

Technická univerzita v Liberci

FAKULTA PEDAGOGICKÁ

Katedra: Pedagogiky a Psychologie
Studijní program: M7503 Učitelství pro 2. stupeň základní školy
Kombinace: anglický jazyk - informatika

DIDAKTICKÉ VYUŽITÍ AUTORSKÉHO SYSTÉMU MACROMEDIA
AUTHORWARE PRO KONSTRUKCI APLIKACÍ
O MACROMEDIA AUTHORWARE

DIDACTIC APPLICATION IN AUTHORIZING SYSTEM
MACROMEDIA AUTHORWARE FOR APPLICATION
CONSTRUCTION ABOUT MACROMEDIA AUTHORWARE

DIDAKTISCHE NUTZUNG DES AUTORENSYSTEMS
MACROMEDIA AUTHORWARE FÜR DEN AUFBAU DER
APPLIKATIONEN ÜBER MACROMEDIA AUTHORWARE

Diplomová práce: 07–FP–KPP– 08

Autor:

Jan JÍLEK

Podpis:

Adresa:

Skalní 494/3
460 01 Liberec 5

Vedoucí práce: Doc. PaedDr. Jiří Nikl, CSc.

Konzultant:

Počet

stran	slov	obrázků	tabulek	pramenů	příloh
148	21 752	85	1	18	3

V Liberci dne: 18.12.2007

TU v Liberci, FAKULTA PEDAGOGICKÁ

461 17 LIBEREC 1, Hálkova 6

Tel.: 485 352 515

Fax: 485 352 332

Katedra: Pedagogiky a psychologie

ZADÁNÍ DIPLOMOVÉ PRÁCE

(pro magisterský studijní program)

pro (diplomant) Jan Jílek
adresa: Skalní 494/3, Liberec 46001
obor (kombinace): Učitelství AJ/IF pro druhý stupeň ZŠ

Název DP: Didaktické využití autorského systému Macromedia Authorware pro konstrukci aplikací o Macromedia Authorware

Název DP v angličtině: Didactic application in authoring system Macromedia Authorware for application construction about Macromedia Authorware

Vedoucí práce: Doc. PaedDr. Jiří Nikl, CSc.

Konzultant:

Termín odevzdání: leden 2008

Pozn. Podmínky pro zadání práce jsou k nahlédnutí na katedrách. Katedry rovněž formulují podrobnosti zadání. Zásady pro zpracování DP jsou k dispozici ve dvou verzích (stručné.resp. metodické pokyny) na katedrách a na Děkanátě Fakulty pedagogické TU v Liberci.

V Liberci dne

.....
děkan vedoucí katedry

Převzal (diplomant):

Datum:

Podpis:

Název DP: Didaktické využití autorského systému Macromedia Authorware pro konstrukci aplikací o Macromedia Authorware

Vedoucí práce: Doc. PaedDr. Jiří Nikl, CSc.

Úvod: Teoretická část diplomové práce bude ve vztahu k aprobaci studujícího (Informatika – Anglický jazyk) zahrnovat problematiku dat (proměnné, funkce) v autorském systému Macromedia Authorware. Studující prokáže schopnost překladů anglických odborných textů v oblasti informační výchovy a jejich didaktickou transformaci do metodických návodů pro učitele.

Použité metody: Odborný překlad, komparace textů, didaktická transformace obsahu textů

Praktická část bude obsahovat rozsáhlý osobitý příspěvek autora k pedagogické praxi - *soubory počítačových aplikací pro pochopení práce v autorském systému Macromedia Authorware.*

Práce přispěje k rozvoji odbornosti studujícího, jeho prostřednictvím zanesou nové myšlenky do praxe školní výuky v souvislosti s rozpracováním rámcového vzdělávacího programu a přispěje katedře KPP TUL k rozšíření didaktického portfolia metodických materiálů pro didaktické využívání výpočetní techniky.

Cíl:

- odborný překlad a didaktická transformace vymezené části anglického návodu k autorskému systému Macromedia Authorware
- vytvoření rozsáhlého souboru výukových videí k autorskému systému Macromedia Authorware, využitelných ve výuce o Macromedia Authorware

Požadavky: Pravidelné konzultace (každé 3 týdny 2 hodiny), zajištění autorského systému Macromedia Authorware

Literatura: Macromedia Authorware 7

(URL: <http://www.digitalmedia.cz>)

Macromedia Authorware 7 – help

(elektronická verze anglického návodu Macromedia Authorware 7)

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mojí diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce.

V Liberci dne: 18.12. 2007

Jan Jílek

Poděkování:

Děkuji doc. PaedDr. Jiřímu Niklovi, CSc. za odborné vedení diplomové práce, poskytování rad a materiálových podkladů k práci. Dále bych rád poděkoval své mamince za podporu, kterou mi během studií poskytla.

DIDAKTICKÉ VYUŽITÍ AUTORSKÉHO SYSTÉMU MACROMEDIA AUTHORWARE PRO KONSTRUKCI APLIKACÍ O MACROMEDIA AUTHORWARE

JÍLEK Jan

DP–2008

Vedoucí DP: Doc. PaedDr. Jiří Nikl, CSc.

Resumé

Teoretická část diplomové práce se zabývá problematikou dat (proměnných a funkcí) v autorském systému Macromedia Authorware (MAW).

Praktická část obsahuje autorem v MAW konstruované soubory výukových videí a počítačových aplikací pro pochopení práce v autorském systému Macromedia Authorware. Práce je určena pedagogům neprogramátorům, kteří budou vytvářet v MAW didaktické aplikace.

Klíčová slova: Macromedia Authorware, proměnné, funkce, didaktická aplikace, autorský systém

DIDACTIC APPLICATION IN AUTHORIZING SYSTEM MACROMEDIA AUTHORWARE FOR APPLICATION CONSTRUCTION ABOUT MACROMEDIA AUTHORWARE

Summary

The theoretical part of my diploma thesis is focused working with data (variables and functions) in an authoring system Macromedia Authorware (MAW).

The practical part adds didactic video tutorials made in MAW by author of this thesis. These video tutorials function as model on how to construct didactic application in an authoring system Macromedia Authorware. This part of the work contains also sample program files for reinforcing the understanding how to construct didactic applications in MAW. This diploma thesis is intended to be used by teachers with no programming skills, who are constructing computer tests in MAW.

Keywords: Macromedia Authorware, variables, functions, didactic application, authoring system

DIDAKTISCHE NUTZUNG DES AUTORENSYSTEMS MACROMEDIA AUTHORWARE FÜR DEN AUFBAU DER APPLIKATIONEN ÜBER MACROMEDIA AUTHORWARE

Zusammenfassung

Der theoretische Teil der Diplomarbeit beschäftigt sich mit der Problematik der Daten (Variablen und Funktionen) in dem Autorensystem Macromedia Authorware (MAW).

Der praktische Teil beinhaltet von dem Autor in dem System MAW konstruierte Dateien der Videosequenzen und Rechnerapplikationen für den Verständniss der Arbeit in dem Autorensystem Macromedia Authorware.

Diese Arbeit ist bestimmt für die Pädagogen Nichtprogrammierer, die im MAW didaktische Applikationen erzeugen wollen.“

Schlüsselwörter: Macromedia Authorware, Variabeln, Funktionen, Didaktische applikationen, Autorensystems

Obsah:

1. Úvod	15
1.1 Zdůvodnění aktuálnosti tématu	15
1.2 Zdůvodnění volby tématu	15
1.3 Cíle a metody vytváření diplomové práce	16
2. Základní pojmy v Macromedia Authorware	18
2.1 Autorský systém	18
2.2 Macromedia Authorware	18
2.3 Prostředky tvorby projektu	18
2.3.1 Pracovní okno	19
2.3.2 Okno prezentace	19
2.3.3 Časová osa	19
2.3.4 Lišta hlavního menu	20
2.3.5 Nástrojová lišta	20
2.3.6 Panel nástrojů	20
2.3.7 Praporky start a stop	21
2.3.8 Paleta barev ikon	21
2.4 Paleta ikon	21
2.4.1 Zobrazovací ikona (Display Icon)	21
2.4.2 Pohybová ikona (Motion Icon).....	21
2.4.3 Mazací ikona (Erase Icon)	21
2.4.4 Čekací ikona (Wait Icon).....	22
2.4.5 Slučovací ikona (Map Icon)	22
2.4.6 Filmová ikona (Digital Movie Icon).....	22
2.4.7 Zvuková ikona (Sound Icon)	22
2.4.8 DVD ikona (DVD Icon)	22
2.4.9 Ikona objektu Knowledge (Knowledge object Icon).....	22
2.4.10 Navigační ikona (Navigation Icon)	22
2.4.11 Stránkovací ikona (Framework Icon).....	23

2.4.12 Rozhodovací ikona (Decision Icon)	23
2.4.13 Interakční ikona (Interaction Icon)	23
2.4.14 Kalkulační ikona (Calculation Icon)	23
3. Proměnné	24
3.1. Co je proměnná	24
3.2 Typy proměnných v Macromedia Authorware	25
3.2.1 Číselné proměnné	25
3.2.2 Řetězcové proměnné	26
3.2.3 Logické proměnné	26
3.2.4 Proměnné typu Seznam	27
3.2.5 Znakové proměnné	28
3.2.6 Proměnné typu Rect	28
3.2.7 Proměnné typu Point	28
3.3 Systémové a uživatelské proměnné	29
3.3.1 Vytváření uživatelských proměnných	30
3.3.1.1 Používání proměnných v kalkulační ikoně	31
3.3.1.2 Zobrazování obsahu proměnných v okně prezentace	32
3.3.1.3 Používání komentářů	33
3.3.2 Kategorie systémových proměnných v Macromedia Authorware	33
3.3.2.1 Rozhodovací proměnné	34
3.3.2.1.1 AllSelected	34
3.3.2.1.2 PathCount	34
3.3.2.1.3 PathSelected	35
3.3.2.1.4 RepCount	36
3.3.2.1.5 SelectedEver	36
3.3.2.1.6 TimesSelected	37
3.3.2.2 Souborové proměnné	37
3.3.2.2.1 DiskBytes	37
3.3.2.3 Stránkovací proměnné	38
3.3.2.3.1 CurrentPageID	38
3.3.2.3.2 CurrentPageNum	39

3.3.2.3.3 PageCount.....	40
3.3.2.4 Obecné proměnné.....	40
3.3.2.4.1 MediaLength.....	40
3.3.2.4.2 MediaPlaying.....	41
3.3.2.4.3 MediaPosition.....	42
3.3.2.4.4 SoundPlaying.....	42
3.3.2.4.5 AltDown	42
3.3.2.4.6 CapsLock.....	42
3.3.2.4.7 ControlDown	43
3.3.2.4.8 Key	43
3.3.2.4.9 MouseDown, MiddleMouseDown, RightMouseDown.....	44
3.3.2.4.10 ShiftDown.....	44
3.3.2.4.11 e	45
3.3.2.4.12 Pi.....	45
3.3.2.4.13 FirstName	45
3.3.2.4.14 UserName	46
3.3.2.4.15 ScreenDepth	46
3.3.2.5 Ikonové proměnné.....	47
3.3.2.5.1 DisplayTop, DisplayLeft, DisplayX, DisplayY, DisplayWidth, DisplayHeight.....	47
3.3.2.5.2 Movable	48
3.3.2.6 Interakční proměnné.....	48
3.3.2.6.1 ResponseTime	49
3.3.2.6.2 TimeInInteraction	49
3.3.2.6.3 CharCount.....	51
3.3.2.6.4 EntryText.....	51
3.3.2.6.5 ForceCaps	52
3.3.2.6.6 NumCount	52
3.3.2.6.7 NumEntry, NumEntry2, NumEntry3.....	53
3.3.2.6.8 PresetEntry	53
3.3.2.6.9 WordCount	54
3.3.2.6.10 AllCorrectMatched	55

3.3.2.6.11 ChoiceCount	55
3.3.2.6.12 ChoiceNumber	55
3.3.2.6.13 ChoicesMatched	56
3.3.2.6.14 Correct	56
3.3.2.6.15 CorrectChoicesMatched, WrongChoicesMatched	57
3.3.2.6.16 FirstTryCorrect, FirstTryWrong	57
3.3.2.6.17 JudgedInteractions	58
3.3.2.6.18 JudgedResponses	58
3.3.2.6.19 PercentCorrect, PercentWrong	59
3.3.2.6.20 TotalCorrect, TotalWrong	59
3.3.2.7 Časové proměnné	60
3.3.2.7.1 Date	60
3.3.2.7.2 Day	60
3.3.2.7.3 FullDate	61
3.3.2.7.4 Year	61
3.3.2.7.5 FullTime	61
3.3.2.7.6 Hour	62
3.3.2.7.7 Minute	62
3.3.2.7.8 Sec	62
3.3.2.7.9 Time	62
3.3.2.8 Video proměnné	63
3.3.2.8.1 VideoDone	63
4. Funkce	64
4.1 Co je funkce	64
4.2 Operátory	64
4.3 Systémové a uživatelské funkce	66
4.3.1 Kategorie systémových funkcí	66
4.3.1.1 Řetězcové funkce	66
4.3.1.1.1 Capitalize	66
4.3.1.1.2 LowerCase, UpperCase	67
4.3.1.1.3 CharCount	68

4.3.1.1.4 NumCount	68
4.3.1.1.5 WordCount	69
4.3.1.1.6 GetNumber	70
4.3.1.2 Souborové funkce	71
4.3.1.2.1 AppendExtFile, WriteExtFile	71
4.3.1.2.2 ReadExtFile	72
4.3.1.3 Obecné funkce	73
4.3.1.3.1 MediaPause	73
4.3.1.3.2 MediaPlay	73
4.3.1.3.3 MediaSeek	74
4.3.1.3.4 PrintScreen	75
4.3.1.3.5 Quit	76
4.3.1.3.6 Restart	76
4.3.1.3.7 SyncPoint, SyncWait	77
4.3.1.3.8 Beep	77
4.3.1.3.9 WaitMouseUp	78
4.3.1.3.10 Initialize	79
4.3.1.3.11 Test	79
4.3.1.3.12 SetCursor	80
4.3.1.3.13 ShowCursor	81
4.3.1.4 Ikonové funkce	81
4.3.1.4.1 EraseAll, EraseIcon	81
4.3.1.4.2 Preload	82
4.3.1.5 Přeskakovací funkce	83
4.3.1.5.1 GoTo	83
4.3.1.5.2 JumpFile, JumpFileReturn	84
4.3.1.5.3 JumpOut, JumpOutReturn	85
4.3.1.5.4 JumpPrintReturn	87
4.3.1.6 Matematické funkce	88
4.3.1.6.1 ACOS	88
4.3.1.6.2 ASIN	88
4.3.1.6.3 ATAN	89

4.3.1.6.4 COS	89
4.3.1.6.5 SIN.....	89
4.3.1.6.6 TAN.....	89
4.3.1.6.7 ABS	90
4.3.1.6.8 MOD.....	91
4.3.1.6.9 SQRT.....	91
4.3.1.6.10 Random.....	91
4.3.1.6.11 INT	94
4.3.1.6.12 Round	94
4.3.1.7 Časové funkce	94
4.3.1.7.1 DateToNum	95
4.3.1.7.2 Date.....	96
4.3.1.7.3 Day	96
4.3.1.7.4 FullDate	97
4.3.1.7.5 Month	97
4.3.1.7.6 Year	97
4.3.1.8 Video funkce	98
4.3.1.8.1 VideoDisplay.....	98
4.3.1.8.2 VideoPause	98
4.3.1.8.3 VideoPlay	99
4.3.1.8.4 VideoSpeed.....	99
5. Soubory počítačových aplikací pro pochopení práce v autorském systému	
Macromedia Authorware.....	101
5.1 Postup konstrukce výukových videí pro pochopení práce v autorském systému	
Macromedia Authorware.....	101
5.2 Pracovní archy k výukovým videům o konstrukci didaktických aplikací	
v autorském systému Macromedia Authorware	103
5.2.1 Pracovní arch Autorizace didaktické aplikace.....	104
5.2.1.1 Komentář k pracovnímu archu číslo 1: Autorizace didaktické aplikace..	106
5.2.2 Testovací typ tlačítko (button).....	107

5.2.2.1 Komentář k pracovnímu archu číslo 2: Přidávání a modifikace testovacích úloh typu Button (tlačítko)	113
5.2.3 Typ Hot Spot (aktivní plocha)	115
5.2.3.1 Komentář k pracovnímu archu č. 3: Přidávání a modifikace testovacích úloh typu Hot Spot (aktivní plocha)	119
5.2.4 Typ Hot object (aktivní objekt)	119
5.2.4.1 Komentář k pracovnímu archu č. 4: Přidávání a modifikace testovacích úloh typu Hot Object (aktivní objekt)	124
5.2.5 Typ Target Area (cílová oblast).....	125
5.2.5.1 Komentář k pracovnímu archu č. 5: Přidávání a modifikace testovacích úloh typu Target Area (cílová oblast).....	130
5.2.6 Typ Key Press (stisk klávesy)	131
5.2.6.1 Komentář k pracovnímu archu č. 6: Přidávání a modifikace testovacích úloh typu Key Press (stisk klávesy).....	135
5.2.7 Typ Text Entry (textová odpověď).....	135
5.2.7.1 Komentář k pracovnímu archu č. 7: Přidávání a modifikace testovacích úloh typu Text Entry (textová odpověď).....	141
5.2.8 Vytváření spustitelných aplikací.	141
6. Závěr.....	144
7. Použité prameny	146
8. Přílohy	148
8.1 CD – ROM	148
8.1.1 Textová část diplomové práce	148
8.1.2 Soubor výukových videí pro pochopení práce v autorském systému MAW	148
8.1.3 Soubory předloh jednotlivých typů úloh pro tvorbu didaktických aplikací v MAW	148
8.1.4 Soubor pracovních archů k instruktážním videím.....	148
8.1.5 Soubor programů pro pochopení práce s vybranými systémovými funkcemi MAW	148

1. Úvod

1.1 Zdůvodnění aktuálnosti tématu

Svoji diplomovou práci jsem se rozhodl psát o tématu, které má velice blízko k počítačům a informačním technologiím vůbec. Myslím, že toto téma je stále velice aktuální. Dnes už si mnozí z nás (včetně mě) nedovedeme představit, že bychom nebyli neustále v kontaktu pomocí mobilního telefonu, že bychom nemohli napsat e-mail. O kolik by bylo studium bez počítačů těžší, protože bychom se jedním klikem nemohli přihlásit na zkoušku a seminární práce bychom psali ručně nebo na psacím stroji? Většina lidí už považuje dostupnost počítačů za tak běžnou věc, že je ani nenapadne přemýšlet o tom, zda-li počítač použít či ne. Počítač tu prostě je. Je kolem nás, je všude. V mobilním telefonu, v ledničce, v mikrovlnné troubě, v autě. Počítače získávají na důležitosti v nejrůznějších oborech lidské činnosti. V lékařství, ve vojenství, dokonce i ve sportu.

Pochopitelně se počítače výrazně prosazují i ve vzdělávání. Ulehčují práci učitelům a zefektivňují snahu žáků. Učitelé využívají počítačové systémy k přípravě přednášek, evidenci, klasifikaci, kontaktu s rodiči, apod. Podle Jacqueline Keane [16] počítače nejen usnadnily a urychlily přípravu hodin, ale dostupnost zdrojů na internetu poskytla nové podněty a inspiraci. Je tedy zřejmé, že počítačová gramotnost je důležitá a jestli bude pokračovat dosavadní trend, tak bude čím dál důležitější. Kdy ale začít s výukou informačních technologií? ERIC CaSSC [3] se domnívá, že není vhodné začlenit počítače do vzdělávání dětí mladších než tři roky, protože jejich styl učení na to není připraven. Patricia Cantor [2] poukazuje na fakt, že většina výukových programů je zaměřena na děti do pěti let. Je evidentní, že děti se začínají seznamovat s počítači ve stále mladším věku.

1.2 Zdůvodnění volby tématu

Nyní uvedu několik důvodů, které způsobily, že jsem si za téma své diplomové práce rozhodl vzít autorský systém Macromedia Authorware:

Prvním z nich je výše zmíněná aktuálnost tématu. Počítač a práce s ním patří již od útlého věku mezi důležité součásti mého života. Když jsem se v seminářích doc. Nikla seznámil se základy autorského systému Macromedia Authorware, zaujala mě možnost tvorby vlastních didaktických aplikací bez nutnosti programování. Na FP TUL studuji obor učitelství anglického jazyka a informatiky pro 2. stupeň ZŠ. Tato kombinace je pro diplomovou práci o autorském systému Macromedia Authorware (MAW) velice vhodná, protože mi umožňuje skloubit znalosti načerpané v obou předmětech aprobace. Kvůli nedostatečné české literatuře o MAW je práce s anglickými texty nutností a znalost informatiky usnadňuje orientaci v technických náležitostech problematiky. Když jsem tedy dostal nabídku psát diplomovou práci o MAW, tak moje rozhodování nebylo příliš složité, už kvůli zmíněné možnosti využít znalosti z obou předmětů, které studuji. Z vlastní zkušenosti vím, že použití didaktické aplikace během výuky je nejen zajímavé pro žáky, kteří velice rádi pracují na počítači, ale může být velice přínosná pro samotného vyučujícího. Například v mém případě se používání počítačových didaktických aplikací osvědčilo ve chvílích, kdy bych jinak musel psát rukou na tabuli. Vzhledem k mému tělesnému postižení (od narození mám dětskou mozkovou obrnu a chodím o dvou francouzských holích) je pro mě psaní na tabuli poměrně namáhavé. S použitím didaktické aplikace mohu výuku efektivně zorganizovat tak, aby byla zajímavá pro žáky a méně fyzicky náročná pro mě.

Dalším z hlavních důvodů volby tohoto tématu byl fakt, že mě autorský systém MAW a jeho možnosti oslovily natolik, že jsem se rozhodl ho zkusit zpopularizovat a doufám, že má diplomová práce fakticky přiblíží MAW dalším pedagogům a věřím, že s její pomocí budou vytvářet vlastní didaktické aplikace.

1.3 Cíle a metody vytváření diplomové práce

V diplomové práci se zabývám problematikou využití autorského systému Macromedia Authorware pro vytváření didaktických aplikací.

Stanovil jsem si následující **cíle**:

A. V teoretické části diplomové práce:

1) Obsahová analýza a komparace odborných textů se zkoumanou problematikou.

- 2) Objasnění problematiky dat (proměnných a funkcí) v autorském systému Macromedia Authorware na základě mého odborného překladu z anglických originálů.
- 3) Didaktická transformace získaných poznatků o proměnných a funkcích.

B. V praktické části diplomové práce:

- 1) Vytvoření výukových videí o tvorbě didaktických aplikací v autorském systému Macromedia Authorware.
- 2) Obsahové naplnění modelových úloh ukázkami procvičovacích úloh pro úroveň jazyka elementary či preintermediate.

Ke splnění cílů jsem použil následující **metody** přiměřené daným cílům:

- 1) Metoda obsahové analýzy dostupných literárních zdrojů.
- 2) Metoda překladu, logického uspořádání a částečné minimalizace obsahu zdrojů v rámci didaktické transformace učiva.
- 3) Metoda vytváření didaktické aplikace výzkumníkem.
- 4) Metoda ověření vlastností didaktické aplikace vlastním výkonem činnosti výzkumníka.

2. Základní pojmy v Macromedia Authorware

2.1 Autorský systém

Za autorský systém je dle Nikla [13] považován počítačový program, který umožňuje uživatelům (neprogramátorům) vytvářet počítačové aplikace. Programovací prvky autorského systému bývají skryty za tlačítka, ikonami a dalšími nástroji tak, že uživatel autorského systému dle [18] nepotřebuje umět programovat v programovacím jazyce a může při tvorbě aplikace využít zabudovaných nástrojů autorského systému, jako je např. umístování grafiky, možnost přehrávání zvuků a animací apod. I přesto, že autorský systém značně usnadňuje tvorbu aplikací snížením znalosti programování potřebné k vytváření počítačových aplikací na minimum, tak jsou stále nutné schopnosti logického myšlení a návrhu algoritmů. Vytváření aplikace v autorském systému je totiž ve skutečnosti pouze urychlený typ programování. Podle Skoumala [15] není sice nutné znát všechny složitosti programovacího jazyka, ale stále je třeba chápat, jak programy fungují a pracují.

2.2 Macromedia Authorware

Macromedia Authorware je dle Brdičky[1] jedním z nejpoužívanějších autorských systémů pro tvorbu didaktických a e-learningových aplikací. Integruje text, grafiku, animace, zvuk a video. Spojením těchto prvků vznikají velmi přesvědčivá multimediální řešení výukových kurzů. Authorware lze označit za optimalizovaný nástroj poskytující vhodné prostředí pro vývoj interaktivních aplikací. Díky intuitivnímu ovládní ikon přetahováním myši umožňuje rychlou a intuitivní tvorbu aplikací bez znalosti programovacího jazyka.

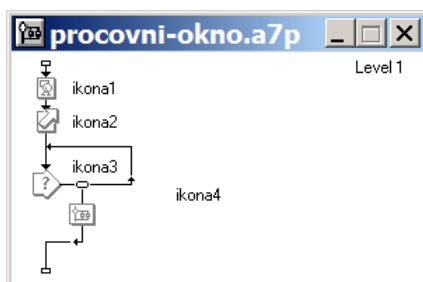
2.3 Prostředky tvorby projektu

Popis prostředků vychází z [9] překladem anglického textu, jeho transformací a minimalizací.

2.3.1 Pracovní okno

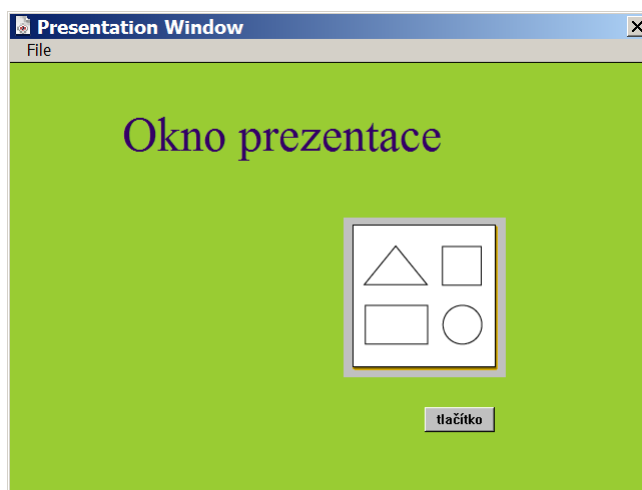
V pracovním okně (*design window*) se vytváří strukturogram aplikace.

Obrázek 1: Pracovní okno



2.3.2 Okno prezentace

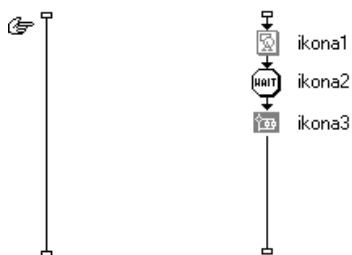
Okno prezentace (*presentation window*) má dvě funkce. Při tvorbě aplikace se v něm uspořádávají objekty na vhodná místa na obrazovce. Když je aplikace hotová, tak se v tomto okně prezentuje obsah vytvořené aplikace.



Obrázek 2: Prezentační okno

2.3.3 Časová osa

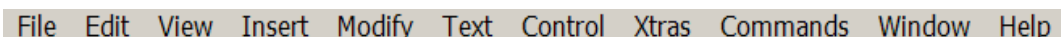
Časová osa (*flowline*) je prezentována čarou ve svislou čarou, na níž se tažením myši umisťují jednotlivé ikony. Pořadí v jakém jsou ikony na časové ose uspořádány, udává pořadí, v jakém budou spuštěny při běhu aplikace.



Obrázek 3: Časová osa

2.3.4 Lišta hlavního menu

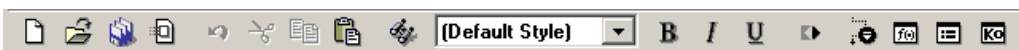
Hlavní menu je v MAW zobrazeno jako pruh tlačítek, který umožňuje ovládání všech základních funkcí programu. Hlavní menu je tvořeno jedenácti tlačítky, kterými jsou Soubor (*File*), Upravit (*Edit*), Zobrazit (*View*), Vložit (*Insert*), Změnit (*Modify*), Text (*Text*), Ovládací prvky (*Control*), Extra (*Xtras*), Příkazy (*Commands*), Okno (*Window*), Nápověda (*Help*). Po stisknutí zvoleného tlačítka, se rozbalí seznam příkazů, které jsou s ním spojeny.



Obrázek 4: Lišta hlavního menu

2.3.5 Nástrojová lišta

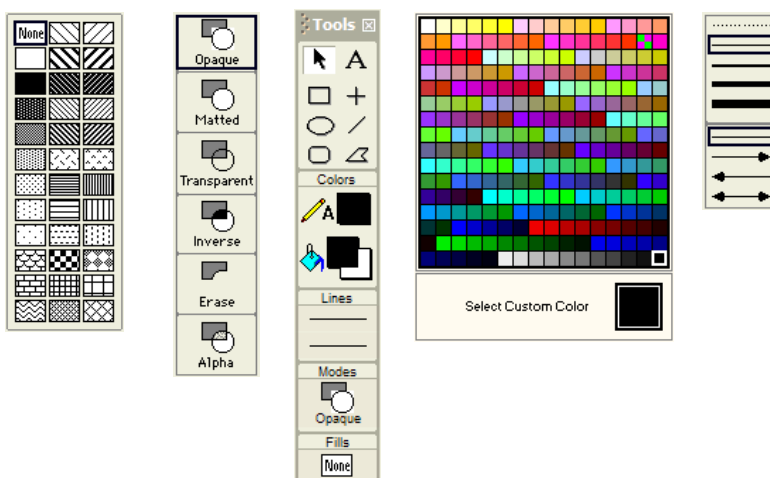
Nástrojová lišta (*toolbar*) zprostředkovává rychlý přístup k nejčastěji používaným příkazům.



Obrázek 5: Nástrojová lišta

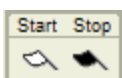
2.3.6 Panel nástrojů

Slouží k výběru, tvorbě, úpravám a nastavením grafiky a textu v okně prezentace. Panel je rozdělen na čtyři viditelné části a to na Barvy (*Colors*), pomocí které se dají měnit barvy písma a grafiky, Čáry (*Lines*), zde se mění nastavení síly a typu čar, Styly (*Modes*), které umožňují vybrat styl zobrazení objektu, a Výplně (*Fills*) objektů.



Obrázek 6: Panel nástrojů

2.3.7 Praporky start a stop



Při spuštění začíná aplikace vždy od začátku (první ikony na časové ose). To není nejvhodnější v situaci, kdy potřebujeme otestovat funkci části aplikace. Pro takovou situaci Authorware nabízí praporky (*flags*) start a stop. Prvním z nich se označí začátek a druhým konec testované části a po zahájení aplikace se spustí pouze její vymezená část.

2.3.8 Paleta barev ikon



Paleta barev ikon umožňuje obarvovat jednotlivé ikony strukturogramu různými barvami. Barevné rozlišování ikon (např. podle typu) pomáhá lepší orientaci ve strukturogramu.

2.4 Paleta ikon

Panel obsahuje čtrnáct ikon, jejichž přetahováním na časovou osu se vytváří strukturogram projektu.

2.4.1 Zobrazovací ikona (Display Icon)



Pomocí této ikony se na obrazovce zobrazuje grafika a text.

2.4.2 Pohybová ikona (Motion Icon)



Ikona slouží k přesunu objektů po určené dráze nebo ke konkrétnímu bodu.

2.4.3 Mazací ikona (Erase Icon)



Dovoluje mazat objekty z obrazovky. Je možné použít efekt při přechodu (Transition).

2.4.4 Čekací ikona (Wait Icon)



Čekací ikona umožňuje vkládat do běhu aplikace časovou prodlevu mezi zpracováním ikon ve strukturogramu.

2.4.5 Slučovací ikona (Map Icon)



Slučovací nebo též seskupovací ikona umožňuje slučování jiných ikon, což vede ke zjednodušení vytvářeného strukturogramu.

2.4.6 Filmová ikona (Digital Movie Icon)



Filmová ikona umožňuje přehrávání videa ve vytvářené aplikaci ve formátech AVI, MOV, FLC, MPEG, QuickTime2 a animací vytvořených v softwaru Macromedia Director.

2.4.7 Zvuková ikona (Sound Icon)



Zvuková ikona umožňuje doplnit vytvářenou aplikaci o zvukové efekty, hudbu nebo mluvený komentář.

2.4.8 DVD ikona (DVD Icon)



Ikona umožňuje přidávat do aplikace obsah disku DVD (obraz, zvuk, video).

2.4.9 Ikona objektu Knowledge (Knowledge object Icon)



Tato ikona dává možnost vložit do projektu objekt Knowledge. Ten může zahrnovat statické obrazy, zvuk nebo video.

2.4.10 Navigační ikona (Navigation Icon)



Pomocí této ikony se nastavuje a řídí pohyb mezi ikonami v aplikaci.

2.4.11 Stránkovací ikona (Framework Icon)



Ikona zajišťuje automatické vytvoření navigační struktury ikon, s jejichž pomocí se může uživatel pohybovat v prezentaci podle svých potřeb.

2.4.12 Rozhodovací ikona (Decision Icon)



Rozhodovací ikona umožňuje nastavovat podmínky větvení programu v závislosti na podmínkách a událostech.

2.4.13 Interakční ikona (Interaction Icon)



Interakční ikona vyhodnocuje akce uživatele a také mu dává možnost aktivně zasáhnout do chodu aplikace.

2.4.14 Kalkulační ikona (Calculation Icon)



Do kalkulační ikony je možné vkládat vlastní programový kód, upravovat hodnoty proměnných a volat funkce ovlivňující chování aplikace.

3. Proměnné

Obsah kapitoly vychází z [5],[6],[7], [8], [9] a [10] překladem anglického textu, jeho sloučením a následnou transformací a minimalizací.

3.1. Co je proměnná

Dříve než se začneme podrobněji zabývat problematikou proměnných v autorském systému Macromedia Authorware s důrazem na ty, které jsou využitelné při konstrukci didaktických aplikací, vysvětlíme si základní pojmy.

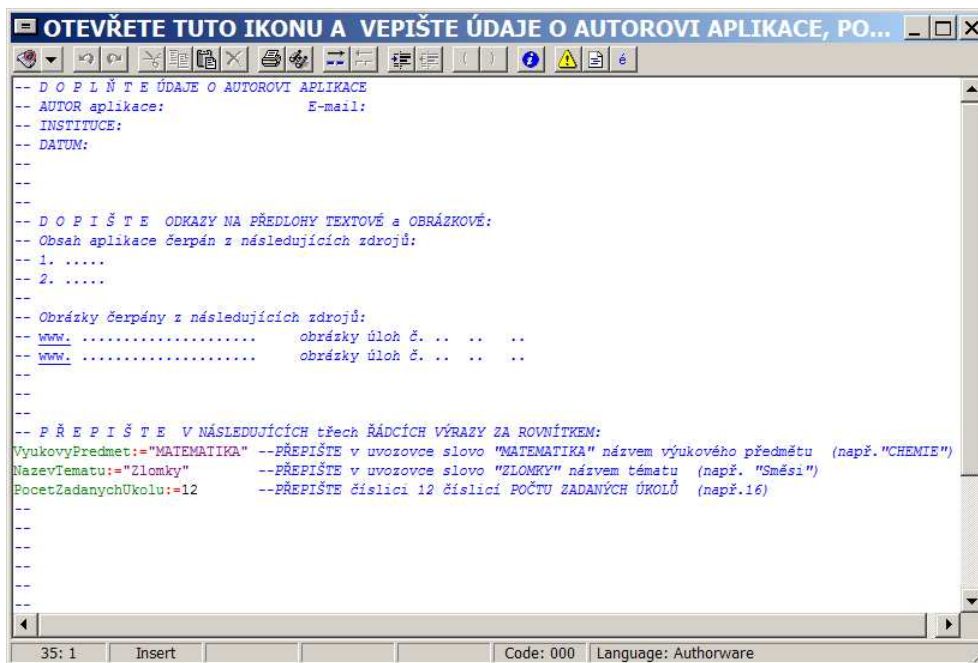
Proměnné dle [11] reprezentují hodnoty, které se mohou měnit v závislosti na situaci. Proměnné můžeme vnímat jako šuplíky, do kterých je možné ukládat nějaké údaje. Například číslo, část textu či něco jiného, co je právě potřeba si někam uložit k zapamatování.

Autorský systém Macromedia Authorware (MAW) obsahuje podle údajů [11] přes 250 předdefinovaných proměnných, které jsou ihned připraveny k použití. Například systémová proměnná `FirstTryCorrect` v sobě uchovává hodnotu celkového počtu správných odpovědí, které žák dokázal vyřešit napoprvé.



Obrázek 7: Ukázka využití systémové proměnné `FirstTryCorrect`.

Nechceme-li použít vestavěné proměnné, můžete velice snadno vytvořit proměnnou vlastní. Příkladem takové vlastní proměnné může být proměnná `VyukovyPredmet`, do které si můžeme uložit jméno předmětu v testu.



Obrázek 8: Ukázka využití uživatelské proměnné `VyukovyPredmet`.

Více se systémovým a uživatelským proměnným budeme věnovat v kapitole „3.3 Systémové a uživatelské proměnné“.

Do proměnných lze ukládat údaje nejrůznějšího charakteru - například číslo nebo text. Autorský systém Macromedia Authorware rozeznává 7 typů proměnných podle toho, jakým způsobem jsou využívány.

