

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky

Tvorba interaktivního symbolického jazyka pro uplatnění systémových  
proměnných v AutoCADu

Martin Kopecký

Bakalářská práce  
2023

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2022/2023

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Martin Kopecký**  
Osobní číslo: **I19105**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Téma práce: **Tvorba interaktivního symbolického jazyka pro uplatnění systémových proměnných v AutoCADu**  
Zadávající katedra: **Katedra informačních technologií**

## Zásady pro vypracování

Bakalářská práce bude zpracována s cílem vytvořit interaktivní symbolický jazyk srozumitelný pro celý svět, který najde uplatnění v hlavních oblastech nápovědy a popisu, zejména v systémových proměnných AutoCADu. Používáním tohoto symbolického jazyka nebude tak velká potřeba vytvářet podrobné textové nápovědy do všech národních lokalizací tj. i do české verze.

V teoretické části práce se seznámíme s možnostmi uplatnění interaktivního symbolického jazyka (piktogramy, animace, obrázky, ...) pro AutoCAD.

V implementační části se práce zaměří na vytvoření nové nápovědy u vybraných částí stávající nápovědy a popisu vybraných systémových proměnných pomocí interaktivního symbolického jazyka.

Rozsah pracovní zprávy: **min. 30 stran**  
Rozsah grafických prací:  
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

ABDULLAH, Rayan a Roger HÜBNER. Pictograms, icons & signs: a guide to information graphics. New York: Thames & Hudson, 2006. ISBN 978-0500286357.

HLAVENKA, Jiří a David ŘEHÁČEK. AutoCAD Release 14: průvodce příkazy a funkcemi. Praha: Computer Press, 1997. ISBN 80-7226-014-6.

Autodesk AutoCAD 2022: Nápověda [online]. Autodesk Inc., 2022 [cit. 2022-10-20]. Dostupné z: <https://help.autodesk.com/view/ACD/2022/CSY/>.

Vedoucí bakalářské práce: **Ing. Zbyněk Kopecký**  
Katedra informačních technologií

Datum zadání bakalářské práce: **16. prosince 2022**  
Termín odevzdání bakalářské práce: **12. května 2023**

**Ing. Zdeněk Němec, Ph.D. v.r.**  
děkan

L.S.

**Ing. Jan Panuš, Ph.D. v.r.**  
vedoucí katedry

V Pardubicích dne 28. února 2023

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 3. 5. 2023

Martin Kopecký

## **PODĚKOVÁNÍ**

Tímto bych rád poděkoval Ing. Zbyňku Kopeckému za cenné rady a pomoc při zpracování této práce a za jeho trpělivost během konzultací.

## **ANOTACE**

Tato práce se zabývá symbolickými jazyky a jejich uplatněním pro vytvoření nové nápovědy a popisu u vybraných systémových proměnných v AutoCADu. Práce je rozdělena na dvě části. První část je zaměřena na symbolické jazyky, jako jsou obrázky, symboly, piktogramy, animace, a jejich definici, vlastnosti, historii, využití a také se zabývá tím, co je to AutoCAD a jak s ním lze pracovat. Druhá část je zaměřena na praktickou ukázkou, kde je popsána samotná implementace a je ukázána nově vytvořená nápověda. Součástí jsou také uživatelské ohlasy ohledně této nové nápovědy. Získané ohlasy na jednotlivé formy provedení nápovědy jsou poté mezi sebou porovnány.

## **KLÍČOVÁ SLOVA**

Symbolický jazyk, obrázek, symbol, piktogram, animace, AutoCAD, interaktivní, systémové proměnné, nápověda

## **TITLE**

Creating an interactive symbolic language for applying system variables in AutoCAD

## **ANNOTATION**

This bachelor thesis deals with symbolic languages and their application to create a new help and description for selected system variables in AutoCAD. The work is divided into two parts. The first part focuses on symbolic languages such as images, symbols, pictographs, animations, and their definition, characteristics, history, usage, and also deals with what AutoCAD is and how it can be worked with. The second part is focused on a practical demonstration where the actual implementation is described, and the newly created help is shown. It also includes user feedback about this new help. The received responses to each form of help are then compared with each other.

## **KEYWORDS**

Symbolic language, image, symbol, pictogram, animation, AutoCAD, interactive, system variables, help

# OBSAH

<b>Seznam obrázků</b> .....	<b>9</b>
<b>Seznam zdrojových kódů</b> .....	<b>11</b>
<b>Seznam zkratek</b> .....	<b>12</b>
<b>Úvod</b> .....	<b>13</b>
<b>1 Obrázky</b> .....	<b>14</b>
1.1 Definice obrázku .....	14
1.2 Formáty digitálních obrazových souborů .....	14
1.2.1 Rastrový soubor .....	14
1.2.2 Vektorový soubor .....	15
1.2.3 RAW soubor .....	16
<b>2 Symboly</b> .....	<b>18</b>
2.1 Definice.....	18
2.2 Typy symbolů .....	18
2.2.1 Typogram.....	18
2.2.2 Fonogram .....	19
2.2.3 Rébus .....	19
2.2.4 Ideogram .....	20
2.2.5 Ikona .....	20
2.2.6 Logo .....	21
2.2.7 Piktogramy.....	23
2.3 Využití symbolů.....	24
2.4 Rozdíl mezi symbolem a značkou .....	24
2.5 Studium symbolů .....	25
<b>3 Animace</b> .....	<b>26</b>
3.1 Definice.....	26
3.2 Typy animací .....	26
3.2.1 Tradiční animace.....	26
3.2.2 Počítačová animace.....	27
3.2.3 Stop Motion .....	31
3.2.4 Pohyblivá grafika.....	32
3.3 Historie animace .....	33
3.4 Využití animací.....	41
<b>4 Piktogramy</b> .....	<b>42</b>
4.1 Definice.....	42
4.2 Historie.....	42
4.3 Moderní užití.....	43

4.4	Standardizace .....	43
<b>5</b>	<b>AutoCad .....</b>	<b>45</b>
<b>6</b>	<b>Možnosti implementace .....</b>	<b>46</b>
6.1	AutoLISP/VisualLISP .....	46
6.2	Visual Basic for Applications (VBA).....	46
6.3	ActiveX.....	47
6.4	ObjectARX® .....	47
6.5	C# .NET .....	47
<b>7</b>	<b>Praktická část.....</b>	<b>49</b>
7.1	Implementace.....	49
7.2	Struktura formuláře s nápovědou.....	54
7.3	Formy nápovědy a jejich uživatelské preference.....	56
7.3.1	Systémové proměnné obsahující číselnou hodnotu .....	56
7.3.2	Systémová proměnná pro nastavení pozice kótovacího textu .....	58
7.3.3	Systémová proměnná pro nastavení počtu desetinných míst rozměrů .....	59
7.3.4	Systémové proměnné nabývají stavu Zapnuto/Vypnuto .....	61
7.4	Shrnutí uživatelského průzkumu .....	65
	<b>Závěr .....</b>	<b>67</b>
	<b>Použitá literatura .....</b>	<b>68</b>
	<b>Přílohy.....</b>	<b>77</b>



## SEZNAM OBRÁZKŮ

Obrázek 1: Ukázka rastrového obrázku [7] .....	15
Obrázek 2: Ukázka vektorového obrázku [7] .....	16
Obrázek 3: Ukázka typogramu [16].....	18
Obrázek 4: Ukázka fonogramu [18] .....	19
Obrázek 5: Ukázka rébusu IBM [19].....	19
Obrázek 6: Ukázka rébusu I Love New York [20] .....	20
Obrázek 7: Ukázka ideogramů [23].....	20
Obrázek 8: Ukázka ikony pro tiskárnu [26].....	21
Obrázek 9: Ukázka ikony pro vyhledávání [25] .....	21
Obrázek 10: Wordmark loga [28] .....	21
Obrázek 11: Lettermark loga [30] .....	22
Obrázek 12: Brandmark loga [32] .....	22
Obrázek 13: Kombinovaná loga [33].....	23
Obrázek 14: Emblém loga [34].....	23
Obrázek 15: Ukázka maskotů [35] .....	23
Obrázek 16: Tvoření snímků pro tradiční animaci [55] .....	27
Obrázek 17: Získání pohybu přes motion capture [59] .....	28
Obrázek 18: Magická lucerna [75] .....	33
Obrázek 19: Thaumatrope [76].....	34
Obrázek 20: Phenakistoscope [77] .....	34
Obrázek 21: Zoetrope [79].....	35
Obrázek 22: Flipbbok [80].....	35
Obrázek 23: Praxinoskop [81] .....	36
Obrázek 24: Gertie the Dinosaur [83].....	37
Obrázek 25: Felix the Cat [84] .....	37
Obrázek 26: Betty Boop [85].....	39
Obrázek 27: Bugs Bunny [87] .....	39
Obrázek 28: Piktogramy domorodých Američanů v Great Gallery, Horseshoe Canyon, Canyonlands National Park [95].....	42
Obrázek 29: Mezinárodní piktogram pro varování před ionizujícím zářením [96].....	43
Obrázek 30: ISO 7001 piktogram pro odlety letů [98].....	44
Obrázek 31: DOT piktogram pro odlety letů [99] .....	44

Obrázek 32: Ukázka nápovědy pro systémovou proměnnou DIMTOH (Zdroj vlastní) .....	54
Obrázek 33: Ukázka nápovědy pro systémovou proměnnou DIMEXE (Zdroj vlastní).....	56
Obrázek 34: Ukázka nápovědy pro systémovou proměnnou DIMTP (Zdroj vlastní).....	57
Obrázek 35: Ukázka nápovědy pro systémovou proměnnou DIMRND (Zdroj vlastní) .....	58
Obrázek 36: Ukázka nápovědy pro systémovou proměnnou DIMITAD (Zdroj vlastní) .....	58
Obrázek 37: Ukázka nápovědy pro systémovou proměnnou DIMITAD (Zdroj vlastní) .....	59
Obrázek 38: Ukázka nápovědy pro systémovou proměnnou DIMDEC (Zdroj vlastní) .....	60
Obrázek 39: Ukázka nápovědy pro systémovou proměnnou DIMDEC (Zdroj vlastní) .....	60
Obrázek 40: Ukázka nápovědy pro systémovou proměnnou DIMLIM (Zdroj vlastní) .....	61
Obrázek 41: Ukázka nápovědy pro systémovou proměnnou DIMITIH (Zdroj vlastní).....	62
Obrázek 42: Ukázka nápovědy pro systémovou proměnnou DIMITIX (Zdroj vlastní).....	62
Obrázek 43: Ukázka nápovědy pro systémovou proměnnou DIMTOFL (Zdroj vlastní) .....	63
Obrázek 44: Ukázka nápovědy pro systémovou proměnnou DIMTOH (Zdroj vlastní) .....	64
Obrázek 45: Ukázka nápovědy pro systémovou proměnnou DIMITOL (Zdroj vlastní).....	65

## **SEZNAM ZDROJOVÝCH KÓDŮ**

Zdrojový kód 1: Ukázka atributu CommandMethod a s ním spojené funkce (Zdroj vlastní)..51	
Zdrojový kód 2: Ukázka části kódu nápovědy pro systémovou proměnnou DIMTOH (Zdroj vlastní) .....	55

## SEZNAM ZKRATEK

DPI	Dots Per Inch
PPI	Pixels Per Inch
JPEG	Joint Photographic Experts Group
PNG	Portable Network Graphics
GIF	Graphical Interchange Format
BMP	Bitmap
TIFF	Tag Image File Format
PSD	Photoshop Document
MB	Megabyte
SVG	Scalable Vector Graphics
EPS	Encapsulated PostScript
AI	Adobe Illustrator Artwork
DAE	Digital Asset Exchange
PS	PostScript
EMF	Enhanced Windows Metafile
IBM	International Business Machines Corporation
CAD	Computer-Aided Design
FPS	Frames Per Second
Hz	Hertz
CGI	Computer-Generated Imagery
AR	Augmented Reality
VR	Virtual reality
ISO	International Organization for Standardization
GHS	Globally Harmonized System of Classification and Labelling of Chemicals
DOT	United States Department of Transportation
DWG	Drawing
DXF	Drawing Exchange Format
ASCII	American Standard Code for Information Interchange
DLL	Dynamic-Link Library
ADS	AutoCAD Development System
CLI	Common Language Infrastructure
IDE	Integrated Development Environment
API	Application Programming Interface
SDK	Software Development Kit
LSP	Lisp Program Source Code

# ÚVOD

Žijeme v době, kdy téměř každý vlastní nebo má aspoň přístup, ať už k počítačům, telefonům či jiným podobným zařízením. Pro práci s nimi existují tisíce programů a aplikací nejrůznějších typů, funkcí, použití a využíváme je pro všemožné účely a potřeby. Jelikož je mohou používat lidé z celého světa, tak je již většina programů dostupná v široké řadě národních lokalizací. Z toho ovšem vyplývá, že nestačí vytvořit manuál, nápovědu a další pomocné materiály pouze v jedné lokalizaci, ale je nutné je udělat v každé lokalizaci, kterou daný program nabízí. To ale znamená, že při velkém počtu národních lokalizací může být vytváření manuálů a nápověd poměrně časově i finančně náročné, a to může mít za následek, že se počet dostupných národních lokalizací omezí a lidé z nezahrnutých lokalizací mohou mít při používání programu potíže.

Řešení tohoto problému je cílem této práce, a to ukázat využití symbolických jazyků a jiných grafických prvků při tvorbě manuálů a nápověd. To může jejich tvorbu značně usnadnit a udělat více univerzální, jelikož když bude nápověda tvořena symbolickými jazyky, a to buď kompletně, nebo aspoň z větší části, tak její převedení do jiného jazyka bude daleko jednodušší. Použití symbolických jazyků umožňuje i kolikrát snazší pochopení funkce nebo vlastnosti, než kdyby byla popsána pouze psaným textem. Výhodou symbolických jazyků je také to, že s nimi každodenně přicházíme do styku a jejich použití je velmi rozšířené napříč všemi formami komunikace, takže se nejedná o nic, s čím by se lidé již dříve nesetkali. Doufám, že se mi touto prací podaří předvést, jaký přínos by takové manuály a nápovědy měly.

Práce je rozdělena na teoretickou a praktickou část. Teoretická část se zabývá jednotlivými symbolickými jazyky, jako jsou obrázky, symboly, piktogramy, animace, a jejich charakteristickými znaky, využitím, historií a další problematikou s nimi spojenou. Součástí této části je i obeznámení s tím, co je AutoCAD a jaké existují možnosti pro práci a vývoj AutoCADu.

Na to navazuje praktická část, která je zaměřena na použití symbolických jazyků pro vytvoření nápovědy v softwaru AutoCAD, a to konkrétně pro popsání vybraných systémových proměnných. Tato část zahrnuje popsání způsobu implementace, popis vytvoření nápovědy pro určitou systémovou proměnnou a také průzkum a ohlasy uživatelů na jednotlivá provedení nápovědy společně i s její ukázkou.

Celá práce je zakončena shrnutím a porovnáním získaných ohlasů od uživatelů, které z prezentovaných provedení nápovědy je v podle nich v každé kategorii nejlepší.

# 1 OBRÁZKY

## 1.1 Definice obrázku

Obrázek lze definovat jako vizuální zachycení nebo reprezentaci něčeho, ať už věci, osoby, přírody a mnoho dalšího. Dnes je ale potřeba rozlišovat, zda se jedná o klasický, fyzický obrázek nebo o digitální obrázek, což je binární reprezentace obrazových dat. [1]

Pro účely této práce se budu dále zabývat pouze digitálními obrázky a problematikou s nimi spojenou. Digitální obrázek lze získat různými způsoby. Nejčastější způsob získání je za použití digitálního fotoaparátu nebo pomocí mobilního telefonu. Dále můžeme digitální obrázek získat tak, že ho vytvoříme na počítači pomocí grafických editorů. Mezi ty nejznámější grafické editory patří například Adobe Photoshop, Adobe Illustrator, Adobe InDesign, Mega Creator nebo také jednoduchý program Malování. Další možností získání digitálního obrázku je použití scanneru nebo získání jednotlivých snímků z video záznamu. [2]

Všechny takto získané digitální obrázky jsou uloženy na paměťovém médiu v elektronické formě. [1]

## 1.2 Formáty digitálních obrazových souborů

Mezi dva nejoblíbenější formáty pro práci s vizuálními daty patří rastrové a vektorové soubory. Často používaný je také specifický RAW soubor. Každý tento formát disponuje různými vlastnostmi, obrázek zobrazují a vytváří zcela odlišnými způsoby a nabízí rozdílné typy souborů, z nichž každý má dále své určité vlastnosti. Proto u výběru vhodného formátu velmi záleží na tom, na co je obrázek určený, jakým způsobem byl vytvořen, popřípadě jak často se bude měnit jeho velikost. [3] [4]

### 1.2.1 Rastrový soubor

Rastrový soubor, nebo také rastrová grafika, je obrázek, jenž je tvořen dvourozměrnou soustavou, která obsahuje pevný počet obrazových bodů, z nichž každý má přiřazenou určitou barvu a souřadnici x a y. [5]

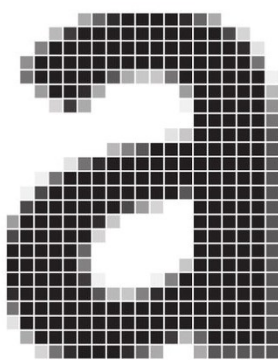
Celý rastrový soubor je poté popsán rozlišením, které se v rastrové grafice udává v DPI nebo PPI, rozměry a barevnou hloubkou. Pro rastrový obrázek platí, že čím více obrazových bodů obsahuje a čím větší má barevnou hloubku, tím roste jeho kvalita a detaily, přičemž počet obrazových bodů se odvíjí od použitého typu souboru. Mezi typy rastrových souborů spadají soubory JPEG, PNG, GIF, BMP, TIFF, PSD. [4] [6]

Asi největší výhodou rastrové grafiky je její schopnost zobrazovat realitu ve velmi vysoké kvalitě. Při použití správných rozměrů umožňuje zobrazení fotografií nebo obrázků ve vysokém

rozlišení se složitými barvami a detaily. Další výhodou je vysoká kompatibilita, která umožňuje soubory editovat a otvírat napříč velkým množstvím aplikací, programů a webových prohlížečů. Nabízí také větší možnosti při editaci oproti vektorovému souboru, jelikož se dají měnit jednotlivé obrazové body, čímž lze dosáhnout přesnějších úprav.[5]

Kromě výhod ale rastrová grafika disponuje i řadou nevýhod. Tou největší je zhoršení kvality při změně rozměrů obrázku. Lze provést malé změny bez zásadního dopadu na kvalitu, nicméně při dalším zvětšování dochází k viditelnosti jednotlivých obrazových bodů a ke zhoršení detailů, čitelnosti a barev. Nevýhodou může být i velikost souboru, která se pohybuje v jednotkách nebo i desítkách MB v případě, že obrázek má vysoké rozlišení a velkou barevnou hloubku. [5]

Rastrová grafika se nejčastěji využívá při práci a editaci digitálních fotografií, tvorbě nových rastrových obrázků nebo při digitalizaci fyzických materiálů, jako jsou knihy, spisy a další. Veliké využití je také na internetu, jako například na webových stránkách či sociálních sítích, v herním průmyslu nebo i v lékařství. [5]



Obrázek 1: Ukázka rastrového obrázku [7]

### 1.2.2 Vektorový soubor

Vektorový soubor, nebo také vektorová grafika, neobsahuje žádné obrazové body, ale obrázek je složen z přesně definovaných tvarů, jako jsou body, mnohoúhelníky, přímky, ale zejména křivky. [8]

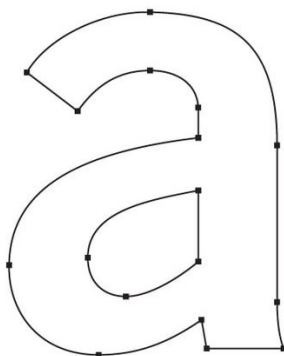
Takovýto objekt je definován ve formě matematického zápisu, který spolu s tím obsahuje i jeho grafické informace, mezi které patří barva, výplň, tloušťka hrany, zakřivení spolu s kotevními body a další. V případě křivky bude matematický záznam ještě obsahovat počáteční a koncový bod a k tomu ještě její vektor. Díky tomu, že je objekt definován pomocí matematického zápisu a není nutné mít definovány jednotlivé obrazové body, jak tomu je u rastrového souboru, tak se změna rozměrů obrázku provede pomocí matematického výpočtu, což znamená, že nedojde ke zhoršení jeho kvality. [9]

Do skupiny typů vektorových souborů spadají soubory SVG, EPS, AI, DAE, PS, EMF. [4]

Výhody a nevýhody vektorové grafiky jsou často opakem těch u rastrové grafiky. Hlavní výhodou, jak již bylo zmíněno, je možnost v podstatě neomezeně měnit rozměry obrázku bez jakékoliv ztráty jeho kvality. Dalšími výhodami jsou menší velikost souboru a možnost pracovat s každým objektem samostatně, nezávisle na ostatních a dávat jim různé vlastnosti. [9]

Mezi nevýhody oproti rastrové grafice spadá to, že není jednoduché vytvořit složitý obrázek. Takový obrázek bude složen z velkého množství objektů, čímž vznikají vyšší požadavky na výpočetní výkon. Další nevýhoda je menší kompatibilita, jelikož k otevření a editaci jsou zapotřebí speciální programy. [9]

Vektorová grafika má hojné využití v marketingu pro vytváření log či propagačních materiálů, v reklamních agenturách i v průmyslu, kde na této grafice jsou založeny grafické, konstrukční, návrhářské a další modelové programy. Dobrý příklad těchto programů jsou programy CAD. Používá se také ve webových prohlížečích a stránkách, kde mají tu výhodu, že stačí jediný soubor a není nutné mít více souborů s různými rozměry. [9]



Obrázek 2: Ukázka vektorového obrázku [7]

### 1.2.3 RAW soubor

RAW soubor je sice typem rastrového souboru, jenže se nejedná přímo o obrázek. RAW soubor slouží k uchování nekomprimovaných dat pořízených digitálními fotoaparáty a skenery. Tyto obrazová data jsou přímo ze snímačů kamery a jsou minimálně zpracována a v bezztrátové kvalitě. To při jejich editaci poskytuje mnohem větší možnosti úprav. Jelikož se u RAW souboru nejedná o obrázek, je pro jeho otevření a editaci potřeba použití vhodného grafického programu. [10] [11]

Mezi nesporné výhody patří hlavně nejvyšší detaily ze všech typů rastrových souborů, bezztrátová komprese, mnohem větší rozsah barev, který nabízí miliony nových barev, možnosti konverze, a právě ona větší svoboda a možnosti při editaci díky minimálnímu



zpracování dat. [10]

Ani tomuto formátu se ale nevýhody nevyhnuly. Tou první je velikost RAW souboru, který potřebuje mnohem víc paměti, což může zpomalit fotoaparát v případě, že se snímky v tomto formátu budou pořizovat v rychlém sledu za sebou. Další je problém s kompatibilitou, jelikož RAW soubory nejsou standardizované a je potřeba specializovaných programů, aby se soubory bylo možné pracovat. Nevýhodou je také nutnost zkonvertovat soubor na jiný formát, pokud bude dále používán například pro tisk nebo sdílení. [10] [12]

Využití tohoto formátu vyplývá už z jeho výhod a to, že RAW formát se používá, pokud je zapotřebí nekomprimovaných snímků s vysokými detaily, které se budou dále editovat.

## 2 SYMBOLY

### 2.1 Definice

Symboly byly a jsou velice důležitou součástí veškeré komunikace, kterou v mnoha případech značně usnadňují, zrychlují a setkáváme se s nimi každý den. Symbolem může být v podstatě cokoli, co označuje nebo je chápáno jako něco, co je přijato určitou skupinou lidí nebo většinou populací. Symboly mohou mít podobu gesta, slov, znaků, barev, zvuků, věcí či obrazce, které se používají k vyjádření nějaké akce, emoce, myšlenky, vzpomínky nebo k předání informace, bez nutnosti použití mluveného nebo psaného textu. Nicméně symboly jsou využívány i v kombinaci s textem, aby tak zpráva byla smysluplnější, byla lépe zapamatovatelná, popřípadě tak získala hlubší význam. Tím, že značné množství symbolů může mít pro různé skupiny lidí odlišné významy, dochází k rozličným interpretacím oněch symbolů. Jako příklad takových symbolů lze zmínit symbol růže, který může mít několik významů. Symbol růže může představovat soucit, náklonost, vůni anebo také jen květinu či trny. Existuje ale i spousta symbolů, u kterých k těmto rozdílným významům docházet buď nebude, nebo jen minimálně. Mezi ty patří například symbol WiFi připojení, symbol vypínacích/zapínacích tlačítek nebo písmeno, které představuje chemický prvek. [13] [14]

### 2.2 Typy symbolů

Vzhledem k tomu, že symbolů je ohromné množství a používají se ve všemožných odvětvích, jsou rozděleny do několika různých typů.

