

Demonstrace možností využití knihovny Windows UI Library

Lukáš Kobylík

Bakalářská práce
2020



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Lukáš Kobylík**
Osobní číslo: **A16350**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Softwarové inženýrství**
Forma studia: **Prezenční**
Téma práce: **Demonstrace možností využití knihovny Windows UI Library**
Téma práce anglicky: **Illustrating Windows UI Library – Possibilities of Use**

Zásady pro vypracování

1. Popište technologie vývoje desktopových aplikací pro operační systém Windows.
2. Prozkoumejte a popište knihovnu WinUI a obsažené komponenty.
3. Zhodnoťte způsoby použití knihovny WinUI se staršími technologiemi tvorby desktopových aplikací a různými programovacími jazyky jako C++ a C#.
4. Navrhněte ukázkovou aplikaci s využitím knihovny WinUI demonstrující její nové vlastnosti.
5. Vytvořte a popište vizuálně řešenou demonstrující klíčové prvky navržené aplikace.
6. Demonstrujte výsledky.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. NAGEL, Christian. Professional C# 7 and .NET Core 2.0. 11th edition. Indianapolis: Wrox, a Wiley Brand, 2018. ISBN 978-1119449270.
2. CHATTERJEE, Ayan. Building apps for the universal Windows platform: explore Windows 10 Native, IoT, HoloLens, and Xamarin. Berkeley, California: Apress, [2017]. ISBN 978-1484226285.
3. PETZOLD, Charles. Mistrovství ve Windows Presentation Foundation: [aplikace = kód + markup]. Brno: Computer Press, 2008. Mistrovství. ISBN 978-80-251-2141-2.
4. GitHub – microsoft/microsoft-ui-xaml: Windows UI Library: the latest Windows 10 native controls and Fluent styles for your applications. The world's leading software development platform - GitHub [online]. GitHub, Inc. [cit. 20.11.2019]. Dostupné z: <https://github.com/microsoft/microsoft-ui-xaml>
5. WinUI 3.0 Alpha (November 2019) – Windows UWP applications | Microsoft Docs. [online]. Dostupné z: <https://docs.microsoft.com/en-us/uwp/toolkits/winui3/#xaml-controls-gallery-winui-30-alpha-branch>
6. WinUI on Windows 7 – Yes, it's possible with Uno Platform – Uno Platform. Uno Platform – Home [online]. Dostupné z: <https://platform.uno/WinUI-on-Windows7-via-UnoPlatform/>

Vedoucí bakalářské práce:

Ing. Erik Král, Ph.D.
Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce: 28. listopadu 2019
Termín odevzdání bakalářské práce: 15. května 2020



doc. Mgr. Milan Adámek, Ph.D.
děkan

prof. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

Ve Zlíně dne 9. prosince 2019

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 10. 8. 2020

Lukáš Kobylík, v. k.

ABSTRAKT

V teoretické části se práce zabývá řešením současných technologií pro tvorbu desktopových aplikací a možnostmi tvorby uživatelského prostředí s moderními prvky a funkcemi. V praktické části se práce věnuje návrhu jednoduchých aplikací za využití různých technologií a programovacích jazyků, ve kterých bude využito moderních UWP elementů a knihovny WinUI.

Klíčová slova: WinUI, C#, C++, UI, uživatelské prostředí

ABSTRACT

The theoretical part contains recherche about current technologies for creating desktop applications and the possibilities of creating a user interface with modern elements and functions. The practical part deals with the design of simple applications using various technologies and programming languages, which will use modern UWP elements and the WinUI library.

Keywords: WinUI, C#, C++, UI, user interface

Děkuji Ing. Eriku Královi, Ph.D. za cenné rady a čas, který mi věnoval při vedení mé bakalářské práce.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

| | |
|---|-----------|
| ÚVOD | 9 |
| I TEORETICKÁ ČÁST | 10 |
| 1 TECHNOLOGIE VÝVOJE DESKTOPOVÝCH APLIKACÍ | 11 |
| 1.1 TECHNOLOGIE VÝVOJE DESKTOPOVÝCH APLIKACÍ | 11 |
| 2 PRŮZKUM GRAFICKÉHO PROSTŘEDÍ | 14 |
| 2.1 DEFINICE GRAFICKÉHO PROSTŘEDÍ..... | 14 |
| 2.1.1 Textové uživatelské rozhraní | 15 |
| 2.1.2 Grafické uživatelské rozhraní..... | 16 |
| 2.1.3 Ovládání uživatelského prostředí | 17 |
| 2.2 VÝVOJ MODERNÍHO PROSTŘEDÍ..... | 17 |
| 2.3 MICROSOFT FLUENT DESIGN SYSTEM..... | 21 |
| 2.3.1 Nasazení Fluent Designu..... | 23 |
| 2.3.2 Nekonzistence UI | 25 |
| 2.4 KNIHOVNA WINUI | 28 |
| 2.4.1 Knihovna WinUI 2.0..... | 29 |
| 2.4.2 Knihovna WinUI 2.1 | 32 |
| 2.4.3 Knihovna WinUI 2.2 | 33 |
| 2.4.4 Knihovna WinUI 2.3 | 36 |
| 2.4.5 Knihovna WinUI 3.0..... | 37 |
| 2.4.6 Současné limitace knihovny WinUI 3.0..... | 38 |
| 2.5 XAML ISLANDS..... | 39 |
| 2.6 XAML CONTROL GALLERY..... | 40 |
| 2.7 WINUI A STARŠÍ TECHNOLOGIE TVORBY DESKTOPOVÝCH APLIKACÍ A RŮZNÉ PROGRAMOVACÍ JAZYKY | 42 |
| II PRAKTICKÁ ČÁST | 43 |
| 3 ÚVOD DO PRAKTICKÉ ČÁSTI | 44 |
| 4 PŘÍPRAVA KNIHOVNY WINUI 3.0 | 45 |
| 4.1 WINUI 3.0 PRO DESKTOPOVÉ APLIKACE | 45 |
| 5 PŘÍPRAVA | 46 |
| 6 APLIKACE UWP S WINUI 3.0 | 48 |
| 6.1 XAML UWP APLIKACE | 49 |
| 6.2 C# KÓD UWP APLIKACE..... | 57 |
| 6.3 POPIS VÝSLEDNÉ APLIKACE..... | 59 |
| 7 KLASICKÁ DESKTOPOVÁ APLIKACE V C# | 60 |
| 7.1 POPIS VÝSLEDNÉ APLIKACE..... | 61 |
| 8 KLASICKÁ DESKTOPOVÁ APLIKACE V C++ | 63 |

| | | |
|---|--------------------------------------|-----------|
| 8.1 | XAML DESKTOPOVÉ APLIKACE V C++ | 65 |
| 8.2 | C++ KÓD DESKTOPOVÉ APLIKACE | 65 |
| 8.3 | POPIS VÝSLEDNÉ APLIKACE | 67 |
| ZÁVĚR | | 69 |
| SEZNAM POUŽITÉ LITERATURY | | 70 |
| SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK | | 74 |
| SEZNAM OBRÁZKŮ | | 75 |
| SEZNAM TABULEK | | 77 |
| SEZNAM PŘÍLOH | | 78 |

ÚVOD

Již několik desítek let se staly počítače nedílnou našich životů. Za tu dobu prošlo také jejich ovládání značným vývojem. Od velkých sálových počítačů na děrné štítky, po počítače osobní. Ty zprvu nebyly zdaleka tak pohodlné na ovládání, jako je tomu dnes. Ovládání počítače skrze příkazový řádek bylo později nahrazeno grafickým uživatelským prostředím. Uživatelské prostředí je to, v čem se uživatel pohybuje. Skrze něj se ovládá počítač. A proto je velmi důležité, aby se v něm uživatel vyznal a orientoval. Aby pro něj bylo intuitivní.

Každé uživatelské prostředí má své výhody a nevýhody a disponuje různými prvky a elementy, aby byl přívětivější pro uživatele. Přesto se najdou také určité nedostatky.

Tato práce má sloužit k tomu, aby se v teoretické části porovnaly možnosti tvoření desktopových aplikací a možnosti knihovny WinUI. Také však proto, aby se ukázalo na nedostatky jednotlivých řešení.

V praktické části bakalářské práce je tvorba ukázkových aplikací za použití WinUI 3.0, která je vytvořena pomocí IDE Microsoft Visual Studio. Jednotlivé návrhy jsou napsány za využití technologií C# a C++. Ty jsou dále tvořeny, jako klasická desktopová aplikace či aplikace univerzální.

I. TEORETICKÁ ČÁST

1 TECHNOLOGIE VÝVOJE DESKTOPOVÝCH APLIKACÍ

1.1 Technologie vývoje desktopových aplikací

1.1.1 Tvorba desktopových aplikací

Pro tvorbu nativních desktopových aplikací je možno vybrat z několika dostupných možností.

V dnešní době je na výběr pro vytvoření desktopové aplikace z více různých platform. Každá z nich nabízí své unikátní možnosti a hodí se pro různé případy užití a zaměření.

- Universal Windows Platform (UWP)
- WPF (.NET)
- Windows Forms (.NET)
- Win32

WPF

Jedná se framework z roku 2006. WPF používá pro tvorbu uživatelského prostředí jazyk Extensible Application Markup Language (XAML). WPF zpětně funguje až do systému Windows XP. Je zároveň plně zpětně kompatibilní se staršími Windows Forms. Podporuje také MVVM.

WPF je založena na značkovacím jazyku XAML. Můžeme psát v C # nebo VB.NET. WPF je vhodnou volbou pro velké aplikace. [1]

Windows Forms

Windows Forms, nebo také WinForms, je framework, který umožňuje tvorbu formulářových aplikací. Tvorba grafického prostředí je usnadněna grafickým designérem. Framework je založen na .NETU.

WinForms se spoléhá na drag and drop rozhraní Visual Studio Designeru. Proto je také velmi jednoduchý pro vytvoření uživatelského prostředí a jeho následné úpravy. Windows Form byl poté nahrazen WPF. [2]

Win32

Win32 API. Rozhraní Win32 API (také nazývané Windows API) je původní platforma pro nativní aplikace Windows C/C ++. Většinu rozhraní API Win32 obsahuje MFC (Microsoft

Foundation Library). MFC poté dodává knihovny s klíčovými elementy. Okna, tlačítka či písma. [3]

UWP

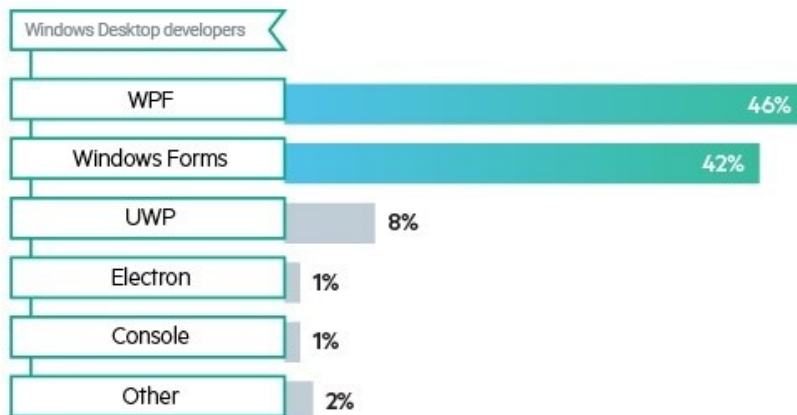
UWP je nejmodernější nativní způsob vývoje aplikací pro Windows. Cílí však pouze na Windows 10. Tato platforma umožňuje vytvořit aplikaci, která sdílí společný kód, API a aplikační model pro všechna zařízení, která používají jádro Windows 10. UWP používá Windows Runtime. WinRT je nativní API poskytované operačním systémem. Je založen na XAML, jako WPF, a je možno je psát také v C#, Visual Basic, C ++. DirectX UI a C ++. JavaScript a HTML. Aplikace UWP obsahují také moderní elementy a funkce. Aplikace deklaruje v manifestu přístupy k zařízením, jako je například kamera nebo mikrofon. Musí dát nejdříve vědět uživateli o žádosti k přístupu. UWP lze tedy označit za bezpečnější. [4]
[5]

Electron

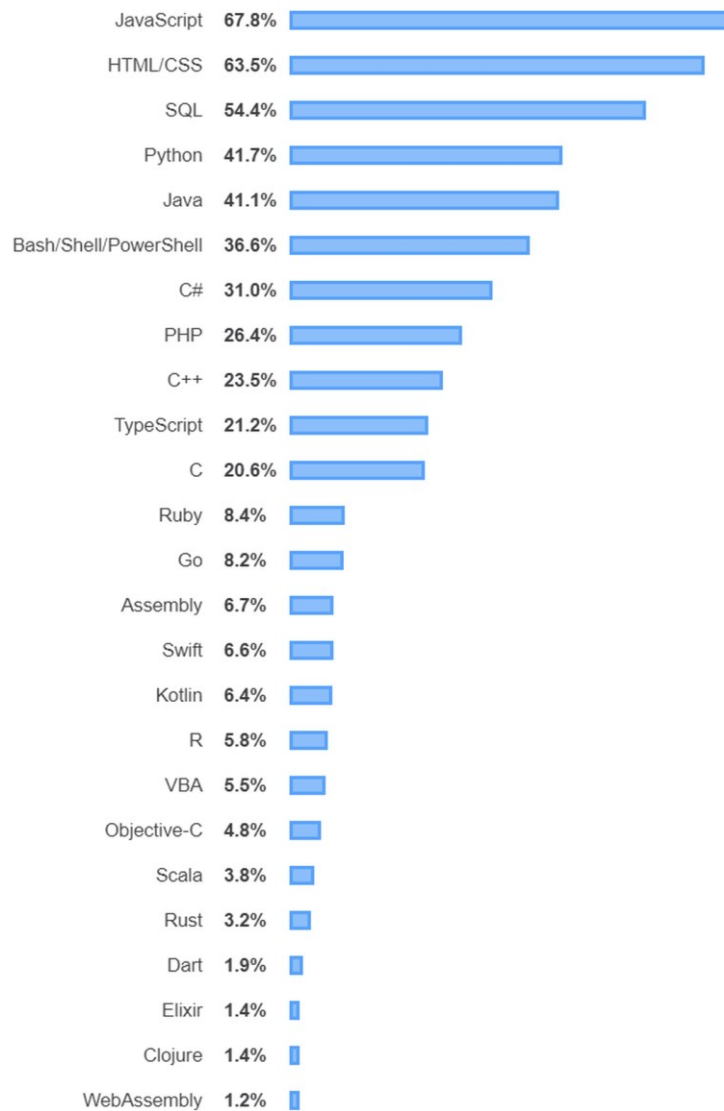
Electron umožňuje vývoj desktopových aplikací s webovými technologiemi. Electron používá Node.js a Chromium k vytvoření webového zobrazení v klasickém okně plochy. Electron je navíc multiplatformní. Funguje tak také na Windows, Mac OS X i Linuxu. Pouze na Windows 10 však mohou Electron aplikace využívat nových API, jako je Cortana a notifikace. [6]

Dle průzkumu Stack Overflow Survey z roku 2020 [8] zobrazeného na Obrázku 2 je nejpopulárnějším jazykem JavaScript. C# je pak na sedmé pozici, následováno PHP a C++. Z průzkumu server Telerik z roku 2016 [7] pak vyplývá, že by většina vývojářů volila tvorbu WPF aplikací. Windows Forms jsou na druhém místě a UWP pak na třetím, což je vidět na obrázku číslo 1.

What Technology Would You Choose if Building for Windows Desktop?



Obrázek 1 – Popularita technologií pro tvorbu aplikací pro Windows 2018 [7]



Obrázek 2 – Popularita programovacích jazyků 2019 [8]

2 PRŮZKUM GRAFICKÉHO PROSTŘEDÍ

2.1 Definice grafického prostředí

Na otázku, zda bude nový software či operační systém úspěšný nelze z daleka tak snadno odpovědět. Výsledek závisí na mnoha aspektech. Nejde pouze o kód, který se stará o celkový běh a funkce systému či aplikace. Velmi důležitou roli zde hraje zdánlivě mnohem méně významnější prvek a tím je uživatelské prostředí.

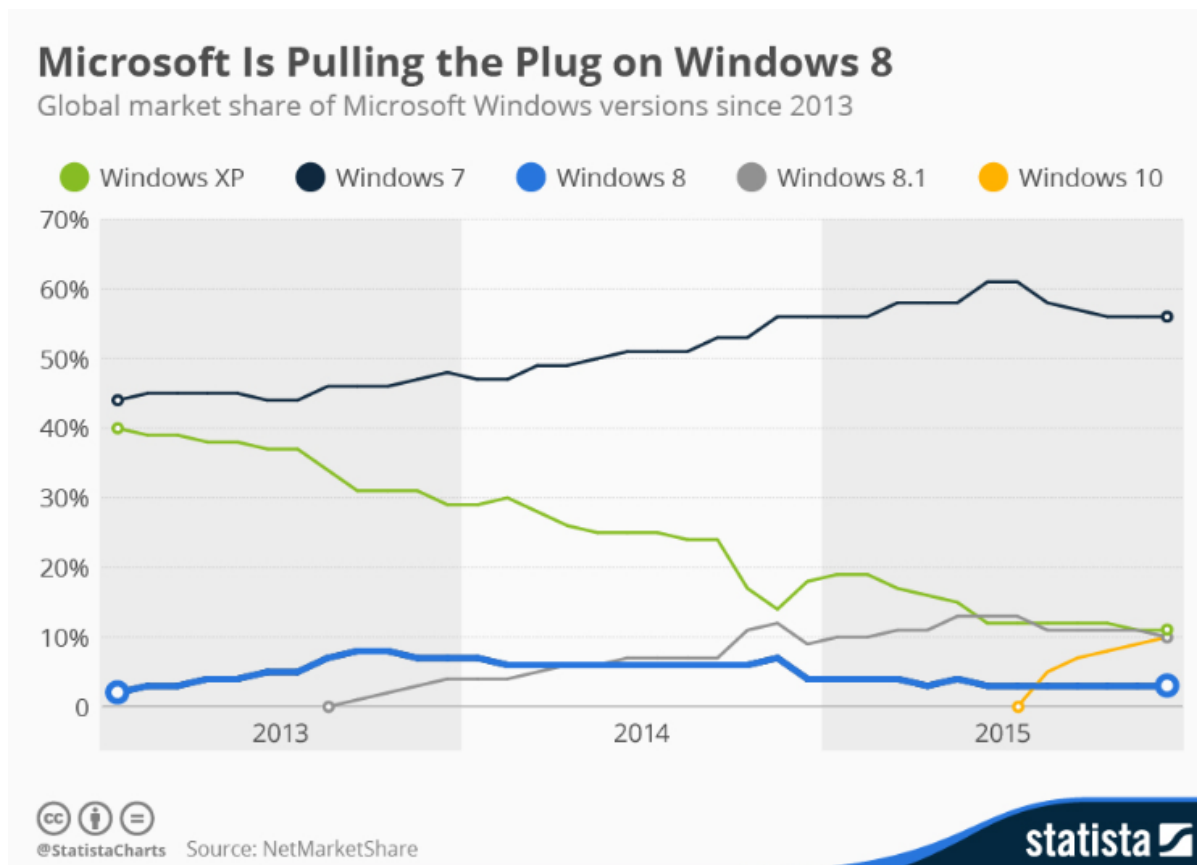
Běžný uživatel samozřejmě ocení funkce či různá vylepšení, ovšem je to právě uživatelské prostředí (Dále jen UI), které je uživateli vždy na očích. Se správně navrženým UI poté úzce souvisí uživatelský zážitek (Dále jen UX). Kladme si proto otázky.