3.2 Typy proměnných v Macromedia Authorware

3.2.1 Číselné proměnné

Číselné proměnné obsahují čísla. Čísla mohou být buď celá (-3; 7) nebo reálná (0,006; 3,14). Rozsah číselných proměnných je od $-1,7 \cdot 10^{308}$ do $+1,7 \cdot 10^{308}$. Budeme-li s číselnými proměnnými provádět matematické operace jako je sčítání, odčítání, násobení a dělení, pak k tomu využijeme aritmetické operátory (operátorům je věnována

kapitola 4.2 Operátory). Číselné proměnné mohou obsahovat například bodové skóre nebo hodnotu matematické operace.

3.2.2 Řetězcové proměnné

Řetězcové proměnné obsahují znaky. Řetězec je sekvence jednoho nebo více znaků, tvořená písmeny („Jan Novák“), číslly („49931“), symboly („\$“), speciálními znaky (“*/“) nebo jejich kombinací („Cena celkem je \$5.00“). Maximální rozsah řetězcové proměnné je 30 000 znaků. Řetězcové proměnné mohou obsahovat například jméno a příjmení žáka, slovíčka nebo www adresu.

3.2.3 Logické proměnné

Logické proměnné nabývají hodnot True nebo False (pravda nebo nepravda). Tento typ lze přirovnat k světelnému vypínači, který přepíná mezi dvěma stavy: svítí nebo nesvítí. Číslo 0 je přiřazeno hodnotě False (nepravda) a číslo různé od nuly (obvykle 1) je True (pravda).

Použijeme-li logickou proměnnou tam, kde ji Authorware očekává (např. Conditional Response Properties dialog), vyhodnotí ji následujícím způsobem: jestliže proměnná obsahuje číselnou hodnotu, je výraz pravdivý, pokud obsahuje nenulovou hodnotu. Jestliže proměnná obsahuje řetězec, pak ji Authorware vyhodnotí jako pravdivou tehdy, jestliže řetězec uložený v proměnné je `TRUE`, `T`, `Yes`, nebo `On` (velikost písmen nehraje roli). Jakákoli jiná hodnota řetězce je vyhodnocena jako nepravda.

Logických proměnných se nejčastěji využívá při aktivaci a deaktivaci různých prvků.

Příklad:

Představme si, že máme pro žáka připravenou aplikaci s videem, potřebujeme zobrazovat tlačítko nápovědy, ale pouze ve chvílích, když se video nepřehrává. Pro tuto podmínku lze celkem snadno využít systémovou proměnnou `VideoDone`, která nabývá pravdivých hodnot, pokud se zvolené video nepřehrává. Pro vyhodnocování této proměnné ji stačí vložit do voleb tlačítka do části „*Active if True*“. Přidáním této podmínky dosáhneme toho, že tlačítko s nápovědou bude během přehrávání videa skryté.

3.2.4 Proměnné typu Seznam

Do proměnné typu Seznam se ukládají pole proměnných nebo konstant. Pole (Array) se používají v případech, kdy je vhodné v jediné proměnné soustředit více dílčích údajů, které spolu nějakým způsobem souvisí. Jednoduchým příkladem pole může být řetězec. Řetězec je v podstatě pole znaků.

Zápis:

```
JmenoPole := Array(hodnota, rozmer1 [, rozmer2, rozmer3, ...rozmer10])
```

Příklad:

Následující příklad vytvoří lineární seznam:

```
MojePole := Array(0,3)
```

Výsledkem bude tříprvkové pole [0, 0, 0].

Pro větší názornost uvedu ještě jeden příklad. Mějme pole pojmenované *Semafor*.

Potom můžeme mít v poli Semafor uložené různé hodnoty podle barev.

```
Semafor(1) := cervena;
```

```
Semafor(2) := zluta;
```

```
Semafor(3) := zelena;
```

V tomto případě je využití pole daleko přehlednější a úspornější, než zakládání tří nových proměnných.

Authorware podporuje dva typy seznamů.

1) Lineární seznam, ve kterém je každý prvek jedna hodnota.

Například: [1, 2, 3, "a", "b", "c"]

2) Seznam vlastností, ve kterém jsou prvky tvořeny vlastností a hodnotou, oddělené dvojtečkou.

Například: [#krestni:"Jan",#pri jmeni:"Novak",#tel:1248612]

Oba dva druhy seznamů mohou být abecedně tříděny.

3.2.5 Znakové proměnné

Tento typ proměnných, který je podobný řetězcovým proměnným, začíná znakem (#). Znakové proměnné Authorware dokáže zpracovat rychleji než řetězcové proměnné.

Například: znak #Alena ve výrazu:

```
MojePromenna := #Alena
```

je zpracován rychleji než řetězec “Alena“ ve výrazu:

```
MojePromenna := “Alena“
```

3.2.6 Proměnné typu Rect

Rect je typ proměnné, který je navrácen systémovou funkcí Authorwaru Rect (funkcemi se budeme podrobně zabývat ve 4. kapitole). Proměnné tohoto typu se využívají k definování pravoúhlých oblastí (slovo rect je odvozeno od slova rectangle, což v angličtině znamená pravoúhelník, obdélník). Proměnná je reprezentována pravoúhlou oblastí vymezenou body.

Například:

```
MojePromenna:=Rect(hodnota1, hodnota2, hodnota3, hodnota4) nebo
```

```
MojePromenna:=Rect(bod, bod).
```

3.2.7 Proměnné typu Point

Point je typ proměnné, který je navrácen systémovou funkcí Authorwaru Point. Proměnné tohoto typu se využívají k nastavení umístění pixelů (obrazových bodů) na obrazovce. Point (point v angličtině znamená bod) je typ proměnných, které určují bod daný souřadnicemi.

Například:

```
MujBod := Point(x, y).
```

3.3 Systémové a uživatelské proměnné

V autorském systému MAW (Macromedia Authorware) rozlišujeme dva typy proměnných, podle jejich původu. Systémové proměnné, které jsou součástí MAW a uživatelské proměnné, které si definuje uživatel sám.

Systémové proměnné

Některé systémové proměnné dovolují poukazovat na hodnotu v konkrétní ikoně v aplikaci napsáním jména proměnné, následované znakem „zavináče“ (@) a názvem ikony: `JmenoPromenne@"NazevIkony"`

Authorware aktualizuje hodnoty systémových proměnných automaticky. Názvy systémových proměnných v Authorware vždy začínají velkým písmenem a skládají se z jednoho nebo více slov bez mezer. Např. `FirstTryCorrect` a `VideoDone`.

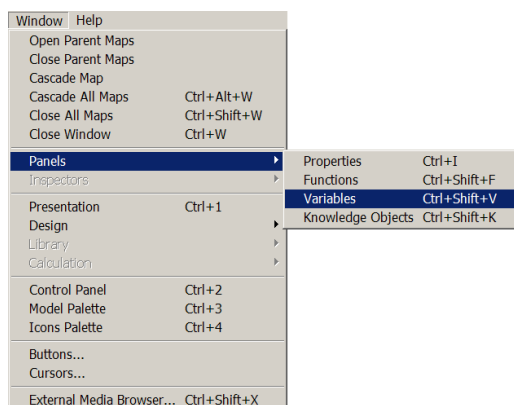
Některé proměnné se doplňují, jako např. `PercentCorrect` a `PercentWrong`, jiné proměnné spolupracují s funkcemi jako např. časové proměnné. Některým systémovým proměnným můžeme přiřadit hodnotu (např. proměnná `Movable`) a jiným ne (např. proměnná `Pi`). Přestože autorský systém MAW obsahuje velké množství systémových proměnných, můžeme občas narazit na situaci, kdy budeme potřebovat uchovat nějakou hodnotu, pro kterou ale nenalezneme odpovídající systémovou proměnnou. (např. proměnná `VyukovyPredmet` použitá v 5. kapitole). V té chvíli můžeme přistoupit k vytvoření vlastní, tzv. uživatelské proměnné.

Uživatelské proměnné

Hodnoty těchto proměnných přiřazuje a mění pouze uživatel. Například budeme chtít vědět, kolikrát Authorware prošel přes určitou strukturu, kde je proměnná `Pocitadlo`. Aktualizaci této proměnné můžeme provést následujícím způsobem: `Pocitadlo:=Pocitadlo+1`, tzn. že při každém průchodu se hodnota proměnné `Pocitadlo` zvýší o 1 (hodnota je do proměnné `Pocitadlo` přiřazena pomocí přiřazovacího operátoru `:=`). O operátorech bude pojednáno v kapitole 4.2.

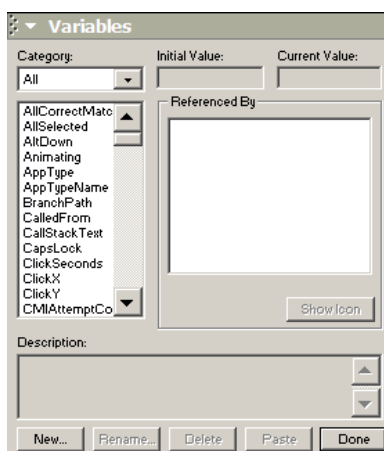
3.3.1 Vytváření uživatelských proměnných

Pokud potřebujeme pracovat s daty a informacemi, které neobstarává žádná systémová proměnná, můžeme si vytvořit proměnnou vlastní. Tyto proměnné se chovají stejně jako systémové proměnné, pouze s tím rozdílem, že se jejich hodnoty mění výhradně příkazem uživatele. Pro vytvoření nové proměnné stačí otevřít panel s proměnnými přes nabídku *Window-Panels-Variables*.



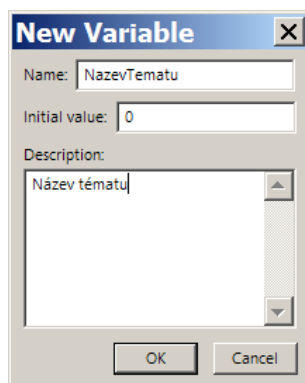
Obrázek 9: Zobrazení panelu proměnných

a v panelu Variables (proměnné) kliknout na New (nová).



Obrázek 10: Kategorie proměnných

V následujícím dialogovém okně se Authorware zeptá na název proměnné (Name), její počáteční hodnotu (Initial value) a popis (Description).



Obrázek 11: Vytváření proměnné

Vyplněním těchto položek a kliknutím na tlačítko OK vytvoříme novou proměnnou. Jméno vytvořené proměnné nesmí být delší než 40 znaků a musí být jedinečné (nelze vytvořit dvě a více proměnných se stejným názvem). Zároveň pro uživatelskou proměnnou nesmí být použit název jakékoli existující systémové proměnné. Název proměnné musí začínat písmenem a dále může obsahovat jakékoli kombinace písmen, čísel, podtržítok a mezer. Pokud není stanovena počáteční hodnota proměnné (initial value), potom je automaticky chápána jako 0. Délka popisu (description) není nijak omezena.

Uživatelské proměnné se ukládají do zvláštní skupiny proměnných (jsou oddělené kategorií systémových proměnných), která je pojmenována podle jména souboru, ve kterém vytváříme didaktickou aplikaci. Například jmenuje-li se soubor, ve kterém vytváříme aplikaci test.a7p, potom po vytvoření uživatelské proměnné vznikne v proměnných nová kategorie pojmenovaná slovem „test.a7p“.

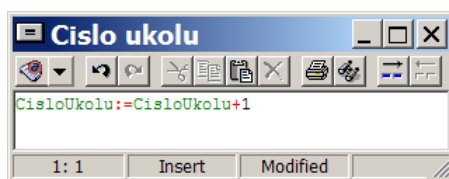
Za názvem uživatelských proměnných můžeme umístit symbol @ a název ikony. Tímto zápisem získáme hodnotu proměnné z konkrétní ikony specifikované názvem.

3.3.1.1 Používání proměnných v kalkulační ikoně

Proměnné mají velké uplatnění v kalkulační ikoně (2.4.14 kalkulační ikona). Kalkulační ikona se využívá při spouštění výrazů a skriptů, které mění hodnoty proměnných nebo provádí výpočty. Tuto ikonu můžeme vložit přímo na časovou osu strukturogramu nebo ji připojit k jiným ikonám.

Kalkulační ikonu otevřeme dvojitým poklepáním myši. Poté se zobrazí okno pro psaní kódu. V tomto okně mohou být zadávány proměnné a funkce. Můžeme například vložit

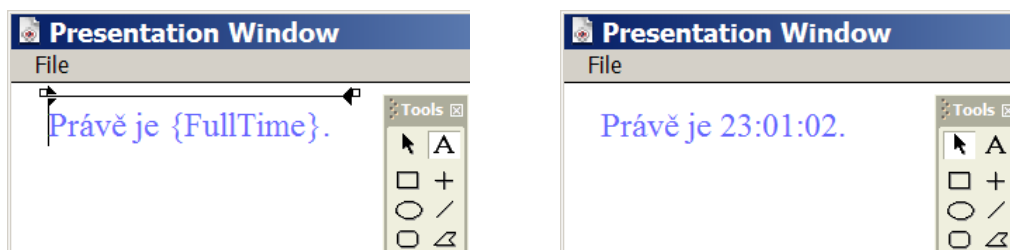
kalkulační ikonu ke správné odpovědi na otázku a aktualizovat počet správně zodpovězených otázek.



Obrázek 12: Otevřená kalkulační ikona

3.3.1.2 Zobrazování obsahu proměnných v okně prezentace

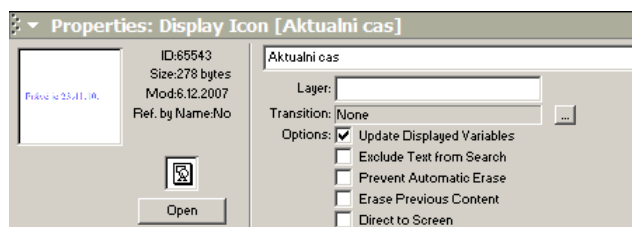
V určitých chvílích můžeme potřebovat zobrazit hodnotu proměnné v okně prezentace (presentation window). Nejčastěji to bývá ze dvou důvodů. Prvním důvodem může být „ladění“ aplikace, když definujeme novou proměnnou, a nejsme si jisti, jakou hodnotu v dané chvíli obsahuje. Druhým důvodem bývá situace, kdy chceme uživateli zobrazit nějakou důležitou informaci, kterou proměnná obsahuje (například zobrazení aktuálního času). Zobrazení aktuální hodnoty proměnné v okně prezentace je realizováno pomocí jména proměnné (např. FullTime) uzavřeného dvěma složenými závorkami („{“ a „}“). Na velikosti písmen v názvu proměnné nezáleží.



Obrázek 13: Zobrazení obsahu proměnné FullTime

Jakmile je editační kurzor (I) umístěn mimo text se jménem proměnné (na obrázku FullTime), Authorware zobrazí aktuální hodnotu proměnné (23:01:02). Pokud se hodnota proměnné nezobrazí, pak je nutné zkontrolovat, jestli je jméno proměnné uzavřeno složenými závorkami a jestli název proměnné je napsán správně. V případě, že jsme omylem zadali název neexistující proměnné, Authorware nabídne dialog pro vytvoření nové proměnné (viz 3.3.1 Vytváření uživatelských proměnných).

V některých případech budeme potřebovat zobrazovat aktualizované hodnoty proměnné. Aktualizaci zobrazovaných hodnot proměnné zajistíme zaškrtnutím volby *Update Displayed Variables* ve vlastnostech ikony.

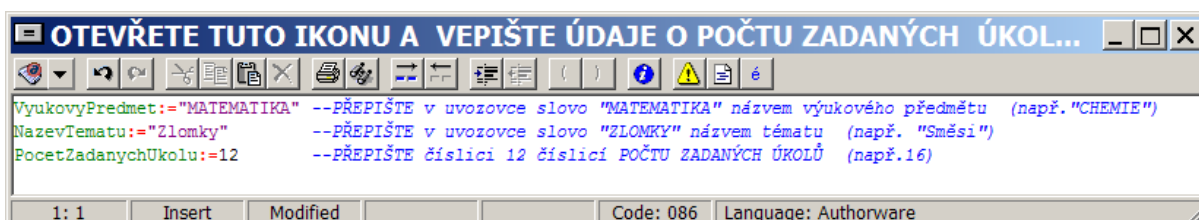


Obrázek 14: Vlastnosti ikony – Update Displayed Variables

3.3.1.3 Používání komentářů

Používání komentářů při práci s proměnnými je velice důležité, zvláště tehdy, je-li projekt (didaktická aplikace) obsáhlejší a používá více proměnných. Správné používání komentářů usnadňuje práci na větších projektech, ve více lidech. Při správně napsaném komentáři nedělá tvůrci aplikace problém se v kódu vyznat, protože všechny důležité informace může vyčíst právě z komentářů. Komentáře mohou být použity kdekoliv v okně skriptu. Komentář se skládá ze dvou pomlček (--) a textu, který následuje za nimi.

Například:



Obrázek 15: Kalkulační ikona s komentáři

Authorware ignoruje vše, co se nachází za pomlčkami (tzn. v komentáři).

3.3.2 Kategorie systémových proměnných v Macromedia Authorware

V autorském systému Macromedia Authorware se proměnné dají rozdělit nejen podle typu (číselné, řetězcové, logické atd.), podle původu (systémové, uživatelské), ale také podle kategorií, do kterých jsou v autorském systému zařazeny. V autorském systému MAW rozlišujeme 11 kategorií proměnných, podle oblasti jejich využití. Jedná se

o kategorie proměnných CMI (z anglického computer-managed instruction, což znamená použití počítače jako prostředku pro řízení vzdělávání), Decision (rozhodovací), File (souborové), Framework (stránkovací), General (obecné), Graphics (grafické), Icons (ikonové), Interaction (interakční), Network (síťové), Time (časové) a Video.

V následujících kapitolách se budeme podrobněji zabývat kategoriemi a konkrétními proměnnými v těchto kategoriích, které jsou využitelné při tvorbě didaktických aplikací.

3.3.2.1 Rozhodovací proměnné

Proměnné v této kategorii se vztahují k rozhodovací ikoně.

3.3.2.1.1 AllSelected

Typ proměnné: logická

Popis: AllSelected je pravdivá (TRUE), jestliže každá ikona připojená na rozhodovací ikonu byla vybrána.

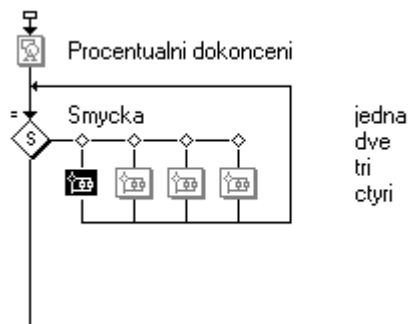
Příklad: Pomocí proměnné AllSelected můžeme určit, kde má uživatel pokračovat v zadaném úkolu při příštím spuštění aplikace. Jestliže uživatel ukončil aplikaci v rozhodovací struktuře a nebyly vybrány všechny cesty, Authorware při příštím spuštění aplikace nastaví pokračování na rozhodovací strukturu. Jestliže uživatel ukončil aplikaci a všechny cesty u rozhodovací ikony byly vybrány, potom při příštím spuštění bude Authorware pokračovat na další ikoně na časové ose.

3.3.2.1.2 PathCount

Typ proměnné: číselná

Popis: proměnná PathCount obsahuje celkový počet ikon připojených na aktuální rozhodovací ikonu. PathCount@*“NazevIkony“* obsahuje počet ikon připojených na specifikovanou rozhodovací ikonu.

Příklad: Můžeme automaticky spočítat a zobrazit, kolik z možných cest ve strukturogramu bylo vybráno. Na obrázku jsou čtyři cesty. Projití jednou cestou se započítá jako 25 procent.



Obrázek 16: Strukturogram u příkladu PathCount

Tento výpočet získáme připojením kalkulační ikony na rozhodovací ikonu a vložíme do ní následující výraz:

$$\text{CelkemProcent} := (\text{Repcount} / \text{PathCount}) * 100$$

CelkemProcent je uživatelská proměnná, která obsahuje výsledek výrazu „ $(\text{Repcount}/\text{PathCount}) * 100$ “. *RepCount* je systémová proměnná (viz 3.3.2.1.4 *RepCount*) která obsahuje počet, kolikrát Authorware spustil rozhodovací strukturu. Proměnná *PathCount* obsahuje počet ikon připojených na rozhodovací ikonu.

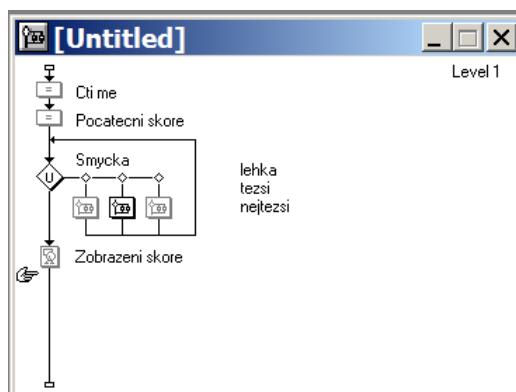
Zobrazovací ikona „*Procentualni dokonceni*“ zobrazuje vloženou proměnnou *CelkemProcent* (viz 3.3.1.2 Zobrazování obsahu proměnných v okně prezentace).

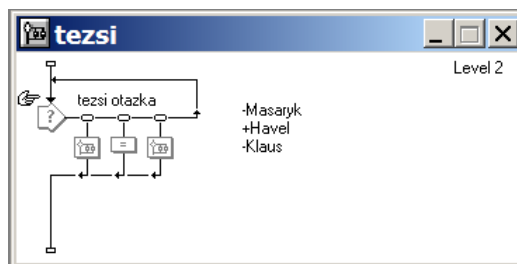
3.3.2.1.3 PathSelected

Typ proměnné: číselná

Popis: Obsahuje číslo poslední vybrané cesty v aktuální rozhodovací ikoně. *PathSelected@“NazevIkony”* obsahuje poslední vybranou cestu u specifikované rozhodovací ikony. Cesty jsou očíslované od jedné, zleva doprava.

Příklad: Správné odpovědi na otázky v procvičovacím testu mohou být bodovány podle náročnosti (nejjednodušší za 1 bod, těžší za 2 body, nejtěžší za 3 body).





Obrázek 17: Bodování otázek s použitím funkce PathSelected

Výraz v kalkulační ikoně *Pocatecni skore* definuje počáteční hodnotu proměnné skore:

```
skore := 0
```

Rozhodovací struktura *Smycka* obsahuje tři cesty, jednu pro každou otázku. Každá cesta obsahuje tři možné odpovědi. Správná odpověď je zpracována přes kalkulační ikonu, která přidá k uživatelské proměnné skore hodnotu uloženou v proměnné PathSelected (hodnota je stejná jako pořadí ikon ve strukturogramu, to znamená, že odpověď na nejtěžší otázku bude ohodnocena třemi body.

```
skore := skore + PathSelected@“Smycka“
```

Zobrazovací ikona *Zobrazeni skore* zobrazuje obsah proměnné skore uživateli.

3.3.2.1.4 RepCount

Typ proměnné: číselná

Popis: Proměnná RepCount obsahuje informaci o počtu, kolikrát Authorware opakoval rozhodovací strukturu. RepCount se vztahuje k aktuální rozhodovací ikoně a musí být použita v cyklické rozhodovací struktuře.

Příklad: viz proměnná PathCount.

3.3.2.1.5 SelectedEver

Typ proměnné: logická

Popis: SelectedEver je pravdivá (TRUE), jestliže aktuální cesta rozhodovací ikony byla předtím vybrána. SelectedEver@“NazevIkony“ je pravdivá, jestliže specifikovaná ikona připojená na rozhodující ikonu byla vybrána.

Příklad: Proměnnou `SelectedEver` můžeme použít v případě, že potřebujeme změnit výpis na obrazovku podle toho, co uživatel viděl. Poté mu můžeme nabídnout např. krátké shrnutí.

3.3.2.1.6 TimesSelected

Typ proměnné: číselná

Popis: obsahuje počet (až do 255) kolikrát byla vybrána cesta v rozhodovací ikoně. `TimesSelected@“NazevIkony“` obsahuje počet, kolikrát byla vybrána cesta na specifikované ikoně.

Příklad: Pomocí `TimesSelected` můžeme kontrolovat, kolikrát uživatel vybral určitou cestu v rozhodovací ikoně. Například je-li uživateli dvakrát položena otázka a on stále neodpověděl správně, můžeme mu poskytnout dodatečnou nápovědu.

3.3.2.2 Souborové proměnné

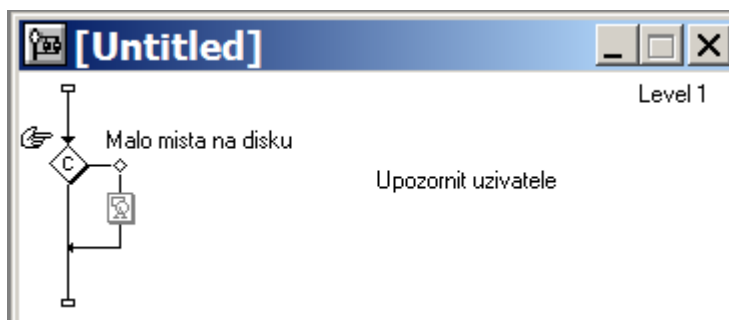
Proměnné z této kategorie obsahují informace o souborech nebo o vstupu a výstupu. Vzhledem k technické povaze těchto informací je učitel při tvorbě didaktických aplikací nevyužije příliš často. Zmíním jednu základní proměnnou, která nám dovolí sledovat velikost volného místa na disku.

3.3.2.2.1 DiskBytes

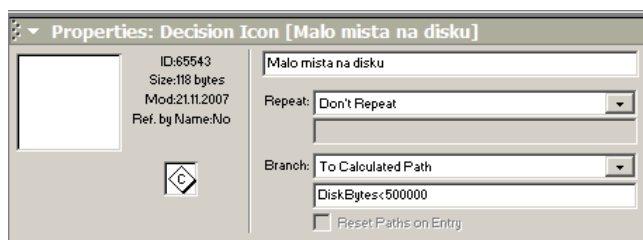
Typ proměnné: číselná

Popis: Udává velikost volného místa na disku, kde je umístěný soubor aplikace. Velikost je udávána v bytech.

Příklad: Pomocí této proměnné můžeme kontrolovat velikost volného místa na uživatelském disku a v případě nedostatku volného prostoru ho upozornit. V následující ukázce je uveden příklad, který upozorní uživatele na nedostatek volného prostoru na disku, jestliže velikost tohoto prostoru je menší než 500 KB.



Obrázek 18: Upozornění



Obrázek 19: Nastavení větvení

3.3.2.3 Stránkovací proměnné

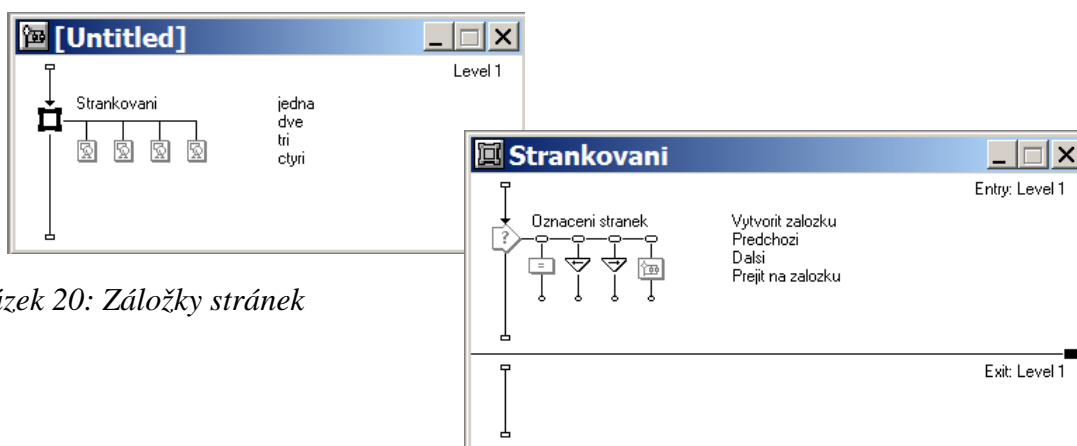
Proměnné v této kategorii obsahují informace o navigačních konstrukcích jako je identifikační číslo stránky, názvy stránek v určitých konstrukcích, počet kolikrát Authorware našel text a informace o propojení mezi ikonami, které jsou nastaveny jako volání (volání je obousměrné propojení; když Authorware dokončí operaci na volané ikoně, vrátí se zpět k ikoně, která volání spustila).

3.3.2.3.1 CurrentPageID

Typ proměnné: číselná

Popis: Obsahuje identifikační číslo ikony (ID) aktuálně zobrazené stránky. Jestliže není zobrazena žádná stránka, pak je hodnota v proměnné 0. `CurrentPageID@“NazevIkony“` obsahuje ID poslední ikony poslední stránky kterou uživatel zobrazil na specifikované stránkovací ikoně, nebo 0, pokud nebyly zobrazeny žádné stránky připojené na stránkovací ikonu.

Příklad: Můžeme uživateli povolit vytvářet záložky a později se k nim vracet. V příkladu jsou označené stránky uloženy v seznamu „zalozka“ pomocí hodnot uložených v proměnné `CurrentPageID`. Interakční ikona obsahuje čtyři kalkulační ikony pro označování stránek a přepínání mezi nimi.



Obrázek 20: Záložky stránek

Kalkulační ikona *Vytvorit zalozku* používá kód pro přidání stránky do seznamu záložek:

```
zalozka := InsertLine(zalozka, 0, CurrentPageID)
```

Když bude chtít uživatel přejít zpět na označenou stránku, použije kalkulační ikonu, která přidala poslední stránku a odebere ji ze seznamu:

```
oznac := GetLine(zalozka, 1)
```

```
zalozka := DeleteLine(zalozka, 1)
```

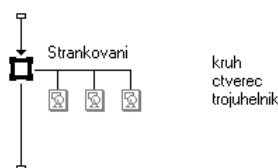
Potom použije navigační ikonu pro přeskočení na stránku identifikovanou v proměnné *oznac*.

3.3.2.3.2 CurrentPageNum

Typ proměnné: číselná

Popis: Obsahuje číslo poslední zobrazené stránky v aktuální stránkovací ikoně nebo 0, jestliže nebyla zobrazena žádná stránka. *CurrentPageNum@“NazevIkony“* obsahuje číslo poslední stránky zobrazené na specifikované ikoně nebo 0, jestliže na specifické ikoně nebyla zobrazena žádná stránka. Stránky jsou číslovány od jedné zleva doprava. Číslo stránky nejvíce vpravo je obsaženo v proměnné *PageCount* (3.3.2.3.3 *PageCount*).

Příklad: V následujícím příkladu je hodnota proměnné *PageCount* 3. *Kruh* je ve stránce 1, *ctverec* ve stránce 2 a *trojuhelnik* ve stránce 3. Když je uživatel na stránce *kruh*, potom je hodnota proměnné *CurrentPageNum* rovna jedné.



Obrázek 21: Vysvětlení počítání PageCount

Pomocí proměnné *CurrentPageNum* můžeme například znepřístupnit tlačítko pro přechod na další stránku, jestliže se momentálně nacházíme na poslední stránce prezentace. To uděláme tak, že do políčka *Aktivní když (Active if)* vložíme následující výraz:

```
CurrentPageNum<PageCount
```

3.3.2.3.3 PageCount

Typ proměnné: řetězcová

Popis: Obsahuje počet stránek připojených na aktuální (nebo poslední) stránkovací ikonu. `Pagecount@“NazevIkony“` obsahuje počet stránek připojených na ikonu specifikovanou názvem. Pokud nespecifikujeme název, potom je hodnota proměnné 0.

Příklad: viz proměnná `CurrentPageNum`.

3.3.2.4 Obecné proměnné

Proměnné v této kategorii mohou obsahovat nejrůznější informace. Tato skupina proměnných je poměrně obsáhlá, proto si ji můžeme dále rozdělit do podkategorií podle toho, jakou oblast v didaktické aplikaci proměnné v každé podkategorii obstarávají. Jedná se o skupiny proměnných:

- Multimédia
- Vstup periferií
- Konstanty
- Informace o uživateli a počítači

Multimédia

V podkategorii multimedia jsou proměnné, které souvisí s multimédií, s přehráváním filmů a zvuků.

3.3.2.4.1 MediaLength

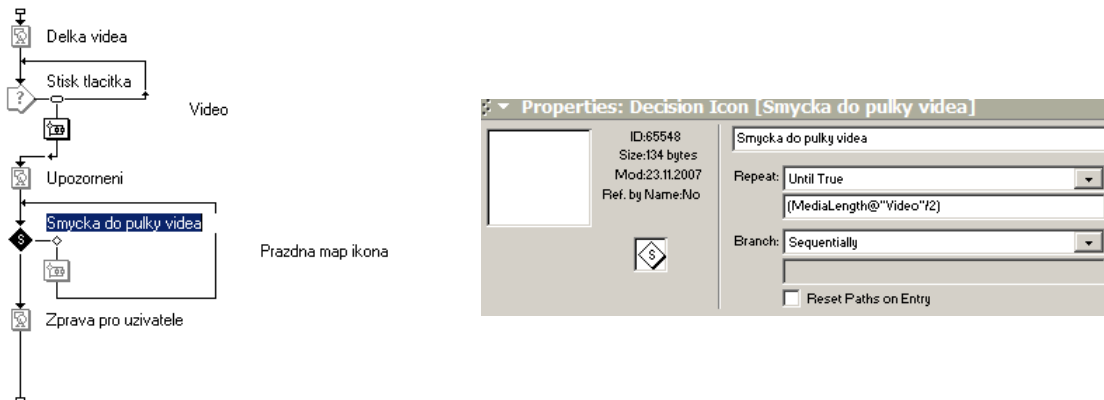
Typ proměnné: číselná

Popis: Obsahuje informaci o délce videozáznamu nebo zvukového záznamu.

Příklad: Budeme-li chtít uživateli uprostřed sledování videa sdělit nějakou důležitou informaci, můžeme k tomu využít právě proměnnou `MediaLength`. Uživatel spustí video, objeví se upozornění, že v půli videa dostane další instrukce. Rozhodovací ikona *Smyčka do pulky videa* je nastavena na opakování, dokud není pravdivý následující výraz:

```
MediaPosition@“Video“>INT (MediaLength@“Video“/2)
```


Potom uživatel obdrží zprávu v ikoně *Zprava pro uzivatele*.



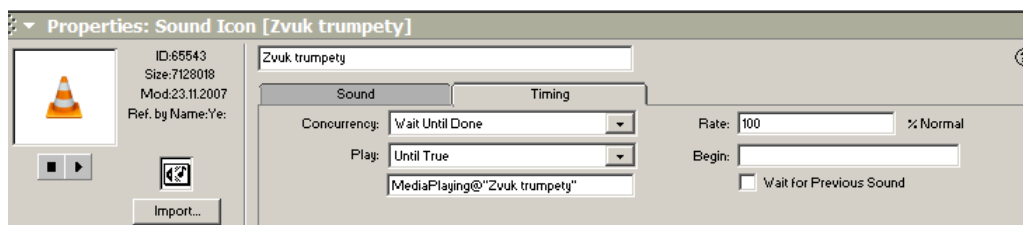
Obrázek 22: Zobrazení textu uživateli během sledování videa

3.3.2.4.2 MediaPlayering

Typ proměnné: logická

Popis: Proměnná je pravdivá (TRUE) jestliže se film, video nebo zvuk specifikovaný v `MediaPlayering@“NazevIkony“` právě přehrává nebo je pozastaven. Proměnná je nepravdivá (FALSE), jestliže se film, video nebo zvuk právě nepřehrává nebo bylo přehrávání dokončeno.

Příklad: Pomocí této proměnné můžeme synchronizovat události aktivováním nebo deaktivováním ikon, podle toho, je-li hodnota proměnné `MediaPlayering` pravdivá (TRUE) nebo nepravdivá (FALSE). Použitím ve vlastnostech zvukové ikony (*Sound Icon properties*) a nastavením na *Play: Until True* zajistíme přehrání zvuku, kdykoliv je spuštěno video nebo zvuk.



3.3.2.4.3 MediaPosition

Typ proměnné: číselná

Popis: Tato proměnná obsahuje číslo aktuálního snímku videa, které je specifikováno v `MediaPosition@“NazevIkony“`

Příklad: viz proměnná `MediaLength`.

3.3.2.4.4 SoundPlaying

Typ proměnné: logická

Popis: Proměnná je pravdivá (TRUE), jestliže je přehráván zvuk.

Příklad: Použitím proměnné `SoundPlaying` můžeme zajistit přehrávání zvuku souběžně s jinými událostmi a zároveň zajistíme, že přehrávání zvuku přestane před posunutím na další ikonu na časové ose.

Vstup periferií

Tato podkategorie obecných proměnných se vztahuje k práci se vstupy z periferií (klávesnice a myši).

3.3.2.4.5 AltDown

Typ proměnné: logická

Popis: `AltDown` je pravdivá (TRUE), když je stisknutá klávesa Alt. Klávesa Alt aktivuje menu v MAW (Macromedia Authorware) a také uživatelská menu během prezentace.

Příklad: Pomocí této proměnné můžeme reagovat na uživatelův stisk klávesy Alt.

3.3.2.4.6 CapsLock

Typ proměnné: logická

Popis: Proměnná `CapsLock` je pravdivá, jestliže je zapnuta klávesa Capslock.

