačítek, textu, až po obrázky, uživatelské vstupy, či animace.

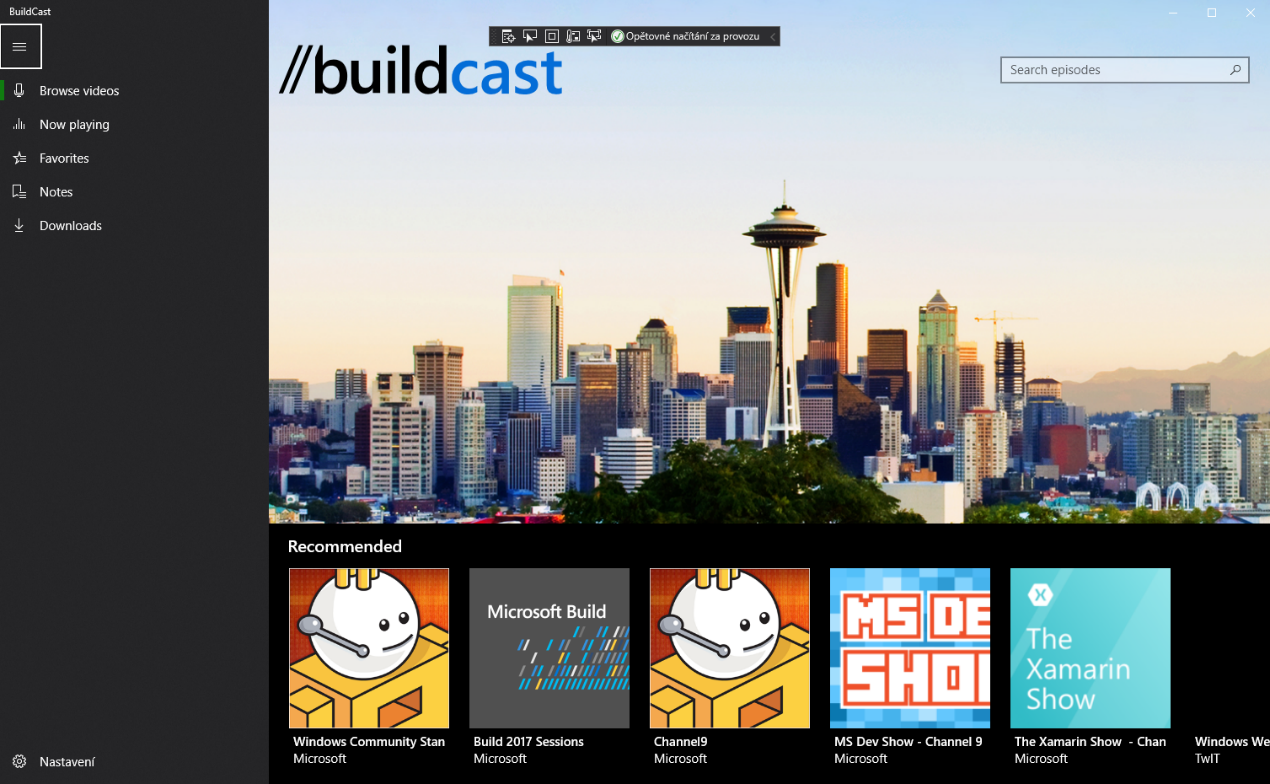
Grid, jak již napovídá název nám dovolí udělat mřížku, která se skládá z řádků a sloupců, kterým můžeme dát specifickou velikost a můžeme je spojovat dohromady.

Naproti tomu StackPanel, může být buď pouze horizontálně či vertikálně.  
(ukázka vertical StackPanel, viz obrázek 22)

StackPanel použije nejčastěji v Gridu, když potřebujeme udělat seznam položek  
které budou buď pod sebou anebo vedle sebe.  
Například slider, kde budeme ukazovat např. seznam charakterů ve hře,

22

* <https://github.com/Microsoft/BuildCast> je dobrý příklad

Níže vidíte aplikaci, kde např. můžeme StackPanel použít (viz obrázek 23)

23

## XAML výchozí styly, Light/Dark Mode

Jistě jste si všimli, že kdykoliv, když vytvoříte novou komponentu,  
tak téměř vždy je tam již argument RequestedTheme="Default"

Zde se theme (vzhled) řídí systémem / hlavním nastavením, takže pokud máme Windows v Dark mode, tak aplikace se pokusí aplikovat tento styl na danou komponentu, pokud nastavíme RequestedTheme  
na "Light" či "Dark" tak komponenta zůstane v tomto stylu, nehledě na systémové nastavení.

Jestli chceme aplikaci pouze v Light/Dark modu, tak vše můžeme specifikovat v App.xaml

* <https://docs.microsoft.com/en-us/windows/uwp/design/controls-and-patterns/xaml-theme-resources>

## XAML ResourceDictionary

Jedná se o něco jako obdobu CSS, při psaní webových stránek.

Využíváme tehdy, pokud máme hodně stejných komponentů a chceme, aby vypadali stejně, tak místo toho abychom, pro každou část nastavovali vzhled a vlastnosti samostatně, tak je nastavíme jednou a poté už jen přeřazujeme.

Představte si, že máme na stránce 6 uživatelských vstupů a chceme, aby pozadí zde bylo vždy tmavé, tak bez použití ResourceDictionary, bychom museli každý uživatelský vstup nastavit samostatně.

Je to obdoba toho, jak jsme vytvářeli navigační menu. Také jsme pro něj vytvořili samostatnou stránku, abychom po změně nemuseli nastavení měnit na každé stránce zvlášť.

Níže vidíme příklad z dokumentace, kde ResourceDictionary definujeme <Page.Resources>  
kde se bude jedna o string a název bude x:Key="greeting" a hodnota "Hello world".  
Na konci xaml kódu vidíme TextBlock, který zobrazí text Hello world. (viz obrázek 24)



24

## Závěr vzorové hodiny

Dnes jsme si ukázali, jak vytvořit navigační menu v UWP, jak používat tlačítko zpět a vracet se na předchozí stránky. Nebo již víme jak reagovat na kliknutí tlačítka, či to jaký je rozdíl mezi Gridem a StackPanelem.

Příště se podíváme na WindowsInk a Canvas a postupně se bude naše aplikace rýsovat.

## Dokumentace Microsoft

* Navigace: <https://docs.microsoft.com/en-us/windows/uwp/design/controls-and-patterns/navigationview>
* Frame: <https://docs.microsoft.com/en-us/uwp/api/windows.ui.xaml.controls.frame?view=winrt-19041>
* StackPanel: <https://docs.microsoft.com/en-us/uwp/api/windows.ui.xaml.controls.stackpanel?view=winrt-19041>
* ResourceDictionary: <https://docs.microsoft.com/en-us/windows/uwp/design/controls-and-patterns/resourcedictionary-and-xaml-resource-references>
* Demo projekty UWP na githubu: <https://github.com/microsoft/Windows-universal-samples>