

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Nástroj pro automatizaci uživatelského rozhraní

Tool for Automation of User Interface

Zadání bakalářské práce

Student: **Jakub Šustr**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Nástroj pro automatizaci uživatelského rozhraní**
Tool for Automation of User Interface

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je navrhnout a sestavit aplikaci, která bude umožňovat automatizované ovládání elementárních prvků uživatelského rozhraní podle předem připraveného scénáře. Aplikace bude podporovat dvě varianty vstupních dat. První varianta bude pracovat na základě zpracování vstupního textového skriptu. Druhá varianta rozšíří funkcionalitu aplikace o identifikaci elementárních prvků uživatelského rozhraní na základě rozpoznávání vstupních snímků.

1. Seznamte se s problematikou automatizovaného ovládání uživatelského rozhraní a prostudujte základní teorii funkce.
2. Prozkoumejte existující nástroje a analyzujte jejich funkce a možnosti v oblasti rozpoznávání a ovládání grafických prvků uživatelského rozhraní.
3. Sestavte seznam požadavků na jednotlivé komponenty aplikace. Prostudujte vhodné existující platformy pro běh aplikace a zvažte možnost využití již existujících knihoven pro rozpoznávání obrazu.
4. Na základě předchozí analýzy vytvořte implementaci aplikace pro obě varianty vstupních dat.
5. Proveďte testování aplikace při ovládání uživatelského rozhraní a zhodnoťte spolehlivost funkce aplikace jak při operaci se vstupním skriptem, tak při rozpoznávání grafických prvků uživatelského rozhraní ze vstupních obrázků.
6. Zhodnoťte dosažené výsledky a porovnejte je s konkurenčními projekty.

Seznam doporučené odborné literatury:

- [1] BISHOP, Christopher M. Pattern recognition and machine learning. New York: Springer, c2006. Information science and statistics. ISBN 0387310738.
- [2] KAEHLER, Adrian a Gary BRADSKI. Learning OpenCV 3. ISBN 978-1-4493-1465-1.
- [3] LI, Kanglin a Mengqi WU. Effective GUI Testing Automation Developing an Automated GUI Testing Tool. Hoboken: John Wiley, 2006. ISBN 978-078-2150-674.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

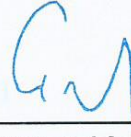
Vedoucí bakalářské práce: **doc. Ing. Jan Platoš, Ph.D.**

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 25. dubna 2017

.....*Jakub Gudek*.....

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla. Především panu doc. Ing. Platošovi Ph.D. a panu Ing. Osadníkovi.

Abstrakt

Cílem bakalářské práce je vyvinout software pro automatizaci uživatelského rozhraní. Účelem tohoto softwaru je umožnit automatizované ovládání elementárních prvků uživatelského rozhraní podle předem připraveného scénáře. Vytvořený software podporuje dvě varianty vstupních dat. Těmito variantami jsou zpracování vstupních dat textového skriptu a zpracování vstupních dat na základě rozpoznávání vstupních obrázků.

Klíčová slova: Automatizer, automatizace, uživatelské rozhraní, testování, software

Abstract

The aim of this bachelor thesis is to develop software for automatization of user interface. Purpose of this software is to allow automated control of elementary elements of user interface according to planned scenario. These variants are processing input data of text script and processing input data based on recognition of input images.

Key Words: Automatizer, automation, user interface, testing, software

Obsah

Seznam použitých zkratk a symbolů	9
Seznam obrázků	10
Seznam tabulek	11
Seznam výpisů zdrojového kódu	12
Úvod	13
1 Automatizace GUI	14
1.1 Proč automatizovat GUI?	14
2 Existující aplikace	15
2.1 Aplikace automatizující GUI OS aplikací	15
2.2 Aplikace automatizující Web	18
2.3 Srovnání	20
3 Použité technologie	22
3.1 Microsoft UI Automation	22
3.2 Windows Input Simulator	22
3.3 Global MouseKeyHook	23
3.4 OpenCV	23
4 UI programu Automatizer	25
4.1 Hlavní menu	25
4.2 Menu nástrojů	26
4.3 Oblast funkcí	28
4.4 Oblast textu skriptu	29
4.5 Spodní lišta	29
5 Funkční celky programu Automatizer	31
5.1 Výběr cílového programu pro automatizaci	31
5.2 Zaměřovací systémy	31
5.3 Nahrávání skriptu	33
5.4 Ukládání/načítání skriptů	35
5.5 Uživatelské funkce	35
5.6 Dynamické volání funkcí	38

6	Rozdíly počítačové a lidské simulace vstupů	41
6.1	Myš	41
6.2	Klávesnice	42
7	Test programu Automatizer	43
7.1	Test automatizace aplikace Poznámkový blok	43
7.2	Test automatizace aplikace Microsoft Word 2016	44
7.3	Výsledek testu	44
	Závěr	45
	Literatura	46
	Přílohy	46
A	Obsah příloženého DVD	47

Seznam použitých zkratek a symbolů

GUI	– Graphical User Interface
CLI	– Command Line Interface
OS	– Operating System
UI	– User Interface
LDTP	– Linux Desktop Testing Project
API	– Application Programming Interface
WPF	– Windows Presentation Foundation
ID	– Identification
HTML	– Hypertext Markup Language
DVD	– Digital Versatile Disc

Seznam obrázků

1	Logo vývojáře AscentialTest	15
2	Logo AutoIt	16
3	Logo Maveryx	17
4	Logo Selenium	18
5	Logo iMacros	18
6	Logo Sahi	19
7	UI programu Automatizer	25
8	Hlavní menu programu Automatizer	25
9	Menu nástrojů programu Automatizer	26
10	Oblast funkcí programu Automatizer	28
11	Textová oblast skriptu programu Automatizer	29
12	Spodní lišta programu Automatizer	29
13	Ukázka červeného ohraničení při výběru cílového programu	31
14	Ukázka modrého ohraničení při výběru elementu	32
15	Ukázka položky soubor v hlavním menu	35
16	Ukázka kategorií uživatelské dokumentace	38
17	Ukázka uživatelské dokumentace funkce putKey()	38
18	Výsledek testu rychlosti dynamického volání funkcí	40
19	Ukázka počítačové simulace nakreslení kříže	41
20	Ukázka lidské simulace nakreslení kříže	42
21	Poznámkový blok s textem pro automatizační test	43
22	Microsoft Word 2016 s textem pro automatizační test	44

Seznam tabulek

1	Základní přehled softwaru AscentialTest	15
2	Základní přehled softwaru AutoIt	16
3	Základní přehled softwaru Linux Desktop Testing Project	16
4	Základní přehled softwaru Maveryx	17
5	Základní přehled softwaru Automatizer	17
6	Základní přehled softwaru Selenium	18
7	Základní přehled softwaru iMacros	18
8	Základní přehled softwaru Sahi	19
9	Srovnání automatizačního softwaru pro OS aplikace	20
10	Srovnání automatizačního softwaru pro web aplikace	20

Seznam výpisů zdrojového kódu

1	První postup dynamického volání	39
2	Druhý postup dynamického volání	39

Úvod

V dnešní uspěchané době, kdy je kladen velký důraz na efektivitu práce je velmi důležité umět šetřit čas. Proto se čím dál tím více rozšiřují a zdokonalují technologie pro automatizaci. Existuje velké množství uživatelských činností, které nevyžadují pozornost uživatele. Veškeré tyto činnosti lze automatizovat a tím šetřit čas.

Ruku v ruce s automatizací softwaru jde také automatizované testování softwaru, kdy lze provádět testy uživatelského rozhraní bez nutnosti neustálého dohledu. Automatizovanými testy softwaru lze šetřit nemalé peníze. Pokud se nalezne chyba při testování ještě předtím, než je software doručen zákazníkovi, lze tak ušetřit nejen peníze, ale také čas, který je potřebný pro opravu softwaru a jeho znovunasazení.

Problémem automatizace se zabývá také tato bakalářská práce, jejím cílem je vývoj automatizačního softwaru. V první kapitole se bakalářská práce zabývá automatizací GUI a otázkami proč je vhodné GUI automatizovat. V další kapitole jsou popsány již existující aplikace zabývající se automatizováním a testováním, na tuto kapitolu navazuje kapitola se srovnáním již existujícího softwaru se softwarem vyvíjeným k této bakalářské práci. Čtvrtá kapitola se zaměřuje na použité technologie při vývoji softwaru k bakalářské práci. Ve čtvrté kapitole je popsáno UI programu Automatizer, který je vyvíjen jako program k této bakalářské práci. Po této kapitole následuje kapitola, kde jsou popsány funkční celky programu Automatizer, a to především výběr cílového programu, zaměřovací systémy programu Automatizer, nahrávání skriptů a jejich pozdější ukládání a načítání, dále uživatelské funkce a jejich dokumentace a dynamické volání funkcí a problémy s tím spojené. Předposlední kapitola se zabývá rozdíly počítačové a lidské simulace vstupu. V poslední kapitole je ukázán automatizovaný test dvou aplikací.

1 Automatizace GUI

Automatizace uživatelského rozhraní je proces, při kterém dochází k automatizování uživatelského rozhraní podle předem daného scénáře. Automatizace šetří čas uživatele tím, že vykonává činnosti sama bez nutnosti uživatelského zásahu. Automatizování jde ruku v ruce s testováním. Testování grafického uživatelského rozhraní je proces testování, při kterém dochází k testování GUI dle předem daných specifikací. Tyto specifikace se nazývají testovací scénáře. Testovací scénáře musí pokrýt veškerou funkčnost aplikace a také plně vyzkoušet všechny možnosti grafického rozhraní. Oproti testování aplikací, které používají pouze textový režim (aplikace příkazového řádku), mají aplikace s grafickým uživatelským rozhráním mnohem více možných kroků pro testování. I relativně jednoduché grafické aplikace mohou mít stovky možných GUI operací, které jsou potřeba otestovat a zahrnout do testovacích scénářů.