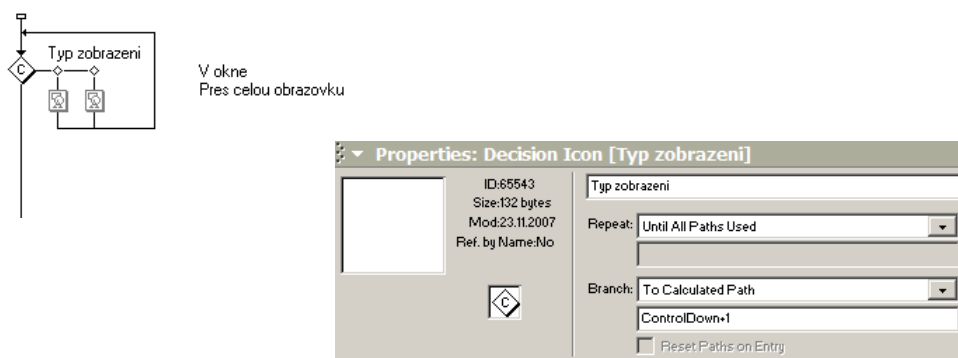
Příklad: V určitých situacích (např. při zadávání hesla) můžeme chtít upozornit uživatele, že má zapnutou klávesu Capslock, tehdy použijeme proměnnou `CapsLock`.

3.3.2.4.7 ControlDown

Typ proměnné: logická

Popis: ControlDown je pravdivá, jestliže je stisknuta klávesa Control.

Příklad: Pomocí této proměnné můžeme ovlivňovat funkce programu, jestliže uživatel stiskne klávesu Ctrl (Control). V situaci uvedené níže stiskem klávesy Ctrl určí uživatel cestu ve strukturogramu rozhodovací ikony a ovlivní tak způsob zobrazení prezentace – místo normálního zobrazení v okně se uživateli zobrazí aplikace v celoobrazovkovém módu.



Obrázek 23: Ovlivňování běhu aplikace stiskem klávesy

3.3.2.4.8 Key

Typ proměnné: řetězcová

Popis: Obsahuje název poslední klávesy stisknuté uživatelem (např. a, e, E, Insert, 3 nebo Enter).

Příklad: Níže je uveden postup, jak vytvořit uživatelské heslo ukládáním hodnot kláves stisknutých uživatelem do proměnné Key. Z bezpečnostních důvodů bude Authorware zobrazovat místo písmen hvězdičky (*).



Obrázek 24: Použití proměnné Key

Zobrazovací ikona *Zobrazení hvězdiček* obsahuje vloženou uživatelskou proměnnou {hvezdicky}. Proměnná je nastavena na automatické aktualizování (*Update Displayed Variables*). Ať uživatel vepíše jakýkoliv znak, na obrazovce se zobrazí hvězdička (*). Když uživatel stiskne klávesu, Authorware vyhodnotí kalkulační ikonu *Vložení znaku*, která obsahuje výrazy pro vytváření hesla. První řádek:

```
heslo:=heslo^Key
```

přidá všechny klávesy které uživatel stiskl do uživatelské proměnné heslo.

Druhý řádek:

```
hvezdicky:=hvezdicky^" * "
```

nahrazuje hvězdičkou každý znak vložený do uživatelské proměnné *hvezdicky*. Když stiskne uživatel klávesu Enter, provede se slučovací ikona *Enter*, která ukončí interakci. Jestliže uživatel stiskne při vkládání hesla klávesu Backspace, provede se map ikona *Backspace*, která obsahuje kalkulační ikonu, která odmaže jeden znak pokaždé, když uživatel stiskne klávesu Backspace.

Display ikona *Zobrazení hesla* obsahuje vloženou proměnnou {heslo}, která zobrazí zadané heslo.

3.3.2.4.9 MouseDown, MiddleMouseDown, RightMouseDown

Typ proměnných: logické

Popis: Proměnná `MouseDown` je pravdivá, když je stisknuté levé tlačítko myši.

Proměnná `MiddleMouseDown` je pravdivá, když je stisknuté prostřední tlačítko myši.

Proměnná `RightMouseDown` je pravdivá, když je stisknuté pravé tlačítko myši.

Příklad: Proměnnou `MouseDown` můžeme použít k přehrání zvuku v momentě, kdy uživatel stiskne levé tlačítko myši.

3.3.2.4.10 ShiftDown

Typ proměnné: logická

Popis: `ShiftDown` je pravdivá, když uživatel stiskne klávesu Shift.

Příklad: viz proměnná `ControlDown`.

Konstanty

Proměnné v této podkategorii obsahují matematické konstanty.

3.3.2.4.11 e

Typ proměnné: číselná

Popis: Obsahuje hodnotu Eulerova čísla (2.718281828459), které se používá jako základ přirozených logaritmů.

Příklad: Tuto konstantu můžeme použít při exponenciálních výpočtech:

```
Vysledek := 75 * e**3.2
```

3.3.2.4.12 Pi

Typ proměnné: číselná

Popis: Obsahuje poměr obvodu kružnice k jejímu průměru (Π - 3.1415926536).

Příklad: Tuto konstantu můžeme využít při matematických výpočtech:

```
ObvodKruznice := Pi*prumer
```

Informace o uživateli

Proměnné v této podkategorii obecných proměnných souvisejí s informacemi o uživateli a jeho počítači.

3.3.2.4.13 FirstName

Typ proměnné: řetězcová

Popis: Obsahuje uživatelovo křestní jméno uvedené v systémové proměnné `UserName` (viz dále). Jestliže byla hodnota do proměnné `UserName` vložena v pořadí příjmení, křestní jméno, potom `Authorware` uloží do proměnné `FirstName` první slovo za čárkou. První písmeno ve `FirstName` je automaticky převedeno na velké. Hodnotu proměnné `FirstName` můžeme přiřadit přímo, čímž přepíšeme hodnotu převzatou z proměnné `UserName`.

Příklad: Zobrazením proměnné `FirstName` (a tedy jména uživatele) v prezentaci můžeme uživateli zpříjemnit používání aplikace.

3.3.2.4.14 UserName

Typ proměnné: řetězcová

Popis: Přiřazením hodnoty do proměnné `UserName` můžeme uložit informaci o uživateli celým jménem. Při přiřazování hodnoty `FirstName` z proměnné `UserName` Authorware automaticky převádí první písmeno na velké.

Příklad: viz proměnná `FirstName`.

3.3.2.4.15 ScreenDepth

Typ proměnné: číselná

Popis: Obsahuje informaci o počtu bitů použitých k prezentaci každého obrazovkového bodu (pixelu) v zobrazovací paměti podle nainstalovaného ovladače grafické karty. Nejčastější hodnoty jsou:

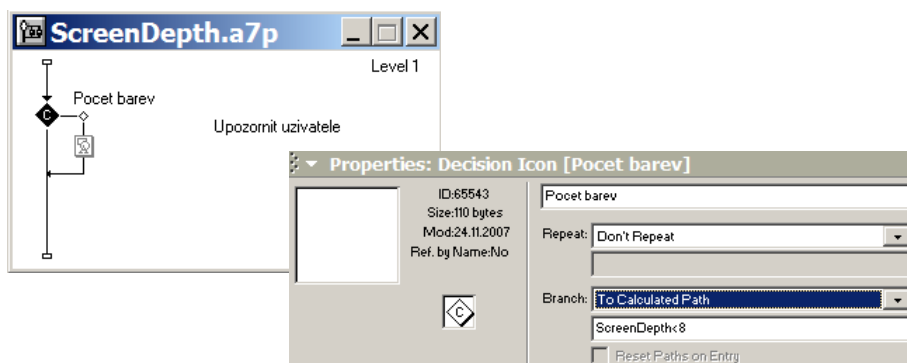
1 bit – černobílé zobrazení

4 bity – zobrazení v 16 barvách

8 bitů - zobrazení v 256 barvách

16, 24 a 32 bitů – zobrazení v tisících až miliónech barev

Příklad: Pomocí této proměnné můžeme před spuštěním aplikace zkontrolovat, jestli uživatelův počítač (a jeho grafická karta) splňuje minimální požadavky pro spuštění prezentace.



Obrázek 25: Varování uživatele při nízkém počtu barev monitoru

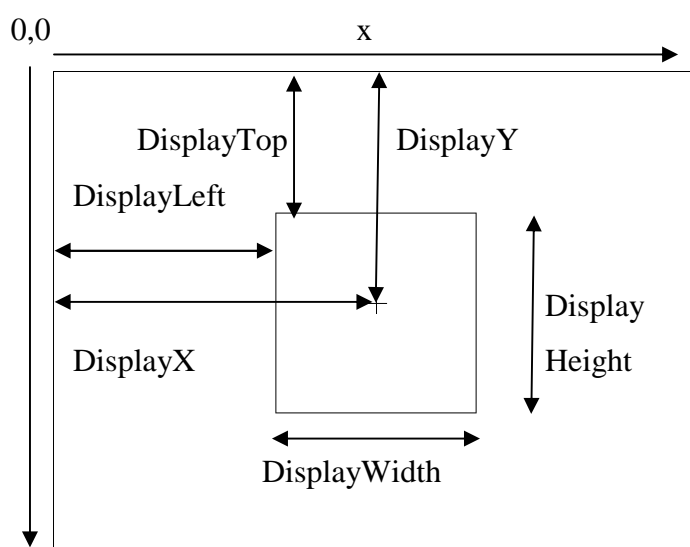
3.3.2.5 Ikonové proměnné

Ikonové proměnné obsahují informace o pozicích objektů, pohybu ikon a vztahů mezi nimi.

3.3.2.5.1 DisplayTop, DisplayLeft, DisplayX, DisplayY, DisplayWidth, DisplayHeight

Typ proměnných: číselné

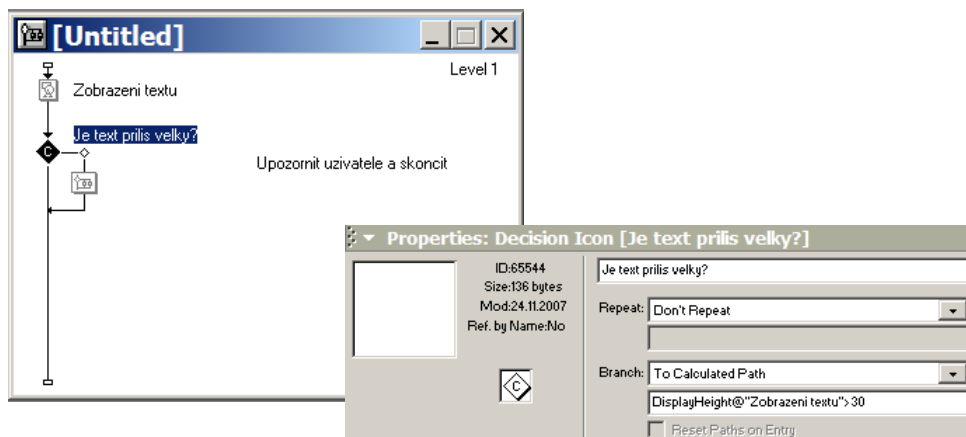
Popis: Pomocí těchto proměnných můžeme určit velikost videa nebo ukázky obsažené v ikoně specifikované názvem nebo její pozicí v okně prezentace. Hodnoty jsou udávány v pixelech (obrazkových bodech) a jsou počítány od počátku okna (souřadnice 0,0). Počátek je zpravidla vlevo nahoře. Proměnná `DisplayTop` obsahuje informaci o vzdálenosti mezi horní hranou videa/ukázky a horní hranou okna prezentace. Proměnná `DisplayLeft` obsahuje informaci o vzdálenosti mezi levou hranou videa/ukázky a levou hranou okna prezentace. Proměnná `DisplayX` obsahuje informaci o vzdálenosti mezi levou hranou okna prezentace a středem videa/ukázky. Proměnná `DisplayY` obsahuje informaci o vzdálenosti mezi horní hranou okna prezentace a středem videa/ukázky. Proměnné `DisplayWidth` a `DisplayHeight` obsahují informace o rozměrech videa/ukázky.



Obrázek 26: Orientace Display proměnných

Příklad: Ukážeme si, jak se dá využít proměnná `DisplayHeight` pro upozornění uživatele, jestliže je velikost písma příliš velká pro zobrazení v prezentaci.

V rozhodovací ikoně je nastaveno `DisplayHeight@“Zobrazeni textu”>30`.



Obrázek 27: Kontrola velikosti textu

3.3.2.5.2 Movable

Typ proměnné: logická

Popis: Proměnná `Movable@“NazevIkony”` je pravdivá (TRUE), pokud může uživatel hýbat (z anglického move – pohyb) s objektem ve specifikované ikoně. Přiřazením hodnot do této proměnné můžeme povolit nebo zakázat hýbání s objekty.

Příklad: Pomocí konstrukce s využitím této proměnné můžeme zabránit nechtěnému posunutí objektů během vytváření didaktické aplikace:

`Movable@“Pozadi” := FALSE` (uzamkne polohu pozadí prezentace)

`Movable@“ZadaniOtazky” := TRUE` (umožní přesouvat text se zadáním otázky, přičemž nehrozí nechtěné hnutí pozadím prezentace).

3.3.2.6 Interakční proměnné

Authorware automaticky sleduje a ukládá data o interakcích do interakčních proměnných. Údaje v těchto proměnných mohou být použity pro nejrůznější účely, např. k vyhodnocování výkonu uživatele během testu, personalizované zpětné vazbě, nebo k předání hodnot funkcím pro jejich další zpracování (funkcemi se budeme

zabývat ve 4. kapitole). Skupina interakčních proměnných může být dále dělena na podkategorie. Jednotlivé podkategorie Interakčních proměnných seskupují ikony podle typu informací, které uchovávají.

Podkategorie časových interakčních proměnných

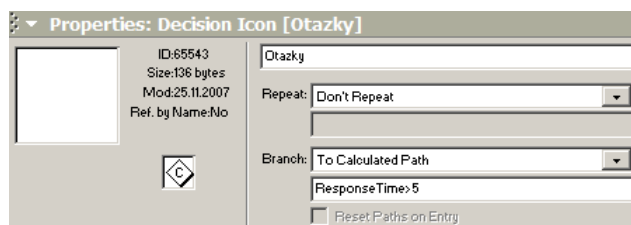
Proměnné v této podkategorii umožňují sledovat různé časové údaje.

3.3.2.6.1 ResponseTime

Typ proměnné: číselná

Popis: Obsahuje informaci o tom, jak dlouho trvala uživateli reakce (odpověď) na interakci s přesností na desetinu sekundy. Časový úsek pro první odpověď se měří od doby, kdy začala interakce, u dalších interakcí se čas měří od doby poslední reakce na předcházející interakci. Po každé interakci je hodnota proměnné nastavena na nulu (každá reakce se měří zvlášť).

Příklad: Pomocí této proměnné můžeme stanovit různé pořadí otázek (např. podle obtížnosti), podle toho, jak rychle dokázal žák správně odpovědět na předchozí otázku.



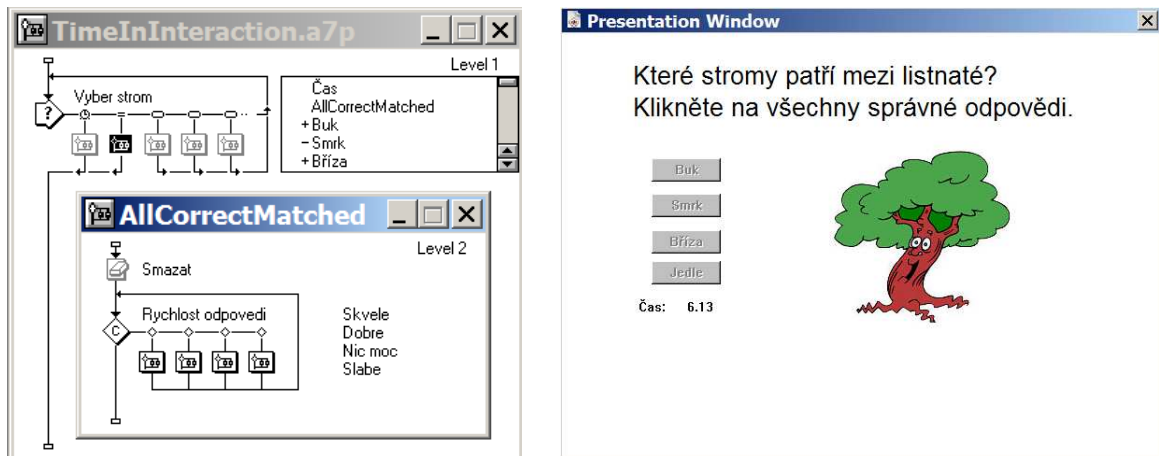
Obrázek 28: Rychlost odpovědi

3.3.2.6.2 TimeInInteraction

Typ proměnné: číselná

Popis: Obsahuje informaci o celkovém čase, který uživatel strávil poslední interakcí. Čas se počítá od začátku interakce do jejího konce a je měřen s přesností na jednu desetinu sekundy. `TimeInInteraction@“NazevIkony“` je obsahuje informaci o čase, který uživatel strávil interakcí na specifikované ikoně.

Příklad: Chceme-li ohodnotit uživatele podle rychlosti jeho správných odpovědí, pak



Obrázek 29: Ukázka použití proměnné *TimeInInteraction*

každou odpověď (v tomto případě druh stromu) připojíme na interakční ikonu. Ještě před tyto ikony připojíme slučovací ikonu (*AllCorrectMatched*) s podmínkou. V dialogovém okně asociovaném k této ikoně nastavíme *AllCorrectMatched* v políčku *Condition* a *Automatic When True*. Když je tato podmínka splněna, Authorware ukončí interakci a zobrazí obsah vložené proměnné *TimeInInteraction*. Časová osa v ikoně *AllCorrectMatched* obsahuje čtyři ikony s konstrukcí pro posuzování času, který spotřeboval žák. Potom můžeme zobrazit hodnocení podle rychlosti správných odpovědí žáka. V rozhodovací ikoně (*Rychlost odpovedi*) je v políčku *To Calculated Path* nastaveno:

`TimeInInteraction@“Vyber strom“/5+1`

Authorware potom provede příslušnou větev ve strukturogramu, která obsahuje hodnocení. Například:

Odpověděl/a jste správně za 6.13 sekund. To je velmi dobrý
výsledek!

Podkategorie textových interakčních proměnných

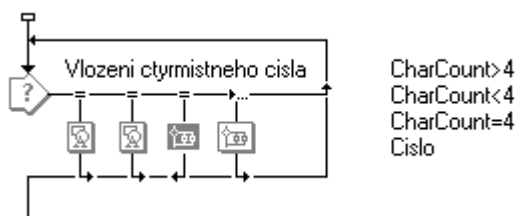
Tyto proměnné zajišťují informace o textu, který uživatel napsal během interakce.

3.3.2.6.3 CharCount

Typ proměnné: číselná

Popis: Obsahuje informaci o počtu znaků, které uživatel vepsal při textové interakci. `CharCount@“NazevIkony“` obsahuje informaci o počtu znaků pro ikonu specifikovanou názvem. Proměnná `CharCount` pracuje s textem uloženým v proměnné `EntryText` (viz 3.3.2.6.4 `EntryText`).

Příklad: Proměnnou `CharCount` můžeme použít v případě, když potřebujeme mít jistotu, že uživatel vložil požadovaný počet znaků (např. čtyřmístné číslo).



Obrázek 30: Ověřování délky řetězce

Slučovací ikona `Cislo` obsahuje políčko pro vpisování informací, do kterého uživatel vloží čtyřmístné číslo. Tři podmínkové ikony nalevo mají nastaveno políčko `When True` podmínky, které porovnávají hodnotu v proměnné `EntryText` s hodnotou 4.

`EntryText > 4`

`EntryText < 4`

`EntryText = 4`

Je-li hodnota různá od 4, `Authorware` vykoná jednu z prvních dvou větví ve strukturogramu. Jestliže je hodnota 4, `Authorware` zobrazí uživateli uvítací zprávu, která je obsažena v zobrazovací ikoně napojené na slučovací ikonu `Charcount=4`.

3.3.2.6.4 EntryText

Typ proměnné: řetězcová

Popis: Obsahuje text poslední odpovědi uživatele v poslední interakci. `EntryText@“NazevIkony“` obsahuje text poslední odpovědi uživatele pro specifikovanou ikonu.

Příklad: Tuto proměnnou používáme, pokud potřebujeme pracovat s textovou odpovědí. Mezi nejběžnější příklady tohoto typu patří získávání informací o uživateli, např. jejich jména. Jakmile uživatel zadá text, můžeme jej přiřadit do jiné proměnné:

```
UserName:=EntryText
```

3.3.2.6.5 ForceCaps

Typ proměnné: logická

Popis: Pokud je proměnná `ForceCaps` nastavena na `TRUE`, potom text, který uživatel píše je převeden na velká písmena. `ForceCaps@“NazevIkony“` převede na velká písmena text ve specifikované interakci.

Příklad: Nastavením `ForceCaps` na `TRUE` automaticky převede text na velká písmena, což může být užitečné, nechceme-li se zabývat odpověďmi uživatele napsanými malými písmeny. Velmi podobnou službu jako proměnná `ForceCaps` odvede také systémová funkce 4.3.1.1.2 `UpperCase`.

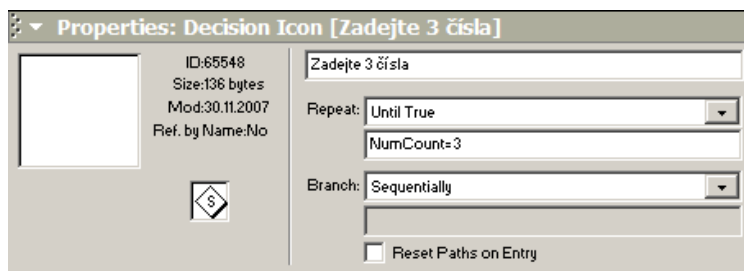
3.3.2.6.6 NumCount

Typ proměnné: číselná

Popis: Obsahuje celkový počet čísel obsažených v poslední odpovědi uživatele. `NumCount@“NazevIkony“` počítá čísla v uživatelově odpovědi na interakci specifikované názvem ikony. Proměnná `NumCount` pracuje s textem uloženým v proměnné `EntryText`.

Authorware vyhodnocuje všechny nečíselné znaky (kromě tečky) jako počátek nového čísla. Napíše-li například uživatel 32,8,634abc52, pak má proměnná `NumCount` hodnotu 4. Znaménka plus (+) a mínus (-) jsou přiřazena k číslu, které předcházejí (mezi znaménkem a číslem nesmí být mezera ani žádný jiný znak). Pokud by uživatel napsal například .3, 59,-6, potom by hodnota proměnné `NumCount` byla 3.

Příklad: Pomocí proměnné `NumCount` můžeme ověřit, jestli uživatel vložil požadovaný počet čísel.



Obrázek 31: Ověrování počtu čísel

3.3.2.6.7 NumEntry, NumEntry2, NumEntry3

Typ proměnných: číselné

Popis: Proměnná NumEntry obsahuje první číslo, které uživatel vložil v textové interakci. NumEntry@“NazevIkony“ obsahuje první číslo, které uživatel vložil do ikony specifikované názvem. Proměnné NumEntry2 a NumEntry3 obsahují druhé a třetí číslo vložené uživatelem. Za číslo Authorware považuje posloupnost numerických znaků, znaménka +, – a desetinnou tečku.

Příklad: Proměnná NumEntry se používá k získávání a ukládání číselných údajů.

3.3.2.6.8 PresetEntry

Typ proměnné: řetězcová

Popis: Jestliže je této proměnné přiřazena hodnota, pak se uživateli během prezentace zobrazí v textovém poli text obsažený v této proměnné. Uživatel může text libovolně měnit (i smazat). Text může být užitečný například pro předvyplnění nepravděpodobnější odpovědi.

Příklad: V následujícím příkladu zobrazíme do políčka, kam má uživatel napsat svoje jméno text „Sem vepiš svoje jméno“ pomocí proměnné PresetEntry. Na interakční ikonu „Prihlaseni zaka“ pomoci pravého tlačítka myši připojíme kalkulační ikonu a do ní napíšeme:

PresetEntry: = "Sem napiš své jméno".

Výsledek může vypadat například takto:



Obrázek 32: Použití proměnné PresetEntry

3.3.2.6.9 WordCount

Typ proměnné: číselná

Popis: Obsahuje počet slov obsažených v poslední odpovědi uživatele. Jako slova Authorware počítá jakékoli skupiny znaků, které neobsahují mezeru. WordCount@"NazevIkony" obsahuje počet slov v odpovědi uživatele na specifikované ikoně. Proměnná WordCount počítá slova uložená v proměnné EntryText.

Příklad: Proměnnou WordCount můžeme použít, když chceme zkontrolovat, zda uživatel zadal předepsaný počet slov. Budeme-li po uživateli požadovat zadání celého jména, můžeme zkontrolovat, jestli vložený text obsahuje minimálně dvě slova.

Podkategorie výkonových interakčních proměnných

Pomocí proměnných z této podkategorie můžeme měřit výkon uživatele.

3.3.2.6.10 AllCorrectMatched

Typ proměnné: logická

Popis: Proměnná AllCorrectMatched je pravdivá (TRUE), jestliže uživatel prošel všechny odpovědi připojené na interakční ikonu, které měly nastaveny status „Correct“ (Správně). AllCorrectMatched@“NazevIkony“ poskytuje hodnotu proměnné na specifikované ikoně.

Příklad: Pomocí této proměnné můžeme uživateli sdělit, jestli zvolil všechny správné odpovědi. Na základě hodnoty této proměnné se také můžeme rozhodnout, zda mají být uživateli položeny další otázky podle toho, zvládl-li odpovědět na všechny otázky správně.

3.3.2.6.11 ChoiceCount

Typ proměnné: číselná

Popis: Obsahuje informaci o celkovém počtu odpovědí, které jsou připojeny na aktuální interakci. ChoiceCount@“NazevIkony“ obsahuje informaci o celkovém počtu cílených odpovědí připojených na specifickou interakci.

Příklad: Pomocí této proměnné můžeme uživateli poskytnout informaci, kolik možností má na výběr. Například obsahuje-li prezentace šest tlačítek, ze kterých si může uživatel vybrat, necháme zobrazit následující zprávu:

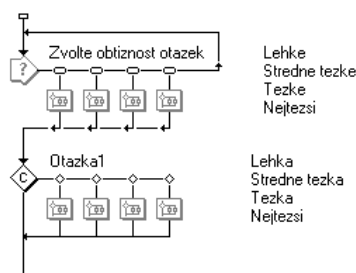
Vyberte jednu z {ChoiceCount} odpovědí. (viz 3.3.1.2 Zobrazování obsahu proměnných v okně prezentace).

3.3.2.6.12 ChoiceNumber

Typ proměnné: číselná

Popis: Obsahuje informaci o poslední odpovědi zvolené uživatelem v aktuální interakci. ChoiceNumber@“NazevIkony“ obsahuje informaci o poslední odpovědi ve specifikované interakci. Ikony odpovědí jsou číslovány od jedné zleva doprava.

Příklad: Podle toho, jakou odpověď uživatel v interakci zvolil a s pomocí čísla odpovědi, můžeme zvolit určitou cestu ve strukturogramu. Na začátek prezentace umístíme rozcestník, kde se uživatele zeptáme, jak těžké otázky chce pokládat.



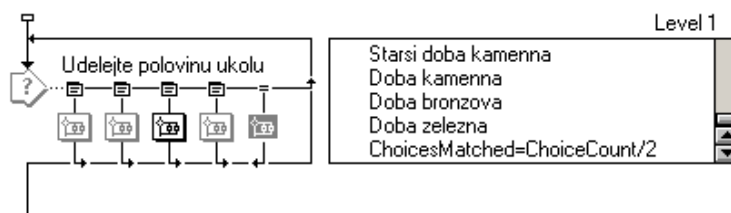
Obrázek 33: Rozcestník otázek

3.3.2.6.13 ChoicesMatched

Typ proměnné: číselná

Popis: V této proměnné je uložena informace o počtu různých odpovědí, které uživatel zvolil v aktuální interakci. V ChoicesMatched@“NazevIkony“ je uložena informace o počtu různých odpovědí, které uživatel zvolil ve specifikované interakci.

Příklad: V následujícím příkladu musí uživatel udělat polovinu úkolů v prezentaci, potom se interakce ukončí.



Obrázek 34: Podmínka pro polovinu úkolů

3.3.2.6.14 Correct

Typ proměnné: řetězcová

Popis: Obsahuje název první odpovědi, která má status *Correct* (Správně, je vyznačena znaménkem +). Correct@“NazevIkony“ obsahuje název první správné odpovědi ve specifikované interakci.

Příklad: V interakci s jedinou správnou textovou odpovědí připojíme správnou odpověď na první ikonu v interakci a v případě, že uživatel neodpoví správně na

stanovený počet pokusů, můžeme mu nechat zobrazit správnou odpověď vložením správné odpovědi do textu.

3.3.2.6.15 CorrectChoicesMatched, WrongChoicesMatched

Typ proměnných: číselné

Popis: `CorrectChoicesMatched` obsahuje počet správných odpovědí (odpovědi se statusem *Correct*), které uživatel zadal během interakce. `WrongChoicesMatched` obsahuje počet chybných odpovědí (odpovědi se statusem *Wrong*), které uživatel zadal během interakce. `CorrectChoicesMatched@“Nazevikony“` a `WrongChoicesMatched@“Nazevikony“` obsahují výše zmíněné číselné údaje pro specifikované ikony.

Příklad: Tyto proměnné slouží ke shrnutí uživatelova výkonu během testu. Sledováním hodnot uložených v proměnných můžeme určit, je-li test validní. Pokud uživatelé často odpovídají chybně, potom můžeme přemýšlet o tom, jestli uživatelé nepotřebují více nápověd nebo jestli bychom neměli test změnit (ať už otázky samotné nebo třeba jen vzhled prezentace).

3.3.2.6.16 FirstTryCorrect, FirstTryWrong

Typ proměnných: číselné

Popis: Proměnná `FirstTimeCorrect` obsahuje údaj o celkovém počtu odpovědí, které uživatel zvládl zodpovědět správně napoprvé. `FirstTimeWrong` obsahuje údaj o celkovém počtu odpovědí, které uživatel napoprvé zodpověděl chybně. Tyto proměnné se používají k vypočítávání celkového výkonu uživatele v testu (didaktické aplikaci). Odpoví-li uživatel správně napoprvé, je hodnota proměnné `FirstTimeCorrect` zvětšena o 1. Podobně, v případě, že uživatelova odpověď při prvním pokusu není správná, je zvětšena hodnota proměnné `FirstTimeWrong` o 1.

Příklad: Pomocí těchto proměnných můžeme uživateli zobrazit jeho procentuální úspěšnost prvních odpovědí v testu. K tomu budeme ještě potřebovat proměnnou `JudgedResponses` (proměnnou, která obsahuje informaci o počtu vyhodnocovaných odpovědí, viz 3.3.2.6.18 `JudgedResponses`). Procentuální vyjádření výkonu můžeme získat tak, že počet otázek zodpovězených správně napoprvé

(FirstTimeCorrect) respektive chybně napoprvé (FirstTimeWrong) vydělíme celkovým počtem hodnocených odpovědí a tuto hodnotu vynásobíme číslem 100.



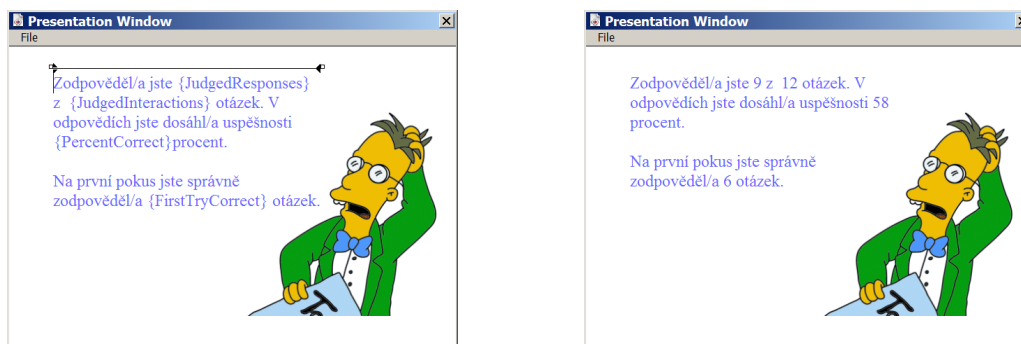
Obrázek 35: Procentuální vyjádření výkonu

3.3.2.6.17 JudgedInteractions

Typ proměnné: číselná

Popis: Obsahuje informaci o počtu vyhodnocovaných odpovědí, na které uživatel narazil během práce v didaktické aplikaci. Vyhodnocovanou odpovědí se rozumí odpověď, která má nastaven status na *Correct* (Správně) nebo *Wrong* (Chybně). Pokaždé, když uživatel během interakce narazí na vyhodnocovanou otázku (otázku s nastaveným vyhodnocením *Correct/Wrong*), se hodnota proměnné JudgedInteractions zvýší o 1 (v případě, že uživatel odpovídal na některou otázku několikrát, hodnota JudgedInteractions pokaždé odpovědi se zvýší o 1).

Příklad: Uživateli můžeme na konci testu zobrazit informace o jeho výkonu. Hodnocení může vypadat například takto:



Obrázek 36: Ukázka použití proměnných během vyhodnocování

3.3.2.6.18 JudgedResponses

Typ proměnné: číselná

Popis: Obsahuje informaci o celkovém počtu odpovědí uživatele na otázky s nastaveným statusem odpovědi (*Correct* nebo *Wrong*). Pokaždé, když uživatel odpoví na takto vyhodnocovanou otázku, hodnota proměnné `JudgedResponses` se zvýší o 1.

Příklad: viz proměnná `JudgedInteractions`.

3.3.2.6.19 `PercentCorrect`, `PercentWrong`

Typ proměnných: číselné

Popis: Proměnná `PercentCorrect` obsahuje informaci o procentuálním vyjádření odpovědí (viz proměnná `JudgedResponses`), které byly vyhodnoceny jako správné. Proměnná `PercentWrong` obsahuje informaci o procentuálním vyjádření odpovědí, které byly vyhodnoceny jako chybné.

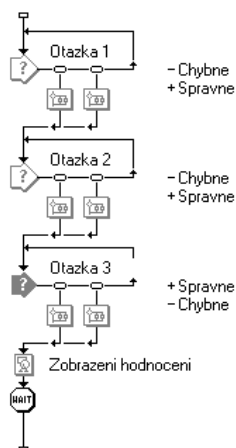
Příklad: viz proměnná `JudgedInteractions`.

3.3.2.6.20 `TotalCorrect`, `TotalWrong`

Typ proměnných: číselné

Popis: Proměnná `TotalCorrect` obsahuje informaci o celkovém počtu správných odpovědí, které uživatel zadal při práci s didaktickou aplikací. Proměnná `TotalWrong` obsahuje informaci o celkovém počtu chybných odpovědí, které uživatel zadal při práci s didaktickou aplikací. Na rozdíl od proměnných `FirstTryCorrect` a `FirstTryWrong` proměnné `TotalCorrect` a `TotalWrong` obsahují celkový počet správných a chybných odpovědí, tedy včetně těch, které nebyly zodpovězeny správně nebo chybně napoprvé. Hodnota proměnné `TotalCorrect` se zvýší o 1 pokaždé, když uživatel odpoví správně, stejně tak hodnota proměnné `TotalWrong` se zvýší o 1, když uživatel zadá chybnou odpověď.

Příklad: Pomocí těchto proměnných můžeme automaticky sledovat výkony uživatelů během testu a pak zobrazit závěrečné hodnocení na základě údajů získaných z těchto proměnných. Na strukturogramu je uveden příklad, kde jsou v didaktické aplikaci položeny 3 otázky. U každé jsou možné dvě odpovědi, jedna správná (označená znaménkem +) a jedna chybná (označená znaménkem -)



Obrázek 37: Užití proměnné *TotalCorrect*

Když Authorware vyhodnotí odpověď na poslední otázku (*Otazka 3*), zobrazí uživateli hodnocení připravené v zobrazovací ikoně *Zobrazení hodnoceni*. To může vypadat následovně:

Z celkového počtu {JudgedInteractions} otázek jste zodpověděl/a {TotalCorrect} otázek správně.

3.3.2.7 Časové proměnné

Časové proměnné obsahují informace o aktuálním datu a času v různých formátech.

3.3.2.7.1 Date

Typ proměnné: číselná

Popis: Tato proměnná obsahuje aktuální datum v číselném formátu (28.11.2007).

Formát zápisu data je automaticky nastaven podle nastavení počítače uživatele.

Příklad: Pomocí této proměnné můžeme v aplikaci zobrazit aktuální datum.

Dnes je {Date}. Dnes je 28.11.2007.

Obrázek 38: Užití proměnné *Date*

3.3.2.7.2 Day

Typ proměnné: číselná

Popis: Proměnná *Day* obsahuje číslo aktuálního dne v měsíci od 1 do 31.

Příklad: Spolu se systémovou funkcí `DateToNum` můžeme proměnnou `Day` použít pro výpočty nejruznějších datumů (minulých i budoucích).

3.3.2.7.3 FullDate

Typ proměnné: řetězcová

Popis: Proměnná `FullDate` obsahuje aktuální datum, které obsahuje den, měsíc (názvy měsíců jsou bohužel v angličtině, což víceméně předurčuje použití této proměnné v didaktických aplikacích zaměřených na anglický jazyk) a rok.

Příklad: Pomocí této funkce můžeme nechat uživateli zobrazit aktuální datum s názvem měsíce v angličtině.



```
[Today is {FullDate}]
```

 Today is 5. December 2007

Obrázek 39: Použití funkce `FullDate`

3.3.2.7.4 Year

Typ proměnné: číselná

Popis: Obsahuje číslo aktuálního roku.

