

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

Hlasový asistent Termix

Ivo Meixner
Moravskoslezský kraj

Lichnov 2018

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

Hlasový asistent Termix

Termix voice assistant

Autoři: Ivo Meixner

Škola: Gymnázium a Střední průmyslová škola elektrotechniky a informatiky, Frenštát pod Radhoštěm, příspěvková organizace, Křižíkova 1258, Frenštát pod Radhoštěm, 744 01

Kraj: Moravskoslezský kraj

Konzultant: Mgr. Richard Štěpán

Lichnov 2018

1 PROHLÁŠENÍ

Prohlašuji, že jsem svou práci SOČ vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Lichnově dne 21.3.2018

Ivo Meixner

2 PODĚKOVÁNÍ

Tímto bych chtěl poděkovat všem lidem, kteří jakkoliv přispěli k úspěšnému vypracování tohoto projektu. Jmenovitě bych chtěl zmínit především svého třídního učitele Mgr. Richarda Štěpána, který mi umožnil pracovat na mém projektu během vyučování programování. Dále bych chtěl poděkovat Martinu Kostelníkovi za jeho spolupráci při realizaci hlasem ovládaných šachů. Klíčovou roli také sehrála internetová komunita Stack Overflow, která mi pomohla najít řešení nemalého množství problémů.

3 ANOTACE

Tento projekt se zabývá hlasovým asistentem a ovládáním počítače pomocí hlasu. Cílem bylo navrhnout a naprogramovat počítačový program, který by běžnému uživateli umožnil ovládat počítač pomocí hlasových příkazů. K vytvoření tohoto programu jsem použil programovací jazyk C#, platformu Microsoft .NET Framework, vývojové prostředí Microsoft Visual Studio a systém pro převod hlasu na text Google Speech API. Můj hlasový asistent je určen jak pro fanoušky moderních technologií ve světě počítačů, tak pro všední uživatele, kteří si chtějí zjednodušit práci. Jedinými předpoklady pro používání tohoto programu jsou základní znalost anglického jazyka a schopnost zapamatovat si potřebné příkazy.

4 KLÍČOVÁ SLOVA

hlasový asistent; ovládání hlasem; programování; C#

5 ANNOTATION

This project is focused on creating a voice assistant and controlling a computer using voice. The goal of this project was to design and create a computer program, which would allow a regular user to control their computer using voice commands. To make program I have used the C# programming language, Microsoft .NET Framework platform, integrated development environment Microsoft Visual Studio and a speech-to-text conversion system Google Speech API. My voice assistant is intended for both fans of modern computer technologies and everyday computer users, who want to ease their work. The only requirements for a user are a basic knowledge of the English language and the ability to remember all the necessary commands.

6 KEYWORDS

voice assistant, voice control, programming, C#

Obsah

1	Prohlášení.....	2
2	Poděkování.....	3
3	Anotace.....	4
4	Klíčová slova.....	4
5	Annotation.....	4
6	Keywords.....	4
7	Úvod.....	6
8	Co je to hlasový asistent.....	7
9	Historie.....	8
10	Můj hlasový asistent.....	10
11	Čím se můj hlasový asistent liší od Cortany.....	11
12	Uživatelské rozhraní.....	11
	12.1 Aktivační tlačítko.....	12
	12.2 Převod řeči na text v reálném čase.....	12
	12.3 Zobrazovač aktuálního jména asistenta.....	12
	12.4 Historie příkazů a odpovědí.....	12
13	Příkazy.....	13
	13.1 Regulární výrazy.....	13
	13.2 Konkrétní regulární výrazy.....	14
	13.3 Změna jména asistenta.....	15
	13.4 Ukončení hlasového asistenta.....	15
	13.5 Změna citlivost aktivačního příkazu.....	16
	13.6 Příkaz pro řešení matematických problémů.....	17
14	Tabulka příkazů.....	18
15	Závěr.....	19
16	Seznam obrázků a tabulek.....	20

7 ÚVOD

Téma hlasový asistent jsem si pro svůj projekt zvolil na základě mé záliby v různých druzích umělé inteligence a všech věcech, jenž s ní souvisí. Zároveň se dlouhodobě zajímám o ovládání digitálních zařízení hlasem a podobný projekt jsem již v minulosti realizoval.

Tehdy se ovšem jednalo o poměrně jednoduchý program, který měl k dispozici pouze malé množství příkazů, takže ovládání nebylo příliš pohodlné. Dále bylo velmi problematické a nespolehlivé jakékoliv diktování textu. V tomto projektu jsem se rozhodl tyto problémy vyřešit a umožnit uživateli pohodlnější práci s počítačem.

