**C# příručka pro učitele**

**3. Lekce – Využití nástrojů UWP, Canvas, Ink a ListBoxu**

Vytvořeno v rámci programu Microsoft Studentské Trenérské Centrum, dokončeno 29.10.2020

Autor: Adam Libra

Úvodní informace

* Ve třetí lekci se podíváme na pokročilejší funkce v rámci naší aplikace Malování. Naučíme se jak zanechat barevnou stopu na plátně a jak ji uložit. Též si ukážeme, jak a k čemu slouží ListBox.

Znalostní Prerekvizity

* Znalost objektového programování
* Znalost všeho z předchozích hodin

Technické prerekvizity

* Visual Studio s C# UWP

Probírané funkce

* WindowsInk – Canvas, Toolbar
* ListBox
* GridView, ListView
* Toast content

Očekávaný výstup

* Hotová aplikace se vší funkčností

Odhad časové dotace

* 40minut

Věková doporučení

* 14 – 99let

Obsah

[Úvod 2](#_Toc60087172)

[Využití WindowsInk 2](#_Toc60087173)

[Využití ListBox 2](#_Toc60087174)

[GridView vs ListView 4](#_Toc60087175)

[Ukládání do .png 4](#_Toc60087176)

[Notifikace 6](#_Toc60087177)

[Quiz 7](#_Toc60087178)

[Závěr 7](#_Toc60087179)

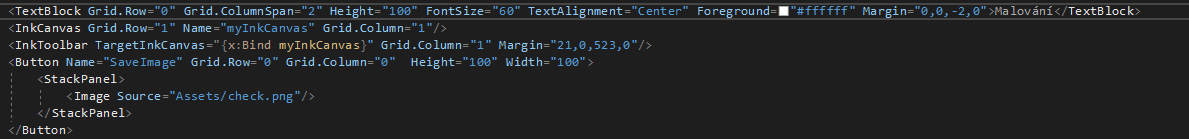
## Úvod

V dnešní hodině si dokončíme naší aplikaci. Přidáme do ní možnost kreslit a využívat tooly jako třeba zvýrazňovač nebo pravítko, přidáme možnost ukládání výtvoru a odešleme notifikaci do Windowsu.

## Využití WindowsInk

Nyní si do aplikace přidáme možnost kreslení. Pro to si nejdříve budeme muset definovat plátno, na které budeme kreslit, tzv. InkCanvas. V MainPage.xaml si definujeme <InkCanvas/> a pojmenujeme ho myInkCanvas, přidáme ho do řady a kolonky 1. Viz. obr.1.1


1

Jako další si přidáme nástroje jako je pero, zvýrazňovač nebo guma. K tomu využijeme třídy <InkToolBar/>. Tu si umístíme do kolonky 1 a spojíme ho s naším plátnem za pomocí atributu TargetInkCanvas="{x:Bind myInkCanvas}". Dáme si pozor, abychom měli tyto dvě třídy umístěny nad buttonem, jelikož jinak by nám tlačítko nefungovalo. Výsledný kód by tedy měl vypadat takto: 

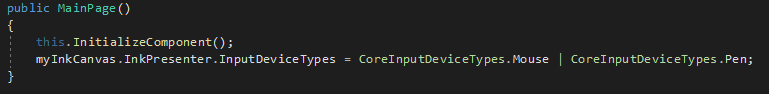
2

Nyní se podíváme na funkční stránku. V souboru MainPage.xaml.cs. v MainPage() si zprovozníme naše kreslící nástroje za pomocí dvou řádků,

this.InitializeComponent();

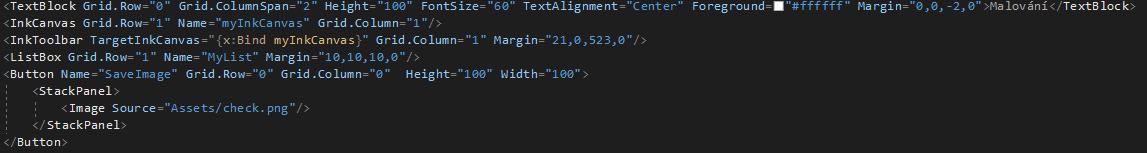
myInkCanvas.InkPresenter.InputDeviceTypes = CoreInputDeviceTypes.Mouse | CoreInputDeviceTypes.Pen;

viz. obr. 3



3

## Využití ListBox

ListBox je prvek umožňující zobrazení seznamu položek a jeho dynamické upravování. V našem MainPage.xaml si přidáme položku <ListBox/> do řady a kolonky 0. Pojmenuje me si ho Name="MyList". Náš výsledný kód by tedy měl vypadat takto: 