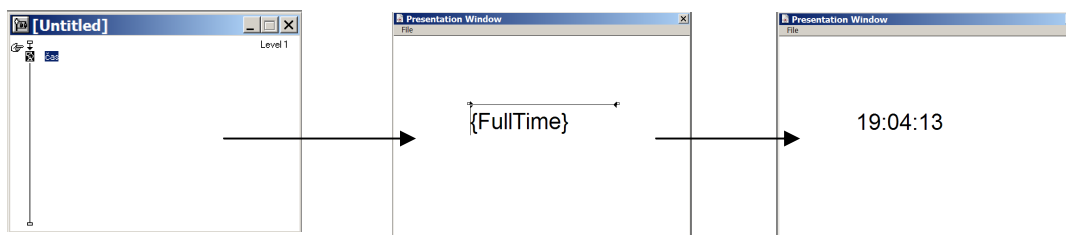
Příklad: Tuto proměnnou můžeme použít v případě, budeme-li chtít zobrazit pouze rok, nebo ji můžeme použít spolu se systémovou funkcí `DateToNum` pro výpočet nejruznějších datumů.

3.3.2.7.5 FullTime

Typ proměnné: řetězcová

Popis: Proměnná `FullTime` obsahuje aktuální čas ve dlouhém tvaru, který zahrnuje hodiny, minuty a sekundy. Formát proměnné je určený nastavením počítače uživatele.

Příklad: Pro zobrazení aktuálního času můžeme použít proměnnou `FullTime`, když zároveň nastavíme *Update Displayed Variables* v dialogovém okně *Icon Properties*.



Obrázek 40: Do display ikony „čas“ vepíšeme proměnnou *FullTime* a v prezentačním okně se zobrazí aktuální čas .

3.3.2.7.6 Hour

Typ proměnné: číselná

Popis: Proměnná *Hour* obsahuje číslo aktuální hodiny od 0 do 23.

Příklad: Pomocí této proměnné můžeme zobrazit samotné hodiny.

3.3.2.7.7 Minute

Typ proměnné: číselná

Popis: Proměnná *Minute* obsahuje číslo aktuální minuty v hodině od 0 do 59.

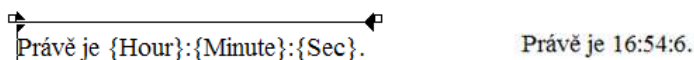
Příklad: Pomocí této proměnné můžeme zobrazit pouze minuty.

3.3.2.7.8 Sec

Typ proměnné: číselná

Popis: Proměnná *Sec* obsahuje číslo aktuální sekundy v minutě od 0 do 59.

Příklad: Pomocí této proměnné můžeme zobrazit pouze sekundy. Na následujícím obrázku je ukázáno použití proměnných *Hour*, *Minute* a *Sec* k zobrazení aktuálního času.



Obrázek 41: Využití 3 proměnných namísto jedné

Výsledek použití těchto tří proměnných je prakticky stejný, jako kdybychom použili proměnnou *FullTime*.

3.3.2.7.9 Time

Typ proměnné: číselná

Popis: Proměnná *Time* obsahuje aktuální čas v hodinách a minutách. Formát času je dán nastavením počítače uživatele.

Příklad: Pomocí této proměnné můžeme zobrazit aktuální čas v hodinách a minutách.

3.3.2.8 Video proměnné

Proměnné patřící do této kategorie se souvisejí s aktuálním stavem videa/filmu.

3.3.2.8.1 VideoDone

Typ proměnné: logická

Popis: Proměnná VideoDone je pravdivá (TRUE), je-li dokončeno přehrávání aktuální videosekvence.

Příklad: Pomocí této proměnné můžeme synchronizovat přehrávání videa s jinými událostmi, jako je například přehrávání zvuku, což můžeme udělat pomocí vložení podmínky *Until TRUE: VideoDone* do políčka *Play*.



Obrázek 42: Podmínka u příkladu k proměnné VideoDone

4. Funkce

Obsah kapitoly vychází z [5],[6],[7], [8], [9] a [10] překladem anglického textu, jeho sloučením a následnou transformací a minimalizací.

4.1 Co je funkce

Funkce je část programu, kterou je možné opakovaně volat z různých míst kódu. Funkce provádí akce s daty. Například funkce `SQRT` vypočítává druhou odmocninu ze zadaného čísla (viz funkce `SQRT`) a funkce `Capitalize` převádí první znak zadaného řetězce nebo proměnné na velké písmeno (viz funkce `Capitalize`). Některé funkce mohou aktivovat speciální volby, jako je například předběžné nahrání grafiky, zvuku nebo videa do paměti počítače pro rychlejší přístup k těmto datům (viz funkce `Preload`).

Funkce může mít parametry (též argumenty) – údaje, které jí jsou předávány při volání – a návratovou hodnotu, kterou naopak vrací. Parametr funkce je označení pro vstupní data funkce. Většina systémových funkcí v Authorwaru vyžaduje parametry (argumenty), pomocí nichž se předávají informace pro další výpočty. Parametry se píší za název funkce do okrouhlých závorek. Například pokud použijeme funkci `SQRT`, musíme jí předat číslo, ze kterého chceme druhou odmocninu získat (např. `SQRT(64)`). U funkcí rozlišujeme dva druhy parametrů: povinné a nepovinné. Nepovinné parametry se zpravidla píší za povinné parametry a jsou v popisu použití naznačeny uzavřením v hranatých závorkách [].

Názvy systémových funkcí (viz 4.3 Systémové a uživatelské funkce) v autorském systému Macromedia Authorware jsou tvořeny jedním nebo více slovy bez mezer, přičemž každé slovo začíná velkým písmenem (např. funkce 4.3.1.3.4 `PrintScreen`).

4.2 Operátory

Pro získání výsledků s použitím funkcí a proměnných často potřebujeme využít výrazů. Výraz je příkaz, jehož realizací vzniká výsledek provedením výpočtu nebo provedením operace. Spolu s funkcemi a proměnnými se ve výrazech používají operátory, čísla, řetězce, konstanty a komentáře.

Například ve výrazu zmíněném v kapitole 3.3 se hodnota proměnné `Pocitadlo` zvětšuje o 1 (`Pocitadlo := Pocitadlo + 1`).

Aby Authorware věděl, co má s funkcemi a proměnnými dělat, používáme operátory.

Typy operátorů

Operátor je symbol, který určuje akci, která se má provést s jednou nebo více hodnotami (v konstantách, proměnných nebo funkcích). Souhrn operátorů udává tabulka 1 v [11].

Druh operátoru	Operátor	Význam
Přiřazovací operátor	<code>:=</code>	Uvádí se napravo od proměnné. Přiřazuje hodnotu na pravé straně do proměnné na levé straně.
Relační operátory	<code>=</code> <code><></code> <code><</code> <code>></code> <code><=</code> <code>>=</code>	Rovná se Nerovná se Menší než Větší než Je menší nebo rovno Je větší nebo rovno
Logické operátory	<code>~</code> <code>&</code> <code> </code>	Not And Or
Aritmetické operátory	<code>+</code> <code>-</code> <code>/</code> <code>*</code> <code>**</code>	Součet Rozdíl Podíl Součin Exponent
Operátor zřetězení	<code>^</code>	Spojí dva řetězce v jeden

Tabulka operátorů podle manuálu MAW.

4.3 Systémové a uživatelské funkce

Podobně jako u proměnných můžeme funkce rozdělit na funkce systémové a funkce uživatelské. Z důvodu složitosti problematiky vytváření uživatelských funkcí se budeme zabývat pouze funkcemi systémovými, které jsou dostatečně silným nástrojem pro tvorbu didaktických aplikací v MAW. Systémové funkce lze použít pro manipulaci s textem, soubory, k přehrávání a synchronizaci médií, přepínání mezi soubory, k vykonávání matematických výpočtů a k řadě dalších úkolů. Některé funkce se doplňují (např. funkce `WriteExtFile` a `ReadExtFile`), jiné spolupracují s proměnnými (např. časová funkce `DateToNum`) a některé provádí určité operace (např. funkce `Random`). Názvy systémových funkcí zachovávají konvence pojmenování nastvené u proměnných, tzn. začínají velkým písmenem a obsahují jedno a více slov bez mezer (např. `Beep` a `SetCursor`). Panel s funkcemi lze otevřít projitím menu *Window-Panels-Functions*. Podobně jako systémové proměnné, lze i systémové funkce rozdělit do kategorií podle toho, s jakými informacemi pracují. V následujících kapitolách se budeme zabývat řetězcovými, souborovými, obecnými, ikonovými, přeskakovacími/přepínacími, matematickými, časovými a video funkcemi.

4.3.1 Kategorie systémových funkcí

4.3.1.1 Řetězcové funkce

Funkce v této kategorii pracují s řetězcem, ať už přímo či s řetězcem uloženými v proměnných.

4.3.1.1.1 Capitalize

Použití: `VyslednyRetezec := Capitalize(„řetězec“)`

`VyslednyRetezec := Capitalize(„řetězec“;[1])`

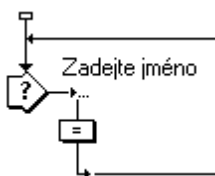
Popis: Tato funkce převádí první písmeno každého slova v řetězci zadaném parametrem na velké. Původní řetězec zůstává beze změny, nový řetězec je předán zpět, aby mohl být použit v proměnné nebo prezentačním okně. Jestliže je zadán i nepovinný parametr (1), pak je na velké písmeno převedeno pouze první písmeno v prvním slově řetězce.

Parametry:

řetězec – název řetězce, ve kterém mají být první písmena slov převedena na velká.

l – nepovinný parametr – přikazuje převod na velké písmeno pouze v prvním slově v řetězci.

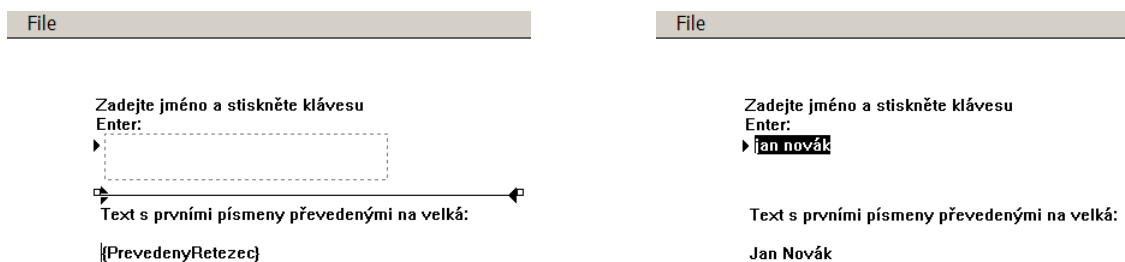
Příklad: Pomocí této funkce můžeme převést první písmena ve slovech na velká. To může být výhodné, chceme-li například po uživateli zadat jeho jméno. Funkce Capitalize zajistí, že omylem napsaný text „jan novák“ se uloží ve správném formátu s velkými písmeny („Jan Novák“).



Obrázek 43: Užití funkce Capitalize v kalkulačnické ikoně

Kalkulačnická ikona obsahuje výraz:

```
PrevedenyRetezec := Capitalize(EntryText)
```



Obrázek 44: Výstup jak jej vidí uživatel

Ukázka použití funkce Capitalize je k dispozici na přiloženém CD v adresáři Funkce\Capitalize.a7p

4.3.1.1.2 LowerCase, UpperCase

Použití: `VyslednyRetezec := LowerCase(„řetězec“)`

`VyslednyRetezec := UpperCase(„řetězec“)`

Popis: Funkce LowerCase převádí řetězec zadaný parametrem na malá písmena. Funkce UpperCase převádí řetězec zadaný parametrem na velká písmena.

Parametry: *řetězec* – název řetězce, ve kterém mají být písmena převedena na malá (LowerCase) nebo velká (UpperCase) písmena.

Příklad: Pomocí funkce UpperCase můžeme text vložený uživatelem převést na velká písmena a nechat ho zobrazit jako nadpis druhé obrazovky.

Ukázka použití funkce UpperCase je k dispozici na příloženém CD v adresáři Funkce\UpperCase.a7p

4.3.1.1.3 CharCount

Použití: *Cislo* := CharCount(„řetězec“)

Popis: Tato funkce počítá znaky (včetně mezer) obsažené v řetězci zadaném parametrem.

Parametry: *řetězec* – název řetězce, ve kterém mají být spočítány znaky.

Příklad: V případě, že má uživatel zadat předepsaný počet znaků (např. šestimístné heslo), můžeme použít funkci CharCount.

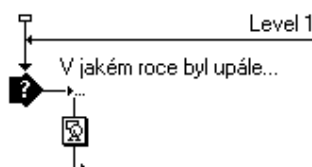
4.3.1.1.4 NumCount

Použití: *Cislo* := NumCount(„řetězec“)

Popis: Funkce NumCount počítá čísla obsažená v řetězci zadaném parametrem. Jako konec čísla Authorware chápe jakýkoliv nenumernický znak kromě tečky, která slouží k zápisu desetinných čísel.

Parametry: *řetězec* – název řetězce, ve kterém mají být spočítána čísla.

Příklad: Pomocí této funkce můžeme ověřit, jestli uživatel zadal alespoň jedno číslo. Takovéto ověření je užitečné zejména ve chvíli, kdy na vstupu z klávesnice číslo očekáváme jako odpověď (například na otázku „V jakém roce byl upálen Jan Hus?“).

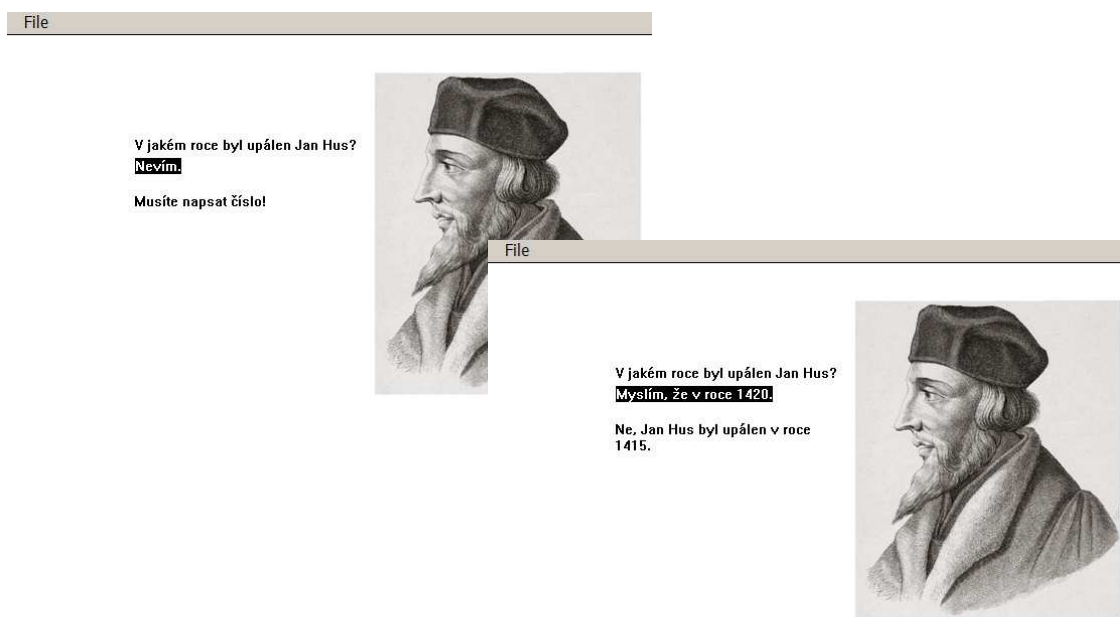


Obrázek 45: Využití Funkce NumCount

Zobrazovací ikona obsahuje následující výraz, který zajistí správné vyhodnocení podmínky:

```
{Test ( NumCount ( EntryText ) = 0; "Musíte napsat číslo!";
Test ( GetNumber ( 1; EntryText ) = 1415; "Ano, správně!"; "Ne, Jan Hus byl
upálen v roce 1415."))}
```

Funkce NumCount je v tomto případě použita s dalšími funkcemi Test a GetNumber, kterými se budeme zabývat později. Níže jsou zobrazeny odpovědi aplikace tak, jak je uvidí uživatel.



Obrázek 46: Vyhodnocení odpovědí

Ukázka použití funkce NumCount je k dispozici na přiloženém CD v adresáři Funkce\NumCount.a7p

4.3.1.1.5 WordCount

Použití: *Cislo* := WordCount („řetězec“)

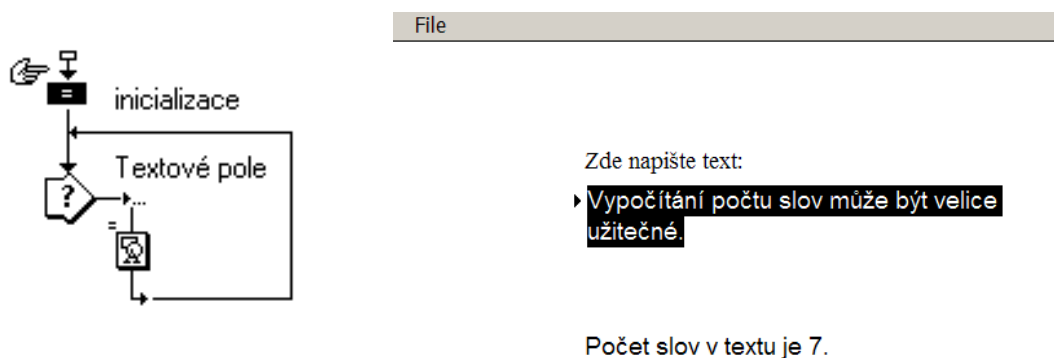
Popis: Funkce WordCount počítá slova obsažená v řetězci zadaném parametrem.

Parametry: *řetězec*, ve kterém mají být spočítána slova. Slovem se rozumí jakýkoli řetězec znaků oddělený mezerou nebo tečkou.

Příklad: Občas můžeme chtít spočítat počet slov, která uživatel zadal (např. při zadání úlohy „*napište o sobě 50 slov v angličtině*“). Pro tento případ použijeme právě funkci WordCount.

V příkladu uvedeném níže je v zobrazovací ikoně použit následující kód:

```
PocetSlov := WordCount(EntryText)
```



Obrázek 47: Struktura pro počítání počtu slov a výstup na obrazovku.

Ukázka použití funkce WordCount je k dispozici na přiloženém CD v adresáři Funkce\WordCount.a7p

4.3.1.1.6 GetNumber

Použití: $Cislo := GetNumber(n, „řetězec“)$

Popis: Tato funkce vrací hodnotu n-tého čísla v řetězci.

Například $GetNumber(5, „8, 15, 22, 36, 51, 64“)$ vrátí hodnotu pátého čísla, tzn. 51.

Parametry:

n – určuje, kolikáté číslo v řetězci zadaném parametrem má funkce vrátit. Pokud je parametr n větší než počet čísel v řetězci, pak funkce GetNumber vrátí hodnotu 0.

$řetězec$ – proměnná obsahující řetězec, nebo konkrétní řetězec, ve kterém se má najít n -té číslo.

Příklad: viz funkce NumCount.

4.3.1.2 Souborové funkce

Souborové funkce zahrnují funkce pro práci se soubory, pomocí nich můžeme číst ze souboru a zapisovat do souboru.

4.3.1.2.1 AppendExtFile, WriteExtFile

Použití: `ZapisDoSouboru := AppendExtFile („jméno souboru“, „řetězec“)`

`ZapisDoSouboru := WriteExtFile („jméno souboru“, „řetězec“)`

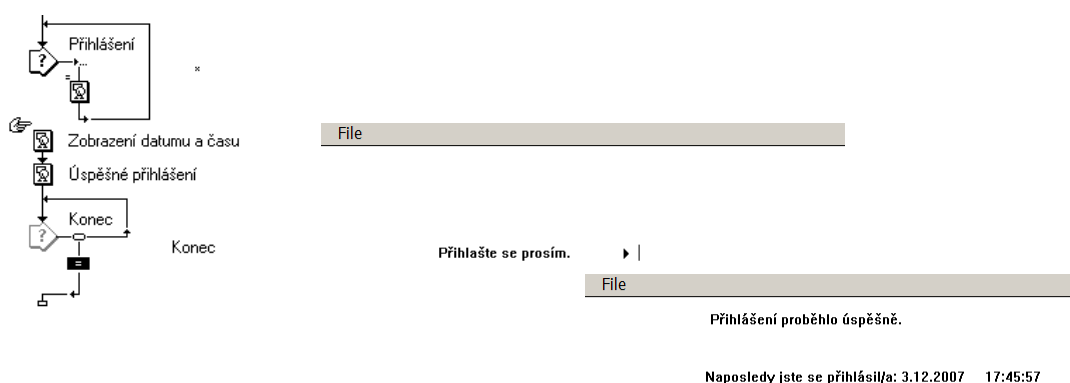
Popis: Funkce `AppendExtFile` a `WriteExtFile` slouží k ukládání řetězce zadaného parametrem („řetězec“) do určeného souboru. `AppendExtFile` připisuje řetězec na konec souboru, `WriteExtFile` přepisuje obsah souboru. V případě, že zadaný soubor neexistuje, tak ho funkce vytvoří.

Parametry:

jméno souboru – jméno souboru, do kterého se má zapisovat.

řetězec – konkrétní obsah toho, co se má zapsat do specifikovaného souboru.

Příklad: Využití zapisování informací do externího souboru není příliš časté, ale v určitých chvílích může mít své opodstatnění. Takovým příkladem může být třeba ukládání času, kdy naposledy se uživatel přihlásil do didaktické aplikace.



Obrázek 48: Systém na zjišťování posledního přihlášení uživatele

Ukázka použití funkce `WriteExtFile` je k dispozici na přiloženém CD v adresáři `Funkce\WriteExtFile.a7p`

4.3.1.2.2 ReadExtFile

Použití: *Retezec := ReadExtFile([URL]název souboru)*

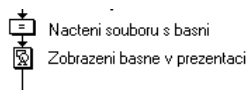
Popis: Funkce `ReadExtFile` umožňuje načtení obsahu textového souboru specifikovaného parametrem název souboru. V parametru název souboru se zadává i umístění souboru a to buď na pevném disku počítače nebo na internetové adrese (pomocí nepovinného parametru URL).

Parametry:

název souboru – umístění a název souboru, jehož obsah se má načíst

URL – internetová adresa souboru jehož obsah se má načíst (např. `http://maw.webzdarma.cz/maj.txt`).

Příklad: Pomocí této funkce může tvůrce didaktické aplikace dynamicky měnit obsah prezentace změnou zdrojového souboru. Na příkladu níže je ukázán případ, kdy se z internetové adresy načte soubor `maj.txt`, který obsahuje úryvek básně K.H.Mácha *Máj*, a následně se zobrazí uživateli v okně prezentace.



Obrázek 49: Strukturogram příkladu funkce `ReadExtFile`

V kalkulační ikoně *Nacteni souboru s basni* je definováno umístění souboru:

```
Ma j := ReadExtFile("http://maw.webzdarma.cz/maj.txt")
```

A text básně je přiřazen do uživatelské proměnné `Ma j`. Zobrazovací ikona *Zobrazeni basne v prezentaci* zajistí výpis obsahu proměnné `Ma j` v okně prezentace pomocí vložení obsahu proměnné `Ma j` (`{Ma j}`). Výsledek může vypadat například takto:

Karel Hynek Mácha - *Máj*

Byl pozdní večer – první máj –
večerní máj – byl lásky čas.
Hrdliččin zval ku lásce hlas,
kde borový zaváněl háj.
O lásce šeptal tichý mech;
květoucí strom lhal lásky žel,
svou lásku slavík růži pěl,
růžinu jevil vonný vzdech.



Obrázek 50: Načtený text z internetu v okně aplikace

Ukázka použití funkce `ReadExtFile` je k dispozici na přiloženém CD v adresáři `Funkce\ReadExtFile.a7p`

4.3.1.3 Obecné funkce

Obecné funkce mají širokou škálu využití a mohou být dále rozděleny podle toho, jaký druh práce vykonávají. Rozdělíme si je do čtyř podkategorií: multimédia, funkce ovládající systém, synchronizační a ostatní.

Multimédia

Funkce v podkategorii multimédia obstarávají přehrávání filmů a zvuků.

4.3.1.3.1 MediaPause

Použití: `MediaPause(IconID@“NazevIkony“, stav)`

Popis: Funkce `MediaPause` pozastavuje a pouští přehrávání filmu a videa.

Parametry:

NazevIkony - název filmové nebo video ikony specifikované názvem.

stav – pokud je stav pravdivý (TRUE), potom je video zastaveno na aktuálním snímku.

Jestliže je nepravdivý (FALSE), potom pokračuje přehrávání videa od aktuálního snímku.

Příklad: Pomocí této funkce může uživatel zastavovat a pouštět video. Detailní příklad bude uveden u funkce `MediaSeek`.

4.3.1.3.2 MediaPlay

Použití: `MediaPlay(IconID@“NazevIkony“)`

Popis: Funkce `MediaPlay` spouští přehrávání filmu a videa.

Parametry: *NazevIkony* – název filmové nebo video ikony.

Příklad: Pomocí této funkce lze spouštět přehrávání videa nebo filmu. Detailní příklad bude uveden u funkce `MediaSeek`.

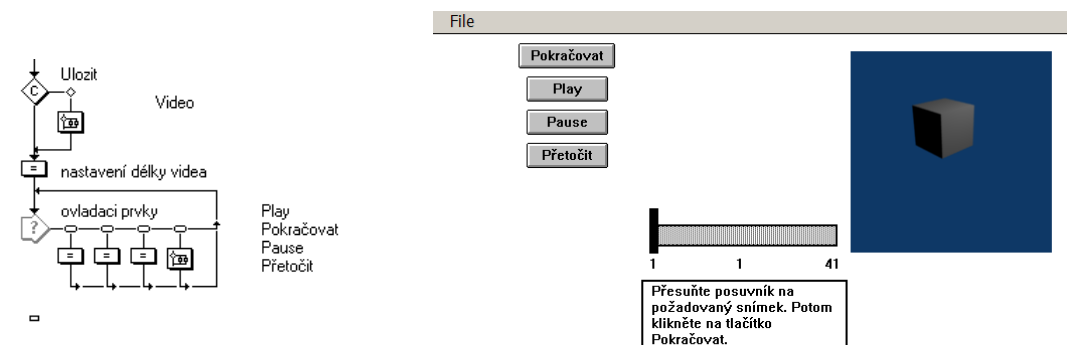
4.3.1.3.3 MediaSeek

Použití: `MediaPlay(IconID@"NazevIkony", číslo snímku)`

Popis: Pomocí této funkce může uživatel přetáčet přehrávání filmu a videa na požadovaný snímek .

Parametry: *NazevIkony* – název filmové nebo video.

Příklad: Pomocí funkcí `MediaSeek`, `MediaPlay` a `MediaPause` můžeme dát uživateli kontrolu nad přehráváním videa. V níže uvedeném příkladu je ukázán strukturogram v němž jsou zobrazeny ovládací prvky videa (*Play*, *Pokračovat*, *Pause*, *Přetočit*) a prezentační okno jak jej uvidí uživatel aplikace.



Obrázek 51: Ukázka interaktivní aplikace pro přehrávání videa

Slučovací ikona *Video* obsahuje filmovou ikonu *Video2*, která obsahuje cestu k souboru s videem. Kalkulační ikona pod ní (*nastaveni delky videa*) obsahuje výraz pro výpočet délky videa (ve snímcích):

```
delka := MediaLength@"Video2"
```

Jednotlivé ovládací prvky využívají funkcí, které jsme probrali dříve, konkrétně funkce `MediaPlay` pro tlačítko *Play*:

```
MediaPlay(@"Video2")
```

MediaPause pro tlačítko *Pause*:

```
MediaPause(@"Video2" ; TRUE)
```

MediaSeek pro tlačítko *Přetočit*:

```
MediaSeek(@"Video2" ; 41)
```

Ukázka použití funkcí MediaPause, MediaPlayer a MediaSeek je k dispozici na příloženém CD v adresáři Funkce\MediaSeek.a7p

Ovládání systému

Funkce v této kategorii ovládají běh aplikace (Quit a Restart) a tisk (PrintScreen).

4.3.1.3.4 PrintScreen

Použití: PrintScreen()

Popis: Funkce PrintScreen umožňuje vytisknout aktuální obrazovku na tiskárnu, která je nastavena jako výchozí. Pokud prezentace Authorwaru nezabírá celou obrazovku, vytiskne se nejen prezentace, ale celá obrazovka.

Parametry: Tato funkce nemá žádné parametry.

Příklad: Pomocí této funkce můžeme uživateli umožnit vytisknout si své výsledky testů, výtvorů apod.



Obrázek 52: Nastavení aplikace s tlačítkem pro tisk

V tomto jednoduchém případě je na tlačítko *Tisk* připojena kalkulační ikona, která obsahuje funkci PrintScreen().

Ukázka použití funkce PrintScreen je k dispozici na příloženém CD v adresáři Funkce\PrintScreen.a7p

4.3.1.3.5 Quit

Použití: Quit(volba)

Popis: Tato funkce umožňuje ukončit práci s Authorwarem, bez nutnosti resetovat počítač.

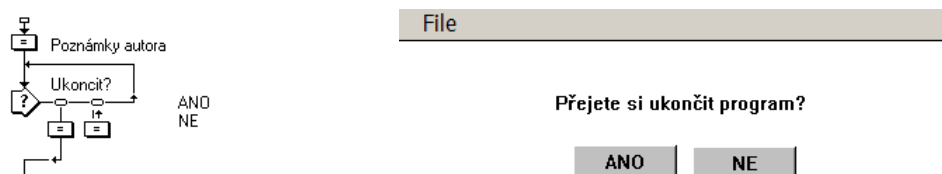
Parametry: volba – 0,1 – návrat na plochu Windows

2 – restart Windows

3 – vypnutí počítače

Příklad: V následujícím příkladu má uživatel dvě volby (ANO/NE). V kalkulační ikoně u tlačítka ANO je použita funkce Quit s parametrem 0 („Quit(0)“). U tlačítka NE je v kalkulační ikoně pouze komentář „--nic se neděje“.

Ukázka použití funkce Quit je k dispozici na příloženém CD v adresáři Funkce\Quit.a7p



Obrázek 53: Ukázka použití funkce Quit

4.3.1.3.6 Restart

Použití: Restart()

Popis: Funkce Restart přinutí MAW přeskočit na začátek strukturogramu souboru a jeho znovuspuštění. Při jejím použití se obnoví počáteční hodnota všech proměnných. Smí být používána v kalkulační ikoně, ne ve výrazech nebo pro zobrazování v okně prezentace.

Parametry: Tato funkce nemá žádné parametry.

Příklad: Tuto funkci využijeme, je-li potřeba aplikaci jednoduše znovu nastartovat. To platí zejména u různých her, kde přidáním tlačítka pro novou hru s využitím funkce Restart dovolíme uživateli hrát novou hru, bez nutnosti zapínání a vypínání prezentace. Viz příklad u funkce Random.

Synchronizační funkce

V této podkategorii jsou dvě funkce `SyncPoint` a `SyncWait`, které umožňují synchronizaci prezentace ikon.

4.3.1.3.7 SyncPoint, SyncWait

Použití: `SyncPoint(volba)`

`SyncWait(počet sekund)`

Popis: Funkce `SyncPoint` a `SyncWait` slouží k synchronizaci prezentace jednotlivých ikon.

Parametry:

volba – 0 – nastaví časovač před zobrazením aktuální ikony

1 – nastaví časovač po zobrazení aktuální ikony

2 – nastaví časovač při odpovědi uživatele nebo když uživatel opustí interakci

počet sekund – počet sekund, na jak dlouho se má zastavit prezentace

Příklad: Synchronizace zobrazování ikon a prezentace je důležitá pro správné pořadí zobrazování elementů prezentace (grafiky, videa a zvuku). Příklad použití funkcí `SyncPoint` a `SyncWait` bude uveden u funkce `WaitMouseUp`.

Ostatní funkce

V této podkategorii jsou obecné funkce, které se nevešly do předchozích podkategorií.

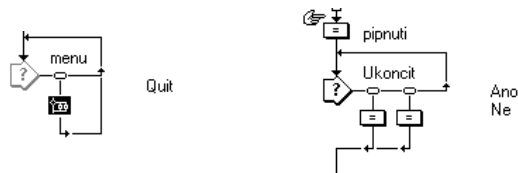
4.3.1.3.8 Beep

Použití: `Beep ()`

Popis: Tato funkce spouští zvuk pípnutí.

Parametry: Tato funkce nemá žádné parametry.

Příklad: Funkce je užitečná ve chvílích, kdy potřebujeme upoutat uživatelskou pozornost na určitou akci, např. na stisknutí tlačítka. V následujícím příkladu systém pípne, když uživatel klikne na tlačítko *Quit*.



Obrázek 54: Použití funkce Beep()

V kalkulační ikoně *pipnuti* je použita funkce Beep ().

Ukázka použití funkce Beep je k dispozici na přiloženém CD v adresáři Funkce\Beep.a7p

4.3.1.3.9 WaitMouseUp

Použití: WaitMouseUp ()

Popis: Funkce WaitMouseUp pozastaví prezentaci, pokud je stisknuto levé tlačítko myši.

Parametry: Tato funkce nemá žádné parametry.

Příklad: Využití této funkce spočívá především v tom, že můžeme uživateli zabránit klikat levým tlačítkem myši během prezentace, respektive když uživatel klikne během prezentace, tak se prezentace pozastaví. Tato funkce se dá také využít pro přerušování prezentace jako je například galerie obrázků, kde se obrázky automaticky střídají. V příkladu uvedeném níže se zobrazují uživateli obrázky plemen psů. Mezi zobrazením jednotlivých obrázků je časová prodleva 1 sekunda. V případě, že uživatel drží stisknuté levé tlačítko myši, tak se přehrávání obrázků pozastaví.



Obrázek 55: Fotogalerie využitím funkce WaitMouseUp

Na rozhodovací ikonu *Obrázky psů* je připojen výraz:

```
SyncPoint(0)
```

```
SyncWait(1)
```

```
WaitMouseUp()
```

Viz funkce `SyncPoint` a `SyncWait`.

Ukázka použití funkce `WaitMouseUp` je k dispozici na přiloženém CD v adresáři `Funkce\WaitMouseUp.a7p`

4.3.1.3.10 Initialize

Použití: `Initialize([proměnná1, [proměnná2,...])`

Popis: Tato funkce obnovuje nastavení hodnot proměnných, které jdou měnit uživatelem, na výchozí.

Parametry: Pokud funkci nezadáme žádné parametry, tak přenastavení hodnot proměnných na jejich výchozí hodnoty proběhne u všech. V případě, že jako parametr zadáme jména proměnných, pak se přenastavení hodnot provede jen u vyjmenovaných proměnných. Názvy jednotlivých proměnných se oddělují čárkou.

Příklad: Tuto funkci využijeme v případě, když budeme chtít dát uživateli možnost vyzkoušet si cvičení nebo test znovu a dovolíme mu například vynulovat skóre.

Funkce `Initialize` je použita v ukázce u funkce `WordCount`.

4.3.1.3.11 Test

Použití: `Test(podmínka, pravdivý výraz, nepravdivý výraz)`

Popis: Funkce `test` vyhodnocuje podmínku zadanou parametrem. Pokud je podmínka pravdivá, pak je vyhodnocen pravdivý výraz. Pokud je nepravdivá a je zadaný i nepravdivý výraz, pak je vyhodnocen nepravdivý výraz.

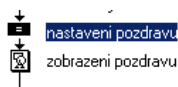
Parametry:

podmínka – podmínkou může být jakýkoliv logický výraz, to znamená výraz, o kterém lze prohlásit, že je pravdivý(TRUE) nebo nepravdivý(FALSE).

pravdivý výraz – výraz, který se vyhodnotí, jestliže je podmínka pravdivá(TRUE)

nepravdivý výraz – výraz, který se vyhodnotí, jestliže je podmínka nepravdivá(FALSE)

Příklad: Tato funkce má široké využití při vyhodnocování interakce s uživatelem nebo při vyhodnocování data a času. Na konkrétním příkladu si ukážeme, jak nastavit správné oslovení uživatele v závislosti na době ve kterou aplikaci spustil.



Obrázek 56: Strukturogram u příkladu funkce Test

Kalkulační ikona nastaveni pozdravu obsahuje následující kód:

```

Test (Hour < 12; pozdrav := "Dobré ráno."; pozdrav := "Dobré
poledne. ")
Test (Hour > 13; pozdrav := "Dobré odpoledne. ")
Test (Hour > 17; pozdrav := "Dobrý večer. ")
  
```

Ukázka použití funkce Test je k dispozici na přiloženém CD v adresáři Funkce\Test.a7p

4.3.1.3.12 SetCursor

Použití: SetCursor(*číslo kurzoru*)

Popis: Tato funkce umožňuje měnit tvar kurzoru.


Parametry:

číslo kurzoru – tento parametr určuje, jak bude vypadat zobrazovaný kurzor.


0 – standardní systémový kurzor (šipka) 


1 – editační kurzor 

2 – křížek 

3 – plus 

4 – prázdný kurzor

5 – přesýpací hodiny 

6 – ruka 

Příklad: Pomocí této funkce můžeme snadno přitáhnout pozornost uživatele. Prázdný kurzor můžeme použít v případě, kdy nechceme aby kurzor rušil uživatele při sledování prezentace.

4.3.1.3.13 ShowCursor

Použití: ShowCursor(*ON/OFF*)

Popis: Použitím této funkce s parametrem *OFF* docílíme prakticky shodného výsledku jako při použití funkce SetCursor(4) – které vede ke skrytí kurzoru.

Parametry:

ON – při použití tohoto parametru bude kurzor viditelný.

OFF – při použití tohoto parametru bude kurzor skrytý.