Bylo nutné najít způsob, jak spolehlivěji převádět řeč na text. Ukázalo se, že této technologii vládne společnost Google, která veřejně nabízí své služby. Rozhodl jsem se tedy využít jejich systém. Oproti systému společnosti Microsoft, který jsem použil pro vytvoření svého předchozího hlasového asistenta, systém Google Speech API vyžaduje stabilní připojení k internetu a jeho konfigurace je značně složitější. Toto byla ovšem daň, kterou jsem byl ochoten podstoupit, jelikož kvalita a spolehlivost systému společnosti Google byla dramaticky vyšší, což mi umožnilo vytvořit mnohem kreativnější a komplexnější hlasové příkazy. Například příkaz použitý pro ovládání šachových figurek by byl naprosto neproveditelný s použitím systému společnosti Microsoft.

V následujících kapitolách postupně proberu veškeré prvky mého hlasového asistenta a podrobně vysvětlím funkci jednotlivých příkazů.

8 CO JE TO HLASOVÝ ASISTENT

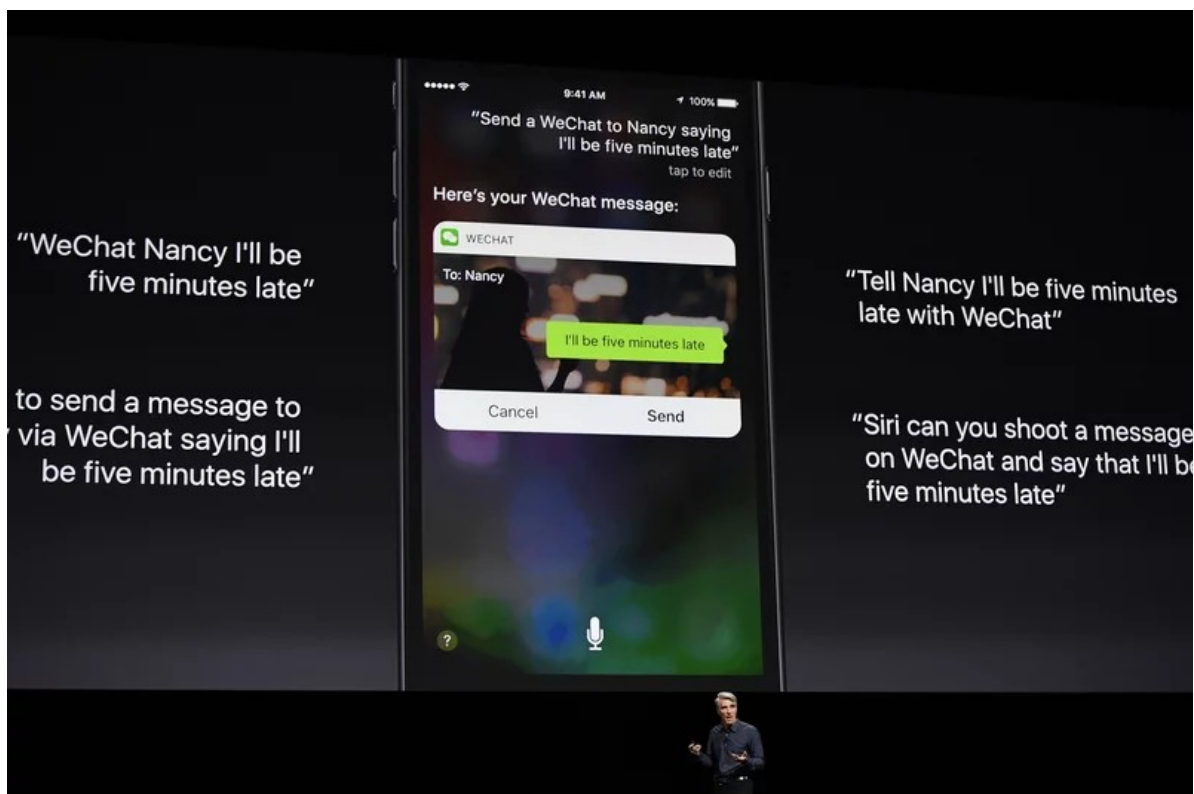
Hlasový asistent je program, který umožňuje uživateli ovládat dané zařízení pomocí hlasových příkazů. Většina lidí zná tuto technologii ze světa mobilních telefonů. V poslední době se však začínají objevovat i jiná zařízení, které je možno ovládat hlasem.



Obr. 1: Používání hlasového asistenta na mobilním zařízení – Dostupné z: https://performancein.com/assets/uploads/img/news/2017/12/crop/Untitled_design_51896x504.jpg

9 HISTORIE

Prvním skutečně populárním hlasovým asistentem byla Siri v roce 2011. Jednalo se o ženského hlasového asistenta, jenž byl zabudován do mobilního operačního systému iOS společnosti Apple. Umožňoval ovládat Apple iPhone pomocí hlasu, což byla funkce, jakou nikdy předtím žádný jiný mobilní telefon spolehlivě nezvládal. Siri se stala velmi populární, protože usnadňovala a urychlovala vyhledávání informací na internet, vytváření poznámek, přehrávání hudby a mnoho dalšího.