#### 2.2.1 Typogram

Typogram je slovo, které je napsané takovým způsobem, aby vizuálně zobrazovalo to, co dané slovo znamená. [15]



m i s s i n g

Obrázek 3: Ukázka typogramu [16]

### 2.2.2 Fonogram

Fonogram je písmenný symbol, který představuje zvuk tak, jak zní při mluveném projevu. To znamená, že celá latinská abeceda je tvořena fonogramy. [15]

Fonogram může být složen z jednoho, nebo i více písmenných symbolů. Takže například u českého slova „dveře“ je slyšet 5 samostatných zvuků, a proto je tvořeno 5 fonogramy (d-v-e-ř-e). Naopak u anglického slova „right“ jsou na základě své výslovnosti slyšet 3 samostatné zvuky, a je tedy složeno ze 3 fonogramů (r-igh-t). [17]

ti      ci  
si      ough

Obrázek 4: Ukázka fonogramu [18]

### 2.2.3 Rébus

Rébus je složený z obrázků, písmen, čísel nebo dalších symbolů, které představují slovo, část slova nebo zvuk, který při vyslovení zní stejně, nebo podobně jako nahrazované slovo. Příklad takového českého rébusu může být obrázek kola, doplněný o slovo „toč“, což jako výsledné slovo vyjde „kolotoč“. Je nutno ovšem podotknout, že většina rébusů bude fungovat pouze v jednom jazyce a v jiných jazycích nebudou dávat smysl. Existují ale i výjimky, jako jsou mezinárodně užívané symboly, které lze v rébusu použít a jeho význam zůstane stejný napříč více jazyky. [15]



Obrázek 5: Ukázka rébusu IBM [19]

Rébus na Obrázku 5 představuje zkratku firmy IBM. Pokud by ovšem tento rébus byl použit v jiném jazyce, než je angličtina, ztratil by svůj smysl.



Obrázek 6: Ukázka rébusu I Love New York [20]

Naopak rébus na Obrázku 6 si svůj význam ponechá i v jiných jazycích, než je ten původní, a to především díky tomu, že symbol srdce se běžně používá jako vyjádření lásky napříč světovou populací.

#### 2.2.4 Ideogram

Ideogram je symbol, který ve zjednodušené formě vyjadřuje myšlenku, pojem, akci či věc, bez nutnosti použití psaného textu nebo mluveného zvuku. [15]

Vhodný příklad ideogramu je symbol v kombinaci s červeným kruhem a přeškrtnutím, což znamená zákaz, popřípadě se žlutým nebo oranžovým kruhem jako varování nebo nebezpečí. Ideogram lze použít i pro předání informace, například jakou akci je nutno provést nebo pro udání směru na cestě. [22]

Mezi ideogramy patří třeba i aritmetická znaménka jako mínus. [21]



Obrázek 7: Ukázka ideogramů [23]

#### 2.2.5 Ikona

Ikona je symbol, který zobrazuje objekt, funkci nebo objekt a jeho funkci. Nejčastější využití ikon je na digitálních zařízeních, jako jsou počítače, tablety a mobilní telefony. [15]

Tam ikony umožňují snadnější určení, k čemu daná aplikace nebo program slouží. [24]



Obrázek 9: Ukázka ikony pro vyhledávání [25]



Obrázek 8: Ukázka ikony pro tiskárnu [26]

## 2.2.6 Logo

Logo je tvořeno symbolem, kombinací symbolu a textu nebo jen čistě textem a slouží k identifikaci a reprezentaci firmy, značky, produktu nebo organizace. Cílem dobrého loga je, aby bylo snadno rozpoznatelné a zapamatovatelné, aby si ho lidé spojili s danou značkou pokaždé, když ho vidí. [27] I loga se dále dělí na různé typy:

1. Slovní logo (Wordmark) – v tomto logu je použit pouze text, což je ve většině případů název firmy, který je stylizován unikátním fontem. [15]



Obrázek 10: Wordmark loga [28]

2. Dopisní logo (Lettermark) – toto logo taktéž využívá jenom text, který ovšem neobsahuje celý název firmy, ale jen její iniciály. Těmi může být buď anagram, nebo monogram. [29]



Obrázek 11: Lettermark loga [30]

3. Značka (Brandmark) – je obrazový symbol, jenž je tvořen výraznou grafikou tak, aby byla jasně spojitelná s danou firmou. [29]

Využívá se zde buď abstraktní grafika, jako je například logo Pepsi, nebo obrázek reálné věci, který používá Apple nebo Twitter. [31]



Obrázek 12: Brandmark loga [32]

4. Kombinované logo (Combination mark) – tahle varianta loga je složena ze značky a názvu firmy. Oba prvky stojí ve výsledném logu vedle sebe a jak značka, tak název může být použit i samostatně. [15]



Obrázek 13: Kombinovaná loga [33]

5. Emblém – u emblému je text vložen do symbolu a je jeho součástí, což znamená, že prvky nelze použít odděleně. [15]



Obrázek 14: Emblém loga [34]

6. Maskot – tento typ loga zahrnuje ilustrovanou postavu, která samotnou firmu i reprezentuje. Postava bývá často barevná, kreslená, ale téměř vždy je vyobrazena tak, aby byla zábavná. [31]



Obrázek 15: Ukázka maskotů [35]

### 2.2.7 Piktogramy

Jelikož je piktogramům věnována samostatná kapitola, tak není potřeba je zde podrobně popisovat. Postačí definice, že piktogramy mají podobu objektu nebo místa a jsou podobné tomu, co reprezentují. [15]

## 2.3 Využití symbolů

Symboly se za dobu své existence rozšířili do všemožných odvětví lidské činnosti. Jelikož by bylo obtížné je všechny vyjmenovat, tak postačí uvést několik hlavních sektorů.

- Náboženství – v náboženství zaujímají symboly velkou roli. Nejvíce ikonické náboženské symboly jsou ty, které reprezentují dané náboženství a dále také symboly, které představují předměty nebo činnosti v rámci onoho náboženství. [36] [37]
- Architektura – zde jsou symboly velmi důležité a používají se v nákresech a plánech, kde zobrazují objekty, předměty, konstrukce, typ materiálu, někde i naznačení akce. Třeba u symbolu dveří či oken je zakreslen i směr otvírání. [38]
- Literatura – v rámci literatury mají symboly podobu slov nebo i obrázků a symbolizují pojem, myšlenku, událost či předmět. Spisovatelé je často využívají, aby lépe vyjádřili to, co chtějí, dodali nějaké pasáží větší váhu a aby své dílo udělali živější, dodali mu hlubší význam nebo aby si čtenář mohl v mysli vytvořit přesnější obraz. [39]

Příkladem takového symbolu je zelené světlo v díle *The Great Gatsby* (1925) od F. Scotta Fitzgeralda, kde světlo může ilustrovat bohatství a peníze, nebo také představuje vše, co je pro hlavní postavu nedosažitelné. [40]

- Umění – v umění jsou symboly reálné a snadno rozpoznatelné věci, jako jsou zvířata, rostliny, předměty nebo i postavy, a slouží k vyjádření něčeho, co by bylo jinak těžké namalovat. [41]

Symboly mohou celé dílo udělat zajímavější a umožnit mu větší dopad. Takové symboly jsou například čtyřlístek, který představuje štěstí, kniha, která zobrazuje vzdělání či vědomosti, nebo lebka, jenž představuje smrt. [42]

- Kartografie – v kartografii se používají k zobrazení geografických rysů. Vyjadřují přírodní prvky (řeky, les, hory), aktivity v určitých lokacích (rybaření, lyžování), prvky infrastruktury (dálnice, železnice), budovy (nemocnice, obecní budovy). [43]
- Doprava – symboly se v dopravě používají hlavně na dopravních značkách a značení k rychlému předání potřebné informace a usnadnění provozu. Takové značky mohou být příkazové, zákazové, varovné nebo informativní. [44]

## 2.4 Rozdíl mezi symbolem a značkou

Stejně jako symboly, jsou značky už od nepaměti používány ke komunikaci ve všemožných odvětvích. Často však dochází k chybnému názoru, že symbol a značka jsou synonyma.



Existuje ovšem několik hlavních rozdílů, které tyto dva pojmy odlišují. [45]

Jak již bylo zmíněno dříve, u symbolu může u různých skupin lidí docházet k jeho rozdílným interpretacím. Značka má ovšem všeobecně přijatý význam, který se musí povinně a bez otázek následovat a neumožňuje odlišné výklady. [45] [46]

Liší se taktéž vzhled a použití. Symboly povětšinou vypadají jinak, než jaký je jejich význam a slouží jako pomůcka, jak podat důležitou informaci snadno a srozumitelně. Značka naopak vypadá přesně jako to, co vyjadřuje. Takže pokud značka obsahuje invalidní vozík, znamená to, že se jedná o bezbariérový prostor. [45]

Další rozdíl je ten, že u symbolu může hrát velkou roli jakou má symbol barvu, jelikož při její změně může také dojít ke změně samotného významu symbolu. Naopak značka bude znamenat pořád to stejné i při použití jiné barvy. [45]

Na základě těchto rozdílů je zřejmé, že se jedná o rozdílné pojmy, a je dobré je znát.

## **2.5 Studium symbolů**

Symboly se zabývá celá studie, která se nazývá sémiotika, nebo také sémiologie, a je zaměřená na studium znaků a symbolů, jejich používání a další způsoby jazykové a nejazykové symbolické komunikace. Sémiotici se věnují tomu, jak lidé znaky a symboly vytváří, jak je používají a interpretují a jaký mají vliv na lidské životy. [47]

Zakladateli této studie jsou švýcarský lingvista Ferdinand de Saussure a americký filozof Charles Sanders Peirce. [48]

## 3 ANIMACE

### 3.1 Definice

Celá podstata animace spočívá ve vytvoření iluze pohybu tak, že se v rychlém sledu po sobě zobrazují statické obrázky, které se od sebe liší. [49]

Této iluzi pohybu je dosaženo díky tzv. setrvačnosti lidského oka, která je způsobena tím, že lidské smysly a nervová soustava mají určitou setrvačnost, takže při dostatečně rychlém zobrazování obrázků je od sebe lidské oko a mozek nedokáže odlišit a v mozku vzniká právě ona iluze pohybu. Rychlost zobrazování se udává jako snímková frekvence neboli počet snímků za sekundu a jako jednotka se používá FPS nebo Hz. [50]

Plynulost animace závisí právě na její frekvenci, kde, pokud je příliš nízká, animace nebude plynulá a půjdou rozlišit jednotlivé snímky. V podstatě platí, že čím vyšší frekvence, tím bude animace plynulejší a citlivější. Běžně se využívají frekvence 24, 25, 30 a 60 FPS, kde například u filmů se obvykle používá frekvence 24 FPS. [49]

### 3.2 Typy animací

Snímky animace mohou být vytvořeny různými postupy, a to třeba ruční malování snímků, fotografování objektů nebo se počítačově generují. S každým těchto postupů je možné vytvořit 2D a 3D snímky a poté i celé animace. [51]

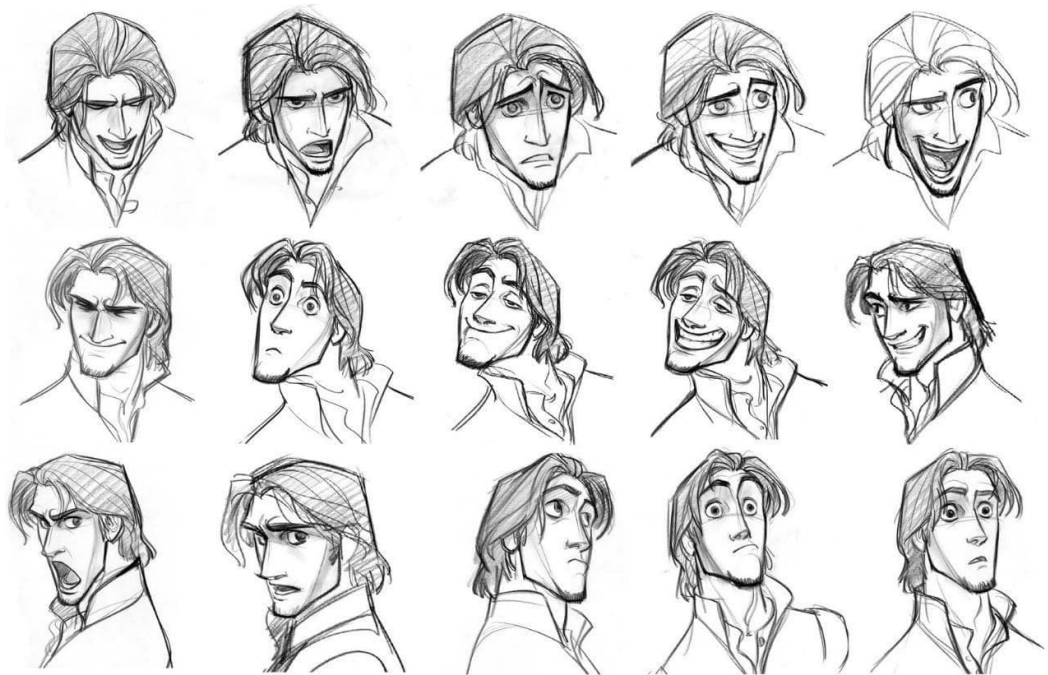
Na použitém postupu také závisí, o jaký typ animace se bude jednat. Rozlišuje se několik základních typů.

#### 3.2.1 Tradiční animace

Jedná se o jeden z nejstarších typů animace a byla nejvíce využívána během 20. století. [52]

Jak již z názvu může vyplývat, jednotlivé snímky jsou ručně malovány animátorem. Nákresy jsou poté nahrány do počítače, kde se pomocí softwaru zkombinují se zvukem a vzniká výsledná animace. [53]

Dříve se nákresy dělaly na průsvitný papír, které byly položeny na podsvíceném stole tak, aby animátor viděl předchozí kresbu. Kresby se poté překreslily na průsvitné celuloidy, kde se i vybarvily, a následně se tyto celuloidy dávaly postupně před připravené pozadí a byly vyfotografovány. S příchodem moderních technologií se proces o něco zjednodušil a přibýly i možnosti pro tvorbu nákresů a pozadí, a to vytvářet je na grafických tabletech či přímo na počítačích, a v případě papírových kreseb přibyla možnost je do počítače dostat pomocí skeneru. [52] [54]



Obrázek 16: Tvoření snímků pro tradiční animaci [55]

Co lze považovat za nevýhodu je to, že tradiční animace je pracná a časově náročná právě kvůli ručnímu vytváření snímku, obzvláště pokud je animace dlouhá a má běžnou frekvenci 24 FPS. [53]

Nejlepším příkladem této animace jsou známé animované filmy, jako jsou *Lví král* (1994), *Alladin* (1992), *Pinocchio* (1940) či *Farma zvířat* (1954). [52] [54]

### 3.2.2 Počítačová animace

Počítačová animace, též známá jako digitální animace, je forma tzv. CGI animace. Tyto dvě techniky animace se od nicméně liší, a to v jejich zaměření. Zatímco CGI se zabývá jak statickými, tak dynamickými scénami a obrázky, počítačová animace je zaměřená jen na pohyblivé snímky. [56] [57]

Příchod počítačové animace značně usnadnil tvorbu již existujících typů animací a zároveň přinesl i nespočet nových možností pro vytváření a práci s animacemi. Z toho vyplývá, že počítačové animace nabízí různé techniky, jak onu animaci vytvořit.

- Tradiční (Frame-by-frame) – princip vytváření je shodný s klasickou tradiční animací čili ruční vytváření každého snímku zvláště, jen zde se snímky tvoří rovnou v digitální podobě. [56]

- Motion capture – jedná se techniku, která využívá senzory tzv. trackery k zaznamenání realistického a přesného pohybu lidí, zvířat nebo i některých objektů. Zaznamenávat lze nejen samotný pohyb, ale i třeba mimiku obličeje, přičemž trackery se umísťují buď přímo na kůži, nebo na látku. Takto získaná data jsou poté převedena a použita pro vytvoření pohybu grafického objektu v animaci. [58] [56]



Obrázek 17: Získání pohybu přes motion capture [59]

Jedná se o velmi oblíbenou techniku díky relativně snadnému získání pohybových dat objektu i přes problémy spojené s touto technikou. Mezi ně patří především chybová data, ke kterým dochází při posunu trackerů během pohybu zaznamenávaného objektu, přičemž často k tomu dochází u lidí. Obtíže s animací nastávají také v případech, že grafický objekt má jiné rozměry než objekt, jehož pohyb se snímá. Jisté problémy také mohou nastat na základě použitých senzorů. U magnetických senzorů se jedná o šum v datech způsobený kovem v prostředí, při použití senzorů, které potřebují připojení kabelem k počítači, může dojít omezení možností pohybu a při použití optických senzorů dochází ke zmatkům v datech, který je způsobený částmi těla nebo jinými předměty, které naruší signál senzorů. [58]

- Procedurální animace – jedná se o skupinu technik, které vytváří počítačem vygenerované pohyby v reálném čase, přičemž počítač při generování postupuje podle kroků a pravidel, které jsou nastavené v algoritmu. Tyto techniky jsou často používány pro vytvoření skupiny podobných pohybů nebo pro pohyby, který by byly příliš

komplikované na to, aby byly tvořeny ručně, například pohyb vody a jiné podobné systémy a povrchy. [56] [58]

- Behaviorální animace – je to technika, patřící do skupiny procedurálních animací a umožňuje, že autonomní objekt může do jisté míry určovat své vlastní akce a chování. Jsou nastavena pravidla, jak se objekt může v daném prostředí chovat, a to dává objektu jistou možnost improvizovat a není tím pádem potřeba specifikovat každý detail pohybu objektu. Příkladem takové animace může být stádo zvířat, dav lidí, hejno ptáku nebo ryb, kde budou použita pravidla pro velké množství pohybujících se objektů. [56] [58]
- Fyzikálně založená animace (FZA) – jedná se taktéž o techniku ze skupiny procedurálních animací, která pro generování pohybu používá fyzikální zákony. Při zachycení důležitých fyzikálních charakteristik prostředí či situace lze vytvořit realistický pohyb objektu. FZA taktéž umožňuje pohyb objektu i dynamicky přizpůsobovat změnám v prostředí. Využívána je ve videohrách (*FIFA*, *Assassin's Creed*), filmech či simulacích. Tato technika se používá například pro vytvoření animace kouře, tekutiny, vlasů, listů, ale i lidí nebo zvířat. Nevýhodou této techniky je ovšem fakt, že vytvoření takové animace může být náročné a vyžaduje velkou znalost a chápání fyzikálních zákonů. [58] [60]
- Keyframing – velmi důležitá a používaná technika při níž animátor navrhne pouze klíčové snímky a zbylé mezi snímky jsou vytvořeny počítačem za pomoci interpolace mezi pozicemi na klíčových snímcích. [56] [58]
  - V klíčových snímcích jsou obsaženy významné změny objektu, jako jsou změna velikosti, pozice, směru, tvaru a dalších vlastností. [61]
  - Kvalita výsledné animace a vytvořeného pohybu objektu závisí také na použité interpolační technice. Mezi ty patří lineární interpolace, splines nebo třeba inverzní kinematika. Příklad keyframingu může být třeba odpal míčku v baseballu. Klíčovými snímky budou počáteční pozice pálkaře, střet pálky s míčkem a poté finální pozice pálkaře po odpalu. [58]
- Morphing – je to technika animace, kde je objekt plynule transformován do jiného objektu. Taková transformace je komplexní proces, pro který je potřeba spousta přesných měření a dat, mezi které spadá například znalost zdrojového a cílového objektu, důkladně zmapované a změřené tvary objektů či definované důležité body, jak

ve zdrojovém, tak cílovém objektu, aby souhlasily klíčové oblasti, jako jsou třeba oči. Ukázkový příklad morphingu je ve filmu *Terminator 2: Judgement Day* (1991), kde jsou pomocí morphingu vytvořeny scény, ve kterých se hlavní postava T-1000 transformuje do jiných lidí a objektů. [62]

Dalšími příklady jsou film *Flight of the Navigator* (1986) nebo hudební videoklip *Black or White* (1991) od Michaela Jacksona. [62] [58]

- 3D animace – forma animace, která je založená na 3D modelech. Tyto modely jsou vymodelovány umělci a disponují kostrou nebo jiným podobným systémem tak, aby animátoři mohli pohybovat jednotlivými částmi, jako jsou třeba končetiny, části obličeje, oblečení a další. [63]

Pro vytvoření animace je poté využito keyframing, takže animátor vytvoří klíčové snímky s významnými pozicemi modelu a počítač vypočítá přechod mezi klíčovými pozicemi a vygeneruje mezi snímky. Tento proces generování je známý jako „tweening“ a pokud jsou klíčové snímky pečlivě a chytře rozloženy, je možné vytvořit plynulou a realistickou animaci. [63]

Další způsob, jak vytvořit pohyb modelu, je využití motion capture. Herec předvádí požadované pohyby, které jsou snímány a zaznamenány. Takto nahrané pohyby jsou poté aplikovány na 3D model. Velké množství 3D animací je dnes tvořeno pomocí kombinace keyframingu a motion capture. [56]

Nakonec přichází na řadu tzv. rendering, jelikož tvorba 3D animace se skládá ze 3 částí: modelování, rozvržení a animace, rendering. [56]

Rendering převádí modely a jejich pohyb na obrázky, ze kterých je poté složena sekvence obrázků, obvykle při rychlosti 30 FPS. [58] [63]

Pokud by se 3D a 2D animace mezi sebou porovnaly, nelze jednoznačně říct, která z nich je lepší. Každá má svoje výhody a nevýhody. 3D animace například nabízí větší realističnost, možnosti a opětovné použití modelů, nicméně může být často složitější a dražší na vytvoření. Oproti tomu 2D animace je většinou levnější, jednodušší na výrobu a snáze se na nich provádí změny, ale zase trpí nedostatkem dynamiky, omezeným opětovným použitím a taky tím, že dnes už zkrátka není tak zajímavá a je tím pádem méně žádaná. To ovšem neznamená, že by se přestala používat a o typu zvolené animace budou rozhodovat především potřeby projektu, který bude animaci obsahovat, než vyloženě ony výhody a nevýhody daného typu animace. [63]

Nakonec, co se týče příkladů 3D animace, tak s nimi se setkal už dost

pravděpodobně každý a nezáleží na tom v jaké formě. Asi nejlepší příklady jsou filmy *Avatar*, které využívají 3D animaci na nejvyšší možné úrovni. Dále jsou také dobrými příklady animované filmy, jako jsou *Toy Story* (1995), *Shrek* (2001), *Madagascar* (2005), *Finding Nemo* (2003), anebo také počítačové hry jako *Call of Duty*, *The Legend of Zelda: Ocarina of Time* nebo *Metal Gear Solid*. [64]

### 3.2.3 Stop Motion

Podobně jako u tradiční animace se zde tvoří snímek po snímku. Liší se v tom, že zde se manipuluje s fyzickým objektem a postupně se fotografuje a zachycuje jeho stav tak, aby výsledkem byla ona iluze pohybu daného objektu. Jelikož na každém snímku je objekt trochu změněný, je vhodné si zvolit objekty, se kterými se dobře manipuluje a jsou poddajné. Právě na základě použitého objektu či materiálu se stop motion animace dále dělí na podtypy, nicméně princip vytváření animace zůstává u všech stále stejný. [65] [66]

- Pixilace – jako objekty jsou zde využíváni herci, kteří postupně zaujmají pózy a po každém snímku se trochu pohnou. Tato technika je využita například v krátkých filmech *Hôtel électrique* (1908), *Neighbours* (1952) nebo v českém krátkém filmu *Jidlo* (1992). [66] [67]
- Claymation – objekty jsou vyrobeny z materiálu, který je tvárný a nabízí široké pole úprav. Nejčastěji se používá jíla nebo plastelína, která se poté běžně dává na drátěnou kostru a následně se s těmito objekty pracuje. Mezi známé filmy a série, využívající tuto techniku, patří například *Shaun the Sheep* nebo *Wallace & Gromit*. [65] [66]
- Objektová animace – jedná se o nekreslené, pevné objekty, které nejsou plně tvárné, jako jsou hračky, LEGO kostky, panenky, postavičky a další. Příkladem staré objektové animace je krátký film *The Humpty Dumpty Circus* (1898) a novější ukázkou je série *Robot Chicken* (2005-2022). [65] [66]
- Cutout motion – používají se ploché části či pozadí, které jsou vystřižené z papíru, látky nebo fotografií. Tato technika je využita v *Monty Python's Flying Circus* (1969–1974), *South Park* (1997) anebo ve *Fantastic Planet* (1973). [65] [66]
- Loutková animace – využívá loutky, které se pohybují v kloubech a animátor s ní po každém snímku lehce pohne. [66]

Existuje ještě technika, která souvisí s loutkovou animací, a to tzv. Puppeton, kde se místo práce s jednou loutkou použije pro každý snímek jiná verze dané loutky. [65]

[66]

S loutkovou animací se lze setkat například ve filmech *Fantastic Mr. Fox* (2009), *Coraline* (2009) či *The Nightmare Before Christmas* (1993). [66]

- Animace siluety – jedná se o variantu cutout motion animace, která pracuje se siluetami podsvícených plochých částí. Pomocí této animace jsou vytvořeny filmy *The Adventures of Prince Achmed* (1926), *Princes and Princesses* (2000) nebo i *Tales of the Night* (2011). [66]

### 3.2.4 Pohyblivá grafika

Lze to také nazývat jako grafika v pohybu. Jedná se o jakýkoliv pohyb grafických elementů, jako jsou obrázky, videa, text a další, které jsou často zkombinovány se zvukem či hudbou. [68]

Pohyblivá grafika ovšem není plnohodnotná animace, jako jsou třeba animované filmy. Od klasického animovaného filmu, který má hlavně upoutat a pobavit, odlišuje pohyblivou grafiku především její účel a ten je v první řadě informativní. Pohyblivá grafika může být udělána zábavně, poutavě nebo i zahrnovat určité emoce, nicméně primárním účelem zůstává vizuální komunikace a předání informace. [68] [69]

Pro lepší pochopení a představu, jak může taková pohyblivá grafika vypadat, je dobré se seznámit s jejím využitím. Používá se jako úvodní titulky v seriálech, konkrétně třeba v *Game of Thrones* (2011-2019), *True Detective* (2014), a v dalších podobných televizních relacích a kanálech. Uchytila se také v reklamní sféře, kde se používá k vytváření videí o produktu a umožňuje video udělat zábavnější, zajímavější a může lépe zaujmout zákazníky. Těž se používá i na digitálních billboardech, kde často nahradila statické formáty, a rovněž lze s její pomocí oživit nebo rovnou představit celou značku. Velmi často se pohyblivá grafika používá taktéž v edukativních videích jakožto pomůcka pro sledující k lepšímu pochopení a vizualizaci probíraného tématu. S rozmachem internetu a sociálních sítí se pohyblivá grafika přesunula i tam. Pomáhá především zlepšit uživatelské prostředí webových stránek nebo také může sloužit i jako forma zábavy. [69] [70]