Příklad: viz příklad u funkce SetCursor.

4.3.1.4 Ikonové funkce

Ikonové funkce dovolují manipulovat ikonami a jejich obsahem.

4.3.1.4.1 EraseAll, EraseIcon

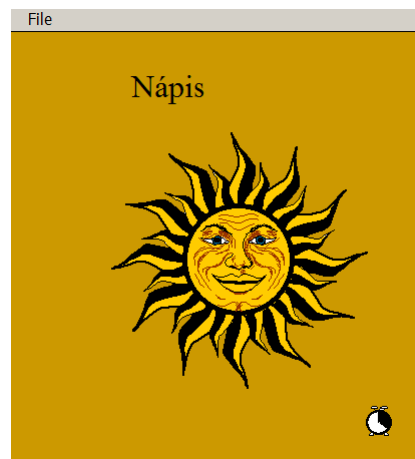
Použití: EraseAll()

EraseIcon(IconID@“*NazevIkony*“)

Popis: Funkce EraseAll vymazává objekty z obrazovky (text, grafiku a video). Vymazání se netýká barvy pozadí. Funkce EraseIcon vymazává z obrazovky ikony specifikované názvem. Obě funkce mohou být používány pouze v kalkulační ikoně, ne ve výrazech.

Parametry: Funkce EraseAll nemá žádné parametry. Funkce EraseIcon má jako parametr název ikony, která má být vymazána.

Příklad: Pomocí funkce EraseAll vymažeme objekty z okna prezentace, což využijeme například při vytváření přechodů v mezi obrazovkami prezentace. V následující ukázce je funkce EraseAll použita pro smazání nápisu(text) a sluníčka(grafika) po 5 vteřinovém čekání nastaveném v čekací ikoně. V kalkulační ikoně *smaz vse* je vložen kód EraseAll().



Obrázek 57: 5 vteřinový odpočet před smazáním obrazovky

Ukázka použití funkce `EraseAll` je k dispozici na přiloženém CD v adresáři `Funkce\EraseAll.a7p`

4.3.1.4.2 Preload

Použití: `Nahravani := Preload(IconID@“NazevIkony“ [,volba])`

Popis: Funkce `Preload` nahraje do paměti počítače data potřebná ke spuštění ikony specifikované názvem (ikony použité s funkcí `Preload` mohou být zobrazovací, zvukové a filmové). V případě, že je specifikována seskupovací ikona, potom jsou nahrány do paměti počítače všechny zobrazovací, zvukové a filmové ikony, které jsou připojeny na specifikovanou slučovací ikonu. Externí soubory jako např. `.avi` a `.mpeg` nejsou nahrávány do paměti počítače. Nahrání ikon do paměti počítače je výhodné kvůli rychlejšímu přístupu k datům v paměti počítače.

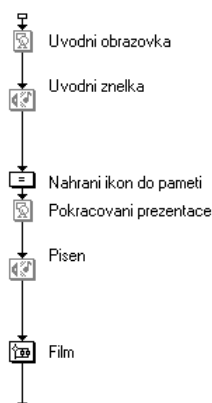
Parametry:

volba – 0 – MAW nepřevéde grafické soubory při nahrávání do paměti do optimálního formátu

1- MAW převede grafické soubory při nahrávání do paměti do optimálního formátu, tato volba je brána jako výchozí.

Příklad: Při správném použití funkce `Preload` můžeme dosáhnout plynulejšího běhu prezentace tím, že nahrajeme paměťově náročné elementy aplikace (grafika, zvuk, video) do paměti počítače. Nejčastěji je tato funkce využívána ve chvílích, kdy je uživatel zaměstnán prezentací (např. plněním úkolu) a tak může být strojový čas

efektivně využít k nahrání potřebných elementů prezentace. V následujícím příkladu jsou nahrány do paměti obsahy ikon s velkou zátěží na počítač (*Pokracovani prezentace, Pisen, Film*).



Obrázek 58: Využití funkce Preload

Kalkulační ikona *Nahrani ikon do pameti* obsahuje kód:

```
Preload(IconID@"Pokracovani prezentace")
```

```
Preload(IconID@"Pisen")
```

```
Preload(IconID@"Film")
```

Ten zajistí, že vyjmenované ikony budou nahrány do paměti počítače mezitím, co je uživatel zaměstnán úvodní obrazovkou a znělkou.

4.3.1.5 Přeskakovací funkce

Přeskakovací funkce umožňují přerušení sekvenční prezentace ikon podle časové osy. Přeskakování může být mezi ikonami v souboru (*GoTo*), přeskakování mezi soubory Authorwaru (*JumpFile*, *JumpFileReturn*) a přeskakování do jiných aplikací (*JumpOut*, *JumpOutReturn* a *JumpPrintReturn*).

4.3.1.5.1 GoTo

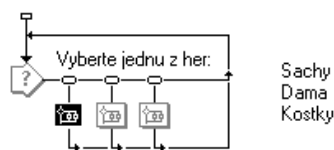
Použití: *GoTo(IconID@"NazevIkony")*

Popis: Pomocí funkce *GoTo* můžeme přeskakovat mezi ikonami. Když Authorware narazí na funkci *GoTo*, přeskočí na ikonu specifikovanou názvem. Pomocí funkce *GoTo* nelze přeskakovat mezi ikonami, které jsou v různých souborech.

Parametry: *NazevIkony* – název ikony, na kterou se má ve strukturogramu přeskočit.

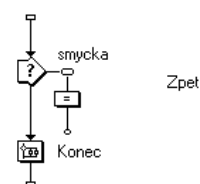
Příklad: Protože funkce GoTo umožňuje obejít běžného (lineárního) zpracování kódu, využijeme ji, když potřebujeme dát uživateli volnost ve výběru (např. různá menu).

V následující ukázce si uživatel vybere jednu z her.



Obrázek 59: Rozcestník ovládaný pomocí funkce GoTo

Podívejme se na první volbu (Sachy), kde kalkulační ikona Zpet obsahuje kód `GoTo(IconID@"Konec")`.



Obrázek 60: Ukázka nelineárního pŕchodu strukturogramu

4.3.1.5.2 JumpFile, JumpFileReturn

Použití: `JumpFile („název souboru“[,proměnná1] [,proměnná2])`

`JumpFileReturn („název souboru“[,proměnná1] [,proměnná2])`

Popis: Pomocí této funkce lze přeskokovat mezi soubory Authorwaru, na soubory zadané parametrem *název souboru*. Při přeskočení z jednoho souboru na druhý pokračuje prezentace ve druhém souboru.

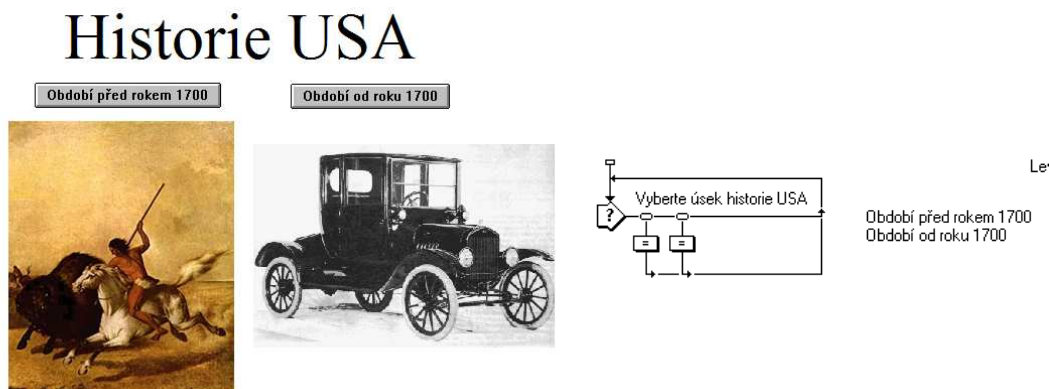
Funkce `JumpFileReturn` také umožňuje přeskokovat mezi soubory Authorwaru, avšak s tím rozdílem, že po skončení prezentace v druhém souboru Authorware přeskočí zpět na první soubor. Obě funkce mohou být použity pouze v kalkulační ikoně. Pokud je parametrem specifikována konkrétní proměnná, pak se předá její hodnota volanému souboru.

Parametry:

název souboru – umístění a název souboru, na který se má přeskočit

proměnná – nepovinný parametr – název proměnné, jejíž hodnota se má předat volanému souboru.

Příklad: V rozsáhlých didaktických aplikacích může být konstrukce aplikace v jednom souboru nepřehledná. V takové situaci můžeme obsah jednoho velkého souboru MAW rozdělit do menších souborů a pomocí funkcí `JumpFile` a `JumpFileReturn` na soubory vzájemně odkazovat. Jako příklad může posloužit historie USA. Na obrázku níže má uživatel na výběr, jaký časový úsek historie USA chce sledovat, buď období před rokem 1700 nebo od roku 1700.



Obrázek 61: Větení aplikace za použití více souborů MAW

Obě tlačítka směřují na další soubory Authorwaru, které obsahují prezentaci pro zvolené období. Tlačítko *Období před rokem 1700* obsahuje následující kód:

```
JumpFile("c:\\projekty\\AmerikaPred1700"), který odkazuje na soubor AmerikaPred1700.
```

Podobně tlačítko *Období od roku 1700* obsahuje kód:

```
JumpFile("c:\\projekty\\AmerikaPo1700"), který odkazuje na soubor AmerikaPo1700.
```

4.3.1.5.3 JumpOut, JumpOutReturn

Použití: `JumpOut(„program“)`

`JumpOut(„program“, „dokument“)`

`JumpOutReturn(„program“)`

`JumpOutReturn(„program“, „dokument“)`

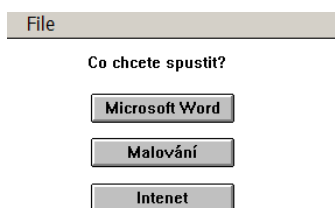
Popis: Funkce `JumpOut` a `JumpOutReturn` slouží ke spuštění dalších aplikací, které jsou definované v parametru *program* (takovou aplikací může být např. MS Word). Na rozdíl od funkce `JumpOut`, která při spuštění volané aplikace ukončí běh Authorwaru, `JumpOutReturn` volanou aplikaci spustí a prezentaci Authorwaru nechá dál spuštěnou na pozadí kvůli návratu. Obě funkce mohou být používány pouze v kalkulačních ikonách, nedají se vkládat přímo do okna prezentace. Parametr *dokument* se předává aplikaci definované parametrem *program*. Nejčastěji to bývá jméno souboru, který chceme ve volané aplikaci spustit. Pokud zadáme pouze parametr dokument, Authorware se nás vyzve k nalezení aplikace, ve které chceme soubor spustit.

Parametry:

program – název a cesta k aplikaci, kterou chceme spustit

dokument – parametr předávaný Authorwarem spuštěné aplikaci.

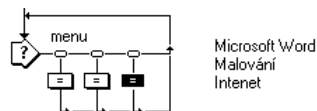
Příklad: Funkci `JumpOut` můžeme použít pro vytváření jednoduchých nabídek pro uživatele. Na obrázcích níže můžeme vidět nabídku, která uživateli umožní vybrat jednu ze tří aplikací:



Obrázek 62: Jednoduché menu se základními programy

Strukturogram menu skládá ze tří kalkulačních ikon, kde každá obsahuje název a umístění aplikace, kterou chceme pomocí Authorware spustit.

Například kalkulační ikona *Internet* obsahuje tento kód:



Obrázek 63: Strukturogram rozcestníku

```
JumpOut("C:\\Program Files\\Internet Explorer\\iexplore.exe")
```

Ukázka použití funkce `JumpOut` je k dispozici na přiloženém CD v adresáři Funkce\\JumpOut.a7p

4.3.1.5.4 JumpPrintReturn

Použití: JumpPrintReturn („*program*“, „*dokument*“)

JumpPrintReturn („“, „*dokument*“)

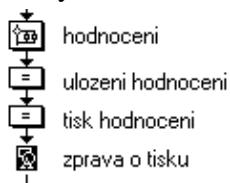
Popis: Tato funkce otevře aplikaci určenou parametrem *program* a vytiskne soubor určený parametrem *dokument*. Když je spuštěna volaná aplikace, tak se spustí její možnosti tisku. Po dokončení tisku se volaná aplikace zavře a pokračuje prezentace MAW. Pokud není definována aplikace ze které se má tisknout (jako parametr *program* jsou zadány pouze uvozovky „“), pak Authorware sám vybere aplikaci ze které se bude tisknout. Pokud takovou aplikaci nenalezne, tak vyzve uživatele, aby našel aplikaci pro tisk. Funkce JumpPrintReturn může být používána pouze v kalkulační ikoně.

Parametry:

program – program ve kterém se má provést tisk

dokument – název souboru, který se má vytisknout

Příklad: Pomocí této funkce můžeme vytisknout hodnocení uživatele a další informace.



Obrázek 64: Strukturogram příkladu funkce JumpPrintReturn

Na strukturogramu uvedeném výše je uveden příklad tisku hodnocení žáka. V seskupovací ikoně *hodnoceni* je kalkulační ikona, která obsahuje kód:

```
hodnoceni := "Zde by mohlo být například hodnocení žáka"
```

Do uživatelské proměnné *hodnoceni* je uložen řetězec, který chceme vytisknout.

V kalkulační ikoně *ulozeni hodnoceni* je kód, který pomocí funkce WriteExtFile uloží obsah proměnné *hodnoceni* do souboru *report.txt*:

```
WriteExtFile("c:\\report.txt"; hodnoceni)
```

Příkaz k tisku je obsažen v kalkulační ikoně *tisk hodnoceni*:

```
JumpPrintReturn("notepad.exe"; "c:\\report.txt")
```

Funkce `JumpPrintReturn` otevře *Poznámkový blok* (standardní aplikace MS Windows, která se spouští souborem `notepad.exe`) a vytiskne v něm obsah souboru `report.txt`.

Zobrazovací ikona *zprava o tisku* může obsahovat různá hlášení, například, že se tisk úspěšně podařil.

Ukázka použití funkce `JumpPrintReturn` je k dispozici na příloženém CD v adresáři `Funkce\JumpPrintReturn.a7p`

4.3.1.6 Matematické funkce

Matematické funkce mohou být rozděleny do čtyř podkategorií, na trigonometrické funkce, další funkce, funkce vytváření náhodných čísel a funkce ovlivňující formát čísel.

4.3.1.6.1 ACOS

Použití: `cislo := ACOS(x)`

Popis: Funkce `ACOS` vypočítává hodnotu arkus kosinus z čísla zadaného parametrem x , v rozmezí od 0 do π .

Parametry: x - číslo, ze kterého se má vypočítat hodnota arkus kosinus

Příklad: Funkce `ACOS` spolu s dalšími trigonometrickými funkcemi (viz funkce `COS`, `SIN`, `TAN`, `ASIN` a `ATAN`) se používá převážně v trigonometrii a dalších výpočetních úlohách.

4.3.1.6.2 ASIN

Použití: `cislo := ASIN(x)`

Popis: Funkce `ASIN` vypočítává hodnotu arkus sinus z čísla zadaného parametrem x , v rozmezí od $-\pi/2$ do $\pi/2$.

Parametry: x - číslo, ze kterého se má vypočítat hodnota arkus sinus

Příklad: Funkce `ASIN` spolu s dalšími trigonometrickými funkcemi (viz funkce `COS`, `SIN`, `TAN`, `ACOS` a `ATAN`) se používá převážně v trigonometrii a dalších výpočetních úlohách.

4.3.1.6.3 ATAN

Použití: $cislo := ATAN(x)$

Popis: Funkce ATAN vypočítává hodnotu arkus tangens z čísla zadaného parametrem x , v rozmezí od $-\pi/2$ do $\pi/2$.

Parametry: x - číslo, ze kterého se má vypočítat hodnota arkus tangens

Příklad: Funkce ATAN spolu s dalšími trigonometrickými funkcemi (viz funkce COS, SIN, TAN, ACOS a ASIN) se používá převážně v trigonometrii a dalších výpočetních úlohách.

4.3.1.6.4 COS

Použití: $cislo := COS(\acute{uhel})$

Popis: Funkce COS vypočítává hodnotu kosinus z úhlu zadaného parametrem \acute{uhel} . Výsledek výpočtu je v radiánech.

Parametry: \acute{uhel} – úhel, ze kterého se má vypočítat hodnota kosinus

Příklad: Funkce COS spolu s dalšími trigonometrickými funkcemi (viz funkce ATAN, SIN, TAN, ACOS a ASIN) se používá převážně v trigonometrii a dalších výpočetních úlohách.

4.3.1.6.5 SIN

Použití: $cislo := SIN(\acute{uhel})$

Popis: Funkce SIN vypočítává hodnotu sinus z úhlu zadaného parametrem \acute{uhel} . Výsledek výpočtu je v radiánech.

Parametry: \acute{uhel} – úhel, ze kterého se má vypočítat hodnota sinus

Příklad: Funkce SIN spolu s dalšími trigonometrickými funkcemi (viz funkce ATAN, COS, TAN, ACOS a ASIN) se používá převážně v trigonometrii a dalších výpočetních úlohách.

4.3.1.6.6 TAN

Použití: $cislo := TAN(\acute{uhel})$

Popis: Funkce TAN vypočítává hodnotu tangens z úhlu zadaného parametrem \acute{uhel} . Výsledek výpočtu je v radiánech.

Parametry: *úhel* – úhel, ze kterého se má vypočítat hodnota tangens

Příklad: Funkce TAN spolu s dalšími trigonometrickými funkcemi (viz funkce ATAN, SIN, COS, ACOS a ASIN) se používá převážně v trigonometrii a dalších výpočetních úlohách.

Další matematické funkce.

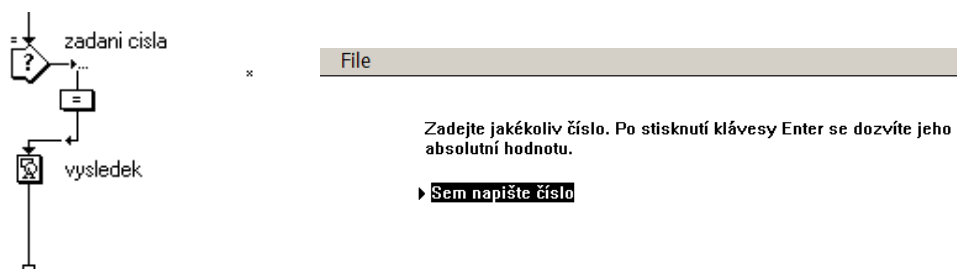
4.3.1.6.7 ABS

Použití: $cislo := ABS(x)$

Popis: Funkce ABS vypočítává absolutní hodnotu z čísla zadaného parametrem x .

Parametry: x – číslo, ze kterého se má vypočítat absolutní hodnota

Příklad: Funkce ABS využijeme, potřebujeme-li znát absolutní hodnotu čísla (například vzdálenost). Na příkladu uvedeném níže si ukážeme, jak zjišťovat absolutní hodnotu čísla.



Obrázek 65: Výpočet absolutní hodnoty čísla

V interakční ikoně *zadani cisla* je žák vyzván aby zadal jakékoliv číslo.

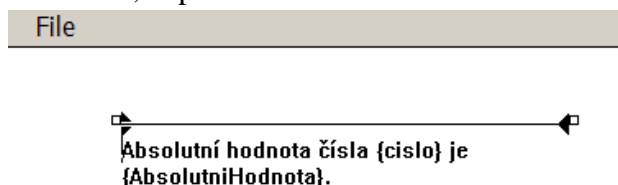
V kalkulační ikoně připojeném na interakční ikonu je následující kód:

```
cislo := NumEntry
```

```
Test(cislo < 0; AbsolutniHodnota := ABS(cislo))
```

```
Test(cislo > 0; AbsolutniHodnota := cislo)
```

Který hodnotu vloženou uživatelem (NumEntry) přiřadí do proměnné *cislo* a absolutní hodnotu čísla přiřadí do proměnné *AbsolutniHodnota*. V zobrazovací ikoně *vysledek* zobrazíme absolutní hodnotu čísla uživateli, například takto:



Obrázek 66: Výpis výsledku na obrazovku

Ukázka použití funkce ABS k dispozici na příloženém CD v adresáři Funkce\ABS.a7p

4.3.1.6.8 MOD

Použití: $cislo := MOD(x;y)$

Popis: Funkce MOD slouží k vypočítávání zbytků po celočíselném dělení.

Parametry: x – dělenec

y – dělitel

Příklad: Pomocí počítání zbytku po celočíselném dělení můžeme převádět čas z 24 hodinového formátu na formát 12 hodinový. Výsledkem následujícího kódu:

```
12HodinovyZapis := MOD(23;12)
```

Bude 11 (celočíselný zbytek po výpočetní operaci 23:11)

4.3.1.6.9 SQRT

Použití: $Cislo := SQRT(x)$

Popis: Pomocí funkce SQRT vypočítáme druhou odmocninu z čísla zadaného parametrem x .

Parametry: x – číslo, ze kterého se má vypočítat druhá odmocnina

Příklad: Funkce SQRT se používá v nejrůznějších početních úlohách (např. v pythagorově větě, kvadratických rovnicích apod.)

Funkce pro vytváření náhodných čísel.

4.3.1.6.10 Random

Použití: $Random(min;max;jednotky)$

Popis: Pomocí funkce Random lze generovat náhodná čísla.

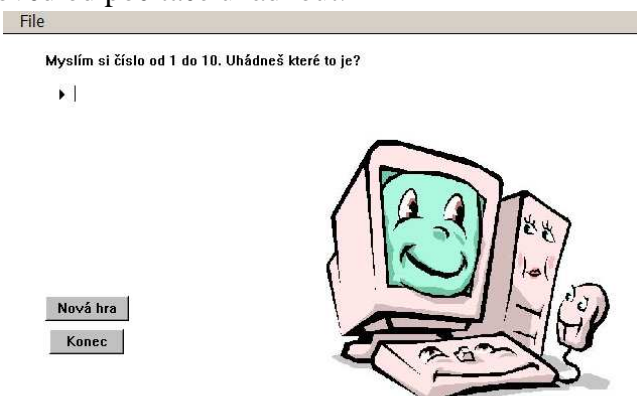
Parametry:

min – spodní hranice rozsahu pro generování náhodných čísel

max – horní hranice rozsahu pro generování náhodných čísel

$jednotky$ – parametr jednotky určuje, v jakých jednotkách bude generované číslo (např. v desetínách, jednotkách, stovkách atd.)

Příklad: Pomocí funkce `Random` můžeme vytvářet různé hry, které potřebují určitý stupeň náhodnosti. Jako příklad může posloužit hra hádání čísel, kdy si počítač myslí nějaké číslo a žák ho má pomocí nápověd od počítače uhádnout.



Obrázek 67: Úvodní obrazovka hry hádání čísel

Strukturogram hry v Authorwaru vypadá takto:



Obrázek 68: Strukturogram hry

V kalkulační ikoně *Vygenerovani cisla* je kód, který vygeneruje číslo od 1 do 10:

```
cislo := Random(1;10;1)
```

Jednička ve výrazu úplně vpravo zajistí, že vygenerované číslo bude celé.

Kalkulační ikona *Nová hra* obsahuje kód pro znovuspuštění hry:

```
Restart() – viz funkce Restart
```

Kalkulační ikona *Konec* obsahuje kód pro ukončení hry:

```
Quit(0) – viz funkce Quit
```

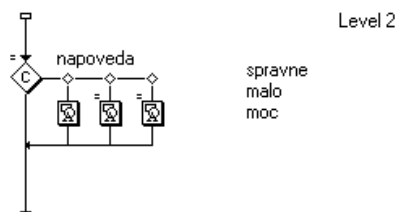
Ve slučovací ikoně *** jsou nastaveny všechny případy, které mohou během hádání čísla nastat (uživatel hádá moc, málo nebo správně). K rozodovací ikoně *napoveda* je připojen výraz:

```
Odhad := NumEntry
```

PocetPokusu:=1+Malo+Moc -- přičtená 1 je za správný pokus

Do proměnné Odhad se přiřadí číslo zadané uživatelem a v proměnné PocetPokusu se ukládá počet pokusů, které uživatel potřeboval k uhádnutí čísla (viz dále). U interakční ikony *napoveda* je nastaveno větvení na *To Calculated Path*, které obsahuje kód pro vybrání větve strukturogramu podle toho, jaké číslo uživatel hádal:

$(\text{Odhad} = \text{cislo}) + 2 * (\text{Odhad} < \text{cislo}) + 3 * (\text{Odhad} > \text{cislo})$



Obrázek 69: Struktura možností odhadů

V případě, že uživatel tipoval správně, tak se zvolí první větev *spravne*. Jestliže bylo číslo, které hádal menší než to, které si myslí počítač, pak se vybere druhá větev a v případě, že číslo hádané uživatelem je vyšší než myšlené číslo, potom se vybere třetí, poslední větev.

V každé ze tří zobrazovacích ikon (*spravne*, *moc*, *malo*) připojených na interakční ikonu *napoveda* je nastaveno počítadlo pokusů a příslušná nápověda uživateli.

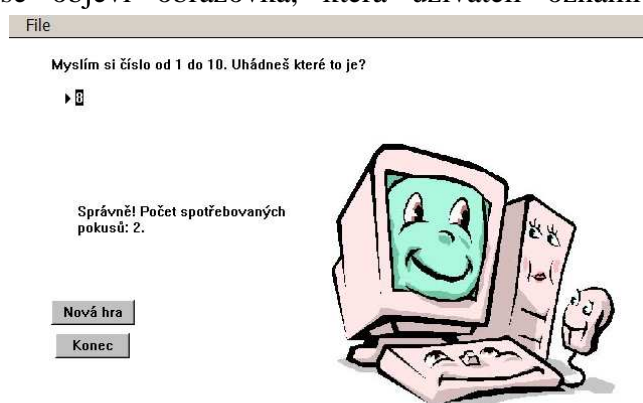
Například pro ikonu *moc* je počítadlo nastaveno takto:

$\text{Moc} := \text{Moc} + 1$

A nápověda uživateli takto:

To je moc. Zkus to znovu.

Po úspěšném dohrání hry se objeví obrazovka, která uživateli oznámí počet spotřebovaných pokusů.



Obrázek 70: Úspěšné dohrání hry

Ukázka použití funkce Random je k dispozici na přiloženém CD v adresáři Funkce\Random.a7p

Funkce pro formátování čísel.

Funkcemi INT a Round můžeme převádět čísla na celá, resp. je můžeme zaokrouhlovat.

4.3.1.6.11 INT

Použití: $Cislo := INT(x)$

Popis: Tato funkce převede číslo zadané parametrem x na celé. Při použití funkce INT se číslo nezaokrouhlí, ale desetinné číslo je oříznuto.

Parametry: x – číslo, které se má převést na celé.

Příklad: Tuto funkci použijeme budeme-li potřebovat pracovat pouze s celočíselnou hodnotou. Můžeme například vyzvat uživatele, aby zadal svůj věk. V případě, že by zadal např. 11.8, potom by proměnná $Vek := INT(11.8)$ obsahovala hodnotu 11.

4.3.1.6.12 Round

Použití: $Cislo := Round(x, [desetinná\ místa])$

Popis: Funkce Round zaokrouhlí číslo zadané parametrem x na počet desetinných míst specifikovaný nepovinným parametrem *desetinná místa*. Pokud parametr *desetinná místa* ne zadáme, potom je číslo funkcí Round zaokrouhleno na 0 desetinných míst, tzn. na celé číslo.

Parametry:

x - číslo, které se má zaokrouhlit

desetinná místa – počet desetinných míst, na kolik se má číslo zaokrouhlit

Příklad: Zaokrouhlování se používá ve výpočetních úlohách.

4.3.1.7 Časové funkce

Časové funkce umožňují převádět kalendářní data na čísla. Jsou úzce propojeny s časovými proměnnými jako jsou např. Day, Month a Year.

4.3.1.7.1 DateToNum

Použití: $Cislo := DateToNum(den; měsíc; rok)$

Popis: Funkce `DateToNum` převádí datum na číslo. Číslo odpovídá počtu dní od 1. ledna 1970.

Parametry:

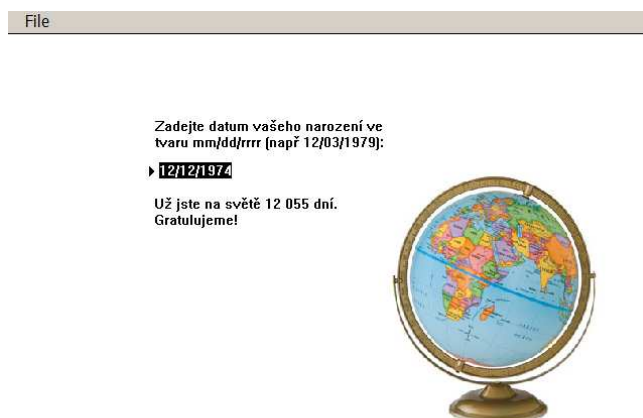
den – den v datumu, ze kterého se počítá číslo

měsíc – měsíc v datumu, ze kterého se počítá číslo

rok – rok v datumu, ze kterého se počítá číslo

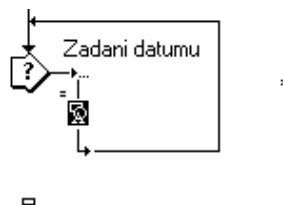
Platné rozmezí dat je od 1. ledna 1970 do 5. února 2036 (25568 do 49710)

Příklad: Pomocí funkce `DateToNum` můžeme počítat časové rozmezí mezi dvěma daty. Níže uvedená aplikace uživateli spočítá počet dní od narození.



Obrázek 71: Počítání rozdílů dat

Strukturogram vypadá takto:



Obrázek 72: Strukturogram počítání rozdílů dat.

V zobrazovací ikoně * je kód, který vypočítá rozdíl mezi aktuálním datem a datem, které zadal uživatel:

```
mesic := NumEntry
den := NumEntry2
rok := NumEntry3
PocetDni := DateToNum(Day; Month; Year) - DateToNum(den;
mesic; rok)
```

Do uživatelské proměnné `PocetDni` se přiřadí rozdíl mezi aktuálním datem (viz funkce `Day`, `Month` a `Year`) a datem které zadal uživatel (pomocí uživatelských proměnných `den`, `mesic` a `rok`).

Ukázka použití funkce `DateToNum` je k dispozici na přiloženém CD v adresáři `Funkce\DateToNum.a7p`

4.3.1.7.2 Date

Použití: `Retezec := Date(číslo)`

Popis: Funkce `Date` převádí celé číslo zadané parametrem číslo na datum. Číslo odpovídá počtu dnů od 1. ledna 1970. Výsledné datum je ve formátu den.měsíc.rok. Funkce `Date` se často používá ve spojení s dalšími systémovými proměnnými a funkcemi, které pracují s daty, obzvláště pak s funkcí `DateToNum`.

Parametry: *číslo* – počet dní od 1. ledna 1970, které se má převést na datum. Platné rozmezí je od 25568 do 49709 (od 1. ledna 1970 do 5. února 2036).

Příklad: Spolu s funkcí `DateToNum` má funkce `Date` využití, když chceme pracovat s daty v číselné podobě (například při počítání uběhlého času).

4.3.1.7.3 Day

Použití: `Cislo := Day(číslo)`

Popis: Funkce `Day` převádí číslo zadané parametrem na číslo dne. Číslo se počítá od 1. ledna 1900, platné rozmezí je od 25568 do 49709 (od 1. ledna 1970 do 5. února 2036).

Parametry: *číslo* – číslo dne od 1.ledna 1900

Příklad: Následující výraz vrátí hodnotu dne pro 1. ledna 1970:

```
Vysledek := Day(25568)
```


Proměnná `Vysledek` bude mít hodnotu 1. Další použití viz funkce `DateToNum`.

4.3.1.7.4 FullDate

Použití: `Retezec := FullDate (číslo)`

Popis: Funkce `FullDate` převádí číslo zadané parametrem na řetězec obsahující anglický zápis datumu.

Parametry: *číslo* – číslo dne od 1. ledna 1900 pro které se má vypočítat anglické datum. Platné rozmezí je od 25568 do 49709 (od 1. ledna 1970 do 5. února 2036)

Příklad: Vzhledem k tomu, že MAW ve funkci `FullDate` používá anglické názvy měsíců, tak je tato funkce použitelná především v didaktických aplikacích zabývajících se angličtinou.

Například ve výrazu `Datum := FullDate(25570)` by uživatelská proměnná `Datum` obsahovala řetězec 3. January 1970.

4.3.1.7.5 Month

Použití: `Cislo := Month(číslo)`

Popis: Funkce `Month` převádí číslo zadané parametrem na číslo měsíce počítáním dní od 1. ledna 1900. Platné rozmezí je od 25568 do 49709 (od 1. ledna 1970 do 5. února 2036).

Parametry: *číslo* – číslo dne od 1.ledna 1900

Příklad: Následující výraz vrátí hodnotu měsíce pro 1. leden 1970:

`Vysledek := Month(25568)`

Uživatelská proměnná `Vysledek` bude mít hodnotu 1. Další použití viz funkce `DateToNum`.

4.3.1.7.6 Year

Použití: `Cislo := Year(číslo)`

Popis: Funkce `Year` převádí číslo zadané parametrem na číslo roku počítáním dní od 1. ledna 1900. Platné rozmezí je od 25568 do 49709 (od 1. ledna 1970 do 5. února 2036).

Parametry: *číslo* – číslo dne od 1.ledna 1900

Příklad: Následující výraz vrátí hodnotu roku pro 1. ledna 1970:

```
Vysledek := Year(25568)
```

Uživatelská proměnná `Vysledek` bude mít hodnotu 1. Další použití viz funkce `DateToNum`.

4.3.1.8 Video funkce

Pomocí těchto funkcí můžeme ovlivňovat způsob, jakým jsou přehrávány sekvence filmů.

4.3.1.8.1 VideoDisplay

Použití: `VideoDisplay(stav)`

Popis: Funkce `VideoDisplay` slouží k vypínání a zapínání obrazu videa.

Parametry: `stav` – nabývá hodnot `on`(zapnuto) nebo `off`(vypnuto)

Příklad: Pomocí této funkce můžeme vypnout přehrávání obrazu videa při zachování přehrávání zvukové stopy videa.



Obrázek 73: Vypínání obrazu videa

Uživateli dáme možnost výběru pomocí tlačítkové nabídky. Její správný chod zrealizujeme pomocí funkce `VideoDisplay`. V kalkulační ikoně pro zapnutí obrazu videa (*Ano (on)*) bude kód `VideoDisplay(on)` a v kalkulační ikoně pro vypnutí obrazu videa (*Ne (off)*) bude kód `VideoDisplay(off)`.

4.3.1.8.2 VideoPause

Použití: `VideoPause()`

Popis: Funkce `VideoPause` zastavuje přehrávání videa na aktuálním snímku.

Parametry: Tato funkce nemá žádné parametry.

Příklad: Pomocí této funkce můžeme pozastavit přehrávání videa.

4.3.1.8.3 VideoPlay

Použití: `VideoPlay(snímek)`

Popis: Funkce `VideoPlay` umožňuje nastavit konečný snímek přehrávání videa tzn. video bude přehráno až do pozice definované parametrem *snímek*. Pokud je číslo parametru snímek menší než aktuální snímek, potom se video přehraje pozpátku.

Parametry: *snímek* – konečný snímek po který se má video přehrát

Příklad: Pomocí této funkce můžeme uživatelům podle zvolených parametrů přehrát určitou část videa. Např. výraz `VideoDisplay(800)` zajistí, že se video přehraje až po snímek 800.

Výraz `Test(New User; VideoPlay(1200), Quit(0))`

Pomocí funkce `Test` zjistí, je-li uživatel nový (systémová proměnná `New User` je pravdivá) a pustí mu videosekvenci po snímek 1200. Pokud uživatel není nový, tak se MAW ukončí (pomocí funkce `Quit`).

4.3.1.8.4 VideoSpeed

Použití: `VideoSpeed(rychlost)`

Popis: Funkce `VideoSpeed` umožňuje měnit rychlost přehrávání videa podle zadaného parametru *rychlost*. Parametr *rychlost* může nabývat hodnot od -5 do +5.

Parametry: rychlost

-5	nejrychlejší přehrávání pozpátku
-4	rychlé přehrávání pozpátku
-3	přehrávání pozpátku
-2	pomalé přehrávání pozpátku
-1	nejpomalejší přehrávání pozpátku
0	pozastavení přehrávání
1	nejpomalejší přehrávání
2	pomalé přehrávání
3	normální přehrávání
4	rychlé přehrávání
5	nejrychlejší přehrávání

Příklad: Pomocí této funkce můžeme dát uživateli kontrolu nad rychlostí přehrávání videa. To má své opodstatnění zejména ve chvíli, kdy uživatel již přehrávané video viděl a může ho shlédnout v rychlejším tempu než uživatel, který zmiňované video vidí poprvé.