Při testování GUI se oproti CLI lze setkat také s dalšími problémy, jako jsou například:

- Sekvenční přístup – kdy nelze vidět určitou část GUI ihned, ale musí se k ní dojít pomocí řady GUI kroků. Například uložení souboru, kdy uživatel musí nejprve najet do menu aplikace, zde vybrat položku v menu "Uložit", poté se otevře nové okno aplikace, kde je nutné vyplnit jméno souboru, vybrat cestu a teprve následující kliknutí na tlačítko "Uložit" uloží daný soubor.
- Dalším problémem je také možnost změn v GUI aplikace, kdy funkčnost aplikace může být zachována, ale tlačítko volající tuto činnost může být přesunuto do jiného menu. V tomto okamžiku je nutné vytvořit odlišný scénář od původního, aby mohl být test proveden.

1.1 Proč automatizovat GUI?

Především automatizování činností šetří čas. Pokud budou automatizovány činnosti, které nepotřebují uživatelský zásah, je možné pracovat rychleji a efektivněji. Při automatizování jedné činnosti může uživatel vykonávat činnost jinou, která vyžaduje jeho pozornost. Dále si lze položit několik otázek při automatizovaném testování softwaru:

1. Je to potřeba? – Je třeba si uvědomit, že to, jaký má uživatel dojem, závisí převážně na uživatelském rozhraní aplikace. Pokud tedy uživateli bude práce s uživatelským rozhraním nepříjemná, s největší pravděpodobností nebude chtít s aplikací pracovat.
2. Není dostačující testování funkčnosti a logiky softwaru? – Při testování funkčnosti a logiky aplikace nelze odhalit chyby, které mohou být v uživatelském rozhraní, jako například nefunkční tlačítka, špatně přiřazené menu a podobně. Všechny tyto chyby mohou být testováním GUI odhaleny.

2 Existující aplikace

Aplikací zabývajících se automatizací uživatelského rozhraní existuje poměrně velké množství. Některé z nich jsou zde uvedeny.

Tyto aplikace lze rozdělit na tyto základní skupiny, dle cílové platformy:

1. Operační systémy

- Windows
- Linux
- OS X
- Android
- iOS
- Multiplatformní

2. Web

2.1 Aplikace automatizující GUI OS aplikací

2.1.1 AscentialTest

AscentialTest[1] je multiplatformní testovací software umožňující plánované testování, správu testovaných dat, vývoj manuálních a automatizovaných testů, vykonávání testů, sledování chyb a jejich reportování.



Obrázek 1: Logo vývojáře AscentialTest

Tabulka 1: Základní přehled softwaru AscentialTest

Podporované platformy	Windows
Vývojář	Zeenyx Software, Inc.
Licence	Proprietary
Webová stránka	http://www.zeenyx.com

2.1.2 AutoIt

AutoIt[2] je skriptovací jazyk podobný BASICu[3], který je navržen pro automatizování GUI na Windows aplikacích a jejich skriptování. Používá kombinaci simulování psaní na klávesnici, pohybu myši a ovládání oken. AutoIt je velmi malý (verze 3 má po rozbalení ~35 MB) a funguje na všech verzích operačního systému Windows. Mezi velkou výhodou patří také to, že skripty lze převést na samostatně spustitelné soubory a poté je spouštět bez nutnosti nainstalovaného AutoIt softwaru.



Obrázek 2: Logo AutoIt

Tabulka 2: Základní přehled softwaru AutoIt

Podporované platformy	Windows
Vývojář	AutoIt Consulting Ltd
Licence	Freeware
Webová stránka	https://www.autoitconsulting.com

2.1.3 Linux Desktop Testing Project

Linux Desktop Testing Project[4] je linuxový GUI testovací software, který si dává za cíl tvořit vysoce kvalitní framework pro testování se špičkovými nástroji. LDTP také podporuje různé programovací jazyky, jako jsou například: Python, Ruby, Java, C# a další.

Tabulka 3: Základní přehled softwaru Linux Desktop Testing Project

Podporované platformy	Linux
Vývojář	nagappan@gmail.com
Licence	GNU LGPL
Webová stránka	https://ldtp.freedesktop.org/wiki/

2.1.4 Maveryx

Maveryx[5] je multiplatformní testovací software pro testování GUI, regresivního testování a daty řízeného testování pro všechny Java a Android aplikace. Používá také některé pokročilé funkce automatizace, například inteligentní rozpoznávání UI objektů a testování klíčových slov.



Obrázek 3: Logo Maveryx

Tabulka 4: Základní přehled softwaru Maveryx

Podporované platformy	Windows, Linux, OS X
Vývojář	Maveryx srl.
Licence	Proprietary
Webová stránka	http://www.maveryx.com

2.1.5 Automatizer

Automatizer je program vyvíjený pro účely této bakalářské práce. Umožňuje nahrávání a znovuspouštění uložených skriptů, ale také ruční psaní skriptů. Podporuje 3 zaměřovací systémy a to souřadnicový systém, systém zaměřování elementů a systémem rozpoznávání v obraze.

Tabulka 5: Základní přehled softwaru Automatizer

Podporované platformy	Windows, částečně web
Vývojář	Jakub Šustr
Licence	Open Source
Webová stránka	-

2.2 Aplikace automatizující Web

2.2.1 Selenium

Selenium[6] je přenosný testovací softwarový framework pro webové aplikace. V tomto softwaru lze nahrávat a poté zpětně přehrávat uživatelské akce. Pokud uživatel nechce, případně neumí psát vlastní skripty. Skripty lze psát v různých programovacích jazycích jako jsou například C#, Java, Perl, PHP, Ruby a další. Testy lze spouštět na většině moderních webových prohlížečů.



Obrázek 4: Logo Selenium

Tabulka 6: Základní přehled softwaru Selenium

Podporované platformy	Windows, Linux, OS X
Vývojář	http://www.seleniumhq.org/about/contributors.jsp
Licence	Apache License 2.0
Webová stránka	http://www.seleniumhq.org

2.2.2 iMacros

iMacros[7] je software umožňující automatizaci webových aplikací a to pomocí nahrání makra a jeho následovného spuštění, ale také pomocí psaní skriptů. iMacros dokáže provádět nejrůznější operace při automatizaci webových aplikací, jako jsou například vyplňování formulářů, vyhledávání textů a podobně. Podporované prohlížeče jsou Internet Explorer, Firefox a Chrome. Makra uložená v iMacros lze také převést do Selenium WebDriver kódu.



Obrázek 5: Logo iMacros

Tabulka 7: Základní přehled softwaru iMacros

Podporované platformy	Windows
Vývojář	IPSWITCH, INC.
Licence	Proprietary
Webová stránka	http://imacros.net

2.2.3 Sahi

Sahi[8] je software pro testování webových aplikací. Umožňuje nahrávání a znovuspouštění marker. Skripty jsou založeny na JavaScriptu. Sahi podporuje většinu moderních prohlížečů. Software má jak placenou Pro verzi tak Open Source verzi, kde chybějí některé funkce z Pro verze.



Obrázek 6: Logo Sahi

Tabulka 8: Základní přehled softwaru Sahi




Podporované platformy	Web
Vývojář	Tyto Software Pvt. Ltd.
Licence	Proprietary, Open Source
Webová stránka	http://sahipro.com

2.3 Srovnání

V následujících tabulkách lze vidět srovnání, kde jsou přehledně zobrazeny informace testovacích softwarů pro aplikace OS a pro web.




Tabulka pro aplikace OS

Tabulka 9: Srovnání automatizačního softwaru pro OS aplikace

	 AscentialTest	 AutoIt	LDTP	 Maveryx	Automatizer
Licence	Proprietary	Freeware	GNU LGPL	Proprietary	Open Source
Nahrávání maker	X	✓	✓	X	✓
Samo-spustitelné skripty	X	✓	X	X	X
Skripty ve známých programovacích jazycích	X	X	✓	X	X
Platforma	Windows, web	Windows	Linux	Windows, Linux, OS X	Windows, částečně Web

Tabulka pro web aplikace

Tabulka 10: Srovnání automatizačního softwaru pro web aplikace

	 Selenium	 iMacros	 Sahi
Licence	Apache License 2.0	Proprietary	Proprietary, Open Source
Nahrávání maker	✓	✓	✓
Skripty ve známých programovacích jazycích	✓	✓	X
Platforma	Internet Explorer, Safari, Opera, Chrome, Firefox	Internet Explorer, Firefox, Chrome	Prohlížeče, které podporují JavaScript

Při výběru určitého automatizačního softwaru nelze určit jasného vítěze, jelikož každý software má svá pro a proti. Rozhodování, který software si uživatel/firma vybere, také záleží na jejich potřebách. Je zde velké množství rozhodnutí, které se musí provést před výběrem softwaru, jako jsou například:

- Je potřeba automatizovat Webové, OS aplikace nebo obojí?
- Znájí testeři již některý z těchto softwarů?
- Jaká je cena softwaru?
- Bude se software používat tak, jak je nebo se bude upravovat?
- Jak rychle se testeři naučí software používat?

Je také dobré si uvědomit, že naučit se používat software trvá určitou dobu, která se může měnit v závislosti na prostředí, ve kterém bude software využíván. Například, pokud jsou testeři seznámeni s určitým programovacím jazykem, může být výhodnější vybrat software, který podporuje psání skriptů v daném jazyce. Dále pokud je potřeba si software upravit potřebám na míru, je výhodnější vybrat software s otevřenou licencí a upravit si jej tak přesně na míru. A v neposlední řadě je také důležité zvážit cenu softwaru, kdy je třeba si uvědomit, zda se cena za software s enormním množstvím funkcí vyplatí, když se při běžném používání bude využívat pouze malé procento z těchto funkcí.

Výběr automatizačního softwaru je nutno pečlivě zvážit, jinak si uživatel/firma může zkomplikovat celý následný proces nasazení, a poté automatizaci či testování softwaru.

3 Použité technologie

K vypracování programu k bakalářské práci byl zvolen programovací jazyk C# s frameworkem .NET[9]. Jedná se o typově bezpečný objektově orientovaný jazyk podobný programovacímu jazyku Java. Dále byly použity některé další frameworky a knihovny jako Microsoft UI Automation[10], Windows Input Simulator[11], Global MouseKeyHook[12], OpenCV[13] a EmguCV[14].

