

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačové grafiky a interakce

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Jan Svoboda**

Studijní program: Otevřená informatika
Obor: Softwarové inženýrství

Název tématu: **Úlohy pro kognitivní cvičení**

Pokyny pro vypracování:

Soubor úloh pro kognitivní cvičení:

1. Prostudujte úlohy zaměřené na kognitivních schopností člověka, zejména úlohy k procvičování každodenních aktivit a činností: úklid, nakupování, vaření, domácí zařízení atd.
2. Tyto úlohy rozdělte do skupin podle jejich: zaměření, činnosti, náročnosti atd. Případně cílové skupiny uživatelů.
3. Vyberte buď jednu skupinu úloh, nebo i jednotlivé úlohy z více skupin vhodné pro osoby postižené zejména ztrátou paměti, neuvědomění si posloupnosti již vykonaných akcí.
4. Implementujte tedy aspoň 5 úloh lišících se typem. U každé z nich dbejte na jejich možnou modifikaci (doplnění, rozšíření, přizpůsobení) a hlavně snadné ovládání (pouze jednoduché posloupnosti dotyků). Vytvořené úlohy musí vykazovat interaktivní chování, aby byly pro uživatele zajímavé (zvukové a grafické podněty).
5. Na počtu aspoň pěti uživatelů ověřte (pouze) srozumitelnost a snadnost ovládání úloh (nikoli kognitivní přínos, který je dán pouze jejich dlouhodobým využíváním).

Seznam odborné literatury:

- 1) Dostupné WWW stránky projektů vytvářející úlohy pro cvičení této tematiky
- 2) WWW stránky organizací využívající různé typy úloh s touto tematikou (CEREBRUM, Jedličkův ústav a školy, ...)
- 3) Nagel, Evjen, Glynn, Watson, Skinner, Professional C# 4 .NET 4, Wrox, 2010
- 4) Matthew MacDonald, Pro WPF in C# 2010, Apress, 2013
- 5) Matthew MacDonald, Pro WPF: Windows Presentation Foundation in .NET 3.0, APress, 2007

Vedoucí: Ing. Petr Novák, Ph.D.

Platnost zadání: do konce letního semestru 2015/2016

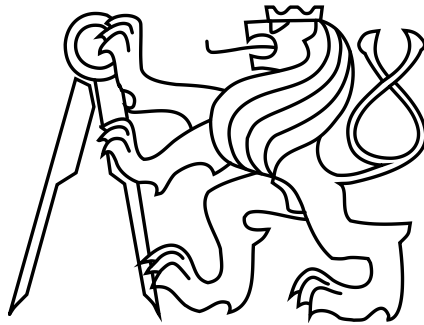


prof. Ing. Jiří Žára, CSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 25. 3. 2015

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačové grafiky a interakce



Diplomová práce

Úlohy pro kognitivní cvičení
Tasks for cognitive exercises

Bc. Jan Svoboda

Vedoucí práce: Ing. Petr Novák, Ph.D.

Studijní program: Otevřená informatika

Studijní obor: Softwarové inženýrství

Praha 2015

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne

.....

Podpis autora práce

Poděkování

Chtěl bych poděkovat vedoucímu diplomové práce Ing. Petru Novákovi, Ph.D. za veškerou odbornou pomoc a cenné rady při zpracování mé diplomové práce. Dále bych chtěl poděkovat svým rodičům a přítelkyni za psychickou podporu a vytvoření podmínek po celou dobu mého studia.

Anotace

Práce obsahuje stručný úvod k problematice kognitivního tréninku doplněný o porovnání existujících řešení zabývajících se tréninkem kognitivních funkcí. Hlavním cílem práce je vytvoření (naprogramování) jednoduše ovladatelné interaktivní sady úloh, určené právě pro kognitivní trénink. Vytvářené úlohy jsou určeny zejména pro zařízení s dotykovou obrazovkou, za použití OS Microsoft Windows, a jsou implementovány v programovacím jazyce C#, za použití prostředí .NET Framework. Řešení jednotlivých úloh se sestává z několika jednoduchých posloupností dotyků a je doprovázeno interaktivními signály ve formě zvukových a grafických podnětů. Vytvořené úlohy jsou rozděleny do 5 skupin, kde každá skupina obsahuje 2-3 typy úloh. Díky snadné modifikovatelnosti jednotlivých úloh se dá vytvořit velké až skoro neomezené množství zadání.

Klíčová slova

Kognitivní funkce, Interaktivní úlohy, Windows, C#, .NET

Annotation

Work contains short introduction to theory of cognitive training supplied with comparison of existing solutions interested in training of cognitive functions. Main goal of this work is creating (programming) simple controlled interactive set of tasks, intended for cognitive training. Created tasks are assigned especially for devices with touchscreen, by using OS Windows and are implemented in programming language C# and .NET Framework environment. Solving of each task consist of few simple sequences of touches and goes along with interactive signals in sound and graphic form. Created tasks are disjoined in 5 groups, where each group contains 2-3 types of task. Due to easy process of modification there can be created big and almost infinite count of assignments.

Keywords

Cognitive functions, Interactive tasks, Windows, C#, .NET

Obsah

Seznam obrázků	9
Seznam tabulek	10
Seznam použitých zkratk	11
Použité pojmy u úloh	11
Úvod	12
1 Trénink kognitivních funkcí.....	13
1.1 Kognitivní funkce.....	13
1.2 Dělení kognitivních funkcí.....	13
1.2.1 Paměť.....	13
1.2.2 Pozornost.....	14
1.2.3 Jazyk a řečové schopnosti	15
1.2.4 Zrakově-prostorové schopnosti	15
1.2.5 Myšlení	16
1.3 Existující řešení a jejich porovnání	16
1.3.1 Vzpomínkovi	16
1.3.2 Neurop	18
1.3.3 Brain jogging	19
1.4 Rozdělení a výběr úloh pro tuto práci	20
1.4.1 Posloupnosti akcí.....	20
1.4.2 Křížovky a osmisměrky.....	21
1.4.3 Spojovačky a obrázky.....	21
1.4.4 Paměť.....	21
1.4.5 Odpovědi a dvojice.....	21
2 Cíle práce.....	22
3 Návrh řešení.....	23
3.1 Použité technologie	23
3.1.1 C# a .NET	23
3.1.2 Prostředí pro vývoj	23
3.1.3 Stavový automat	25
3.2 Režimy činnosti úlohy.....	26
3.2.1 Režim konfigurace.....	26
3.2.2 Režim běhu	26
3.3 Úlohy vybrané pro vytvoření	27
3.3.1 Posloupnosti akcí.....	27
3.3.2 Křížovky a osmisměrky.....	28
3.3.3 Spojovačky a obrázky.....	28
3.3.4 Paměť.....	29
3.3.5 Odpovědi a dvojice.....	29

4 Implementace	30
4.1 Tvorba a ovládání úloh	30
4.1.1 Posloupnosti akcí	32
4.1.2 Křížovky a osmisměrky	33
4.1.3 Spojovačky a obrázky	34
4.1.4 Paměť	36
4.1.5 Odpovědi a dvojice	37
4.2 Ukládání dat úloh.....	38
4.2.1 Posloupnosti akcí	38
4.2.2 Křížovky a osmisměrky	40
4.2.3 Spojovačky a obrázky	42
4.2.4 Paměť	42
4.2.5 Odpovědi a dvojice	44
5 Prvotní testování	46
5.1 Posloupnosti akcí.....	46
5.2 Křížovky a osmisměrky.....	47
5.3 Spojovačky a obrázky.....	47
5.4 Paměť.....	48
5.5 Odpovědi a dvojice.....	48
6 Popis editace úloh.....	49
6.1 Posloupnosti akcí.....	50
6.2 Křížovky a osmisměrky.....	52
6.3 Spojovačky a obrázky.....	54
6.4 Paměť.....	56
6.5 Odpovědi a dvojice.....	56
7 Popis ovládání.....	59
7.1 Posloupnosti akcí.....	59
7.2 Křížovky a osmisměrky.....	61
7.3 Spojovačky a obrázky.....	64
7.4 Paměť.....	65
7.5 Odpovědi a dvojice.....	68
Závěr.....	71
Zdroje.....	72
Příloha A – Ukázka konfigurace úlohy posloupnosti akcí.....	74
Příloha B – Ukázka konfigurace úlohy křížovky a rébusy.....	75
Příloha C – Ukázka konfigurace úlohy spojovačky	76
Příloha D – Ukázka konfigurace úlohy paměť'	77
Příloha E – Ukázka konfigurace úlohy slova a texty	78
Příloha F – Hierarchie vytvářené sady úloh	79
Příloha G – Obsah příloženého CD	80

Seznam obrázků

Obrázek 1 – Vzpomínkovi	17
Obrázek 2 – Neurop	19
Obrázek 3 - Online Brain jogging.....	20
Obrázek 4 - Uspořádání skupin úloh v projektu MS Visual Studio	30
Obrázek 5 - Přidání nového zadání	49
Obrázek 6 - Uložení zadání	49
Obrázek 7 - Prázdná úloha.....	50
Obrázek 8 - Přidání oblasti a její úpravy	51
Obrázek 9 - Kompletní úloha ze skupiny Posloupnosti akcí	52
Obrázek 10 - Nastavení velikosti matic.....	52
Obrázek 11 - Kompletní úloha typu Křížovky	53
Obrázek 12 - Kompletní úloha typu Osmisměrky	53
Obrázek 13 - Kompletní úloha typu Vyhledávání v textu	54
Obrázek 14 - Přidání oblasti a její úpravy	54
Obrázek 15 - Kompletní úloha typu Posloupnosti.....	55
Obrázek 16 - Kompletní úloha typu Kreslení	55
Obrázek 17 - Nastavení velikosti matic.....	56
Obrázek 18 - Kompletní úloha ze skupiny Paměť	56
Obrázek 19 - Nastavení počtu odpovědí a vyplnění zadání.....	57
Obrázek 20 - Kompletní úloha typu Dvojice.....	57
Obrázek 21 - Kompletní úloha typu Jedna odpověď	58
Obrázek 22 - Počáteční stav úlohy ze skupiny Posloupnosti akcí.....	59
Obrázek 23 – Grafické reakce úloh ze skupiny Posloupnosti akcí.....	60
Obrázek 24 - Počáteční stav úlohy typu Křížovky	61
Obrázek 25 - Grafické reakce úlohy typu Křížovky	61
Obrázek 26 - Počáteční stav úlohy typu Osmisměrky	62
Obrázek 27 - Vybrání slova z písmen matice	62
Obrázek 28 - Grafické reakce úlohy typu Osmisměrky	62
Obrázek 29 - Zobrazení tajenky u úlohy typu Osmisměrky	63
Obrázek 30 - Grafické reakce úlohy typu Posloupnosti	64
Obrázek 31 - Grafické reakce úlohy typu Kreslení	65
Obrázek 32 - Počáteční stav úlohy typu S umístěním	66
Obrázek 33 - Po uplynutí časového limitu.....	66
Obrázek 34 - Počáteční stav úlohy typu Bez umístění	67
Obrázek 35 - Po uplynutí časového limitu.....	67
Obrázek 36 - Grafické reakce úloh ze skupiny Paměť	67
Obrázek 37 - Pomocný obrázek.....	68
Obrázek 38 - Počáteční stav úlohy typu Dvojice.....	69
Obrázek 39 - Grafické reakce úlohy typu Dvojice	69