Obr. 2: Oficiální prezentace Siri společnosti Apple – Dostupné z: <https://imagesvc.timeincapp.com/v3/mm/image?url=https%3A%2F%2Ftimedotcom.files.wordpress.com%2F2016%2F06%2Fgettyimages-539911842-2.jpg&w=800&q=85>

Později se z hlasových asistentů na mobilních zařízeních stal velký trend a tak společnost Google vytvořila Google Now, později známý jako Google Assistant, hlasového asistenta pro mobilní operační systém Android, a po nějakém čase se přidala i společnost Microsoft s hlasovou asistentkou Cortana pro Windows Phone.

Netrvalo dlouho a hlasoví asistenti se rozšířili i mimo mobilní zařízení. Nyní jsou hlasoví asistenti zabudováni i do operačních systémů pro stolní počítače. Microsoft Windows nabízí Cortanu a Apple macOS obsahuje Siri. Zájem o hlasové asistenty na stolních počítačích není ovšem zdaleka tak vysoký jako na mobilních zařízeních, takže jejich vývoj probíhá velmi pomalu.

Microsoft Windows se prakticky nedají pomocí Cortany ovládat. Je možné, že ji někdo v každodenním životě používá, ale dle mého názoru se jedná pouze o jakousi zajímavost, kterou se Microsoft rozhodl do Windows zabudovat. Ony totiž Windows obsahovaly jistou formu

hlasového asistenta již od Windows Vista, ale v té době se skutečně nedalo mluvit o pořádně funkční technologii. Příkazy byly velmi omezené a rozpoznávací schopnosti činily tohoto hlasového asistenta téměř nepoužitelným. Dnešní Cortana je sice o něco lepší, ale stále se se svými konkurenty nemůže pořádně srovnávat.

Nedávno se na trhu hlasových asistentů objevil nový konkurent, Amazon Echo. Poprvé se jedná o dedikovaného hlasového asistenta, neboli zařízení, které samo o sobě nic jiného nedělá. To znamená, že nemá žádné jiné využití. Je to čistě hlasový asistent, nelze jej ovládat ručně. Někdo by tento tah mohl považovat za velmi odvážný, ale společnosti Amazon se rozhodně vyplatil. Později Amazon vydal menší verzi tohoto zařízení, která se nazývá Amazon Echo Dot. Hlasový asistent, který je v těchto zařízeních obsažen se nazývá Alexa. Jedná se tedy o dalšího ženského hlasového asistenta. Hlavním účelem je plánování událostí, vytváření různých seznamů, přehrávání hudby a provádění dalších každodenních činností. Popularita tohoto zařízení stoupá každým dnem, takže je dost možné, že toto je cesta, kterou se budou takzvané chytré domácnosti v blízké budoucnosti ubírat.



Obr. 3: Různé varianty produktu Amazon Echo – Dostupné z: <https://edge.alluremedia.com.au/m/g/2018/01/amazon-echo-range-Recovered-2.png>

10 MŮJ HLASOVÝ ASISTENT

Když jsem za tomto projektu začal pracovat, měl jsem se svým hlasovým asistentem trochu jiné záměry. Od začátku byl sice založený na Google Speech API, ale původně se mělo pouze jednat o mnohem jednodušší Pythonový skript, jehož účelem bylo na základě hlasových příkazů ovládat Linuxový terminál. Z tohoto důvodu jsem mu dal pracovní název Termix a jelikož jsem na žádný lepší název nepřišel, zůstal jsem u toho původního.

Narozdíl od populárních hlasových asistentů, u toho mého nemá název projektu žádný vliv na to, jak se samotný hlasový asistent oslovuje a jaký příkaz se používá k jeho aktivaci. To znamená, že pokud se vám název Termix nelíbí, není to vůbec problém, protože asistenta si můžete pojmenovat, jak budete chtít. Kdyby jste chtěli, můžete mu klidně dát jméno Siri, Cortana nebo Alexa, volba je čistě na vás. Ve výchozí konfiguraci je jméno nastaveno na Assistant, aby si nikdo nemohl stěžovat.

Celý hlasový asistent je založený na platformě .NET Framework, což znamená, že je možné jej používat pouze v rámci operačního systému Microsoft Windows. Apple macOS má Siri a lidí, kteří by chtěli ovládat Linux hlasem není mnoho, takže bych toto nepovažovat za žádné zásadní omezení.