- Je UI dobře navrženo?
- Je přehledné?
- Jsou jednotlivé prvky a jejich funkce pochopitelné?
- Působí příjemně a zapadá do prostředí systému?
- Vytváří celkový dojem příjemné pracovní UI pro uživatele?

Pokud bychom na tyto otázky odpověděli negativně, mohlo by mít UI velmi negativní efekt na produktivitu uživatele. Následná nespokojenost by se snadno promítla do celkového neúspěchu systému či našeho softwaru, ač by jeho kód byl velmi dobře navržen.

V retrospektivě můžeme použít jako příklad Microsoft Windows 8 z roku 2012. Windows 8 byl revoluční systém, který se pokusil sjednotit dotykové UI pro tablety a UI pro klasické ovládání myší a klávesnicí. Od této kombinace si Microsoft sliboval příchod nových dotykových zařízení Windows. Microsoft totiž reagoval na stoupající popularitu ultrabooků. Zároveň se však jednalo o odpověď Microsoftu na iPady od Applu. Kombinace dvou prostředí přinášela jak možnost snadné konzumace obsahu pomocí dotykového prostředí pro jednoduché ovládání prsty a zároveň produktivní režim klasické pracovní plochy. [9] Výrobci notebooků na příchod nového systému reagovali také novými form factory, které se vyrábí dodnes i se systémem Windows 10. Jedná se o různé otočné či oddělitelné konstrukce, pomocí kterých lze snadno přejít do tabletového režimu. Microsoft také uvedl na trh své zařízení Microsoft Surface, které z velké části posloužilo, jako inspirace pro nová zařízení s Windows. [10]

Uživatelé běžných nedotykových stanic však toto řešení příliš nepřijali. Velké prvky a celoobrazkový Start mohly být pro uživatele s myší a klávesnicí spíše limitující. UI zobrazovalo mnohem méně informací a velké prvky znamenaly větší pohyby myši. Systém nebyl přijat příliš kladně a nepřekonal počet strojů s Windows 7 ani Windows XP [11], jako můžeme vidět na obrázku 3.



Obrázek 3 – Tržní podíl operačních systémů 2013–2015 [11]

2.1.1 Textové uživatelské rozhraní

Textové rozhraní je v podstatě nejstarším způsobem interakce mezi obrazovkou počítače a uživatelem. Textové uživatelské rozhraní pracuje v takzvaném textovém režimu. Jedním z nejznámějších příkladů textového režimu je příkazová řádka. Počítač s uživatelem komunikuje pouze skrze vypsání textové informace a jako vstup očekává speciální příkazy. Takovým uživatelským rozhraním disponoval například operační systém MS-DOS. Později však MS-DOS získal grafické uživatelské rozhraní v podobě Windows 1.0 či jiných aplikací, které sice pracovaly v textovém režimu, ale snažily se jej přetvořit do grafického rozvržení.

I textové uživatelské rozhraní tedy může být obohacena o uživatelské rozhraní, které v určité podobě známe dodnes. [12]

Operační systémy od Windows 1.0 až Windows 98 byly, co se týče rozhraní, pouhou grafickou nástavbou operačního systému MS-DOS [12]. Ukázka prostředí z MS-DOS je vyobrazena na obrázku 4.

```
Starting MS-DOS...

Current date is Tue 06-09-2009
Enter new date (mm-dd-yy):
Current time is 5:47:40.20p
Enter new time:

Microsoft(R) MS-DOS(R) Version 6.30
(C)Copyright Microsoft Corp 1981-1995.

A:\>ver /r

MS-DOS Version 6.30
Revision A
DOS is in low memory

A:\>
```

Obrázek 4 – Textové uživatelské prostředí DOS [12]

2.1.2 Grafické uživatelské rozhraní

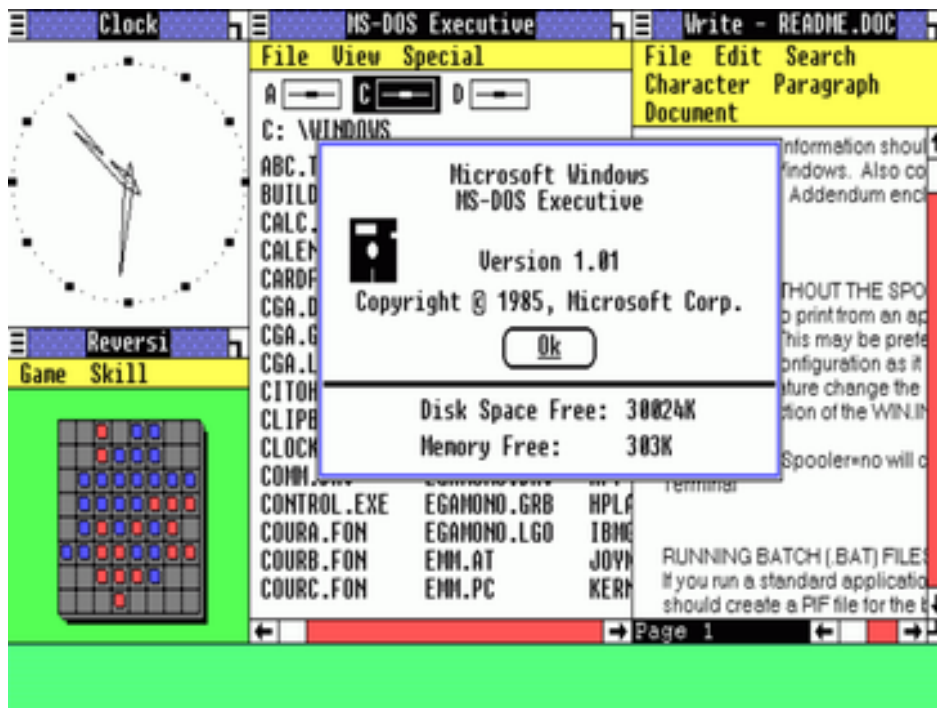
S grafickým UI se stává práce s počítačem mnohem jednodušší a intuitivnější. Zatímco počítače s textovým režimem sloužily převážně odborníkům, kteří příkazy znali, grafické UI umožnilo používání počítače i běžnému uživateli. Ten se tak nemusí učit složité příkazy. Grafické UI se stává jakýmsi prostředím, ve kterém se uživatel pohybuje. Počítač je ovládán klávesnicí a myší a jejím klikáním na jednotlivé prvky v UI.

V minulosti byla k dispozici v daném okamžiku vždy pouze jedna aplikace na obrazovce. Uživatel tak musel přepínat mezi jednotlivými aplikacemi. Tímto se vyznačovaly právě operační systémy v textovém režimu, například operační systém DOS.

Takzvaný okenní uživatelský režim se začal používat až později. Ten přišel ruku v ruce s UI operačním systémem. Tento grafický režim nabízel také Xerox či Apple. Microsoft se jej

v plné podobě implementoval až ve Windows ve verzi 2.0. Do té doby se okna ve Windows nemohla překrývat, ale byla řazena vedle sebe, jako je vyobrazeno na obrázku číslo 5.

Okenní uživatelský režim umožňuje mnohem snazší práci s Větším množstvím aplikací. Uživatel má tak na očích mnohem více informací a své práce. Může interagovat mezi jednotlivými programy a přenášet mezi nimi data. [12]



Obrázek 5 – Uživatelské prostředí systému Windows 1.01 [12]

2.1.3 Ovládání uživatelského prostředí

2.2 Vývoj moderního prostředí

S příchodem Windows 8 začal Microsoft pracovat také na nových, moderních aplikacích, které přinášely zcela rozdílné UI od těch klasických. Tyto aplikace nazýval Microsoft Metro aplikace a nové prostředí Metro UI. UI bylo nejspíše inspirováno jednoduchostí, ikonografií a typografií a čtvercovými symboly z plánů odjezdů a příjezdů v metru. Hlavní obrazovku tak tvořily pohyblivé dlaždice, sloužící jako ikona aplikace a widget zobrazující její informace. [18]



Obrázek 6 – Úvodní obrazovka Windows 8 [13]

Prvky Metro UI však nepřišly poprvé s Windows 8. Microsoft s tímto prostředím experimentoval již dříve ve svých hudebních přehrávačích Zune [14] či mobilních telefonech Microsoft Kin [15], kde bylo poprvé použito prostředí s dlaždicemi, což je zobrazeno na obrázku číslo 8.

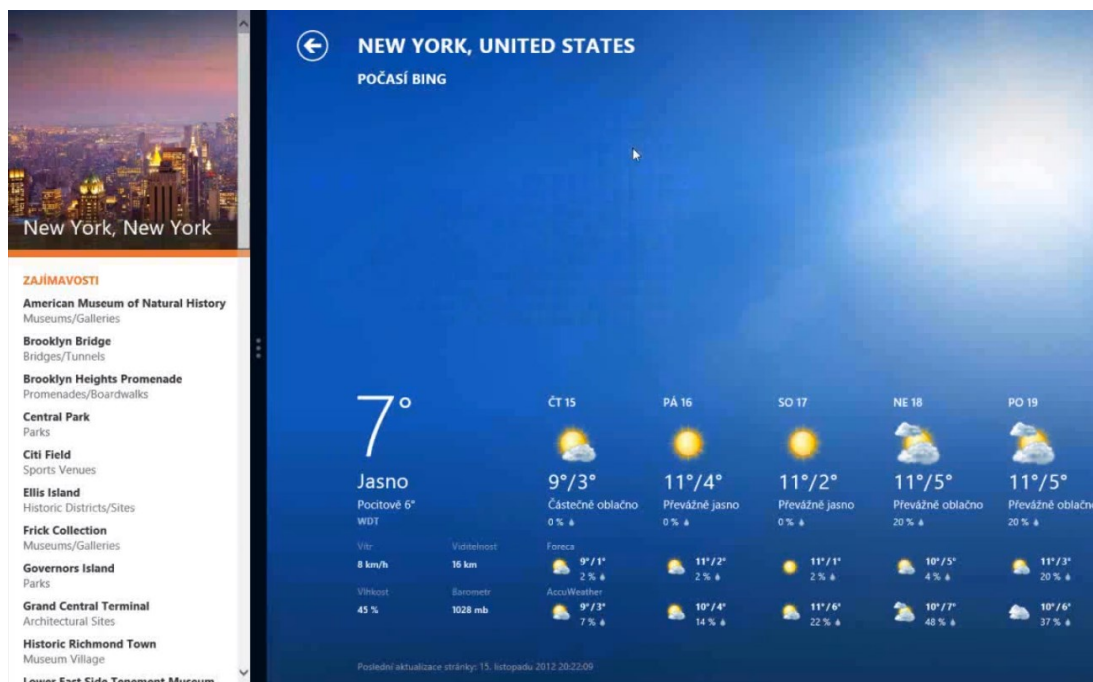


Obrázek 7 – Microsoft Zune [14]



Obrázek 8 – Microsoft Kin a jeho prostředí s dlaždicemi [15]

Prvky Metro aplikací byly zvětšeny pro dotykové ovládání a aplikace byly roztaženy přes celou obrazovku. Jediná možnost multitaskingu byla rozdělit obrazovku na jednu menší a jednu větší aplikaci, což je zobrazeno na obrázku 9.



Obrázek 9 – Aplikace ve Windows 8 a multitasking

Prostředí Metro UI se postupně vyvíjelo a promítlo své prvky také do dalších služeb a produktů stejně, jako dnešní design. [21]



Obrázek 10 – UI Windows Phone / Windows 10 Mobile [16]



Obrázek 11 – UI Xbox OS [17]

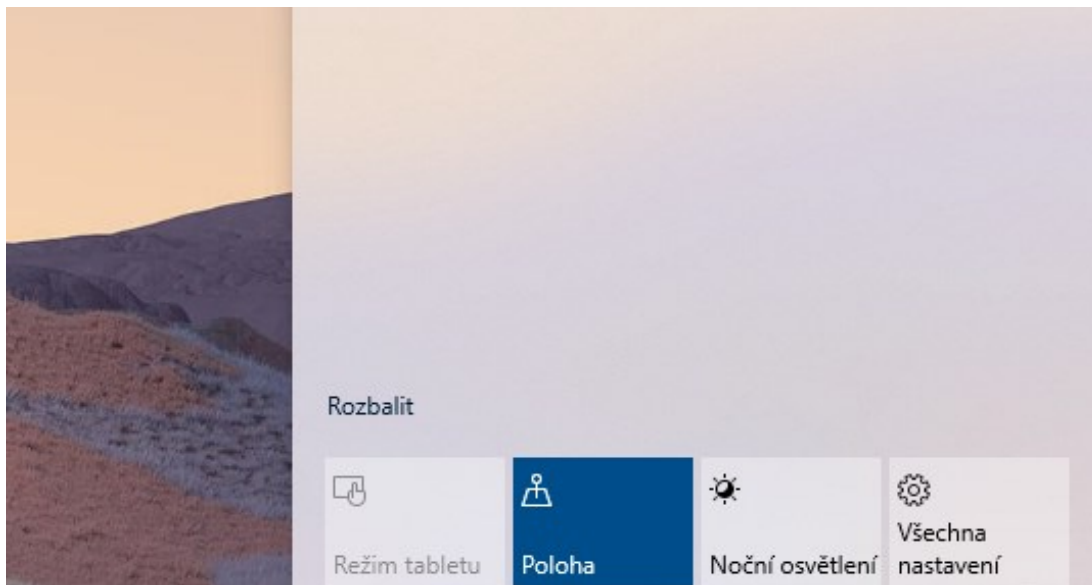
Microsoft definoval prostředí Metro takto [18]: „Metro je náš designový jazyk. Říkáme mu Metro, protože je moderní a čistý. Je rychlý a v pohybu. Je to o obsahu a typografii“.

V srpnu 2012 došlo ke změnám názvu tohoto prostředí. Kvůli neshodám o označení Metro s německou firmou Metro AG, Microsoft od tohoto označení upustil a dále označoval toto

prostředí za Modern UI. Microsoft poté prostředí kompletně postupně přepracoval v nynější Fluent Design System. [19]

2.3 Microsoft Fluent Design System

Fluent Design System, nebo také Fluent Design, znamenal návrat prvkům, od kterých se upustilo s příchodem Windows 8. UI se navrácí opět k průhlednostem a efektům, které imitují reálný materiál, jako například sklo nebo jakousi námrazu, jako jen vyobrazeno na obrázku 12. Fluent Design je pro vývojáře open-source a funguje napříč platformami. Design se zároveň adaptuje prostředí či stylu ovládání a je vhodný pro použití například na telefonech a tabletech, ale i počítačích. Tento nový design se také soustředí na efekty světla, stínů a prostoru a animace. [20] [23]



Obrázek 12 – Průhlednosti Fluent Designu ve Windows 10

Fluent design se skládá z několika důležitých prvků:

Průhlednosti

Průhlednosti s imitací průsvitných materiálů. Aplikace mohou použít průsvitnosti jednotlivých panelů či záhlaví a podobně. Průhlednosti využívá také samotný systém například pro Hlavní panel či Akční centrum.

Stíny a hloubka

Stíny jednotlivých prvků a oken. Stíny udávají důležitost jednotlivým oknům aplikací či dialogovým upozorněním. V operačním systému Windows 10 stíny vrhají okna aplikací, Akční centrum či vyskakovací okna a hlášky. Stíny také zvýrazňují tlačítka. Zvláště za použití plochého designu pomůže takové zvýraznění prvku ke zlepšení navigace v prostředí.

S aktualizací Windows 10 ve verzi 1903 přidal Microsoft efekty stínů do systému. Zároveň Microsoft automaticky aktivoval stíny u vyskakovacích oken či popisků prvků v aplikacích třetích stran. Rozhodnutí, kde dále se budou stíny zobrazovat, nechal Microsoft na vývojářích, aby nedošlo k nechtěnému narušení prostředí.

Světlo

Kurzor myši u některých prvků následuje efekt světla. Zvýrazňuje tak různé prvky, jako například tlačítka či prvky v seznamu. Ve světlém tématu označuje tyto prvky podobný efekt, ale v tmavších barvách.

Animace

Animace, které spojují jednotlivé části aplikací či systému. Přepínání režimů a podobně. Tyto animace, které spojují nějaké akce či obrazovky, místo náhlého přepnutí, způsobují dojem větší plynulosti prostředí.

Přizpůsobitelné posuvníky

Přizpůsobitelné posuvníky, které se zobrazí pouze v případě potřeby. Jinak se zobrazí pouze ukazatel pozice na stránce a šetří tak místo na obrazovce. Při dotyku jsou například skryty.

Imitace materiálu

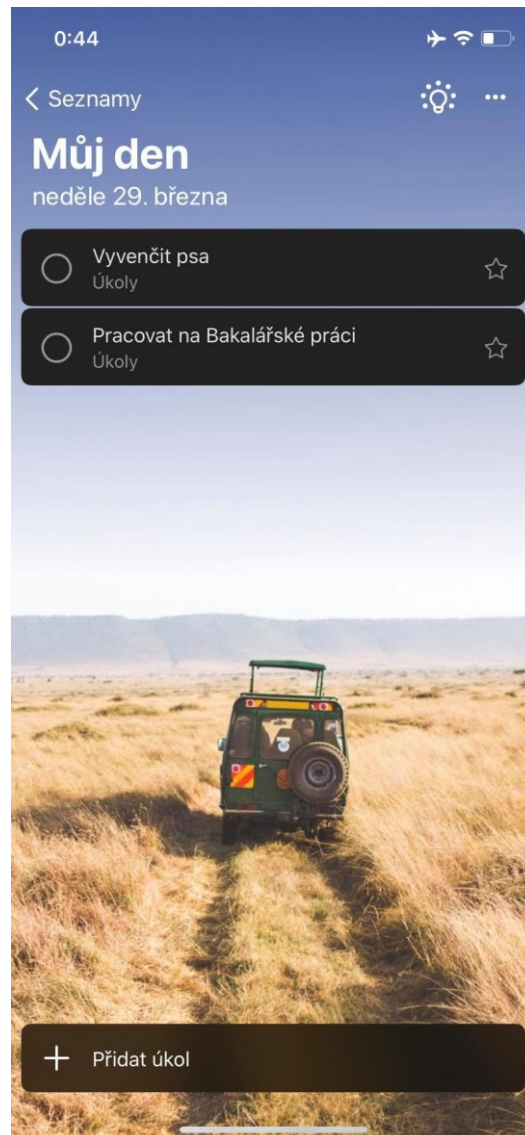
Imitace materiálu se pokouší napodobit prvky reálného světa a přenést je na ikony aplikací či do efektů průhledností.