Seznam tabulek

Tabulka 1 - Defaultní zvuky úlohy typu Nezáleží na pořadí	60
Tabulka 2 - Defaultní zvuky úlohy typu Částečně záleží na pořadí.....	60
Tabulka 3 - Defaultní zvuky úlohy typu Zcela záleží na pořadí	60
Tabulka 4 - Defaultní zvuky úlohy typu Křížovky	63
Tabulka 5 - Defaultní zvuky úlohy typu Osmisměrky	63
Tabulka 6 - Defaultní zvuky úlohy typu Vyhledávání v textu	64
Tabulka 7 - Defaultní zvuky úlohy typu Posloupnosti.....	65
Tabulka 8 - Defaultní zvuky úlohy typu Kreslení.....	65
Tabulka 9 - Defaultní zvuky úlohy typu S umístěním	68
Tabulka 10 - Defaultní zvuky úlohy typu Bez umístění	68
Tabulka 11 - Defaultní zvuky úlohy typu Dvojice.....	70
Tabulka 12 - Defaultní zvuky úlohy typu Jedna odpověď	70

Seznam použitých zkratek

PDF	Portable Document Format
IDE	Integrated Development Environment
CLR	Common Language Runtime
CIL	Common Interface Language
WPF	Windows Presentation Foundation
WF	Win Form
DPI	Dots Per Inch
XAML	Extensible Application Markup Language
XML	Extensible Markup Language

Použité pojmy u úloh

Sada úloh	Vytvářená sada úloh v rámci práce
Skupina úloh	Určitá skupina úloh z vytvářené sady úloh
Typ/Varianta úlohy	Určitý typ úlohy ze skupiny úloh
Zadání úlohy	Jedna instance zadání určitého typu úlohy

Pro lepší pochopení těchto pojmů nahlédněte do přílohy (str. 79), kde je zobrazena hierarchická struktura vytvořených úloh. Na první úrovni je sada vytvořených úloh, druhá úroveň znázorňuje jednotlivé skupiny vytvořených úloh a na třetí úrovni jsou znázorněny jednotlivé typy vytvořených úloh. Pomyslnou čtvrtou úrovní v této hierarchii by pak byla zmíněná zadání.

Úvod

Se stárnutím člověka stárne i samotný mozek a jeho funkčnost. Při běžném stárnutí se lidem zpomalují reakce a mají určité poruchy paměti, ale i část mladší populace trpí nějakým postižením mozku. Mezi nejběžnější postižení mozku patří například Alzheimerova choroba nebo stařecká demence. U lidí postižených touto formou onemocnění postupem času dochází ke zhoršení paměti a to až do takové míry, že nejsou schopni si vybavit nedávné události. Mezi další příznaky patří také neschopnost řešit běžné problémy, se kterými se člověk setkává každý den. Tito lidé postupem času ztrácí jistou soběstačnost a jsou odkázáni na pomoc druhých.

Ke zpomalení nástupu mozkových onemocnění slouží různé formy mozkového tréninku. Pravidelným tréninkem lze zvýšit samostatnost a soběstačnost člověka. Trénování mozku sestává z trénování jeho jednotlivých částí a to buď jednotlivě izolovanými cvičeními, nebo souhrnnými cvičeními zaměřenými na více oblastí najednou. Jednotlivá cvičení poté sestávají z úloh, které lze rozdělit do skupin podle zaměření na trénovanou oblast mozku. Tento trénink je ale samozřejmě vhodný i pro zdravé jedince jako forma prevence.

Je tedy vhodné vytvořit sadu úloh vhodných pro trénování různých oblastí mozku. Výsledná sada úloh by měla být dostatečně interaktivní, aby byly úlohy pro uživatele zajímavější a uživatelsky přívětivější, a jednoduše ovladatelná. Složitější ovládání by uživatele pouze více mátl (možná i odrazovalo) a snadnější ovládání bude pro něj jednodušeji zapamatovatelné. Zároveň by však měla být zadání jednotlivých úloh co nejvíce originální, aby si jejich řešitelé neautomatizovali jejich řešení a úlohy si zachovali svůj pozitivní přínos.

1 Trénink kognitivních funkcí

Kapitola se zabývá úvodem do problematiky kognitivních funkcí. Popisuje tedy, jaké kognitivní funkce existují a také jak se dané funkce procvičují. V další části této kapitoly jsou popsány a porovnány již existující komerční řešení, určené právě k tréninku kognitivních funkcí. Závěr kapitoly popisuje skupiny úloh vybrané pro vytvoření v rámci této diplomové práce.

1.1 Kognitivní funkce

Kognitivní neboli poznávací (případně myšlenkové) funkce jsou schopnosti člověka pro zpracování okolních informací. Kognitivními funkcemi se zabývá tzv. kognitivní psychologie. Kognitivní psychologie tedy zkoumá, jak lidský mozek zpracovává okolní informace, které člověk přijímá smyslovými orgány (oči, uši, atd.). Zabývá se tedy vnímáním nejen podnětů, ale rovněž i uvažováním.

„Člověk si pomocí intelektuálních kognitivních schopností a jejich aktivity uvědomuje sám sebe i prostředí, v němž se nachází, a vytváří si tak postupně hodnotovou orientaci, která mu pomáhá řešit problémy, do nichž se v průběhu života dostává.“ [1] (str. 43)

Díky kognitivním schopnostem se lidé orientují ve světě kolem sebe i v sobě samých, přežívají v těžkých životních situacích a budují hodnotové žebříčky.

1.2 Dělení kognitivních funkcí

Kognitivní funkce dělí Klucká na 5 skupin: paměť, pozornost, jazyk a řečové schopnosti, zrakově-prostorové schopnosti a myšlení [2].

1.2.1 Paměť

Paměť je schopnost centrální nervové soustavy. Umožňuje nám přijímat, uchovávat a vybavovat si informace. Existují celkem 4 fáze paměti. První fází je kódování neboli vštípení. V této fázi se informace ukládají do paměti. Ukládání informací může být úmyslné (záměrné, např. učení) nebo neúmyslné (bezděčné, např. citově podbarvené okolnosti). Druhou fází paměti je retence neboli uchování. S touto fází úzce souvisí opakování sloužící k lepšímu zapamatování informací. Třetí a poslední fází procesu zapamatování je reprodukce neboli vybavení. V této fázi dochází k přesunu uchovaných informací do vědomí. Poslední fází paměti představuje zapominání. Zapominání znamená snížení znalostí dané informace. Účinným pomocníkem proti zapominání je opakování nebo uplatnění nově nabytých poznatků v praxi [3] [4].

Z hlediska délky trvání uchování informace v paměti dělíme paměť na 3 typy. Prvním typem je sensorická paměť uchovávající smyslová data (např. vizuální,

sluchová, hmatová, atd.). Obsah této paměti je uchován v rozmezí zlomku vteřiny až po dobu několika vteřin. Dalším typem paměti je krátkodobá paměť. V krátkodobé paměti se uchová aktuální mentální aktivita po dobu přibližně 30 až 40 vteřin. Krátkodobá paměť má omezenou kapacitu označovanou jako magické číslo a její rozsah je 7 ± 2 položky. To znamená, že většina lidí udrží v krátkodobé paměti 5 až 9 položek současně. Posledním typem paměti je paměť dlouhodobá. Do dlouhodobé paměti se ukládají informace z paměti krátkodobé, které byly několikrát opakovány nebo jsou pro člověka významné. Dlouhodobá paměť má různou dobu udržení informací v rozmezí několika dní až po délku celého života v závislosti na vztahu k dané informaci.

Příkladem trénování paměti je úloha, kdy se řešitel úlohy dozví (uslyší, přečte) sadu slov, které spolu mohou, ale také nemusejí souviset. Jeho úkolem je si tato slova zapamatovat a poté je vyslovit nahlas. Dle obtížnosti úlohy je řešitel nucen vyslovit slova ve stejném, obráceném nebo libovolném pořadí.

1.2.2 Pozornost

Pozornost je funkce vědomí sloužící k jeho usměrnění. Na člověka denně působí mnoho podnětů od různých zdrojů. Vědomí je třeba usměrňovat k vybírání pouze těch informací, které mají pro něj v dané situaci nějaký význam a ty ostatní ignorovat. K tomu slouží jedna ze základních vlastností pozornosti – selektivita. Další důležitou vlastností pozornosti je koncentrace neboli soustředění. Ta umožňuje člověku soustředit se na nějaký předmět po dobu několika vteřin. V závislosti na intenzitě soustředění se nepřímo úměrně mění počet objektů, kterým je člověk schopen věnovat pozornost. Posledními dvěma vlastnostmi jsou distribuce neboli rozdělení a vigilita neboli přenos pozornosti. Rozdělení pozornosti znamená rozdělit pozornost mezi více současných činností. Čím více má člověk zautomatizovaných činností, tím více jich je schopen vykonávat najednou. Přenosem pozornosti je myšlen přechod mezi jednotlivými činnostmi. To úzce souvisí s přizpůsobením se měnícím okolním podmínkám [5] [6].

Pozornost se dělí na 2 typy. První typ představuje pozornost bezděčná. Bezděčná pozornost má vrozený základ a je řízena reflexy člověka. Lze ji vyvolat novým, náhlým nebo neočekávaným podnětem. Například člověk uslyší ránu nebo se v místnosti rozsvítí světlo. Dalším typem je pozornost záměrná. Záměrná pozornost, jak napovídá název, je řízena vědomým cílem. Příkladem je každá posloupnost akcí, které musí člověk vykonat, aby dosáhl svého cíle. Například student učící se na zkoušku, aby u ní uspěl.