Dalším omezením je požadavek mít nainstalovaný balík Google Cloud SDK a nutnost být přihlášený do vývojářského Google účtu. Bez toho není aktuálně možné přistupovat k systému Google Speech API. Existuje možnost vygenerovat soubor obsahující přístupové údaje k účtu, který se potom používá pro komunikaci s Google Speech API systémem, ale vzhledem k tomu, že zatím neplánuji svého hlasového asistenta nijak šířit, nepovažoval jsem to za vhodné z bezpečnostních důvodů.

Nastavení týkající se jména asistenta, citlivosti aktivačního příkazu a zdali se mají nebo nemají číst odpovědi asistenta nahlas jsou uloženy na daném počítači a zůstávají uloženy i po vypnutí a zapnutí programu. Pokud chce uživatel nastavení vyresetovat, musí k tomu použít příslušný příkaz.

11 ČÍM SE MŮJ HLASOVÝ ASISTENT LIŠÍ OD CORTANY

Microsoft Windows obsahují hlasového asistenta jménem Cortana. Proč jsem tedy vytvářel vlastního hlasového asistenta pro operační systém Microsoft Windows? Odpověď je jednoduchá, nebyl jsem ani zdaleka spokojený s možnostmi, které Cortana nabízí.

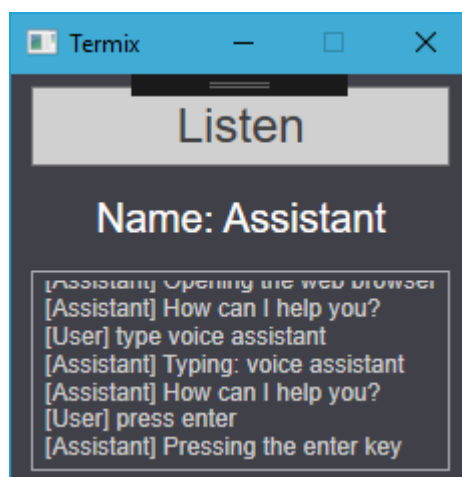
Svého hlasového asistenta jsem mimo jiné navrhl tak, aby umožňoval práci s počítačem i lidem, kteří mají problém ovládat klávesnici či myš. Důvod může být různý, ale zpravidla se jedná o namožení zápěstí nebo zranění libovolné části ruky, což má za následek značné ztížení práce s počítačem. Spousta lidí si dnes již nedokáže bez počítače představit život, takže poranění zápěstí pro ně představuje zásah do jejich každodenního života.

Proto jsem se rozhodl, že můj hlasový asistent by měl lidem usnadňovat i práci s různými aplikacemi, ne jen vyhledávat informace na internetu. Především jsem se pak zaměřil na ovládání prohlížeče, protože ten je v dnešní době nepostradatelný.

12 UŽIVATELSKÉ ROZHŘANÍ

Vzhledem k tomu, že se jedná o hlasového asistenta, rozhodl jsem se navrhnout uživatelské rozhraní co nejjednodušeji. Program je určen pro ovládání hlasem, takže nejsou potřeba žádná zbytečná tlačítka ani textová pole. V tomto ohledu se můj hlasový asistent příliš neliší od těch populárních, protože i jejich uživatelská rozhraní bývají zpravidla jednoduchá.

Uživatelské rozhraní mého hlasového asistenta se skládá ze čtyř částí: aktivační tlačítko, převod řeči na text v reálném čase, zobrazovač aktuálního jména asistenta a historie příkazů a odpovědí na ně.



Obr. 4: Grafické uživatelské rozhraní hlasového asistenta

12.1 Aktivační tlačítko

Jak název napovídá, jedná se o tlačítko, které slouží k aktivaci hlasového asistenta. Za normálních okolností, když všechno funguje správně, jej není třeba vůbec používat. Jeho účelem je umožnit uživateli aktivaci hlasového asistenta v situaci, kdy asistent nereaguje na své jméno. Taková situace může nastat když se aktivační citlivost nastaví na příliš nízkou hodnotu a asistent tudíž nezachytí uživatelův pokus o jeho aktivaci.

12.2 Převod řeči na text v reálném čase

Jedná se o textové pole, které se po aktivaci asistenta zobrazí na místě aktivačního tlačítka. Ve chvíli, kdy je příkaz dokončen, textové pole zmizí a opět se objeví aktivační tlačítko.

Toto textové pole v reálném čase zobrazuje, jaká slova hlasový asistent uživateli rozuměl. Zobrazená slova jsou čistě orientační a pokud uživatel nevidí to, co říká, neznamená to ještě nutně, že příkaz nebude nakonec rozpoznán správně. Převod řeči na text v reálném čase není příliš přesný a je třeba počkat až na finální zpracování celého příkazu pro získání nejpřesnějšího převodu. Částečný převod je zobrazován pouze jako indikace správné funguje asistenta, aby uživatel viděl, že se něco děje.