4

ListBoxu využijeme při přidávání další funkce, funkce pro rozpoznávání textu, který nakreslíme. V C# kódu ( MainPage.xaml.cs ) si vytvoříme novou metodu, pojmenujeme si ji InkPresenter\_StrokesCollected a do jejího parametru vpíšeme InkPresenter sender, InkStrokesCollectedEventArgs args , čímž si zpřístupníme možnosti zpracování dat z InkCanvas.

Vždy při vyvolání této metody si nejdříve vyčistíme náš ListBox, proto napíšeme MyList.Items.Clear(); Dále si přidáme funkci pro rozpoznávání nakreslených symbolů a zapsání výsledku do ListBoxu.

InkRecognizerContainer inkRecognizerContainer = new InkRecognizerContainer();

if (inkRecognizerContainer != null)

{

IReadOnlyList<InkRecognitionResult> recognitionResults = await inkRecognizerContainer.RecognizeAsync(myInkCanvas.InkPresenter.StrokeContainer, InkRecognitionTarget.All);

if (recognitionResults.Count > 0)

foreach (var result in recognitionResults)

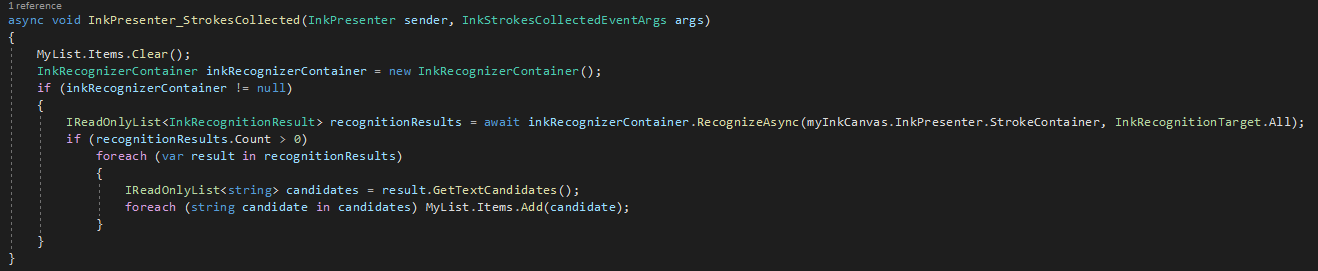
{

IReadOnlyList<string> candidates = result.GetTextCandidates();

foreach (string candidate in candidates) MyList.Items.Add(candidate);

}

}

Celá metoda by tedy měla vypadat takto: 

5

Metodu teď použijeme v MainPage() objektu.

myInkCanvas.InkPresenter.StrokesCollected += InkPresenter\_StrokesCollected;

## GridView vs ListView

ListView i GridView jsou způsoby organizace UI ( user interface ) aplikace. ListView skládá data do sloupce, například list zpráv. Pro uspořádání ve více sloupcích nebo v tabulce je lepší používat GridView.

## Ukládání do .png

Další metodu, kterou si vytvoříme, bude metoda pro ukládání výtvoru do .png obrázku. Metodu si pojmenujeme Button\_Click(), protože bude vyvolávána kliknutím na tlačítko. Tato metoda bude mít několik důležitých podmínek.

1. Podmínku, jež nám zkontroluje zda je na plátně něco nakresleného
2. Podmínku, jež nám zkontraluje, zda ukládaný obrázek nemá velokost 0

Po těchto dvou podmínkách se nám obrázek uloží do .png. V parametrech metody bude object sender, RoutedEventArgs e , díky kterým metoda získá informace o Canvasu.