[20]

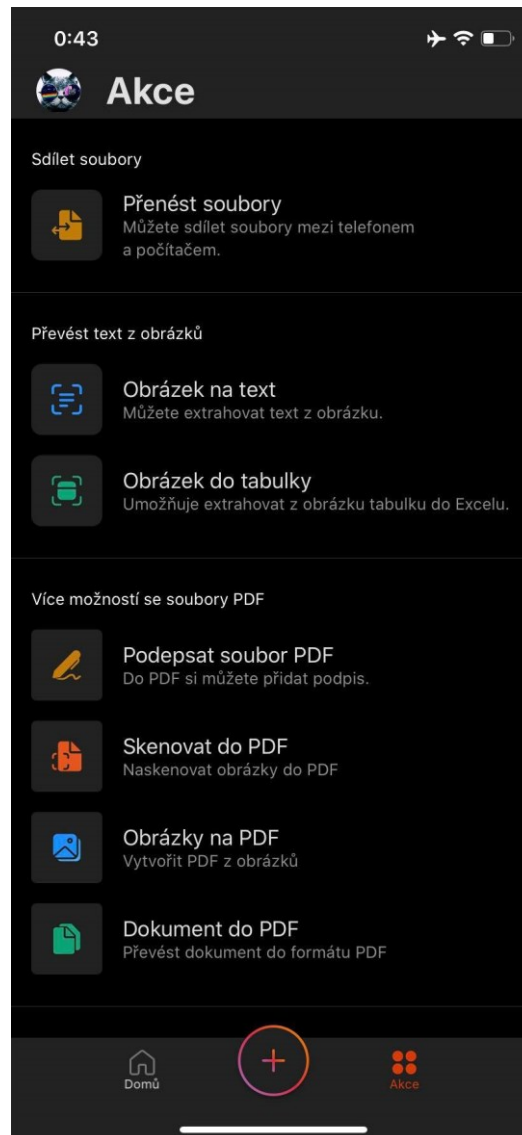
2.3.1 Nasazení Fluent Designu

Fluent design je nasazován Microsoftem v operačním systému Windows 10, Xboxu a také například na web. Microsoft využívá Fluent designu také ve svých aplikacích pro konkurenční operační systémy Android a iOS.

Vzhledem k tomu, že není vhodné rozbíjet stanovený design Applu, je použití Fluent designu v systému iOS střídmejší. Microsoft narušil konzistenci UI, které určuje systém. Proto se v tomto případě jedná převážně pouze o minimální použití Fluent designu. Jde například o ikony aplikací, zvolené barvy a text či menší změna tvarů prvků. [21]



Obrázek 13 – Microsoft ToDo na iOS



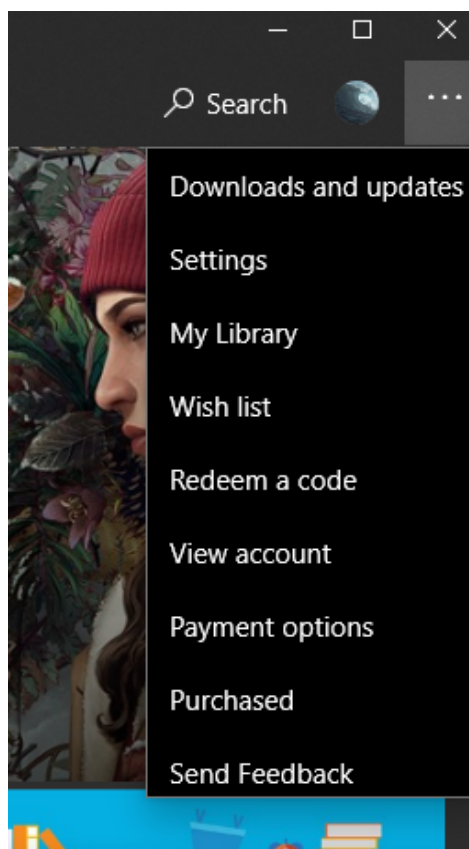
Obrázek 14 – Microsoft Office na iOS

2.3.2 Nekonzistence UI

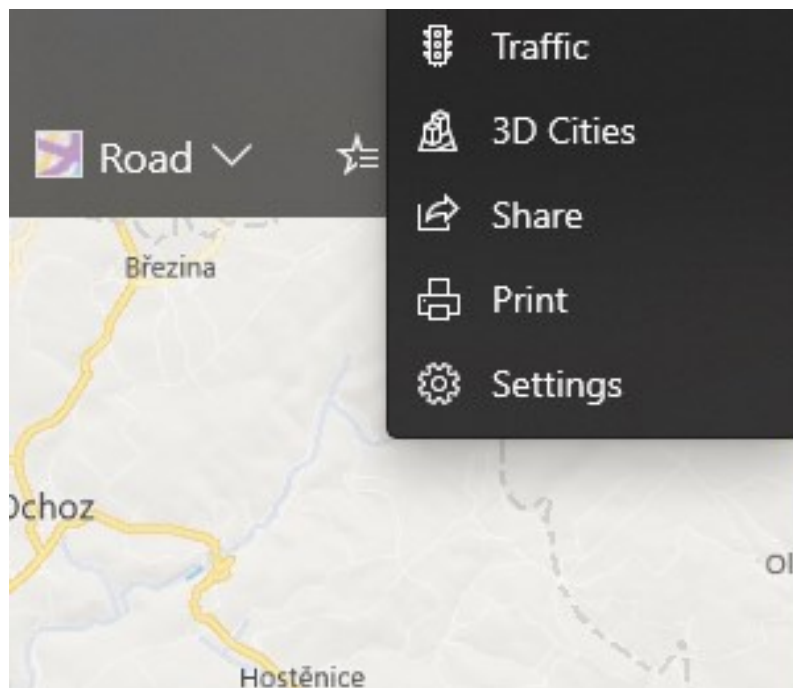
Časté aktualizace designu a jednotlivých prvků na jedné straně může přinášet moderní prvky. Bohužel však s sebou nese velké zvyšování nekonzistence, jelikož v systému a aplikacích zůstávají také prvky starší.

Samotný Microsoft má poněkud problémy s dodržováním konzistence UI, jako je vidět na obrázcích 15–18. Porovnáme-li předinstalované aplikace Windows, často nedodržují stanovený design, animace či rozložení. Microsoft však nejspíše podle aktuálního stavu aplikací ve Windows spatřuje větší důležitost v unikátním rozložení prvků pro každou

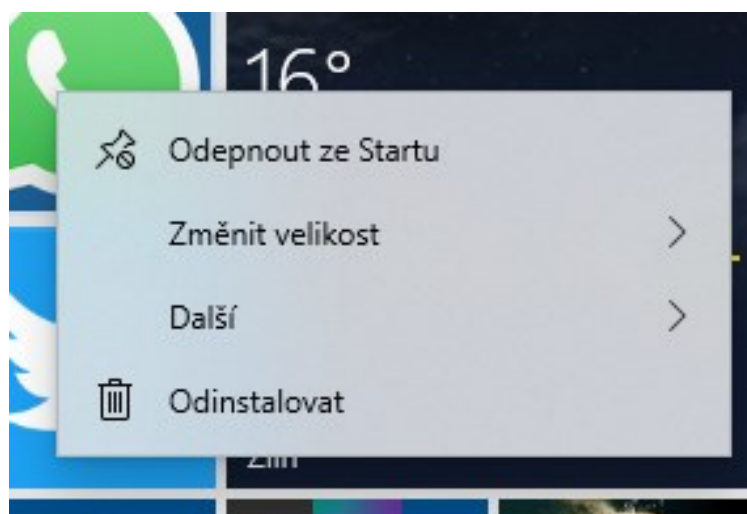
aplikaci. To může přinést lepší přehlednost a použití aplikace pro každý styl užití. Fluent Design je zprostředkován prostřednictvím knihovny WinUI.



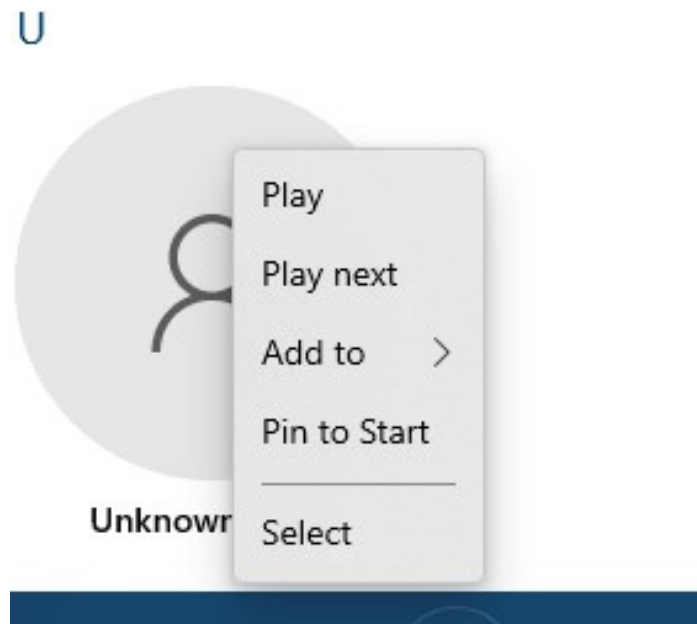
Obrázek 15 – Nekonistence UI – hranaté Drop Down Menu



Obrázek 16 – Zaoblené UI – hranaté Drop Down Menu



Obrázek 17 – Nekonzistence UI – hranaté Kontextové Menu v nabídce Start



Obrázek 18 – Nekonzistence UI – zaoblené Kontextové Menu

2.4 Knihovna WinUI

WinUI je nativní platforma uživatelského rozhraní pro Windows 10. Poskytuje velké množství nejnovějších prvků a funkcí pro UX Windows a aplikací. Obsahuje aktuální ovládací prvky a styly Fluent Design [23].

Použití knihovny WinUI je velmi snadné. K jeho aplikaci si postačí stáhnout balíček NuGet WinUI 2.1. Nově je také možné snadno stáhnout balíček WinUI 3 Preview 2 VSIX package do Visual Studia přímo ze stránek Microsoftu. [22] Zároveň je třeba mít nainstalovanou nejnovější preview verzi Visual Studia. Knihovna WinUI, která je aktuálně veřejně dostupná ve verzi 2.3, je momentálně využívána také samotným Microsoftem. Díky této knihovně Microsoft, ale i vývojáři, mohou využívat moderních prvků ve svých produktech. A to jak v desktopových aplikacích, tak UWP. Je možné WinUI využít jak u nových, tak u již existujících aplikací [23], jako MFC, WPF, WinForms, WPF, za využití jazyků C#, C++, Visual Basic nebo Javascript.

Vývojářům tak nejen usnadňuje použít hotové funkční prvky pro své aplikace, ale zároveň umožňuje zachovat konzistenci prostředí. WinUI je nezávislý na platformě XAML. Z tohoto důvodu není třeba využívat XAML platformu, která je součástí OS. To znamená nejen mnohem lepší flexibilitu pro aktualizace a obměny jednotlivých prvků. Dříve se muselo spoléhat na aktuální a dostupný set prvků, který přišel s aktuálním systémem. [24]

To však mohlo právě také způsobovat nekonzistenci grafického prostředí jako celku ve Windows, což narušuje uživatelský zážitek.

2.4.1 Knihovna WinUI 2.0

WinUI 2.0 je první veřejné vydání knihovny Windows UI. Přináší některé prvky, které známe a aktivně používáme v systému a aplikacích dodnes. Obsahuje dva balíčky NuGet:

- Microsoft.UI.Xaml: Ovládací prvky a Fluent Design pro UWP aplikace.
- Microsoft.UI.Xaml.Core.Direct: Nízkoúrovňová API pro použití v komponentách middlewaru. [25]

Ovládací prvky v této verzi zahrnují elementy v tabulce 1.

| | |
|------------------|--|
| AcrylicBrush | Vykreslení prostředí poloprůhledným materiálem, který využívá více efektů včetně rozostření a struktury šumu. |
| BitmapIconSource | Jedná se o zdroj ikon, který používá jako svůj obsah bitmapu. |
| ColorPicker | Představuje ovládací prvek, který umožňuje uživateli vybrat barvu pomocí barevného spektra, posuvníků a zadávání textu. |
| CommandBarFlyout | Představuje speciální plovoucí nabídku, která poskytuje rozvržení pro AppBarButton a související příkazové prvky. |
| DropDownButton | Tlačítko se šipkou sloužící k otevření nabídky. |
| FontIconSource | Představuje zdroj ikon, který používá symbol z určeného písma. |
| MenuBar | Představuje specializovaný kontejner, který představuje sadu nabídek v horizontálním řádku, obvykle v horní části okna aplikace. |
| MenuItem | Představuje nabídku nejvyšší úrovně v ovládacím prvku MenuBar. |
| NavigationView | Představuje kontejner, který umožňuje procházení obsahu aplikace. Má záhlaví, zobrazení hlavního obsahu a podokno nabídek pro navigační příkazy. |
| ParallaxView | Představuje kontejner, který spojuje polohu posouvání prvku popředí, například seznamu, s prvkem pozadí, jako je obrázek. |

| | |
|-----------------------|--|
| | Při procházení prvkem popředí animuje prvek pozadí a vytváří efekt paralax. |
| PersonPicture | Představuje ovládací prvek, který zobrazuje avatar pro osobu, pokud je k dispozici; pokud ne, zobrazí iniciály osoby nebo výchozí symbol. |
| RatingControl | Představuje ovládací prvek, který umožňuje uživateli zadat hodnocení hvězdičkami. |
| RefreshContainer | Představuje ovládací prvek kontejneru, který poskytuje funkci RefreshVisualizer a pull-to-refresh pro rolovatelný obsah. |
| RefreshVisualizer | Ovládací prvek, který poskytuje animované stavové indikátory pro aktualizaci obsahu. |
| RevealBackgroundBrush | Vykreslování prvků pomocí světelných efektů. |
| RevealBorderBrush | Obkreslování prvků pomocí světelných efektů. |
| RevealBrush | Základní třída pro štětce, které využívají efektů kompozice a osvětlení k odhalování elementů a vizuálního designu. |
| SplitButton | Tlačítko se dvěma částmi, které lze vyvolat samostatně. Jedna část se chová jako standardní tlačítko a druhá část vyvolá plovoucí nabídku. |

| | |
|----------------------|---|
| SwipeControl | Představuje kontejner, který poskytuje přístup ke kontextovým příkazům prostřednictvím dotykových interakcí. |
| SymbolIconSource | Představuje zdroj ikon, který jako svůj obsah používá symbol z písma Segoe MDL2 Assets. |
| TextCommandBarFlyout | TextCommandBarFlyout je speciální plovoucí lišta příkazového řádku, která obsahuje příkazy pro úpravu textu. |
| ToggleSplitButton | Jedná se o tlačítko, rozdělené na dvě části. Ty lze vyvolat samostatně. Jedna část se chová jako přepínací tlačítko a druhá část vyvolá plovoucí nabídku. |
| TreeView | Představuje hierarchický seznam s rozšiřujícími a sbalenými uzly, které obsahují vnořené položky. |

Tabulka 1: Ovládací prvky WinUI 2.0 [25]

2.4.2 Knihovna WinUI 2.1

WinUI 2.1 - byla vydána 8. dubna 2019. Jedná se o první vydání s otevřeným zdrojovým kódem z GitHubu. WinUI 2.1 se soustředí na animace, plynulé přechody. Především pak Microsoft vyzdvihuje stíny a zvýrazňování aktivních prvků. [26]

Ovládací prvky v této verzi zahrnují elementy, jako jsou vypsány v tabulce 2.

| | |
|----------------------|---|
| PoložkyRepeater | Pomocí položky ItemRepeater je možné vytvářet vlastní kolekce s využitím flexibilního systému rozvržení, vlastních pohledů a virtualizace. |
| AnimatedVisualPlayer | Program AnimatedVisualPlayer hostuje a řídí přehrávání animovaných vizuálů, což umožňuje přidat do aplikace vysoce výkonnou vlastní grafiku pohybu. |

| | |
|---------------------|--|
| TeachingTip | TeachingTip poskytuje poutavý a plynulý způsob, jak aplikace vést a informovat uživatele pomocí neinvazivních a obsahově bohatých tipů. |
| RadioMenuFlyoutItem | Zahrnuje možnost mít v nabídce tlačítka možnosti stylu tlačítka „Radio Button“. To umožňuje tvořit skupiny voleb s odrážkami, které jsou svázány dohromady jako skupina přepínačů. |
| CompactDensity | Tento režim umožňuje vytvořit větší hustotu prvků. |
| Stíny | Vytvoření vizuální hierarchie prvků v uživatelském rozhraní. |

Tabulka 2: Ovládací prvky WinUI 2.1 [26]

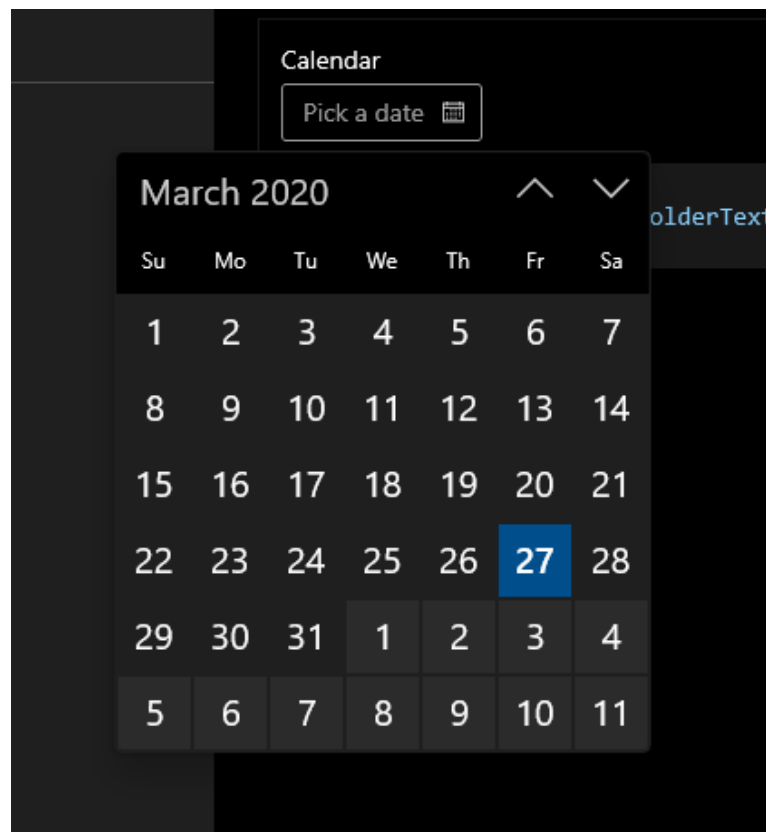
2.4.3 Knihovna WinUI 2.2

S verzí WinUI 2.2 se hranaté prvky vrací ke kulatosti. Progress bary, tlačítka či date pickery tak mají rádius. [27]