3.1 Microsoft UI Automation

Microsoft UI Automation je framework, který zajišťuje přístup k většině UI prvků na ploše počítače. Microsoft UI Automation lze používat na všech operačních systémech s podporou WPF. Přes tento framework se lze dostat také ke stromové struktuře všech UI prvků, kdy jako kořen tohoto stromu je *desktop*. Dále se dá pomocí tohoto frameworku také zjistit i o jaký prvek se jedná, například tlačítko, editační pole a podobně. V programu k bakalářské práci je tento framework využíván hlavně v těchto bodech:

- Nahrávání skriptu – V tomto bodě je framework využíván při zachycování činností uživatele, pokud je uživatelem tato volba zapnuta, kdy při posunu myši či kliknutí tlačítka na myši, je framework použit pro zjištění elementu na pozici myši a následném získání informací o tomto elementu.
- Vybírání elementu – Při této činnosti si uživatel vybere daný element ručně, a to tak, že pomocí funkce v programu aktivuje vyhledávání elementu a následně na tento element najede myší, aby byly zjištěny informace o tomto elementu.
- Spouštění skriptu – Zde jsou využity zjištěné informace o elementech z předchozích bodů a následně jsou použity pro vyhledání konkrétních elementů ve stromové struktuře. Pokud jsou elementy nalezeny, jsou vráceny pro další zpracování.

Použitím tohoto frameworku při psaní programu se zjednodušuje přístup k prvkům UI.

3.2 Windows Input Simulator

Windows Input Simulator je jednoduché C# rozhraní, které dokáže simulovat vstup klávesnice a myši. Toto rozhraní využívá *SendInput* metody z *Win32* systémové knihovny. Skrze poskytované API lze posílat běžné klávesy a události těchto tlačítek, ale také tlačítka speciální jako například funkční klávesy, dále klávesy jako alt, shift, ctrl a podobně. Lze také posílat události kláves jako jsou události *KeyDown*, *KeyUp* atd. V programu k bakalářské práci je toto rozhraní použito hlavně v těchto bodech:

- Funkce pro simulování vstupu – V programu má uživatel možnost vkládat funkce pro simulaci vstupu z klávesnice a myši ze sady funkcí. Rozhraní je použito jak pro jednodušší funkce typu kliknutí, tak pro složitější funkce typu posílání klávesových zkratk.

- Běh skriptu – V programu je rozhraní také použito, pokud se při běhu skriptu narazí na funkci pro obsluhu vstupu, poté je tato funkce zpracována a pro nasimulování vstupu je použito toto rozhraní.

Toto rozhraní poskytuje snadnější přístup k používání simulace uživatelského vstupu. Při použití tohoto rozhraní není třeba používat *unmanaged* kód z již uvedené systémové knihovny, ale lze použít obvyklý přístup *managed* kódu v programu.

3.3 Global MouseKeyHook

Tato knihovna umožňuje detekovat a nahrávat aktivitu myši a klávesnice, i když aplikace používající tuto knihovnu není v popředí. Knihovna také poskytuje rozhraní pro vytvoření globálních *hooků* na určité události, kdy si poté lze k těmto událostem přiřadit vlastní funkce, které se vyvolají při uskutečnění této události. Knihovna také dokáže vracet události myši a klávesnice v jejich standardní C# podobě, takže lze jednoduše získávat informace, které jsou potřeba například:

- Souřadnice myši
- Stisknutí tlačítek na myši
- Pohyb kolečka na myši
- Stisknutí a uvolnění kláves
- Speciální stavy kláves

Navíc lze při použití rozšířených funkcí pro klávesnici a myš z knihovny zjistit/provést také následující vlastnosti/činnosti:

- Potlačení vstupu
- Časové razítko
- Zjištění v jakém stavu je tlačítko myši (*up/down*)
- Zjištění v jakém stavu je klávesa (*up/down*)

Tato knihovna je v programu hojně zastoupena hlavně při nahrávání skriptu uživatelem, kdy se veškerá uživatelská činnost zaznamenává s pomocí použití této knihovny.

3.4 OpenCV

OpenCV je open source knihovna počítačového vidění a strojového učení. Tato knihovna obsahuje více než 2500 optimalizovaných algoritmů, které lze používat pro různé účely jako například:

- Rozpoznávání tváří

- Identifikování objektů
- Extrahování 3D modelů z objektů
- Skládání obrazů
- Vyhledávání obrazu v obraze
- A mnoho dalšího.

Tato knihovna má rozhraní v těchto programovacích jazycích C++, C, Python, Java a MATLAB. Pro účely psaní programu k bakalářské práci je ovšem zvolen programovací jazyk C#, pro který OpenCV neposkytuje oficiální rozhraní, proto je nutné najít vyhovující *wrapper* na tuto knihovnu.

3.4.1 EmguCV

EmguCV je multiplatformní *wrapper* pro OpenCV knihovnu poskytující API, přes které lze volat z C# (a dalších programovacích jazyků) příslušné funkce z OpenCV.

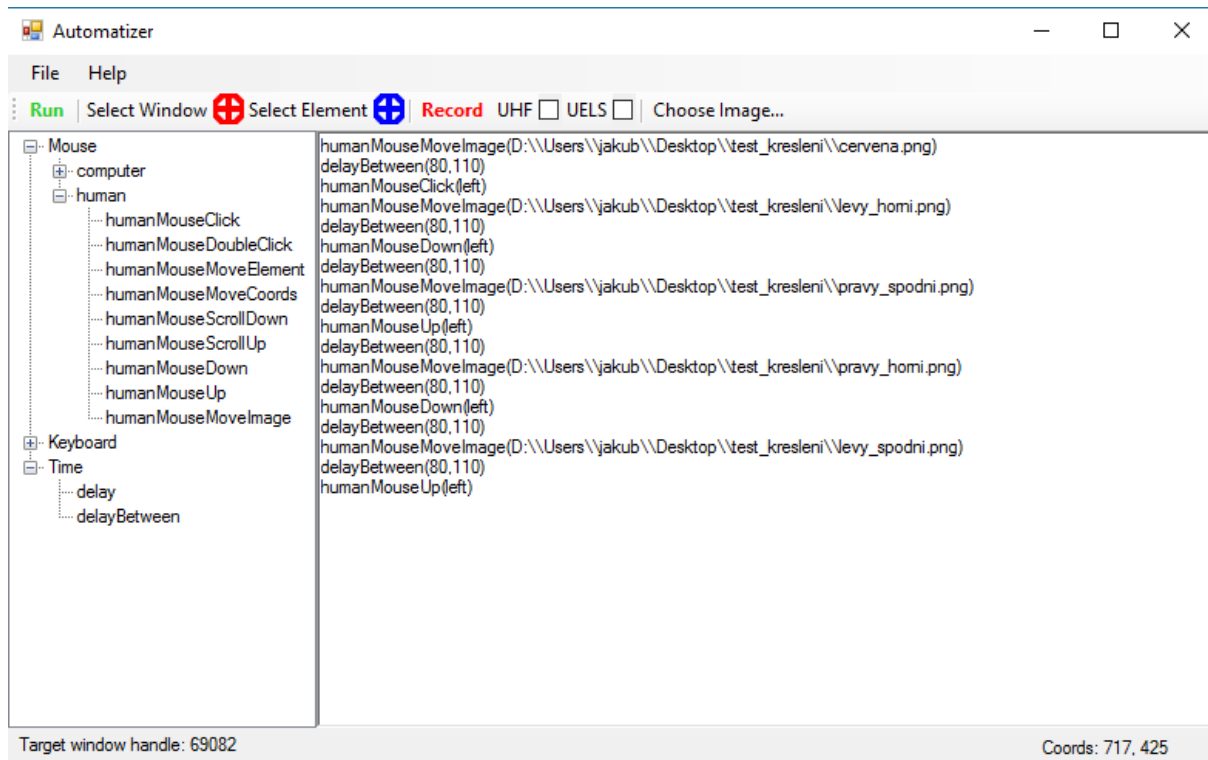
V programu k bakalářské práci je nejdůležitější prvek z OpenCV, potažmo EmguCV, prvek vyhledávání obrazu v obraze. Tento prvek je použit jako jeden ze tří možných vyhledávacích systémů programu. Mezi další využívané prvky z EmguCV patří:

- Skládání obrazu - Tento prvek je využit pro sestavení obrazu z více obrazů oken aplikace, která je cílem pro automatizaci
- Úprava obrazu - Tato funkce je potřeba k úpravě a odstraňování nechtěných částí obrazu

Pro potřeby programu k bakalářské práci je využito tohoto *wrapperu* pro jednodušší spolupráci s knihovnou OpenCV.

4 UI programu Automatizer

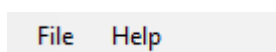
Uživatelské rozhraní je navrženo tak, aby bylo co nejintuitivnější, jak je to jen možné, a zároveň obsahovalo všechny důležité prvky pro snadné ovládání programu.



Obrázek 7: UI programu Automatizer

Dále jsou rozebrány jednotlivé části UI s informacemi, k čemu tyto části slouží.

4.1 Hlavní menu



Obrázek 8: Hlavní menu programu Automatizer

Jedná se o hlavní menu programu Automatizer, ve kterém je výběr ze 2 možností a to Soubor (File), kde se nachází položky menu:

- Otevřít (Open) – Slouží pro načtení uloženého skriptu.
- Uložit (Save) – Slouží pro uložení skriptu.
- Uložit jako (Save as) – Slouží pro uložení skriptu s rozšířenými možnostmi uložení.
- Ukončit (Quit) – Slouží pro ukončení programu Automatizer

A dále se zde nachází menu Pomoc (Help), ve které se nachází položka menu pro otevření dokumentace funkcí.

4.2 Menu nástrojů



Obrázek 9: Menu nástrojů programu Automatizer

V menu nástrojů jsou důležité prvky pro vytváření, spuštění a nahrávání skriptů. Následně je popsána funkcionality jednotlivých prvků.