Příkladem pro trénování pozornosti je úloha, kdy se řešiteli předloží věta a jeho úkolem je vyznačit zde všechny samohlásky. Dalším příkladem může být úloha, v níž má řešitel nakreslit zrcadlově převrácený obraz předloženého obrázku. Jako měřítko této úlohy slouží délka a směr nakreslených čar.

1.2.3 Jazyk a řečové schopnosti

Jazyk je prostředek komunikace a řeč je konkrétní jazyková dovednost. Pomocí jazyka člověk vyjadřuje své myšlenky, pocity nebo emoce. Rovněž je také schopen komunikovat s dalšími osobami, což mu umožňuje lépe poznávat okolní svět. Jazyk se skládá ze znaků nebo slov, která nesou význam a vyjadřují konkrétní nebo abstraktní skutečnost.

Používání jazyka lze rozdělit na 2 kategorie. První kategorie zahrnuje produkci. Proces produkce sestává z uspořádání myšlenky do věty skládající se ze slov a vytvoření zvuku, který se nakonec předá posluchači. Příkladem může být zodpovězení otázky zkoušejícímu u ústní části zkoušky. Druhá kategorie zahrnuje porozumění. Proces porozumění probíhá v přesně opačném pořadí než proces produkce. Na začátku člověk slyší zvuky, kterým poté přiřadí význam slov. Ze slov si poskládá věty a z těch nakonec získá smysl původního sdělení.

Řeč zůstává při normálním stárnutí a bez poruch mozku zachována až do velmi vysokého věku. Postupným stárnutím ale upadá verbální fluence neboli slovní plynulost. Lidé trpící poškozením mozku nebo demencí mohou pociťovat obtíže s nalezením vhodných slov. Neschopnost rozumět jazyku a neschopnost jej užívat je označována jako afázie. Jde hlavně o ztrátu již naučených schopností dorozumívání se řečí a to mluvenou, čtenou nebo psanou [7].

Příkladem úlohy na procvičování řečových schopností může být úloha typu „vymyslete co nejvíce slov začínajících nebo končících nějakým písmenem“. Úlohu lze dále ztížit výběráním slov pouze z určité skupiny (např. názvy věcí nebo slovesa).

1.2.4 Zrakově-prostorové schopnosti

Do oblasti zrakově-prostorových schopností patří schopnosti vizuálně-konstrukční, vizuálně-motorické a percepční neboli vjemové. I když člověku správně fungují smyslové orgány, tak mohou být poškozeny části mozku zpracovávající informace z těchto smyslových orgánů. Tímto se zhoršuje vnímání okolí. Obtížná je rovněž konstrukce nebo umístění předmětů v prostoru.

Nejhorším případem poruchy prostorové orientace je neglekt syndrom. Lidé trpícími touto poruchou nevidí jednu polovinu předmětů v prostoru. Důsledkem této poruchy je pak například snědené jídlo jen z půlky talíře. Další možnou poruchou je ztráta schopnosti vidět předměty z určitého úhlu. Člověk s touto poruchou hůře nachází věci, jelikož se musí na všechna místa dívat z více úhlů. U osob s demencí se často vyskytuje vizuokonstrukční apraxie. Člověk trpící touto poruchou ztratil schopnosti používat dříve osvojené komplexní pohyby využívané v každodenním životě. Příkladem je dressing apraxie neboli neschopnost se obléknout [8].

Příkladem úlohy na procvičování zrakově-prostorových schopností může být úloha zaměřená na zrakové vyhledávání. Řešiteli je předložena tabulka obsahující několik obrázků. Cíl spočívá v označení symbolů vyskytujících se v tabulce dvakrát.

1.2.5 Myšlení

Jedná se o mentální manipulaci s informacemi. Myšlení člověku umožňuje dávat si informace do vzájemných souvislostí, vyvozovat závěry a řešit problémy. Druhy myšlení lze dělit z více hledisek. Příkladem dělení typů myšlení je konvergentní a divergentní myšlení. V případě konvergentního myšlení člověk hledá jedno správné řešení problému. V případě divergentního myšlení vyhledává více možných alternativ řešení problému. V praxi se uplatňují většinou oba typy najednou. Nejdříve divergentní ke zjištění možných alternativ a poté konvergentní k výběru jedné varianty [9].

Usuzování spočívá ve zpracování obecných informací, jehož výsledkem je určitý závěr. Usuzování dělíme na 2 typy. Prvním typem je dedukce. Pomocí dedukce člověk vyvozuje závěry z obecných informací. Dedukce nepřináší nové závěry. Druhým typem je indukce. Indukcí člověk vyvozuje od jedné určité informace více dalších obecných informací. Induktivní důsledky nejsou nikdy stoprocentní.

K řešení teoretických i praktických problémů směřujeme pomocí myšlenkových operací. Myšlenkové operace dělíme na 2 typy. Prvním typem jsou logické operace. Logické operace se řídí přesnými pravidly, kdy člověk postupuje jako počítač podle nějakého algoritmu. Druhým typem jsou heuristické operace. Člověk se s jejich pomocí dostane k výsledku bez zvážení všech možných alternativ a variant řešení. Tento typ je rychlejší, ale se zvýšenou pravděpodobností chyb.

Příkladem úlohy na procvičování myšlení je úloha, kdy tazatel sdělí řešiteli, že si myslí v hlavě nějaké slovo a cílem řešitele je klást tazateli vhodné otázky. Na základě získaných odpovědí se pak musí dobrat k tomuto slovu.

1.3 Existující řešení a jejich porovnání

Jako příklady byly vybrány 3 různé zdroje zabývající se problematikou kognitivního tréninku. První představuje web s online úlohami a několika elektronickými dokumenty zabývajícími se problematikou ztráty paměti (vše dostupné zdarma). Druhým představitelem je komerční sada neuropsychologických úloh určená k instalaci na počítač. Posledním zástupcem je web nabízející jak online trénink, tak i vlastní produkt určený pro instalaci na počítač (oboje placené).

1.3.1 Vzpomínkovi

Tato česká webová stránka se zabývá kognitivní rehabilitací a procvičováním paměti. Na webu jsou dostupné celkem 3 online úlohy a několik dokumentů s úlohami

ve formátu PDF. Všechny materiály byly vytvořeny ve spolupráci s Mgr. Jitkou Suchou, ergoterapeutkou z Gerontologického centra Praha [10].

První aplikací jsou interaktivní puzzle, kdy má uživatel za úkol složit obrázek podle předlohy a pokud chce, může si změřit dobu skládání. Další aplikací je test na procvičování vizuální a prostorové paměti vypracovaný ve dvou stupních obtížnosti. Obrazovka jednodušší varianty obsahuje okno rozdělené na 9 políček a 6 různých symbolů. Program vybere 3 různé symboly a ty umístí do políček. Úkol uživatele spočívá v zapamatování si symbolů zobrazených na obrazovce i s jejich umístěním a pomocí přetažení dané symboly rozmístit na původní správné pozice v tabulce. Časový limit na splnění úlohy je 3 minuty. Obtížnější varianta je založena na stejném principu. Okno je zde rozdělené na 16 políček a 8 různých symbolů. Časový limit na splnění úlohy jsou 2 minuty.

Poslední úlohou je aplikace na procvičování paměti a vybavení si informací. Aplikace zahrnuje dvě herní varianty - vybavení slov ve správném sledu a vybavení obrázků. Na obrazovce se zobrazí 4, 6 nebo 8 obrázků různých věcí podle volby uživatele. Úkolem uživatele je zobrazené obrázky si zapamatovat a po zobrazení 20 různých obrázků vybrat právě ty obrázky, které si měl zapamatovat. Varianta se slovy využívá stejný princip. Na obrazovce se zobrazí 3-7 slov podle volby uživatele a po zapamatování se zobrazí seznam 12 položek pro výběr.

Velkou nevýhodou těchto úloh je v podstatě jejich neměnnost. Například interaktivní puzzle mají pouze jednu předlohu. Symboly v testu na procvičování vizuální a prostorové paměti se sice po každém spuštění vyskytnou na jiných místech obrazovky, ale jsou vždy vybrány ze skupiny pevně daných symbolů. Aplikaci na procvičování paměti a vybavení informací se dá vytknout totéž jako testu na procvičování vizuální a prostorové paměti. Co se týká elektronických dokumentů, všechny úlohy jsou taktéž neměnné a jejich častým procvičováním už má uživatel zautomatizovaná řešení, tudíž je jejich přínos časem značně omezen.



Obrázek 1 – Vzpomínkovi [10]

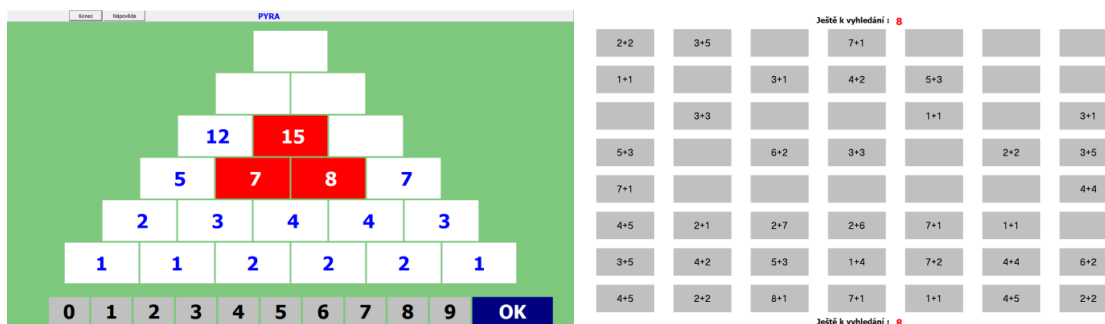
1.3.2 Neurop

NP3/NEUROOP-2 je nástupcem programů NEUROOP-1 a NEUROOP-2. Autorem tohoto programu je Dr.phil. Laco Gaál, neuropsycholog a programátor s dlouholetou klinickou praxí v oboru neuropsychologické rehabilitace [11].