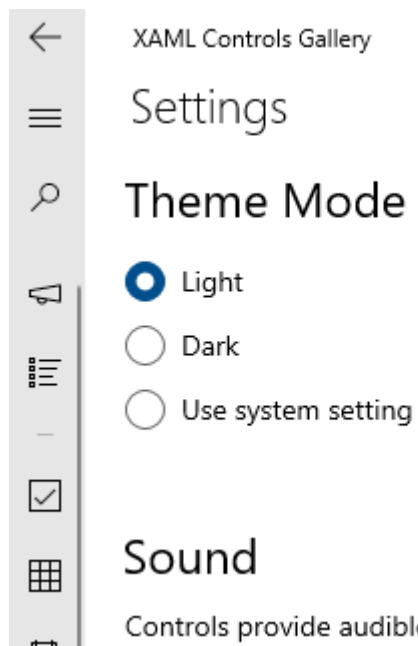
| | |
|--------------------------|---|
| TabView | Ovládací prvek TabView je kolekce karet, z nichž každá představuje vlastní stránku s vlastním obsahem v aplikaci podobně, jako v internetovém prohlížeči. |
| Tlačítko Zpět Navigace | Tlačítko zpět je nyní stále viditelné i po skrytí menu. Pro tuto aktualizaci není třeba provádět žádné změny kódu. |
| Aktualizace radiusu rohů | Byl zde přidán atribut CornerRadius. Výchozí ovládací prvky byly aktualizovány, aby používaly mírně zaoblené rohy. Vývojáři pak samostatně mohou snadno přizpůsobit poloměr rohu. |

| | |
|---|--|
| Aktualizace tloušťky okrajů | Byly aktualizovány okraje a obrysy pro čistší dojem. |
| Vizuální aktualizace tlačítka | Byl aktualizován vzhled tlačítka. Pro čistší vzhled byl odstraněn jeho obrys. |
| Vizuální aktualizace SplitButton | Výchozí vzhled prvku SplitButton byl aktualizován tak, aby byl vizuálně odlišen od DropDownButton. |
| Vizuální aktualizace prvku přepínač (ToggleSwitch) | Výchozí šířka prvku ToggleSwitch byla snížena z 44px na 40px. |
| Vizuální aktualizace zaškrtačacího políčka CheckBox a RadioButton | Vizuální prvky zaškrtačacích políček CheckBox a RadioButton byly aktualizovány, aby byly konzistentní se zbytkem změny vizuálního stylu. |

Tabulka 3: Ovládací prvky WinUI 2.2 [27]



Obrázek 19 – WinUI 2.2 a zakulacení elementů u Date Pickeru



Obrázek 20 – Tlačítko zpět je součástí menu

2.4.4 Knihovna WinUI 2.3

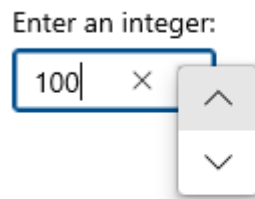
WinUI 2.3 je nejnovější oficiální vydání Windows UI Library (WinUI).

Tato verze přidává některé nové prvky, či jejich úpravy [28].

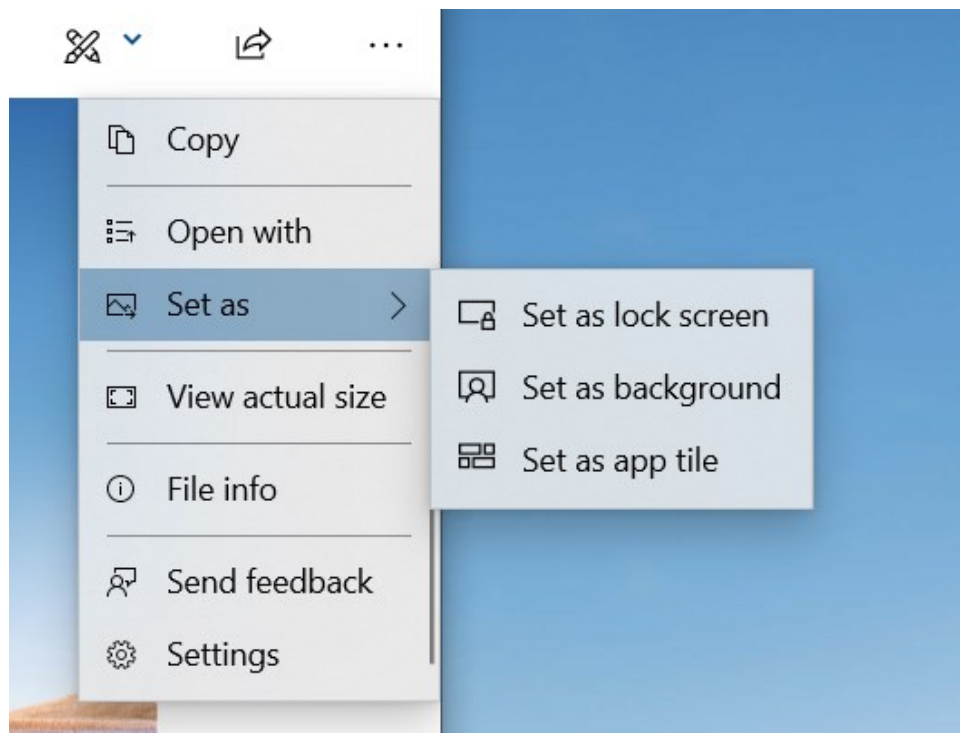
Mezi ně patří elementy, které jsou vypsány v tabulce 4.

| | |
|----------------------------|--|
| ProgressBar | ProgressBar má nyní více vizuálních vlastností. |
| Indeterminate Progress Bar | Ukazuje, že úloha právě probíhá a interakci uživatele neblokuje. |
| Určete ukazatel průběhu | Prvek ukazuje, progres práce. |
| NumberBox | Ovládací prvek sloužící k zobrazení a úpravám čísel. |
| Radiobuttons | RadioButtons je nový ovládací prvek kontejneru, který umožňuje snadno vytvářet související skupiny prvků RadioButton. Přidáno také přepínání focusu pomocí šipek klávesnice a předčítání obsahu programem Předčítání. |

Tabulka 4: Ovládací prvky WinUI 2.3 [28]



Obrázek 21 – Numberbox s efektem stínu



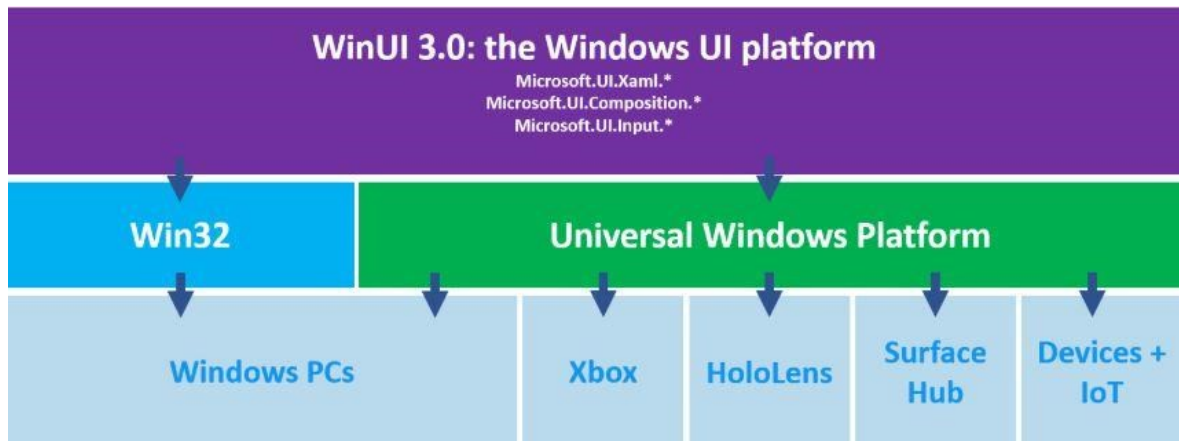
Obrázek 22 – Menu s efekty stínu

2.4.5 Knihovna WinUI 3.0

Knihovna WinUI 3.0 není závislá na UWP a lze ji použít také na klasických desktopových aplikacích Win32. Jedná se o nejnovější verzi knihovny WinUI, stále se však nachází v rané verzi, takže pro oficiální nasazení stále není vhodná. Vzhledem k jednoduchosti použití této knihovny a možnosti ji aplikovat také na klasické aplikace je možné využít moderních prvků i u starších systémů Windows. To s sebou nepřináší pouze moderní vzhled, ale také nové moderní funkce, jako například animace či moderní přehrávač a podporu dotykového ovládání. [29] Původní nutnost přepsání jejich aplikací do UWP pro využití moderních funkcí tak Microsoft nyní nahrazuje mnohem efektivnějším řešením. Díky WinUI 3.0 bude

zcela oddělena nativní platforma UI Windows 10 od UWP. Jakékoliv klasické Win32, Windows Forms, WPF či UWP aplikace tak mohou využít moderních prvků a funkcí. [30]

WinUI 3.0 předně nabízí několik novinek:



Obrázek 23 – Použití knihovny WinUI [28]

V současné době Microsoft spolupracuje také s platformou Uno. [31] Díky níž mohou vývojáři přenést svůj kód a WinUI v podstatě kamkoliv. Aplikace tak dokáže nativně na iOS, Android, macOS a web prostřednictvím WebAssembly. Stejně tak je možné využít WinUI na starších systémech Windows, jako například Windows 7. [32]

2.4.6 Současné limitace knihovny WinUI 3.0

WinUI 3.0 v současné fázi neobsahuje některé grafické elementy či funkce. Převážně se jedná grafické prvky. Ve WinUI 3.0 chybí Acrylic průhlednosti, které jsou v současnosti využity ve Fluent Designu například pro kontextové menu či boční panel Hamburger Menu. Chybí také Reveal Highlight pro označování prvků okolo kurzoru, jako jsou tlačítka či textová pole. Prozatím také chybí moderní UWP přehrávač médií.

Mezi další chybějící funkce patří [33]:

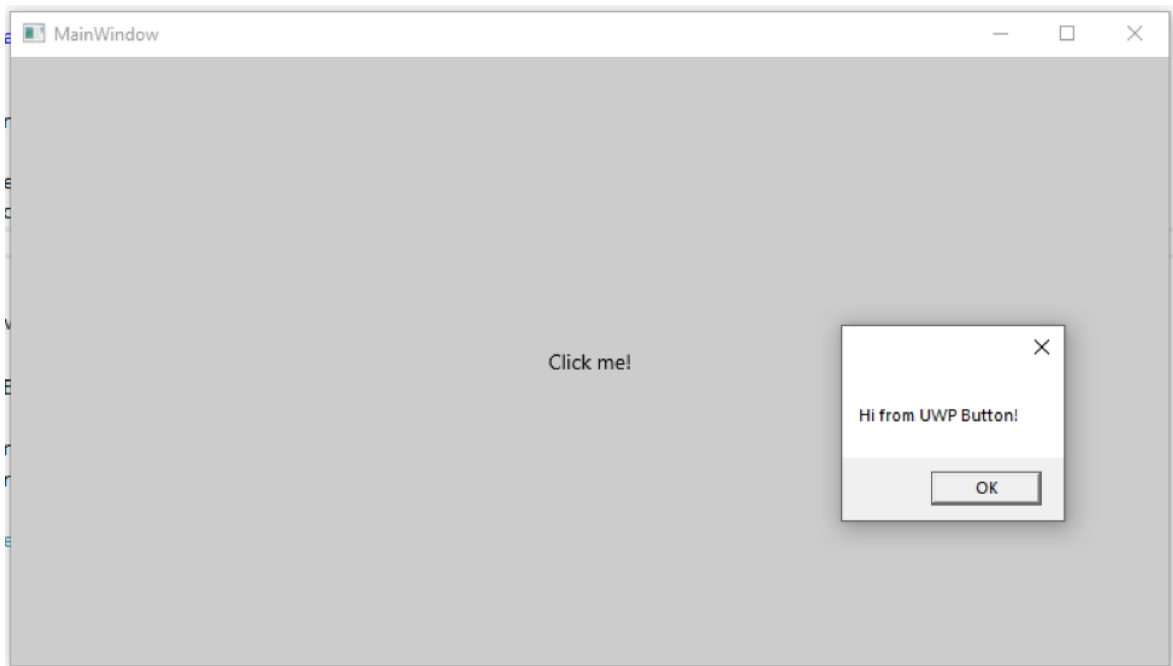
- Podpora Xboxu
- Podpora HoloLens
- Vyskakovací okna
- Podpora inkoustu
- Pozadí Acrylic
- MediaElement a MediaPlayerElement
- RenderTargetBitmap
- MapControl
- SwapChainPanel nepodporuje průhlednost
- Global Reveal je pouze základní
- XAML Islands nejsou v této verzi podporovány
- Funkčnost knihoven ekosystémů třetích stran
- Funkčnost IME

2.5 XAML Islands

Knihovnu WinUI je také možné použít se staršími technologiemi a programovacími jazyky. Je možno ji použít s programovacími jazyky, jako C++ a C#. Druhou řadu WinUI můžeme hostovat skrze XAML Islands také v aplikacích WinForm, WPF a Win32 aplikaci. Toto je možné od verze 1903 Windows 10. [34]

Díky této funkci je možné používat moderní funkce a elementy a ovládací tlačítka, jak v aplikaci Windows forms a WPF, tak v aplikaci Win32 v C++. Je tak možné využít možnosti z UWP či nových elementů z Fluent Design v existujících aplikacích, které nebyly napsány jako univerzální. V podstatě je XAML Islands technologie, která umožňuje vývojářům Windows používat nové elementy uživatelského rozhraní z platformy UWP na jejich stávajících aplikacích Win32, včetně technologií Windows Forms a WPF. [34]

Na obrázku 24 si můžeme všimnout WPF aplikace, která využívá UWP tlačítko.



Obrázek 24 – Ukázka elementů WinUI [34]

```
myUwpButton.ChildChanged += MyUwpButton_ChildChanged;
...
private void MyUwpButton_ChildChanged(object sender, System.EventArgs e)
{
    if (myUwpButton.Child is Windows.UI.Xaml.Controls.Button button)
    {
        button.Content = "Click me!";
        button.Click += (s, args)=>
        {
            MessageBox.Show("Hi from UWP Button!");
        };
    }
}
```

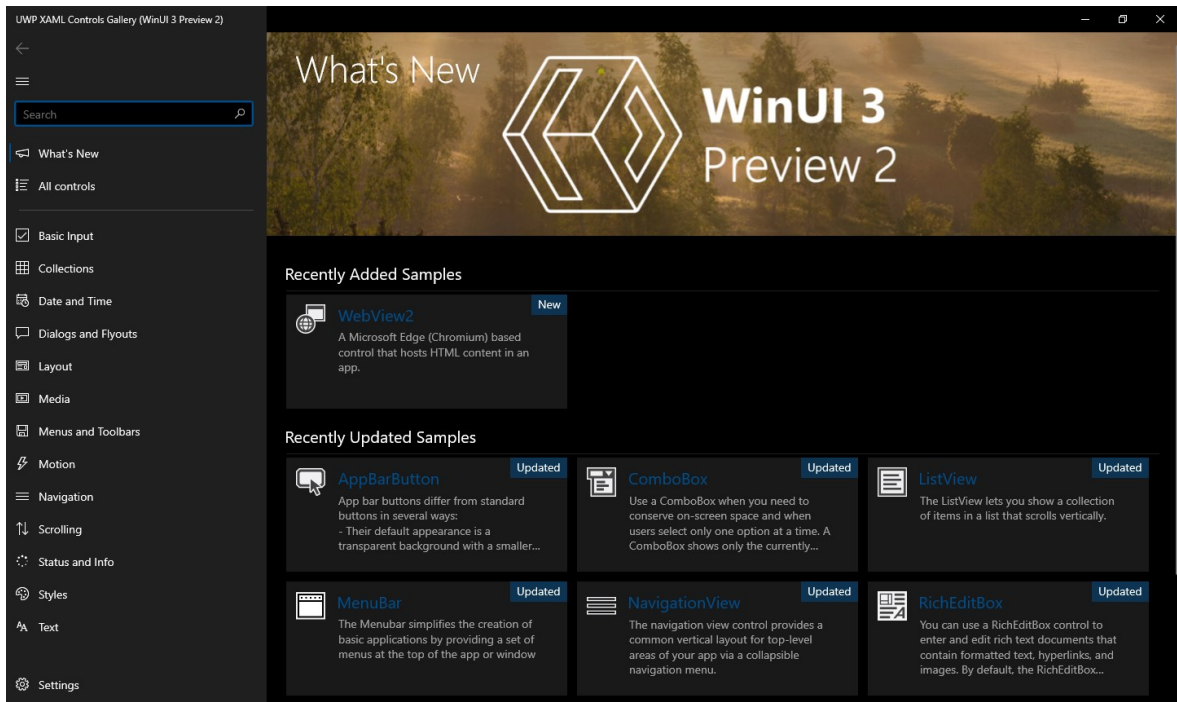
Kód 1. – Na ovládacím prvku je specificky událost zvaná ChildChanged [34]

2.6 XAML Control Gallery

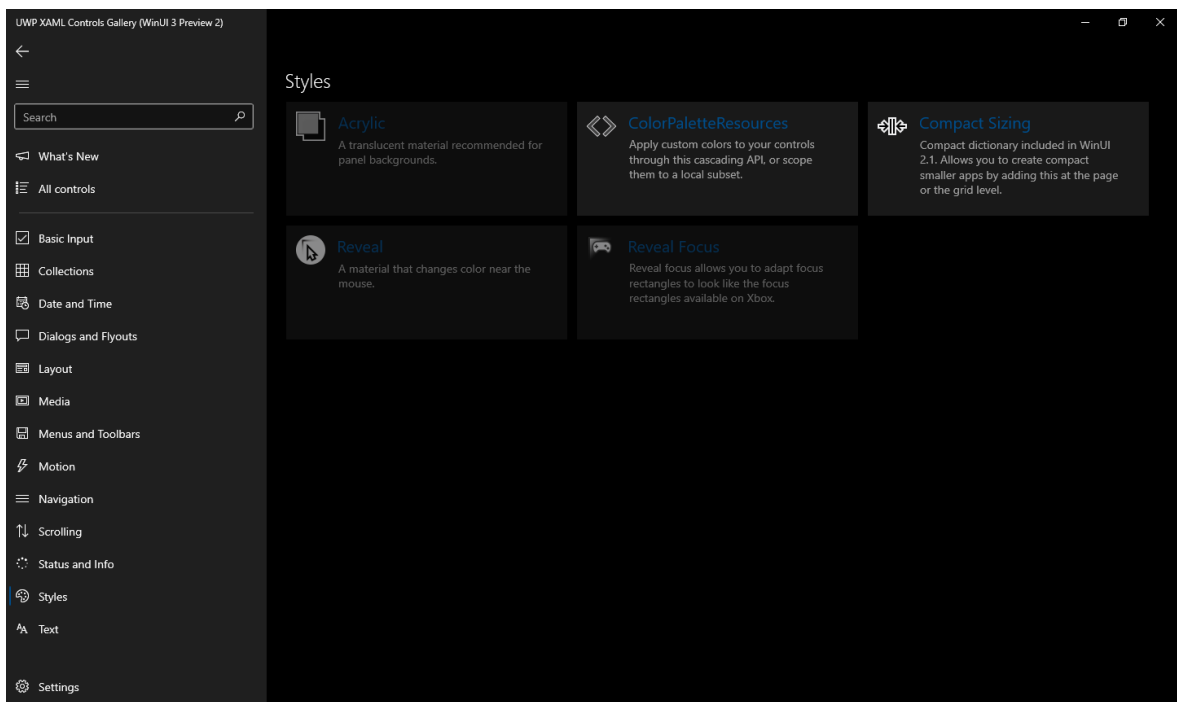
XAML Control Gallery obsahuje praktické ukázky elementů a kódů knihovny WinUI. Je možné je stáhnout z Microsoft Storu. Zde se však jedná o verzi, obsahující poslední elementy veřejně dostupné verzi knihovny WinUI.