4.2.1 Spuštění/zastavení skriptu

Run

Toto tlačítko slouží pro spuštění skriptu. Při spuštění skriptu se text změní z původního textu "Spustit" (Run) na text "Zastavit vykonávání" (Stop executing).

Stop executing

Toto tlačítko slouží pro zastavení vykonávání skriptu. Při ukončení skriptu se text změní z textu "Zastavit vykonávání" (Stop executing) na původní text "Spustit" (Run).

4.2.2 Výběr okna

Select Window

Slouží pro výběr okna cílového programu pro automatizaci. Výběr se provádí tak, že se přesune červený zaměřovací kříž na cílové okno. V průběhu výběru okna se ukazuje červené ohraničení kolem okna, nad kterým je právě umístěn kurzor, po uvolnění tlačítka myši se program, nad kterým je kurzor, nastaví jako cíl pro automatizaci.

4.2.3 Výběr prvku

Select Element

Slouží pro výběr prvku UI cílové aplikace. Výběr se provádí tak, že se přesune modrý zaměřovací kříž na cílový prvek. V průběhu výběru okna se ukazuje modré ohraničení kolem prvku, nad kterým je právě umístěn kurzor, po uvolnění tlačítka myši se informace o prvku vloží do okna po psaní skriptu. Tyto informace lze pak využívat v kombinaci s funkcemi, které ve svém názvu obsahují slovo Element.

4.2.4 Spuštění/zastavení nahrávání skriptu

Record

Toto tlačítko slouží pro spuštění nahrávání skriptu. Při spuštění nahrávání se text změní z původního textu "Nahrávat" (Record) na text "Zastavit nahrávání" (Stop recording). Při nahrávání se vkládají informace o událostech na klávesnici, myši a časových událostech do okna pro psaní skriptu.

Stop recording

Toto tlačítko slouží pro zastavení nahrávání skriptu. Při kliknutí na tlačítko se text změní z textu "Zastavit nahrávání" (Stop recording) na původní text "Nahrávat" (Record) a ukončí se vkládání informací o událostech do okna pro psaní skriptu.

4.2.5 Použití lidských funkcí

UHF

Použití lidské funkce (Use Human Functions) jedná se o zaškrtnutou položku. Pokud je zaškrtnuta, používají se při nahrávání událostí funkce se simulací lidského vstupu. Lidský pohyb myši, lidský stisk klávesy a podobně.

4.2.6 Použití prvkového zaměřovacího systému

UELS

Použití prvkový zaměřovací systém (Use Element Location System) jedná se o zaškrtnutou položku. Pokud je zaškrtnuta, používají se při nahrávání událostí funkce, které získávají informace o prvcích a ty se pak použijí pro vyhledání prvku (na rozdíl od souřadnicového systému).

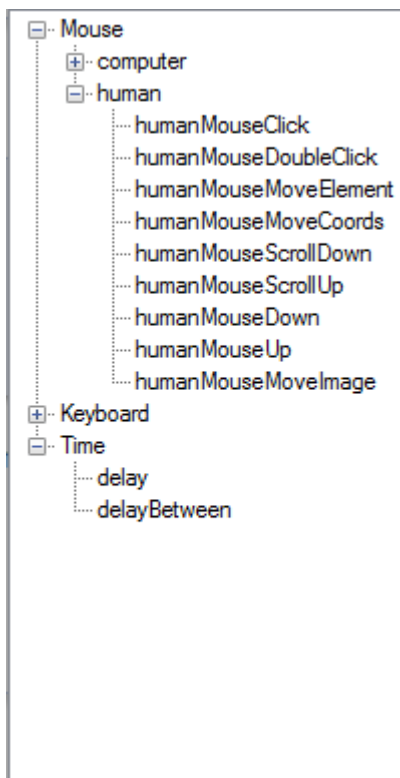
4.2.7 Výběr obrázku

Choose Image...

Vybrat obrázek (Choose Image) jedná se o tlačítko vyvolávající průzkumníka pro vyhledání daného obrázku. Tato funkce se používá v kombinaci s funkcemi, které mají ve svém názvu slovo "obrázek" (image).

4.3 Oblast funkcí

V této oblasti se nachází všechny uživatelsky dostupné funkce. Vyvolání nápovědy pro jednotlivé funkce či sekce se provádí pravým kliknutím. Vložení do textové části se provádí levým dvojklikem.



Obrázek 10: Oblast funkcí programu Automizer

Funkce jsou členěny do 3 kategorií a to:

- Myš (Mouse)
- Klávesnice (Keyboard)
- Čas (Time)

V těchto kategoriích s výjimkou času nalezneme dále podkategorie:

- Počítač (Computer)
- Člověk (Human)

V těchto podkategoriích se již nachází dané automatizační funkce zastupující určitou činnost dle názvu kategorie a podkategorie. Například funkce simulující lidský pohyb myši se souřadnicovým zaměřováním (humanMouseMoveCoords) se nachází v kategorii Myš, podkategorii Člověk.

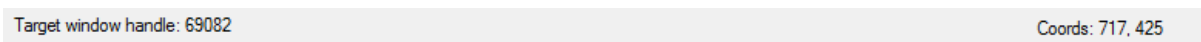
4.4 Oblast textu skriptu

Do oblasti pro psaní skriptu se vkládají automatizační funkce, a to buď z oblasti funkcí nebo automaticky při nahrávání skriptu. Při ukončení běhu skriptu kvůli chyby nebo ukončení uživatelem se označí řádek, kde nastala chyba a v případě uživatelského zastavení 1. řádek v pořadí, který se díky zastavení nevykonal.

```
humanMouseMoveImage(D:\\Users\\jakub\\Desktop\\test_kresleni\\cervena.png)
delayBetween(80,110)
humanMouseClick(left)
humanMouseMoveImage(D:\\Users\\jakub\\Desktop\\test_kresleni\\levy_homi.png)
delayBetween(80,110)
humanMouseDown(left)
delayBetween(80,110)
humanMouseMoveImage(D:\\Users\\jakub\\Desktop\\test_kresleni\\pravy_spodni.png)
delayBetween(80,110)
humanMouseUp(left)
delayBetween(80,110)
humanMouseMoveImage(D:\\Users\\jakub\\Desktop\\test_kresleni\\pravy_homi.png)
delayBetween(80,110)
humanMouseDown(left)
delayBetween(80,110)
humanMouseMoveImage(D:\\Users\\jakub\\Desktop\\test_kresleni\\levy_spodni.png)
delayBetween(80,110)
humanMouseUp(left)
```

Obrázek 11: Textová oblast skriptu programu Automatizer

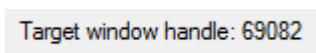
4.5 Spodní lišta



Obrázek 12: Spodní lišta programu Automatizer

Jedná se o jednoduchou lištu s některými praktickými informacemi. Následně jsou popsány jednotlivé prvky lišty.

4.5.1 *Handle* cílového okna



Popisek s informací o *handlu* cílového okna programu k automatizaci.

4.5.2 Informace o běhu skriptu

A small rectangular box with a light gray background and a thin border, containing the text "Run in progress..." in a green, sans-serif font.

Popisek s informací, že skript aktuálně probíhá. Text je umístěn uprostřed spodní lišty, pokud skript právě neprobíhá, text zmizí.

4.5.3 Aktuální souřadnice myši

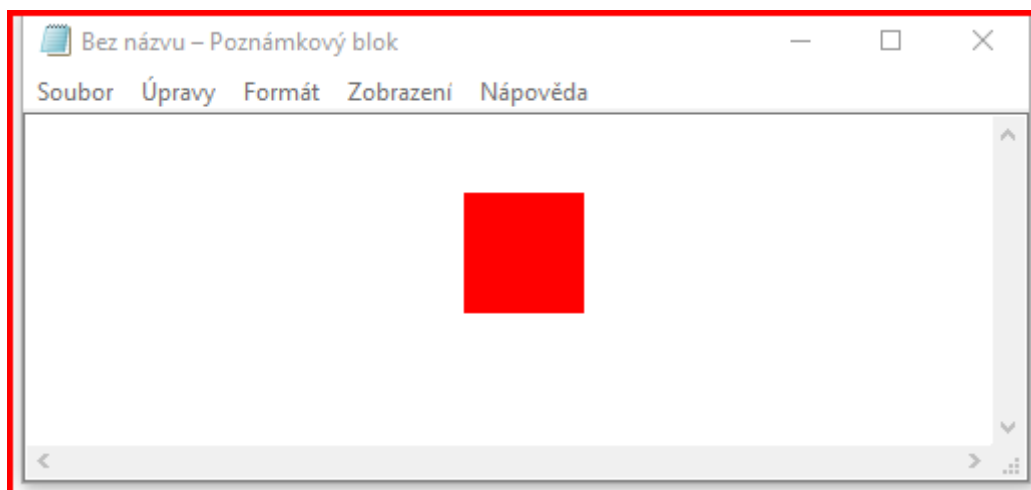
A small rectangular box with a light gray background and a thin border, containing the text "Coords: 717, 425" in a black, sans-serif font.

Popisek s informací o aktuální poloze myši, jedná se o souřadnicový systém X, Y.

5 Funkční celky programu Automatizer

5.1 Výběr cílového programu pro automatizaci

Výběr programu pro automatizaci je poměrně jednoduchá záležitost, pro uživatele stačí, když přetáhne červený zaměřovací kříž na cílové okno programu. Při posouvání myši se ukazuje červené ohraničení a červený čtverec uprostřed okna programu, na kterém je právě kurzor.



Obrázek 13: Ukázka červeného ohraničení při výběru cílového programu

Z programátorského hlediska je výběr okna řešen tak, že v okamžiku, kdy uživatel začne přetahovat *pictureBox*, změní se kurzor na červený zaměřovací kříž. Při tažení myši se periodicky zjišťují souřadnice myši. Poté, co se zjistí souřadnice myši, předají se funkci, která vyhledá příslušný element na zjištěných souřadnicích. Tento element se předá funkci pro vyhledání *rootu* daného elementu a tato funkce vrací jako výsledek získaný *root* element. Z tohoto *root* elementu se zjistí vlastnost *handle*, která se pošle funkci pro vykreslování červeného ohraničení.