5. Soubory počítačových aplikací pro pochopení práce v autorském systému Macromedia Authorware

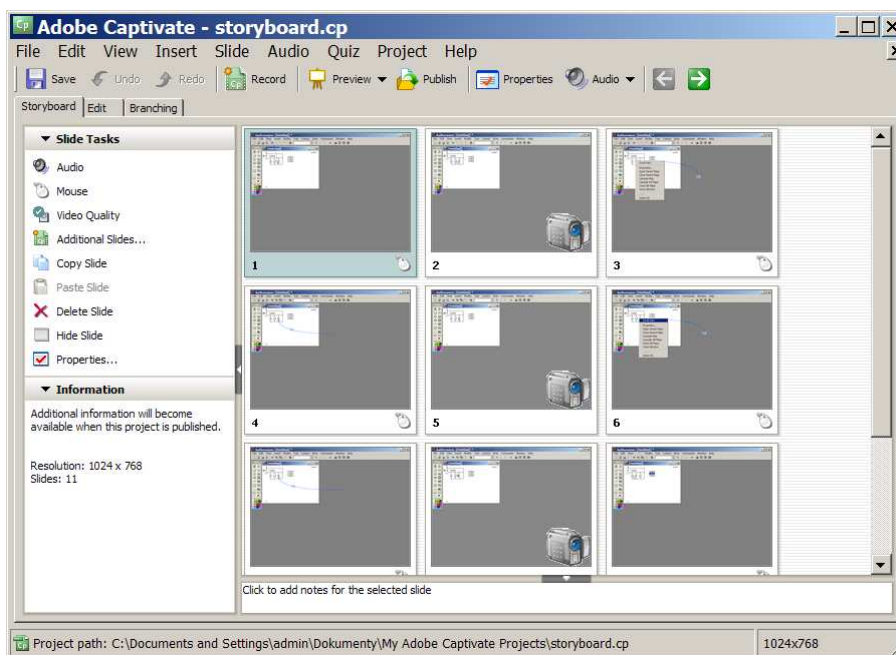
Praktická část mojí diplomové práce se skládá ze dvou částí. První část tvoří soubor výukových videí určených vyučujícím, kteří budou mít zájem svépomocně tvořit didaktické počítačové aplikace typu procvičovací testy v MAW.

Druhou, podpůrnou část k videozáznamům, tvoří pracovní archy. Jsou didaktickou transformací videozáznamu a představují návod pro práci s MAW v situaci, kdy již učitel nemá přístupný videozáznam a musí využívat znalosti, získané při jeho shlédnutí.

5.1 Postup konstrukce výukových videí pro pochopení práce v autorském systému Macromedia Authorware

Výuková videa pro praktickou část diplomové práce jsem natočil v programu Adobe Captivate (dříve Macromedia Captivate) a poté jsem je převedl do prostředí MAW pomocí knihovny `captivate_control` od společnosti GanTek Multimedia.

Captivate je program sloužící k snímání obrazovky a ukládání filmu do formátu Macromedia Flash (.swf). Macromedia Captivate dovoluje namluvit k výukovým videům komentáře a tím zvýšit jejich účinnost.

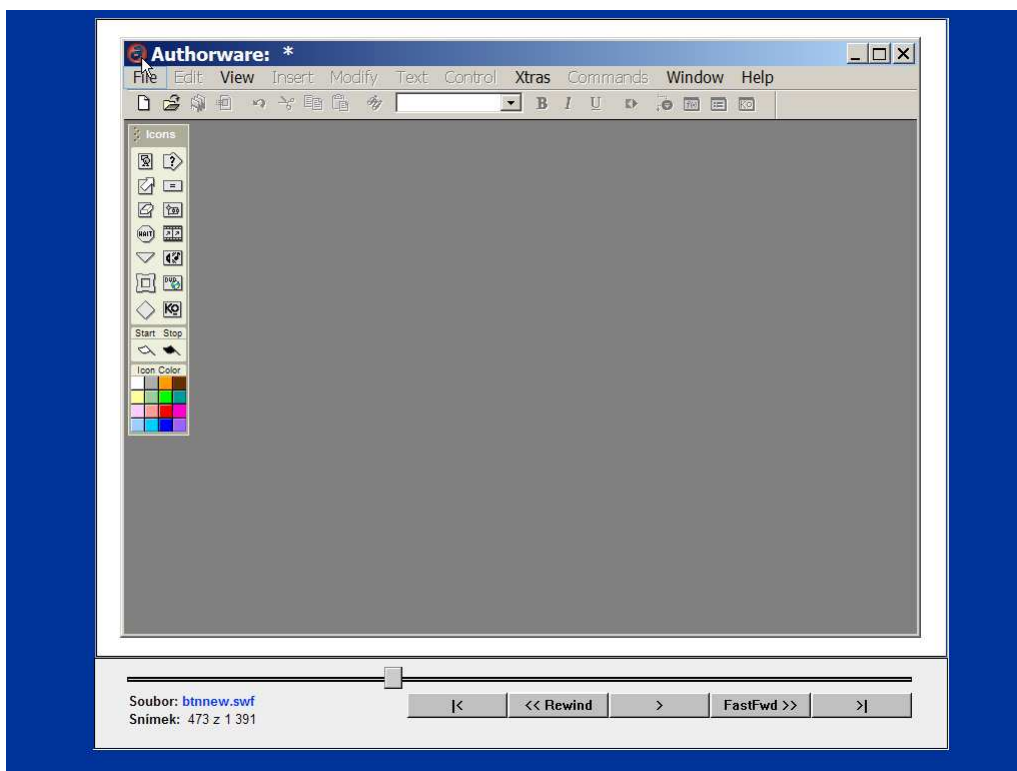


Obrázek 74: Pracovní prostředí Adobe Captivate

Práce v Adobe Captivate probíhá po jednotlivých snímcích. Při práci s ním bylo tedy nezbytné připravit si pro natáčená videa scénáře.

Většinou jsem při tvorbě videí natáčel nejdříve samotný obraz a poté jsem k jednotlivým snímkům přidával komentář. Tato práce byla časově náročná, ale výsledný efekt myslím si, stojí za vynaloženou námahu. Myslím, že vytvořená prezentace je pro budoucího uživatele snadná na pochopení (díky zapojení více smyslů) a dostatečně motivující (kvůli realističnosti).

Když jsem dokončil natáčení souboru v Adobe Captivate, naimportoval jsem pomocí knihovny `captivate_control` vytvořený `swf` soubor do prostředí MAW. Poté jsem musel udělat několik změn v rozhraní knihovny (např. překlad ovládacích a informačních prvků). Dále jsem nastavil cestu k souboru `swf` a vyzkoušel funkčnost ovládacích prvků rozhraní při přehrávání videa. Jako poslední krok při tvorbě videa jsem rozhraní knihovny a videosoubor překonvertoval na samospustitelný `.exe` soubor.



Obrázek 75: Uživatelské rozhraní *captivate_control*

Ukázkové příklady úloh jsou určeny pro žáky 2. stupně ZŠ. Z didaktického hlediska je však možné vzory při změně zadání použít i pro nižší stupeň. Cílem projektu bylo pro učitele vytvořit souhrn typů úloh procvičovacích testů. Soubor je rozdělen do šesti částí podle typů jednotlivých úloh. Jsou jimi button – tlačítkový typ, hot spot – aktivní plocha, hot object – aktivní objekt, target area – cílová oblast, keypress – stisk klávesy a text entry – textová odpověď. Nyní se budu blíže zabývat pracovními archy a typy jednotlivých úloh.

5.2 Pracovní archy k výukovým videům o konstrukci didaktických aplikací v autorském systému Macromedia Authorware

Mnou vytvořené pracovní archy jsou zamýšlené jako podpůrný prostředek pro vytvořená videa o tvorbě didaktických aplikací v MAW. Nyní budou následovat úryvky pracovních archů k videům. Pokud neuvedu jinak, text v uvozovkách napsaný kurzívou bude patřit citaci fráze pracovního archu úlohy, o které se právě zmiňuji.

5.2.1 Pracovní arch Autorizace didaktické aplikace

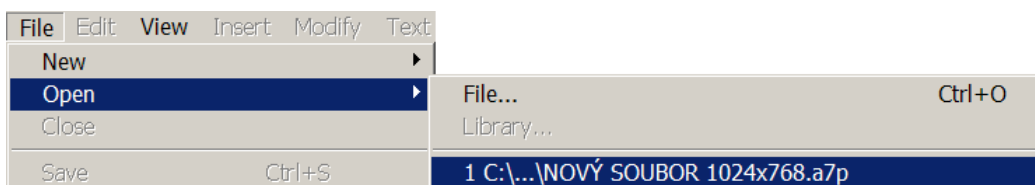
Autorizace didaktické aplikace je nedílnou součástí tvoření aplikace. Autorizací v tomto konkrétním případě rozumíme nejen poskytnutí informací o autorovi aplikace (jako je jeho jméno, kontaktní údaje apod.), ale také zadání základních údajů o vytvářeném procvičovacím testu, jako je například název výukového předmětu, název tématu testu a počet úkolů v testu. Tato část (autorizace aplikace) je shodná pro všechny typy úloh (button, hot object apod.). Proto jsem se rozhodl pro zvýšení přehlednosti pracovních archů tím, že jsem pro tuto část vytvořil samostatné výukové video a pracovní arch. To povede ke zmenšení rozsahu pracovních archů jednotlivých typů úloh, protože v nich nebude tato společná část zahrnuta a v každém pracovním archu úlohy bude pouze slovní odkaz na pracovní arch o autorizaci aplikace.

Uživatelé budou aplikaci zpracovávat podle pracovního archu číslo 1 následujícího znění:

Pracovní arch č. 1

Autorizace didaktické aplikace

- 1) V Macromedia Authorware **otevřete soubor "NOVÝ SOUBOR 1024x768.a7p"**, přes nabídku File/Open/.



- 2) V otevřeném souboru **poklepejte levým tlačítkem myši na první kalkulační ikonu** a otevřete ji.



OTEVŘETE TUTO IKONU A VEPIŠTE ÚDAJE O AUTOROVÍ APLIKACE, POČTU ZADANÝCH ÚKOLŮ, NÁZVU VÝUK. PŘEDMĚTU, NÁZVU TÉMATU aj.

- 3) **Doplňte požadované údaje** o autorovi aplikace (*Jméno, E-mail, Instituci a Datum* vytvoření aplikace).


```

-- D O P L Ň T E ÚDAJE O AUTOROVĚ APLIKACE
-- AUTOR aplikace:                E-mail:
-- INSTITUCE:
-- DATUM:

```

4) Dále **dopíšte odkazy** na textové a obrázkové zdroje použité ve Vaší aplikaci.

```

-- D O P I Š T E ODKAZY NA PŘEDLOHY TEXTOVÉ a OBRÁZKOVÉ:
-- Obsah aplikace čerpán z následujících zdrojů:
-- 1. ....
-- 2. ....
--
-- Obrázky čerpány z následujících zdrojů:
-- www. ..... obrázky úloh č. .. ..
-- www. ..... obrázky úloh č. .. ..

```

5) **Přepište hodnoty proměnných** *VyukovyPredmet*, *NazevTematu* a *PocetZadanychUkolu* hodnotami, které vyhovují Vaší aplikaci.

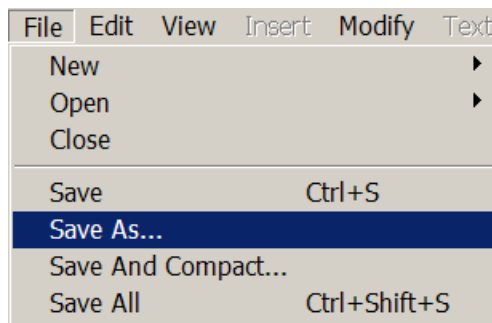
```

-- P Ř E P I Š T E V NÁSLEDUJÍCÍCH třech ŘÁDCÍCH VÝRAZY ZA ROVNÍTKEM:
VyukovyPredmet:="MATEMATIKA" --PŘEPIŠTE v uvozovce slovo "MATEMATIKA" názvem výukového předmětu (např. "CHEMIE")
NazevTematu:="Zlomky"        --PŘEPIŠTE v uvozovce slovo "ZLOMKY" názvem tématu (např. "Směsi")
PocetZadanychUkolu:=12      --PŘEPIŠTE číslíci 12 číslíci POČTU ZADANÝCH ÚKOLŮ (např.16)

```

6) Vyplněnou **kalkulační ikonu zavřete**.

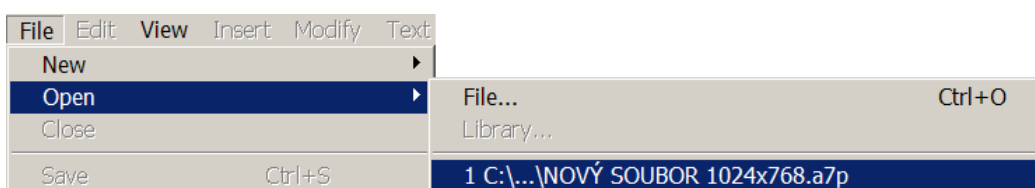
7) **Soubor uložte pod JINÝM názvem**, přes nabídku *File/Save As.../*.



8) Podle instrukcí dalších videonahrávek a dalších pracovních listů budete doplňovat do této předlohy různé typy testových úloh.

5.2.1.1 Komentář k pracovnímu archu číslo 1: Autorizace didaktické aplikace

„V Macromedia Authorware otevřete soubor „NOVÝ SOUBOR 1024x768.a7p“, přes nabídku File/Open.“



Obrázek 76: Otevření Nového souboru

Zvolil jsem formu vykání, protože oslovuji učitele a některým by tykání nemuselo být příjemné. Rozhodl jsem se nešetřit obrázky kvůli názornosti. Prostředí Authorwaru může novému uživateli na první pohled připadat složité, proto poskytuji uživatelům MAW i obrázky nabídek.

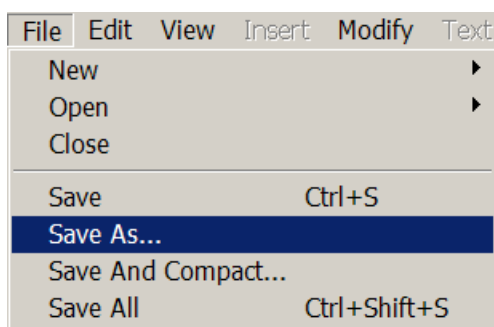
„Doplňte požadované údaje o autorovi aplikace (Jméno, E-mail, Instituci a Datum vytvoření aplikace).“

```
-- D O P L Ň T Ě ÚDAJE O AUTOROVI APLIKACE
-- AUTOR aplikace:                E-mail:
-- INSTITUCE:
-- DATUM:                          ..
```

Obrázek 77: 1. část autorizace

Rozhodl jsem se uživatele příliš nezatěžovat technickými podrobnostmi (např. tím, že vše co je v MAW napsáno za komentáři je ignorováno), protože vytváření aplikací za podpory výukových videí s pomocí pracovních archů nebylo podmiňováno žádnými hlubšími znalostmi MAW.

„Soubor uložte pod JINÝM názvem, přes nabídku File/Save As.../.“



Obrázek 78: Ukládání souboru

Uložení vytvářené aplikace pod jiným jménem je velice důležité, protože kdyby uživatel přepsal soubor předloh, nemohl by vytvářet aplikace další.

5.2.2 Testovací typ tlačítko (button)

V tomto typu úlohy je žákovi je prezentována jednoduchá otázka, na kterou musí odpovědět kliknutím na jedno z nabízených tlačítek. Úloha je konstruována tak, že vždy jen jedna odpověď je správná, zatímco zbylé (ve výchozím nastavení tři) jsou chybné. Pokud žák zvolí nesprávné tlačítko, zobrazí se mu informace o chybě. To ovšem neznamená konec úlohy, neboť žák dostává další šanci na její vyřešení. Ve chvíli, kdy žák vybere správnou možnost, se mu zobrazí informace o správném řešení. A přičte se mu jeden bod. Obě upozornění, jak o chybě, tak o správné odpovědi jsou doprovázeny zvukem.

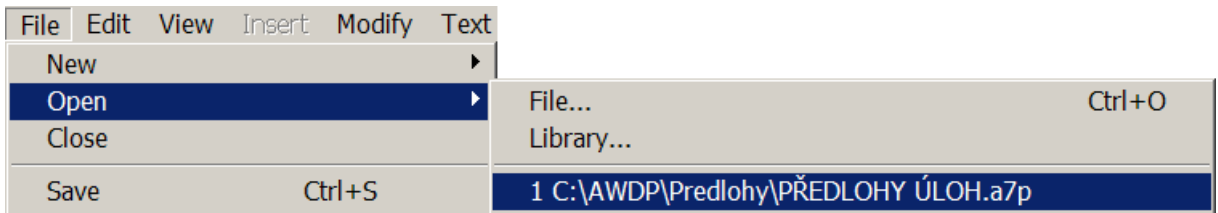
Uživatelé budou aplikaci zpracovávat podle pracovního archu číslo 2 následujícího znění:

Pracovní arch č. 2

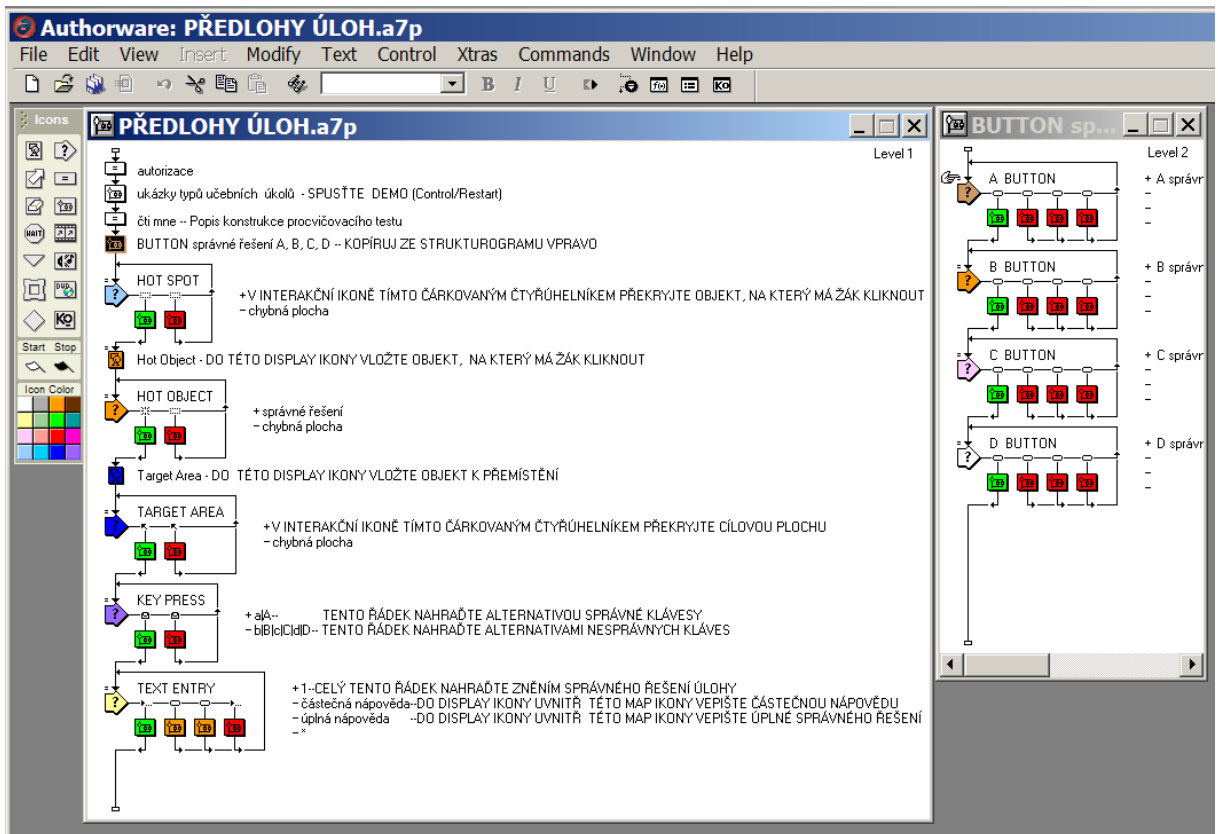
Přidávání a modifikace testovacích úloh typu Button (tlačítko)

Do autorizovaného a přejmenovaného nového souboru (viz pracovní arch č.1) vložíte výběrovou úlohu tlačítkového typu (Button) následujícím postupem:

- 1) Otevřete soubor "PŘEDLOHY ÚLOH.a7p" prostřednictvím nabídky *File/Open/PŘEDLOHY ÚLOH.a7p*

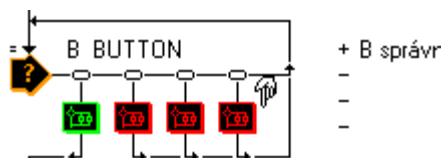
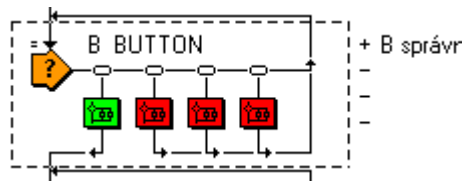


2) Získáte strukturogram s předlohami všech typů úloh:



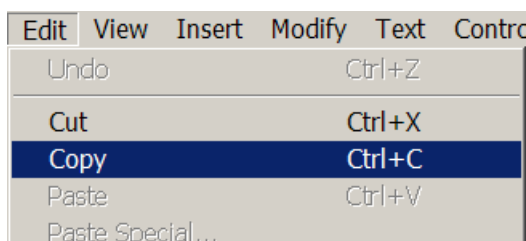
3) Vyhledejte předlohu **BUTTON**. Vpravo vyberte jeden ze čtyř strukturogramů požadované varianty typu **BUTTON** (tlačítko) podle toho, která z odpovědí má být správná (A, B, C nebo D).

4) Strukturogram potřebné předlohy úlohy (např. **B BUTTON**) označte pomocí myši (držte stisknuté levé tlačítko myši, objedťe ikony strukturogramu a závěrem tlačítko pusťte):



5) Ztmavěle označené ikony zkopírujte do schránky

přes nabídku *Edit/Copy*:
nebo je zkopírujte pomocí klávesové kombinace *Ctrl+C*.

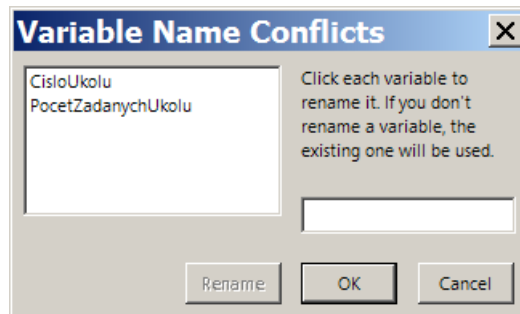


6) Vraťte se do nově vytvořeného, již autorizovaného souboru a poklepnáním myši otevřete žlutou Map ikonu (slučovací ikonu) *TEST*:

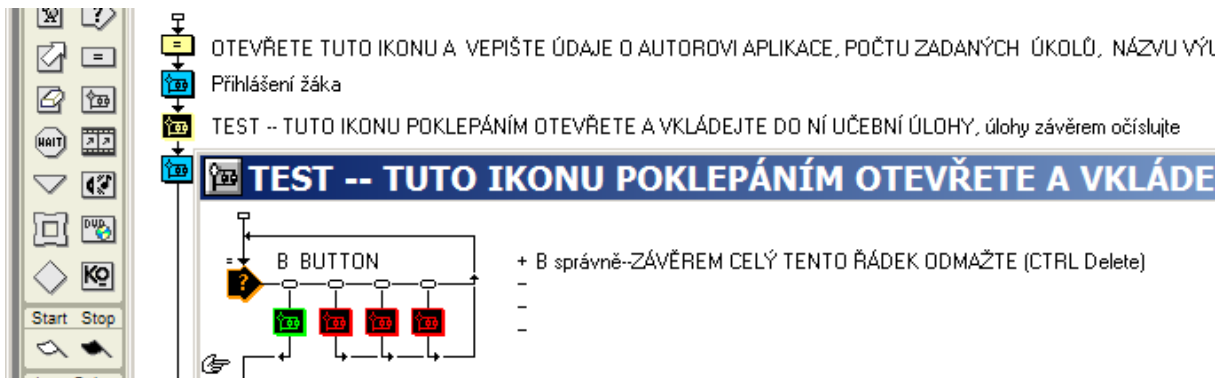


7) Na časovou osu ikony *TEST* vložte ze schránky strukturogram zkopírované úlohy buď přes nabídku *Edit/Paste* nebo pomocí klávesové kombinace *Ctrl+V*.

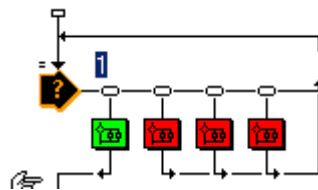
Bude prezentováno dialogové okno pro indexování proměnných: **Vždy klikněte OK.**



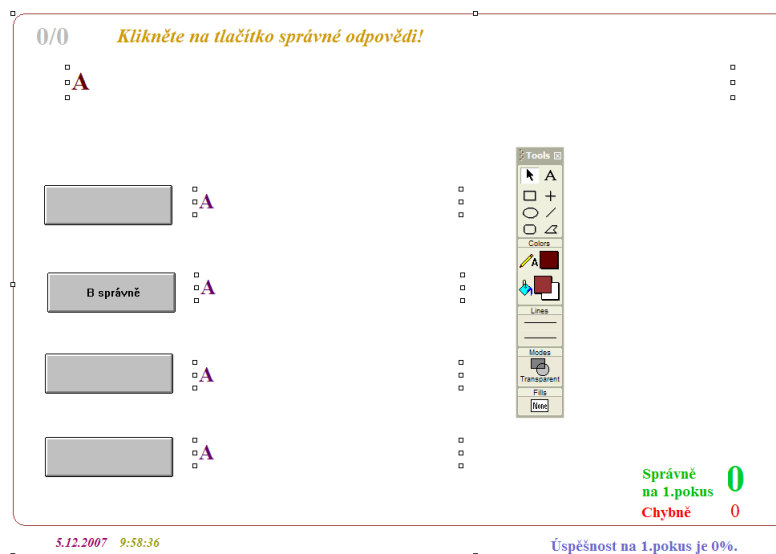
8) Na časové ose ikony *TEST* se objeví strukturogram požadované předlohy úloh:



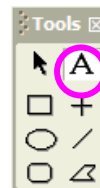
- 9) Kliknutím interakční ikonu označte a **přejmenujte ji pořadovým číslem úlohy**
(např. 1):



- 10) Otevřete *interakční ikonu* poklepnutím.



Pomocí nástroje *Text (A)* získajte psací kurzor.



- 11) V prostoru zadání úlohy psacím kurzorem klikněte na písmeno A. Písmeno A odmažte pomocí klávesy

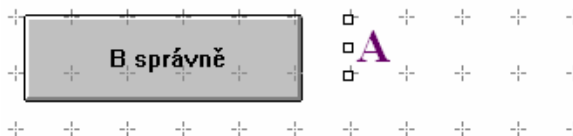


Backspace (tím zachováte formátovací charakteristiky).

- 12) V místě původního písmene A **vepište zadání úlohy:**



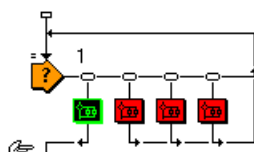
- 13) U tlačítka správné odpovědi písmeno A odmažte pomocí klávesy *Backspace*.



Vepište znění správné varianty řešení úlohy.

- 14) U tlačítek nesprávných odpovědí postupně písmeno A odmažte pomocí klávesy *Backspace* a vepište znění nesprávné varianty.

- 15) **Uzavřete interakční ikonu** stiskem kláves



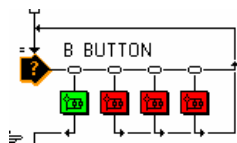
+ B správně-ZÁVĚREM CELÝ TENTO ŘÁDEK ODMAŽTE (CTRL Delete)

CTRL+J.

- 16) Pokud potřebujete odstranit v úloze některá tlačítka, **odmazávejte (DELETE) zprava červené Map ikony** (slučovací ikony).

17) Dále se řiďte pokynem psaným velkými písmeny:

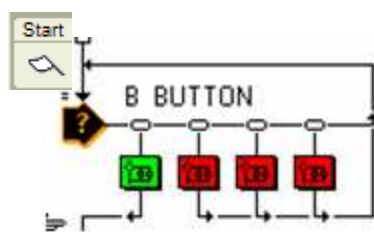
**Označte
horní řádek
kliknutím a**



+ B správně--ZÁVĚREM CELÝ TENTO ŘÁDEK ODMAŽTE (CTRL Delete)

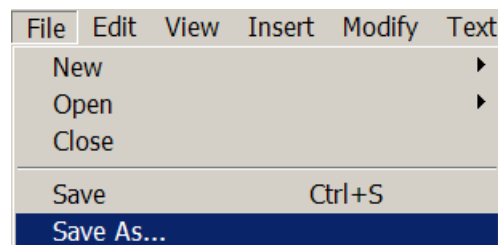
odstraňte ho stiskem kláves *CTRL+DELETE*. Tím odstraníte označení tlačítka správné odpovědi.

18) Umístěte před interakční ikonu bílý praporek a spusťte aplikaci:



19) Pověřte bezchybný běh aplikace. Případné nedostatky opravte.

20) **Bílý praporek odstraníte** kliknutím do lišty ikon. **Aplikaci znovu uložte prostřednictvím nabídky File/Save As/...**



21) Pokud chcete do aplikace vložit další tlačítkovou úlohu, **opakujte postup popsaný v bodech 3) až 20).**

22) Pokud chcete do aplikace vložit **jiný typ úlohy** (např. Hot Spot, Target Area) postupujte dle pracovního archu příslušného typu úlohy.

23) Pokud jste s výslednou aplikací spokojeni a **chcete vytvořit plnohodnotnou windowsovou aplikaci (.exe soubor), postupujte podle pracovního archu č. 8 „Vytváření spustitelných aplikací“.**

5.2.2.1 Komentář k pracovnímu archu číslo 2: Přidávání a modifikace testovacích úloh typu Button (tlačítko).

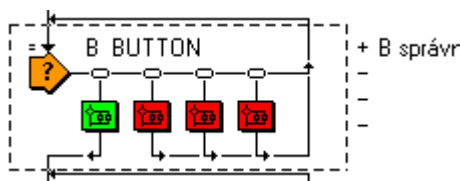
„Do autorizovaného a přejmenovaného nového souboru (viz pracovní arch č.1) vložíte výběrovou úlohu tlačítkového typu (Button) následujícím postupem:“

První bod každého pracovního archu úloh upozorní uživatele na autorizaci vytvářené aplikace. Autorizace je dobrá i proto, že si autor/tvůrce aplikace při vyplňování položek musí důkladně promyslet styl a náplň své aplikace.

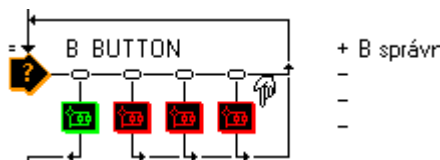
„Vpravo vyberte jeden ze čtyř strukturogramů požadované varianty typu **BUTTON** (tlačítko) podle toho, která z odpovědí má být správná (A, B, C nebo D).

V předlohách doc. Nikla nemusí uživatel řešit, jak změnit chování tlačítkového typu podle toho, které z tlačítek má být správné. Jednoduše si vybere jednu ze čtyř předdefinovaných variant a tu použije. (A, B, C nebo D)

„Strukturogram potřebné předlohy úlohy (např. B **BUTTON**) označte pomocí myši (držte stisknuté levé tlačítko myši a objed'te ikony strukturogramu...).“



„Ztmavělé označené ikony zkopírujte...“



Jak již jsem napsal dříve, při návrhu pracovních archů jsem se rozhodl pro maximální názornost, z toho důvodu uvádím i jemné nuance vybírání a označování strukturogramu úlohy.

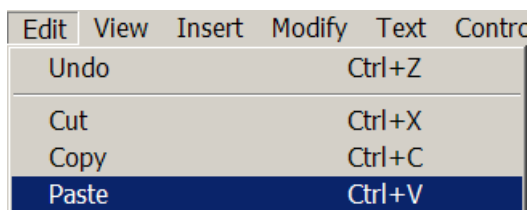
„Vraťte se do nově vytvářeného, již autorizovaného souboru a poklepáním myši otevřete žlutou Map ikonu (slučovací ikonu) **TEST**.“



Obrázek 79: Slučovací ikona **TEST** do které se vkládají úlohy

Opět pro lepší názornost jsem se v pracovních arších rozhodl dodržovat velikosti písmen názvů přesně s velikostí písmen použitou v souboru ukázkových úloh od doc. Nikla.

„...bud' přes nabídku Edit/Paste nebo pomocí klávesové kombinace Ctrl+V“



Obrázek 80: Vkládání ze schránky

V případech, kdy jsem popisoval klávesovou zkratku, jsem používal i její ekvivalent dosažitelný přes menu Authorwaru.

„V místě původního písmene A vepište zadání úlohy“



Obrázek 81: Formulace otázky

Při zadávání otázek jsem dbal na to, aby z popisu nad otázkami a ze stylu otázek mohl autor didaktické aplikace vycítit, co vlastně ten který druh procvičovací úlohy dělá (např. při formulaci otázky u typu Button jsem do popisu zapojil frázi „klikněte na tlačítko“, při zadávání otázky u typu Key Press (stisk klávesy) jsem v popisu použil formulaci „odpovězte stiskem klávesy“ atd.) Zároveň jsem dbal na to, aby na ilustračním obrázku byl zachycen nástroj Text a kurzor v editačním módu.

„Aplikaci znovu uložte prostřednictvím nabídky File/Save As...“

Včasné ukládání rozdělané práce je důležité a mělo by patřit k základním návykům uživatelů počítače. Proto jsem se rozhodl ukládání připomínat.

5.2.3 Typ Hot Spot (aktivní plocha)

Hot Spot – aktivní plocha je interakční plocha na obrazovce. V tomto typu úloh musí žák kliknout na správné místo na obrazovce. Při tvorbě této úlohy je velmi důležitý pečlivý výběr obrázků, tak aby byly jednoduše a jednoznačně identifikovatelné. Dobré bývají obrázky konkrétních podstatných jmen. Systém bodování a oprav je stejný jako u úlohy typu Button. Také oznámení o chybách je obdobné. Je důležité si uvědomit, že v některých případech by mohlo oznámení o chybě/správné odpovědi zaniknout na pozadí zaplněném nejrůznějšími obrázky. Je proto důležité věnovat umístění upozornění o správnosti odpovědi zvýšenou pozornost.

Uživatelé budou aplikaci zpracovávat podle pracovního archu číslo 3 následujícího znění:

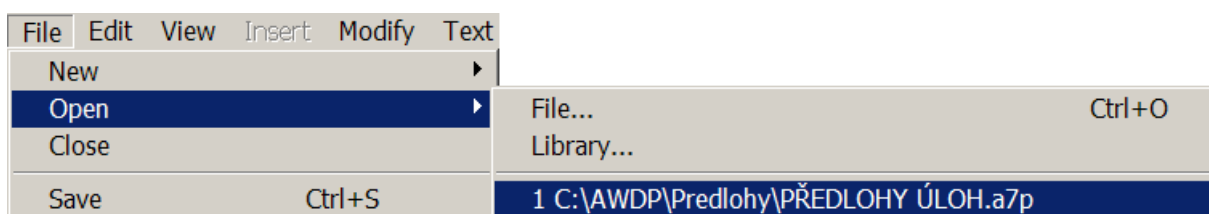
Pracovní arch č. 3

Přidávání a modifikace testovacích úloh typu Hot Spot (aktivní plocha)

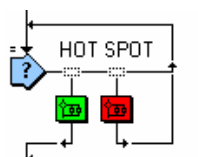
Do autorizovaného a přejmenovaného nového souboru (viz pracovní arch č.1) vložíte výběrovou úlohu typu Hot Spot (aktivní plocha) následujícím postupem:

- 1) Otevřete soubor "**PŘEDLOHY ÚLOH.a7p**" přes nabídku *File/Open/*

PŘEDLOHY ÚLOH.a7p:

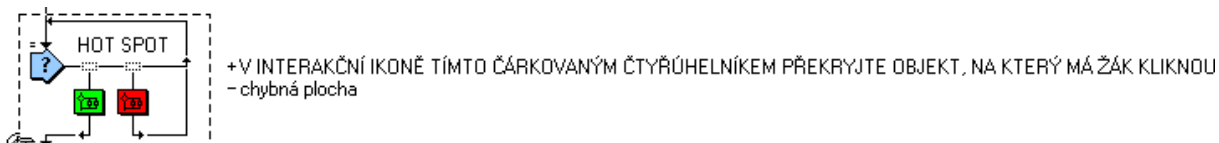


- 2) V získaném strukturogramu všech předloh **vyhledejte strukturogram úlohy typu *HOT SPOT***:

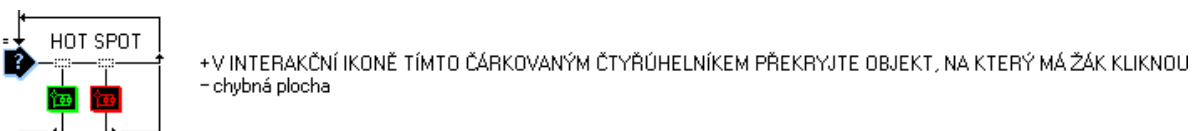


+ V INTERAKČNÍ IKONĚ TÍMTO ČÁRKOVANÝM ČTYŘÚHELNÍKEM PŘEKRYJTE OBJEKT, NA KTERÝ MÁ ŽÁK KLIKNI
- chybná plocha

- 3) **Strukturogram předlohy úlohy typu *HOT SPOT* označte pomocí myši** (držte stisknuté levé tlačítko myši a objedťte strukturogram, závěrem tlačítko pusťte):



Ikony strukturogramu předlohy po označení ztmavnou:



- 4) **Zkopírujte strukturogram předlohy úlohy typu *HOT SPOT* do schránky** pomocí klávesové kombinace *Ctrl+C* (nebo přes nabídku *Edit/Copy*).