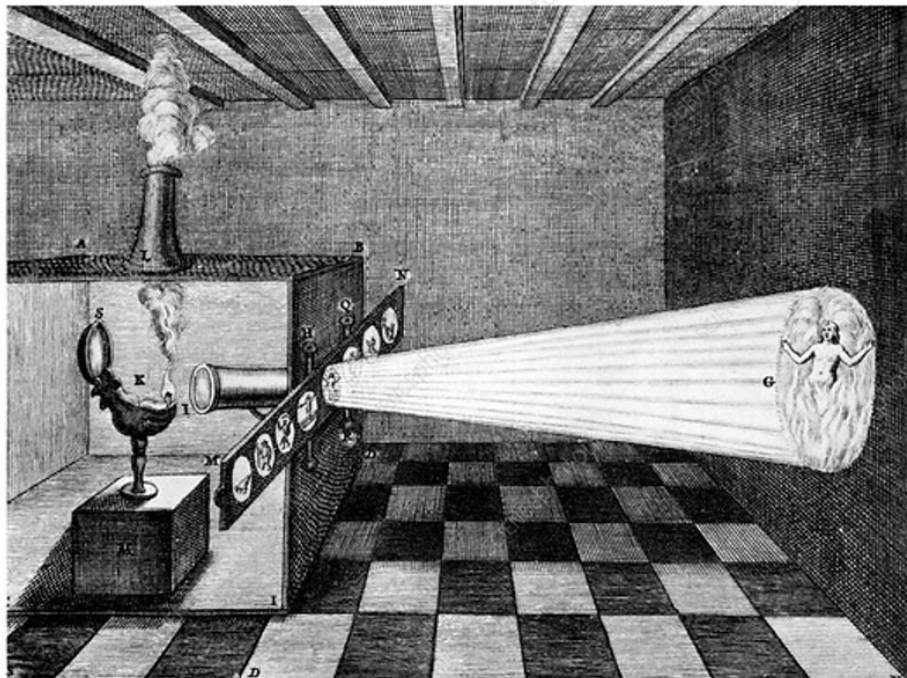


### 3.3 Historie animace

První náznaky snahy o zachycení pohybu se objevují již na jeskyních malbách z období paleolitu, na kterých jsou zakreslena zvířata s několika končetinami, které se překrývají. Na archeologickém nalezišti Shahr-e Sukhteh v jihovýchodním Íránu, které je nalezištěm městského osídlení z doby bronzové, byla nalezena hliněná mísa, na které jsou vyobrazeny kozy ve skoku. Stáří této mísy je odhadováno na 5000 let. [71] [72]

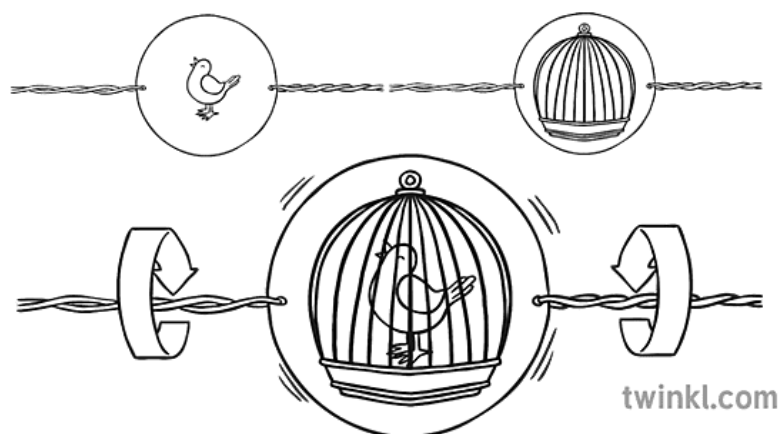
Tímto obdobím se nicméně dále zabývat nebudu a přesunu se do doby, konkrétně do 19. století, ve kterém vznikly jedny z prvních zařízení umožňující vytvořit animace, které byly pro toto odvětví klíčové a umožnily jeho další rozvoj.

Ještě předtím je ale dobré zmínit rok 1603, kdy byla vynalezena tzv. magická lucerna, která sloužila k promítání obrazu. Obsahovala zrcadlo za světelným zdrojem, které směřovalo světlo přes skleněné tabulky a promítalo ilustrace, jež byly na tabulkách nakresleny. Složením více takových tabulek dohromady vznikl pohyb, a proto se magická lucerna považuje za první příklad „pohyblivých obrázků“. [73] [74]



Obrázek 18: Magická lucerna [75]

Teď už ale přejdu do onoho 19. století, a to do roku 1824, kdy vznikl thaumatrope. Jedná se o otočný disk s rozdílným obrázkem na každé straně a držení dvěma provázky. Když se pomocí provázku disk roztočil, bylo možné vidět zkombinovaný obrázek, čehož bylo docíleno pomocí optické iluze zvané perzistence vidění. [73] [74]



Obrázek 19: Thaumatrope [76]

V roce 1832 vynalezl belgický fyzik Joseph Plateau přístroj pojmenovaný phenakistoscope. Je to rotující, malovaný disk, který při odrážení v zrcadle vytvářel iluzi pohybu. [73] [74]



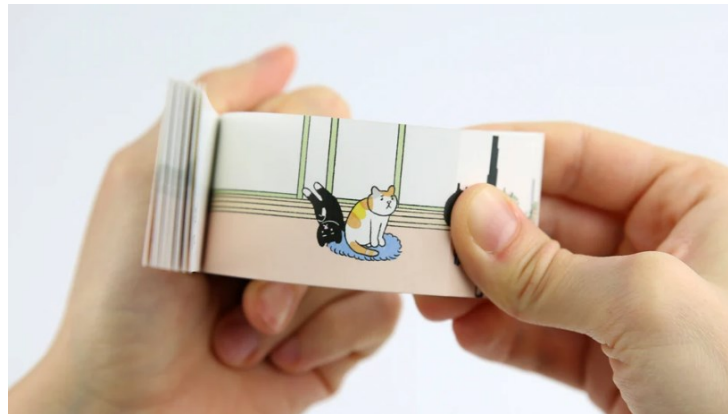
Obrázek 20: Phenakistoscope [77]

Nástupcem phenakistoskopu je zoetrop, který v roce 1834 sestrojil britský matematik William George Horner. Jedná se o dutý válec, který obsahuje dlouhý pás pokrytý obrázky a při rotaci vzniká iluze pohybu. Ve stranách válce jsou svislé štěrbiny, kterými se jednak diváci dívali dovnitř válce a také při rotaci zabraňovali rozmazání obrázků. Oproti phenakistoskopu má výhodu v tom, že pohyb obrázků může sledovat více lidí najednou. [74] [78]



Obrázek 21: Zoetrope [79]

Jako další byl kineograf nebo také flipbook, který si v roce 1868 nechal patentovat britský litografický tiskař John Barnes Linnett. Nejedná se o mechanické zařízení, nýbrž o knihu, která má na každé stránce kresbu v jiné poloze při pohybu, takže při rychlém obracení stran vzniká animace. [74]



Obrázek 22: Flipbbook [80]

Nakonec francouzský vynálezce Charles-Émile Reynaud představil v roce 1877 nástupce zoetropu, praxinoskop. Pracuje na stejném principu jako zoetrop, ale vylepšení spočívá v nahrazení svislých štěrbin vnitřním válcem posetým na úhlovými zrcadly, na kterých se tvořila ona animace. Tato zrcadla poskytovala divákovi jasnější a méně rozmazaný obraz než štěrbinu u zoetropu. [73] [74]

Praxinoskop byl poté použit v dalším vynálezu Charlese-Émile Reynauda. Jednalo se o systém pohybového obrazu zvaný Théâtre Optique, který si Reynaud nechal v roce 1888 patentovat a sloužil k promítání před divadelním publikem. Pomocí Théâtre Optique promítal Reynaud v roce 1892 v Musée Grévin v Paříži svůj krátký animovaný film

*Pauvre Pierrot* (1892), který je často označován za první animovaný film. Trval zhruba 15 minut a skládal se z 500 samostatně malovaných obrázků. [71] [74]



Obrázek 23: Praxinoskop [81]

Nedlouho poté následovali další významná jména, jejichž díla byla klíčová ve vývoji animace. Mezi ně patří britsko-americký filmový producent James Stuart Blackton a jeho díla *Humorous Phases of Funny Faces* (1906), což byl první zcela animovaný film nahraný na standardní filmovou pásku, a *The Haunted Hotel* (1907). V obou případech se jednalo o krátké filmy, které byly vytvořeny pomocí stop motion animace. [74] [78]

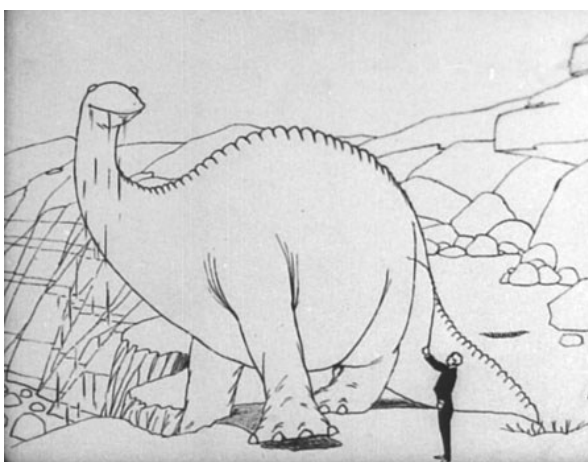
Podobnou formu animace, jako byla ta Blacktonova, vyvíjel i francouzský karikaturista Émile Cohl, celým jménem Émile Eugène Jean Louis Cohl, který byl Blacktonem značně inspirován. Cohl ale používal poměrně zjednodušený styl postav, narozdíl od Blacktonových propracovaných karikatur. [78]

Cohlův film *Fantasmagorie* (1908), se kterým debutoval, využívá tento styl a jedná se o první animaci vytvořenou pomocí tradičního způsobu čili ruční malování snímků a je považován za první kreslený film. [71] [73]

Mezi první průkopníky v oblasti animace patří také americký karikaturista Winsor McCay. Jeho skvělé kreslířské schopnosti jsou vidět ve významných dílech *Little Nemo in Slumberland* (1905–1911), *Dream of the Rarebit Fiend* (1904–1911) a *Gertie the Dinosaur* (1914), což je první kreslený film se zajímavou postavou, která vypadá, že má osobnost, vlastní život a vyvíjí se. Všechna tato díla byla klíčová pro transformaci animačního průmyslu. [78] [82]

McCayova tvorba čítá řadu dalších děl, například *The Sinking of Lusitania* (1918), a na tuto tvorbu navázal a rozšířil ji australsko-americký karikaturista a animátor Pat Sullivan. Ten pomohl talentovanému americkému animátorovi jménem Otto James Messmer, který je známý

díky své vymyšlené kreslené postavě černého kocoura Felixe. [78] [82]



Obrázek 24: Gertie the Dinosaur [83]

Felix the Cat, představený poprvé v roce 1919 v krátkých filmech *Feline Follies* (1919) a *Musical Mews* (1919), se rychle stal standardním modelem pro další kreslené postavy a je často považován za první kreslenou filmovou hvězdu. [73] [78]

Dalšími umělci byly například americký karikaturista Reuben Lucius Goldberg, americký karikaturista Harry Conway „Bud“ Fisher, který je tvůrce populárního novinového komiksu *Mutt a Jeff* (1907–1932), anebo americký karikaturista George Herriman, známý jako tvůrce novinového komiksu *Krazy Kat* (1913–1944). [78] [82]



Obrázek 25: Felix the Cat [84]

Celé toto období mezi počátkem 20. století a koncem 20. let 20. století se nazývá „The Silent Era“, jelikož filmy v tomto období byly nahrány bez synchronizovaného zvuku a jiných dialogů. Na tuto éru navazuje období zvané „The Golden Age“, které trvalo zhruba od počátku 30. let do konce 50. let 20. století, a vystupují v něm jedni z nejznámějších animátorů v historii. [73]

Tím prvním je Walt Disney. Spolu zakladatel firmy Walt Disney Productions a jeden z nejvýznamnějších producentů a inovátorů animace, který vytvořil řadu charakterů a postav,

které jsou populární dodnes, jako například Mickey Mouse, Donald Duck, Goofy, Pluto a další. [73]

Jeho první významný charakter byl Oswald the Lucky Rabbit, ze kterého poté vzešel další charakter, Mickey Mouse. Právě jeho třetí film *Steamboat Willie* (1928) byl revoluční v tom, že se jednalo o první kreslený film se synchronizovaným zvukem. [73] [78]

Na to navázal význačnými technikami, jako je přidání synchronizované hudby (*The Skeleton Dance*, 1929), filmový proces zvaný technicolor (*Flowers and Trees*, 1932) či více plošná kamera (*The Old Mill*, 1937). [78] [82]

Disneyho další ikonické dílo je *Snow White and the Seven Dwarfs* (1937), jelikož v něm byly použity nejlepší techniky, které v té době animace nabízela, a jednalo se o první ručně kreslený celovečerní animovaný film a zároveň o první takovýto film, který získal široké vydání v hollywoodském stylu. [73] [78]

Kromě zábavných filmů vytvářel Disney i filmy, které poskytovali drama a hlubší zážitek, jako jsou *Pinocchio* (1940), *Fantasia* (1940), *Dumbo* (1941), *Bambi* (1942), a u kterých byl kladen velký důraz na fotografický realismus. [78]

Mezi významné tvůrce patří i bratři Fleischerové, Max a Dave, kteří se stali úspěšnými animátory ještě před Waltem Disneym. Jsou to tvůrci techniky zvané rotoskopování, která se používá dodnes, a jedná se o překreslování živých záběrů jako obraz a poprvé ji použili v jejich seriálu *Out of the Inkwell* (1919–29). [78] [82]

Jejich nejznámější charakter, se kterým debutovali v roce 1930, je zpěvačka Betty Boop a její dva společníci Bimbo the Dog a Koko the Clown. Tyto postavy jsou příkladem Fleischerovic tvorby v první polovině 30. let, která byla opakem Disneyho tvorby. Byla temná, městská, drsná, sexuální, posedlá smrtí. [78] [82]

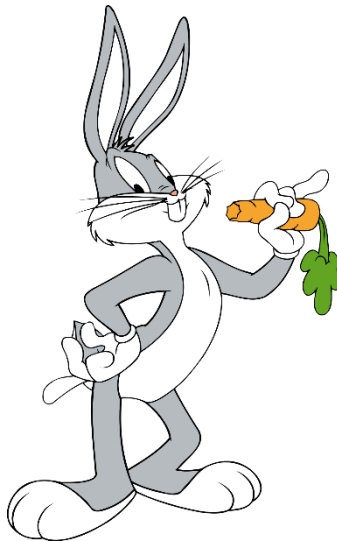
Nicméně v druhé polovině 30. let se svojí tvorbou začali stále více přibližovat té Disneyho, ale ne s velkým úspěchem. Vytvořili dva celovečerní filmy, *Gulliver's Travels* (1939) a *Mr. Bug Goes to Town* (1941). Jejich hlavní tvorbou nicméně zůstala Betty Boop, která skončila v roce 1939, a postava Popeye the Sailor, která byla představena v roce 1933, a seriál trval až do roku 1942, kdy jejich studio skončilo. [78]



Obrázek 26: Betty Boop [85]

S ještě více chaotickými animacemi přišlo studio Warner Bros. a především Tex Avery, který se přidal k Warner Bros. v roce 1936 jako režisér a lídr týmu animátorů. Jelikož se Warner Bros. nemohli po technické stránce rovnat Disney studios, snažil se Avery, aby jeho postavy a kreslené filmy byly co nejzábavnější a nejlépe napsané. [86]

V této snaze vytvořil se svým týmem nový druh postav. Byly cynické, rychlé, vtipné, často agresivní a násilné, jako byl Bugs Bunny ve filmu *A Wild Hare* (1940), který byl součástí série *Merrie Melodies* (1931-1969). Dalšími postavami byly Daffy Duck, Porky Pig, Elmer Fudd, Tweety, Sylvester, Pepe LePew, Foghorn Leghorn, Road Runner, Wile E. Coyote, které se objevovali napříč seriály *Looney Tunes* (1930-1969) a *Merrie Melodies* (1931-1969) od Warner Bros. [78][82]



Obrázek 27: Bugs Bunny [87]

V roce 1942 přešel Avery z Warner Bros. do studia Metro-Goldwyn-Mayer (MGM), kde se jeho tvorba stala ještě divočejší, a během osmi let v MGM vytvořil 67 kreslených filmů, jako jsou třeba *Who Killed Who?* (1943), *Batty Baseball* (1944), *Screwball Squirrel* (1944), *Red Hot Riding Hood* (1943) nebo *Bad Luck Blackie* (1949). [82] [86]

Avery měl po Disneyem druhý největší vliv na vývoj animace v Americe. [86]

Koncem 50. let 20. století se animační průmysl musel vypořádat s novou výzvou, a to byla zvyšující se popularita televize jako druhu zábavy. Začal se tedy této skutečnosti přizpůsobovat a tím v Americe začala „The television era“. [73]

Studia začala vytvářet animace pomocí techniky zvané „limited animation“, což je opětovné použití celých, nebo jen částí již vytvořených snímků a kreslení nových snímků se provádí jen v případě, že je to nutné. Pokud tedy postava mluví a zbytek jeho těla se nehýbe, není nutné překreslovat celý snímek, ale bude se překreslovat jen pusa, popřípadě výraz v obličeji, a zbytek snímku se použije z předchozích. [82] [88]

Jako prvním animovaným seriálem, který se v televizi vysílal v hlavním čase, byl *Flinstones* v roce 1960 od studia Hanna-Barbera. Dalším populárním seriálem od stejného studia byl v roce 1961 *The Yogi Bear Show*. [73] [82]

Známými tituly byly také film *Fritz the Cat* (1972), jelikož se jednalo o první dospělý animovaný film, které se v té době staly v televizi velmi populární, a krátký film *The Pink Panther* (1964), což byl film ze série *Pink Panther* a jednalo se první krátký animovaný film, který získal Academy Award. [73]

S příchodem televizním kanálů, jako byly The Disney Channel nebo Nickelodeon, se animované pořady staly v televizi běžnou a všudypřítomnou věcí. [73]

Od roku 1980 se animační průmysl začal transformovat do podoby, v jaké ho známe dnes, a to díky obrovským pokrokům ve vývoji 2D a 3D počítačové animace a CGI, které byly pro animaci revoluční. [82]

Jelikož vývoj je v tomto období už velmi rychlý a tvorba je celosvětově známá, zmíním jen pár příkladů.

Například *The Adventures of André & Wally B.* (1984), což byl krátký film, který byl jako první vytvořen kompletně pomocí CGI. Jako prvním počítačově animovaným celovečerním filmem byl *Toy Story* (1995) od studia Pixar. [73]

Další příklady 2D animací jsou *Beauty and the Beast* (1991), *The Lion King* (1994) či *Mulan* (1998). Za 3D animace to jsou třeba *Ice Age* (2002), *Shrek* (2001), *Madagascar* (2005). [82]



Většina výše popsané historie byla o vývoji animace v Americe. Vývoj animace v Evropě se ubíral jiným směrem a namísto kreslených animací, experimentovali evropští animátoři se značně odlišnými způsoby a technikami. Této části historie se však v mojí práci již dále věnovat nebudu.

### **3.4 Využití animací**

Ačkoli bylo využití animací zmiňováno napříč celou touto kapitolou, je dobré si ho shrnout, jelikož oblast využití je značně rozsáhlá.

Velmi rozsáhlé užití animací je v již zmíněném zábavním a herním průmyslu, což zahrnuje právě filmy, animované filmy, počítačové hry a další podobné formy animace dostupné na různých platformách, ať už v televizi nebo na internetu. [89] [90]

Velkou roli hrají animace také v reklamě a v marketingu. Umožňují organizacím ukázat produkt nebo značku kreativním a unikátním způsobem. Reklama je díky nim zajímavější, osloví více lidí a snáz zaujme divákovu pozornost a zapojí jeho představivost. Takovéto reklamy jsou populární a jsou stále více používané, jak v televizi, tak na internetu, a to na webových stránkách či sociálních sítích. [90]

Běžně jsou animace využívány i ve vzdělávání, jakožto nástroj buď pro lepší pochopení a vizuální zobrazení předávané informace, nebo přímo pro vysvětlení složité informace, která by se obtížně vysvětlovala jen za pomoci textu. Je totiž známý fakt, že lidé si spíše pamatují obrázky a videa než psaný text. Jedná se tak o efektivní a záživnější formu studijních a vzdělávacích materiálů jak pro studenty, tak i pro učitele. [90]

Animace jsou používány i v simulacích. Jedná se například o vojenské simulace, simulace zbraní, tréninkové simulace, letecké simulace pro nácvik pilotů nebo simulace akcí a událostí, u kterých je potřeba vědět, jak pravděpodobně dopadnou. Takovou akcí je třeba start a let rakety, kde jsou simulace nezbytné a mohou zabránit špatnému scénáři. [89] [90]

Široké uplatnění si animace našli ve vědecké vizualizaci, kde slouží pro vytváření realistických 3D modelů i pro účely výzkumu, analýzy a studia. [90]

Některé další oblasti, ve kterých se animace používají, jsou ve zkratce zdravotnictví, umění, architektura, AR a VR, strojírenský průmysl nebo softwarové inženýrství. [90]

## 4 PIKTOGRAMY

### 4.1 Definice

Piktogramy jsou typ symbolů a také s nimi velice často přicházíme do styku. Jelikož se s piktogramy bude v této práci dále pracovat, je potřebné si je podrobněji rozebrat.

Piktogramy vyjadřují objekt, místo nebo prostor bez použití psaného textu a informaci předávají tím, že jsou danému prvku vizuálně podobné. [91] [92]

Tím, že piktogramy sdílí vizuální podobnost se zobrazovaným objektem a nemíří na jeho název, tak by teoreticky mělo být možné je použít mezinárodně, jenže ne vždy je ale piktogram navržen tak, aby byl srozumitelný pro všechny a bylo opravdu možné ho globálně použít.

### 4.2 Historie

Piktogramy, společně s dalšími grafickými symboly, byly využívány už starověkými civilizacemi, jako byly starověký Egypt, Čína, Sumerové, Mezopotámie nebo i domorodými Americkými civilizacemi, například Aztékové a Mayové, a sloužily ať už jako forma umění, nebo i jako forma obrazového jazyka pro komunikaci. [91] [93]

Nicméně za úplně první piktogramy jsou běžně považovány pravěké jeskynní malby, které povětšinou zobrazovaly zvířata, předměty a další věci spojené s lovem. Takovými malbami jsou například jeskynní malby v jeskyních Altamira a Lascaux ve Francii. Z těchto prvních piktogramů šel dále jejich vývoj postupně od dalších piktogramů pro zobrazení lidí, předmětů a činností, následně přes jejich skládání do série pro předání určité informace, až po vytvoření komplexních psaných systémů. [94]



Obrázek 28: Piktogramy domorodých Američanů v Great Gallery, Horseshoe Canyon, Canyonlands National Park [95]

Takovým systémem je kupříkladu čínština, která je kompletně tvořena piktogramy a je používána dodnes. Jedná se o nejstarší z dnes používaných písem a od svého vzniku se téměř nezměnilo, vyjma navýšení počtu znaků a případných lehkých úprav. Podobný systém používali i Egypťané, a to konkrétně hieroglyfický, který byl taktéž složený z grafických symbolů a jedná se o ještě starší systém, než je právě zmíněná čínština, nicméně se dnes již hieroglyfy nepoužívají. [94]

Postupem času ale piktogramy začaly buď ztrácet původní význam, nebo docházelo k více různým výkladům, a proto se jejich počet postupně snižoval a psané písmo se stávalo více a více fonologické. [91]

To ovšem neznamená, že by se piktogramy přestaly v jakékoliv formě používat.

### 4.3 Moderní užití

Piktogramy si svůj účel zachovaly dodnes a v dnešní společnosti je jejich hlavní cíl přilákat na sebe pozornost a rychle předat potřebnou informaci. Nejčastěji jsou využívány jako značky, instrukce nebo diagramy. Příklady takovýchto piktogramů jsou silniční, varovné a informační značky nebo i matematická a statistická schémata a grafy, kde reprezentují data. [91] [93]

Piktogramy dnes také značně zjednodušují komunikaci s jakkoli znevýhodněnými lidmi, ať už se jedná o lidi s problémy se čtením, bez znalosti daného jazyka či z jiných kultur. V tom tkví ale taky nevýhoda piktogramů, jelikož skutečně univerzálních piktogramů není mnoho a vytvořit takové piktogramy, které budou srozumitelné a snadno pochopitelné pro všechny, i pro lidi z jiných kultur a zemí, je poměrně složité a časově náročné. Nicméně ani u těchto univerzálních piktogramů nelze úplně zamezit odlišným interpretacím či jejich nepochopení. [91]



Obrázek 29: Mezinárodní piktogram pro varování před ionizujícím zářením [96]

### 4.4 Standardizace

Proto, aby piktogramy bylo možné používat mezinárodně a byly opravdu univerzální a jazykově a kulturně nezávislé, jsou piktogramy často součástí globálních standardů, u kterých se předpokládá, že budou srozumitelné pro všechny.