Vykreslování červeného ohraničení probíhá tak, že pomocí *handlu*, který je získán pomocí výše uvedených funkcí, se zjistí o elementu vlastnost *boundingRectangle* neboli ohraničující obdélník. Po zjištění souřadnic a rozměrů tohoto obdélníku se přesunou a zviditelní červené formuláře (forms) bez okrajů (borderless form), kterým se nastaví příslušná velikost a souřadnice. Tyto formuláře se neustále přesouvají na správné umístění, dokud není funkce pro výběr cílového programu ukončena. Poté se zneviditelní a zruší se jim vlastnost *topMost*.

Při uvolnění tlačítka myši se funkce pro výběr cílového programu ukončí a kurzor změní zpět na původní. Dále se také uloží vlastnost *handle* cílového programu do paměti a přepíše se popisek s informací o cílovém *handlu* ve spodní liště programu.

5.2 Zaměřovací systémy

V programu k bakalářské práci jsou využity 3 způsoby zaměřování, tyto způsoby jsou:

- Souřadnicový systém
- Systém vyhledávání elementů
- Systém vyhledávání obrazu

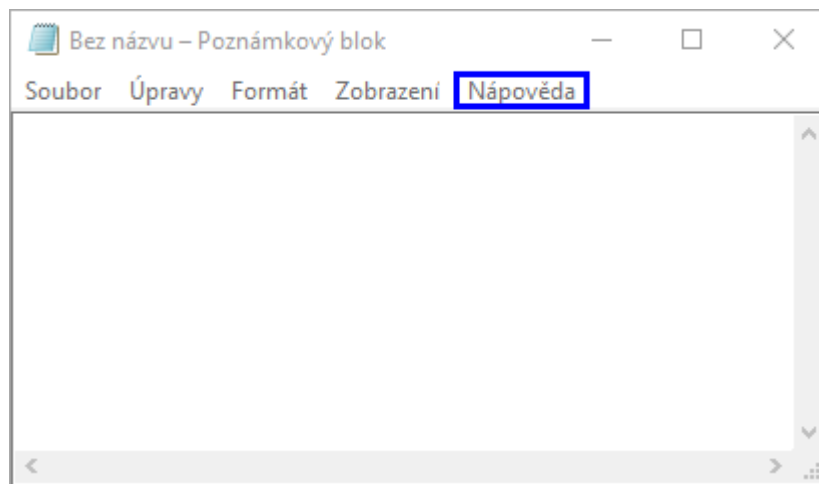
5.2.1 Souřadnicový systém

Tento systém je založen na souřadnicovém systému, kdy uživatel zapisuje funkce, které vyžadují polohu souřadnice ve formátu X, Y.

Z programátorského hlediska je tento systém nejjednodušší, stačí pouze získané souřadnice použít a poslat jednotlivým funkcím, které je vyžadují.

5.2.2 Systém vyhledávání elementů

Tento systém je založený na vyhledávání elementů. Podobně jako při výběru programu pro automatizaci i tento systém využívá funkce z menu nástrojů. Pro výběr elementu stačí, když uživatel přetáhne modrý zaměřovací kříž na daný element.



Obrázek 14: Ukázka modrého ohraničení při výběru elementu

Vyhledávání elementu z programátorského hlediska funguje obdobně jako při výběru programu pro automatizaci s rozdílem, že se nevyhledává *root*, ale vrací se element zjištěný pomocí souřadnic.

Tento element se pak předá stejné funkci pro vykreslování ohraničení jako v případě výběru programu s tím rozdílem, že ohraničení, které se vykresluje, se zobrazuje modrou barvou a již není zobrazen vycentrovaný čtverec.

5.2.3 Systém vyhledávání obrazu

Tento systém je založený na vyhledávání obrazu v obraze. Pro tento zaměřovací systém jsou využity knihovny OpenCV, respektive EmguCV.

Funkce pro vyhledávání v obrazu jsou z hlediska uživatele vcelku jednoduché. Pro uživatele stačí, když si vybere danou funkci z oblasti funkcí a poté pomocí vybrání obrázku (Choose image) z menu nástrojů vybere obraz, který chce v cílovém programu vyhledat.

Z programátorského hlediska je pro vyhledávání nutné určit hledaný obrázek, metodu vyhledávání, práh citlivosti a obraz, ve kterém se vyhledává. Po zavolání funkce z knihovny EmguCV pro vyhledávání v obraze je vrácena shoda, pokud je nalezena.

Obraz, ve kterém se vyhledává, se musí nejdříve získat a poté upravit. V programu se nejprve zjistí podle uloženého *handle* v paměti ID procesu cílového programu a pomocí funkce se vyhledají všechny *handle* daného procesu. Poté se vezme výtisk obrazovky (print screen). Následně se zjišťují informace o všech *handle* procesu, kdy se pomocí funkcí zjistí jejich vlastnost *boundingRectangle* a souřadnice. Po zjištění těchto informací se vytvoří nový průhledný obrázek s velikostí, která je vypočítána na základě rozmístění oken daného programu na obrazovce. Pokud se nejedná o extrém (jedno okno programu je umístěno v levém horním rohu obrazovky/obrazovky a druhé v pravém dolním rohu), jsou výrazně sníženy rozměry obrázku a tím pádem dochází k lepší optimalizaci (menší obraz k prohledávání => rychleji nalezená shoda). Následně se nakopírují obdélníky ze souřadnic z původního obrázku do tohoto nového obrázku. Tento nový obrázek je spolu s informací o posunu souřadnic odeslán do funkce pro vyhledávání z EmguCV knihovny.

Pokud je shoda nalezena, zpracuje se její výsledek a získají se informace se souřadnicemi a velikostí obdélníku. Tyto informace jsou poté odeslány do funkce dle uživatelova výběru, aby následně byla funkce provedena.

5.3 Nahrávání skriptu

V programu je také umožněno nahrávat skripty. Spuštění a zastavení nahrávání je umožněno skrze tlačítka v menu nástrojů. Při nahrávání skriptů dochází k tomu, že uživatelské vstupy z klávesnice, myši, ale také časové vstupy jsou zaznamenány, zpracovány a následně vloženy do textové oblasti. Pro nahrávání skriptů jsou zde 2 další možnosti nastavení nahrávání:

- Použití počítačových funkcí – jedná se o výchozí možnost nahrávání, kdy se nesimuluje lidský vstup. V tomto případě se při nahrávání skriptu nevytvářejí intervalové prodlevy a používají se funkce nesimulující lidský vstup.
- Použití lidských funkcí – tato možnost se použije, pokud je zaškrtnuto políčko pro použití lidských funkcí (UHF). Při této možnosti se nahrávají také prodlevy mezi vstupy a používají se funkce pro simulaci lidského vstupu, jako je lidský pohyb myši a podobně.

Další možnosti nastavení je možnost výběru lokačního systému, kdy na výběr jsou opět 2 možnosti:

- Použití souřadnicového systému – jedná se o výchozí možnost. Při této možnosti se zaznamenává pozice vstupů myši pomocí souřadnicového systému, tedy používají se funkce,

které pohybují s myší podle zadaných souřadnic: pohyb myši na zadané souřadnice (mouseMoveCoords) a lidský pohyb myši na zadané souřadnice (humanMouseMoveCoords).

- Použití lokačního systému elementů – tato možnost se aktivuje zaškrtnutím políčka pro použití lokačního systému elementů (UELS). Při této možnosti se používají funkce pro zaměření elementů, těmi jsou například tlačítko, okno, menu a podobně.

5.3.1 Bezpečnost nahrávání

Při psaní programu a jeho části pro nahrávání je jistým překvapením, že funkcionality nahrávání se obejde bez zvýšení oprávnění programu. Nástroj pro řízení uživatelských účtů (UAC) nepožaduje zvýšení oprávnění v žádném kroku nahrávání a to i přesto, že aplikace Automatizer není v popředí. Aplikace tedy dokáže nahrávat vstup uživatele ačkoliv není vidět.

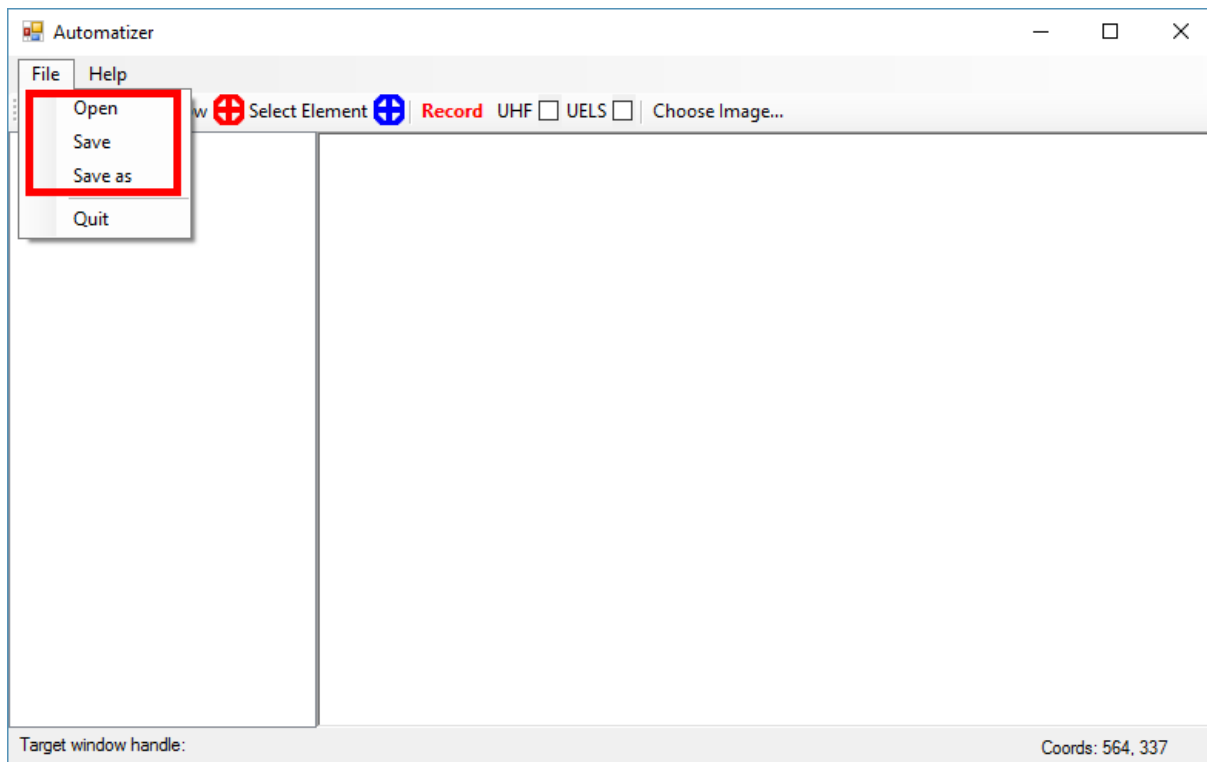
Lze si uvědomit, že některé základní body, které dokáže nahrávání skriptu a potažmo rozpoznávání v obraze v programu Automatizer provádět, mají schopnost:

- Nahrávání uživatelských vstupů (klávesnice, myš, čas)
- Ukládání snímků obrazovky

Tak lze při mírných úpravách program Automatizer považovat za určitý druh spywaru.