NP3 je sada neuropsychologických programů. Uchovává statistiky o úspěšnosti řešení úloh jednotlivých pacientů a zároveň je zpracovává. Momentálně obsahuje 57 úloh. Úlohy lze rozdělit dle zaměření do několika skupin. Zajímavé jsou úlohy ze skupin strategické myšlení a plánování a konání. Ty nutí pacienta předem promýšlet posloupnosti jednotlivých kroků k dosažení požadovaného výsledku. Do této skupiny úloh patří například úlohy Hanoi (Hanoiské věže), R4 (Čtyři v řadě) či R5 (Pět v řadě). Podrobnější popis všech 57 úloh lze najít v uživatelské příručce. Uživatelská příručka je k dispozici v němčině, češtině a slovenštině a je součástí sady. Většina programů v sadě používá externí konfigurační soubory, úlohy se tak dají jednoduše modifikovat dle instrukcí v uživatelské příručce. V rámci modifikace je možné upravit textový obsah, obrázky a zvuky u úloh, které tyto změny umožňují.

Další dostupnou verzí programu je THNP/HNP, program navržený pro kontrolovaný neuropsychologický kognitivní trénink v domácím prostředí. Program sestává ze dvou modulů THNP a HNP. THNP je kolekce programů pro terapeutů. Jejich úkolem je vytvořit úlohy pro pacienty tzv. na míru. V konfiguraci úloh je definována obtížnost, obsah, pořadí úloh, jejich počet a další parametry. Konfigurace je zabezpečena proti narušení zvenčí a k jejímu nastavení mají přístup pouze terapeuti. Úlohy jsou předávány pacientům na paměťovém médiu. HNP je verze programu pro pacienty. Pacient má možnost spustit vybrané úlohy nebo komunikovat přímo s terapeutem. Celá procedura testování je kontrolována automaticky na pozadí (zahájení úlohy, uložení výsledků, uložení konfigurace).

Na webových stránkách je možné stáhnout demoverzi programu NP3 s 31 úlohami a uživatelskou příručku. Dále je možné zdarma stáhnout plnou verzi programu TOUCH-NEUROOP, což je jednoduchý komunikátor obsahující sadu vybraných úloh z NP3 určených pro dotyková zařízení. Možnou nevýhodou těchto aplikací může být pro někoho dnes již zastaralé a hůře přehledné grafické rozhraní. Naopak výhodou je určitě velká komplexnost aplikací a to jak z pohledu pokrytí testovaných oblastí, tak i měřených statistik jednotlivých pacientů. Další nespornou výhodou je možnost rozšiřování úloh díky jednoduché modifikaci. Mínusem pro pacienty je také cena těchto programů. První licence základní edice NP3 je dostupná v ceně 25 000 Kč, další licence vyjdou na 18 000 Kč. Kompletní edice vyjde na 55 000 Kč a další licence na 29 000 Kč (ceny platné k 14.4.2015).



Obrázek 2 – Neurop [11]

1.3.3 Brain jogging

Tato česká webová stránka se zabývá zvýšením povědomí lidí o mentálním tréninku. Na webu je dostupný online trénink nebo je možné objednat si program HAPPYneuron [12].

HAPPYneuron je program určený k tréninku kognitivních funkcí na počítači. Program je možné objednat na webu Brain jogging ve formě CD. Program je zaměřený na trénink posílení paměti, koncentrace, řečových funkcí, logického myšlení, vizuálně-prostorové orientace a mnoha dalších. Dále jsou zaznamenávány a vyhodnocovány výsledky pacienta.

Online trénink obsahuje samostatná cvičení a také cvičení s trenérem. Samostatná cvičení jsou rozdělena do 5 kategorií podle zaměření. U každé úlohy se před spuštěním zobrazí stručný popis úlohy, tedy návod k řešení a cílová kognitivní oblast, kterou daná úloha rozvíjí. Při spuštění úlohy se rovněž zobrazí konfigurační dialog, kde si uživatel může nastavit obtížnost dané úlohy. Po úspěšném absolvování úlohy se uživateli zobrazí jeho statistiky ve formě správnosti řešení a průměrného času řešení jednotlivých úkolů.

První kategorií je cvičení paměti. Příkladovou úlohou z této skupiny je úloha Barvy a čáry. Cílem této úlohy spočívá v zapamatování si 6, 8 nebo 10 postupně zobrazených obrázků. Po fázi zapamatování se postupně zobrazují nejen zapamatované obrázky, ale s nimi i několik dalších. Cílem je vybrat ty správné.

Druhou kategorií je pozornost. Testovanou úlohou je úloha Na lovu berušek. V průběhu řešení této úlohy se postupně zobrazují na hrací ploše beruška a její stín, který má uživatele rozptýlit. Cílem je včas kliknout na berušku. Pokud uživatel klikne mimo, ztrácí postupně životy. Po ztrátě tří životů je úloha dokončena.

Třetí kategorií je jazyk, který reprezentuje úloha Doplnovačka. Uživateli se zobrazují texty, kde jsou vynechána slova. Zároveň mu je k dispozici seznam chybějících slov a jeho úkolem je slova správně umístit do textu.

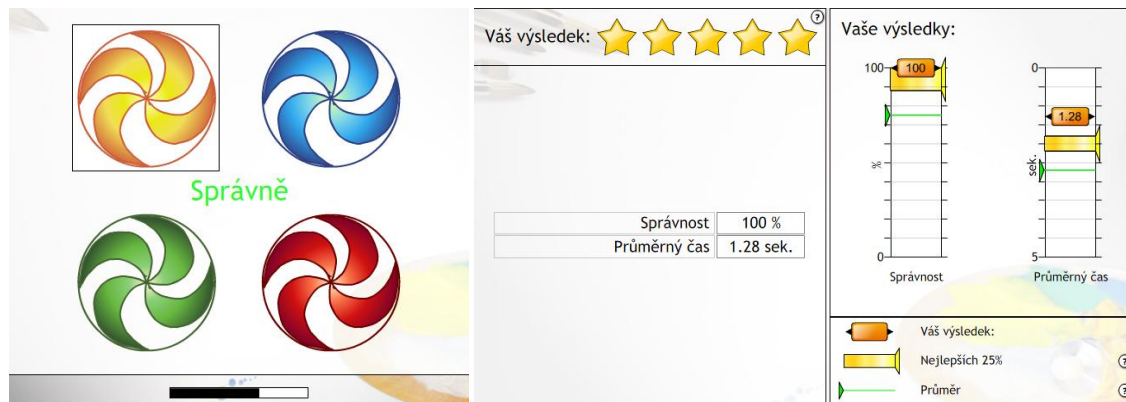
Další kategorií je uvažování. Zástupnou úlohou pro tuto kategorii je Basketbal. Řešiteli se zobrazí dvě podoby rozmístění očíslovaných míčů do basketbalových košů.

Jeho úkolem je najít nejmenší počet přesunů míčů, aby dosáhl situace na druhého rozmístění za podmínek, že z koše může vždy vytáhnout pouze horní míč a po upuštění míče do koše je míč úplně nahoře.

Poslední kategorií je prostorová představivost. Příkladovou úlohou z této skupiny je úloha Hra s perspektivou. Uživateli se zobrazí půdorys scény a obrázek, co vidí ze svého pohledu. Cílem je vybrat správné místo v půdorysu, odkud uživatel scénu vidí. Ve druhé části této úlohy je situace opačná. Uživatel ví, odkud se dívá a má za úkol vybrat obrázek toho, co z daného místa vidí.

Cvičení s trenérem je posloupnost několika úloh, které se vhodně doplňují, za sebou. Počet úloh se odvíjí od doby tréninku. Například v tréninku na 20 minut je uživatel proveden 3 úlohami.

Zájemci si mohou vyzkoušet online trénink zdarma po dobu 7 dnů. V této omezené verzi jsou dostupné výše popsané úlohy (v každé kategorii však pouze jedna) a dvě dvacetiminutová cvičení s trenérem. Pokud se tento program zalíbí, měsíční přístup k online tréninku stojí 349 Kč. Roční přístup pak vyjde na 1899 Kč. CD s programem HAPPYneuron stojí 1499 Kč (ceny platné k 14.4.2015). Nevýhoda této verze spočívá v dostupnosti všech statistik uživatele pouze na počítači, kde je program nainstalovaný. V porovnání s programy Neurop má Brain jogging moderní uživatelské rozhraní podložené příjemným zvukovým doprovodem, což je určitě značné plus.



Obrázek 3 - Online Brain jogging [12]

1.4 Rozdělení a výběr úloh pro tuto práci

Následuje popis rozdělení úloh vybraných pro tvorbu, v rámci této diplomové práce, z hlediska kognitivního zaměření jednotlivých skupin.

1.4.1 Posloupnosti akcí

Tato skupina obsahuje úlohy vhodné k nasimulování běžných každodenních situací, se kterými se člověk setkává právě skoro každý den. Tato cvičení jsou vhodná především pro lidi neuvědomující si posloupnosti svých vykonaných akcí (například lidé trpící Alzheimerovou chorobou nebo stařeckou demencí). Postupným cvičením

těchto úloh, kdy pacient trénuje takovéto každodenní situace běžné jako například uklízení, oblékání nebo nakupování, se zpomaluje zapomínání těchto činností a prodlužuje se tak pacientova soběstačnost a samostatnost.

1.4.2 Křížovky a osmisměrky

V této skupině jsou úlohy zaměřené hlavně na schopnost vybavit si a dále na zrakové vyhledávání. První část úloh je vhodná ke cvičení pro osoby mající problémy s vybavováním slov, která znají/znali. Trénink vybavnosti probíhá formou textové nápovědy, ke které je jen jedna správná odpověď. Druhá část obsahuje úlohy určené pro osoby s poruchami zrakového vnímání. Tato cvičení probíhají formou hledání písmen a z nich poté následného tvoření slov.

1.4.3 Spojovačky a obrázky

Tato skupina obsahuje úlohy určené hlavně k procvičení pozornosti a zrakového vyhledávání. Tyto úlohy, jako v podstatě úlohy z předchozí skupiny, jsou určeny pro osoby s poruchami zrakového vnímání. Avšak na rozdíl od úloh z předchozí skupiny, obsahují čísla a zároveň je potřeba zvýšené soustředěnosti kvůli řazení čísel v posloupnosti.

1.4.4 Paměť

Zde jsou úlohy zaměřené hlavně na procvičení paměti a vizuální a prostorové orientace. Všechny typy úloh jsou vhodné ke cvičení paměti. Cíl spočívá vždy v zapamatování si a vybavení již zobrazených předmětů. Jednotlivé typy se ale liší dalšími informacemi k zapamatování (poloha a tvar předmětů v prostoru) nebo rozpoznáním předmětů mezi mnoha dalšími.

1.4.5 Odpovědi a dvojice

Tato skupina je, co se zaměření na kognitivní funkce týká, poněkud univerzální skupinou úloh. Zde je možné vytvořit úlohy v podstatě pro všechny skupiny kognitivních funkcí. To je dáno tím, že úlohy v této skupině jsou řešeny formou otázky a cílem je vybrat správnou odpověď (popřípadě jinak modifikovanou variantu výběru možností).