Jedním takovým standardem je ISO 7001 „Public information symbols“, který definuje sadu piktogramů a značek společně s jejich užitím a umístěním, přičemž tyto piktogramy jsou testovány v několika zemích a různých kulturách tak, aby splnili stanovené podmínky. Do této sady patří například piktogramy pro označení parkování, informační symboly pro handicapované osoby a další veřejné informační symboly či označení dopravních zařízení. [91]

Další standartní sady jsou symboly užívané na oděvních štítcích, symboly pro označení chemického nebezpečí definovaného normou GHS nebo i tzv. DOT piktogramy, což je americká verze ISO 7001 symbolů, s nimiž v mnoha případech sdílí značnou podobnost. [93] [97]



Obrázek 30: ISO 7001 piktogram pro odlety letů [98]



Obrázek 31: DOT piktogram pro odlety letů [99]

## 5 AUTOCAD

Jelikož je praktická část této práce určena pro software AutoCAD, je dobré si specifikovat, o jaký software se vlastně jedná.

AutoCAD je vyvíjený společností Autodesk a jedná se o komerční návrhový software určený pro vytváření 2D a 3D grafiky, výkresů, plánů či modelů. Je taktéž součástí tzv. CAD programů, mezi kterými se řadí na absolutní špičku. Software je založený na vektorovém systému, avšak dokáže pracovat i s bitmapovými obrázky, které je možné importovat. [100] [101]

Díky svým vlastnostem a funkcím je AutoCAD velmi oblíbený a oblast jeho využití je značně rozsáhlá. To zahrnuje strojírenství a elektroinženýrství pro návrh plánů a modelů součástek, grafické designéry a jiné umělce, architekty a návrháře interiérů pro návrh plánů budov a vnitřních prostor, aeronautiku a mnoho dalších okruhů a profesí. [100] [102]

Historie AutoCADu sahá do roku 1982, kdy byla vydána jeho první verze skupinou programátorů, kterou vedl John Walker, jenž je zároveň spoluzakladatel společnosti Autodesk. Od té doby se jak Autodesk, tak AutoCAD podstatně rozrostly a po rok 2022 bylo vydáno 37 verzí AutoCADu a od roku 2010 pro iOS a od roku 2011 pro Android je dostupná i mobilní aplikace AutoCADu. [103]

AutoCAD používá formát DWG jako nativní typ souboru, který obsahuje všechny informace o nákresu, geometrii a použitých obrázcích. Společně s DXF, což může být buď binární, nebo ASCII reprezentace výkresového souboru, tvoří hlavní formáty pro výměnu dat AutoCADu s jinými programy. Software nicméně poskytuje širokou řadu dalších podporovaných formátů, do kterých umožňuje export a import. [104] [105]

## 6 MOŽNOSTI IMPLEMENTACE

V AutoCadu existuje více způsobů, jak praktickou část implementovat, jelikož k programování AutoCADu je možné použít několik různých rozhraní a programovacích jazyků.

### 6.1 AutoLISP/VisualLISP

AutoLISP je aplikační rozhraní pro automatizaci návrhových úloh a je založen na programovacím jazyce LISP. Kód AutoLISPU lze zadat buď do příkazového řádku, nebo ho načíst z externích souborů, což je možné díky vestavěnému interpretu LISP přímo v AutoCADu. [106]

Aplikace AutoLISP umožňuje přistupovat k vestavěným příkazům AutoCADu, editovat, pracovat a vytvářet nové objekty přímo ve výkresové databázi, přidávat specifické příkazy či vyzvat uživatele k zadání vstupní hodnoty. [106]

AutoLISP byl představen v AutoCADu verze 2.18 v roce 1986 a společnost Autodesk postupně AutoLISP vylepšovala a rozšiřovala. Nicméně od AutoCADu verze 14 v roce 1997 už Autodesk přestal AutoLISP dále vyvíjet a zaměřil se na další generaci jazyka LISP, Visual LISP. [107]

*„Visual LISP je rozšíření programovacího jazyka AutoLISP.“* [108]

Visual LISP je vývojové prostředí, které je zabudované v AutoCADu, a jeho součástí je i debugger, kompilátor a další nástroje umožňující například volání příkazů AutoCADu, systémových proměnných nebo dialogových oken. Rozšíření oproti AutoLISPU spočívá hlavně v tom, že Visual LISP má přístup k objektovému modelu AutoCADu. [108] [109] [110]

Jak Visual LISP, tak i AutoLISP, zůstal mezi uživateli AutoCADu velmi populární a jejich použití se odvíjí od toho, jak velkou kontrolu nad AutoCADem chtějí uživatelé mít. [108]

### 6.2 Visual Basic for Applications (VBA)

Jedná se o objektově orientované programovací prostředí vytvořené společností Microsoft a nabízí široké možnosti pro vývoj aplikací a automatizaci operací. Při spojení s AutoCADem poté VBA umožňuje si podle potřeby přizpůsobit AutoCAD všemožnými způsoby a poskytuje inteligentní a rychlé programovací prostředí, jelikož VBA běží ve stejném procesním prostoru jako AutoCAD. [111] [112]

I přesto, že je VBA podporováno produkty Autodesku, není součástí instalace AutoCADu a je potřeba si tento modul doinstalovat samostatně.

### 6.3 ActiveX

ActiveX je označení pro skupinu objektově orientovaných programovacích technologií, které umožňují programově manipulovat s AutoCADem. Toho je docíleno tím, že se objekty AutoCADu vystavují vnějšímu prostředí. Tyto objekty jsou hlavní elementy každé ActiveX aplikace, kde každý z nich představuje přesnou část AutoCADu. [113] [114]

Rozhraní ActiveX umožňuje programově přistupovat k datům AutoCADu i jiným programovacím prostředím a stejně tak usnadnilo sdílení dat s dalšími Microsoft aplikacemi, jako je Word nebo Excel. [114]

### 6.4 ObjectARX®

ObjectARX je objektově orientované programovací prostředí v kompilovaném jazyce. Prostředí slouží k vývoji AutoCAD aplikací a disponuje jak programovacími rozhraními C++, C# a VB .Net, tak i řadou dynamických knihoven (DLL), které s AutoCADem sdílí adresní prostor, díky čemuž mohou pracovat přímo s hlavní datovou strukturou AutoCADu a umožňují vytvářet přímé volání funkcí do AutoCADu. [115] [116]

Jelikož tyto knihovny využívají otevřené architektury AutoCADu, poskytují proto přímý přístup ke geometrii, grafickému systému, k databázovým strukturám AutoCADu, k definici nativních příkazů a mimo to i sady všestranných nástrojů, které jsou užitečné zejména pro vývojáře AutoCAD API. [115] [116]

ObjectARX knihovny se také dají použít v kombinaci s jinými programovacími rozhraními, jako je VBA, AutoLisp nebo ADS (AutoCAD Development System). [115]

### 6.5 C# .NET

Zde se jedná o 2 samostatné pojmy. C# je objektově orientovaný programovací jazyk běžící na .NET Frameworku a jeho využití je značně rozsáhlé. Jedná se hlavně tvorbu webových aplikací, mobilních aplikací, databázových programů a mnoha dalších. [117]

.NET je open source vývojářská platforma vytvořená společností Microsoft a je určena pro vytváření webových, desktopových a mobilních aplikací, které lze spustit na různých operačních systémech. [118] [119]

K tomu .NET obsahuje knihovny, nástroje, a jazyky, které umožňují tvorbu výkonných a moderních aplikací. [119]

Aplikace .NET lze vytvářet v několika podporovaných jazycích, a to v C#, F#, C++, Visual Basic a jazyky CLI (Common Language Infrastructure) mezi které patří například ClojureCLR, Eiffel či IronPython. [118] [119]

.NET je k dispozici v několika implementacích, které především určují, na jakých operačních systémech bude možné aplikace spustit. .NET Framework je první implementace .NET, která dnes již není aktivně vyvíjena a běží pouze na Windows. .NET Core je novější multiplatformní verze, která nahrazuje starší .NET Framework. Další implementace jsou .NET Standard, Mono.  
[119] [120]



## 7 PRAKTICKÁ ČÁST

V teoretické části jste se seznámili s několika symbolickými jazyky. Cílem praktické části je ukázat, jak za pomoci těchto symbolických jazyků vytvořit nápovědu pro AutoCAD, která by nahradila stávající čistě textovou nápovědu. Má také ukázat, jak by symbolické jazyky usnadnily práci tvůrcům, kteří by nemuseli vymýšlet složité definice a popisy a zároveň by buď plně odpadla, nebo by se výrazně snížila náročnost překladu nápovědy do všech národních lokalizací. Pro uživatele by poté takové nápovědy byly příjemnější, rychlejší a lepší pro pochopení dané věci.

### 7.1 Implementace

Se způsoby implementace jste se již seznámili, takže nyní již přistoupím ke zdůvodnění zvolené implementace, která je v praktické části použita. Zvolil jsem C# .NET, a to primárně z toho důvodu, že s ním mám, oproti ostatním způsobům implementace, největší zkušenosti a velkou výhodou je také to, že k němu existuje velké množství manuálů a jiných materiálů. Jako vývojové prostředí, dále IDE, jsem si vybral Visual Studio 2019, také na základě předchozích zkušeností. Co se týče operačního systému, tak používám Windows 10 Professional. AutoCAD jsem zvolil ve verzi 2022 v české lokalizaci.

Tolik ke zvolené implementaci a prostředí a teď se přesunu k rozboru vytvoření a nastavení nejen projektu, ale i samotného IDE. Ještě před vytvořením projektu je dobré si zkontrolovat, zda máte v IDE nainstalované moduly, které jsou potřebné pro vytváření a kompilování aplikací pomocí jazyka C#. Jedná se o moduly s názvem *Vývoj desktopových aplikací pomocí .NET* a *Vývoj pro Univerzální platformu Windows*, které lze nainstalovat buď při prvotní instalaci softwaru, nebo je možné si je nainstalovat dodatečně v instalátoru IDE.

Po zkontrolování přítomnosti modulů je dalším krokem vytvoření projektu. Ze zobrazeného listu různých typů šablon projektů vyberte šablonu zvanou *Knihovna tříd (.NET Framework)*, projekt pojmenujte, vyberte jeho umístění a architekturu nastavte na .NET Framework 4.7.2. V mém případě ke projekt pojmenován *Autodesk\_help*.

Tím je projekt vytvořený, ale ještě předtím, než ho začnete používat, je nutné provést počáteční nastavení. Jako první je potřeba přidat odkazy na AutoCAD .NET API soubory, se kterými se bude dále pracovat. Jedná se o tři knihovny, nebo také soubory DLL. Jsou to knihovny *AcCoreMgd.dll*, *AcDbMgd.dll*, *AcMgd.dll*.

Obecná cesta do adresáře, ve kterém se knihovny nachází je:

```
<disk>:\Program Files\Autodesk\<verze>
```

V mém případě vypadá cesta následovně:

```
C:\Program Files\Autodesk\AutoCAD 2022
```

Druhá možnost je, že pokud máte v AutoCADu nainstalovaný ObjectARX SDK, tak najdete potřebné knihovny i ve složce *inc* v hlavní instalační složce. Po přidání odkazů na tyto knihovny je nutné u nich nastavit vlastnost *Kopírovat místně* na false. Jako další ve vlastnostech projektu, v sekci *Ladit*, v části *Spustit akci* vyberte možnost *Spustit externí program*, kam zadejte cestu ke spustitelnému souboru AutoCADu. V mém případě je cesta k souboru následující:

```
C:\Program Files\Autodesk\AutoCAD 2022\acad.exe
```

Je to důvodu, aby při testování projektu rovnou došlo ke spuštění programu AutoCAD.

Nyní máte projekt nastavený a zbývá ještě připravit soubor a třídu, do které budete psát zdrojový kód. Aby kompilátor věděl, kde má hledat třídy, které budete ve zdrojovém kódu používat, musíte přidat jmenné prostory do hlavního souboru, který vznikne během vytvoření projektu a jmenuje se stejně jako projekt. Základní jmenný prostor, který budete potřebovat, je *Autodesk.AutoCAD.Runtime* a přidáte ho pomocí klíčového *using*.

Vedle tohoto jmenného prostoru je dobré zmínit i další jmenné prostory, které budete potřebovat v případě, že byste chtěli kód dále rozšiřovat o pokročilejší funkce. Jedná se o jmenné prostory *Autodesk.AutoCAD.ApplicationServices*, který obsahuje třídy, jenž umožňují pracovat s aplikací jako s celkem. Další jsou *Autodesk.AutoCAD.DatabaseServices* a *Autodesk.AutoCAD.EditorInput*, který zahrnuje třídy a prostředky pro čtení a zápis do příkazového řádku a interakci s uživatelem, například aby uživatel poskytl souřadnici, vybral objekt atd.

Po přidání jmenných prostorů máte přípravu souboru hotovou a můžete se přesunout k samotné tvorbě funkcí a příkazů.

```

using Autodesk.AutoCAD.Runtime;

namespace Autodesk_help
{
    public class Autodesk
    {
        //atribut CommandMethod, který má jako argument název příkazu
        [CommandMethod("DimTadHelpComboBox")]
        //funkce spojená s atributem CommandMethod, spustí se při zadání
        //příkazu v argumentu atributu
        public void DimTadComboBox()
        {
            DIMTAD_ComboBox dimTadCB = new DIMTAD_ComboBox();
            //zobrazení formuláře
            dimTadCB.Show();
        }
    }
}

```

Zdrojový kód 1: Ukázka atributu CommandMethod a s ním spojené funkce (Zdroj vlastní)

Ve Zdrojovém kódu 1 je jednak názorně ukázáno přidání jmenného prostoru a ve třídě *Autodesk* můžete vidět funkci, která zobrazí formulář s nápovědou pro systémovou proměnnou DIMTAD. Tato funkce je spojena s atributem *CommandMethod*, který má nastavený jeden argument, a to název příkazu. To znamená, že při zadání tohoto názvu příkazu, *DimTadHelpComboBox*, do příkazového řádku v AutoCADu, dojde k zavolání funkce spojené s tímto s příkazem, tedy funkce *DimTadCombo*.

Poslední věcí, kterou je potřeba udělat, je sestavit řešení projektu, čímž vznikne soubor DLL onoho projektu, který bude poté použit v AutoCADu. Tím je část C# projektu hotová a zbývá ještě připravit druhou část, AutoCAD.

V případě AutoCADu to bude značně jednodušší, jelikož stačí provést standartní instalaci, otevřít nový nebo již existující výkres a přidat zmíněný soubor DLL vytvořeného C# projektu. To lze provést dvěma způsoby. První způsob je ruční přidání souboru DLL, které se provádí pomocí příkazu *NETLOAD* zadaného do příkazového řádku AutoCADu. Tím se otevře průzkumník souborů a Vy přejdete do adresáře C# projektu a vyberete soubor DLL. Obecné umístění souboru je:

Název\_projektu\bin\Debug\Název\_projektu.dll.

Moje cesta k souboru je:

Autodesk\_help\bin\Debug\ Autodesk\_help.dll

Druhý způsob přidání je automatické načtení souboru DLL při otevření AutoCAD souboru. Pro tento způsob je potřeba nejdřív zkopírovat soubor DLL z adresáře projektu, který je uveden výše, do adresáře:

```
C:\Program Files\Autodesk\ApplicationPlugins
```

Jako další musíte otevřít soubor *acad20xxdoc.lsp*, který najdete v adresáři:

```
C:\Program Files\Autodesk\<verze>\Support\<jazyk>
```

Tato cesta do adresáře a název souboru jsou v obecné podobě. Můj soubor je pojmenován *acad2022doc.lsp* a cesta do adresáře je:

```
C:\Program Files\Autodesk\AutoCAD 2022\Support\cs-cz
```

Soubor *acad2022doc.lsp* otevřete v libovolném editoru ASCII souborů, jako je například Notepad. Pokud byste v adresáři tento soubor nenašli, musíte takto pojmenovaný soubor v editoru ASCII souborů vytvořit. Do tohoto souboru vložte příkaz:

```
(command "_netload""C:/ProgramFiles/Autodesk/ApplicationPlugins/Autodesk_help.DLL)
```

Pokaždé, když teď otevřete nový nebo již existující soubor v AutoCADu, načte se automaticky soubor DLL C# projektu.

V případě automatického načtení je ovšem potřeba dát si pozor, pokud v C# projektu načítáte jiné soubory, například obrázky, a používáte k těmto souborům relativní cesty, které vypadají třeba takto *../Pics/DIMRND.jpg*. Jelikož přesouváme soubor DLL do jiného adresáře, dojde k tomu, že relativní cesty nebudou fungovat a musíte je vyměnit za absolutní cesty k souboru, které vypadají například takhle

```
C:/ProgramFiles/Autodesk/ApplicationPlugins/Pics/DIMRND.jpg.
```

Nastavení AutoCADu je tímto hotové a zbývá poslední věc. Součástí přílohy je složka *Pics*, která obsahuje potřebné obrázky, jež jsou v nápovědě využity. Tuto složku si vložte k sobě do počítače a popřípadě ve zdrojových kódech C# upravte cesty k těmto souborům.

V mém případě je složka umístěna v adresáři:

```
C:\Program Files\Autodesk\ApplicationPlugins
```

Ve zdrojových kódech poté cesty k souborům vypadají takto:

```
C:/ProgramFiles/Autodesk/ApplicationPlugins/Pics/Název_souboru.jpg.
```

Jelikož v práci používám automatické načtení souboru DLL do AutoCADu, jsou proto ve zdrojových kódech využity absolutní cesty. Při použití absolutních cest je možné si složku se soubory vložit do libovolného adresáře, nicméně doporučuji ji mít ve stejném adresáři jako soubor DLL C# projektu.

V případě, že byste chtěli ve zdrojových kódech použít relativní cesty, museli byste využít ruční přidání souboru DLL a složku by bylo nutné umístit do adresáře:

```
Název_projektu\bin
```

Cesty k souborům budou poté vypadat takto:

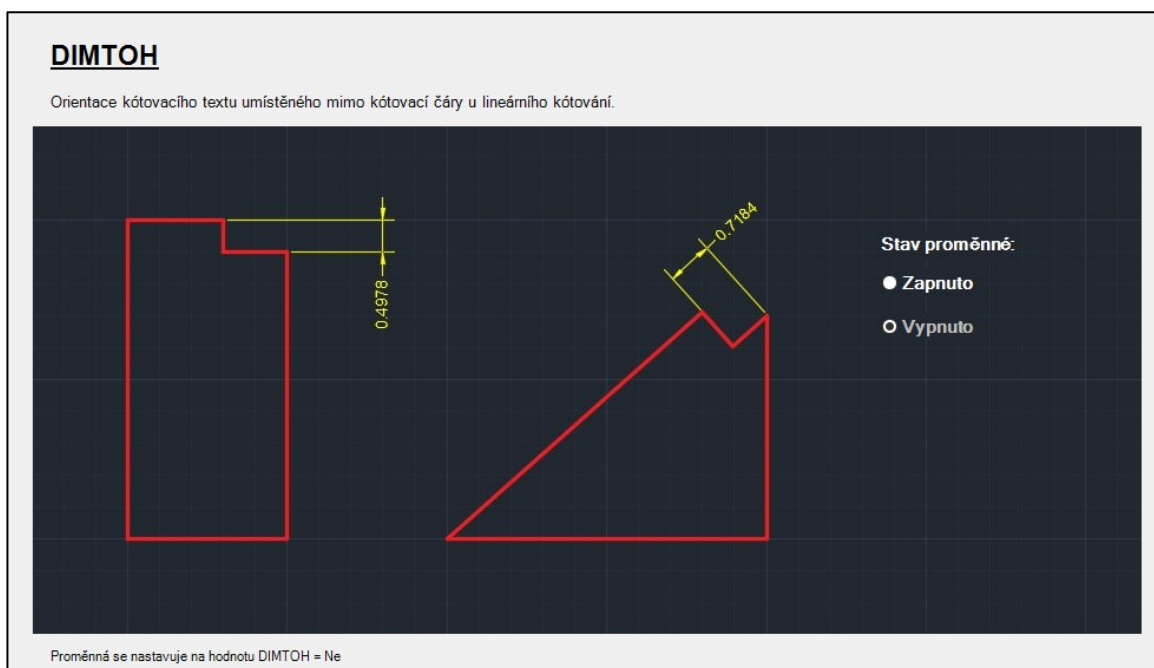
```
../Pics/Název_souboru.jpg
```

Tím je implementace projektu hotová. Jak jsem již na začátku zmínil, celá implementace probíhala na operačním systému Windows 10. Na jiných operačních systémech, jako třeba Linux nebo MacOS, bude implementace velmi podobná. Lišit se bude akorát v umístění souborů, cestách do adresářů a do IDE bude případně potřeba nainstalovat moduly potřebné pro vývoj aplikací pomocí C# na daném operačním systému.

V příloze naleznete obrázky z nastavení Visual Studia 2019, projektu a AutoCADu.

## 7.2 Struktura formuláře s nápovědou

Samotná nápověda pro systémové proměnné je vytvořena za pomoci formuláře Windows Forms, což je framework pro tvorbu formulářových aplikací prostřednictvím grafického designeru, který pro to obsahuje sadu ovládacích prvků, označovaných také jako komponenty. Pro přiblížení, z jakých komponent je formulář s nápovědou sestaven a jaká je jeho funkčnost, vybral jsem si k tomu nápovědu pro systémovou proměnnou DIMTOH. Základem je prázdný formulář, ve kterém je hlavní komponentou PictureBox, která zobrazuje největší obrázek s objekty v AutoCADu. Další komponenta je Label, která je využita především k zobrazování textu.



Obrázek 32: Ukázka nápovědy pro systémovou proměnnou DIMTOH (Zdroj vlastní)

Na Obrázku 32 je Label konkrétně využit pro nadpis a vysvětlivku na začátku nápovědy, nadpis nad RadioButtonem a doplňující informace dole pod obrázkem. Poslední komponenta je RadioButton, které jsou v tomto případě interaktivní a slouží k výběru stavu proměnné. Při kliknutí na RadioButton se změní barva jeho názvu, změní se text v komponentě Label dole pod obrázkem tak, aby odpovídal hodnotě, která se nastaví do proměnné v AutoCADu, ale především se změní obrázek, který zobrazuje to, co proměnná ovlivňuje a jak se změna hodnoty proměnné projeví na nákrese v AutoCADu. Toho je docíleno pomocí dvou událostí, které zajišťují onu interaktivitu RadioButtonů. Kód těchto událostí můžete vidět ve Zdrojovém kódu 2 společně i s počátečním nastavením komponent a hodnot.

```

public partial class DIMTOH : Form
{
    public DIMTOH()
    {
        InitializeComponent();

        //nastavení obrázku v hlavním PictureBoxu
        pictureBoxImage.Image = Image.FromFile("C:/Program Files/" +
            "Autodesk/ApplicationPlugins/Pics/DIMTOH_OFF.jpg");
        //nastavení doplňující informace v Labelu pod obrázkem
        labelNastaveni.Text = "Proměnná se nastavuje na hodnotu DIMTOH = Ne";
        //nastavení výchozího aktivního RadioButtonu
        radioButtonVypnuto.Checked = true;
        //nastavení ikony kurzoru při najetí na komponentu
        radioButtonVypnuto.Cursor = Cursors.Hand;
        radioButtonZapnuto.Cursor = Cursors.Hand;
    }

    //událost při kliknutí na RadioButton Zapnuto
    private void radioButtonZapnuto_Click(object sender, EventArgs e)
    {
        //změna obrázku v hlavním PictureBoxu podle aktivního RadioButtonu
        pictureBoxImage.Image = Image.FromFile("C:/Program Files/" +
            "Autodesk/ApplicationPlugins/Pics/DIMTOH_ON.jpg");
        //změna doplňující informace v Labelu pod obrázkem
        labelNastaveni.Text = "Proměnná se nastavuje na hodnotu DIMTOH = Ano";
        //změna barvy textu u RadioButtonů
        radioButtonZapnuto.ForeColor = System.Drawing.Color.Silver;
        radioButtonVypnuto.ForeColor = System.Drawing.Color.White;
    }

    //událost při kliknutí na RadioButton Vypnuto
    private void radioButtonVypnuto_Click(object sender, EventArgs e)
    {
        //změna obrázku v hlavním PictureBoxu podle aktivního RadioButtonu
        pictureBoxImage.Image = Image.FromFile("C:/Program Files/" +
            "Autodesk/ApplicationPlugins/Pics/DIMTOH_OFF.jpg");
        //změna doplňující informace v Labelu pod obrázkem
        labelNastaveni.Text = "Proměnná se nastavuje na hodnotu DIMTOH = Ne";
        //změna barvy textu u RadioButtonů
        radioButtonVypnuto.ForeColor = System.Drawing.Color.Silver;
        radioButtonZapnuto.ForeColor = System.Drawing.Color.White;
    }
}

```

Zdrojový kód 2: Ukázka části kódu nápovědy pro systémovou proměnnou DIMTOH (Zdroj vlastní)

Z kódu je patrné, že při otevření nápovědy je výchozí hodnota nastavena na Vypnuto a hodnota proměnné DIMTOH je Ne. Při spuštění události *radioButtonZapnuto\_Click* nebo *radioButtonVypnuto\_Click* dojde ke změně obrázku v PictureBoxu, textu v Labelu pod obrázkem a změní se taktéž barva textu obou RadioButtonů tak, aby byl aktivní RadioButton barevně odlišen.