12.3 Zobrazovač aktuálního jména asistenta

Tato položka je skutečně velmi jednoduchá a není na ní nic k vysvětlování. Jednoduše se zde zobrazuje jméno, na které v danou chvíli hlasový asistent reaguje a pomocí kterého jej uživatel může aktivovat.

12.4 Historie příkazů a odpovědí

Jedná se o další textové pole. Zde se zobrazuje finální výsledek převodu řeči na text. Hlavním účelem tohoto políčka bylo zjišťování, proč určité příkazy nefungují. Často se stává, že převod řeči na text selže a výsledek neodpovídá příkazu uživatele. V tom případě se zde uživatel podívá a vidí, že určité slovo je špatně a může se pokusit vyslovit jej zřetelněji. Z osobní zkušenosti však musím říct, že určité kombinace slov jsem nebyl nikdy schopen vyslovit tak, aby byly rozpoznány.

Dále se zde zobrazují odpovědi asistenta. Zpravidla se jedná pouze o potvrzení toho, že vám rozuměl. Odpovědi na některé příkazy obsahují i užitečné hodnoty, o které by uživatel nechtěl přijít. Příkladem je příkaz na vyhodnocení matematického problému. Mnohem důležitější roli má však toto textové pole pokud uživatel zakáže hlasové odpovědi. V takovém případě je historie příkazů a odpovědí jedině místo, kde se může dozvědět, jak asistent příkaz vyhodnotil.

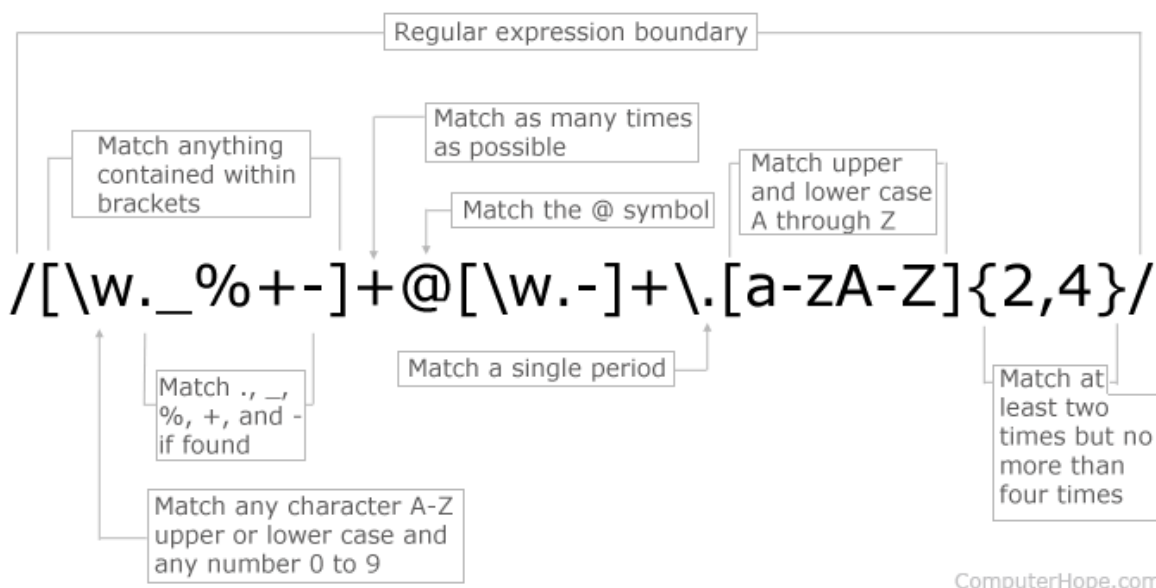
13 PŘÍKAZY

Abych zajistil, co největší pohodlí při používání mého hlasového asistenta, musel jsem zařídit, že budou příkazy správně rozeznány i když uživatel nebude doslovný. Naučit se spoustu příkazu slovo od slova by bylo náročné a otravné, takže jsem musel najít způsob, jak se tomuto problému vyhnout. Řešením jsou regulární výrazy (v angličtině: regular expressions, zkracováno jako regex).

13.1 Regulární výrazy

Tento systém je již poměrně starý, ale osvědčený. Umožňuje efektivně získávat data z textových řetězců na základě takzvaných vzorů, které určují požadovaný formát řetězce. Jedná se o tak komplikovaný systém, že nebudu zbytečně zabíhat do detailů. Takto například vypadá regulární výraz pro vyhodnocení, zda řetězec obsahuje emailovou adresu.

Regular Expression E-mail Matching Example



Obr. 5: Ukázkový regulární výraz – Dostupné z: <https://www.computerhope.com/jargon/r/regular-expression.gif>

13.2 Konkrétní regulární výrazy

V celém kódu mého hlasového asistenta se nachází velmi velké množství těchto regulárních výrazů, avšak zdaleka nejvyšší koncentrace je v části, kde jsou definovány příkazy. Zde je ukázka, jak tato část kódu vypadá.