2 Cíle práce

Cílem této práce je vytvořit úlohy vhodné pro různé typy kognitivního cvičení. Úlohy budou vhodně rozděleny do alespoň 5 skupin podle jejich zaměření a způsobu řešení. Vybrané skupiny úloh se budou zaměřovat hlavně na oblasti trénování paměti, procvičování navazujících akcí a trénování zrakově-prostorových schopností.

Tyto tři oblasti byly vybrány z důvodu jejich přímého propojení s orientací v okolním světě. Vytvořené úlohy budou vhodné hlavně pro osoby trpící ztrátou paměti nebo neuvědoměním si posloupností již vykonaných akcí.

Všechny úlohy budou vytvořeny s ohledem na jejich jednoduchou modifikaci a snadné ovládání. Ovládání úloh bude umožněno pomocí jednoduchých posloupností dotyků. Modifikace úloh bude sestávat z proměnlivosti zadání.

Úlohy budou vykazovat interaktivní chování ve formě zvukových a grafických podnětů. Tím budou pro uživatele zajímavější a uživatelsky přívětivější. U úloh, kde to bude možné, bude umožněno využívat vlastní zvukové podněty. Avšak cílem práce není vytvořit dokonalý editor na tvorbu všech těchto typů úloh.

V rámci této práce budou vytvořené úlohy rovněž základně otestovány na několika uživateli. Předmětem otestování bude zejména srozumitelnost a snadnost jejich ovládání. Kognitivní přínos úloh nebude předmětem tohoto otestování, protože je dán teprve až jejich dlouhodobým využíváním.

3 Návrh řešení

Tato kapitola obsahuje popis technologií vhodných k tvorbě vytvářených úloh. Popisuje prostředí pro vývoj a režimy činnosti úloh. Nakonec je uveden popis úloh navržených k vytvoření.

3.1 Použité technologie

Následuje popis technologií použitých k tvorbě vytvářených úloh.

3.1.1 C# a .NET

Ke tvorbě úloh bude využit již existující základ dodaný vedoucím práce. Toto základní prostředí je napsáno v jazyce C# s využitím .NET frameworku. .NET framework sestává ze 4 komponent [13]:

- Jazyk – v našem případě C# přizpůsobený pro vývoj v .NET.
- Visual Studio – moderní IDE pro vývoj aplikací (úlohy budou vyvíjeny ve VS 2013 Community edition).
- Virtuální stroj – CLR převádí CIL do instrukcí fyzického procesoru.
- Knihovny – obsáhlé knihovny s předpřipravenou řadou struktur a komponent.

Využitým prostředkem pro tvorbu grafických aplikací v C# .NET je WPF (Windows Presentation Framework). WPF je součástí .NET frameworku určená pro tvorbu formulářových a grafických aplikací. Obsahuje velké množství formulářových prvků a umožňuje jejich obsáhlé stylování. Tvorba formulářů probíhá skládáním jednotlivých prvků na plochu obrazovky. Výsledná aplikace pak sestává z několika grafických komponent, které se mohou překrývat.

Velkou výhodou WPF oproti původnímu WF (WinForm) je zavedení jednotky DPI a čistě vektorové grafiky. Toto bylo zavedeno z důvodu přenositelnosti aplikací mezi zařízeními s různým rozlišením, kde s využitím WPF působí aplikace na všech zařízeních stejným vzhledem. WPF také zavedlo definici formulářů pomocí jazyka XAML, což umožňuje lepší oddělení prezentační a logické vrstvy aplikace [14].

3.1.2 Prostředí pro vývoj

Již zmíněné základní prostředí dodané vedoucím práce poskytuje potřebný backend pro tvorbu úloh. Každá skupina úloh je tvořena alespoň jednou programovou třídou. Třídě se při spuštění předá parametr s možnou variantou úlohy ve skupině. Podle hodnoty předaného parametru se tedy může vytvořit příslušná modifikace (typ) úlohy.

Kromě programové třídy musí každá úloha obsahovat také záznam v konfiguračním souboru celé aplikace a soubor obsahující lokalizované texty pro úlohu.

Pro přidání nové úlohy je potřeba vytvořit novou programovou třídu. V prostředí je dostupný soubor **TaskEmptyMain.cs**, což je výchozí vzorový zdrojový soubor. Tento soubor je potřeba vhodně pojmenovat a je pozdějším vstupním místem pro vytvářenou úlohu. Aby byla úloha skutečně začleněna do aplikace, je potřeba rovněž přidat její záznam do konfiguračního XML souboru v následující podobě.

```
<Task Class = "Module.Tasks.MojeUloha.TaskMojeUlohaMain"  
Texts = "Applications/MyApplication/Texts/MojeUloha.txt"  
Manual = " Applications/MyApplication/Docs/MojeUloha.pdf" />
```

Ke každé úloze lze vytvořit textový soubor s lokalizací úlohy do více jazyků. Tento textový dokument není povinný, ale v případě psaní aplikace ve více jazycích je tento postup velmi vhodný. Textový soubor je pojmenovaný podle názvu úlohy a jazyka lokalizace (například **MojeUlohaCZ.txt**).

Při prvním spuštění úlohy se vytvoří instance programové třídy úlohy a při opuštění úlohy je tato instance zachována pro rychlé opětovné znovuspuštění, pouze se z ní odstraní většina grafických komponent. Uchovávají se však stavy všech položek třídy. Třída úlohy se skládá z několika přetížených metod doplněných vlastním kódem. Metody jsou určeny k přecházení mezi stavy stavového automatu, odrážejícího jednotlivé stavy hry, ale také k obsluze vlastních uživatelských akcí. Stavový automat je popsán v další podkapitole. Metody pro obsluhu uživatelských událostí jsou například:

- **TaskOnUserButton(String ID)** – stisk vlastního nadefinovaného tlačítka. Podle ID víme, které tlačítko bylo stisknuto a voláme příslušnou metodu obsluhy.
- **TaskOnDropFile(FileInfo[] files)** – obsluha upuštění souborů na plochu úlohy.

Každé modifikaci (typu) úlohy se po spuštění programu vytvoří samostatné spouštěcí tlačítko. Všechny modifikace jedné programové třídy jsou v jedné skupině úloh. K rozeznání modifikací úlohy je třeba ve třídě definovat enum typu **EnumTaskType**. Každá hodnota v **EnumTaskType** je přiřazena k jednomu vstupnímu bodu **TaskEntry**. Statická metoda **TaskEntries()** vrací pole všech nadefinovaných **TaskEntry** v programové třídě. Všechny modifikace mohou mezi sebou sdílet jak manuál, tak i jazykový soubor.

Každá úloha po spuštění obsahuje také informační řádek. Na informačním řádku jsou zobrazeny textové položky. Zde se dají zobrazit například informace typu stav a název úlohy, počet tahů, chyb a čas běhu úlohy.

Na ovládací panel úlohy se do řádků vkládají tlačítka pro řízení úlohy. V prostředí jsou některá tlačítka předdefinována a je rovněž možné dodefinovat si svá vlastní. Poté lze nastavit, která tlačítka se zobrazí. Předdefinovaná tlačítka a jejich účel jsou:

- **Start** – spuštění úlohy.
- **Next** – další krok úlohy.
- **Stop** – zastavení úlohy.
- **LevelMinus** – snížení úrovně obtížnosti úlohy.
- **LevelPlus** – zvýšení úrovně obtížnosti úlohy.
- **Settings** – zobrazení konfigurace pro úlohu.
- **Info** – zobrazení popisu úlohy.
- **FullScreen** – zobrazení přes celou obrazovku.
- **Back** – ukončení úlohy.

Na hlavní grid úlohy lze vložit předdefinované komponenty typu **Canvas**, **Grid** nebo **Viewport3D** a na tyto komponenty vkládat další komponenty poděděné od třídy **UIElement**. Takto se postupně skládá celý grafický obsah úlohy. Ve výsledku tvoří všechny zobrazené komponenty na hlavním gridu v podstatě strom jednotlivých objektů, jehož kořenem je právě hlavní grid.

Pro ukládání a načítání dat lze využít některé ze zdrojových míst v adresáři **AppFilesData**. Každá úloha a každé zadání úlohy by měly být v oddělených adresářích. Pro jednotný přístup k datům lze využít metodu **GetPathFilesData(EnumPathData enum, String[] dirs)**. Jednotlivé položky tohoto enumu obsahují odkazy do jednotlivých složek úložiště:

- **CommonData** – datové soubory pro všechny aplikace.
- **CommonSounds** – zvukové soubory pro všechny aplikace.
- **ThisApp** – datové soubory pro aktuální aplikaci.
- **ThisAppData** – datové soubory pro aktuální aplikaci společně všem úlohám.
- **ThisAppSounds** – zvukové soubory pro aktuální aplikaci společně všem úlohám.
- **ThisAppDataTask** – datové soubory pro aktuální úlohu.

3.1.3 Stavový automat

Možné stavy automatu / úlohy jsou:

- **Create** – vytváření úlohy.
- **Init** – inicializace dat úlohy.
- **Ready** – úloha je připravena ke spuštění.
- **Run** – úloha je spuštěna.
- **Stop** – násilné zastavení uživatelem v průběhu řešení úlohy.
- **Finished** – úspěšně dokončená úloha.
- **Close** – násilné ukončení úlohy stiskem tlačítka zpět.

Metody obsluhující přecházení mezi stavy jsou:

- **TaskOnCreate ()** – první vytvoření instance třídy.
- **TaskOnInit ()** – každé načtení / znovuspuštění úlohy.
- **TaskOnClose ()** – uzavření nebo opuštění úlohy.
- **TaskOnNew ()** – generování nového zadání.
- **TaskOnStart ()** – spuštění úlohy, například uživatelem.
- **TaskOnNext ()** – další krok úlohy.
- **TaskOnStop ()** – zastavení úlohy, například uživatelem.
- **TaskOnFinish ()** – úspěšné ukončení úlohy.
- **TaskOnData ()** – zpracování výsledků úlohy.
- **TaskAction(EnumTaskAction enum)** – změna stavu, volá metody **TaskTo... ()** a ty poté volají **TaskOn... ()**.

3.2 Režimy činnosti úlohy

Všechny vytvářené úlohy mají dva režimy své činnosti.