Tolik k popisu formuláře a jeho funkcionality pro nápovědu systémové proměnné DIMTOH. Jelikož se formuláře s nápovědami u dalších systémových proměnných, po stránce komponent

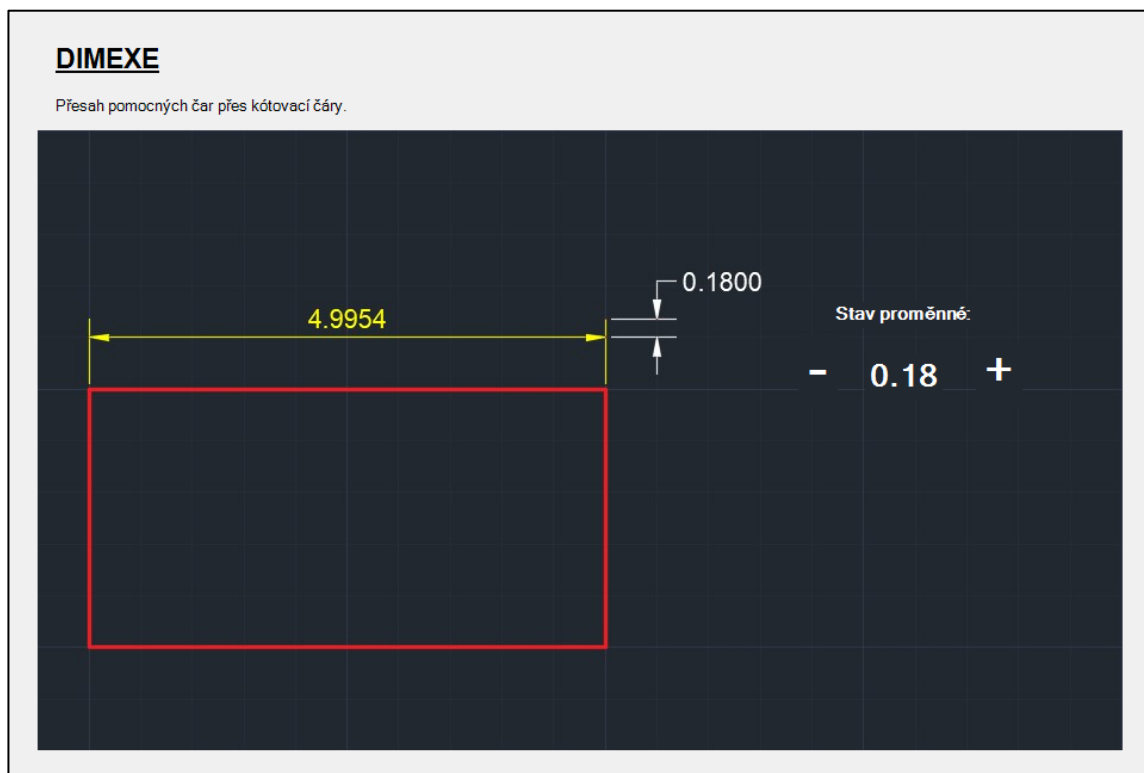
a funkcí, nijak výrazně neliší, usoudil jsem, že je zbytečné, tady takto podrobně popisovat zbývající nápovědy systémových proměnných. Nicméně zde vyjmenuju list všech použitých komponent. Jedná se o komponenty Label, PictureBox, RadioButton, Button, TextBox, ComboBox a CheckBox.

### 7.3 Formy nápovědy a jejich uživatelské preference

V rámci praktické části jsem vytvořil několik různých provedení nápovědy, abych ukázal, že neexistuje jenom jeden způsob provedení, a oproti textové nápovědě, nabízí symbolické jazyky mnohem větší možnosti při tvorbě nápovědy a může ji i značně usnadnit. O tom, jak bude finální podoba nápovědy vypadat, by poté měli rozhodnout uživatelské průzkumy, aby si sami uživatelé vybrali, jaké provedení se jim líbí nejvíce a je podle nich nejsnáze pochopitelné i pro neznalé uživatele v daném softwaru. Tento přístup jsem zvolil i já a provedl jsem menší uživatelský průzkum s mými připravenými nápovědami, které vidíte níže společně i s procentem uživatelů, kteří dané provedení preferují. Nápovědy jsou rozděleny do čtyř kategorií podle typu systémových proměnných.

#### 7.3.1 Systémové proměnné obsahující číselnou hodnotu

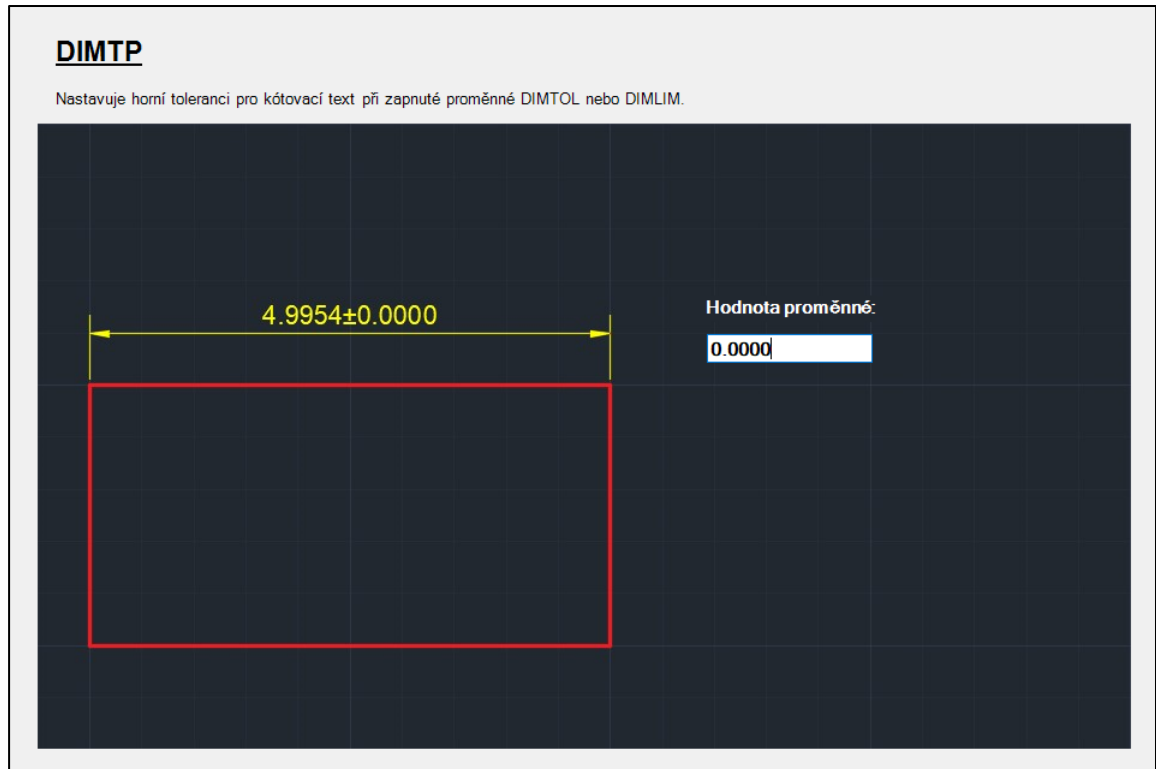
Jedná se o proměnné, do kterých se vkládá číselná hodnota. Pro tento typ systémových proměnných jsem vytvořil tři různé provedení nápovědy, kde každý způsob je prezentován na jiné systémové proměnné.



Obrázek 33: Ukázka nápovědy pro systémovou proměnnou DIMEXE (Zdroj vlastní)



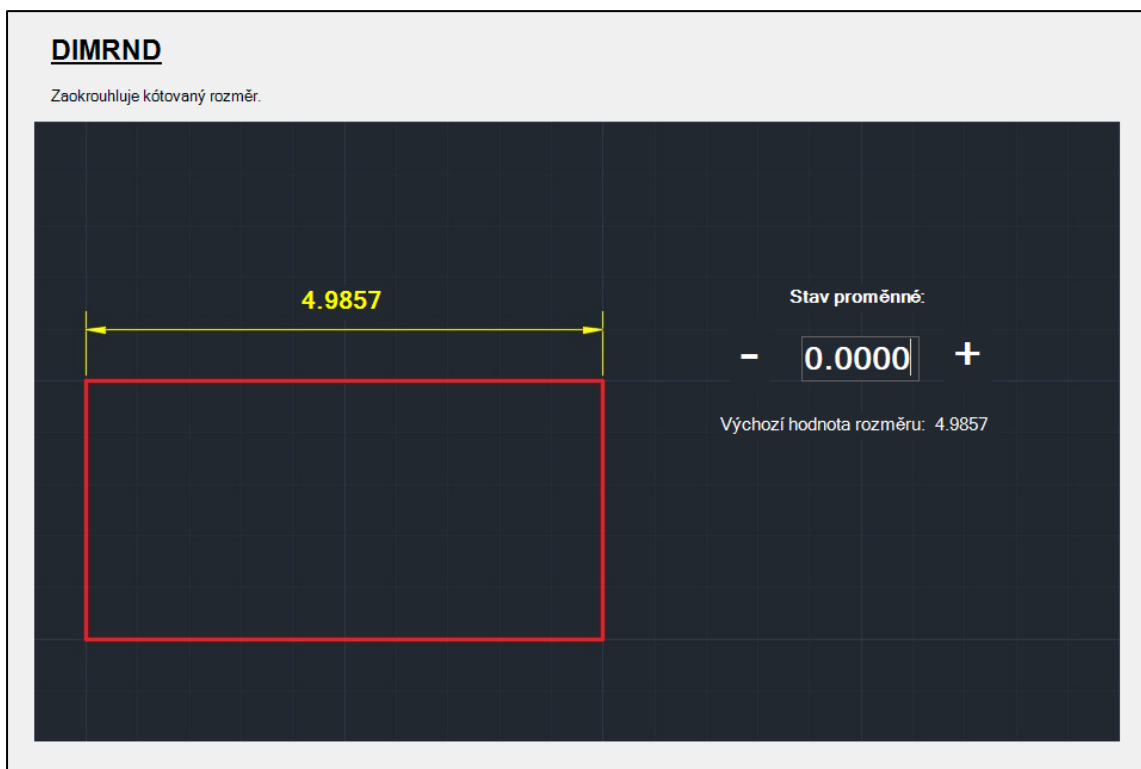
Na Obrázku 33 je nápověda pro systémovou proměnnou DIMEXE, která nastavuje přesah pomocných čar přes kótovací čáry. Zadání hodnoty se provádí pomocí symbolů – a +, kde je pevně nastavené číslo, o kolik se hodnota zvětšuje či snižuje. Tento způsob provedení preferuje 10,4 % uživatelů s IT zkušenostmi a 19,1 % uživatelů bez IT zkušeností.



Obrázek 34: Ukázka nápovědy pro systémovou proměnnou DIMTP (Zdroj vlastní)

Na Obrázku 34 je nápověda pro systémovou proměnnou DIMTP, která nastavuje horní toleranci pro kótovací text. Zadání hodnoty se provádí přes vstupní textového pole. Tento způsob provedení preferuje 24,1 % uživatelů s IT zkušenostmi a 23,8 % uživatelů bez IT zkušeností.

Na Obrázku 35 je nápověda pro systémovou proměnnou DIMRND, která zaokrouhluje kótovaný rozměr. Zadání hodnoty je možné jak přes vstupní textového pole, tak i pomocí symbolů – a +, kde je pevně nastavené číslo, o kolik se hodnota zvětšuje či snižuje. Tento způsob provedení preferuje 65,5 % uživatelů s IT zkušenostmi a 57,1 % uživatelů bez IT zkušeností.



Obrázek 35: Ukázka nápovědy pro systémovou proměnnou DIMRND (Zdroj vlastní)

### 7.3.2 Systémová proměnná pro nastavení pozice kótovacího textu

Jedná se o proměnné, která nastavuje pozici kótovacího textu. Pro tuto systémovou proměnnou jsem vytvořil dvě různé provedení nápovědy.



Obrázek 36: Ukázka nápovědy pro systémovou proměnnou DIMTAD (Zdroj vlastní)

Na Obrázku 36 je nápověda pro systémovou proměnnou DIMTAD. Výběr pozice se provádí přes nabídku ComboBox. Pod obrázkem je uvedeno, jaká hodnota se v AutoCADu pro tuto pozici do systémové proměnné ukládá. Tento způsob provedení preferuje 65,5 % uživatelů s IT zkušenostmi a 66,7 % uživatelů bez IT zkušeností.



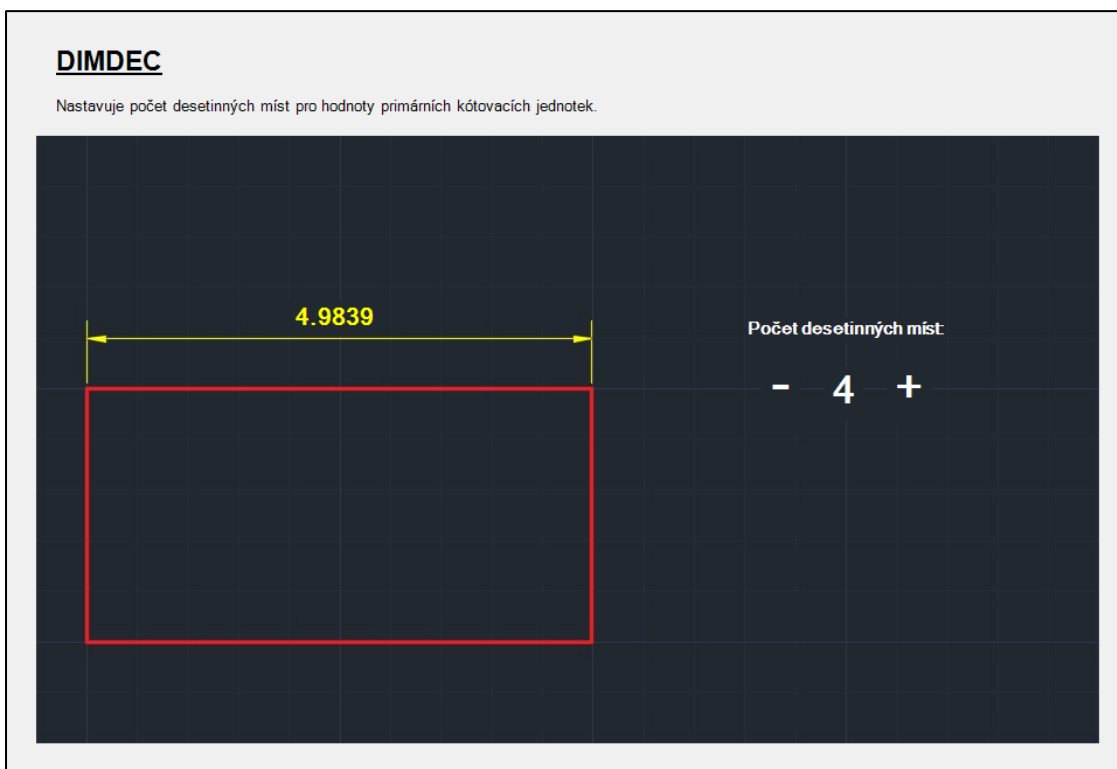
Obrázek 37: Ukázka nápovědy pro systémovou proměnnou DIMTAD (Zdroj vlastní)

Na Obrázku 37 je také nápověda pro systémovou proměnnou DIMTAD, tentokrát ale s jiným způsobem výběru pozice kótovacího textu. Výběr pozice se provádí přes kliknutí na text v obrázku, který se poté zvýrazní. Pod obrázkem je opět uvedeno, jaká hodnota se v AutoCADu pro tuto pozici do systémové proměnné ukládá. Tento způsob provedení preferuje 34,5 % uživatelů s IT zkušenostmi a 33,3 % uživatelů bez IT zkušeností.

### 7.3.3 Systémová proměnná pro nastavení počtu desetinných míst rozměrů

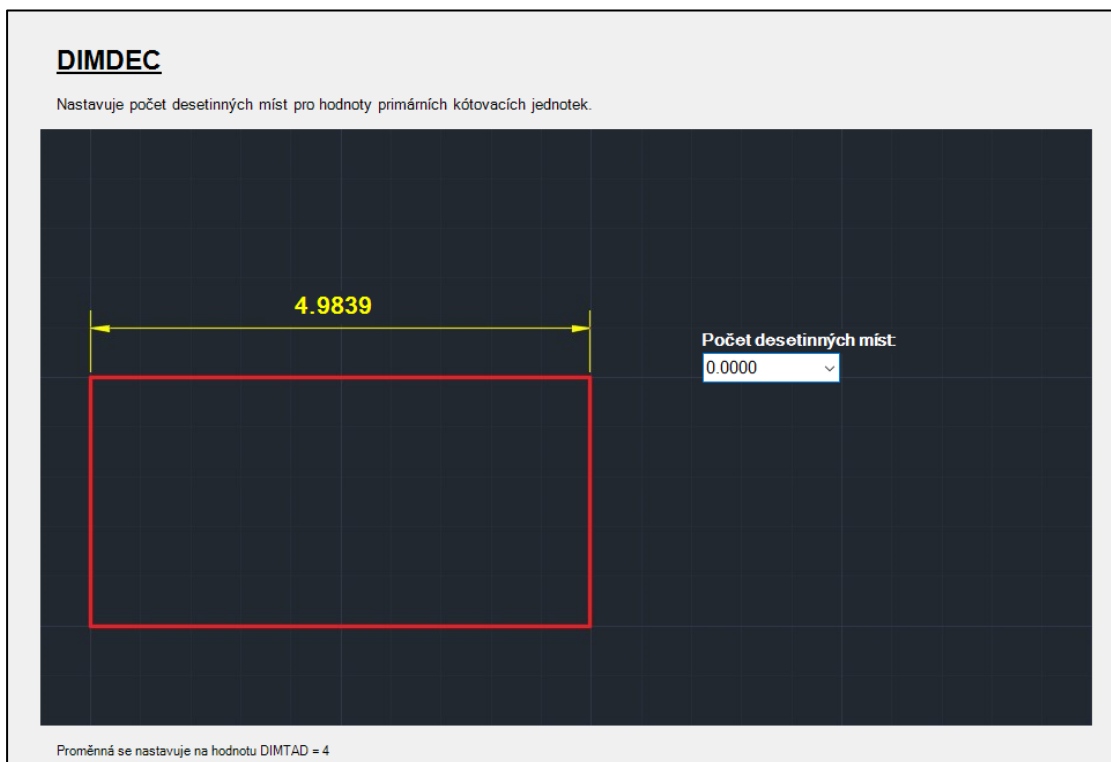
Jedná se o proměnnou, která nastavuje počet desetinných míst u kótovacích rozměrů. Pro tuto systémovou proměnnou jsem vytvořil dvě různé provedení nápovědy.

Na Obrázku 38 je nápověda pro systémovou proměnnou DIMDEC, která nastavuje počet desetinných míst u kótovacích rozměrů. Počet míst se vybírá přes symboly – a +. Tento způsob provedení preferuje 55,2 % uživatelů s IT zkušenostmi a 42,9 % uživatelů bez IT zkušeností.



Obrázek 38: Ukázka nápovědy pro systémovou proměnnou DIMDEC (Zdroj vlastní)

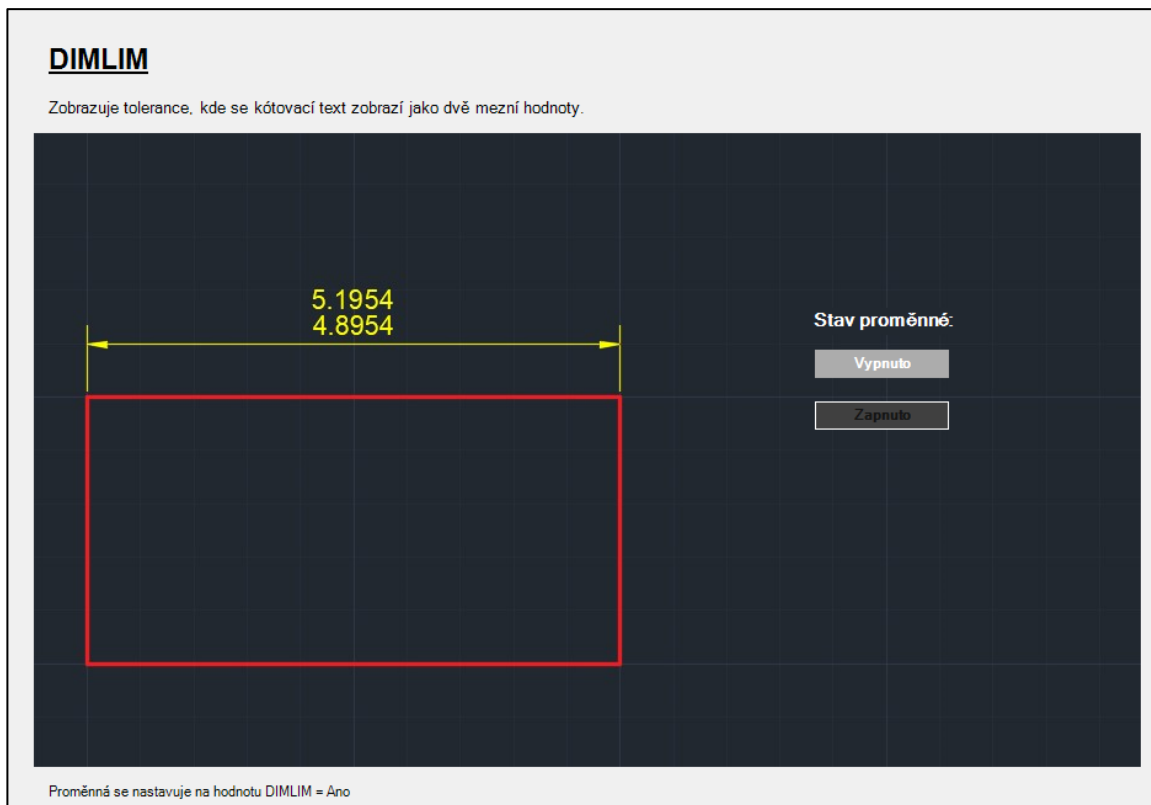
Na Obrázku 39 je také nápověda pro systémovou proměnnou DIMDEC, ale výběr počtu míst je proveden jiným způsobem. Výběr počtu míst se provádí přes nabídku ComboBox. Tento způsob provedení preferuje 44,8 % uživatelů s IT zkušenostmi a 57,1 % uživatelů bez IT zkušeností.



Obrázek 39: Ukázka nápovědy pro systémovou proměnnou DIMDEC (Zdroj vlastní)

### 7.3.4 Systémové proměnné nabývají stavu Zapnuto/Vypnuto

Jedná se o proměnné, které jsou buď zapnuté, nebo vypnuté. Pro tento typ systémových proměnných jsem vytvořil šest různých provedení nápovědy, kde každý způsob je prezentován na jiné proměnné.



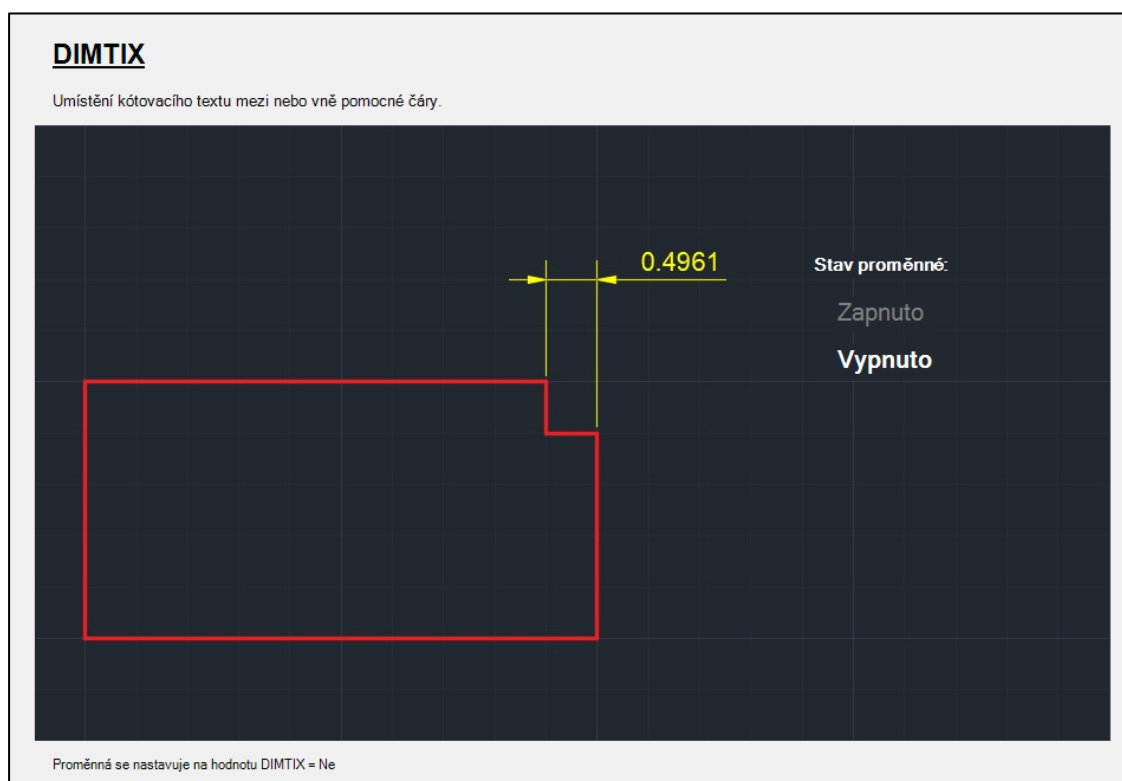
Obrázek 40: Ukázka nápovědy pro systémovou proměnnou DIMLIM (Zdroj vlastní)

Na Obrázku 40 je nápověda pro systémovou proměnnou DIMLIM, která určuje, jestli se budou zobrazovat tolerance kótovacího rozměru či nikoliv. Stav se nastavuje pomocí tlačítek Zapnuto/Vypnuto, kde aktivní tlačítko ztmavne a už na něj nelze znova kliknout. Pod obrázkem je uvedeno, co se při zvoleném stavu v AutoCADu vloží do systémové proměnné. Tento způsob provedení preferuje 0 % uživatelů s IT zkušenostmi a 14,3 % uživatelů bez IT zkušeností.