5.4 Ukládání/načítání skriptů

V programu Automatizer je možné jednoduše ukládat skripty, a také následně načítat uložené skripty. Ukládání a načítání se provádí přes hlavní menu programu Automatizer. Po kliknutí na příslušnou položku menu se může vyvolat dialogové okno pro vybrání cesty pro načtení/uložení skriptu. Přípona skriptů tohoto programu je "atmtz".



Obrázek 15: Ukázka položky soubor v hlavním menu

5.5 Uživatelské funkce

Uživatelské funkce se nacházejí v oblasti funkcí. Tyto funkce slouží k automatizaci cílového programu. Uživatel je může buďto ručně napsat do textové oblasti nebo je automaticky vygenerovat dvojklikem v oblasti funkcí. Více informací o daných funkcích lze získat v uživatelské dokumentaci 5.5.4.

Dále jsou rozebrány kategorie funkcí a jejich jednotlivé funkce.

5.5.1 Myš

Uživatelské funkce v kategorii Myš zastupují funkce, které provádějí simulace vstupu myši. Tato kategorie je dále rozdělena na 2 podkategorie a to:

- Počítačovou (computer) simulaci

- Lidskou (human) simulaci

Počet funkcí je v obou kategoriích stejný. Tyto funkce ovšem vykonávají svou činnost rozdílně. Zatímco počítačově simulované funkce se nestarají o uživatelské zpoždění či obdobné události způsobené uživatelem (například postupné tažení myši) a nesimulují toto chování, funkce lidské simulují také zpoždění a ostatní lidské chování, cílem lidských funkcí je tedy také to, aby nebylo poznat, že danou činnost neprovádí uživatel sám, ale počítač.

Kategorie myš obsahuje tyto funkce:

- **Kliknutí myši (mouseClick a humanMouseClicked)** – jedná se o funkce, které simulují kliknutí (stisknutí a uvolnění) tlačítka myši
- **Dvojklik myši (mouseDoubleClick a humanMouseDoubleClick)** - funkce, které simulují dvojklik na myši
- **Pohyb myši na element (mouseMoveElement a humanMouseMoveElement)** - funkce, které přesouvají kurzor na daný element
- **Pohyb myši na souřadnice (mouseMoveCoords a humanMouseMoveCoords)** - funkce, které přesouvají kurzor na dané souřadnice
- **Rolovat dolů (mouseScrollDown a humanMouseScrollDown)** - funkce, které provádí rolování směrem dolů
- **Rolovat nahoru (mouseScrollUp a humanMouseScrollUp)** - funkce, které provádí rolování směrem nahoru
- **Stisknutí tlačítka myši (mouseDown a humanMouseDown)** – funkce, které provádí simulaci stisknutí tlačítka myši
- **Uvolnění tlačítka myši (mouseUp a humanMouseUp)** – funkce, které provádí simulaci uvolnění tlačítka myši
- **Pohyb myši na obrázek (mouseMoveImage a humanMouseMoveImage)** – funkce, které přesouvají kurzor na daný obrázek

5.5.2 Klávesnice

Uživatelské funkce v této kategorii zastupují funkce, které provádějí simulace vstupu klávesnice. Tato kategorie je dále rozdělena na 2 podkategorie a to:

- Počítačovou (computer) simulaci
- Lidskou (human) simulaci

Počet funkcí je v obou kategoriích stejný. Tyto funkce ovšem vykonávají svou činnost rozdílně. Zatímco počítačově simulované funkce se nestarají o uživatelské zpoždění či obdobné události způsobené uživatelem a nesimulují toto chování, funkce lidské simulují také zpoždění a ostatní lidské chování, cílem lidských funkcí je tedy také to, aby nebylo poznat, že danou činnost neprovádí uživatel sám, ale počítač.

Kategorie klávesnice obsahuje tyto funkce:

- **Napsání textu (`textType` a `humanTextType`)** – funkce, které simulují napsání textu
- **Stisk kombinace kláves (`keyCombination` a `humanKeyCombination`)** – jedná se o funkce, které simulují stisk kombinace kláves
- **Stisk klávesy (`putKey` a `humanPutKey`)** – jedná se o funkce, které simulují stisknutí dané klávesy
- **Stisk a držení klávesy (`putKeyDown` a `humanPutKeyDown`)** – funkce, které simulují stisk a držení dané klávesy
- **Uvolnění klávesy (`putKeyUp` a `humanPutKeyUp`)** – funkce simulující uvolnění dané klávesy

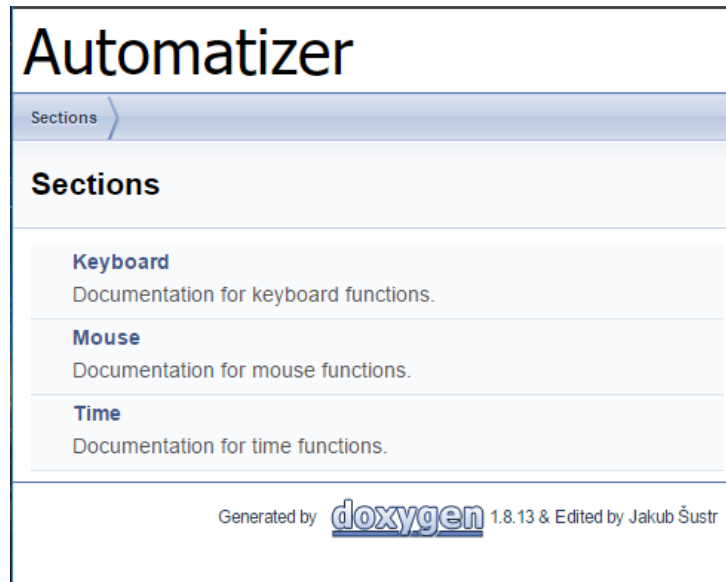
5.5.3 Čas

V této kategorii se vyskytují funkce simulující čas. Jedná se o tyto funkce:

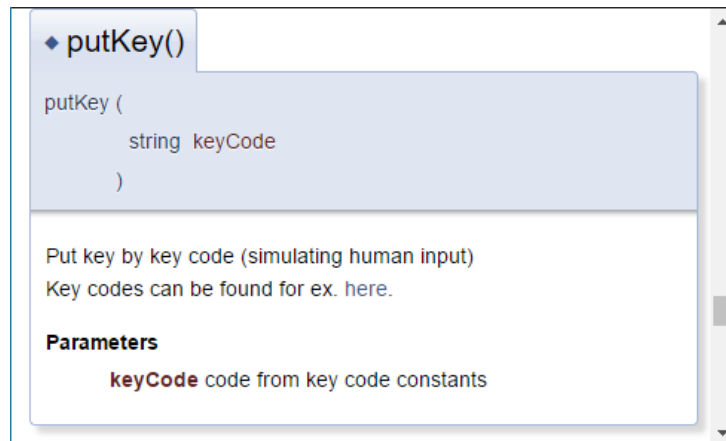
- **Zpoždění (`delay`)** – jedná se o funkci, která vloží do simulace zpoždění
- **Zpoždění v rozmezí (`delayBetween`)** – jedná se o funkci, která vkládá do simulace zpoždění v daném rozmezí

5.5.4 Uživatelská dokumentace

Všechny funkce všech kategorií a podkategorií mají také svou dokumentaci. Dokumentaci lze vyvolat pravým kliknutím na příslušnou funkci či kategorii. V uživatelské dokumentaci lze zjistit všechny potřebné informace k dané funkci jako jsou vstupy dané funkce, jak je získat a podobně. Dokumentace je ve formátu HTML a její otevření se provádí pomocí výchozího webového prohlížeče.



Obrázek 16: Ukázka kategorií uživatelské dokumentace



Obrázek 17: Ukázka uživatelské dokumentace funkce putKey()

5.6 Dynamické volání funkcí

Veškeré funkce, které jsou dostupné uživatelům, je nutno volat dynamicky. Není totiž možné dopředu, a tedy v době překladu, vědět, které funkce uživatel bude chtít používat a kdy.