Microsoft také nabízí verzi obsahující elementy a ukázky kódů knihovny WinUI 3.0. Ta je ke stažení na GitHubu. XAML Control Gallery obsahuje snadný přehled elementů, se

kterými je možno na zkoušku dokonce interagovat. Samozřejmou součástí aplikace jsou také zdrojové kódy. [35]

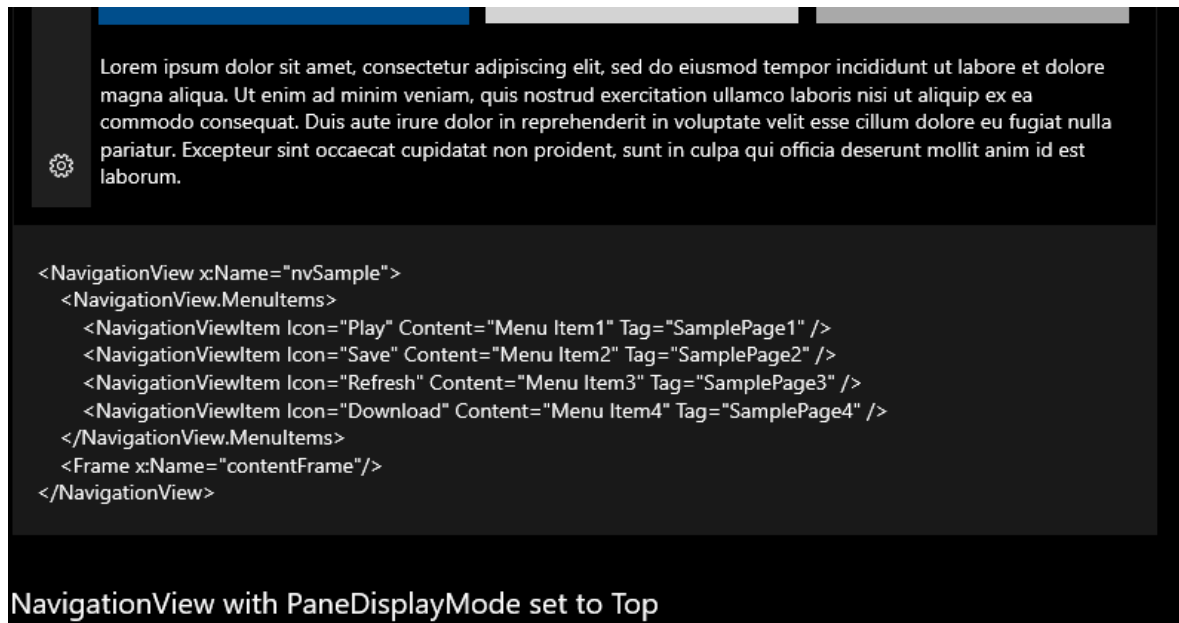


Obrázek 25 – Úvodní obrazovka aplikace UWP Xaml Controls Dallery (WinUI 3 Preview)



Obrázek 26 – Některé funkce či elementy prozatím v aplikaci UWP Xaml Controls Dallery (WinUI 3 Preview) chybí

Bohužel však v současné verzi není možno zkopírovat z aplikace kód. Jedná se zřejmě o dočasnou chybu.



Obrázek 27 – Aplikace UWP Xaml Controls Dallery (WinUI 3 Preview) má ještě své nedostatky. Například zde chybí kopírování kódu.

2.7 WinUI a starší technologie tvorby desktopových aplikací a různé programovací jazyky

Vzhledem k tomu, že se knihovna WinUI odděluje od UWP, umožňuje využívat moderních UWP prvků a funkcí nejen v UWP, ale také v desktopových aplikacích Win32 v C++, Windows Forms či WPF. Velmi ulehčuje práci pro vývojáře, kteří se nechtějí zaměřit pouze na Windows 10 s UWP aplikacemi, ale také na jiné systémy, a přesto chtějí využít moderních funkcí, animací a prvků. Starší WinUI ve verzi 2 byla k dispozici pouze pro UWP.

Pro vývojáře to také znamená mnohem snazší tvorbu aplikací, které budou designově konzistentní se systémem Windows 10, což pro uživatele znamená známé prostředí a menší porušování konzistence. Zároveň však takové aplikace mohou nabízet nové funkce. Jedná se například o dotykové prvky, nově také Web View, který je postaven na Chromiu, či nové dialogy. Některé z funkcí však zatím nejsou k dispozici. Jedná se například o moderní přehrávač médií.

II. PRAKTICKÁ ČÁST

3 ÚVOD DO PRAKTICKÉ ČÁSTI

V praktické části bude vytvořena ukázková aplikace s využitím knihovny WinUI. Demonstrující možnosti knihovny WinUI. Bude vytvořeno několik jednoduchých ukázkových aplikací za využití různých technologií a programovacích jazyků, přičemž výsledkem by měly být aplikace využívající stejné moderní prvky. Aplikace tak budou využívat moderních elementů a jejich UX bude konzistentní, ať se jedná o aplikaci univerzální, klasickou desktopovou v C# či C++.

Na závěr budou zhodnoceny výhody a nevýhody této technologie a možnosti využití knihovny WinUI.

4 PŘÍPRAVA KNIHOVNY WINUI 3.0

Nejprve si musíme knihovnu WinUI 3.0 připravit pro použití pro Visual Studio. Pořídíme si nejnovější vývojové prostředí Microsoftu Visual Studio ze stránek Microsoftu. Do něj poté musíme doinstalovat rozšíření pro Windows UI Library (WinUI).

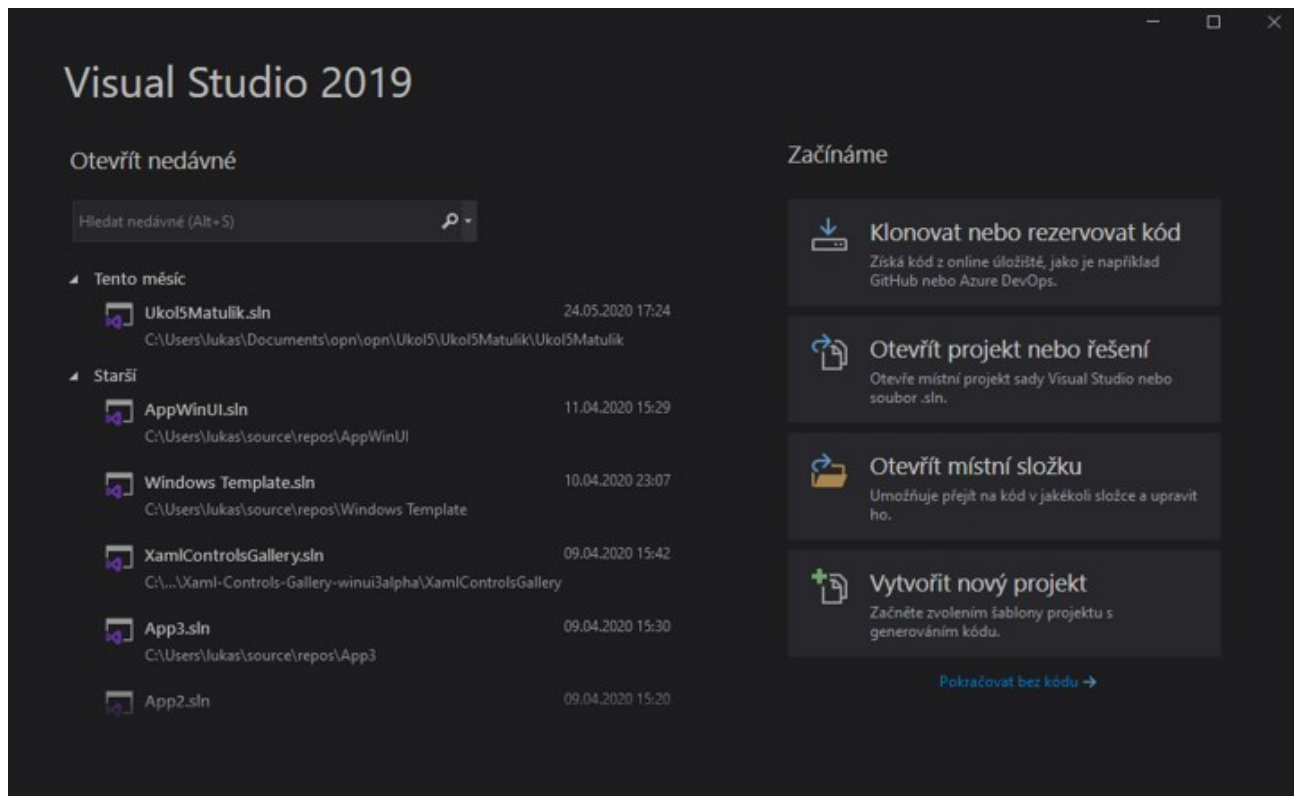
4.1 WinUI 3.0 pro desktopové aplikace

Pro usnadnění tvorby aplikací Microsoft vydal WinUI Preview. V současné chvíli je již k dispozici ve verzi 2. Ten zpřístupní šablony projektů. Pomocí nich je možné vytvářet desktopové C#/.NET aplikace a nativní C++/Win32 desktopové aplikace. Uživatelské rozhraní takovýchto aplikací je celé založeno na WinUI 3.0.

Instalace je velmi jednoduchá. Postačí instalace několika balíčků a nejnovější verze Preview Visual Studia z webových stránek Microsoftu.

5 PŘÍPRAVA

Ve chvíli, kdy máme připraveno WinUI 3.0 for desktop, můžeme začít s tvorbou aplikací využívající WinUI 3.0 a vytvořit nový projekt.



Obrázek 28 – Založení nového projektu s WinUI 3 Preview

Projekt obsahuje klasický XAML s podporou WinUI prvků. Jedná se o klasické rozložení, jako u univerzálních aplikací UWP.

Kód XAML tak píšeme u Win32 aplikací klasicky, jako u aplikací univerzálních.

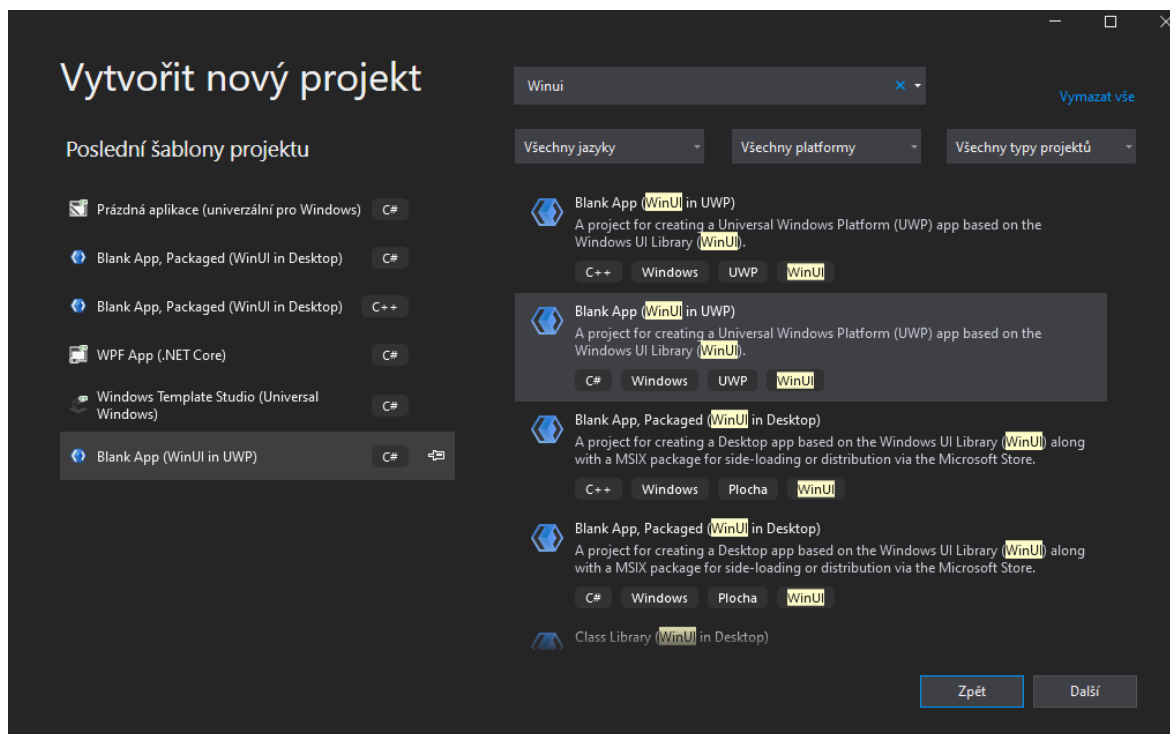
Pro naše jednoduché účely prezentace elementů WinUI 3.0 si vytvoříme Grid se Stack panelem. Bylo na ukázkou vytvořeno jednoduché menu, progress ring, který lze zastavit tlačítky, která jsou automaticky zaoblena dle nového Fluent Designu, dynamický progress bar, který určuje hodnotu, Date Picker pro výběr data, Drop down menu, Flyout pro ukázkou vyskakovacího upozornění s tlačítkem a také ukázkou Text boxu pro vkládání textu. Ten podporuje řadu funkcí převzatých z UWP aplikací. Například automatické opravy textu, textové návrhy a vkládání symbolů. Náš projekt se bude sestávat z jednoduchých elementů,

které jsou součástí WinUI 3.0. Ty se pokusíme implementovat do univerzální aplikace, klasické desktopové či desktopové v C++.

6 APLIKACE UWP S WINUI 3.0

Byl tedy založen nový projekt. Po instalaci rozšíření byla získána možnost tvořit WinUI aplikace ve verzi 3.0. A to jak aplikace klasicky desktopové na C#, C++, tak univerzální.

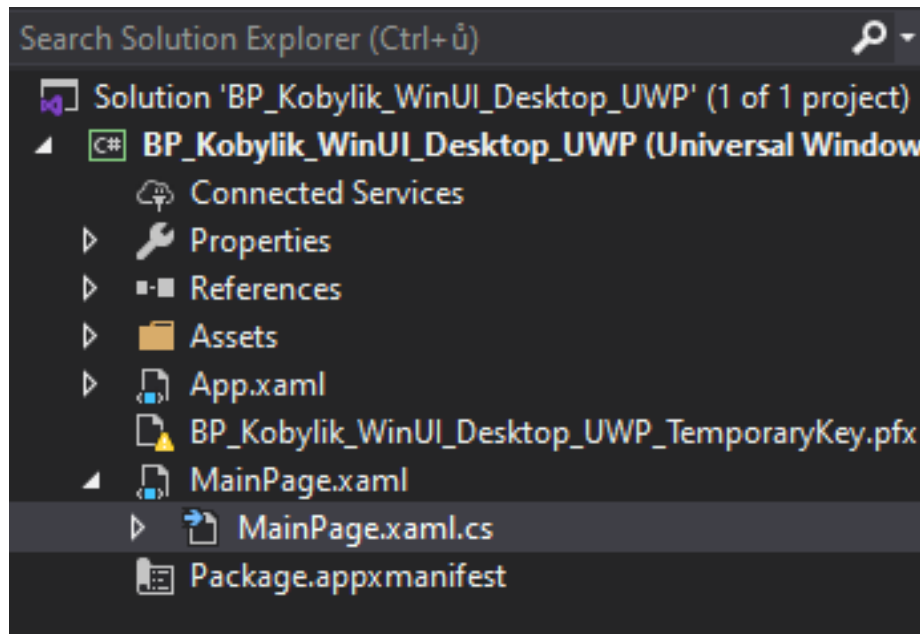
Naše první aplikace bude univerzální UWP v jazyce C#. Projekty univerzálních aplikací UWP již WinUI používá automaticky. Lze tedy využít totožného kódu XAML, jako u řešení pro Win32 aplikace.



Obrázek 29 – Založení nového projektu UWP aplikace

6.1 XAML UWP aplikace

Po vytvoření nového projektu byla vygenerována strana aplikace, která pro náš příklad byla využita.



Obrázek 30 – Solution Explorer UWP aplikace

Nejdříve je třeba se věnovat části XAML. Pro demonstrační účely posloužilo několik UWP elementů. Bylo využito několik animovaných prvků signalizujících činnost, jako například Progress Ring a Progress bar. Progress Ring bude možné pozastavit a znova spustit. Pro lepší interaktivitu může být implementován také stavový řádek progress bar určující konkrétní stav. Samozřejmě budou moderní tlačítka se zaoblenými hranami. Pro další ukázkou bude implementován do naší jednoduché aplikace například element pro výběr data, vyskakovací flyout, textový řádek, kde funguje automatická oprava textu a vkládání Emoji a smybolů z Emoji Pickeru či Color Picker pro výběr konkrétní barvy.