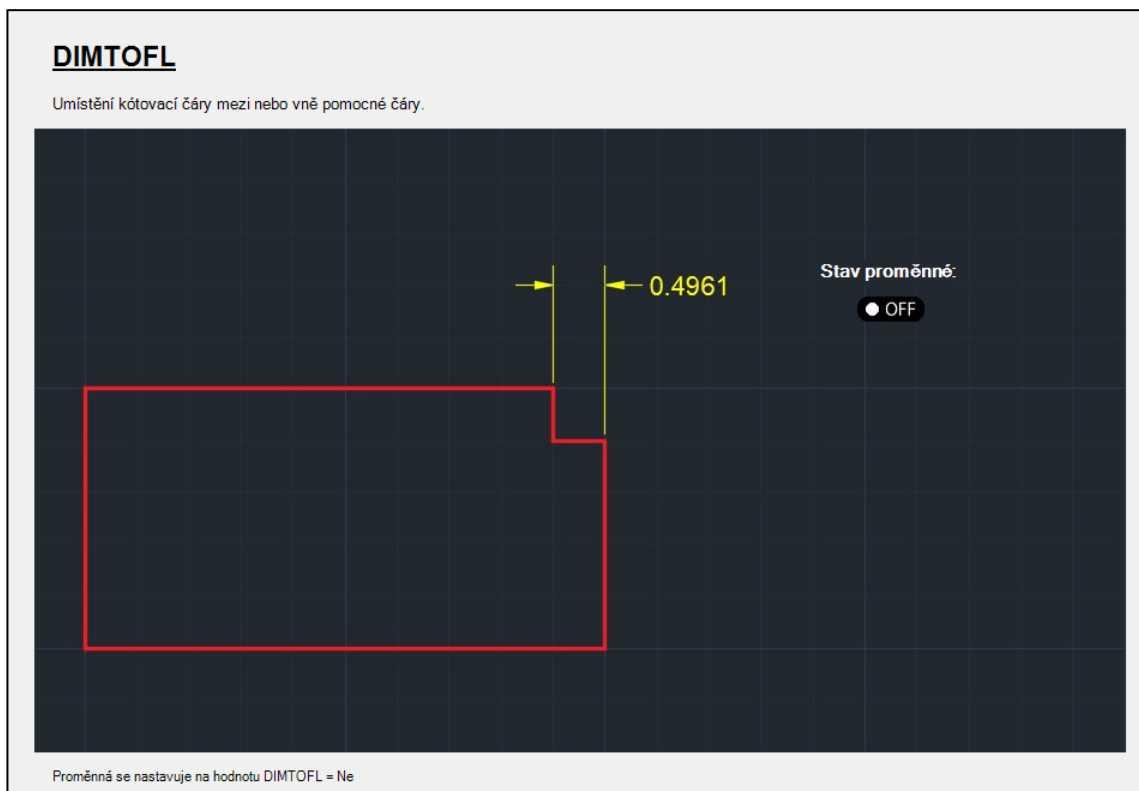
Obrázek 41: Ukázka nápovědy pro systémovou proměnnou DIMTIH (Zdroj vlastní)

Na Obrázku 41 je nápověda pro systémovou proměnnou DIMTIH, která nastavuje orientaci kótovacího textu umístěného mezi kótovacími čarami u lineárního kótování. Stav se nastavuje pomocí CheckBoxu. Pod obrázkem je uvedeno, co se při zvoleném stavu v AutoCADu vloží do systémové proměnné. Tento způsob provedení preferuje 24,1 % uživatelů s IT zkušenostmi a 9,5 % uživatelů bez IT zkušeností.



Obrázek 42: Ukázka nápovědy pro systémovou proměnnou DIMTIX (Zdroj vlastní)

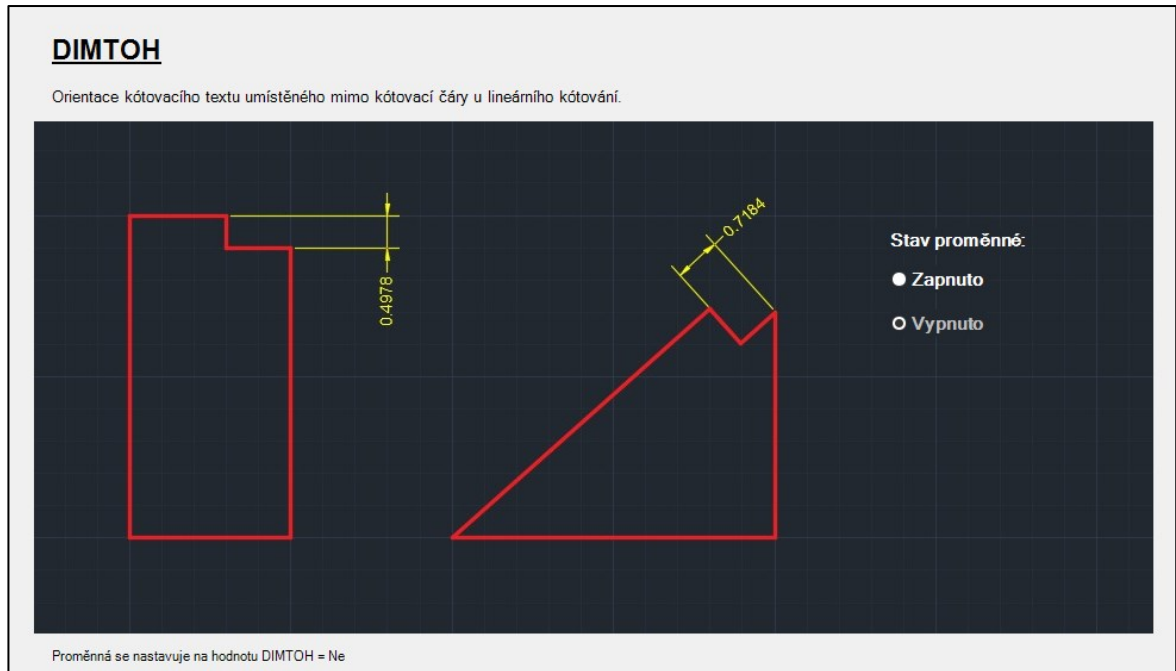
Na Obrázku 42 je nápověda pro systémovou proměnnou DIMTIX, která nastavuje umístění kótovacího textu buď mezi, nebo vně pomocné čáry. Stav se nastavuje pomocí klikání na dva texty Zapnuto a Vypnuto, kde zvolený text se zvětší a změní barvu. Pod obrázkem je uvedeno, co se při zvoleném stavu v AutoCADu vloží do systémové proměnné. Tento způsob provedení preferuje 10,4 % uživatelů s IT zkušenostmi a 28,6 % uživatelů bez IT zkušeností.



Obrázek 43: Ukázka nápovědy pro systémovou proměnnou DIMTOFL (Zdroj vlastní)

Na Obrázku 43 je nápověda pro systémovou proměnnou DIMTOFL, která nastavuje umístění kótovací čáry buď mezi, nebo vně pomocné čáry. Stav se nastavuje pomocí ikony ON/OFF. Ikona, která se nazývá Toggle switch, je zde vytvořena pomocí dvou obrázků, jeden pro každý stav. Je to z důvodu, že Toggle switch není standardní komponentou ve Visual Studiu 2019. Toggle switch lze také vytvořit pomocí třídy *ToggleSwitch* nebo si ho vytvořit jako vlastní komponentu s využitím vlastností komponenty *CheckBox*. Pod obrázkem je uvedeno, co se při zvoleném stavu v AutoCADu vloží do systémové proměnné. Tento způsob provedení preferuje 24,1 % uživatelů s IT zkušenostmi a 14,3 % uživatelů bez IT zkušeností.

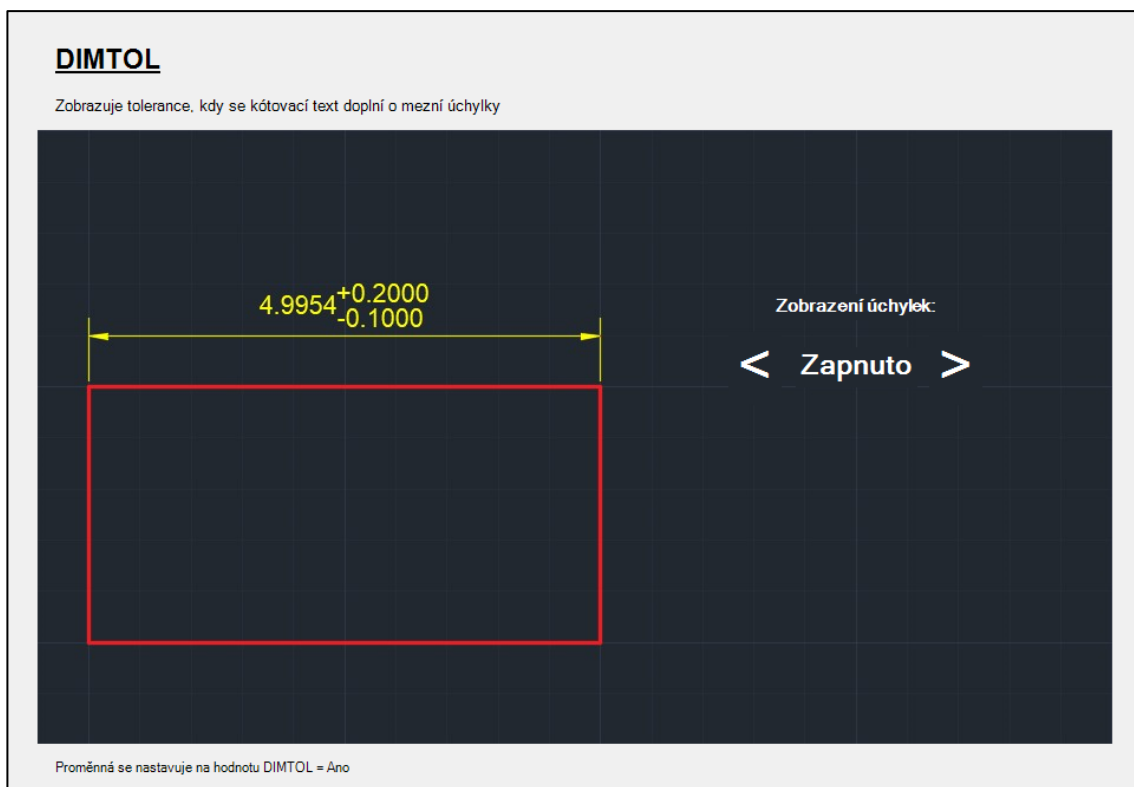
Na Obrázku 44 je nápověda pro systémovou proměnnou DIMTOH, která nastavuje orientaci kótovacího textu umístěného mimo kótovací čáry u lineárního kótování. Stav se nastavuje pomocí RadioButtonu. Pod obrázkem je uvedeno, co se při zvoleném stavu v AutoCADu vloží do systémové proměnné. Tento způsob provedení preferuje 31 % uživatelů s IT zkušenostmi a 23,8 % uživatelů bez IT zkušeností.



Obrázek 44: Ukázka nápovědy pro systémovou proměnnou DIMTOH (Zdroj vlastní)

Na Obrázku 45 je nápověda pro systémovou proměnnou DIMTOL, která určuje, jestli se vedle kótovacího textu budou zobrazovat i mezní úchytky či nikoliv. Stav se nastavuje pomocí symbolů < a >. Pod obrázkem je uvedeno, co se při zvoleném stavu v AutoCADu vloží do systémové proměnné. Tento způsob provedení preferuje 10,4 % uživatelů s IT zkušenostmi a 9,5 % uživatelů bez IT zkušeností.





Obrázek 45: Ukázka nápovědy pro systémovou proměnnou DIMTOL (Zdroj vlastní)

Celkový počet uživatelů zapojených do průzkumů je 50, z toho 29 uživatelů má IT zkušenosti a 21 uživatelů nemá IT zkušenosti. V příloze naleznete grafické znázornění výsledků uživatelského průzkumu.

#### 7.4 Shrnutí uživatelského průzkumu

Mého uživatelského průzkumu se zúčastnilo 50 lidí z různých povolání a ze širokého věkového rozmezí a byly rozděleny do dvou skupin, lidé se zkušenostmi v IT a lidé bez zkušeností v IT. Skupina se zkušenostmi v IT čítala 29 lidí a skupina bez zkušeností v IT měla 21 lidí, takže početně byly skupiny téměř vyrovnané. Tyto skupiny si v každé kategorii nápověd vybraly, jaké provedení nápovědy se jim v rámci oné kategorie nejvíce líbilo.

Ve dvou ze čtyř kategorií nápověd měly obě skupiny nejvíce hlasů u stejného provedení nápovědy. V kategorii *Systémová proměnná pro nastavení pozice kótovacího textu* se jedná o provedení s ComboBoxem (Obrázek 36, 65,5 % hlasů s IT zk. a 66,7 % hlasů bez IT zk.) a v kategorii *Systémové proměnné obsahující číselnou hodnotu* šlo o provedení s možností nastavení hodnoty jak zadáním z klávesnice, tak pomocí myši (Obrázek 35, 65,5 % hlasů s IT zk. a 57,1 % hlasů bez IT zk.).

U zbylých dvou kategorií se už ale uživatelské volby rozcházejí. U kategorie *Systémová proměnná pro nastavení počtu desetinných míst rozměrů* mělo u skupiny s IT zkušenostmi

nejvíce hlasů provedení nápovědy se symboly + a – (Obrázek 38, 55,2 % hlasů), zatímco skupina bez IT zkušeností preferovala provedení s ComboBoxem (Obrázek 39, 57,1 % hlasů).

Stejně tak to je u kategorie *Systémové proměnné nabývají stavu Zapnuto/Vypnuto*, kde si skupina s IT zkušenostmi zvolila provedení nápovědy s RadioButtonem (Obrázek 44, 31 % hlasů) a skupina bez IT zkušeností si vybrala provedení s textem Zapnuto/Vypnuto (Obrázek 42, 28,6 % hlasů).

Z výsledků je patrné, že zkušenosti v IT měly při výběru provedení nápovědy jistý vliv. Tito lidé si spíše budou vybírat provedení, které využívá prvky, se kterými se již setkali, jsou na ně zvyklí nebo s nimi přímo sami pracují. Rozdíly hlasů u jednotlivých provedení nápovědy ovšem nejsou nijak velké a je možné, že při provedení většího průzkumu, by preference provedení byly napříč skupinami více vyrovnané.

## ZÁVĚR

Cílem této práce bylo pomocí interaktivního symbolického jazyka vytvořit nápovědu pro vybrané systémové proměnné v AutoCADu. Pro její realizaci je potřeba mít lehký přehled o symbolických jazycích a znalost alespoň jednoho způsobu programování v AutoCADu. Při práci jsem využíval programovací jazyk C#.

V teoretické části se mi podařilo popsat symbolické jazyky, konkrétně obrázky, symboly, piktogramy a animace. Pozornost je věnována především definici každého symbolického jazyka a jeho charakteristikám, současnému použití, historii a rozdělení na podtypy. Kromě toho se mi podařilo i vysvětlit, co je AutoCad, a také nastínit, jaké existují způsoby pro programování v AutoCADu i s jejich krátkým popisem.

V praktické části bylo dosaženo hlavního cíle, a to vytvořit nápovědu pro vybrané systémové proměnné v AutoCADu za pomoci interaktivního symbolického jazyka. Součástí této části je i podrobný popis implementace a struktury vytvořené nápovědy. Jelikož jsem nápovědu vytvořil v několika různých provedení, obsahuje tato část i uživatelský průzkum, kde spolu s onou ukázkou provedení nápovědy je i uvedeno, kolik procent dotázaných uživatelů si dané provedení zvolilo. V závěru části jsou všechny uživatelské odpovědi shrnuty a porovnány.

Při zpracování praktické části jsem ovšem narazil na problém, a tím je nežádoucí změna rozměrů u některých grafických komponent v případě, že je nápověda zobrazena na obrazovce s jiným měřítkem zobrazení. Tento problém se týká především komponent, které jsou vloženy v jiné komponentě, a ne přímo v hlavním formuláři. Nápověda se zobrazuje správně při 100 % měřítku zobrazení. Pokud je ale nápověda otevřena na obrazovce s jiným měřítkem zobrazení, dojde u některých komponent k samovolné změně rozměrů a komponenta je buď větší, nebo menší a narušuje vizuální stránku nápovědy. Tento problém se mi bohužel nepodařilo vyřešit a lze mu předejít jen tak, že si uživatel zkontroluje, zda má měřítko zobrazení nastaveno na 100 %.

Do budoucna je možné v práci pokračovat a rozšířit ji i na další věci než jen systémové proměnné. Další možností je zlepšení grafické stránky nápovědy anebo úplné vynechání textu tak, aby nápověda byla tvořena čistě symbolickými jazyky. Věřím, že tento způsob provedení nápovědy, to jest za použití symbolických jazyků, by jistě našel své uplatnění i hranicemi AutoCADu, a to v jiných softwarech.

Při zpracování této práce jsem získal základní zkušenosti s prací a ovládáním AutoCADu a značně jsem si rozšířil znalosti v oblasti symbolických jazyků, jejich tvorby a další problematiky s tím spojenou.

## POUŽITÁ LITERATURA

- [1] ZOLA, Andrew. Image. WhatIs.com [online]. Newton: TechTarget, c1999 - 2023, 12 Jul 2022 [cit. 2023-04-06]. Dostupné z: <https://www.techtarget.com/whatis/definition/image>
- [2] Picture. Computer Hope [online]. Utah: Computer Hope, c2023 [cit. 2023-04-06]. Dostupné z: <https://www.computerhope.com/jargon/p/picture.htm>
- [3] Obrazové soubory. Adobe [online]. San Jose: Adobe, c2023 [cit. 2023-04-06]. Dostupné z: <https://www.adobe.com/cz/creativecloud/file-types/image.html>
- [4] Rastrová vs. vektorová grafika. Adobe [online]. San Jose: Adobe, c2023 [cit. 2023-04-06]. Dostupné z: <https://www.adobe.com/cz/creativecloud/file-types/image/comparison/raster-vs-vector.html>
- [5] Bitmapová grafika. WikiSofia [online]. Praha: WikiSofia, c2013 [cit. 2023-04-06]. Dostupné z: [https://wikisofia.cz/wiki/Bitmapov%C3%A1\\_grafika](https://wikisofia.cz/wiki/Bitmapov%C3%A1_grafika)
- [6] ŠTRÁFELDA, Jan. Bitmapová grafika. Štráfelda.cz [online]. [cit. 2023-04-06]. Dostupné z: <https://www.strafelda.cz/bitmapova-grafika>
- [7] MUHAMMADUMAR. Raster VS Vector. In: FileFormat [online]. Lane Cove: Aspose, c2001-2023, August 25, 2021 [cit. 2023-04-12]. Dostupné z: <https://blog.fileformat.com/2021/08/25/raster-vs-vector-images-a-brief-comparison/>
- [8] ŠTRÁFELDA, Jan. Vektorová grafika. Štráfelda.cz [online]. [cit. 2023-04-06]. Dostupné z: <https://www.strafelda.cz/vektorova-grafika>
- [9] Vektorová grafika. WikiSofia [online]. Praha: WikiSofia, c2013 [cit. 2023-04-06]. Dostupné z: [https://wikisofia.cz/wiki/Vektorov%C3%A1\\_grafika](https://wikisofia.cz/wiki/Vektorov%C3%A1_grafika)
- [10] Soubory RAW. Adobe [online]. San Jose: Adobe, c2023 [cit. 2023-04-06]. Dostupné z: <https://www.adobe.com/cz/creativecloud/file-types/image/raw.html>
- [11] Need to Open a RAW File?. AfterShot Pro [online]. Ottawa: Corel Corporation, c2023 [cit. 2023-04-06]. Dostupné z: <https://www.aftershotpro.com/en/pages/raw-file/>
- [12] MOODIE, Michael. Pros & Cons of Shooting in RAW Image Format: What you Need to Know Today. Sleeklens [online]. Sheridan: Sleeklens, c2023, May 5, 2022 [cit. 2023-04-06]. Dostupné z: <https://sleeklens.com/blog/photography-tips-and-tricks/pros-cons-of-shooting-in-raw/>
- [13] MARK, Trevor. Difference between Sign and Symbol. Difference Between.net [online]. Difference Between, c2023, 1 November 2019 [cit. 2023-04-07]. Dostupné z: <http://www.differencebetween.net/miscellaneous/difference-between-sign-and-symbol/>
- [14] Writing 101: What Is Symbolism? Symbolism Definition and Examples in Literature. MasterClass [online]. San Francisco: MasterClass, c2023, Aug 30, 2021 [cit. 2023-04-07]. Dostupné z: <https://www.masterclass.com/articles/writing-101-what-is-symbolism-symbolism-definition-and-examples-in-literature>

- [15] SYMBOLS in visual communication. Julianna Kunstler [online]. Kunstler, c2008-2023 [cit. 2023-04-07]. Dostupné z: [https://juliannakunstler.com/vislit\\_symbols.html](https://juliannakunstler.com/vislit_symbols.html)
- [16] NUGENT, Nicole. Typogram. In: Behance [online]. November 2nd 2012 [cit. 2023-04-07]. Dostupné z: <https://www.behance.net/gallery/5767643/Typogram>
- [17] What is a Phonogram? - Definition & Examples. Study.com [online]. Study.com, c2003-2023, 9 July 2027 [cit. 2023-04-07]. Dostupné z: <https://study.com/academy/lesson/what-is-a-phonogram-definition-examples.html>
- [18] SANSERI, Wanda. The 70 Basic Phonogram Cards. In: The Crown College [online]. Powell: Crown Bookshop, c2023, 1 January 1994 [cit. 2023-04-11]. Dostupné z: <https://bookshop.thecrowncollege.edu/70-basic-phonogram-cards.html>
- [19] IBM rebus. In: Alex Porcelli [online]. ALEX PORCELLI, c2016, Jul 29, 2022 [cit. 2023-04-11]. Dostupné z: <https://porcelli.me/announcement/2022/07/29/new-journey-begins.html>
- [20] GLASER, Milton. I Love New York. In: BEACH [online]. BEACH, c2023, April 13, 2015 [cit. 2023-04-11]. Dostupné z: <https://beachpackagingdesign.com/boxvox/10-rebus-logos-r-they-bad-4-the>
- [21] NORDQUIST, Richard. What Is an Ideogram? Definition and Examples. ThoughtCo [online]. New York: ThoughtCo, August 01, 2019 [cit. 2023-04-07]. Dostupné z: <https://www.thoughtco.com/what-is-an-ideogram-1691050>
- [22] HALDEMANN, Jerome. A brief history of pictograms and ideograms. Saffron Interactive [online]. London: Saffron Interactive, November 11 2014 [cit. 2023-04-07]. Dostupné z: <https://saffroninteractive.com/a-brief-history-of-pictograms-and-ideograms/>
- [23] WATSON, James. Modern Ideograms. In: GDES3010 – Major Project [online]. GDES3010 – Major Project, October 10, 2017 [cit. 2023-04-11]. Dostupné z: <https://jameswatsongdes3010.wordpress.com/2017/10/10/history-ideographs-ideograms/>
- [24] TERRELL HANNA, Katie. Icon. WhatIs.com [online]. Newton: TechTarget, c1999 - 2023, 10 Jun 2022 [cit. 2023-04-07]. Dostupné z: <https://www.techtarget.com/whatis/definition/icon>
- [25] BOYKO, Ivan. Search icon. In: Iconfinder [online]. Virum: Iconfinder, c2023 [cit. 2023-04-11]. Dostupné z: [https://www.iconfinder.com/icons/309077/search\\_icon](https://www.iconfinder.com/icons/309077/search_icon)
- [26] Printer icon. In: Flaticon [online]. Málaga: Freepik Company, c2010-2023 [cit. 2023-04-11]. Dostupné z: [https://www.flaticon.com/free-icon/printer\\_4020167](https://www.flaticon.com/free-icon/printer_4020167)
- [27] What is a logo?. 99designs [online]. 99designs [cit. 2023-04-08]. Dostupné z: <https://99designs.com/blog/tips/what-is-a-logo/>
- [28] Wordmark logos. In: Looka [online]. Looka, c2023, Jul 19, 2022 [cit. 2023-04-12]. Dostupné z: <https://looka.com/blog/wordmark-logo-design/>
- [29] How to choose from 4 types of company logo designs. Deluxe [online]. Minneapolis: Deluxe Corporation, c2023, Feb. 09, 2018 [cit. 2023-04-08]. Dostupné z: <https://www.deluxe.com/blog/how-to-choose-from-four-types-company-logo-designs/>