První postup použitý pro dynamické volání je postup, kdy při každém zavolání funkce je nejprve načtena daná *Assembly*, poté jsou zjištěny informace o konkrétní funkci a nakonec je tato funkce zavolána. Tento postup má výhodu v tom, že pokud jsou třídy v samostatné knihovně, mohou se tyto knihovny za běhu programu měnit. Ovšem nevýhodou je přílišné zpomalení tímto postupem.

```
//First method
Type type = Type.GetType(...);
MethodInfo method = type.GetMethod(...);
method.Invoke(new object(), newArgs);
```

Výpis 1: První postup dynamického volání

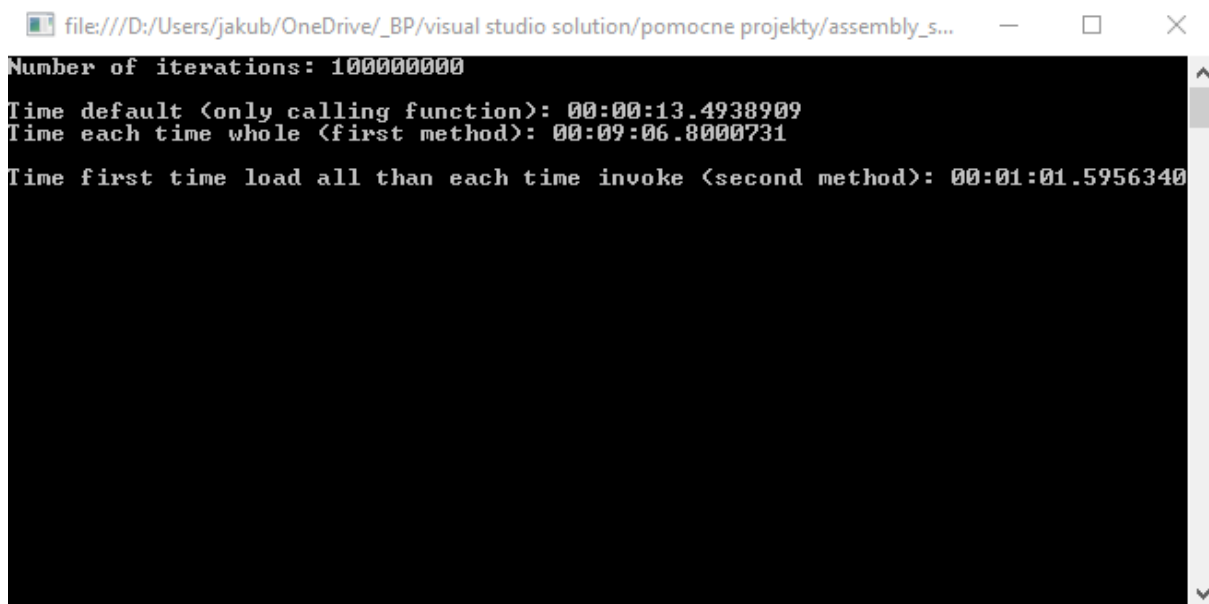
Druhý postup je takový, který při zapnutí programu načte danou *Assembly*, poté se zjistí informace o všech *public static* funkcích a vytvoří se jejich seznam. Následně pokud uživatel volá danou funkci, tak se funkce pouze hledá v seznamu a volá, nemusí se tedy pokaždé znova provádět postup pro načtení jako v případě prvního postupu. Nevýhodou tohoto postupu je ovšem to, že případné knihovny s funkcemi nelze za běhu programu měnit. Jeho výhodou je ale rychlost.

```
//Second method
Dictionary<string, MethodInfo> funcs = new Dictionary<string, MethodInfo>();
.
.
.
funcs[functionName].Invoke(new object(), newArgs);
```

Výpis 2: Druhý postup dynamického volání

Pro srovnání rychlostí obou postupů byla vytvořena konzolová aplikace 3. Na následujícím obrázku lze vidět výsledek tohoto testu při 100 milionech iterací na těchto metodách:

- Základní metoda – pouze volání metody, bez dynamického volání
- První metoda (first method) – První použitý postup
- Druhá metoda (second method) – Druhý použitý postup



```
file:///D:/Users/jakub/OneDrive/_BP/visual studio solution/pomocne projekty/assembly_s...
Number of iterations: 100000000
Time default <only calling function>: 00:00:13.4938909
Time each time whole <first method>: 00:09:06.8000731
Time first time load all than each time invoke <second method>: 00:01:01.5956340
```

Obrázek 18: Výsledek testu rychlosti dynamického volání funkcí

Z tohoto výsledku je patrné, že rychlost volání funkcí se po změně z prvního na druhý postup zvýšila přibližně 9-krát, což se dá považovat za velký výkonnostní rozdíl.

6 Rozdíly počítačové a lidské simulace vstupů

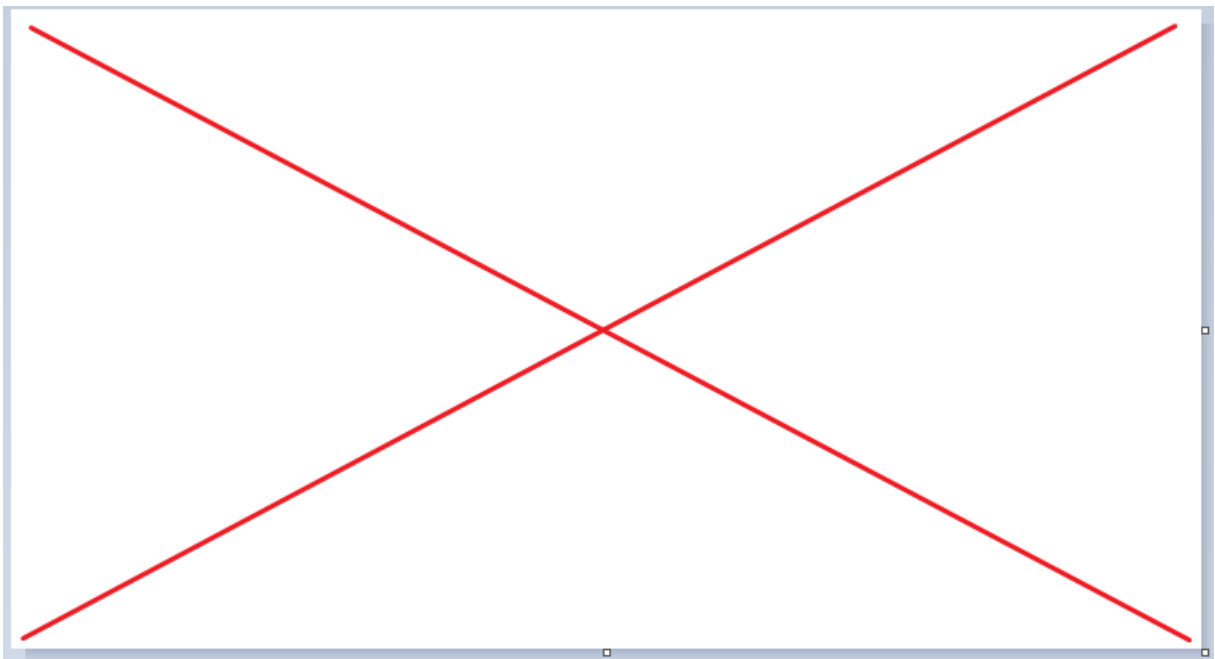
Simulace uživatelských vstupů můžeme rozdělit na 2 kategorie a to:

- Myš
- Klávesnice

Následně bude vysvětleno, jak simulace probíhají v jednotlivých kategoriích.

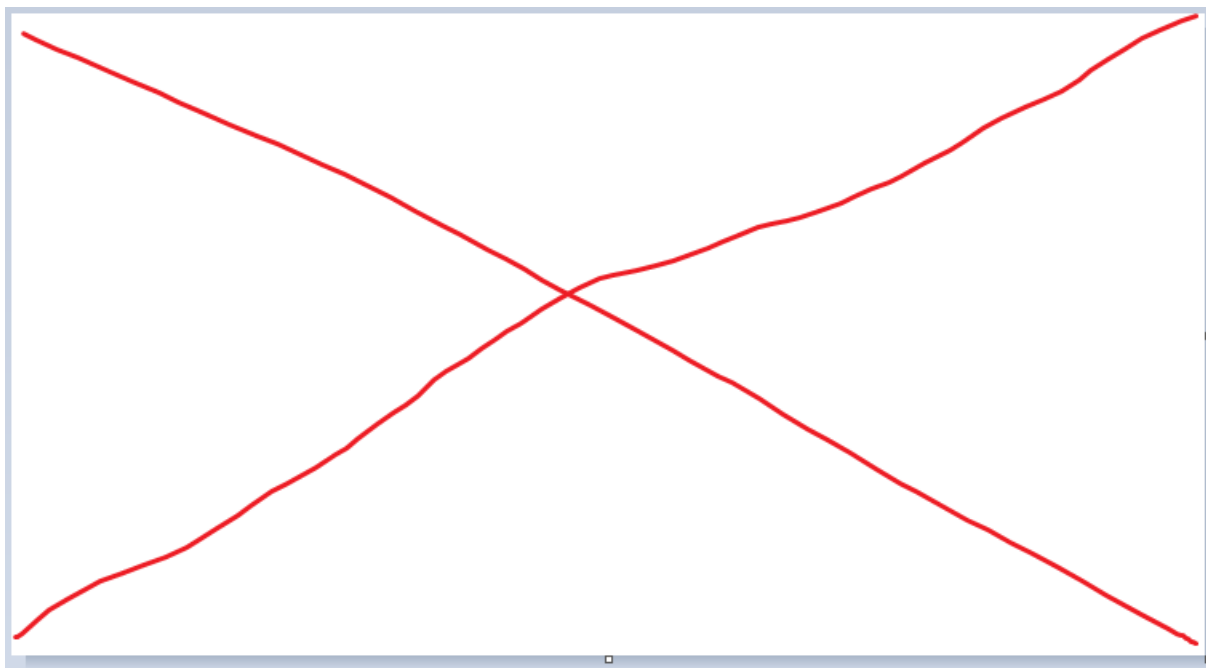
6.1 Myš

První typ simulace je počítačová simulace, která probíhá tak, že po zavolání příslušné funkce je tato funkce prováděna s počítačovou přesností, to znamená, že kurzor se pohybuje přesně na zadaná místa a mezi stiskem a uvolnění tlačítka není téměř žádná prodleva, dále se také kurzor postupně nepřesouvá z bodu A do bodu B, ale rovnou se na cílovém bodě objeví.