První věc, kterou musíme v naší metodě udělat, je získat informace o tazích po plátně a uložit je do proměnné. Toho docílíme řádkem IReadOnlyList<InkStroke> currentStrokes = myInkCanvas.InkPresenter.StrokeContainer.GetStrokes();. Dále vytvoříme naši první podmínku, ve které zkontrolujeme jestli plátno není prázdné. V této podmínce vytvoříme proměnnou třídy FileSavePicker, jež nám umožní vybrat umístění, koncovku a jméno obrázku. Tato část kódu bude vypadat následovně: ( obr. 6. )

if (currentStrokes.Count > 0)

{

FileSavePicker savePicker = new FileSavePicker()

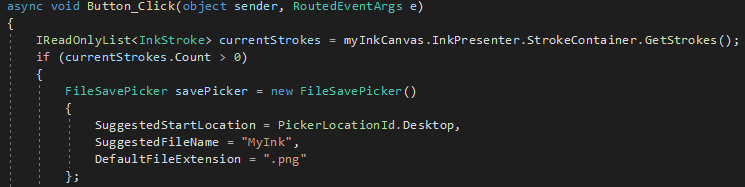
{

SuggestedStartLocation = PickerLocationId.Desktop,

SuggestedFileName = "MyInk",

DefaultFileExtension = ".png"

};



6

Dále si přidáme koncovku .png i do výběru při ukládání a uložíme si místo uložení obrázku do proměnné.

savePicker.FileTypeChoices.Add("PNG (\*.png)", new List<string>() { ".png" });

StorageFile file = await savePicker.PickSaveFileAsync();

Nyní si vytvoříme další podmínku, která nám zkontroluje, aby velikost obrázku nebyla 0. V ní dále soubor uložíme a počkáme na odezvu systému, jestli soubor řádně uložil.

if (file != null)

{

CachedFileManager.DeferUpdates(file);

IRandomAccessStream stream = await file.OpenAsync(FileAccessMode.ReadWrite);

using (IOutputStream outputStream = stream.GetOutputStreamAt(0))

{

await myInkCanvas.InkPresenter.StrokeContainer.SaveAsync(outputStream);

await outputStream.FlushAsync();

}

stream.Dispose();

Windows.Storage.Provider.FileUpdateStatus status = await CachedFileManager.CompleteUpdatesAsync(file);

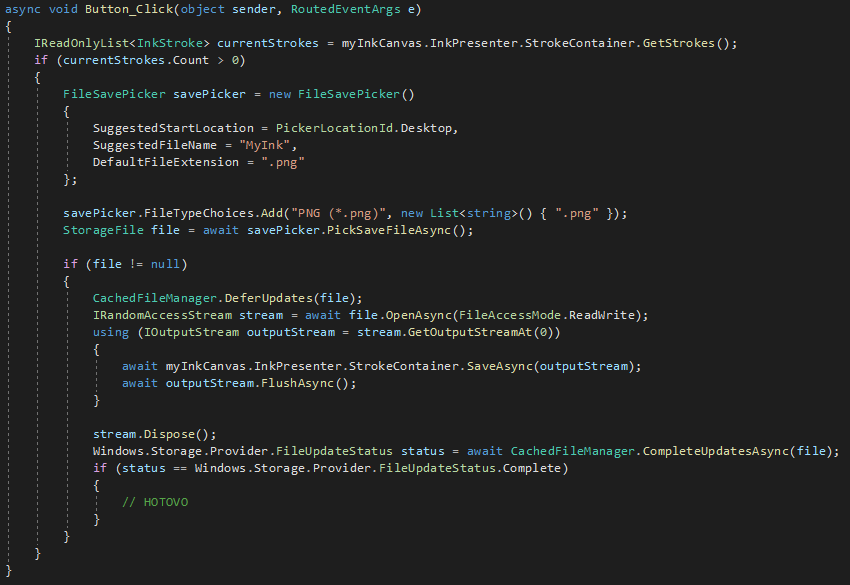
if (status == Windows.Storage.Provider.FileUpdateStatus.Complete)

{

// HOTOVO

}

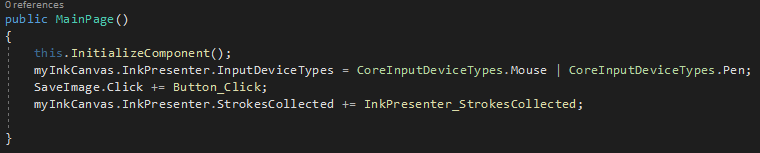
}

Celý kód metody tedy bude vypadat následovně:

7

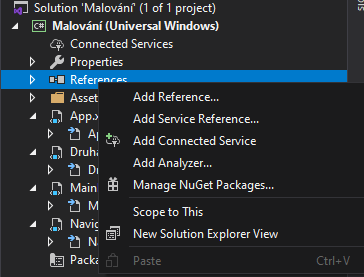
Nyní metody využijeme v MainPage().