- [30] Lettermarks. In: Jessica Jones [online]. Jessica Jones, c2011-2023, August 9, 2017 [cit. 2023-04-12]. Dostupné z: <https://www.jessicajonesdesign.com/types-of-logos/>
- [31] MORR, Kelly. The 7 types of logos (and how to use them). 99designs [online]. 99designs [cit. 2023-04-08]. Dostupné z: <https://99designs.com/blog/tips/types-of-logos/>
- [32] Brandmark. In: Vreasy [online]. Vreasy, c2023 [cit. 2023-04-12]. Dostupné z: <https://web.vreasy.com/logo-vacation-rental-brand/>
- [33] BARBER, Craig. Combination logos. In: Logobly [online]. Cork: DesignBro, c2023 [cit. 2023-04-12]. Dostupné z: <https://logobly.com/blog/logo-design/?ref=marketsplash.com>
- [34] The Emblem. In: Sync'd Design [online]. Huddersfield: Sync'd Design, 12 October 2021 [cit. 2023-04-12]. Dostupné z: <https://www.syncddesign.co.uk/blog/2021-10-12-the-7-types-of-logos>
- [35] Character and mascots. In: The Logo Community [online]. The Logo Community, c2023, March 10, 2020 [cit. 2023-04-12]. Dostupné z: [https://www.thelogocommunity.com/what-is-a-logo-design/?doing\\_wp\\_cron=1674509529.1812889575958251953125%20/](https://www.thelogocommunity.com/what-is-a-logo-design/?doing_wp_cron=1674509529.1812889575958251953125%20/)
- [36] Religious symbol. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 14 March 2023 [cit. 2023-04-08]. Dostupné z: [https://en.wikipedia.org/wiki/Religious\\_symbol](https://en.wikipedia.org/wiki/Religious_symbol)
- [37] GOLDAMMER, Kurt Moritz Artur. Religious symbolism and iconography. Britannica [online]. Chicago: Encyclopædia Britannica, c2023, 12 Nov. 2020 [cit. 2023-04-08]. Dostupné z: <https://www.britannica.com/topic/religious-symbolism>
- [38] Architectural Symbols | Architectural Symbols in the Drawing. CivilMint [online]. CivilMint.Com, c2023 [cit. 2023-04-08]. Dostupné z: <https://civilmint.com/architectural-symbols/>
- [39] KRAMER, Lindsay. What Is Symbolism? Examples of Symbolism as a Literary Device. Grammarly [online]. San Francisco: Grammarly, c2023, June 6, 2022 [cit. 2023-04-08]. Dostupné z: <https://www.grammarly.com/blog/symbolism/>
- [40] GLATCH, Sean. SYMBOLISM IN LITERATURE: WHAT SYMBOLISM IS, AND HOW TO USE IT IN YOUR WRITING. Writers.com [online]. Writers.com, c2023, July 12, 2022 [cit. 2023-04-08]. Dostupné z: <https://writers.com/what-is-symbolism-in-literature>
- [41] MYTHS IN WORDS AND PICTURES [online]. [cit. 2023-04-08]. Dostupné z: <https://smithsonianeducation.org/idealabs/myths/symbolsinart/index.html>
- [42] ROSHANDA. Symbols In Art & Their Meaning. Art by Ro [online]. Lambertville: Art by Ro, c2023 [cit. 2023-04-08]. Dostupné z: <https://artbyro.com/symbols-in-art/>
- [43] WIKIPEDIA. Symbol. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation [cit. 2023-04-07]. Dostupné z: <https://en.wikipedia.org/wiki/Symbol>

- [44] Traffic Signs. Motor Vehicles Department: Government of Kerala [online]. Motor Vehicles Department, c2023 [cit. 2023-04-08]. Dostupné z: <https://mvd.kerala.gov.in/en/traffic-signs>
- [45] TSORME, Derrick. The Difference Between Signs And Symbols. Medium [online]. Medium, May 30, 2020 [cit. 2023-04-07]. Dostupné z: <https://uxderrick.medium.com/the-difference-between-signs-and-symbols-12b645d64718>
- [46] SMITH, Emma. Difference Between Sign and Symbol. Ask Any Difference [online]. Punjab: Ask Any Difference, c2023, March 29 2023 [cit. 2023-04-07]. Dostupné z: <https://askanydifference.com/difference-between-sign-and-symbol/>
- [47] TECHTARGET CONTRIBUTOR. Semiotics. Whatis.com [online]. Newton: TechTarget, c1999-2023, 22 Jul 2017 [cit. 2023-04-06]. Dostupné z: <https://www.techtarget.com/whatis/definition/semiotics>
- [48] BRITANNICA, The Editors of Encyclopaedia. Semiotics. Britannica [online]. Chicago: Encyclopædia Britannica, c2023, Feb 24, 2023 [cit. 2023-04-06]. Dostupné z: <https://www.britannica.com/science/semiotics>
- [49] Zvuk a video. Stránky k výuce informatiky [online]. [cit. 2023-04-08]. Dostupné z: <http://www.ivt.mzf.cz/seminar/11-animace/>
- [50] KRÁLOVÁ, Magda. SETRVAČNOST OKA. Techmania Science Center [online]. Plzeň: Techmania Science Center, c2007 [cit. 2023-04-08]. Dostupné z: <http://edu.techmania.cz/cs/encyklopedie/fyzika/svetlo/lidske-oko/setrvacnost-oka>
- [51] MADISON, Nicole. What is Animation?. MusicalExpert.org [online]. San Francisco: Musical Expert, c2003-2023, March 15, 2023 [cit. 2023-04-08]. Dostupné z: <https://www.musicaexpert.org/what-is-animation.htm>
- [52] The Uses of Animation. In: TAN, Jiang. Aspects of Animation : Steps of Learn Animated Cartoons. Serial Publishers, 2016, s. 7-8. ISBN 978-8183875905.
- [53] MADISON, Nicole. What is Animation?. MusicalExpert.org [online]. San Francisco: Musical Expert, c2003-2023, March 15, 2023 [cit. 2023-04-08]. Dostupné z: <https://www.musicaexpert.org/what-is-animation.htm>
- [54] MAIO, Alyssa. What is Animation? Definition and Types of Animation. Studiobinder [online]. Santa Monica: STUDIOBINDER, c2023, November 18, 2020 [cit. 2023-04-08]. Dostupné z: <https://www.studiobinder.com/blog/what-is-animation-definition/>
- [55] Flynn Rider/Eugene Fitzherbert Sketch. In: Baianat [online]. Baianat, c2023 [cit. 2023-04-12]. Dostupné z: <https://www.baianat.com/books/animation-revolution/traditional-animation>
- [56] WHAT IS COMPUTER ANIMATION?. Unity [online]. San Francisco: Unity Technologies, c2023 [cit. 2023-04-10]. Dostupné z: <https://unity.com/solutions/what-is-computer-animation>
- [57] Complete guide to computer animation: history, techniques and examples. Adobe [online]. San Jose: Adobe, c2023 [cit. 2023-04-09]. Dostupné z: <https://www.adobe.com/uk/creativecloud/animation/discover/computer-animation.html>

- [58] F. O'BRIEN, James, Robert E. BODENHEIMER JR. a Jessica K. HODGINS. Computer Animation. Encyclopedia of Computer Science. Thomson Press, 2001, 0-12. Dostupné z: doi:10.1002/047134608X.W7507
- [59] PENNINGTON, Adrian. Andy Serkis in War for the Planet of the Apes. In: Backstage [online]. Backstage, c2023, November 19, 2018 [cit. 2023-04-12]. Dostupné z: <https://www.backstage.com/uk/magazine/article/trends-and-intelligence-motion-capture-performance-66001/>
- [60] BABADI, Amin. What Is Physically-Based Animation?. Towards Data Science [online]. Towards Data Science, Nov 19, 2018 [cit. 2023-04-09]. Dostupné z: <https://towardsdatascience.com/what-is-physically-based-animation-cd92a7f8d6a4>
- [61] Computer Animation. TutorialsPoint [online]. Haidarábád: TutorialsPoint, c2023 [cit. 2023-04-10]. Dostupné z: [https://www.tutorialspoint.com/computer\\_graphics/computer\\_animation.htm](https://www.tutorialspoint.com/computer_graphics/computer_animation.htm)
- [62] What is morphing? A complete guide to morphing animation. Adobe [online]. San Jose: Adobe, c2023 [cit. 2023-04-09]. Dostupné z: <https://www.adobe.com/creativecloud/animation/discover/morphing-in-animation.html>
- [63] WHAT IS 3D ANIMATION?. Unity [online]. San Francisco: Unity Technologies, c2023 [cit. 2023-04-10]. Dostupné z: <https://unity.com/solutions/what-is-3d-animation>
- [64] TYLER, Dustin. Examples of 2D and 3D Animation. GameDesigning [online]. St. Petersburg, Florida, c2023, March 19, 2023 [cit. 2023-04-16]. Dostupné z: <https://www.gamedesigning.org/animation/2d-and-3d-animation/>
- [65] KEDAJ, Zdeněk. Stop-motion animační techniky. Masarykova Univerzita Fakulta Informatiky, 2009. Dostupné také z: [https://www.fi.muni.cz/lemma/referaty/09/Kedaj\\_Zdenek-stopMotionTechniky.pdf](https://www.fi.muni.cz/lemma/referaty/09/Kedaj_Zdenek-stopMotionTechniky.pdf)
- [66] The Uses of Animation. In: TAN, Jiang. Aspects of Animation : Steps of Learn Animated Cartoons. Serial Publishers, 2016, s. 9-12. ISBN 978-8183875905.
- [67] Stop motion animation explained: definition, types and techniques. Adobe [online]. San Jose: Adobe, c2023 [cit. 2023-04-08]. Dostupné z: <https://www.adobe.com/creativecloud/animation/discover/stop-motion-animation.html>
- [68] Introduction. In: CROOK, Ian a Peter BEARE. Motion Graphics: Principles and Practices from the Ground Up. Bloomsbury Publishing, 2016, s. 10-12. ISBN 9781472569004.
- [69] Motion graphics explained: definition, history and examples. Adobe [online]. San Jose: Adobe, c2023 [cit. 2023-04-08]. Dostupné z: <https://www.adobe.com/uk/creativecloud/animation/discover/motion-graphics.html#01>
- [70] What is motion graphics?. Biteable [online]. Biteable, c2022, 17 Sep 2017 [cit. 2023-04-08]. Dostupné z: <https://biteable.com/blog/what-is-motion-graphics/>
- [71] The Uses of Animation. In: TAN, Jiang. Aspects of Animation : Steps of Learn Animated Cartoons. Serial Publishers, 2016, s. 4-7. ISBN 978-8183875905.



- [72] HISTORY OF ANIMATION. Odessa Animation Studio [online]. Odessa: Odessa Animation Studio, c1991-2021 [cit. 2023-04-09]. Dostupné z: <http://animation-ua.com/en/school-animation/history-of-animation/178-history-of-animation>
- [73] The History Of Animation [online]. [cit. 2023-04-09]. Dostupné z: <https://history-of-animation.webflow.io/>
- [74] A Guide to the History of Animation. MasterClass [online]. San Francisco: MasterClass, c2023, Aug 7, 2021 [cit. 2023-04-09]. Dostupné z: <https://www.masterclass.com/articles/a-guide-to-the-history-of-animation>
- [75] Magic lantern, 1671. In: Science Photo Library [online]. Londýn: Science Photo Library, c2023 [cit. 2023-04-12]. Dostupné z: <https://www.sciencephoto.com/media/979665/view/magic-lantern-1671>
- [76] Thaumatrope Action Black and White. In: Twinkl [online]. Sheffield: Twinkl [cit. 2023-04-12]. Dostupné z: <https://www.twinkl.es/illustration/thaumatrope-action-black-and-white>
- [77] Phenakistoscope disk. In: Sciphile.org [online]. 8 Jun 2013 [cit. 2023-04-12]. Dostupné z: <https://sciphile.org/lessons/zoetropes-phenakistoscopes-and-other-animation-toys>
- [78] KEHR, Dave. Animation. Britannica [online]. Chicago: Encyclopædia Britannica, c2023, Feb 12, 2023 [cit. 2023-04-09]. Dostupné z: <https://www.britannica.com/art/animation>
- [79] DUNN, Andrew. Zoetrope. In: Wikimedia Commons [online]. San Francisco: Wikimedia Foundation, 5 November 2004 [cit. 2023-04-12]. Dostupné z: <https://commons.wikimedia.org/wiki/File:Zoetrope.jpg>
- [80] Cat's Proposal. In: Flipboku [online]. Palma: Flipboku Shop, c2023 [cit. 2023-04-12]. Dostupné z: <https://flipboku.com/products/cat-s-proposal>
- [81] Classic Brown Mini Praxinoscope. In: My Museum Shop [online]. [cit. 2023-04-12]. Dostupné z: <https://mymuseumshop.com/en/my-museum-shop/classic-brown-mini-praxinoscope>
- [82] NAJEEB, Fatima. History of Animation and its Evolution in Film. Motioncue [online]. New York: MotionCue., c2012-2026, 12.18.2020 [cit. 2023-04-09]. Dostupné z: <https://motioncue.com/history-and-evolution-of-animation-in-film/>
- [83] Gertie the Dinosaur. In: AllEars [online]. AllEarsNet, c1996-2023, Posted on September 9, 2013 [cit. 2023-04-12]. Dostupné z: <https://allears.net/2013/09/09/jims-attic-gertie-the-dinosaur/>
- [84] SUSEPONG, Garni. Felix The Cat. In: Pixels [online]. Chicago: Pixels.com, c2023, February 3rd, 2020 [cit. 2023-04-12]. Dostupné z: <https://pixels.com/featured/felix-the-cat-garni-susepong.html>
- [85] Betty Boop. In: Pixabay [online]. 10. dubna 2014 [cit. 2023-04-12]. Dostupné z: <https://pixabay.com/cs/vectors/betty-boop-d%C3%A1ma-%C5%BEensk%C3%BD-%C5%BEena-d%C3%ADvka-295419/>

- [86] Tex Avery. Britannica [online]. Chicago: Encyclopædia Britannica, c2023, Feb 22, 2023 [cit. 2023-04-16]. Dostupné z: <https://www.britannica.com/biography/Tex-Avery>
- [87] Bugs Bunny. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 18 September 2016 [cit. 2023-04-12]. Dostupné z: [https://en.wikipedia.org/wiki/File:Bugs\\_Bunny.svg](https://en.wikipedia.org/wiki/File:Bugs_Bunny.svg)
- [88] SANDERS, Adrien-Luc. Definition and Examples of Limited Animation. Lifewire [online]. New York: Lifewire, August 26, 2019 [cit. 2023-04-16]. Dostupné z: <https://www.lifewire.com/what-is-limited-animation-140520>
- [89] The Uses Of Animation. In: Hive Studio [online]. Cairo: Hive Studio, DECEMBER 7, 2017 [cit. 2023-04-09]. Dostupné z: <https://hivestudio.net/the-uses-of-animation/>
- [90] SHANE, Aadmin. 10 Best Uses of Animation. In: AnimatedJobs.com [online]. Animated Jobs, c2023, April 18, 2020 [cit. 2023-04-09]. Dostupné z: <https://animatedjobs.com/10-best-uses-of-animation/>
- [91] NEW WORLD ENCYCLOPEDIA. Pictogram. Newworldencyclopedia.org [online]. [cit. 2023-04-07]. Dostupné z: <https://www.newworldencyclopedia.org/entry/Pictogram>
- [92] Pictogram. Safeopedia.com [online]. Edmonton: Safeopedia, c2023, August 21, 2017 [cit. 2023-04-07]. Dostupné z: <https://www.safeopedia.com/definition/4992/pictogram>
- [93] Pictogram. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation [cit. 2023-04-07]. Dostupné z: <https://en.wikipedia.org/wiki/Pictogram>
- [94] POŠTOLKA, Jakub. Informační grafika. Plzeň, 2015. Diplomová práce. Západočeská univerzita v Plzni, Fakulta filozofická, Katedra filozofie. Vedoucí práce PhDr. Lada Hanzelínová, Ph.D.
- [95] BLANDFORD, Tom. Ancient Native American Pictographs. In: Flickr [online]. SmugMug, c2023, September 28, 2011 [cit. 2023-04-11]. Dostupné z: <https://www.flickr.com/photos/tomblandford/31453234577>
- [96] BASS, Cary. Radioactive. In: Wikimedia Commons [online]. San Francisco: Wikimedia Foundation, 19 January 2006 [cit. 2023-04-11]. Dostupné z: <https://commons.wikimedia.org/wiki/File:Radioactive.svg>
- [97] ALEXANDER, Paul. The Curious Case of the Failed Pictogram. Medium [online]. Medium, May 14, 2016 [cit. 2023-04-07]. Dostupné z: [https://medium.com/@Follow\\_the\\_arrow/the-curious-case-of-the-failed-pictogram-4457de62fdfe](https://medium.com/@Follow_the_arrow/the-curious-case-of-the-failed-pictogram-4457de62fdfe)
- [98] TF 015: Flight departures. In: Wikimedia Commons [online]. San Francisco: Wikimedia Foundation, 1 January 2023 [cit. 2023-04-11]. Dostupné z: [https://commons.wikimedia.org/wiki/File:ISO\\_7001\\_PI\\_TF\\_015.svg](https://commons.wikimedia.org/wiki/File:ISO_7001_PI_TF_015.svg)
- [99] Departing Flights. In: Wikimedia Commons [online]. San Francisco: Wikimedia Foundation, 27 July 2007 [cit. 2023-04-11]. Dostupné z: [https://commons.wikimedia.org/wiki/File:Aiga\\_departingflights.svg](https://commons.wikimedia.org/wiki/File:Aiga_departingflights.svg)

- [100] Autodesk AutoCAD. Arkance Systems [online]. Praha 4: Arkance Systems, c2023 [cit. 2023-04-10]. Dostupné z: <https://www.arkance-systems.cz/produkty/autodesk-autocad>
- [101] What is AutoCAD and what is it for?. StacBond [online]. [cit. 2023-04-10]. Dostupné z: <https://stacbond.com/en/what-is-autocad-and-what-is-it-for/>
- [102] Who Uses AutoCAD and Why Is It Important?. Charter College [online]. Anchorage: Charter College, c2023, August 12, 2020 [cit. 2023-04-10]. Dostupné z: <https://www.chartercollege.edu/news-hub/who-uses-autocad-and-why-it-important>
- [103] KENNEDY, Luke. A Brief History of AutoCAD. Scan2CAD [online]. Worcester: Scan2CAD, Jan 5, 2014 [cit. 2023-04-10]. Dostupné z: <https://www.scan2cad.com/blog/tips/autocad-brief-history/>
- [104] LASTIČ, Daniel. 3D modely vytvořené v programu AutoCAD pro výuku Mongeova promítání. Olomouc, 2019. Bakalářská práce. Univerzita Palackého v Olomouci, Přírodovědecká fakulta. Vedoucí práce Mgr. Jiří J. Kratochvíl, Ph.D.
- [105] DXF formát. AutoCADblog.cz [online]. Ostrava: Adeon CZ, c2012-2020 [cit. 2023-04-10]. Dostupné z: <https://www.autocadblog.cz/dxf/>
- [106] About AutoLISP Applications. Autodesk AutoCad 2023 [online]. San Rafael: Autodesk, c2023 [cit. 2023-04-06]. Dostupné z: <https://help.autodesk.com/view/OARX/2023/ENU/?guid=GUID-16DC15FC-5329-492E-B66A-401D49CF971F>
- [107] AutoLISP. AfraLISP [online]. c2023 [cit. 2023-04-06]. Dostupné z: <https://www.afralisp.net/autolisp/>
- [108] Visual LISP. AfraLISP [online]. c2023 [cit. 2023-04-06]. Dostupné z: <https://www.afralisp.net/visual-lisp/>
- [109] Úvod (AutoLISP). Autodesk AutoCAD 3D Map 2014 [online]. San Rafael: Autodesk, c2013 [cit. 2023-04-06]. Dostupné z: <http://docs.autodesk.com/MAP/2014/CSY/index.html?url=filesACD/GUID-A0E9D801-8BE9-4BF1-85E8-3807E15F3B71.htm,topicNumber=ACDd30e552135>
- [110] Free AutoLISP and Visual LISP code snippets for AutoCAD. JTB World [online]. JTB World, c2001-2023 [cit. 2023-04-06]. Dostupné z: <https://jtbworld.com/autolisp-visual-lisp>
- [111] About the AutoCAD Visual Basic for Applications Interface (VBA/ActiveX). Autodesk AutoCad 2023 [online]. San Rafael: Autodesk, c2023 [cit. 2023-04-06]. Dostupné z: <https://help.autodesk.com/view/OARX/2023/ENU/?guid=GUID-AAF6095B-ABF9-4063-991E-7C0F58B36401>
- [112] Advantages of Using VBA with AutoCAD (Developing a Simple VBA Application). What-when-how [online]. [cit. 2023-04-06]. Dostupné z: <http://what-when-how.com/autocad-vba/advantages-of-using-vba-with-autocad-developing-a-simple-vba-application/>

- [113] ActiveX and COM (ActiveX Controls) (AutoCAD VBA). What-when-how [online]. [cit. 2023-04-06]. Dostupné z: <http://what-when-how.com/autocad-vba/activex-and-com-activex-controls-autocad-vba/>
- [114] About the AutoCAD ActiveX Technology (ActiveX). Autodesk AutoCad 2023 [online]. San Rafael: Autodesk, c2023 [cit. 2023-04-06]. Dostupné z: <https://help.autodesk.com/view/OARX/2023/ENU/?guid=GUID-9C082B2D-015E-43C1-9168-623A2EA91D94>
- [115] Overview of ObjectARX (Concept). Documentation.HELP! [online]. documentation.help [cit. 2023-04-06]. Dostupné z: <https://documentation.help/AutoCAD-ACG/WS73099cc142f4875513fb5cd10c4aa30d6b-7f5c.htm>
- [116] AutoCAD ObjectARX SDK Developer Center. Autodesk [online]. San Rafael: Autodesk, c2023 [cit. 2023-04-06]. Dostupné z: <https://www.autodesk.com/developer-network/platform-technologies/autocad/objectarx>
- [117] C# Tutorial. W3Schools [online]. Sandnes: Refsnes Data, c1999-2023 [cit. 2023-04-06]. Dostupné z: <https://www.w3schools.com/cs/index.php>
- [118] What is .NET?. Codecademy [online]. New York: Codecademy, c2023 [cit. 2023-04-06]. Dostupné z: <https://www.codecademy.com/article/what-is-net>
- [119] What Is .NET?. Aws [online]. Seattle: Amazon Web Services, c2023 [cit. 2023-04-06]. Dostupné z: <https://aws.amazon.com/what-is/net/>
- [120] ČÁPKA, David. Lekce 1 - Úvod do C# a .NET frameworku. Itnetwork.cz [online]. Praha: itnetwork.cz, c2023 [cit. 2023-04-06]. Dostupné z: <https://www.itnetwork.cz/csharp/zaklady/c-sharp-tutorial-uvod-do-jazyka-a-dot-net-framework>

## **PŘÍLOHY**

Příloha A – Tabulka příkazů nápovědy .....	78
Příloha B – Ukázka celého kódu nápovědy .....	79
Příloha C – Nastavení Visual Studia 2019 a AutoCADu .....	82
Příloha D – Grafy s výsledky uživatelského průzkumu .....	85
Příloha E – Zdrojové kódy nápovědy .....	88

## PŘÍLOHA A – TABULKA PŘÍKAZŮ NÁPOVĚDY

Tato příloha obsahuje tabulku příkazů, pomocí kterých se zobrazí nápovědy systémových proměnných v AutoCADu.

DimDecHelpComboBox	Nápověda pro systémovou proměnnou DIMDAC, řešená pomocí ComboBoxu
DimDecHelpPlusMinus	Nápověda pro systémovou proměnnou DIMDAC, řešená pomocí interaktivních symbolů +/-
DimExeHelp	Nápověda pro systémovou proměnnou DIMEXE
DimLimHelp	Nápověda pro systémovou proměnnou DIMLIM
DimRndHelp	Nápověda pro systémovou proměnnou DIMRND
DimTadHelpComboBox	Nápověda pro systémovou proměnnou DIMITAD, řešená pomocí ComboBoxu
DimTadHelpText	Nápověda pro systémovou proměnnou DIMITAD, řešená pomocí interaktivního textu
DimTihHelp	Nápověda pro systémovou proměnnou DIMITIH
DimTixHelp	Nápověda pro systémovou proměnnou DIMITIX
DimToflHelp	Nápověda pro systémovou proměnnou DIMTOFL
DimTohHelp	Nápověda pro systémovou proměnnou DIMTOH
DimTolHelp	Nápověda pro systémovou proměnnou DIMITOL
DimTpHelp	Nápověda pro systémovou proměnnou DIMTP

## PŘÍLOHA B – UKÁZKA CELÉHO KÓDU NÁPOVĚDY

Tato příloha obsahuje kompletní zdrojový kód nápovědy pro dvě systémové proměnné.