Pro ukázkou lze implementovat také obyčejné menu, které disponuje moderními zvýrazněnými aktivní záložkami. Identifikátor aktivní položky se také animuje při kliknutí na další záložku. Tento element tedy bude dalším vhodným prezentačním prostředkem pro tento projekt. Nejdříve bylo tedy vytvořeno menu.

```
<StackPanel Orientation="Vertical" Padding="10">

<NavigationView x:Name="nvSample" PaneDisplayMode="Top"
SelectionFollowsFocus="Enabled" IsBackButtonVisible="Collapsed">
<NavigationView.MenuItems>
<NavigationViewItem Icon="Preview" Content="Date Picker"
x:Name="SamplePage1Item" IsEnabled="True" />
<NavigationViewItem Icon="Calendar" Content="Item2"
x:Name="SamplePage2Item" />
<NavigationViewItem Icon="Clock" Content="Item3"
x:Name="SamplePage3Item" />
<NavigationViewItem Icon="Calculator" Content="Item4"
x:Name="SamplePage4Item" />
</NavigationView.MenuItems>
<Frame x:Name="contentFrame"/>
</NavigationView>
```

Kód 2. – XAML menu

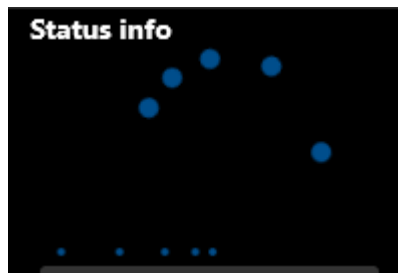


Obrázek 31 – Výsledné horní menu

Animované Elementy jsou důležitou a nedílnou součástí každé aplikace, která provádí jakoukoli složitější práci. V případě, že uživatel čeká na nějaký výsledek, je třeba, aby aplikace o práci informovala. Tento prvek tady dává znát, že aplikace skutečně vykonává určitou činnost. Těmito elementy jsou ve Windows Progress Ring a Progress bar.

```
<TextBlock Text="Status info" FontWeight="Bold" />
<ProgressRing Name="ProgressRing1" IsActive="True" Width="200" Height="100"
/>
<ProgressBar Name="ProgressBar1" Width="200" IsIndeterminate="True"
ShowPaused="False" ShowError="False" />
```

Kód 3. – XAML Progress Ring a Progress Bar



Obrázek 32 – Progress Ring

Pro další demonstrační účely moderních animací byl přidán přechodový efekt, který se zobrazí po načtení prvků.

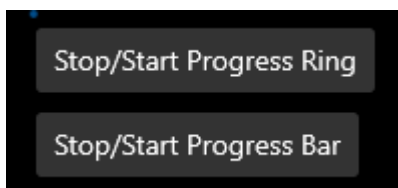
```
<StackPanel x:Name="ProgressRingStackPanel" Orientation="Vertical"
HorizontalAlignment="Center">
  <StackPanel.ChildrenTransitions>
    <TransitionCollection>
      <EntranceThemeTransition IsStaggeringEnabled="True" />
    </TransitionCollection>
  </StackPanel.ChildrenTransitions>
```

Kód 4. – XAML přechodový efekt

Přidáme několik tlačítek, kterými bude možné tyto elementy ovládat. například je pozastavit a znova spustit. tato tlačítka budou samozřejmě také v novém moderním designu, který ctí aktuální Fluent Design.

```
<Button Content="Stop/Start Progress Ring" Name="ButtonRing"
Click="Button_Click_Ring" Margin="5" />
<Button Content="Stop/Start Progress Bar" Name="ButtonBar"
Click="Button_Click_Bar" Margin="5" />
</StackPanel>
```

Kód 5. – XAML tlačítka pro interakci s Progress Ringem a Progress Barem



Obrázek 33 – Tlačítka

Dalším blokem aplikace je ukázka Progress Baru, který zobrazuje konkrétní hodnoty. S těmito hodnotami pak lze manipulovat pomocí moderních tlačítek, a to konkrétně navýšení či snížení hodnoty. Hodnota bude vypisována jak textově, tak ji bude zobrazována Progress Barem. Bylo přidáno také tlačítko na vyresetování hodnoty. Na to reaguje jak textový výpis, tak Progress Bar.

```
<TextBlock Text="A Determinate Progress Bar" FontWeight="Bold" />
<Button Content="Decrement the value" Name="ButtonMinus"
Click="Button_Click_Minus" Margin="5" />
<Button Content="Increment the value" Name="ButtonPlus"
Click="Button_Click_Plus" Margin="5" />
<ProgressBar Width="500" Value="12" Name="ProgressBar2" />
<TextBlock Text="Current value: 12" Name="TextBlockProgressBar" />
</StackPanel>
<Button Content="Reset" Name="ButtonReset" Click="Button_Click_Reset"
Margin="5" />
```

Kód 5. – XAML Progress Baru a ovládání jeho hodnot

Pro tuto sekci by bylo vhodné opět implementovat plynulý přechodový efekt. Proto byl pro tento blok opět implementován.

```
<TextBlock Text="A Determinate Progress Bar" FontWeight="Bold" />
<StackPanel x:Name="ValueStackPanel" Orientation="Horizontal">
<StackPanel.ChildrenTransitions>
<TransitionCollection>
<EntranceThemeTransition IsStaggeringEnabled="True" />
</TransitionCollection>
</StackPanel.ChildrenTransitions>
<Button Content="Decrement the value" Name="ButtonMinus"
Click="Button_Click_Minus" Margin="5" />
<Button Content="Increment the value" Name="ButtonPlus"
Click="Button_Click_Plus" Margin="5" />
<ProgressBar Width="500" Value="12" Name="ProgressBar2" />
<TextBlock Text="Current value: 12" Name="TextBlockProgressBar" />
</StackPanel>
```

Kód 6. – XAML ovládání hodnot Progress Baru s přechodovým efektem



Obrázek 34 – Progress Bar

Moderní prvek, který je také součástí nového designu, je Element pro výběr konkrétního data. Toho se využívá například v kalendáři či v aplikacích, kde je nutné nastavit termín. Element v novém designu opět disponuje zaoblením.

```
<CalendarDatePicker PlaceholderText="Pick a date" Header="Calendar"  
Name="Calendar" IsEnabled="True" FontWeight="Bold" />
```

Kód 7. – XAML Date Picker

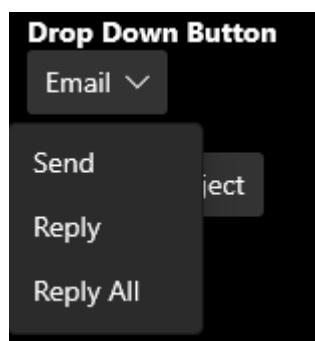


Obrázek 35 – Date Picker

Pro výběr z několika různých hodnot lze využít také rozbalovacího menu, které sdružuje různé funkce či možnosti do jedné kategorie. takováto rozbalovací nabídka opět disponuje moderními animacemi Samozřejmě je opět zaoblení hran.

```
<TextBlock Text="Drop Down Button" FontWeight="Bold" />
<DropDownButton Content="Email">
<DropDownButton.Flyout>
<MenuFlyout Placement="Bottom">
<MenuFlyoutItem Text="Send" />
<MenuFlyoutItem Text="Reply" />
<MenuFlyoutItem Text="Reply All" />
</MenuFlyout>
</DropDownButton.Flyout>
</DropDownButton>
```

Kód 8. – XAML Drop Down Menu

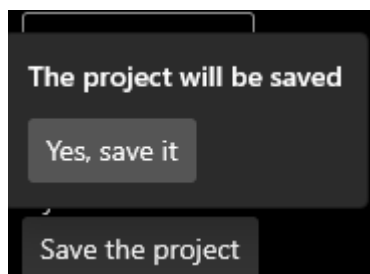


Obrázek 36 – Drop Down Menu

Element Flyout může sloužit k nějakému potvrzení určité akce nebo pro další informace po rozkliknutí. Tyto informace mohou dále sloužit uživateli k rozhodnutí, zda bude v operaci pokračovat či nikoliv. Jedná se o moderní Flyout který používá opět nejnovějšího stylu Fluent design. V našem případě se jedná o simulaci uložení souboru a následné potvrzení této operace.

```
<TextBlock Text="Flyout" />
<Button Content="Save the project">
<Button.Flyout>
<Flyout>
<StackPanel>
<TextBlock Style="{ThemeResource BaseTextBlockStyle}" Text="The project will
be saved" Margin="0,0,0,12" />
<Button Content="Yes, save it" />
</StackPanel>
</Flyout>
</Button.Flyout>
</Button>
```

Kód 9. – XAML Flyout

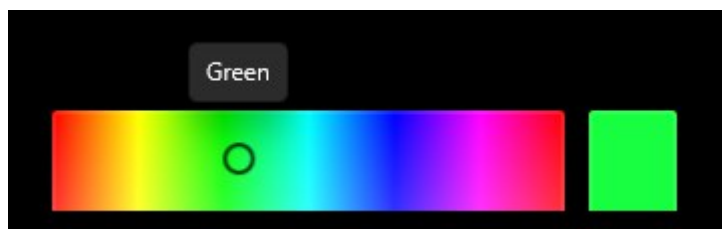


Obrázek 37 – UWP Flyout

Color Picker slouží k výběru konkrétní barvy z barevné palety. Může být využit například při kreslení, či editaci textu. Umožňuje také zobrazit a zadat hodnotu HEXA. Jelikož se jedná také o moderní element nového WinUI, byl využit do projektu pro ukázkou také.

```
<TextBlock Text="Color Picker" FontWeight="Bold" />
<ColorPicker IsMoreButtonVisible="False" IsColorSliderVisible="True"
IsColorChannelTextInputVisible="False" IsHexInputVisible="True"
IsAlphaEnabled="False" IsAlphaSliderVisible="False"
IsAlphaTextInputVisible="False" Width="50%" Height="50%" />
```

Kód 10. – XAML Color Picker

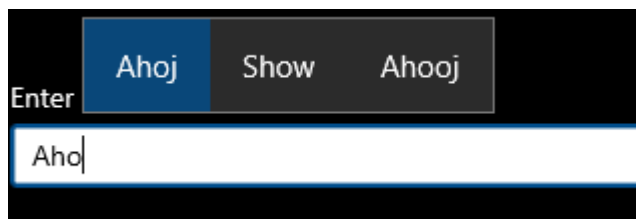


Obrázek 38 – Color Picker

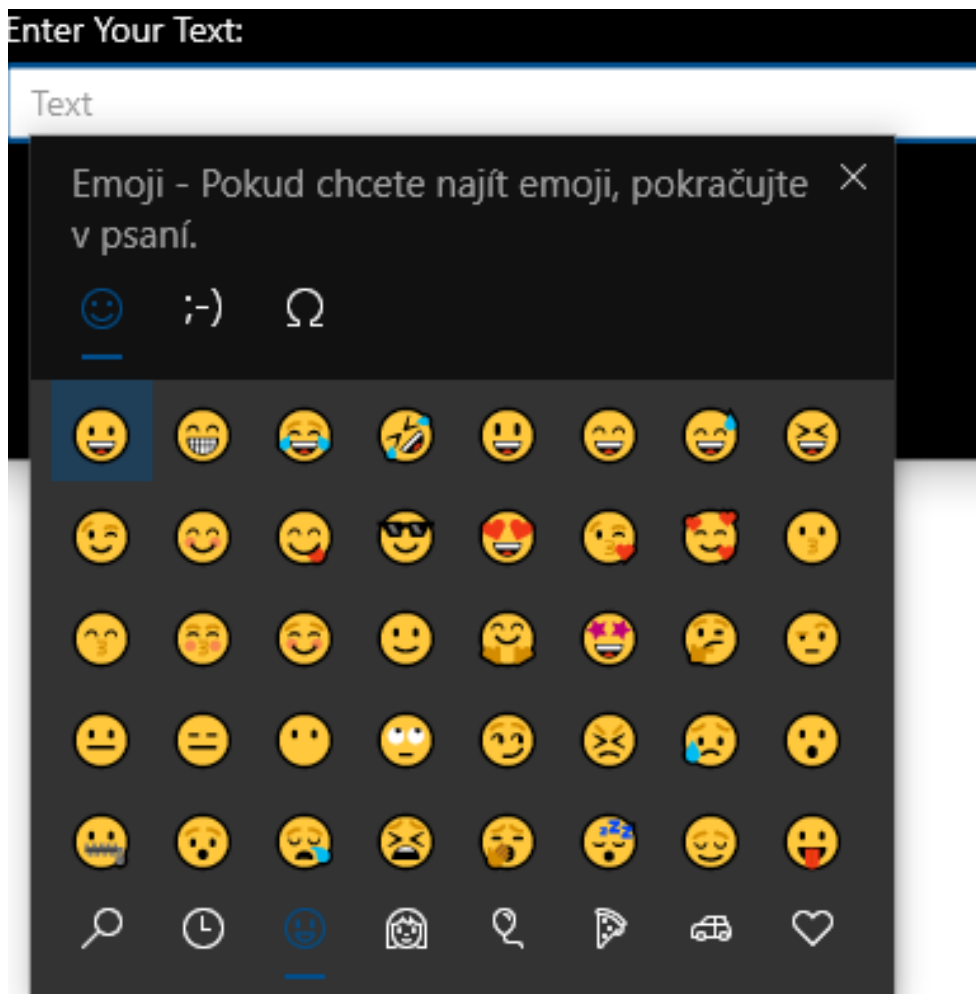
Moderní Text Box umožňuje automatické opravy textu, kontrolu chyb, ale také vkládání smajlíků.

```
<TextBox Header="Enter Your Text:" PlaceholderText="Text" />
```

Kód 11. – Text Box



Obrázek 39 – Text Box s návrhy psaní



Obrázek 40 – Emoji Picker

Aplikace slouží primárně pro demonstrační účely elementů WinUI 3.0 a možnosti jejich použití. Byla však implementována také jednoduchá interakce s elementy pro ukázkou.

6.2 C# Kód UWP aplikace

V souboru MainPage.xaml.cs se vygeneroval kód pro načtení obsahu uživatelského rozhraní XAML.

Nejprve byla provedena deklarace proměnné Value typu integer, které byla přiřazena hodnota 12. S touto proměnnou se později pracuje.

Níže byl dopsán kód pro event handler pro událost tlačítka při kliknutí. V tomto případě bude zkontrolováno, zda je Progress Ring aktivní. Pokud ano, deaktivuje jej a naopak.

```
int Value = 12;

public MainPage()
{
    this.InitializeComponent();
}

private void Button_Click_Ring(object sender, RoutedEventArgs e)
{
    if (ProgressRing1.IsActive == true)
    {
        ProgressRing1.IsActive = false;
    }
    else
    {
        ProgressRing1.IsActive = true;
    }
}
```

Kód 12. – Deklarace proměnné Value a ovládání Progress Ringu

Stejná situace platí také pro tlačítko, které má na starosti vypnout či zapnout Progress Bar.

```
private void Button_Click_Bar(object sender, RoutedEventArgs e)
{
    if (ProgressBar1.ShowPaused == false)
    {
        ProgressBar1.ShowPaused = true;
    }
    else
    {
        ProgressBar1.ShowPaused = false;
    }
}
```

Kód 13. – Ovládání Progress Baru pomocí tlačítka

Kód pro další tlačítko zvyšuje jak hodnotu Progress Baru, tak hodnoty proměnné Value, která je následně vypisována do Text Blocku.

```
private void Button_Click_Plus(object sender, RoutedEventArgs e)
{
    ProgressBar2.Value++;
    Value++;
    TextBlockProgressBar.Text = "Current value: " + Value.ToString()
;
}
```

Kód 14. – Zvyšování hodnoty Progress Baru pomocí tlačítka

To stejné platí také pro další tlačítko. To ovšem naopak hodnotu odečítá.

```
private void Button_Click_Minus(object sender, RoutedEventArgs e)
{
    ProgressBar2.Value--;
    Value--;
    TextBlockProgressBar.Text = "Current value: " + Value.ToString()
;
}
```

Kód 15. – Snižování hodnoty Progress Baru pomocí tlačítka

Pro zajímavější interaktivitu by bylo též vhodné mít k dispozici tlačítko pro vyresetování hodnot. To přiřadí proměnné s názvem Value, Progress Baru i Text Blocku nuly.

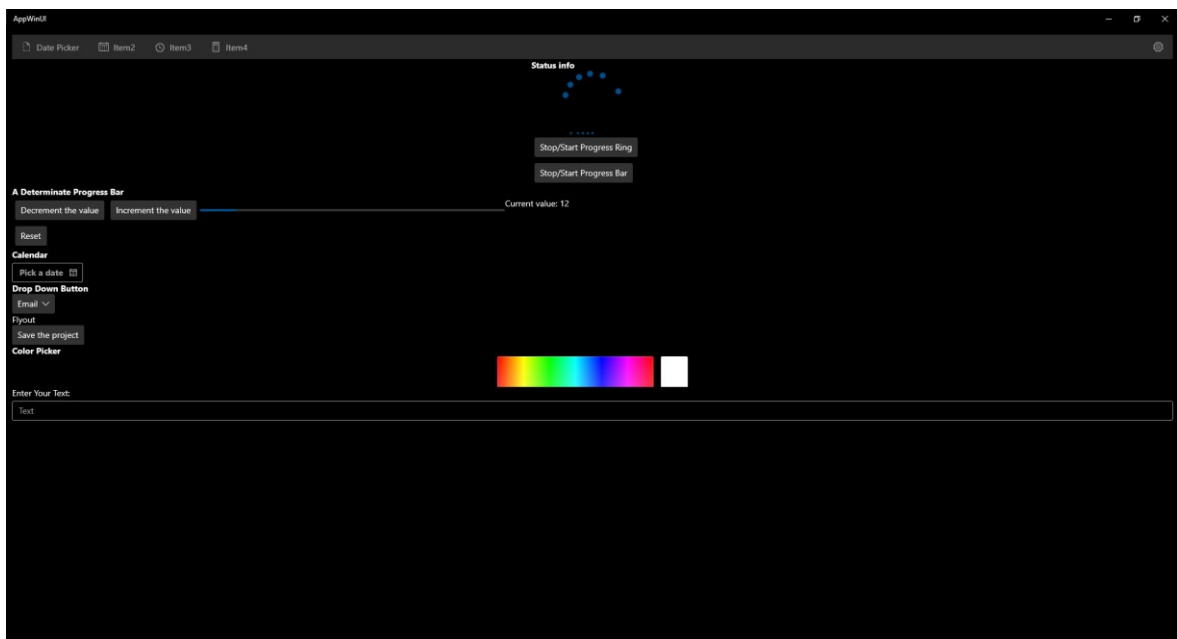
```
private void Button_Click_Reset(object sender, RoutedEventArgs e)
{
    Value = 0;
    ProgressBar2.Value = 0;
    TextBlockProgressBar.Text = "0";
}
```

Kód 16. – Reset hodnoty Progress Baru pomocí tlačítka

6.3 Popis výsledné aplikace

Výsledné aplikace vypadají a fungují totožně, ačkoliv jsou založeny na různých platformách a programovacích jazycích. Aplikace sdílí kód XAML a ve všech případech mají možnost využívat moderních prvků Fluent Design.