- 5) **Vraťte se do nově vytvářeného, již autorizovaného souboru** a poklepáním myši otevřete žlutou Map ikonu (slučovací ikonu) *TEST*:

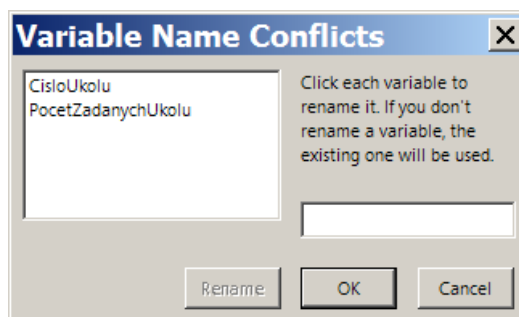


tuto ikonu otevřete:

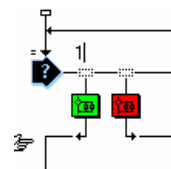


- 6) **Na časovou osu ikony *TEST* vložte ze schránky strukturogram zkopírované úlohy *HOT SPOT*** buď pomocí klávesové kombinace *Ctrl+V* (nebo přes nabídku *Edit/Paste*).

Nejprve Vám bude prezentováno dialogové okno indexování proměnných - klikněte v něm na tlačítko *OK*:

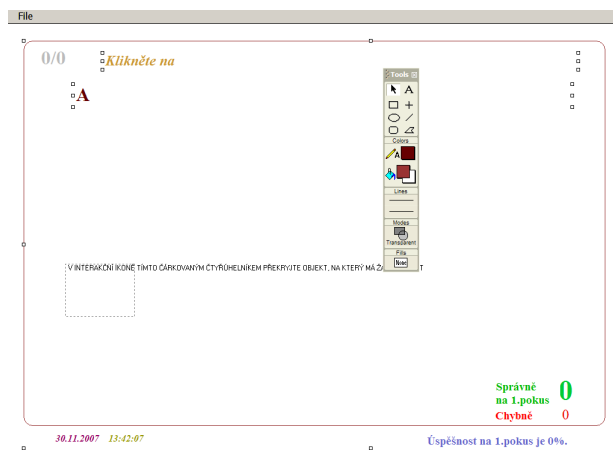


- 7) Kliknutím interakční ikonu označte a **přejmenujte ji pořadovým číslem úlohy** (např. 1):

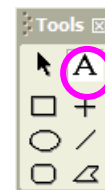


+V INTERAKČNÍ IKONĚ TÍMTO
- chybná plocha

- 8) Poklepnutím myší **otevřete interakční ikonu**:



- 9) V panelu nástrojů (Tools) **klikněte na nástroj Text (reprezentovaný písmenem A)**. Šipkový kurzor se změní na kurzor textový (písmeno velké I):

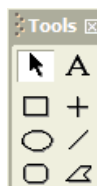


- 10) V horní části, vymezené pro zadání úlohy, **klikněte textovým kurzorem zprava vedle velkého písmene A a odmažte ho klávesou *Backspace*** – tím převezmete do psacího kurzoru formátovací charakteristiky (font, barvu a velikost písma atd.).

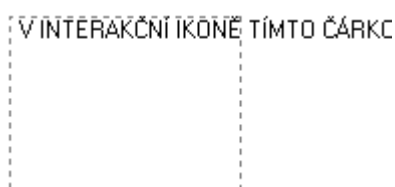
Pokračujte zápisem zadání učební úlohy a případným dalším textem, např.:



- 11) Kliknutím na blokovou šipku v panelu nástrojů **změňte kurzor na blokovou šipku**:



- 12) Přes nabídku **Insert/Image** vložte obrázek, který bude reprezentovat správnou odpověď
- 13) Uchopte **blokovým kurzorem** nápis nad čárkovaným čtyřúhelníkem a přetáhněte ho na objekt správného řešení (např. na obrázek), na který má žák kliknout:



- 14) Obdobným způsobem (přes nabídku **Insert/Image**) vložte obrázky chybných odpovědí. Umístěte je MIMO čárkovaný čtyřúhelník!

15) Uzavřete interakční ikonu stiskem kláves **CTRL+J**.

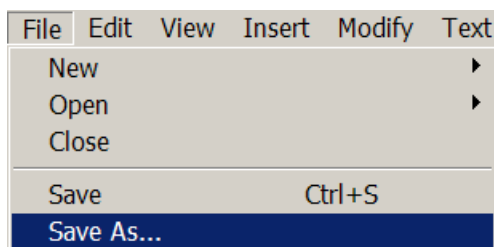
16) Umístěte před interakční ikonu **bílý praporek**



17) Pověřte bezchybný běh aplikace. Případné nedostatky opravte.

18) Uzavřete interakční ikonu stiskem kláves **CTRL+J**.

19) **Bílý praporek** odstraňte kliknutím do lišty ikon. **Aplikaci znovu uložte** prostřednictvím nabídky **File/Save As/...**



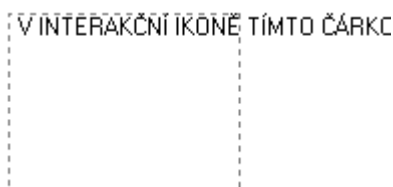
20) Pokud chcete do aplikace vložit další úlohy typu **HOT SPOT**, opakujte postup popsany v bodech 2) až 16).

21) Pokud chcete do aplikace vložit **jiný typ úlohy** (např. Target Area, Text Entry aj.), postupujte dle pracovního archu příslušného typu úlohy.

22) Pokud jste ukončili konstrukci výsledné procvičovací aplikace ke své spokojenosti a přejete si vytvořit plnohodnotnou windowsovou žakovskou aplikaci (.exe soubor), postupujte podle *Pracovního archu č. 8 Vytváření spustitelných aplikací*.

5.2.3.1 Komentář k pracovnímu archu č. 3: Přidávání a modifikace testovacích úloh typu Hot Spot (aktivní plocha)

„Uchopte blokovým kurzorem nápis nad čárkovaným čtyřúhelníkem a přetáhněte ho na objekt správného řešení (např. na obrázek), na který má žák kliknout:“



Obrázek 82: Hot Spot(aktivní plocha)

V typech úloh vázaných na interakci s určitou částí obrazovky je důležité, aby si autor aplikace pracující podle pracovního archu dokázal představit, jak je klíčová oblast na obrazovce značena. Proto jsem sáhl k vytvoření snímku této oblasti.

5.2.4 Typ Hot object (aktivní objekt)

Typ úloh *Hot Object (aktivní objekt)* se pro uživatele z hlediska funkčnosti nijak neliší od typu Hot Spot. Je ale konstrukčně dokonalejší, neboť umožňuje interakčním oblastem přesněji definovat hrany (nemusí se jednat pouze o pravoúhlou oblast). V této úloze má žák za úkol zvolit jednu z nabízených variant. Správným řešením bývá poměrně malá oblast. Proto je vhodné nastavit styl objektu na neprůhledný. Tím se bude za správnou odpověď brát i blízké okolí a ne jen objekt sám.

V případě nesprávné odpovědi se na obrazovce zobrazí informace o chybném řešení úkolu a žák dostává další možnost odpovědi. Tyto možnosti se opakují, dokud není otázka zodpovězena správně. Při správném zodpovězení otázky se zobrazí informace o úspěšném vyřešení úlohy. Za každou správnou odpověď napoprvé získává žák 1 bod.

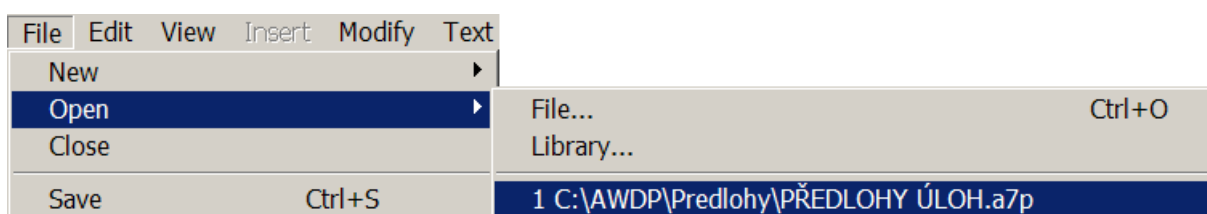
Uživatelé budou aplikaci zpracovávat podle pracovního archu číslo 4 následujícího znění:

Pracovní arch č. 4

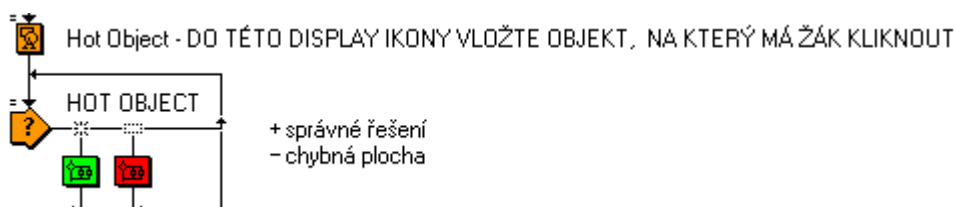
Přidávání a modifikace testovacích úloh typu Hot Object (aktivní objekt)

Do autorizovaného a přejmenovaného nového souboru (viz pracovní arch č.1) vložíte výběrovou úlohu typu Hot Spot (aktivní plocha) následujícím postupem:

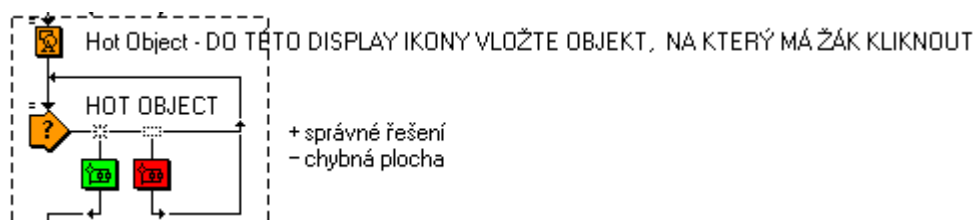
- 1) Otevřete soubor "**PŘEDLOHY ÚLOH.a7p**" přes nabídku *File/Open/*
PŘEDLOHY ÚLOH.a7p:



- 2) V získaném strukturogramu všech předloh **vyhledejte strukturogram** úlohy typu *HOT OBJECT*:

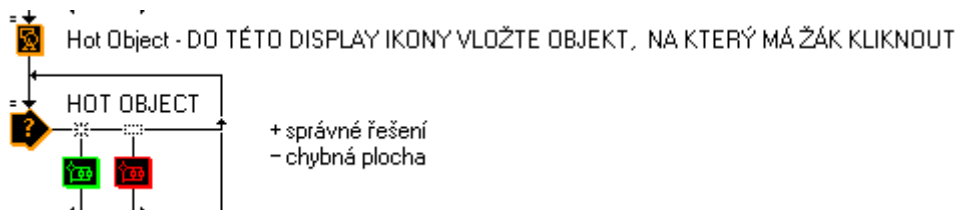


- 3) Strukturogram předlohy úlohy typu *HOT OBJECT* **označte pomocí myši** (držte stisknuté levé tlačítko myši a objedťte strukturogram, závěrem tlačítko pusťte - pozor, je nutné označit také zobrazovací ikonu objektu, na který má žák



kliknout):

Ikony strukturogramu předlohy po označení **ztmavnou**:



- 4) **Zkopírujte strukturogram** předlohy úlohy typu *HOT OBJECT* do schránky pomocí klávesové kombinace *Ctrl+C* (nebo přes nabídku *Edit/Copy*).
- 5) **Přejděte do nově vytvářeného** již autorizovaného testovacího souboru.
- 6) Ve strukturogramu nově vytvářeného souboru poklepáním myši na žlutou slučovací **ikonu TEST**:

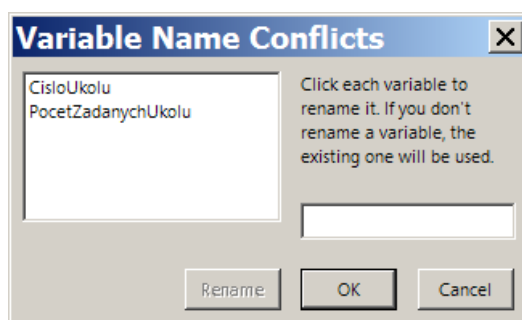


tuto ikonu otevřete:

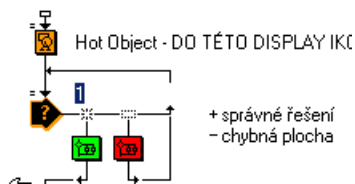


- 7) Na časovou osu ikony **TEST** vložte ze schránky strukturogram zkopírované úlohy *HOT OBJECT* buď pomocí klávesové kombinace *Ctrl+V* (nebo přes nabídku *Edit/Paste*).

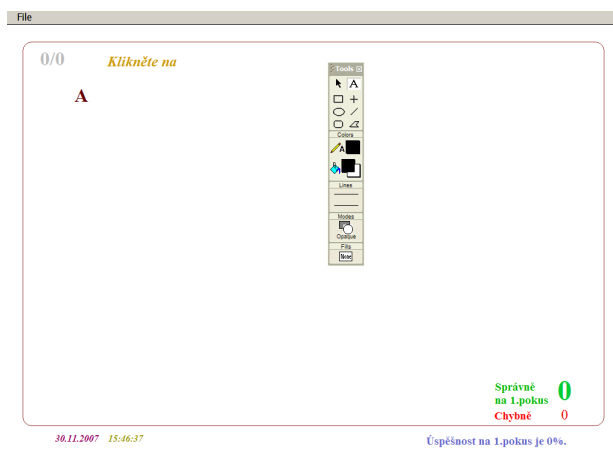
Nejprve Vám bude prezentováno dialogové okno indexování proměnných - klikněte v něm na tlačítko *OK*:



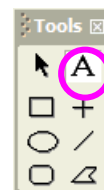
- 8) **Interakční ikonu** vložené předlohy úlohy *HOT OBJECT* přejmenujte **pořadovým číslem úlohy** (např 1):



9) Poklepáním myší **otevřete interakční ikonu:**



10) V panelu nástrojů (Tools) **klikněte na nástroj Text** (reprezentovaný písmenem A). Šipkový kurzor se změní na kurzor textový (písmeno velké I):



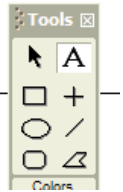
11) V horní části, vymezené pro zadání úlohy, **klikněte textovým kurzorem zprava vedle velkého písmene A** a odmažte ho klávesou *Backspace* – tím převeźmete do psacího kurzoru formátovací charakteristiky (font, barvu a velikost písma atd.).

Pokračujte zápisem zadání učební úlohy a případným dalším textem, např.:

1/12

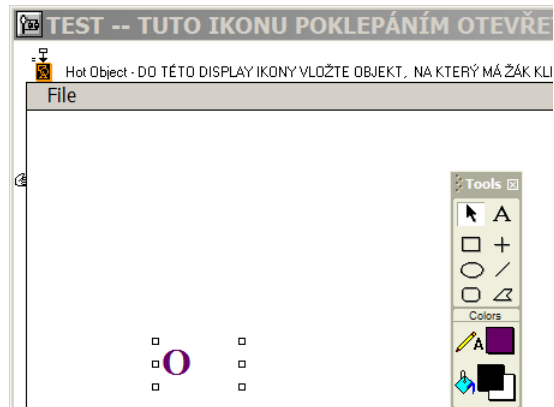
Klikněte na obrázek znázorňující správnou odpověď

Který živočich běží nejrychleji?

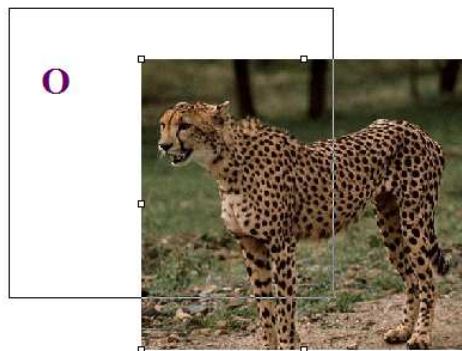


12) Uzavřete interakční ikonu stiskem kláves CTRL+J.

13) Otevřete zobrazovací ikonu:
ikonu:



14) Vložte objekt správného řešení (např. obrázek) přes nabídku **Insert**, a přetáhněte ho na aktivní objekt (znázorněný písmenem O):



15) Uzavřete zobrazovací ikonu stiskem kláves CTRL+J.

16) **Otevřete interakční ikonu** a obdobným způsobem (**přes nabídku Insert**)
vložte na plochu prezentace **obrázky nesprávných odpovědí**.

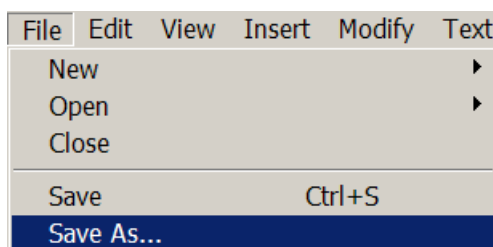
17) **Uzavřete interakční ikonu** stiskem kláves CTRL+J.

18) Umístěte před interakční ikonu **bílý praporek**



19) **Prověřte její bezchybný průběh**. Případné nedostatky opravte.

20) **Bílý praporek odstraňte** kliknutím do lišty ikon. **Aplikaci uložte** přes nabídku
File/Save As/...



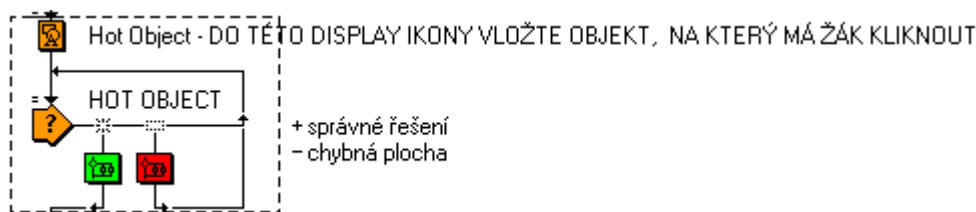
21) Pokud chcete do aplikace vložit **další úlohy typu HOT OBJECT**, opakujte
postup popsany v bodech 2) až 19).

22) Pokud chcete do aplikace vložit **jiný typ úlohy** (např. Target Area, Text Entry
aj.), postupujte dle pracovního archu příslušného typu úlohy.

23) Pokud jste ukončili konstrukci výsledné procvičovací aplikace ke své
spokojenosti a přejete si vytvořit plnohodnotnou windowsovou žákovskou
aplikaci (.exe soubor), postupujte podle *Pracovního archu č. 8 Vytváření
spustitelných aplikací*.

5.2.4.1 Komentář k pracovnímu archu č. 4: Přidávání a modifikace testovacích úloh typu Hot Object (aktivní objekt)

*„...pozor, je nutné označit také zobrazovací ikonu objektu, na který má žák
kliknout.“*



Obrázek 83: Strukturogram úlohy Hot Object

K zvýraznění nutnosti označení zobrazovací ikony mě vedl poněkud atypický tvar strukturogramu ve srovnání se strukturogramy předešlých typů úloh. Já sám jsem napoprvé omylem označil strukturogram HOT OBJECT bez zmíněné zobrazovací ikony.

„... a přetáhněte ho na aktivní objekt (znázorněný písmenem O)“



Obrázek 84: Hot Object

Stejně, jako v předešlém typu úloh, i zde je nutné, aby autor aplikace věděl, jak vypadá objekt, na který má vložit objekt (zpravidla obrázek).

5.2.5 Typ Target Area (cílová oblast)

Dalším typem úlohy je *Target Area* – cílová oblast. V úloze tohoto typu má žák za úkol přesunout určitý objekt. Cílová oblast (Target Area) má tvar obdélníku a na obrazovce zpravidla kopíruje nějaký text nebo část obrázku. Na obrazovce jsou vždy aktivní dvě plochy. Jedna, která překrývá cílovou oblast (správná odpověď) a druhá, která překrývá zbytek obrazovky (chybná odpověď). Je-li objekt přesunut do správné cílové oblasti, potom se žákovi zobrazí informace o tom, že úlohu vyřešil správně a naopak. Pokud by byl objekt přesunut na místo, které nepokrývá ani jedna oblast (chybných a správných odpovědí), pak by přesouvaný objekt zůstal „ležet“ na místě a odpověď by zůstala nevyhodnocena.

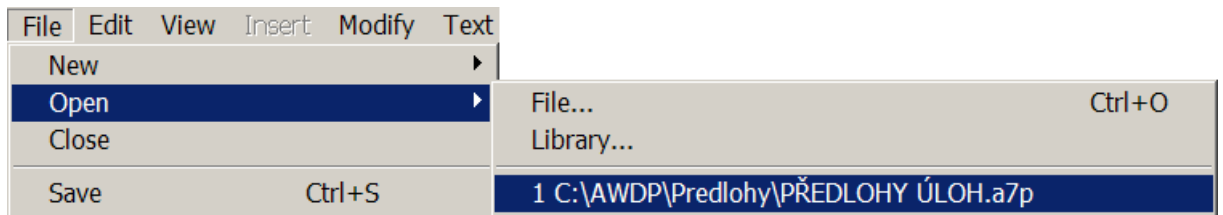
Uživatelé budou aplikaci zpracovávat podle pracovního archu číslo 5 následujícího znění:

Pracovní arch č. 5

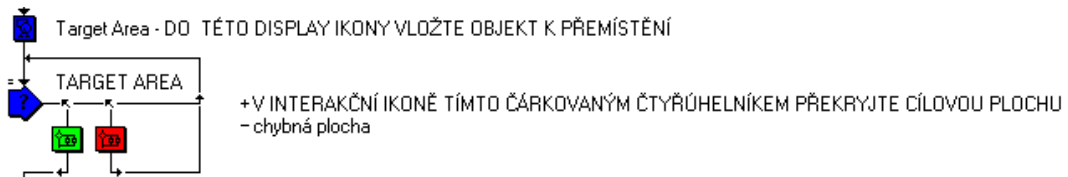
Přidávání a modifikace testovacích úloh typu Target Area (cílová oblast)

Do autorizovaného a přejmenovaného nového souboru (viz pracovní arch č.1) vložíte výběrovou úlohu typu Target Area (cílová oblast) následujícím postupem:

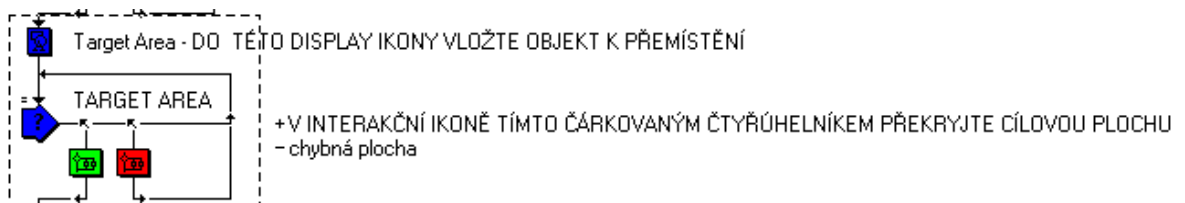
- 1) Otevřete soubor "**PŘEDLOHY ÚLOH.a7p**" přes nabídku *File/Open/*
PŘEDLOHY ÚLOH.a7p:



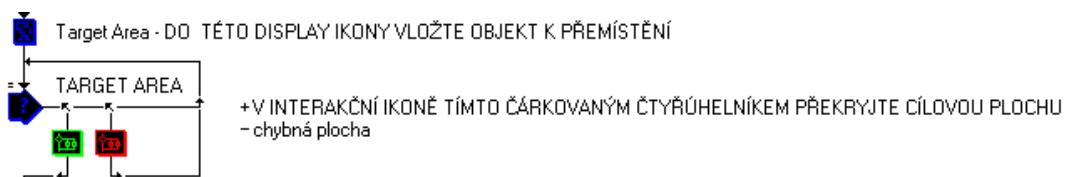
- 2) V získaném strukturogramu všech předloh **vyhledejte strukturogram úlohy typu TARGET AREA**:



- 3) **Strukturogram** předlohy úlohy typu *TARGET AREA* **označte pomocí myši** (držte stisknuté levé tlačítko myši a objeďte strukturogram, závěrem tlačítko pusťte):



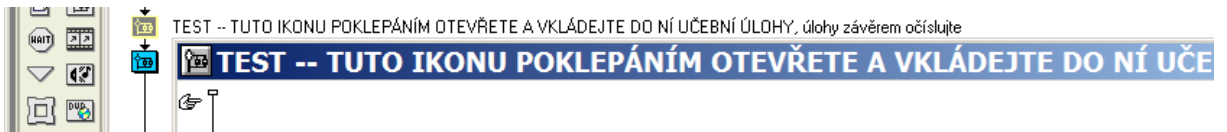
Ikony strukturogramu předlohy po označení **ztmavnou**:



- 4) Zkopírujte strukturogram předlohy úlohy typu *TARGET AREA* do schránky pomocí klávesové kombinace *Ctrl+C* (nebo přes nabídku *Edit/Copy*).
- 5) Přejděte do nově vytvářeného již autorizovaného testovacího souboru.
- 6) Ve strukturogramu nově vytvářeného souboru **poklepáním myši na žlutou slučovací ikonu TEST:**

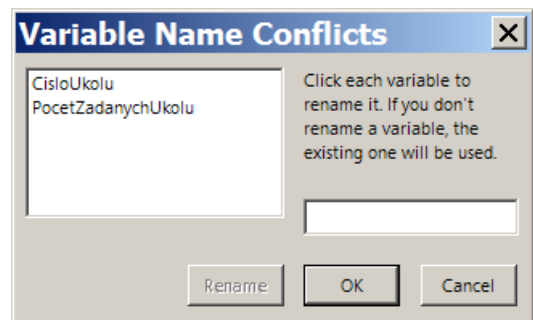


tuto ikonu otevřete:

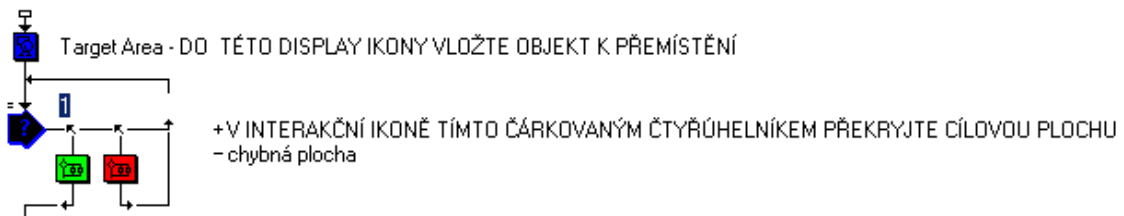


- 7) Na časovou osu ikony **TEST** vložte ze schránky **strukturogram zkopírované úlohy TARGET AREA** buď pomocí klávesové kombinace *Ctrl+V* (nebo přes nabídku *Edit/Paste*).

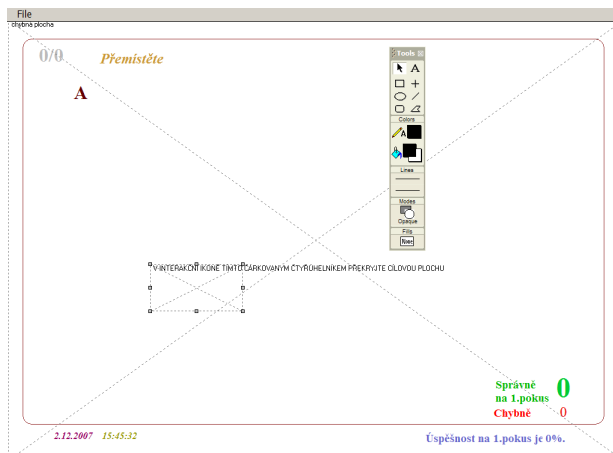
Nejprve Vám bude prezentováno dialogové okno indexování proměnných - klikněte v něm na tlačítko *OK*:



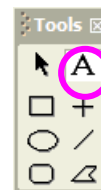
- 8) **Interakční ikonu** vložené předlohy úlohy *TARGET AREA* přejmenujte **pořadovým číslem úlohy** (např. 1.):



9) Poklepáním myší **otevřete interakční ikonu:**

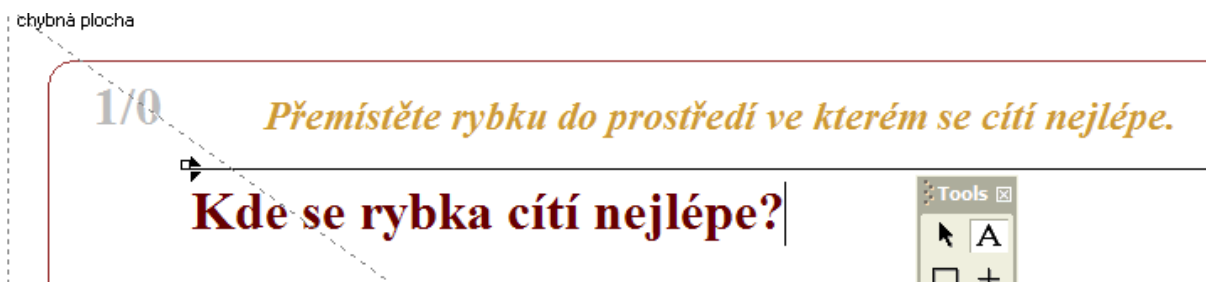


10) V panelu nástrojů (Tools) klikněte **na nástroj Text** (reprezentovaný písmenem A). Šipkový kurzor se změní na kurzor textový (písmeno velké I):



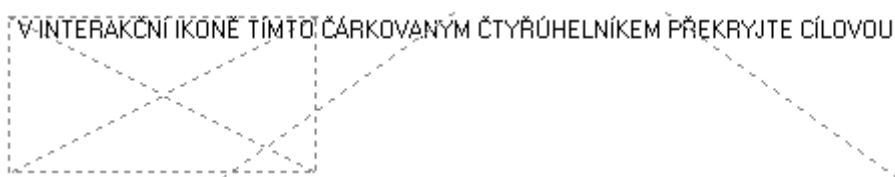
11) V horní části, vymezené pro zadání úlohy, **klikněte textovým kurzorem zprava vedle velkého písmene A a odmažte ho klávesou Backspace** – tím převezmete do psacího kurzoru formátovací charakteristiky (font, barvu a velikost písma atd.).

Pokračujte zápisem zadání učební úlohy a případným dalším textem, např.:



12) **Přes nabídku Insert/Image vložte obrázek**, který bude reprezentovat cílovou oblast správné odpovědi

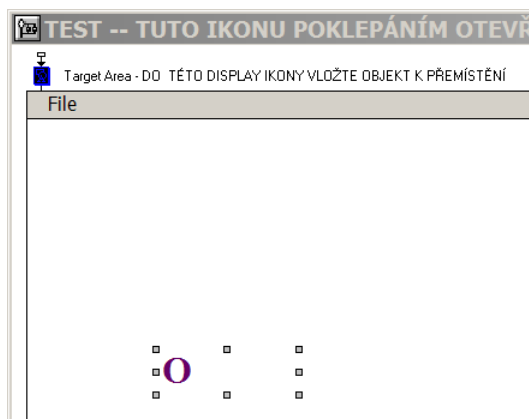
13) Čárkovaným **čtyřúhelníkem překryjte cílovou plochu správné odpovědi.**



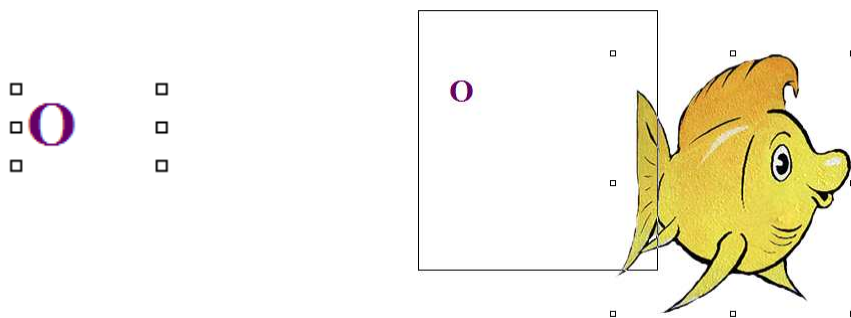
14) Obdobným způsobem (přes nabídku Insert) vložte na plochu prezentace obrázky nesprávných odpovědí – vložte je mimo čárkovaný čtyřúhelník!

15) Uzavřete interakční ikonu stiskem kláves CTRL+J.

16) Otevřete zobrazovací ikonu:



17) Do ní vložte objekt k přemístění (např. obrázek ryby) přes nabídku Insert, a přetáhněte ho na aktivní objekt (znázorněný písmenem O):



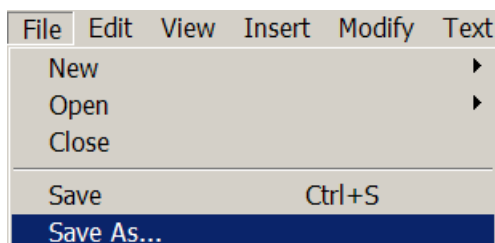
18) Zavřete zobrazovací ikonu stiskem kláves Ctrl+J.

19) Umístěte před interakční ikonu bílý praporek



20) Provéřte její bezchybný průběh. Případné nedostatky opravte.

21) **Bílý praporek odstráňte** kliknutím do lišty ikon. **Aplikaci uložte** přes nabídku File/Save As/...



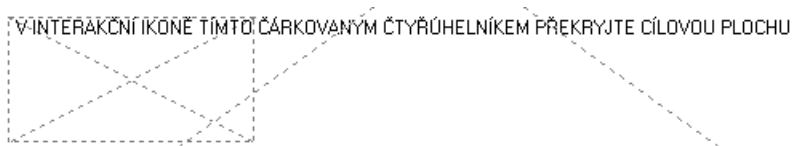
22) Pokud chcete do aplikace vložit **další úlohy typu TARGET AREA**, opakujte postup popsany v bodech 2) až 21).

23) Pokud chcete do aplikace vložit **jiný typ úlohy** (např. Button, Text Entry aj.), postupujte dle pracovního archu příslušného typu úlohy.

24) Pokud jste ukončili konstrukci výsledné procvičovací aplikace ke své spokojenosti a přejete si **vytvořit plnohodnotnou windowsovou žákovskou aplikaci** (.exe soubor), postupujte podle *Pracovního archu č. 8 Vytváření spustitelných aplikací*.

5.2.5.1 Komentář k pracovnímu archu č. 5: Přidávání a modifikace testovacích úloh typu Target Area (cílová oblast)

„Čárkovaným čtyřúhelníkem překryjte cílovou plochu správné odpovědi.“



Obrázek 85: Target Area

Podobně jako u typů Hot Spot a Hot Object i v tomto případě jsem uživateli zobrazil interaktivní objekt. Tím spíš, že v úloze typu Target Area jsou aktivní oblasti dvě.

5.2.6 Typ Key Press (stisk klávesy)

Typ Key Press (stisk klávesy) je založen na snímání jednotlivých znaků z klávesnice a jejich vyhodnocování. To je užitečné ve chvílích, když nám stačí pro identifikaci správné a špatné odpovědi jedno písmeno. Svým způsobem jde o interakci podobnou s úlohou typu Button, pouze s tím rozdílem, že odpovědi zadáme klávesnicí namísto myší.

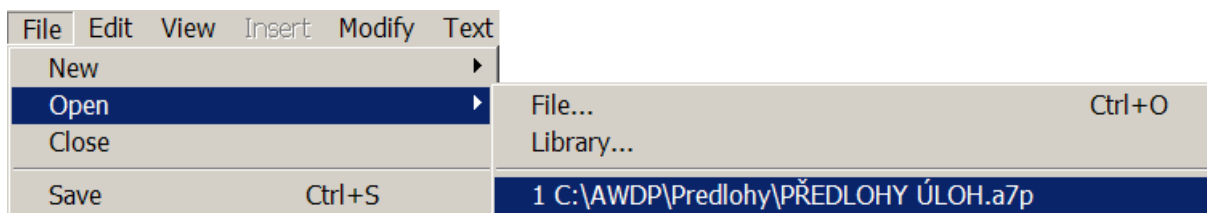
Uživatelé budou aplikaci zpracovávat podle pracovního archu číslo 6 následujícího znění:

Pracovní arch č. 6

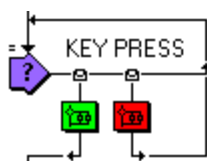
Přidávání a modifikace testovacích úloh typu Key Press (stisk klávesy)

Do autorizovaného a přejmenovaného nového souboru (viz pracovní arch č.1) vložíte výběrovou úlohu Key Press (stisk klávesy) následujícím postupem:

- 1) **Otevřete soubor "PŘEDLOHY ÚLOH.a7p"** přes nabídku File/Open/

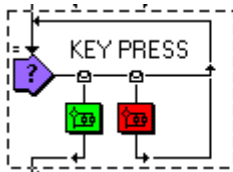


- 2) V získaném strukturogramu všech předloh **vyhledejte strukturogram úlohy typu KEY PRESS:**



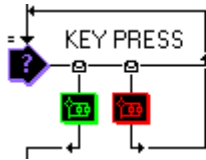
+ a)A-- TENTO ŘÁDEK NAHRAĎTE ALTERNATIVOU SPRÁVNÉ KLÁVESY
- b)B)C)D)D-- TENTO ŘÁDEK NAHRAĎTE ALTERNATIVAMI NESPRÁVNÝCH KLÁVES

- 3) **Strukturogram potřebné předlohy úlohy KEY PRESS označte pomocí myši** (držte stisknuté levé tlačítko myši a objeďte strukturogram).