Obrázek 19: Ukázka počítačové simulace nakreslení kříže

Druhý typ simulace je simulace lidská. V tomto režimu se simulace vstupu myši provádí s časovými prodlevami mezi stiskem tlačítka a jeho uvolněním. Dále pohyb kurzoru je postupný a nepřesný. Pokud se využívají funkce pro přesun kurzoru na element nebo na obrázek, výsledná pozice kurzoru není vždy uprostřed, ale je náhodně umístěná v ohraničujícím obdélníku. Pohyb myši také není rovný, ale je do něj uměle "foukáno" tak, aby pohyb myši vypadal co nejrealističtější.



Obrázek 20: Ukázka lidské simulace nakreslení kříže

6.2 Klávesnice

Pro simulaci vstupu klávesnice nejsou rozdíly tak zásadní, jako u simulace myši. Jediný zásadní rozdíl mezi počítačovou a lidskou simulací vstupu z klávesnice je ten, že při lidské simulaci se vkládají prodlevy s náhodnou časovou délkou z daného rozmezí.

7 Test programu Automatizer

Pro testovací účely jsou vybrány tyto 2 aplikace, které budou automatizovány:

- Poznámkový blok – jedná se o jednoduchý textový editor s jednoduchým GUI pro psaní rychlých poznámek
- Microsoft Word 2016 – jedná se o pokročilý textový editor s pokročilým GUI pro psaní delších dokumentů

Testování je zaměřeno na časovou stránku, tedy za jak dlouho dokáže program Automatizer vykonat stejnou činnost v obou výše uvedených aplikacích. Jako zaměřovací systém je vybrán systém lokace prvků. Veškeré testy jsou prováděny několikrát a jako výsledek je uváděn jejich aritmetický průměr.

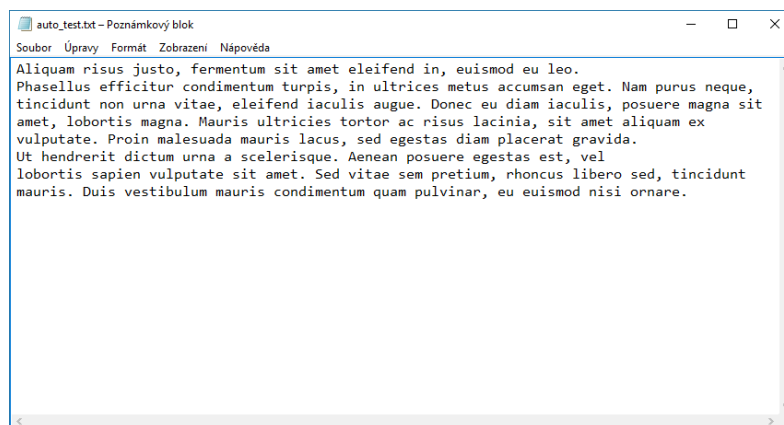
Dále se používají funkce simulující počítačový vstup, jelikož funkce simulující lidský vstup by zkreslovaly výsledky, protože při simulaci vkládají mezi jednotlivé kroky a také v jejich průběhu časové prodlevy pro dosažení co nejrealističtější lidské simulace.

Scénářem testu je provést nahrazení určitého textu textem jiným a následné uložení souboru. Dialogy jsou potvrzovány kliknutím myši a nikoli přes klávesové zkratky, program tedy tyto prvky vyhledává a přesouvá na ně kurzor.

Očekávání výsledku testu je takové, že automatizace programu poznámkový blok proběhne rychleji, než automatizace programu Microsoft Word 2016 a to díky menšímu počtu prvků nutných k prohledání.

7.1 Test automatizace aplikace Poznámkový blok

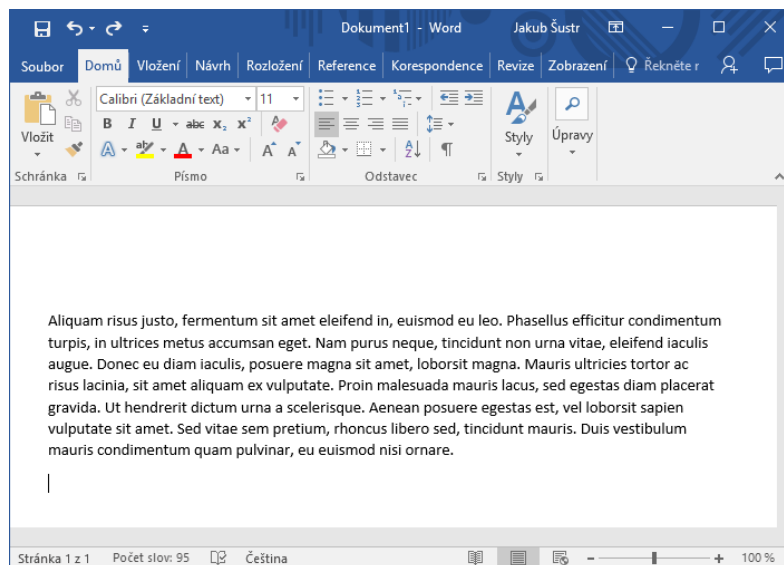
Scénář testu byl proveden úspěšně. Veškeré texty, které byly nahrazovány, byly nahrazeny ve všech případech úspěšně. Průměrná rychlost vykonání automatizace aplikace Poznámkový blok s tímto scénářem je **99 ms**.



Obrázek 21: Poznámkový blok s textem pro automatizační test

7.2 Test automatizace aplikace Microsoft Word 2016

Scénář testu byl proveden úspěšně. Veškeré texty, které byly nahrazovány, byly nahrazeny ve všech případech úspěšně. Průměrná rychlost vykonání automatizace aplikace Microsoft Word 2016 s tímto scénářem je **972ms**.



Obrázek 22: Microsoft Word 2016 s textem pro automatizační test

7.3 Výsledek testu

Jak bylo očekáváno, automatizační test pro aplikaci Poznámkový blok proběhl rychleji než test pro aplikaci Microsoft Word 2016. Hlavní důvod tohoto časového rozdílu je v počtu prvků, které se musí prohledávat pro nalezení určitého elementu. V případě aplikace Microsoft Word 2016 je těchto prvků mnohem více.

Ovšem i přesto, že čím více má aplikace prvků, tím pomalejší je její automatizace v tomto zaměřovacím režimu, si lze všimnout, že těchto výsledků lze jen stěží dosáhnout lidským výkonem.

Závěr

Cílem této bakalářské práce bylo vyvinout software pro automatizaci uživatelského rozhraní. Nejprve byl proveden průzkum konkrétních produktů, které jsou již vyvinuty, a slouží k automatizaci a testování GUI. Následně byl navrhnout software, který slouží k automatizaci GUI a umožní uživateli automatizaci cílového programu. Tato automatizace uživateli značně ušetří čas, který může využít efektivněji jak v osobním životě, tak v pracovní sféře.

Program navržený k této bakalářské práci byl pojmenován Automatizer. Program slouží k automatizaci cílového programu, kdy si uživatel nejprve vybere cílový program, a následně podle zadaného skriptu, nahraného či ručně napsaného, dokáže automatizovat cílový program. V programu je vícero druhu zaměřovacích systémů. Mezi základní patří souřadnicový zaměřovací systém. Dalším systémem určeným k zaměřování je systém zaměřování prvků, kdy uživatel vybírá cílové UI prvky a posledním zaměřovacím systémem je systém zaměřování v obraze, kdy uživatel vybírá obrázek, který se následně sám vyhledá v cílovém programu.

Jak již bylo zmíněno, tento software je navržen primárně pro účely automatizace či testování a tím uspoření času. Uživatelé mohou Automatizer využít například k automatickému otevření emailové schránky či k nastavení pracovní aplikace. Pomocí tohoto programu lze automatizovat v podstatě veškeré činnosti, které nevyžadují uživatelské vyhodnocení situace.

Při psaní této bakalářské práce se bylo možno seznámit s technologiemi zabývajícími se možnostmi automatizace, dále se bylo možno seznámit s tím, jak fungují GUI určitých aplikací na operačním systému Windows, a také byla možnost seznámit se s fungováním frameworků a knihoven pro zpracování obrazu. Také zde byl proveden automatizovaný test aplikací. Tato bakalářská práce byla přínosem a obohacením znalostí.

Literatura

- [1] AscentialTest. Zeenyx [online]. [cit. 2017-04-06]. Dostupné z: <http://www.zeenyx.com>
- [2] AutoIt [online]. [cit. 2017-04-06]. Dostupné z: <https://www.autoitscript.com>
- [3] Wikipedia [online]. [cit. 2017-04-06]. Dostupné z: <https://en.wikipedia.org/wiki/BASIC>
- [4] LDTP [online]. [cit. 2017-04-06]. Dostupné z: <https://ldtp.freedesktop.org/wiki/>
- [5] Maveryx [online]. [cit. 2017-04-06]. Dostupné z: <http://www.maveryx.com/>
- [6] Selenium [online]. [cit. 2017-04-06]. Dostupné z: <http://www.seleniumhq.org/>
- [7] IMacros [online]. [cit. 2017-04-06]. Dostupné z: <http://imacros.net/>
- [8] Sahi [online]. [cit. 2017-04-06]. Dostupné z: <http://sahipro.com/>
- [9] C# .NET [online]. [cit. 2017-04-08]. Dostupné z: <https://msdn.microsoft.com/en-us/library/z1zx9t92.aspx>
- [10] Microsoft UI Automation [online]. [cit. 2017-04-08]. Dostupné z: [https://msdn.microsoft.com/en-us/library/ms747327\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms747327(v=vs.110).aspx)
- [11] Windows Input Simulator [online]. [cit. 2017-04-08]. Dostupné z: <http://inputsimulator.codeplex.com/>
- [12] Global MouseKeyHook [online]. [cit. 2017-04-08]. Dostupné z: <https://github.com/gmamaladze/globalmousekeyhook>
- [13] OpenCV [online]. [cit. 2017-04-08]. Dostupné z: <http://opencv.org>
- [14] EmguCV [online]. [cit. 2017-04-08]. Dostupné z: <http://www.emgu.com>

A Obsah přiloženého DVD

1. Text bakalářské práce
2. Microsoft Visual Studio projekt bakalářské práce
3. Microsoft Visual Studio projekt aplikace pro srovnání rychlosti dynamického volání funkcí