3.2.1 Režim konfigurace

Prvním režim spočívá v konfiguraci úlohy. V tomto režimu uživatel nebo terapeut v podstatě spravuje zadání úlohy. Do oblasti správy úlohy patří přidávání nových zadání, odstranění nebo modifikace existujících zadání. Každé zadání úlohy je uloženo ve vlastním konfiguračním souboru. Bližší detaily ohledně tvorby zadání úloh jsou uvedeny ve vlastní kapitole (str. 49).

3.2.2 Režim běhu

Druhým režimem je vlastní běh úlohy, kdy uživatel řeší zadanou úlohu. Při spuštění úlohy je přehráno zvukové zadání a je spuštěna časomíra. Zvukové zadání je

přehráno buď defaultní, pro každý typ úlohy jiné, nebo může být přehráno zvukové zadání vázané k určitému zadání úlohy. Defaultní zadání je přehráno v okamžiku, kdy zadání úlohy nemá nastavené vlastní zvukové zadání. Průběh hry je doprovázen dalšími pro jednotlivé úlohy specifickými zvukovými nebo vizuálními reakcemi. Specifické reakce jsou popsány v kapitole (str. 59) obsahující popis řešení úloh.

Po úspěšném ukončení úlohy dojde k přehraní zvuku oznamujícího splnění úlohy a poté zobrazení dialogu s uživatelskými aktuálními statistikami v rámci aktuálního běhu aplikace. Výstup řešení jednotlivých typů úloh zahrnuje rovněž jejich hodnocení. Hodnocení je založeno na celkové úspěšnosti uživatele. Úspěšnost se hodnotí jako poměr správných tahů k celkovému počtu tahů. Pomocným výstupem, zobrazeným v dialogu, je průměrný čas řešení úlohy.

3.3 Úlohy vybrané pro vytvoření

Následuje popis jednotlivých úloh vybraných pro vytvoření, v rámci této práce. Podkapitoly jsou rozděleny podle kategorie, kam daná úloha přísluší.

3.3.1 Posloupnosti akcí

Cíl úlohy spočívá v rozmístění předmětů na svá správná místa. Po roztřídění všech předmětů v definovaném pořadí (existuje-li nějaké), je úloha úspěšně dokončena. Vytvořené úlohy jsou rozděleny podle způsobu návaznosti jednotlivých akcí. Jednotlivé varianty reflektují situace, kdy nezáleží na posloupnosti akcí, částečně záleží na posloupnostech akcí a zcela záleží na posloupnosti vykonávaných akcí.

U prvního typu, kdy nezáleží na posloupnosti akcí, záleží pouze na cílovém stavu úlohy. Cílem je všechny položky správně rozmístit do připravených oblastí, kam skutečně patří, ale to v libovolném pořadí. Příkladem úlohy je například běžný úklid pokoje, kdy je třeba pouze rozmístit jednotlivé předměty na svá správná místa. Například kniha patří na poličku a polštář patří na postel. Pro každý předmět existuje pouze jedno správné umístění, ale na každé cílové místo lze umístit i více předmětů.

U druhého typu již částečně záleží na posloupnosti vykonávaných akcí. Zde je opět potřeba rozmístit položky do jednotlivých cílových oblastí, tedy kam patří, ale předměty musejí být do oblasti vkládány již v určitém pořadí. Vhodným příkladem pro tento typ úlohy je například oblékání, kdy oblasti představují části lidského těla a rozmisťované předměty zahrnují kusy oblečení. Pro každou oblast je definováno, v jakém pořadí lze na sebe obléci jednotlivé kusy oblečení. Například nelze obout boty, dokud nebudou navlečeny ponožky.

Třetí typ obsahuje úlohy, kde kompletně záleží na posloupnosti vykonávaných akcí. Zde je pro změnu definováno pořadí oblastí. Do další oblasti v pořadí lze vkládat položky teprve, až když aktuální (neboli předešlá) oblast obsahuje všechny potřebné

položky, které se do ní mají umístit. Jako ukázkový příklad si lze představit libovolný postup, který lze popsat jako posloupnost úkonů. Například proces nakupování. Na začátku je jasně dané, že si člověk vezme nákupní košík. Do nákupního košíku v dalším kroku umísťuje obsah nákupu v podstatě v libovolném pořadí a až nakonec jde k pokladně a za nákup zaplatí. Není tedy možné zaplatit a až poté umístit do košíku brambory nebo jít na nákup bez košíku. Pro tyto úlohy existuje pouze jeden globální (nikoli dílčí) správný postup, který by měl být dodržen. Lze si představit situaci, kdy existuje i nějaký poněkud alternativní postup, ale je lepší používat raději ten jeden ověřený.

Jako pozadí úlohy i jednotlivé obrázky položek lze použít fotografie reálných předmětů, které uživatel běžně používá, což zvyšuje efektivitu tréninku. Všechny zvukové záznamy může namluvit někdo, kdo je uživateli blízký, čímž se pro něj dále zvyšuje reálnost interakce s okolím.

3.3.2 Křížovky a osmisměrky

Po spuštění úlohy je cílem vyřešit tajenku nebo správně nalézt všechna slova ukrytá v textu. Prvním typem úloh jsou běžné doplňovačky ve formě křížovky. Úloha obsahuje textová políčka pro vpisování písmen a seznam nápověd (legendu). Cílem úlohy je ke každé nápovědě vyplnit správné slovo. K dispozici je řešiteli délka slova (počet políček pro zadání písmen). Po vyplnění všech slov vyvstane kompletní tajenka úlohy, která je barevně odlišena od ostatních polí.

Dalším typem úlohy jsou osmisměrky. Každá osmisměrka se skládá z matice písmen a obsahuje soubor slov, která je potřeba v matici vyhledat. Hledání slov je možné horizontálně, vertikálně i diagonálně a to jak zleva doprava, tak obráceně. Po označení všech slov se postupně přečtou po řádcích zleva doprava nevyužitá písmena (písmena nepatřící do žádného vyhledávaného slova) a ta představují ukrytou tajenku.

Posledním typem úlohy je vyhledávání v textu. Zde je jako v osmisměrce seznam slov, která je potřeba vyhledat v matici písmen. Vyhledává se ale v souvislém textu, tzv. řádek po řádku zleva doprava a po vyhledání všech ukrytých slov zbývající písmena netvoří žádnou tajenku.

3.3.3 Spojovačky a obrázky

Po spuštění úlohy je cílem správně seřadit čísla od nejmenšího po největší. U prvního typu úlohy je cílem správně označit čísla, rozmístěná náhodně po ploše, od nejmenšího po největší. Po označení všech čísel ve správném pořadí je úloha úspěšně dokončena.

Druhý typ úlohy jsou obrázkové spojovačky. Zadáním této úlohy je obrázek s body. Každý bod zahrnuje číslo s pořadím daného bodu. Cílem úlohy je označit

postupně body v pořadí od 1 do N (počet bodů obrázku). Při spojení dvou následných bodů se mezi nimi vykreslí spojnice. Po spojení všech bodů je vytvořen obrázek podle zadání a úloha je úspěšně dokončena.

3.3.4 Paměť

Po spuštění úlohy je cílem si zapamatovat zobrazené symboly. Po uplynutí časového limitu jsou symboly přesunuty nebo zpřeházeny a úkolem je vzpomenout si, které symboly byly původně zobrazeny, popřípadě i jejich výchozí umístění.

U prvního typu úlohy jsou nejprve zobrazeny symboly v matici. Po nějaké době symboly z matice zmizí a cílem řešitele je rozmístit zapamatované symboly zpět do matice na svá místa, tedy kde byly původně zobrazeny.

U druhého typu úlohy je zobrazeno rovněž několik symbolů, které si musí řešitel zapamatovat. Po uplynutí časového intervalu je zobrazeno více symbolů, mezi nimiž jsou i ty původně zobrazené symboly. Cílem je ze všech zobrazených symbolů správně vybrat zapamatované, tedy původně zobrazené symboly.

3.3.5 Odpovědi a dvojice

Oba typy úlohy mají zadání v textové formě, avšak ne zcela nutně vždy doplněné o pomocný obrázek, a obsahují několik možností pro výběr odpovědi. Prvním typem úlohy je spojování slovních dvojic. Úloha obsahuje nějak modifikované zadání typu „najděte odpovídající si dvojice“. Příklad zadání úlohy může být: „Přiřaďte k sobě správně dvojice typu: země – hlavní město.“ Dále úloha obsahuje dva seznamy položek, ze kterých je potřeba k sobě přiřadit vždy po jednom slovu.

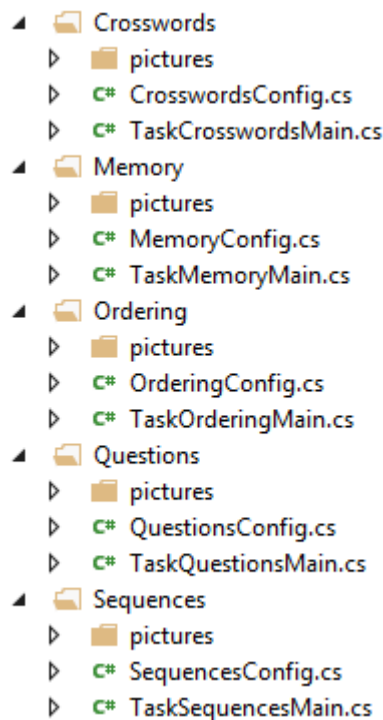
Druhý typ úlohy je výběr správné možnosti. Jako u předchozího typu obsahuje úloha zadání a k němu seznam odpovědí. Cílem je vybrat správnou odpověď ze seznamu nabízených možností. Tento typ pokrývá velké množství variací úloh. Dají se zde aplikovat všechny úlohy typu „odpověz na otázku“. Příkladem je zadání úlohy: „Rozpoznej náladu osoby na fotografii“ doplněné o fotografii obličeje člověka. Úkolem je vybrat odpověď ze seznamu možností: smutek, štěstí nebo zděšení.

4 Implementace

Tato kapitola obsahuje popis implementace úloh a způsob uložení zadání jednotlivých variant úloh.

Zdrojové soubory každé skupiny úloh jsou umístěny v samostatném adresáři (projektu) pro každou skupinu úloh. Každý adresář obsahuje soubor **TaskNameMain.cs**, ve které je v namespace **Module.Tasks.TaskName** definována třída **TaskNameMain**. Tato třída definuje chování úlohy (místo „Name“ je vždy použit název konkrétní úlohy).