+ a)A-- TENTO ŘÁDEK NAHRAĎTE ALTERNATIVOU SPRÁVNÉ KLÁVESY
 - b)B|c|d|D-- TENTO ŘÁDEK NAHRAĎTE ALTERNATIVAMI NESPRÁVNÝCH KLÁVES

Ikony strukturogramu předlohy po označení ztmavnou:

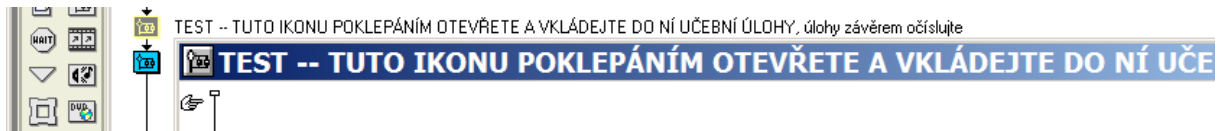


+ a)A-- TENTO ŘÁDEK NAHRAĎTE ALTERNATIVOU SPRÁVNÉ KLÁVESY
 - b)B|c|d|D-- TENTO ŘÁDEK NAHRAĎTE ALTERNATIVAMI NESPRÁVNÝCH KLÁVES

- 4) **Zkopírujte strukturogram** předlohy úlohy typu *KEY PRESS* do schránky pomocí klávesové kombinace *Ctrl+C* (nebo přes nabídku *Edit/Copy*).
- 5) **Přejděte do nově vytvářeného již autorizovaného testovacího souboru.**
- 6) Ve strukturogramu nově vytvářeného souboru **poklepáním myši na žlutou slučovací ikonu TEST:**

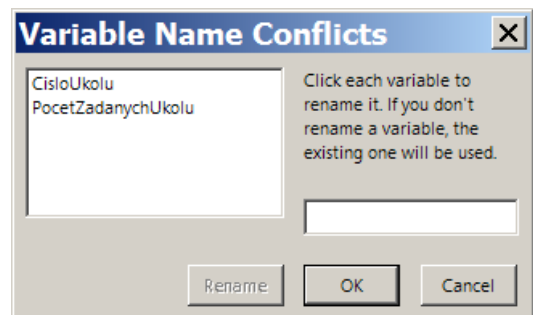


tuto ikonu otevřete:

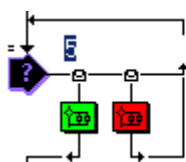


- 7) **Na časovou osu ikony TEST vložte ze schránky strukturogram** zkopírované úlohy *KEY PRESS* buď pomocí klávesové kombinace *Ctrl+V* (nebo přes nabídku *Edit/Paste*).

Nejprve Vám bude prezentováno dialogové okno indexování proměnných - **klikněte v něm na tlačítko OK:**



- 8) **Interakční ikonu** vložené předlohy úlohy *KEY PRESS* **přejmenujte** pořadovým číslem úlohy (např 1):

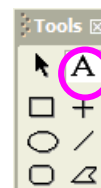


+ a|A-- TENTO ŘÁDEK NAHRAĎTE ALTERNATIVOU SPRÁVNÉ KLÁVESY
 - b|B|c|d|D-- TENTO ŘÁDEK NAHRAĎTE ALTERNATIVAMI NESPRÁVNÝCH KLÁVES

- 9) Poklepáním myší **otevřete interakční ikonu**:



- 10) V panelu nástrojů (Tools) **klikněte na nástroj Text** (reprezentovaný písmenem A). Šipkový kurzor se změní na kurzor textový (písmeno velké I):



- 11) V horní části, vymezené pro zadání úlohy, **klikněte textovým kurzorem zprava vedle velkého písmene A a odmažte ho klávesou Backspace** – tím převezmete do psacího kurzoru

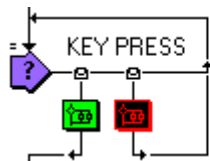
formátovací charakteristiky (font, barvu a velikost písma atd.).

Pokračujte zápisem zadání učební úlohy a případným dalším textem, např.:



- 12) **Uzavřete interakční ikonu** stiskem kláves CTRL+J.

- 13) Ve strukturogramu **přepište možnosti pro stisknutí kláves správných a chybných** odpovědí.



+||- TENTO ŘÁDEK NAHRAĎTE ALTERNATIVOU SPRÁVNÉ KLÁVESY
-yY- TENTO ŘÁDEK NAHRAĎTE ALTERNATIVAMI NESPRÁVNÝCH KLÁVES

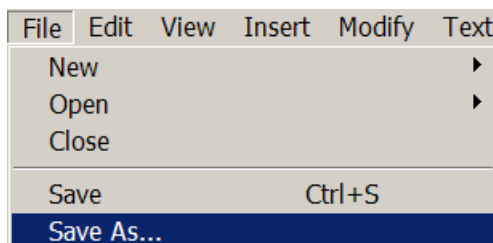
Správné klávesy jsou označeny znaménkem +, chybné jsou označené znaménkem -. Mezi jednotlivé varianty kláves se vkládá logický operátor OR - nebo (|). Ten vložíte pomocí jeho ASCII hodnoty tak, že přidržíte stisknutý levý Alt a napíšete číslo 0124, tedy Alt+0124.

- 14) Umístěte před interakční ikonu **bílý praporek**



- 15) **Proveďte její bezchybný průběh.** Případné nedostatky opravte.

- 16) **Bílý praporek odstráňte** kliknutím do lišty ikon. **Aplikaci uložte** přes nabídku File/Save As/...



- 17) Pokud chcete do aplikace vložit **další úlohu typu KEY PRESS**, opakujte postup popsany v bodech 2) až 16)
- 18) Pokud chcete do aplikace vložit **jiný typ úlohy** (např. Button, Hotspot) postupujte dle pracovního archu příslušného typu úlohy.
- 19) Pokud jste ukončili konstrukci výsledné procvičovací aplikace ke své spokojenosti a **přejete si vytvořit plnohodnotnou windowsovou žákovskou aplikaci** (.exe soubor), postupujte podle *Pracovního archu č. 8 Vytváření spustitelných aplikací*.

5.2.6.1 Komentář k pracovnímu archu č. 6: Přidávání a modifikace testovacích úloh typu Key Press (stisk klávesy)

„Mezi jednotlivé varianty kláves se vkládá logický operátor OR - nebo (/). Ten vložíte pomocí jeho ASCII hodnoty tak, že přidržíte stisknutý levý Alt a napíšete číslo 0124, tedy Alt+0124.“

S pochopením konstrukce této úlohy by tvůrci aplikace neměli mít problémy až na zadávání logického operátoru OR. Proto jsem vypsals jeho hodnotu v ASCII (*American Standard Code for Information Interchange*) tabulce a také kombinaci kláves pro jeho napsání.

5.2.7 Typ Text Entry (textová odpověď)

Typ Text Entry (v překladu textová položka) je velmi podobný typu Key Press. Liší se v tom, že na rozdíl od typu Key Press, který přijímá pouze jeden znak, úloha Text Entry přijímá jako odpověď posloupnost více znaků (slova, věty).

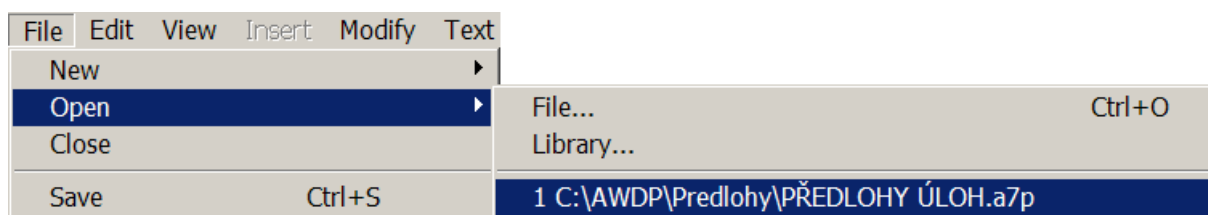
Uživatelé budou aplikaci zpracovávat podle pracovního archu číslo 7 následujícího znění:

Pracovní arch č. 7

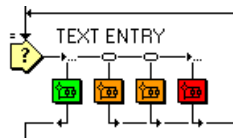
Přidávání a modifikace testovacích úloh typu Text Entry (textová odpověď)

Do autorizovaného a přejmenovaného nového souboru (viz pracovní arch č.1) vložíte výběrovou úlohu typu Text Entry (textová odpověď) následujícím postupem:

- 1) Otevřete soubor "**PŘEDLOHY ÚLOH.a7p**" přes nabídku File/Open/

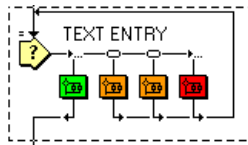


- 2) V získaném strukturogramu všech předloh **vyhledejte strukturogram úlohy typu TEXT ENTRY:**



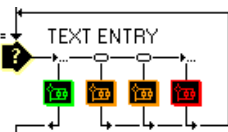
- +1--CELÝ TENTO ŘÁDEK NAHRAĎTE ZNĚNÍM SPRÁVNÉHO ŘEŠENÍ ÚLOHY
- částečná nápověda--DO DISPLAY IKONY UVNITŘ TĚTO MAP IKONY VEPIŠTE ČÁSTEČNOU NÁPOVĚDU
- úplná nápověda --DO DISPLAY IKONY UVNITŘ TĚTO MAP IKONY VEPIŠTE ÚPLNĚ SPRÁVNÉHO ŘEŠENÍ
- *

3) **Strukturogram potřebné předlohy úlohy TEXT ENTRY označte pomocí myši (držte stisknuté levé tlačítko myši a objeďte strukturogram).**



- +1--CELÝ TENTO ŘÁDEK NAHRAĎTE ZNĚNÍM SPRÁVNÉHO ŘEŠENÍ ÚLOHY
- částečná nápověda--DO DISPLAY IKONY UVNITŘ TĚTO MAP IKONY VEPIŠTE ČÁSTEČNOU NÁPOVĚDU
- úplná nápověda --DO DISPLAY IKONY UVNITŘ TĚTO MAP IKONY VEPIŠTE ÚPLNĚ SPRÁVNÉHO ŘEŠENÍ
- *

Ikony strukturogramu předlohy po označení ztmavnou:



- +1--CELÝ TENTO ŘÁDEK NAHRAĎTE ZNĚNÍM SPRÁVNÉHO ŘEŠENÍ ÚLOHY
- částečná nápověda--DO DISPLAY IKONY UVNITŘ TĚTO MAP IKONY VEPIŠTE ČÁSTEČNOU NÁPOVĚDU
- úplná nápověda --DO DISPLAY IKONY UVNITŘ TĚTO MAP IKONY VEPIŠTE ÚPLNĚ SPRÁVNÉHO ŘEŠENÍ
- *

4) **Zkopírujte strukturogram předlohy úlohy typu TEXT ENTRY do schránky pomocí klávesové kombinace Ctrl+C (nebo přes nabídku Edit/Copy).**

5) **Přejděte do nově vytvářeného již autorizovaného testovacího souboru.**

6) Ve strukturogramu nově vytvářeného souboru poklepáním myši **na žlutou slučovací ikonu TEST:**



TEST -- TUTO IKONU POKLEPÁNÍM OTEVŘETE A VKLÁDEJTE DO NÍ UČEBNÍ ÚLOHY, úlohy závěrem očíslujte

tuto ikonu otevřete:

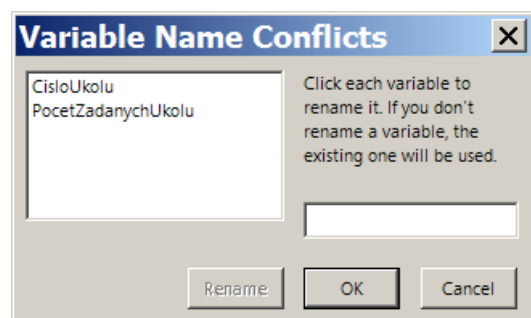


TEST -- TUTO IKONU POKLEPÁNÍM OTEVŘETE A VKLÁDEJTE DO NÍ UČEBNÍ ÚLOHY, úlohy závěrem očíslujte

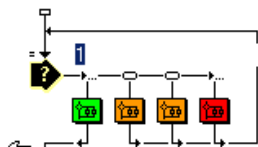
TEST -- TUTO IKONU POKLEPÁNÍM OTEVŘETE A VKLÁDEJTE DO NÍ UČE

7) **Na časovou osu ikony TEST vložte ze schránky strukturogram zkopírované úlohy KEY PRESS buď pomocí klávesové kombinace Ctrl+V (nebo přes nabídku Edit/Paste).**

Nejprve Vám bude prezentováno dialogové okno indexování proměnných
- klikněte v něm na tlačítko **OK**:

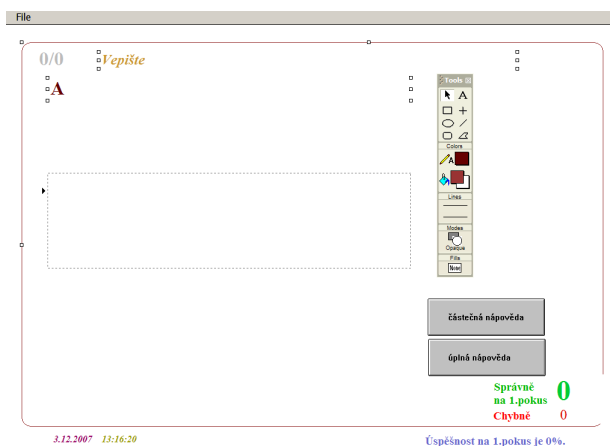


- 8) **Interakční ikonu** vložené předlohy úlohy *TEXT ENTRY* přejmenujte **pořadovým číslem úlohy** (např 1):

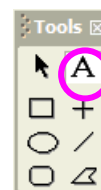


+1--CELÝ TENTO ŘÁDEK NAHRAĎTE ZNĚNÍM SPRÁVNÉHO ŘEŠENÍ ÚLOHY
 - částečná nápověda--DO DISPLAY IKONY UVNITŘ TĚTO MAP IKONY VEPIŠTE ČÁSTEČNOU NÁPOVĚDU
 - úplná nápověda --DO DISPLAY IKONY UVNITŘ TĚTO MAP IKONY VEPIŠTE ÚPLNÉ SPRÁVNÉHO ŘEŠENÍ
 -*

- 9) Poklepnutím myší **otevřete interakční ikonu**:



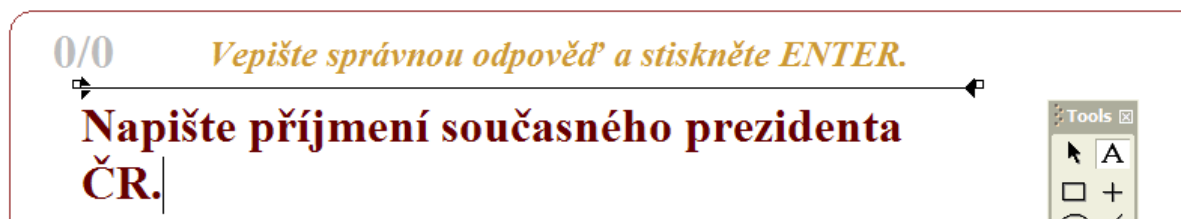
- 10) V panelu nástrojů (Tools) **klikněte na nástroj Text** (reprezentovaný písmenem A). Šipkový kurzor se změnil na kurzor textový (písmeno velké I):



- 11) V horní části, vymezené pro zadání úlohy, **klikněte**

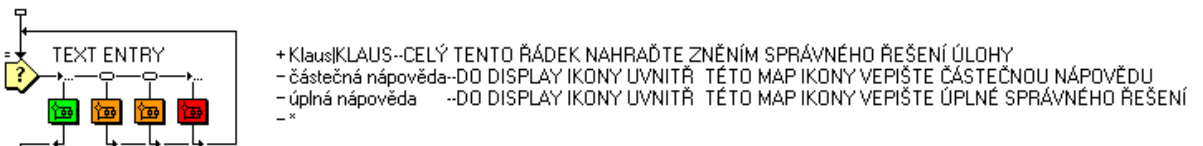
textovým kurzorem zprava vedle velkého písmene A a odmažte ho klávesou Backspace – tím převeďte do psacího kurzoru formátovací charakteristiky (font, barvu a velikost písma atd.).

Pokračujte zápisem zadání učební úlohy a případným dalším textem, např.:



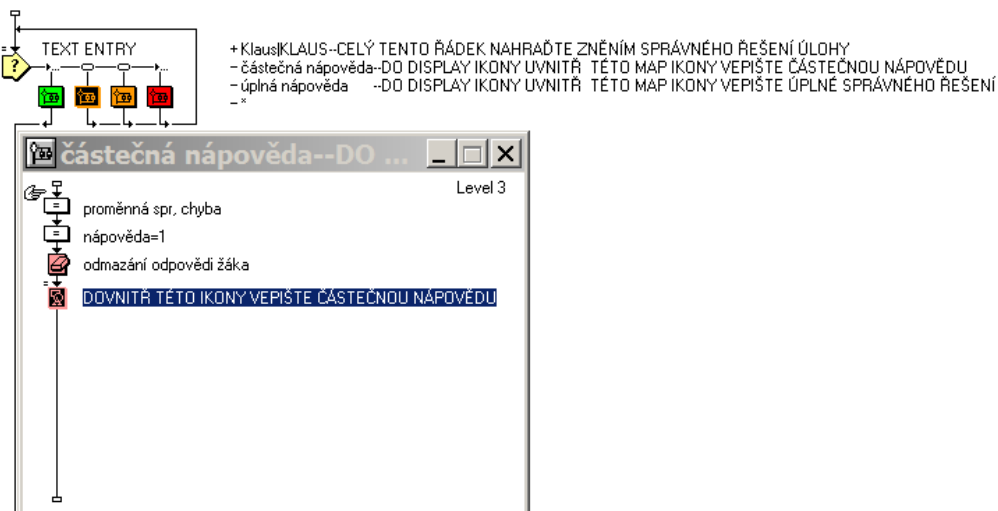
12) Uzavřete interakční ikonu stiskem kláves CTRL+J.

13) Ve strukturogramu přepište možnost pro správnou odpověď (např. Klaus).

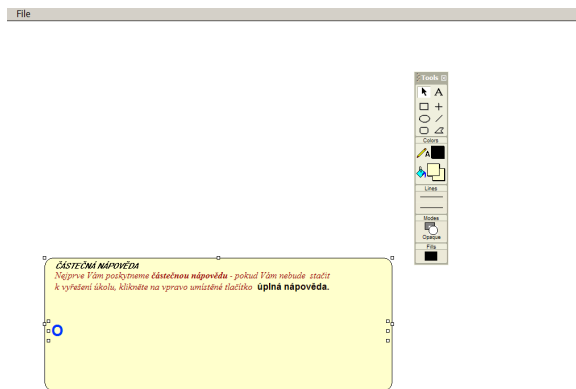


Správná odpověď je označena znaménkem +, chybné odpovědi jsou označené znaménkem -. Mezi jednotlivé varianty kláves se vkládá logický operátor OR - nebo (|). Ten vložíte pomocí jeho ASCII hodnoty tak, že přidržíte stisknutý levý Alt a napíšete číslo 0124, tedy Alt+0124.

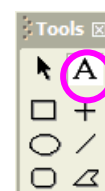
14) Otevřete slučovací ikonu částečné nápovědy:



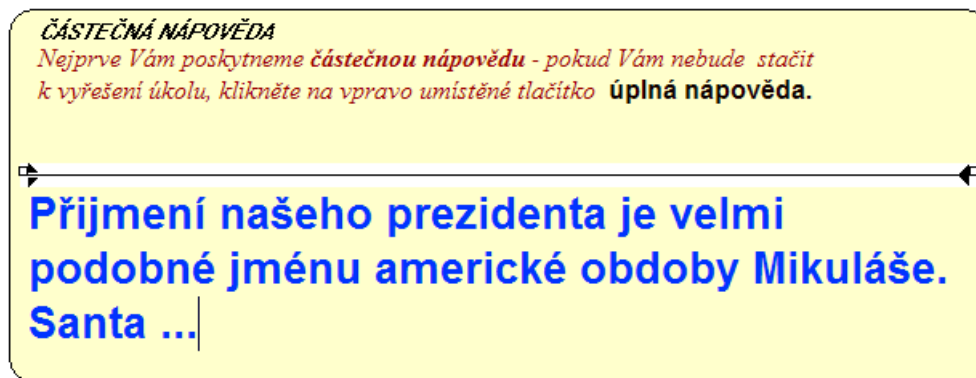
a v ní otevřete zobrazovací ikonu:



- 15) V panelu nástrojů (Tools) **klikněte na nástroj Text** (reprezentovaný písmenem A). Šipkový kurzor se změní na kurzor textový (písmeno velké I):

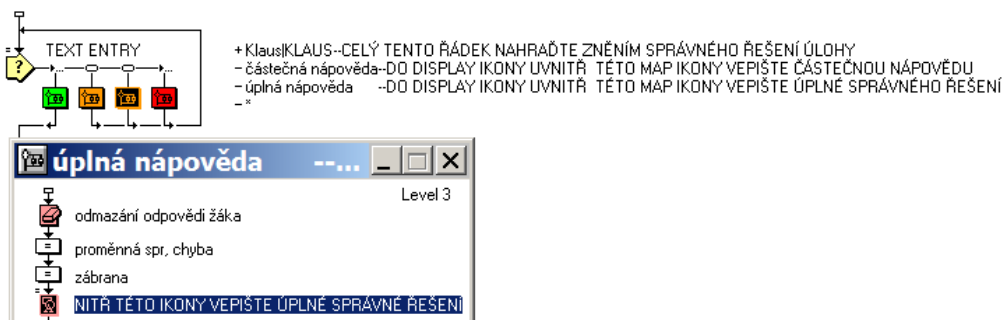


- 16) V části vymezené pro zadání částečné nápovědy **klikněte textovým kurzorem zprava vedle velkého písmene O a odmažte ho klávesou *Backspace*** – tím převezmete do psacího kurzoru formátovací charakteristiky (font, barvu a velikost písma atd.). Pokračujte zápisem zadání částečné nápovědy, např.:

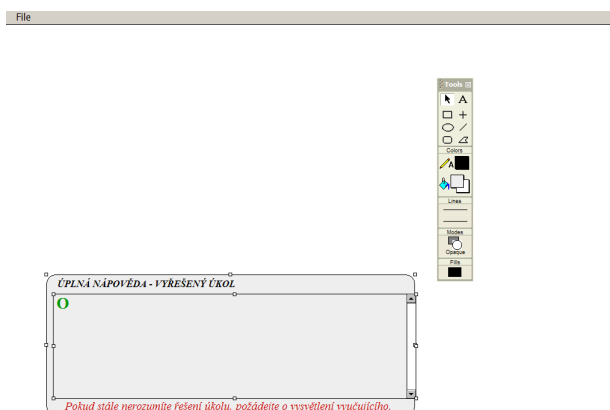


- 17) **Uzavřete obsah zobrazovací ikony** stiskem kláves CTRL+J.

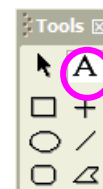
- 18) **Otevřete slučovací ikonu úplné nápovědy:**



a v ní otevřete zobrazovací ikonu:



19) V panelu nástrojů (Tools) **klikněte na nástroj Text** (reprezentovaný písmenem A). Šipkový kurzor se změní na kurzor textový (písmeno velké I):



20) V části vymezené pro zadání úplné nápovědy **klikněte textovým kurzorem zprava vedle velkého písmene O a odmažte ho klávesou Backspace** – tím převezmete do psacího kurzoru formátovací charakteristiky (font, barvu a velikost písma atd.). Pokračujte zápisem zadání úplné nápovědy, např.:



21) **Uzavřete obsah zobrazovací ikony** stiskem kláves CTRL+J.

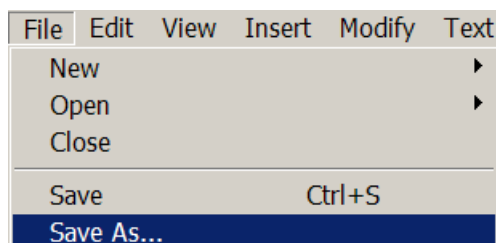
22) Umístěte před interakční ikonu **bílý praporek**



a spustíte aplikaci

23) **Proveřte její bezchybný průběh.** Případné nedostatky opravte.

24) **Bílý praporek odstraňte** kliknutím do lišty ikon. **Aplikaci uložte** přes nabídku File/Save As/...



25) Pokud chcete do aplikace vložit **další úlohu typu TEXT ENTRY**, opakujte postup popsany v bodech 2) až 24)

26) Pokud chcete do aplikace vložit **jiný typ úlohy** (např. Button, Hotspot) postupujte dle pracovního archu příslušného typu úlohy.

27) Pokud jste ukončili konstrukci výsledné procvičovací aplikace ke své spokojenosti a **přejete si vytvořit plnohodnotnou windowsovou žákovskou aplikaci** (.exe soubor), postupujte podle *Pracovního archu č. 8 Vytváření spustitelných aplikací*.

5.2.7.1 Komentář k pracovnímu archu č. 7: Přidávání a modifikace testovacích úloh typu Text Entry (textová odpověď)

Pochopení konstrukce otázky typu Text Entry by nemělo být pro autora aplikace příliš složité, zvláště jestli už předtím konstruoval typ Key Press. Vzhledem k tomu, že pořadí procházení výukových videí a podpůrných pracovních archů není pevně dáno, jsem raději znovu zopakoval způsob zápisu a použití logického operátoru OR.

5.2.8 Vytváření spustitelných aplikací.

Možnost vytváření samostatných spustitelných aplikací ze souborů MAW (.a7p), je velice důležitá kvůli možnosti používat didaktické aplikace vytvořené v MAW bez nutnosti mít nainstalovaný Authorware na počítači, kde ji chceme spustit. Ačkoliv je publikování (vytváření spustitelných souborů) v MAW poměrně jednoduché, mívají

s ním občas problémy i zkušenější uživatelé. Tyto problémy přičítám především rozhraní MAW a nepříliš jednoduchým slovním obrátům použitým v nastavení publikační části. V pracovním archu zabývajícím se vytvářením spustitelných aplikací jsem tedy přesně popsal, co která volba v publikační části Macromedia Authorware dělá.

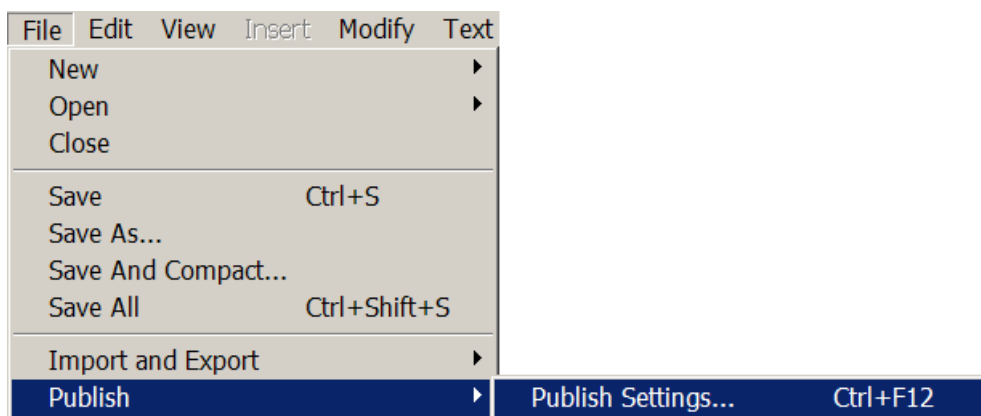
Uživatelé budou aplikaci zpracovávat podle pracovního archu číslo 8 následujícího znění:

Pracovní arch č. 8

Vytváření spustitelných aplikací

Pokud jste s pokojeni se vzhledem a funkcí Vámi vytvořené aplikace, můžete ji převést z formátu Authorwaru (.a7p) na nezávislou spustitelnou aplikaci (.exe soubor). To můžete provést následujícím postupem:

- 1) **Otevřete soubor, ze kterého chcete vytvořit spustitelnou aplikaci** (.exe soubor) přes nabídku File/Open
- 2) **Otevřete nabídku pro publikování** aplikace přes nabídku File/Publish/Publish Settings...



- 3) V nově otevřeném dialogovém okně nastavte parametry převodu souboru.

Zaškrtněte volby:

Package As (sbalit jako) – zde můžete nastavit jméno a umístění finálního souboru

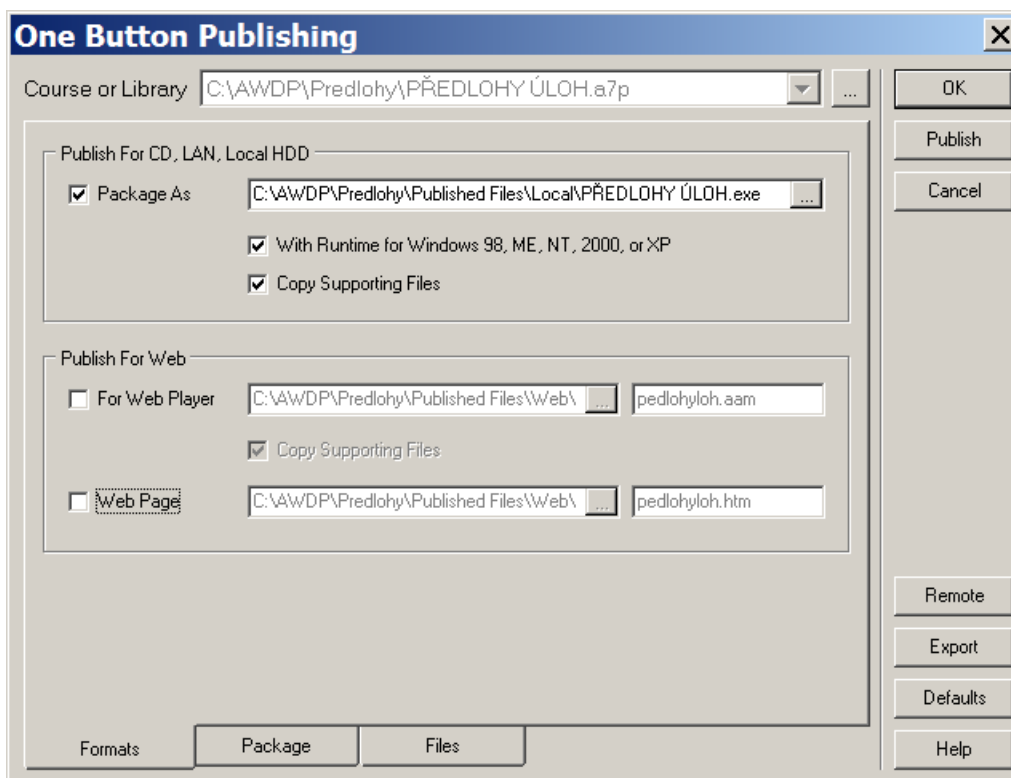
With Runtime for Windows 98, ME, NT, 2000 or XP – tato volba převede soubor na spustitelný .exe soubor pro Windows

Copy Supporting Files – touto volbou určíte, jestli se mají k .exe souboru přikopírovat i všechny knihovny potřebné k správnému chodu programu

- 4) Pokud nepočítáte s rozesíláním aplikací prostřednictvím serveru, odškrtněte volby:

For Web Player

Web Page



- 5) **Klikněte na tlačítko Publish** (publikovat).
- 6) V místě, které jste určili v položce **Package As** vznikne kromě spustitelného .exe souboru také podsložka **XTRAS**, která se musí uživatelům předat spolu se spustitelným .exe souborem, jinak neuvidí obrázky a neuslyší zvuk
- 7) **Aplikaci vyzkoušejte spuštěním nově vytvořeného .exe souboru** umístěného v adresáři, který jste specifikovali u volby *Package As*.

6. Závěr

V diplomové práci jsem v souladu s požadavky vedoucího diplomové práce splnil všechny vytyčené cíle. To znamená:

A. V teoretické části diplomové práce

- 1) Provedl jsem obsahová analýzu a komparaci odborných textů se zkoumanou problematikou.
- 2) Objasnil jsem problematiku dat (proměnných a funkcí) v autorském systému Macromedia Authorware na základě mého odborného překladu z anglických originálů.
- 3) Provedl jsem didaktickou transformaci a minimalizaci poznatků o proměnných a funkcích.

B. V praktické části diplomové práce

- 1) Vytvořil jsem rozsáhlý soubor výukových videí o tvorbě didaktických aplikací v autorském systému Macromedia Authorware.
- 2) Vytvořil jsem rozsáhlý soubor pracovních archů o tvorbě didaktických aplikací v autorském systému Macromedia Authorware.

V úvodní části jsem zdůvodnil aktuálnost problematiky, uvedl jsem důvody volby tématu a vymezil jsem cíle a metody předložené diplomové práce. V kapitole 2 vymezuji a upřesňuji základní pojmy autorského systému Macromedia Authorware. V kapitolách 3 a 4 jsem v souladu s požadavky vedoucího práce rozpracoval problematiku dat (proměnných a funkcí) v autorském systému Macromedia Authorware.

Praktická část diplomové práce obsahuje soubor instruktážních videí pro učitele.

Soubor pracovních archů k instruktážním videím je obsahem přílohy.

Práce v autorském systému Macromedia Authorware mě velmi zaujala. Diplomové práci jsem proto věnoval hodně času a úsilí. Jsem přesvědčen, že to nebylo zbytečné. Získal jsem důležité poznatky o možnostech využití ICT (informační a komunikační

technologie) pro zvyšování efektivity procesu učení, které budu využívat ve své pedagogické praxi.

Chtěl bych se touto problematikou i nadále zabývat a pokračovat tak v započaté práci.

7. Použité prameny

1. BRDIČKA, B. *Authorware Professional*, Bulletin inform. technologií ve škole, 1993.
2. *Computers and the Very Young* [online][cit. 2007-12-10]. Dostupné z <<http://www.acei.org/inf.vol.13.4.htm>>
3. ERIC Counseling and Student Services Clearinghouse [online][cit. 2007-12-9]. Dostupné z <<http://www.kidsource.com/education/computers.children.html>>
4. HOLOUŠOVÁ, D. KROBOTOVÁ, M. *Diplomové a závěrečné práce*. Olomouc : Pedagogická fakulta UP, 2002. ISBN 80-244-0458-3.
5. KELLOG, Orson; BHATNAGAR, Veera. *Macromedia Authorware 6: Training from the Source*. 1st edition. Berkley : Macromedia Press, 2002. 555s. ISBN 0-201-77426-7.
6. Kol: *Authorware Professional (v. 2)*. San Francisco : Macromedia, 1993.
7. Kol: *Authorware Professional (v. 4)*. San Francisco : Macromedia, 1997.
8. Kol: *Authorware Professional Help Pages (v. 5.0)*. San Francisco : Macromedia, 1999.
9. Kol: *Authorware Professional Help Pages (v. 7.0)*. San Francisco : Macromedia, 2007.
10. Kol: *Dokumentace systému Authorware 3.0*, Macromedia Inc. 1995
11. *Macromedia Authothorware 7: Uživatelská příručka* [online]. Olomouc: Digital Media, 2005. Dostupné z : <http://www.fmm.cz/dmdownload/manualy/AW7_CZmanual.zip>.
12. MUSIL, T. *Didaktické využití Autorského systému Macromedia Authorware pro konstrukci didaktických testů z anglického jazyka*. [Diplomová práce] Liberec, Pedagogická fakulta Technické university Liberec, 2007. 85 s.

13. NIKL, J. Autorský systém – nástroj tvorby prostředků racionálního řízení učebních činností žáků. In *Nové trendy vzdělávání učitelů přírodovědných oborů*. Praha : Karolinum, 1998, s.51-55.
14. PRŮCHA, WALTEROVÁ, MAREŠ *Pedagogický slovník*. Praha : Portál, 1995. ISBN 80-7178-029-4.
15. SKOUMAL, J. *Konstrukce didaktických aplikací v Macromedia Authorware*. [Diplomová práce] Liberec, Pedagogická fakulta Technické university Liberec, 2007. 149 s.
16. *Teacher vs. Computer. Where Educators Stand in the Technology Revolution* [online][cit 2007-12-10]. Dostupné z <<http://thejournal.com/articles/16065>>
17. VÍTOVSKÝ, A. *Moderní slovník softwaru* 1. vyd. Praha : AV Software, 2006. 588 s. ISBN 80-901428-8-5.
18. Wikipedia: Authoring systems [online] [cit.2007-12-17]. Dostupné z <http://en.wikipedia.org/wiki/Authoring_systems>.

8. Přílohy

8.1 CD – ROM

8.1.1 Textová část diplomové práce

Textová část diplomové práce je uložena v adresáři TEXT, v souboru DP.doc ve formátu pro Microsoft Word 2003 a DP.pdf ve formátu pro Adobe Acrobat či Adobe Acrobat Reader.

8.1.2 Soubor výukových videí pro pochopení práce v autorském systému MAW

Ve složce VIDEA jsou přiložena výuková videa o tvorbě didaktických aplikací v MAW. Soubor zahrnuje ukázky konstrukcí jednotlivých typů úloh a také výukové video o publikaci aplikací v Macromedia Authorware. Jednotlivá videa jsou ve formátu spustitelných (.exe souborů). Pro snadnou orientaci je přiložen jednoduchý rozcestník spustitelný pod souborem ROZCESTNIK.exe.

8.1.3 Soubory předloh jednotlivých typů úloh pro tvorbu didaktických aplikací v MAW

Ve složce PREDLOHY jsou soubory NOVÝ SOUBOR 1024x768.a7p a PŘEDLOHY ÚLOH.a7p využívané při tvorbě didaktických aplikací podle instruktážních videí.

8.1.4 Soubor pracovních archů k instruktážním videím

Ve složce PRACOVNIARCHY je soubor pracovních archů k instruktážním videím.

8.1.5 Soubor programů pro pochopení práce s vybranými systémovými funkcemi MAW

Ve složce FUNKCE jsou přiloženy soubory ve pro pochopení práce s vybranými funkcemi MAW. Jednotlivé soubory jsou ve formátu pro MAW verze 7 (.a7p).