Na první pohled můžou tyto výrazy vypadat poměrně náhodně, ale opak je pravdou. Jejich správné užití ve velmi náročné a neuvěřitelně snadno se v nich udělá chyba. Najít chybu v těchto, dle mého názoru velmi nepřehledných, výrazech je potom otázka klidně i desítek minut, protože zásadní chyba nemusí být na první pohled vůbec zřejmá.

```
// Assistant
RegisterCommand("do nothing|don't do anything|stop listening|never mind|nevermind", _ => Speak("ok"));
RegisterCommand("(?:close(?:yourself|the assistant)|shut(?:yourself|the assistant)?down)", ActionAssistantShutdown);
RegisterCommand("(?:change(?:your|the)?(?:name|activation command)|rename(?:yourself)?to(?:.+)", ActionAssistantRename);
RegisterCommand("@(increase|decrease)(?:the)?(?:voice)?activation sensitivity(?:by(?:\d+(?:\.\d+|%)?))?", ActionChangeActivationSensitivity);
RegisterCommand("(enable|disable)(?:the)?(?:voice)?feedback|speech synthesis)", ActionSetVoiceFeedback);
RegisterCommand("reset(?:the)?assistant(?:settings|options|configuration)", ActionResetSettings);

// Operating system
RegisterCommand("close(?:the(?:active)?|this)?window", ActionCloseWindow);
RegisterCommand("open(?:my|the)?(documents|music|pictures|videos|downloads|desktop)(?:directory|folder|library)?", ActionOpenUserDirectory);
RegisterCommand("open(?:the)?calculator", ActionOpenCalc);
RegisterCommand("open(?:the)?(?:MS|ms)?paint", ActionOpenPaint);
RegisterCommand("open(?:the|a|my)?(?:web)?browser(?:a)?(?:new)?(?:web)?browser window", ActionOpenWebBrowser);
RegisterCommand("open(Google|YouTube|Wikipedia|Facebook|Twitter|Twitch|Tumblr|Discord|GitHub)", ActionOpenWebpage);

// Keyboard
RegisterCommand("(?:type|write)(.+)", ActionType);
RegisterCommand("scroll down", ActionScrollDown);
RegisterCommand("scroll up", ActionScrollUp);
RegisterCommand("press(?:the)?(?:.+)?(?:key)?(?:zero|one|two|three|four|five|six|seven|eight|nine|ten|\\d+)(?:times|\\|)?", ActionPressKey);

// Problem solving - offline
RegisterCommand("@how much is(?:\d+(?:\.\d+)?)(\+|-|\/)(\d+(?:\.\d+)?)", ActionSolveMathProblem);
RegisterCommand("(?:what is|what's) the time|what time is it", ActionReadTime);
RegisterCommand("(?:tell|read)(?:me)?a joke", ActionReadJoke);

// Problem solving - online
RegisterCommand("open(?:the)?weather forecast", ActionOpenWeatherForecast);
RegisterCommand("how much is(.+)", ActionGoogleMathProblem);
RegisterCommand("search(?:for)?(?:.+)?(?:using|on)(Google|YouTube|Wikipedia)?", ActionSearch);
RegisterCommand("play(?:me)?(?:some(?:thing from)?(?:.+)?(?:a)?(?:YouTube)?mix)?(?:on YouTube)?", ActionPlayYouTubeMix);
RegisterCommand("play(?:a)?(?:YouTube)?video called(?:.+)?(?:video)?(?:on YouTube)?", ActionPlayYouTubeVideo);

// Browser
RegisterCommand("open(?:a)?new tab", ActionBrowserNewTab, AssistantMode.Browser);
RegisterCommand("close(?:the(?:active)?|this)?tab", ActionBrowserCloseTab, AssistantMode.Browser);
RegisterCommand("(?:re)open(?:the(?:last)?|closed)?tab", ActionBrowserReopenTab, AssistantMode.Browser);
RegisterCommand("switch(?:to(?:the)?next)?tab", ActionBrowserNextTab, AssistantMode.Browser);
RegisterCommand("switch to(?:the)?previous tab", ActionBrowserPreviousTab, AssistantMode.Browser);
RegisterCommand("go(?:back|to(?:the)?previous|last) page", ActionBrowserBack, AssistantMode.Browser);
RegisterCommand("go(?:forward|to(?:the)?next) page", ActionBrowserForward, AssistantMode.Browser);

// Chess
RegisterCommand("(?:make a(?:chess)?)?move from(?:[A-H][1-8]) to(?:[A-H][1-8])", x =>
{
    Speak($"Making a chess move from {x[0]} to {x[1]}");
    System.IO.File.AppendAllText("../Chess/log.txt", x[0] + x[1]);
});
```

Obr. 6: Kód definující hlasové příkazy

13.3 Změna jména asistenta

Pro změnu jména asistenta může uživatel používat příkaz „change your name to [nové jméno]“.