Aktuální podobu všech typů dané úlohy definuje konfigurační třída **NameConfig**, umístěna v namespace **Module.Tasks.TaskName**. Každá programová třída úlohy obsahuje referenci na vlastní instanci konfigurační třídy. Z této třídy se následně při uložení v konfiguračním módu automaticky generuje soubor **Settings.xml**, který obsahuje popis jednoho zadání úlohy. Ukázkové výsledné XML soubory s konfigurací zadání pro úlohy jsou dostupné v přílohách (str. 74) k práci.



Obrázek 4 - Uspořádání skupin úloh v projektu MS Visual Studio

4.1 Tvorba a ovládání úloh

V této části budou popsány hlavní třídy definující chování úloh. Nejprve budou popsány věci společné pro všechny skupiny úloh a dále v podkapitolách věci specifické pro každou příslušnou skupinu.

Obecně pro všechny skupiny úloh platí, že jejich hlavní třída je potomkem generické třídy **TaskClassBase**, definující v podstatě chování stavového automatu

tvořícího stavu úloh. Přetěžují se jednotlivé metody vytvářející přechody mezi stavy a obsluhující jiné další uživatelské akce.

Třída obsahuje statické pole **TaskEntry** pro definici jednotlivých typů této úlohy. Jedna položka v tomto poli definuje jednu variantu úlohy. K tomu je přiřazen enum typu **EnumTaskType** uchovávající aktuální typ úlohy po spuštění úlohy. Dalším enumem ve třídě je enum typu **EnumState** uchovávající typ aktuálního stavu úlohy. K obsluze dodatečných uživatelských akcí je ve třídě definován enum typu **EnumEditAction** společně s textovými konstantami pro jednotlivá tlačítka. Dále jsou ve třídě definovány: seznam názvů zadání, odkaz na objekt konfigurace aktuálního zadání, hashmap pro ukládání statistik řešení úloh (existuje pouze v paměti) a info položky zobrazené v informačním řádku úloh (název zadání, počet tahů a počet chyb).

Pouze při prvním vytvoření určitého typu úlohy se volá metoda **TaskOnCreate()**. Zde se nastavuje zobrazení info položek úlohy, zobrazení tlačítek a přidání typů komponent na hlavní grid úlohy.

Metoda, která se volá při každém otevření určitého typu úlohy, je **TaskOnInit()**. Zde se nastavuje aktuální typ spuštěné úlohy a generuje se zadání.

U úloh, kde je na hlavní komponentu umístěn canvas, je potřeba přetížít metodu **TaskOnSizeChange()**. Zde se pak nastavuje měřítko pro využití relativních souřadnic canvasu a překreslují se položky na canvasu.

Metoda sloužící k obsluze stisknutí vlastních tlačítek (pro ovládání úlohy) je metoda **TaskOnUserButton(String ID)**. Do metody se předává textová konstanta s identifikátorem tlačítka. Zde se pak volají další příslušné metody pro obsluhu těchto tlačítek.

Úlohy, které umísťují na canvas obrázek na pozadí, přetěžují metodu **TaskOnDropFile(FileInfo[] files)**. V této metodě se ošetří obsluha přidání obrázku na pozadí a případného zapojení i zvukového zadání.

Metoda volaná při spuštění nového zadání dané varianty úlohy je metoda **TaskOnNew()**. V této metodě se nastavuje viditelnost tlačítek podle aktuálního režimu činnosti, inicializují se zde hodnoty pro zadání úlohy, načítají se hodnoty zadání úlohy z konfigurace a přehrává se zvukové zadání úlohy.

Pro vykonání jednotlivých tahů v režimu běhu i úpravy úloh v režimu konfigurace jsou uživatelem vykonané tahy odchyťávány. Odchyťávají se tahy vykonané prostřednictvím myši nebo prstů (na dotykové obrazovce). Z obou variant těchto handlerů se pak vždy volá příslušná metoda. Jediný typ události vyvolaný pomocí dotyků, který není odchyten, je manipulace odpovídající události vyvolané pohybem kolečka myši, ta je ale potřeba pouze v konfiguračním módu pro změnu

velikosti objektů u 2 skupin úloh a nijak nenarušuje proces řešení úloh na dotykových zařízeních.

Reakcí na vykonané tahy je kontrola, zda nebyl aktuální správný tah posledním při řešení úlohy. Tuto funkcionalitu obsluhuje metoda **TaskOnNext()**. V případě, že byl tah poslední, tak je úloha úspěšně dokončena a volána metoda **TaskOnFinish()**. V této metodě se do hashmapy s výsledky ukládají hodnoty aktuálního řešení (úspěšnost a čas). Po úspěšném vykonání této metody se volá metoda **TaskOnData()**. Tato metoda je určena ke zpracování výsledků a zobrazuje uživateli dialog s jeho aktuálními statistikami.

4.1.1 Posloupnosti akcí

V této skupině úloh je v metodě **TaskOnCreate()** přidán na hlavní grid (vytvořený v frameworku) úlohy canvas (kreslicí plátno). Na canvas se rozmisťují jednotlivé oblasti a jejich obrázky v metodě **TaskOnNew()** při načtení konfigurace aktuálního zadání úlohy. V režimu editace se jednotlivé obrázky umisťují podle tvorby úlohy do svých cílových oblastí. V režimu běhu se při spuštění úlohy všechny obrázky umístí do startovní oblasti.

V obou režimech je umožněno přesouvat obrázky mezi jednotlivými oblastmi. Jediné omezení zde je zakázání posunování obrázků v režimu běhu, kdy už je obrázek umístěn do své správné cílové oblasti. Přesouvání všech obrázků je dáno nastavením handlerů pro události manipulující s jednotlivými objekty typu **Item**. Odchycené události jsou uchopení obrázku, posun obrázku po canvasu a uvolnění obrázku.

```
// zachycení vyzvednutí obrázku z Area
image.MouseLeftButtonDown += (sender, e) =>
{
    // volání metody pro obsluhu vyzvednutí obrázku
    CaptureImage(image, e.GetPosition(canvas));
    e.Handled = true; // zachycení obslouženo
};

// zachycení vyzvednutí obrázku z Area (prstem)
image.TouchDown += (sender, e) =>
{
    // volání metody pro obsluhu vyzvednutí obrázku
    CaptureImage(image, e.GetTouchPoint(canvas).Position);
    e.Handled = true; // zachycení obslouženo
};
```

V režimu konfigurace je umožněna manipulace s vytvořenými oblastmi. Jednotlivé oblasti je možné přesouvat a měnit jejich rozměry. Startovní oblast pro počáteční umístění předmětů navíc umožňuje přijímat upuštěné soubory s obrázky.


```

// zachycení upuštění souboru obrázku nad AreaStart
area.borderPtr.Drop += (sender, e) =>
{
    // jedná se o upuštěný soubor
    if (e.Data.GetDataPresent(DataFormats.FileDrop))
    {
        // volání metody pro obsluhu upuštění souboru obrázku
        DropFileOnArea(e.Data.GetData(DataFormats.FileDrop));
    }
    e.Handled = true; // zachycení obslouženo
};

```

V režimu běhu se při přesunutí obrázku ze startovní do cílové oblasti volá metoda **TaskOnNext()** při úspěšně vykonaném kroku. V ní se kontroluje splnění úlohy počtem zbývajících obrázků ve startovní oblasti. Pokud startovní oblast neobsahuje již žádný další předmět, je úloha úspěšně dokončena.

```

// kontrola posledního kroku
// režim řešení úlohy a je prázdná startovní oblast
if (state == EnumState.Game &&
    config.AreaStart.stackPnlPtr.Children.Count == 0)
{
    PlaySound(SoundState.FINISH); // přehraj zvuk konec
    // návratová hodnota (nebude další krok úlohy)
    return TaskConstants.EnumTaskOnReturn.None;
}
// návratová hodnota (bude další krok úlohy)
return TaskConstants.EnumTaskOnReturn.OK;
};

```

4.1.2 Křížovky a osmisměrky

V této skupině úloh jsou v metodě **TaskOnCreate()** přidány na hlavní grid úlohy dva další gridy. První grid je určen pro zobrazení matice písmen (vlevo), druhý grid umístěný napravo slouží k zobrazení legendy. Do jednotlivých buněk obou gridů se vkládají oblasti a jejich textové položky v metodě **TaskOnNew()** při načtení konfigurace aktuálního zadání úlohy.

V režimu konfigurace se nastavují rozměry obou matic a vyplňují jejich textové položky. Jedinou výjimkou je matice u úlohy typu vyhledávání v textu, jejíž obsah se generuje náhodně. U tohoto typu úlohy je potřeba, aby matice obsahovala celkem dvojnásobek polí, kolik mají písmena všech slov ze seznamu (vpravo). V případě, že toto není dodrženo autorem úlohy, doplní se potřebné sloupce a řádky v matici na levé straně automaticky. Umístění slov do souvislého textu se poté uskuteční jednoduchým umístovacím algoritmem. Náhodně se vybere první buňka, do které se umístí první písmeno prvního slova. Do dalších následujících buněk se poté vyplní zbytek tohoto slova. Poté se vynechá stejný počet buněk jako je počet písmen tohoto dříve umístěného slova a začne se umisťovat další slovo a takto se pokračuje stále dále, dokud nejsou umístěna všechna slova (z pravého seznamu).

V režimu běhu jsou všechny události vykonávané pouze stiskem na jednotlivé buňky. Výjimkou v tomto případě jsou úlohy typu doplňovačky, kdy je potřeba do buněk matice rovněž vepisovat písmena vytvářející jednotlivá slova. Stisknutí buněk je odchyťováno handlerem.

```
// zachycení výběru Area kliknutím
area.borderPtr.MouseLeftButtonDown += (sender, e) =>
{
    // volání metody pro obsluhu výběru Area
    ClickOnArea(area);
    e.Handled = true; // zachycení obslouženo
};

// zachycení výběru Area dotykem (prstem)
area.borderPtr.TouchDown += (sender, e) =>
{
    // volání metody pro obsluhu výběru Area
    ClickOnArea(area);
    e.Handled = true; // zachycení obslouženo
};
```

Při kliknutí do buňky matice se daná buňka označí a je zapojena do aktuálního kroku řešení. Při kliknutí do slova legendy se při úspěšně vykonaném kroku volá metoda **TaskOnNext()**. V ní se kontroluje, zda jsou již vyřešena všechna slova. Pokud ano, úloha je úspěšně dokončena.

```
// kontrola posledního kroku
// projití všech slov legendy
foreach (Area area in config.words)
{
    if (!area.ok)
    {
        // existuje nevyřešená položka v legendě
        // návratová hodnota (bude další krok úlohy)
        return TaskConstants.EnumTaskOnReturn.OK;
    }
}
PlaySound(SoundState.FINISH); // přehraj zvuk konec
// návratová hodnota (nebude další krok úlohy)
return TaskConstants.EnumTaskOnReturn.None;
```

4.1.3 Spojovačky a obrázky

V této skupině úloh je v metodě **TaskOnCreate()** přidán na hlavní grid úlohy canvas. Na canvas se rozmísťují jednotlivé oblasti, představující plochy dotyku, a jejich položky v metodě **TaskOnNew()** při načtení konfigurace aktuálního zadání úlohy.