SaveImage.Click += Button\_Click;

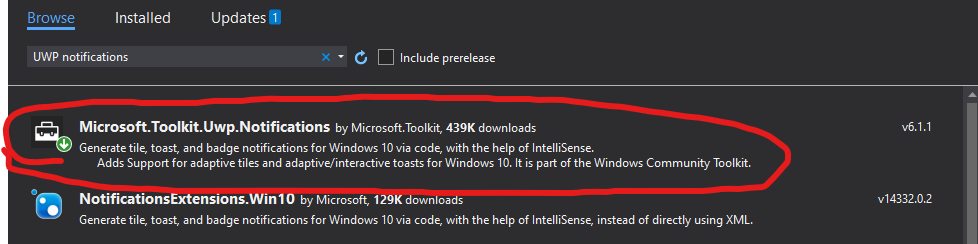
Náš object MainPage() by tedy měl vypadat následovně:

8

## Notifikace

Pro používání notifikací si budeme muset stáhnout NuGet knihovnu Toolkit. Vpravo klikneme pravým myšítkem na „References“ a zvolíme „Manage Nuget Packages...“ ( viz. obr. 9 )

9

Následně v poli „Browse“ vyhledáme „UWP Notifications“ a nainstalujeme hned první výsledek. ( obr. 10 )

10

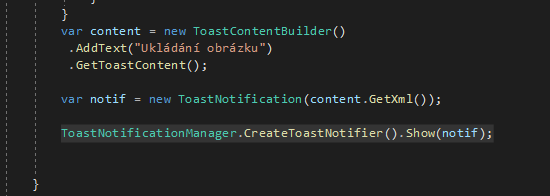
Nyní můžeme přidat notifikace i do našeho programu. Budeme chtít získat notifikaci vždy po uložení obrázku, proto kód přidáme do metody Button\_Click(). Notifikaci vytvoříme pomocí dvou proměnných. První si pojmenujeme content, přidáme ji hodnotu new ToastContentBuilder() a přidáme jí text s pomocí

.AddText("Ukládání obrázku")

.GetToastContent();

Druhá proměnná bude notif, které přiřadíme honotu new ToastNotification(content.GetXml());

A následně odešleme notifikaci systému ToastNotificationManager.CreateToastNotifier().Show(notif);



11

## Quiz

1. Kdybychom chtěli vytvořit záznam dat v podobě tabulky, jakého uspořádání buchom využili? – GridView
2. V jakém uspořádání zobrazujeme v naší aplikaci rozpoznání textu? - ListBox

## Závěr

Dnes jsme si dokončili naší aplikaci malování. Naučili jsme se využívat WindowsInk, ListBox a vysvětlili si rozdíl mezi GridView a ListView.

Dokumentace:

* [System.Windows.Ink Obor názvů | Microsoft Docs](https://docs.microsoft.com/cs-cz/dotnet/api/system.windows.ink?view=net-5.0)
* [ListBox Třída (System.Windows.Forms) | Microsoft Docs](https://docs.microsoft.com/cs-cz/dotnet/api/system.windows.forms.listbox?view=net-5.0)
* [FileSavePicker Class (Windows.Storage.Pickers) - Windows UWP applications | Microsoft Docs](https://docs.microsoft.com/en-us/uwp/api/windows.storage.pickers.filesavepicker?view=winrt-19041)
* [FileSavePicker Constructor (Windows.Storage.Pickers) - Windows UWP applications | Microsoft Docs](https://docs.microsoft.com/en-us/uwp/api/windows.storage.pickers.filesavepicker.-ctor?view=winrt-19041)