Při vykonávání příkazu se nejdřív zkontroluje, zda bylo skutečně zvoleno jméno. Poté se nastaví nové jméno a změny se uloží. Nakonec se aktualizuje informace v textovém poli v okně hlasového asistenta.

```
private void ActionAssistantRename(string[] args)
{
    // User is trying to set the assistant's name to an empty string
    if (string.IsNullOrEmpty(args[0]))
    {
        Speak("The assistant must have a name!");
    }
    else
    {
        Speak("Changing my name to " + args[0]);

        Properties.Settings.Default.Name = args[0];
        Properties.Settings.Default.Save();

        invokeDispatcher(() => LoadAssistantName());
    }
}
```

Obr. 7: Kód příkazu pro změnu jména asistenta

13.4 Ukončení hlasového asistenta

Příkaz pro ukončení hlasového asistenta je velmi jednoduchý. Stačí říct „shut down“ a asistent se vypne. Kód tohoto příkazu je velmi jednoduchý. Prakticky se pouze zavře hlavní okno programu a tím se celý hlasový asistent ukončí.

```
private void ActionAssistantShutDown(string[] args)
{
    Speak("Shutting down the assistant");
    closeMainWindow();
}
```

Obr. 8: Kód příkazu pro ukončení asistenta

13.5 Změna citlivost aktivačního příkazu

Tento příkaz je trochu komplikovanější. Slouží k tomu, aby si uživatel mohl sám definovat, jak jistý si asistent musí být tím, že slyšel svůj aktivační příkaz. Vyšší citlivost znamená, že je jednodušší asistenta aktivovat, ale zároveň je vyšší pravděpodobnost, že jej uživatel aktivuje omylem.

Změnu lze uvést jako desetinné číslo nebo jako hodnotu v procentech. Pokud není uvedena hodnota změny, použije se jako hodnota jeden krok, což je 5%.

Příkaz pro zvýšení citlivosti je „increase activation sensitivity“. Pro snížení citlivosti se na začátku místo „increase“ použije „decrease“. Pokud chce uživatel doplnit příkaz o konkrétní hodnotu, stačí přidat na konec příkazu „by [hodnota změny]“.

Kód tohoto příkazu je trochu složitější, takže jej nebudu vysvětlovat krok po kroku.

```
private void ActionChangeActivationSensitivity(string[] args)
{
    double change = ACTIVATION_SENSITIVITY_STEP;

    if (args[1] != string.Empty)
    {
        change = HelperFunctions.GetDoubleFromString(args[1]);

        if (double.IsNaN(change))
        {
            Speak(args[1] + " is not a valid numeric value");
            return;
        }
    }

    if (args[0] == "decrease")
    {
        change = -change;
    }

    double newSens = Properties.Settings.Default.ActivationSensitivity + change;

    if (newSens > MAXIMUM_ACTIVATION_SENSITIVITY)
    {
        Speak($"Setting the activation sensitivity to its maximum value {MAXIMUM_ACTIVATION_SENSITIVITY}");
        Properties.Settings.Default.ActivationSensitivity = MAXIMUM_ACTIVATION_SENSITIVITY;
    }
    else if (newSens < MINIMUM_ACTIVATION_SENSITIVITY)
    {
        Speak($"Setting the activation sensitivity to its minimum value {MINIMUM_ACTIVATION_SENSITIVITY}");
        Properties.Settings.Default.ActivationSensitivity = MINIMUM_ACTIVATION_SENSITIVITY;
    }
    else
    {
        Speak($"Changing the activation sensitivity to {newSens}");
        Properties.Settings.Default.ActivationSensitivity = newSens;
    }

    Properties.Settings.Default.Save();
}
```

Obr. 9: Kód příkazu pro změnu citlivosti aktivačního příkazu

13.6 Příkaz pro řešení matematických problémů

Ve skutečnosti dokáže tento příkaz řešit pouze velmi jednoduché problémy, jako je sčítání, odčítání, násobení a dělení dvou čísel. Pokud se problém složitější, přenechá jeho řešení Googlu a jeho zabudované kalkulačce.

```
private void ActionSolveMathProblem(string[] args)
{
    if (double.TryParse(args[0], out double leftOperand) && double.TryParse(args[2], out double rightOperand))
    {
        switch (args[1])
        {
            case "+":
                Speak($"{leftOperand} plus {rightOperand} is equal to {leftOperand + rightOperand}");
                break;

            case "-":
                Speak($"{leftOperand} minus {rightOperand} is equal to {leftOperand - rightOperand}");
                break;

            case "*":
                Speak($"{leftOperand} multiplied by {rightOperand} is equal to {leftOperand * rightOperand}");
                break;

            case "/":
                if (rightOperand == 0d)
                {
                    Speak("It is impossible to divide by zero");
                }
                else
                {
                    Speak($"{leftOperand} divided by {rightOperand} is equal to {leftOperand / rightOperand}");
                }
                break;

            default:
                break;
        }
    }
    else
    {
        ActionGoogleMathProblem(new string[] { string.Join(" ", args[0], args[1], args[2]) });
    }
}

private void ActionGoogleMathProblem(string[] args)
{
    Speak("Searching for a solution to " + args[0]);
    HelperFunctions.GoogleSearch(args[0]);
}
```