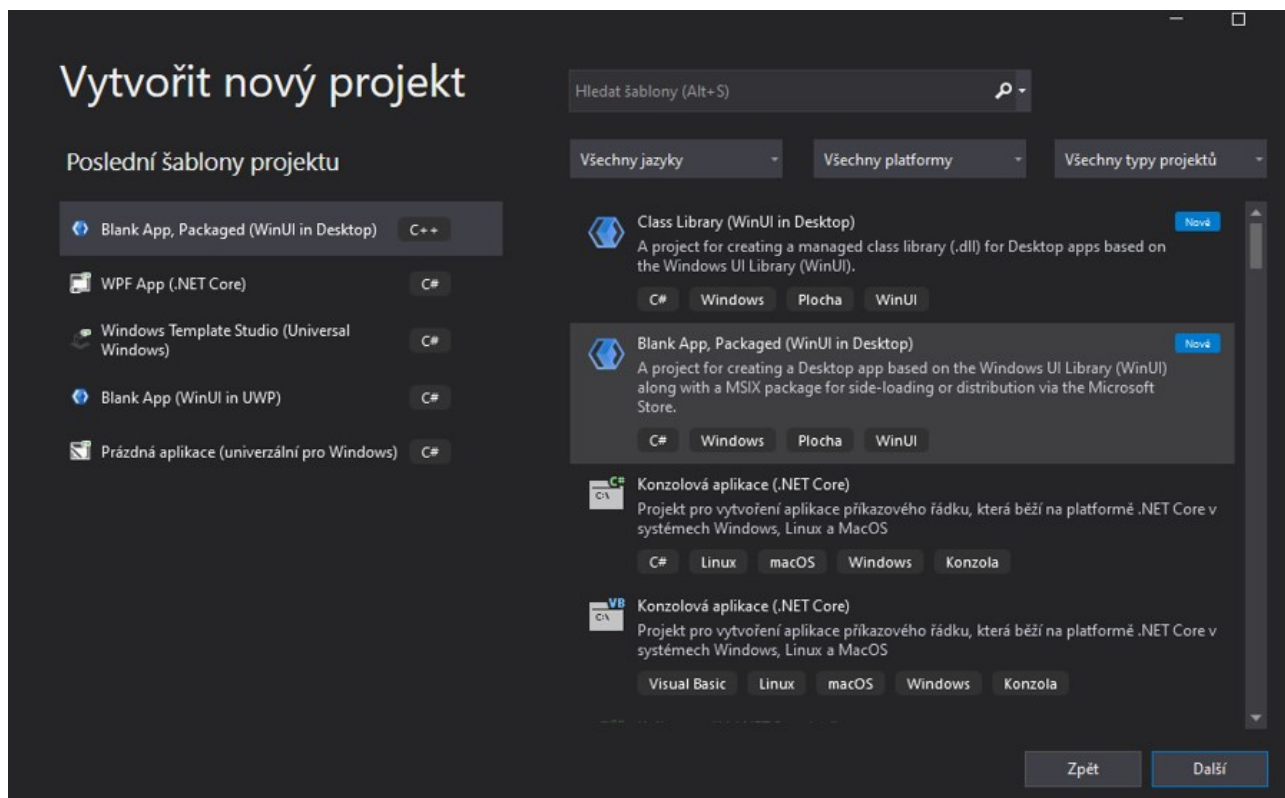
Verze UWP zobrazuje veškeré prvky korektně, včetně animací a responzivního chování. Titulní řádek Aplikací UWP ctí zvolené barevné téma systému a nemění nastavenou výšku ani při maximalizaci přes celou obrazovku. To je vhodné zvláště pro přístroje s dotykovým ovládáním. Tato aplikace je spustitelná jak na počítačích a tabletech s Windows 10, tak Xboxech a v budoucnu také nejspíš na holografických brýlích HoloLens. V podstatě kdekoliv s jádrem Windows 10.



Obrázek 41 – Aplikace UWP v C#

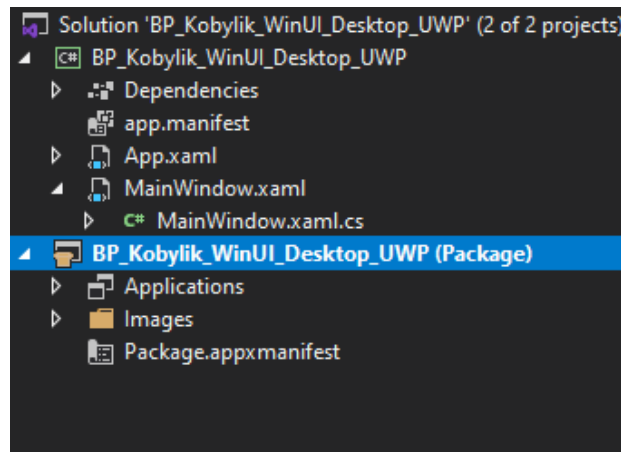
7 KLASICKÁ DESKTOPOVÁ APLIKACE V C#

Aplikace klasické pracovní plochy funguje převážně jako verze UWP. Totožně se také vytváří a vygeneruje podobné soubory. Pro její vytvoření bylo třeba zvolit aplikaci klasické pracovní plochy, jako je znázorněno na obrázku 42.



Obrázek 42 – Založení nového projektu prázdné desktopové C# aplikace s WinUI

Po vytvoření nového projektu se nám vygenerovaly v podstatě totožné soubory, čehož si můžeme všimnout v Solution Exploreru. Opět bylo operováno se soubory `MainWindow.xaml` a `MainWindow.xaml.cs`.

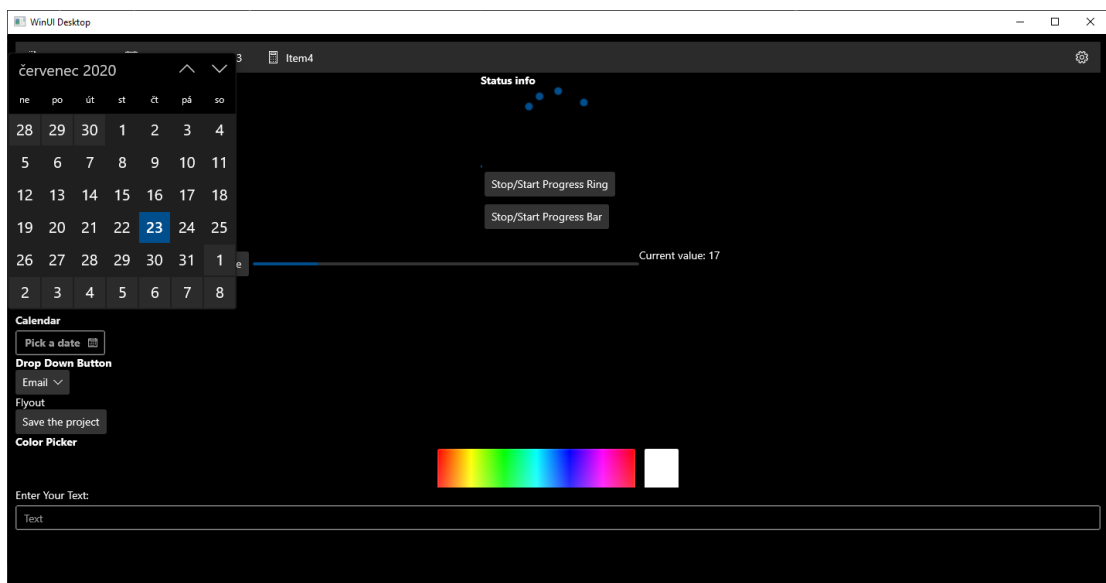


Obrázek 43 – Solution Explorer klasické desktopové aplikace v C#

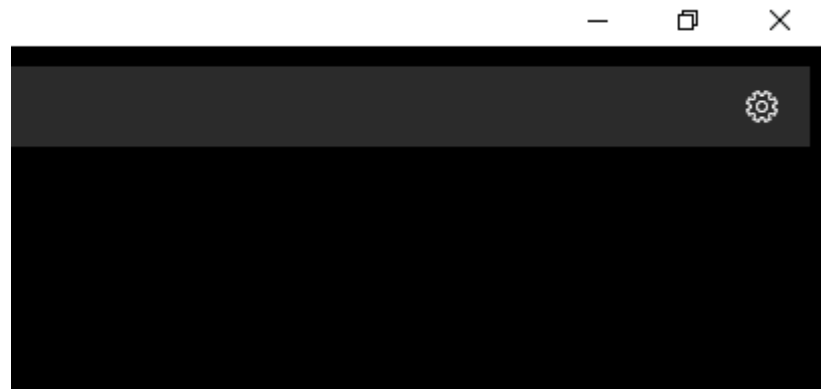
Tvorba aplikace je totožná, jako u UWP verze aplikace. Opět máme k dispozici XAML, ve kterém můžeme využít stejných moderních elementů a stejného rozložení, jako u UWP verze. Je možné využít totožného kódu XAML a C#.

7.1 Popis výsledné aplikace

Výsledná aplikace vypadá a funguje stejně, jako UWP. Bohužel však vykazuje známky nedokončené knihovny WinUI. Některé animace nejsou zobrazeny korektně. Titulní řádek klasických desktopových aplikací převážně nectí nastavené barevné téma a mění svou výšku při maximalizaci.



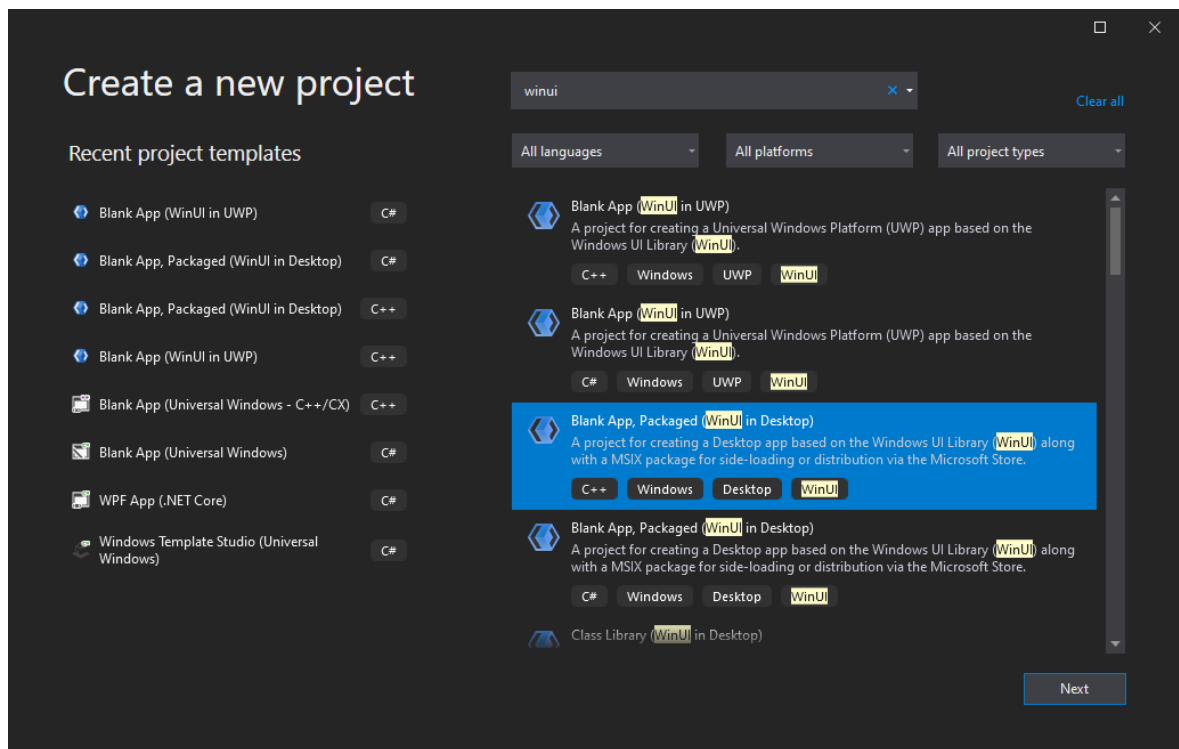
Obrázek 44 – Klasická desktopová aplikace v C#



Obrázek 45 – Detail na titulní řádek klasické desktopové aplikace

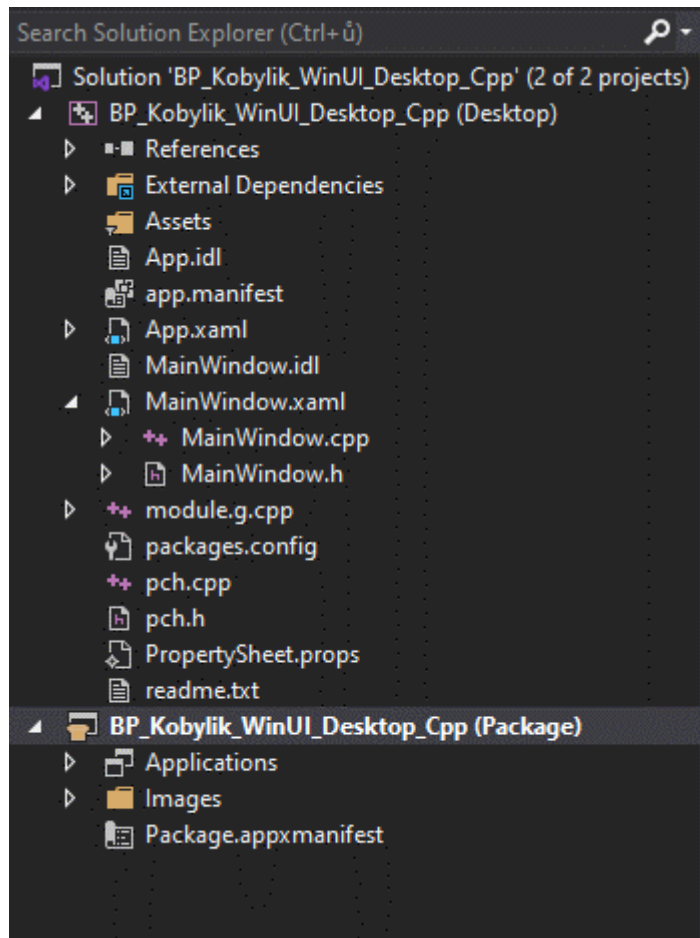
8 KLASICKÁ DESKTOPOVÁ APLIKACE V C++

Nyní bude opět vytvořen nový prázdný projekt pro desktopovou aplikaci s WinUI, ovšem tentokrát napsanou v jazyce C++.



Obrázek 46 – Založení nového projektu prázdné desktopové C++ aplikace s WinUI

Byl opět vygenerován soubor XAML. Tentokrát však byly vygenerovány také další soubory, mezi nimi `MainWindow.cpp` a `MainWindow.h`. Tedy soubor pro kód v C++ a hlavičkový soubor.



Obrázek 47 – Solution Explorer klasické desktopové aplikace v C++

Nejprve bude třeba provést deklaraci tlačítek v hlavičkovém souboru.

```
void Button_Click_Ring(Windows::Foundation::IInspectable const& sender,  
Microsoft::UI::Xaml::RoutedEventArgs const& args);  
void Button_Click_Bar(Windows::Foundation::IInspectable const& sender,  
Microsoft::UI::Xaml::RoutedEventArgs const& args);  
void Button_Click_Minus(Windows::Foundation::IInspectable const& sender,  
Microsoft::UI::Xaml::RoutedEventArgs const& args);  
void Button_Click_Plus(Windows::Foundation::IInspectable const& sender,  
Microsoft::UI::Xaml::RoutedEventArgs const& args);  
void Button_Click_Reset(Windows::Foundation::IInspectable const& sender,  
Microsoft::UI::Xaml::RoutedEventArgs const& args);
```

Kód 17. – Deklarace tlačítek v hlavičkovém souboru

8.1 XAML desktopové aplikace v C++

Kód XAML je naprosto totožný s aplikacemi, které byly vytvořeny dříve. Tedy UWP a klasická desktopová (klasické pracovní plochy). Díky tomu je tvorba grafické části aplikace značně jednodušší. Odpadá tvořit prostředí speciálně pro každý jazyk či jinou technologii.