Seznam zdrojových kódů:

- Zdrojový kód 1: Celý kód nápovědy pro systémovou proměnnou DIMTIH (Zdroj vlastní)
- Zdrojový kód 2: Celý kód nápovědy pro systémovou proměnnou DIMTP (Zdroj vlastní)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Autodesk_help
{
    public partial class DIMTIH : Form
    {
        public DIMTIH()
        {
            InitializeComponent();

            //nastavení obrázku v hlavním PictureBoxu
            pictureBoxImage.Image = Image.FromFile("C:/Program Files/" +
                "Autodesk/ApplicationPlugins/Pics/DIMTIH_OFF.jpg");
            //nastavení doplňující informace v Labelu pod obrázkem
            labelNastaveni.Text = "Proměnná se nastavuje na hodnotu DIMTIH = Ne";
            //nastavení ikony kurzoru při najetí na komponentu
            checkBoxZapnuto.Cursor = Cursors.Hand;
        }

        //událost při kliknutí na CheckBox
        private void checkBoxZapnuto_CheckedChanged(object sender, EventArgs e)
        {
            //kontrola, zda je CheckBox aktivní nebo ne
            if (checkBoxZapnuto.Checked == true)
            {
                //změna obrázku v hlavním PictureBoxu podle stavu CheckBoxu
                pictureBoxImage.Image = Image.FromFile("C:/Program Files/" +
                    "Autodesk/ApplicationPlugins/Pics/DIMTIH_ON.jpg");
                //změna doplňující informace v Labelu pod obrázkem
                labelNastaveni.Text = "Proměnná se nastavuje na hodnotu " +
                    "DIMTIH = Ano";
                //změna barvy textu u CheckBoxu
                checkBoxZapnuto.ForeColor = System.Drawing.Color.Silver;
            }
            else
            {
                //změna obrázku v hlavním PictureBoxu podle stavu CheckBoxu
                pictureBoxImage.Image = Image.FromFile("C:/Program Files/" +
                    "Autodesk/ApplicationPlugins/Pics/DIMTIH_OFF.jpg");
                //změna doplňující informace v Labelu pod obrázkem
                labelNastaveni.Text = "Proměnná se nastavuje na hodnotu " +

```

```

        "DIMTIH = Ne";
        //změna barvy textu u CheckBoxu
        checkBoxZapnuto.ForeColor = System.Drawing.Color.White;
    }
}
}
}

```

Zdrojový kód 1

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Autodesk_help
{
    public partial class DIMTP : Form
    {
        public DIMTP()
        {
            InitializeComponent();

            //nastavení viditelnosti Labelu
            labelHorniMez.Visible = false;
            //nastavení obrázku v hlavním PictureBoxu
            pictureBoxImage.Image = Image.FromFile("C:/Program Files/" +
                "Autodesk/ApplicationPlugins/Pics/DIMTP_0.jpg");
        }

        //událost při zadání hodnoty do TextBoxu
        private void textBoxHodnota_TextChanged(object sender, EventArgs e)
        {
            //získání hodnoty z TextBoxu
            string hodnotaText = textBoxHodnota.Text;
            hodnotaText = hodnotaText.Replace(",", ".");
            double hodnotaCislo;

            //kontrola, zda hodnota z TextBoxu není prázdná
            if (hodnotaText == "")
            {
                hodnotaText = "0";
            }
            //kontrola, zda je hodnota z TextBoxu skutečně o číslo a
            //jestli není záporná
            if (double.TryParse(hodnotaText, out hodnotaCislo)
                || hodnotaText == "-" || hodnotaText == "+")
            {
                if (hodnotaCislo == 0)
                {
                    //změna viditelnosti Labelu
                    labelHorniMez.Visible = false;
                    //změna obrázku v hlavním PictureBoxu
                    pictureBoxImage.Image = Image.FromFile("C:/Program Files/" +
                        "Autodesk/ApplicationPlugins/Pics/DIMTP_0.jpg");
                }
            }
        }
    }
}

```



```

else if (hodnotaCislo < 0)
{
    textBoxHodnota.Text = "0.0000";
    //zobrazení chybové zprávy
    MessageBox.Show("Nelze zadat záporné číslo", "Upozornění");
}
else
{
    if (hodnotaText.Length < 10)
    {
        //změna viditelnosti Labelu
        labelHorniMez.Visible = true;
        //změna obrázku v hlavním PictureBoxu
        pictureBoxImage.Image = Image.FromFile("C:/Program Files/" +
            "Autodesk/ApplicationPlugins/Pics/DIMTP_1.jpg");
        //nastavení zadané hodnoty do Labelu zobrazující
        //horní odchylku
        string cislo = String.Format("{0:F4}", hodnotaCislo);
        labelHorniMez.Text = cislo.Replace(",", ".") + " ";
    }
}
else
{
    MessageBox.Show("Zadaná hodnota není číslo", "Upozornění");
    textBoxHodnota.Text = "0.0000";
}
}
}
}

```

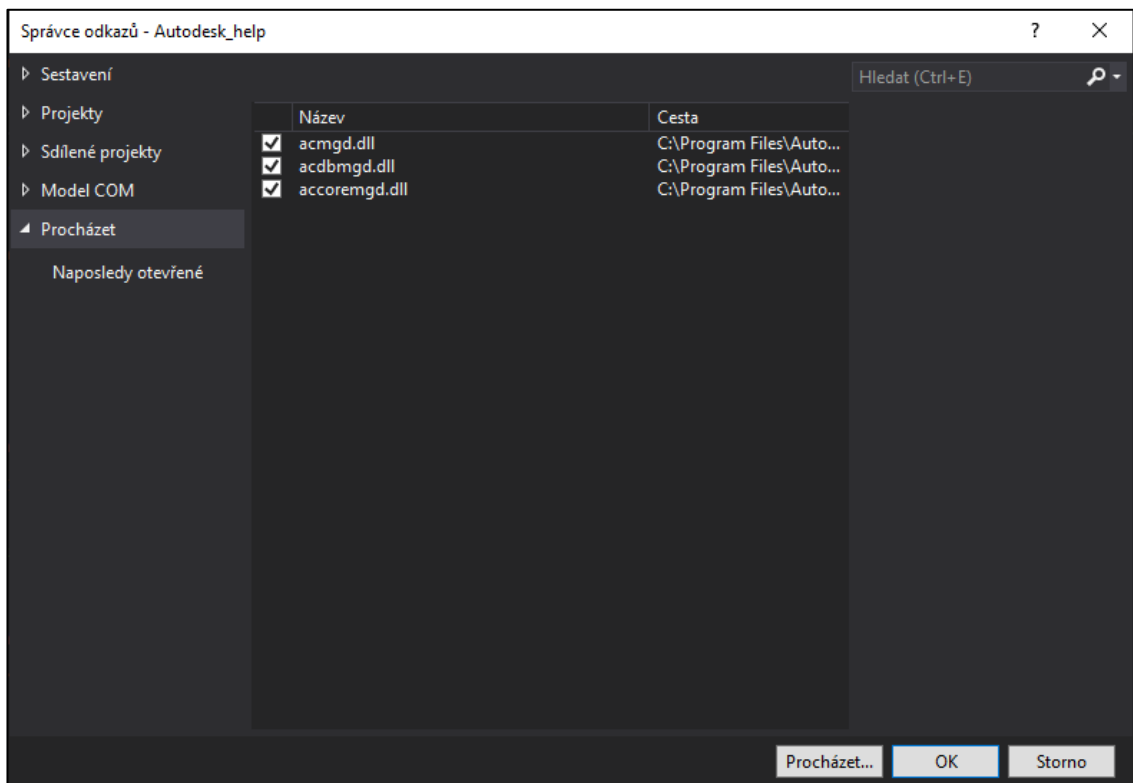
Zdrojový kód 2

# PŘÍLOHA C – NASTAVENÍ VISUAL STUDIA 2019 A AUTOCADU

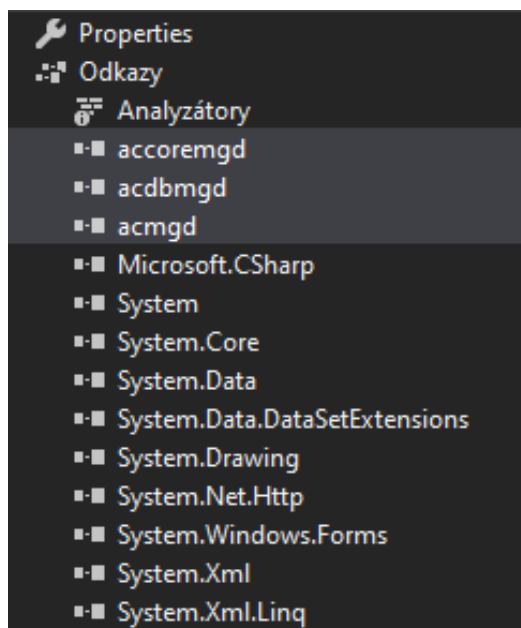
Tato příloha obsahuje obrázky nastavení Visual Studia 2019 (VS) a AutoCADu.

Seznam obrázků:

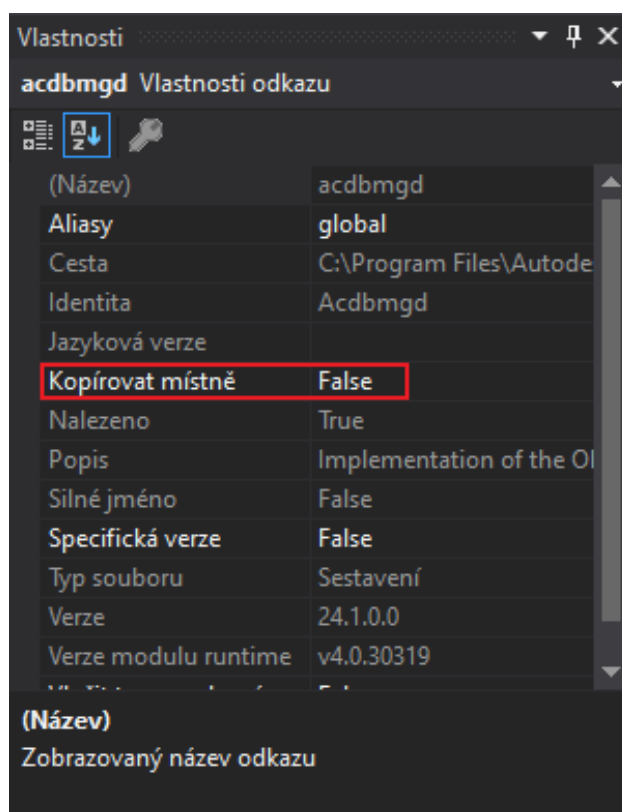
- Obrázek 1 – Dialog pro přidání odkazů do C# projektu ve VS 2019 (Zdroj vlastní)
- Obrázek 2 – Ukázka přidáných odkazů na DLL soubory v C# projektu ve VS 2019 (Zdroj vlastní)
- Obrázek 3 – Nastavení vlastnosti odkazu v C# projektu ve VS 2019 (Zdroj vlastní)
- Obrázek 4 – Ukázka nastavení pro spuštění externího programu v C# projektu ve VS 2019 (Zdroj vlastní)
- Obrázek 5 – Přidání příkazu do souboru lsp (Zdroj vlastní)



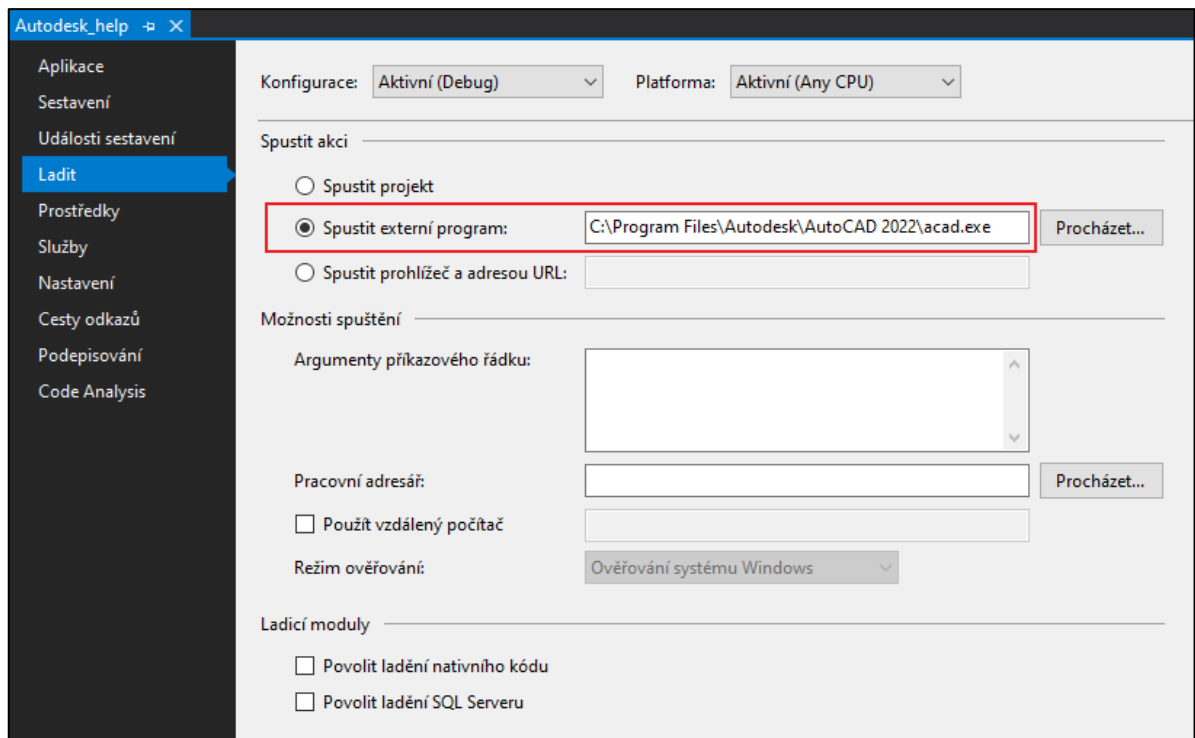
Obrázek 1



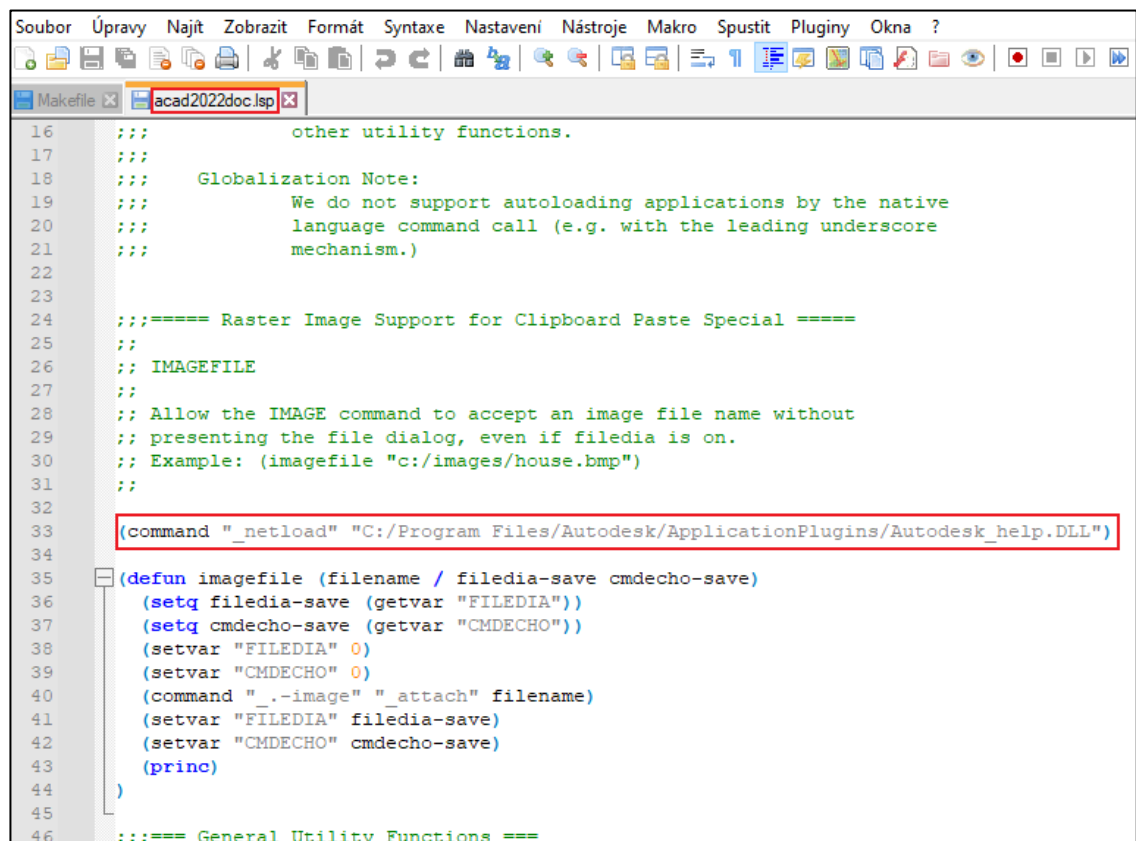
Obrázek 2



Obrázek 3



Obrázek 4



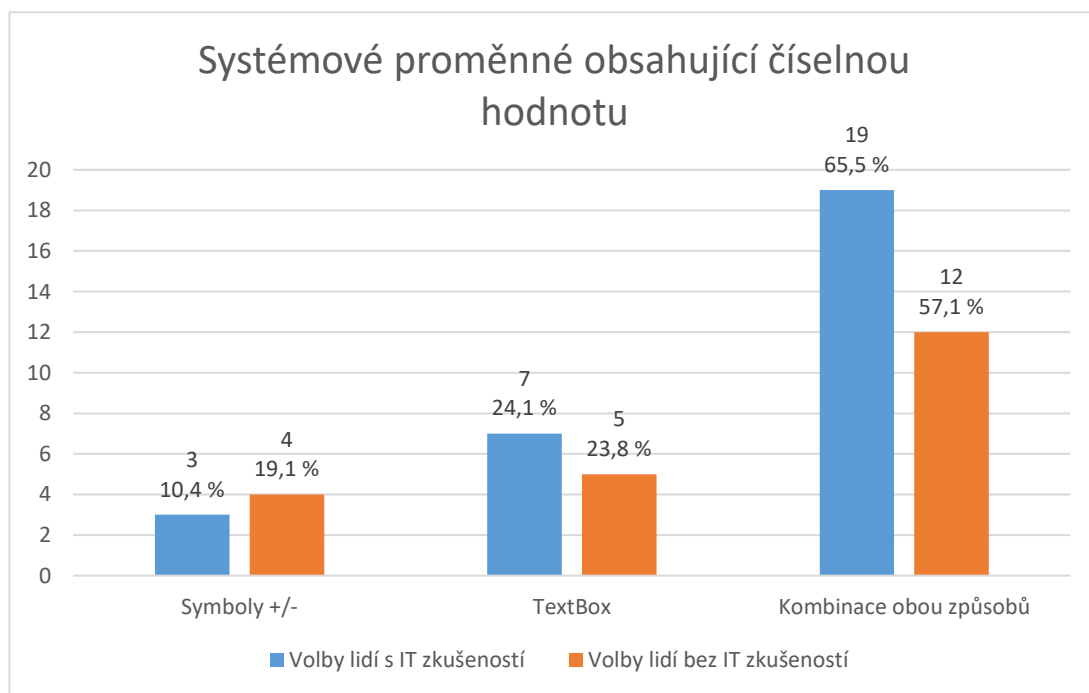
Obrázek 5

## PŘÍLOHA D – GRAFY S VÝSLEDKY UŽIVATELSKÉHO PRŮZKUMU

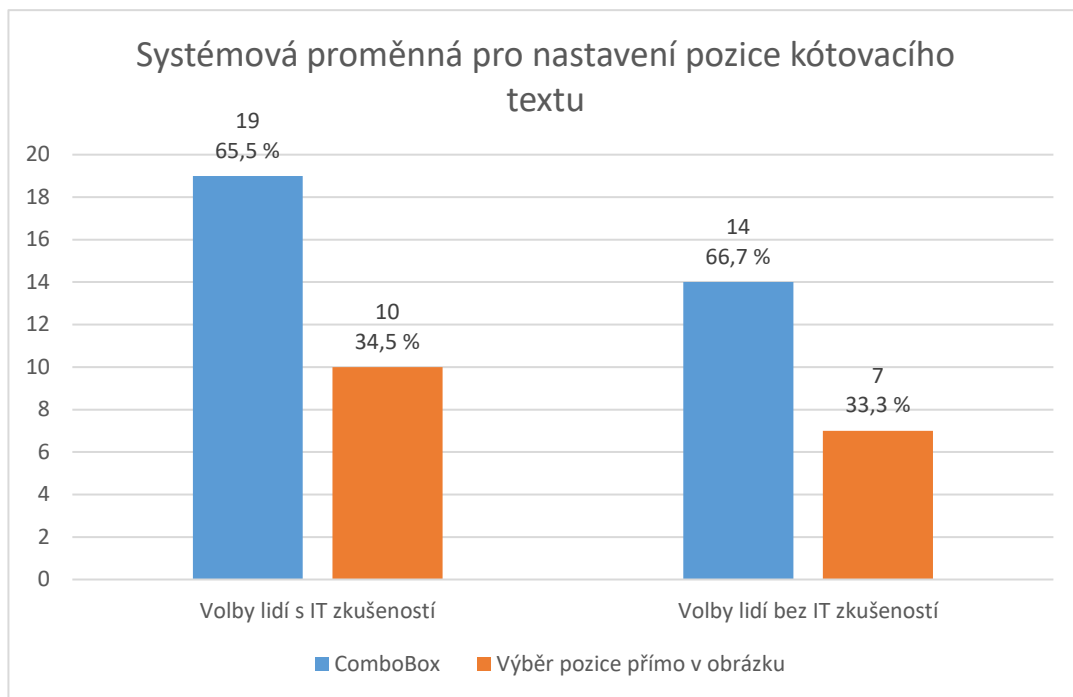
Tato příloha obsahuje grafické znázornění výsledků uživatelského průzkumu.

Seznam grafů:

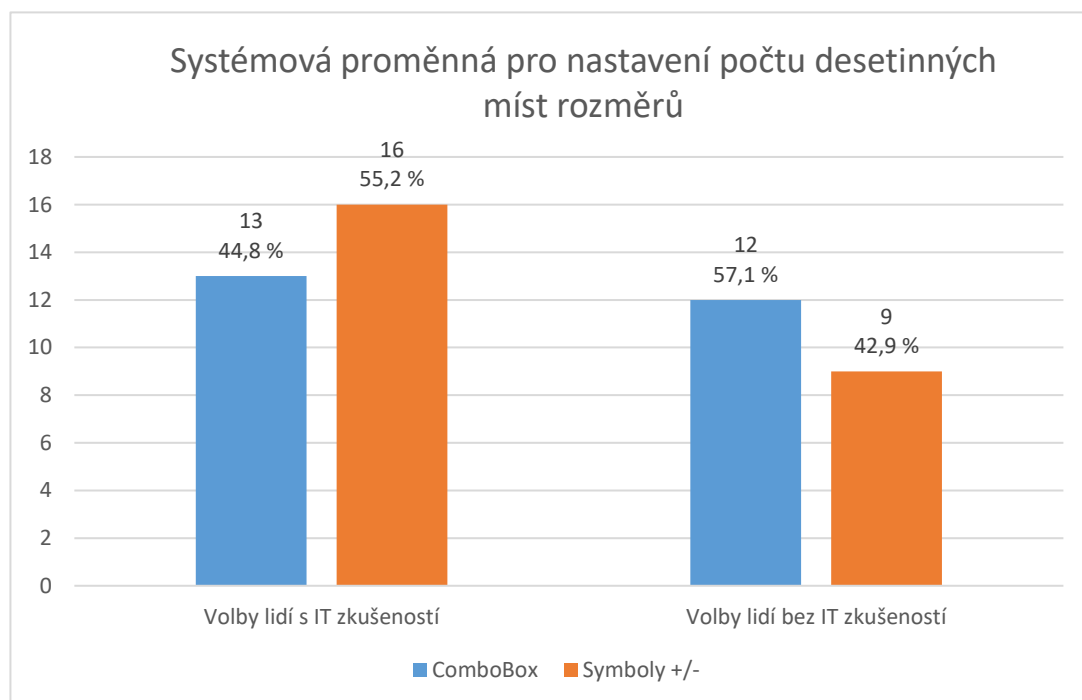
- Graf 1 – Volba provedení nápovědy v kategorii *Systémové proměnné obsahující číselnou hodnotu* (Zdroj vlastní)
- Graf 2 – Volba provedení nápovědy v kategorii *Systémová proměnná pro nastavení pozice kótovacího textu* (Zdroj vlastní)
- Graf 3 – Volba provedení nápovědy v kategorii *Systémová proměnná pro nastavení počtu desetinných míst rozměrů* (Zdroj vlastní)
- Graf 4 – Volba provedení nápovědy v kategorii *Systémové proměnné nabývají stavu Zapnuto/Vypnuto* (Zdroj vlastní)



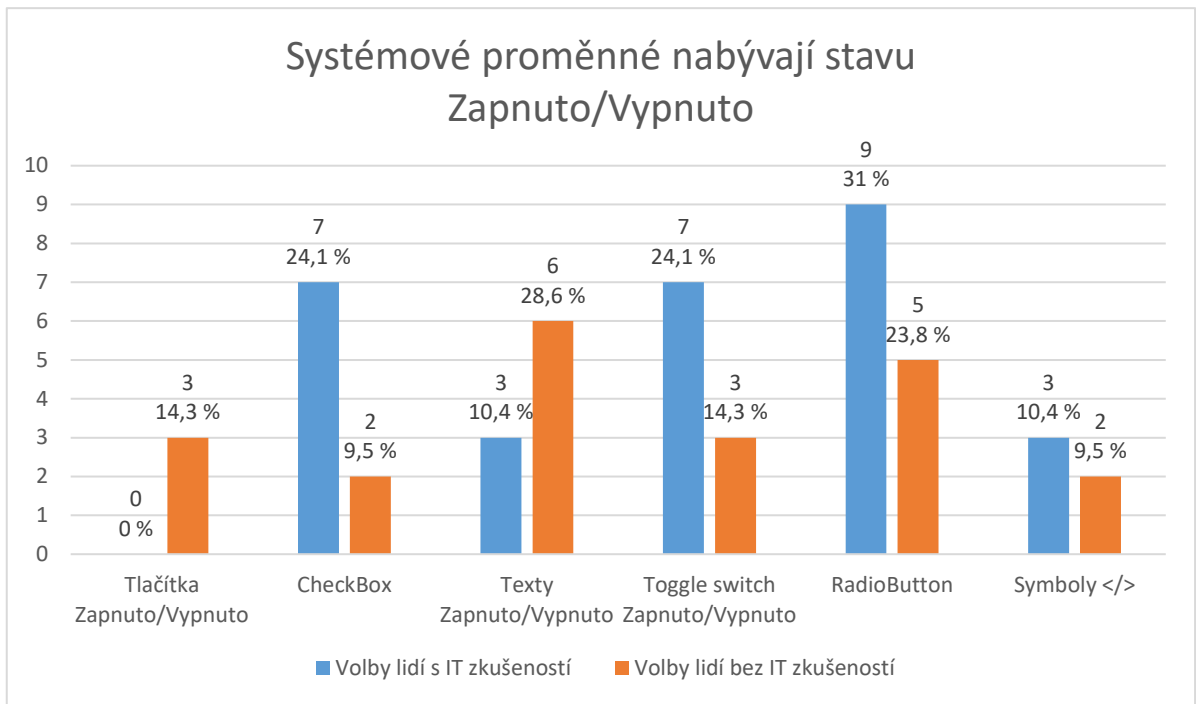
Graf 1



Graf 2



Graf 3



Graf 4

## **PŘÍLOHA E – ZDROJOVÉ KÓDY NÁPOVĚDY**

Příloha obsahuje archiv Kopecký\_st61007\_help.zip.

Obsah archivu:

- Složka Autodesk\_help obsahující všechny zdrojové kódy v jazyce C# spolu se spustitelným řešením ve Visual Studio 2019.
- Složka Pics obsahující všechny obrázky použité v nápovědě.