Obr. 10: Kód příkazu pro řešení matematických problémů

14 TABULKA PŘÍKAZŮ

Seznam příkazů v této tabulce není kompletní a pro každý příkaz může existovat celá řada způsobů, jak jej vyslovit. Vždy je uveden pouze jeden způsob vyslovení daného příkazu pro jednoduchost.

Příkaz	Popis
Do nothing	Slouží k deaktivaci asistenta v případě, že byl omylem aktivován
Shut down	Ukončí celý program hlasového asistenta
Change name to [nové jméno]	Změní jméno asistenta
Increase / decrease activation sensitivity	Zvýšení / snížení citlivosti aktivačního příkazu
Enable / disable voice feedback	Zapnutí / vypnutí čtení odpovědí
Reset assistant settings	Resetování nastavení asistenta do výchozího stavu
Close this window	Zavře aktuálně otevřené okno
Open my documents / music / pictures / ...	Otevře uživatelskou složku přihlášeného uživatele
Open calculator	Otevře kalkulačku
Open MS paint	Otevře program Malování
Open web browser	Otevře nové okno webového prohlížeče
Open Google / YouTube / Wikipedia / ...	Otevře Google / YouTube / Wikipedii ve webovém prohlížeči
Write [text]	Napíše daný text
Press [název tlačítka]	Zmáčkne dané tlačítko klávesnice
How much is [matematický problém]	Vyřešení daný matematický problém nebo najde jeho řešení pomocí Google
What time is it	Přečte aktuální čas
Tell me a joke	Přečte náhodný vtip z databáze
Open weather forecast	Otevře ve webovém prohlížeči předpověď počasí
Search for [text] using Google / YouTube / Wikipedia	Vyhledá daný text pomocí zvoleného vyhledávače
Play something from [interpret]	Začne přehrávat mix písniček od daného hudebního interpreta na YouTube
Play a video called [název videa]	Najde na YouTube video podle názvu a začne jej přehrávat
Make a chess move from [původní pozice] to [nová pozice]	Provede šachový tah

15 ZÁVĚR

Mým cílem bylo vytvořit hlasového asistenta, který by uživateli umožnil ovládat počítač pouze za pomoci hlasu. Přestože jsem svého cíle úplně nedosáhl, naučil jsem se spoustu nových věcí ohledně převodu řeči na text a obecně jsem získal zkušenosti s jazykem C#.

V aktuálním stavu je můj program schopný zpříjemnit uživateli práci s počítačem a do jisté míry mu umožňuje jej ovládat, ale je stále spousta věcí, pro které si uživatel bude muset sáhnout po klávesnici a po myši.

Mám v plánu na tomto projektu nadále pracovat a možná jednoho dne dosáhne stavu, který uživateli umožní odložit klávesnici a myš a spoléhat se čistě na svůj hlas. Je otázkou, zdali je vůbec možné některé úkony provádět pomocí hlasu.

16 SEZNAM OBRÁZKŮ A TABULEK

Obr. 1: Používání hlasového asistenta na mobilním zařízení – Dostupné z: https://performancein.com/assets/uploads/img/news/2017/12/crop/Untitled_design_51896x504.jpg	7
Obr. 2: Oficiální prezentace Siri společností Apple – Dostupné z: https://imagesvc.timeincapp.com/v3/mm/image?url=https%3A%2F%2Ftimedotcom.files.wordpress.com%2F2016%2F06%2Fgettyimages-539911842-2.jpg&w=800&q=85	8
Obr. 3: Různé varianty produktu Amazon Echo – Dostupné z: https://edge.alluremedia.com.au/m/g/2018/01/amazon-echo-range-Recovered-2.png	9
Obr. 4: Grafické uživatelské rozhraní hlasového asistenta.....	11
Obr. 5: Ukázkový regulární výraz – Dostupné z: https://www.computerhope.com/jargon/r/regular-expression.gif	13
Obr. 6: Kód definující hlasové příkazy.....	14
Obr. 7: Kód příkazu pro změnu jména asistenta.....	15
Obr. 8: Kód příkazu pro ukončení asistenta.....	15
Obr. 9: Kód příkazu pro změnu citlivosti aktivačního příkazu.....	16
Obr. 10: Kód příkazu pro řešení matematických problémů.....	17