8.2 C++ Kód desktopové aplikace

Aplikace však bude napsána v C++. Ukázky interakcí s elementy jsou tak napsány v tomto jazyce. Nejprve byla opět přidána deklarace proměnné s názvem Value typu integer. Hodnota byla opět nastavena na 12.

```
int Value = 12;
MainWindow::MainWindow()
{
    InitializeComponent();
}

int32_t MainWindow::MyProperty()
{
    throw hresult_not_implemented();
}

void MainWindow::MyProperty(int32_t /* value */)
{
    throw hresult_not_implemented();
}
```

Kód 18. – Deklarace proměnné Value

K elementům v kódu XAML je třeba přistupovat, jako k funkcím. V následujícím kódu se kontroluje, zda je Progress Ring aktivní či nikoliv. Pokud je aktivní, tlačítkem se vypne. Pokud je deaktivován, naopak se zapne.

```
void MainWindow::Button_Click_Ring(Windows::Foundation::IInspectable
const& sender, Microsoft::UI::Xaml::RoutedEventArgs const& args)
{
    if (ProgressRing1().IsActive() == true)
    {
        ProgressRing1().IsActive(false);
    }

    else
    {
        ProgressRing1().IsActive(true);
    }
}
```

Kód 19. – Ovládání Progress Ringu pomocí tlačítka

Pro následující operaci je třeba provést include `sstream` pro převod čísla integer na textový řetězec `string`.

```
#include "pch.h"
#include "MainWindow.h"
#include "MainWindow.g.cpp"
#include <sstream>
```

Kód 20. – Přidání nového include

V následující části kódu se pracuje s proměnnou `Value` a pomocí tlačítka ubírá její hodnotu. To se také promítá do Progress Baru. Pro zobrazení aktuální hodnoty v aplikaci bylo třeba provést převod číselné hodnoty `Value` typu integer na `string`.

```
void MainWindow::Button_Click_Minus(Windows::Foundation::IInspectable
const& sender, Microsoft::UI::Xaml::RoutedEventArgs const& args)
{
    Value--;
    ProgressBar2().Value(Value);
    std::ostringstream stringstream;
    stringstream << "Current Value: " << Value;
    std::string copyOfStr = stringstream.str();

    std::wstring ws(copyOfStr.begin(), copyOfStr.end());

    TextBlockProgressBar().Text(ws);
}
```

Kód 21. – Snižování hodnoty Progress Baru pomocí tlačítka

V této části se napsal totožný kód, ovšem tentokrát hodnotu inkrementuje.

```
void MainWindow::Button_Click_Plus(Windows::Foundation::IInspectable
const& sender, Microsoft::UI::Xaml::RoutedEventArgs const& args)
{
    Value++;
    ProgressBar2().Value(Value);

    std::ostringstream stringstream;
    stringstream << "Current Value: " << Value;
    std::string copyOfStr = stringstream.str();

    std::wstring ws(copyOfStr.begin(), copyOfStr.end());

    TextBlockProgressBar().Text(ws);
}
```

Kód 22. – Zvyšování hodnoty Progress Baru pomocí tlačítka

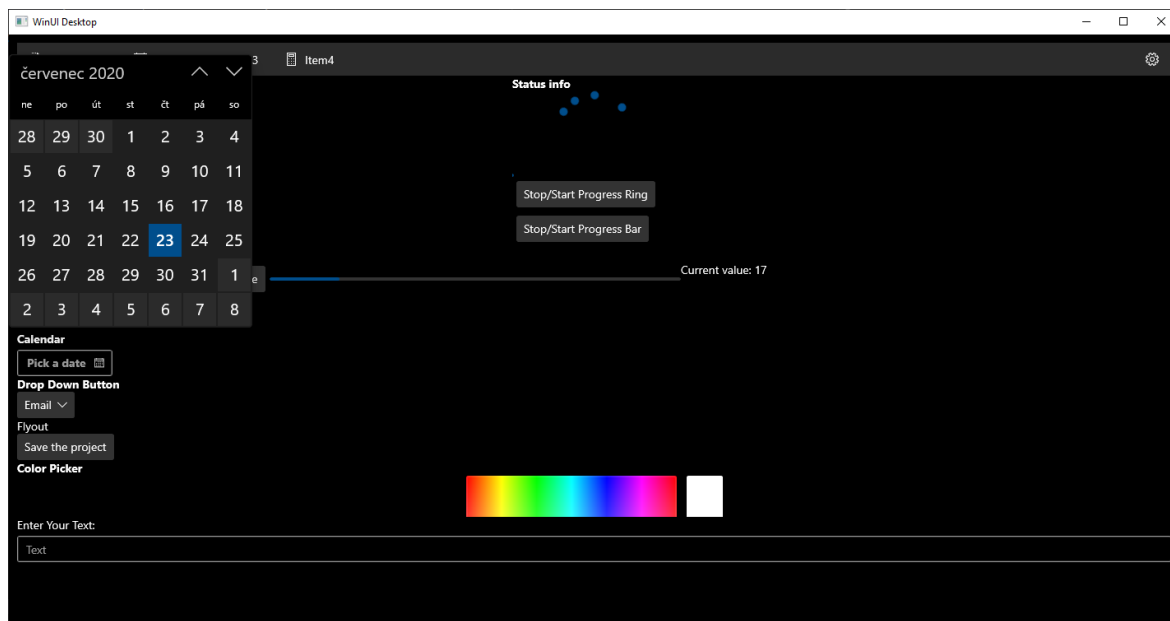
Následující část kódu má na starosti hodnotu Value a Progress Bar vyresetovat.

```
void MainWindow::Button_Click_Reset(Windows::Foundation::IInspectable
const& sender, Microsoft::UI::Xaml::RoutedEventArgs const& args)
{
    Value = 0;
    ProgressBar2().Value(Value);
    TextBlockProgressBar().Text(L"Current value: 0");
}
```

Kód 23. – Reset hodnoty Progress Baru pomocí tlačítka

8.3 Popis výsledné aplikace

Verze desktopové aplikace, jejíž kód byl napsán v C++, vypadá naprosto totožně, jako desktopová v C#, ovšem toto prezentuje možnosti psaní moderních aplikací a možnost výběru programovacího jazyka. Není tedy problém využít moderních prvků za použití C# místo C++.



Obrázek 48 – Klasická desktopová aplikace v C++

ZÁVĚR

WinUI 3.0 je skvělým nástrojem pro tvorbu desktopových aplikací s moderními prvky a funkcemi. Výsledná aplikace vypadá v podstatě stejně, ať je založena na jakékoliv platformě.

Běžný uživatel rozdíl v podstatě nerozezná, ani o jaký druh aplikace se jedná a vývojář má mnohem volnější ruku při tvorbě aplikací. Může si vybrat libovolnou technologii, která mu vyhovuje a zároveň dosáhne na moderní prvky Windows. Jeho aplikace tak nezíská pouze moderní design, ale také moderní funkce a prvky. Zároveň bude mnohem lépe zapadat do designu systému. Pro uživatele to tak bude konzistentní zážitek.

WinUI 3.0 však bohužel stále postrádá některé funkce, elementy či efekty. Vzhledem k tomu, že je knihovna stále ještě ve vývoji lze však očekávat, že se podpora některých z nich objeví v některých z brzkých aktualizací.

Občas se také nedokonale zobrazují přechodové efekty. Ať už se jedná o zvýraznění aktivních prvků či počáteční posun elementů na obrazovku.

V současné době je potřeba k použití knihovny WinUI 3 používat ukázkovou verzi Visual Studio 2019 Preview. Během vytváření ukázkových aplikací však IDE přestalo odpovídat nebo se dokonce několikrát samovolně ukončilo. Toto bude jistě opraveno ve finální verzi.

Přes některá menší negativa se však jedná o velmi ambiciózní projekt Microsoftu, který může vývojářům velmi usnadnit práci při tvorbě nativních a moderních aplikací pro Windows.

SEZNAM POUŽITÉ LITERATURY

- [1] Is WPF dead: the present and future of WPF [online]. [cit. 2012-06-15]. Dostupné z: <http://pragmateek.com/is-wpf-dead-the-present-and-future-of-wpf/>
- [2] Lekce 1 - Úvod do Windows Forms aplikací [online]. [cit. 2012-06-15]. Dostupné z: <https://www.itnetwork.cz/csharp/winforms/c-sharp-tutorial-windows-forms-okenni-aplikace-uvod>
- [3] Win32 Programming: Overview [online]. [cit. 2012-06-15]. Dostupné z: [https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-6.0/aa271855\(v=vs.60\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-6.0/aa271855(v=vs.60)?redirectedfrom=MSDN)
- [4] What's a Universal Windows Platform (UWP) app? [online]. [cit. 2012-06-15]. Dostupné z: <https://docs.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide>
- [5] CHATTERJEE, Ayan. Building apps for the universal Windows platform: explore Windows 10 Native, IoT, HoloLens, and Xamarin. Berkeley, California: Apress, [2017]. ISBN 978-1484226285 [cit. 2012-06-15]
- [6] Compiling Electron Apps for the Windows Store [online]. [cit. 2012-06-15]. Dostupné z: <https://devblogs.microsoft.com/cse/2016/05/08/compiling-electron-apps-for-the-windows-store/>
- [7] Survey Report: Who is the .NET Developer of 2016? [online]. [cit. 2012-06-15]. Dostupné z: <https://www.telerik.com/blogs/survey-report-the-dotnet-developer-of-2016>
- [8] Developer Survey Results 2019 [online]. [cit. 2012-06-15]. Dostupné z: <https://insights.stackoverflow.com/survey/2019>
- [9] Blogging Windows [online]. [cit. 2012-06-15]. Dostupné z: <https://web.archive.org/web/20121027202517/http://blogs.windows.com/windows/b/bloggingwindows/archive/2012/10/25/windows-reimagined-windows8.aspx>
- [10] Microsoft uvedl vlastní tablet s Windows 8, Surface [online]. [cit. 2012-06-15]. Dostupné z: <https://www.lupa.cz/clanky/microsoft-vedl-vlastni-tablet-s-windows-8-surface/>
- [11] Microsoft Is Pulling the Plug on Windows 8 [online]. [cit. 2012-06-15]. Dostupné z: <https://www.statista.com/chart/4225/windows-market-share-by-version/>

- [12] Windows Command-Line: The Evolution of the Windows Command-Line [online]. [cit. 2012-06-15]. Dostupné z: <https://devblogs.microsoft.com/commandline/windows-command-line-the-evolution-of-the-windows-command-line/>
- [13] Stáhnout image disku s Windows 8.1 (soubor ISO) [online]. [cit. 2012-06-15]. Dostupné z: <https://www.microsoft.com/cs-cz/software-download/windows8ISO>
- [14] Zune and iPod Size Comparison [online]. [cit. 2012-06-15]. Dostupné z: <https://www.microsoft.com/security/blog/2007/10/03/zune-and-ipod-size-comparison/>
- [15] Own a piece of tech history: Microsoft Kin One for just \$25 (US Only, Verizon) [online]. [cit. 2012-06-15]. Dostupné z: <https://www.windowscentral.com/tech-history-microsoft-kin-one-just-25-us-only-verizon>
- [16] Microsoft Lumia 950 XL (White) [online]. [cit. 2012-06-15]. Dostupné z: https://www.microsoft.com/en-sg/p/microsoft-lumia-950-xl/8qh5b3zh7s83?ICID=Cat-Mobile-mosaic_linknav-lumia950xl-092016-SG&activetab=pivot%3aoverviewtab
- [17] Update for Skip Ahead members today automatically factory refreshes your Xbox One due to a previous OS corruption [online]. [cit. 2012-06-15]. Dostupné z: <https://mspoweruser.com/update-for-skip-ahead-members-today-automatically-factory-refreshes-your-xbox-one-due-to-a-previous-os-corruption/>
- [18] Why did Microsoft kill the name 'Metro'? [online]. [cit. 2012-06-15]. Dostupné z: <https://www.cnet.com/news/why-did-microsoft-kill-the-name-metro/>
- [19] Microsoft confirms dumping 'Metro' brand from Windows 8 [online]. [cit.2012-06-15]. Dostupné z: <https://www.computerworld.com/article/2505469/microsoft-confirms-dumping--metro--brand-from-windows-8.html>
- [20] The Fluent Design System for Windows app creators [online]. [cit. 2012-06-15]. Dostupné z: <https://docs.microsoft.com/en-us/windows/apps/fluent-design-system>
- [21] Expanding the Fluent design to OneDrive for Android [online]. [cit. 2012-06-15]. Dostupné z: <https://techcommunity.microsoft.com/t5/microsoft-onedrive-blog/expanding-the-fluent-design-to-onedrive-for-android/ba-p/1188176>
- [22] Windows UI Library 3 Preview 2 [online]. (July 2020) [online]. [cit. 2012-06-15]. Dostupné z: <https://docs.microsoft.com/en-us/windows/apps/winui/winui3/>

- [23] About WinUI [online]. [cit. 2012-06-15]. Dostupné z: <https://microsoft.github.io/microsoft-ui-xaml/about.html>
- [24] Windows UI Library (WinUI) [online]. [cit. 2012-06-15]. Dostupné z: <https://docs.microsoft.com/en-us/windows/apps/winui/>
- [25] Windows UI Library 2.0 [online]. [cit. 2012-06-15]. Dostupné z: <https://docs.microsoft.com/en-us/windows/apps/winui/winui2/release-notes/winui-2.0>
- [26] Windows UI Library 2.1 [online]. [cit. 2012-06-15]. Dostupné z: <https://docs.microsoft.com/en-us/windows/apps/winui/winui2/release-notes/winui-2.1>
- [27] Windows UI Library 2.2 [online]. [cit. 2012-06-15]. Dostupné z: <https://docs.microsoft.com/en-us/windows/apps/winui/winui2/release-notes/winui-2.2>
- [28] Windows UI Library 2.3 [online]. [cit. 2012-06-15]. Dostupné z: <https://docs.microsoft.com/en-us/windows/apps/winui/winui2/release-notes/winui-2.3>
- [29] Windows UI Library Roadmap (July 2020) [online]. [cit. 2012-06-15]. Dostupné z: <https://github.com/microsoft/microsoft-ui-xaml/blob/master/docs/roadmap.md>
- [30] Windows UI Library 3 Preview 2 (July 2020) [online]. [cit. 2012-06-15]. Dostupné z: <https://docs.microsoft.com/en-us/windows/apps/winui/winui3/#xaml-controls-gallery-winui-30-alpha-branch>
- [31] Announcing Uno Platform 3.0 – Support for WinUI 3.0 Preview 1 [online]. [cit. 2012-06-15]. Dostupné z: <https://platform.uno/blog/announcing-uno-platform-3-0-support-for-winui-3-0-preview-1/>
- [32] WinUI on Windows 7 - Yes, it's possible with Uno Platform - Uno Platform. Uno Platform - Home [online]. [cit. 2012-06-15] Dostupné z: <https://platform.uno/WinUI-on-Windows7-via-UnoPlatform>
- [33] Preview 2 limitations and known issues [online]. [cit. 2012-06-15]. Dostupné z: <https://docs.microsoft.com/en-us/windows/apps/winui/winui3/#preview-2-limitations-and-known-issues>
- [34] Host UWP XAML controls in desktop apps (XAML Islands) [online]. [cit. 2012-06-15]. Dostupné z: <https://docs.microsoft.com/en-us/windows/apps/desktop/modernize/xaml-islands>

[35] XAML Controls Gallery [online]. [cit. 2012-06-15]. Dostupné z: <https://docs.microsoft.com/en-us/samples/microsoft/xaml-controls-gallery/xaml-controls-gallery/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

IDE Integrated Development Environment

UWP Universal Windows Platform

WPF Windows Presentation Foundation

WinForms Windows Forms

XAML Extensible Application Markup Language

MVVM Model–view–viewmodel

API Application Programming Interface

MFC Microsoft Foundation Class Library

WinRT Windows Runtime

PHP Personal Home Page Tools

UI Uživatelské prostředí

UX Uživatelský zážitek

IME Input Method Editor

SEZNAM OBRÁZKŮ

| | |
|---|----|
| Obrázek 1 – Popularita technologií pro tvorbu aplikací pro Windows 2018 [7]..... | 13 |
| Obrázek 2 – Popularita programovacích jazyků 2019 [8] | 13 |
| Obrázek 3 – Tržní podíl operačních systémů 2013–2015 [11]..... | 15 |
| Obrázek 4 – Textové uživatelské prostředí DOS [12]..... | 16 |
| Obrázek 5 – Uživatelské prostředí systému Windows 1.01 [12]..... | 17 |
| Obrázek 6 – Úvodní obrazovka Windows 8 [13] | 18 |
| Obrázek 7 – Microsoft Zune [14] | 18 |
| Obrázek 8 – Microsoft Kin a jeho prostředí s dlaždicemi [15] | 19 |
| Obrázek 9 – Aplikace ve Windows 8 a multitasking..... | 19 |
| Obrázek 10 – UI Windows Phone / Windows 10 Mobile [16]..... | 20 |
| Obrázek 11 – UI Xbox OS [17]..... | 20 |
| Obrázek 12 – Průhlednosti Fluent Designu ve Windows 10 | 22 |
| Obrázek 13 – Microsoft ToDo na iOS..... | 24 |
| Obrázek 14 – Microsoft Office na iOS | 25 |
| Obrázek 15 – Nekonzistence UI – hranaté Drop Down Menu | 26 |
| Obrázek 16 – Zaoblené UI – hranaté Drop Down Menu..... | 27 |
| Obrázek 17 – Nekonzistence UI – hranaté Kontextové Menu v nabídce Start | 27 |
| Obrázek 18 – Nekonzistence UI – zaoblené Kontextové Menu..... | 28 |
| Obrázek 19 – WinUI 2.2 a zakulacení elementů u Date Pickeru..... | 34 |
| Obrázek 20 – Tlačítko zpět je součástí menu | 35 |
| Obrázek 21 – Numberbox s efektem stínu..... | 37 |
| Obrázek 22 – Menu s efekty stínu | 37 |
| Obrázek 23 – Použití knihovny WinUI [28]..... | 38 |
| Obrázek 24 – Ukázka elementů WinUI [34] | 40 |
| Obrázek 25 – Úvodní obrazovka aplikace UWP Xaml Controls Dallery (WinUI 3 Preview) | 41 |
| Obrázek 26 – Některé funkce či elementy prozatím v aplikaci UWP Xaml Controls Dallery (WinUI 3 Preview) chybí..... | 41 |
| Obrázek 27 – Aplikace UWP Xaml Controls Dallery (WinUI 3 Preview) má ještě své nedostatky. Například zde chybí kopírování kódu. | 42 |
| Obrázek 28 – Založení nového projektu s WinUI 3 Preview | 46 |
| Obrázek 29 – Založení nového projektu UWP aplikace..... | 48 |
| Obrázek 30 – Solution Explorer UWP aplikace | 49 |
| Obrázek 31 – Výsledné horní menu | 50 |
| Obrázek 32 – Progress Ring | 51 |

| | |
|--|----|
| Obrázek 33 – Tlačítka..... | 51 |
| Obrázek 34 – Progress Bar | 53 |
| Obrázek 35 – Date Picker | 53 |
| Obrázek 36 – Drop Down Menu..... | 54 |
| Obrázek 37 – UWP Flyout..... | 55 |
| Obrázek 38 – Color Picker..... | 55 |
| Obrázek 39 – Text Box s návrhy psaní..... | 56 |
| Obrázek 40 – Emoji Picker | 56 |
| Obrázek 41 – Aplikace UWP v C#..... | 59 |
| Obrázek 42 – Založení nového projektu prázdné desktopové C# aplikace s WinUI | 60 |
| Obrázek 43 – Solution Explorer klasické desktopové aplikace v C#..... | 61 |
| Obrázek 44 – Klasická desktopová aplikace v C#..... | 61 |
| Obrázek 45 – Detail na titulní řádek klasické desktopové aplikace | 62 |
| Obrázek 46 – Založení nového projektu prázdné desktopové C++ aplikace s WinUI..... | 63 |
| Obrázek 47 – Solution Explorer klasické desktopové aplikace v C++ | 64 |
| Obrázek 48 – Klasická desktopová aplikace v C++ | 68 |

SEZNAM TABULEK

| | |
|---|----|
| Tabulka 1: Ovládací prvky WinUI 2.0 [25]..... | 32 |
| Tabulka 2: Ovládací prvky WinUI 2.1 [26]..... | 33 |
| Tabulka 3: Ovládací prvky WinUI 2.2 [27]..... | 34 |
| Tabulka 4: Ovládací prvky WinUI 2.3 [28]..... | 36 |

SEZNAM PŘÍLOH

Příloha P1. CD disk.

PŘÍLOHA P I: CD DISK

Přiložené CD obsahuje:

- Bakalářskou práci ve formátu docx: fulltext.docx
- Bakalářskou práci ve formátu pdf: fulltext.pdf
- Vypracované projekty ukázkových aplikací