V režimu konfigurace je umožněna manipulace s vytvořenými oblastmi představujícími body. Jednotlivé oblasti je možné přesouvat a měnit jejich rozměry. V režimu běhu jsou všechny události vykonávané pouze stisknutím na tyto jednotlivé buňky, což je odchyťováno handlerem. Dále se u úlohy typu kreslení rovněž vykreslují

čáry spojující jednotlivé body tvořící výsledný obrázek. Při změně velikosti celého okna je potřeba kromě jednotlivých bodů překreslovat i jejich spojnice.

```
// překreslení spojnic obrázku
int index = 1;
// projití všech prvních potomků Canvasu
foreach (UIElement el in canvas.Children)
{
    // objekt je typu Line (spojnice)
    if (el.GetType() == typeof(Line))
    {
        Line line = (Line) el;
        // úprava souřadnic bodů spojnice
        line.X1 = config.points[index - 1].PosX * scaleX;
        line.Y1 = config.points[index - 1].PosY * scaleY;
        line.X2 = config.points[index].PosX * scaleX;
        line.Y2 = config.points[index].PosY * scaleY;
        index++;
    }
}
```

Při označení oblasti bodu se daný bod aktivuje a je zapojen do aktuálního kroku řešení. Při označení bodu ve správném pořadí se volá metoda **TaskOnNext()**. Ta kontroluje, zda již byly označeny všechny body u posloupností nebo je-li označen poslední bod u kreslení. U prvního typu úlohy je potřeba projít všechny body, protože se hodnoty generují náhodně a není předem jisté, který bod je v posloupnosti poslední.

```
// kontrola posledního kroku u posloupností
// projití všech bodů posloupnosti
foreach (Area area in config.points)
{
    if (!area.selected)
    {
        // existuje označený bod v posloupnosti
        // návratová hodnota (bude další krok úlohy)
        return TaskConstants.EnumTaskOnReturn.OK;
    }
}
PlaySound(SoundState.FINISH); // přehraj zvuk konec
// návratová hodnota (nebude další krok úlohy)
return TaskConstants.EnumTaskOnReturn.None;
}

// kontrola posledního kroku u kreslení
// kontrola označení posledního bodu posloupnosti
if (config.points[config.points.Count - 1].selected)
{
    // byl označen poslední bod v posloupnosti
    PlaySound(SoundState.FINISH); // přehraj zvuk konec
    // návratová hodnota (nebude další krok úlohy)
    return TaskConstants.EnumTaskOnReturn.None;
}
// návratová hodnota (bude další krok úlohy)
return TaskConstants.EnumTaskOnReturn.OK;
```

4.1.4 Paměť

V této skupině úloh jsou v metodě **TaskOnCreate()** přidány na hlavní grid úlohy dva další gridy, které navíc překrývá canvas pro přichycení přesouvaných objektů. Jeden grid je matice pro symboly (horní), druhý grid umístěný dole pod maticí je řádek buněk pro symboly (dolní). Do jednotlivých buněk obou gridů se rozmisťují symboly v metodě **TaskOnNew()** při načtení konfigurace aktuálního zadání úlohy.

V režimu editace se nastavují rozměry obou matic a přidávají se do úlohy symboly jejich vložením do horní matice. U obou úloh je ze sady dostupných symbolů náhodně vybrán určitý počet (sloupečků dolní matice) symbolů. U úlohy s umístěním jsou náhodně vybrány buňky horní matice a zde se umístí vybrané symboly. U úlohy bez umístění se náhodně vybrané symboly umístí do buněk dolní matice. Po stisknutí tlačítka Řešit se symboly přesouvají do druhé spodní matice a je potřeba, jejich uchopením, je přetáhnout na původní místo v horní matici, nebo do libovolného rámečku v případě úlohy bez umístění.

V režimu běhu úlohy je umožněno přesouvat obrázky mezi jednotlivými oblastmi. To je zajištěno odejmutím obrázku symbolu z buňky v gridu (mřížka) a jeho dočasným umístěním na canvas (kreslicí plátno). Později po upuštění symbolu je obrázek symbolu přesunut z canvasu do cílové nebo původní buňky v gridu (pokud byl nesprávně umístěn). Jediné omezení posunu zde je zakázání posunování obrázků v režimu běhu, kdy už je obrázek jednou umístěn do své cílové oblasti. Přesouvání všech obrázků je dáno nastavením handlerů pro události manipulující s jednotlivými objekty typu **Item**. Odchycené události jsou uchopení obrázku, posun obrázku po canvasu a uvolnění obrázku.

```
// umístění obrázku z Grid na Canvas
// buňka obsahuje daný image
if (area.viewPtr.Child == image)
{
    // vymazání odkazů na image v Area
    area.viewPtr.Child = null;
    area.item = null;
}
// přidání komponenty na Canvas
Canvas.Children.Add(image);
```

Při uvolnění obrázku symbolu nad buňkou gridu se vyhodnotí, je-li symbol správně umístěn. Pokud ano, volá se metoda **TaskOnNext()**. V ní se kontroluje u typu úlohy s umístěním, obsahuje-li spodní grid ještě nějaké zbylé obrázky symbolů. U úlohy typu bez umístění se kontroluje, je-li spodní grid maximálně zaplněn. Jsou-li tyto podmínky splněny, je úloha úspěšně dokončena.

```

// kontrola posledního kroku u varianty s umístěním
// projití všech buněk spodního Grid
foreach (Area area in config.starters)
{
    if (area.viewPtr.Child != null)
    {
        // existuje neumístěný předmět
        // návratová hodnota (bude další krok úlohy)
        return TaskConstants.EnumTaskOnReturn.OK;
    }
}
PlaySound(SoundState.FINISH); // přehraj zvuk konec
// návratová hodnota (nebude další krok úlohy)
return TaskConstants.EnumTaskOnReturn.None;

// kontrola posledního kroku u varianty bez umístění
int count = 0; // počet vyplněných buněk ve spodním Grid
// projití všech buněk spodního Grid
foreach (Area area in config.starters)
{
    if (area.viewPtr.Child != null)
    {
        // zvýšení počtu vyplněných buněk
        count++;
    }
}
// spodní Grid neobsahuje maximální možný počet symbolů
if (count < Math.Min(config.items.count, config.starts))
{
    // existuje neumístěný předmět
    // návratová hodnota (bude další krok úlohy)
    return TaskConstants.EnumTaskOnReturn.OK;
}
PlaySound(SoundState.FINISH); // přehraj zvuk konec
// návratová hodnota (nebude další krok úlohy)
return TaskConstants.EnumTaskOnReturn.None;

```

4.1.5 Odpovědi a dvojice

V této skupině úloh jsou v metodě **TaskOnCreate()** přidány na hlavní grid úlohy canvas a grid na pravý okraj plochy. V případě úlohy dvojice je přidán do úlohy ještě jeden grid na levý okraj plochy. V jednotlivých buňkách gridů jsou zobrazena slova jako možnosti. Ta se načítají při volání metody **TaskOnNew()** při načtení konfigurace aktuálního zadání úlohy. Dalšími parametry načtenými při zobrazení úlohy jsou textové zadání umístěné na canvasu a případně obrázek na pozadí úlohy, taktéž umístěný na canvasu.

V režimu konfigurace se zobrazují jednotlivé možnosti v seřazeném tvaru, tzv. tak, jak byly do úlohy vloženy. U úlohy typu dvojice, možnosti na stejném řádku tvoří dvojici. V režimu běhu jsou možnosti v obou gridech náhodně proházeny. V režimu běhu jsou všechny události prováděné pouze stisknutím jednotlivých buněk, což je odchyťováno handlerem.

Při označení možnosti (nebo dvojice možností) se vyhodnotí správnost označení. Pokud bylo správně označeno, volá se metoda `TaskOnNext()`. V ní se kontroluje u typu úlohy dvojice, jestli byly již přiřazeny všechny dvojice. U úlohy typu možnosti se kontroluje správnost vybrané odpovědi. Jsou-li tyto podmínky splněny, je úloha úspěšně dokončena.

```
// kontrola posledního kroku u dvojic
// projití všech možností
foreach (Answer answer in config.answers)
{
    if (!answer.items[0].target.ok)
    {
        // existuje nepřirazení dvojice
        // návratová hodnota (bude další krok úlohy)
        return TaskConstants.EnumTaskOnReturn.OK;
    }
}
PlaySound(SoundState.FINISH); // přehraj zvuk konec
// návratová hodnota (nebude další krok úlohy)
return TaskConstants.EnumTaskOnReturn.None;

// kontrola posledního kroku u jedné odpovědi
// projití všech možností
foreach (Answer answer in config.answers)
{
    // byla označena správná odpověď
    if (answer.items[0].ok && answer.items[0].target.selected)
    {
        PlaySound(SoundState.FINISH); // přehraj zvuk konec
        // návratová hodnota (nebude další krok úlohy)
        return TaskConstants.EnumTaskOnReturn.None;
    }
}
// návratová hodnota (bude další krok úlohy)
return TaskConstants.EnumTaskOnReturn.OK;
```

4.2 Ukládání dat úloh

V této části bude popsána implementace konfiguračních tříd, které definují podobu zadání úlohy.

4.2.1 Posloupnosti akcí

Třída `SequencesConfig.cs` obsahuje definice jednotlivých oblastí. Odkaz `AreaStart` je odkaz na objekt startovní oblasti, seznam `AreaTargets` uchovává zbylé cílové oblasti, na které se umisťují obrázky předmětů v průběhu hry.

```

// třída pro konfiguraci úloh posloupnosti akcí
public class SequencesConfig
{
    // startovní oblast
    [XmlElement("AreaStart")] public Area AreaStart;
    // cílové oblasti
    [XmlArray("AreaTargets"),
    XmlArrayItem("AreaTarget", typeof(Area))]
    public List<Area> AreaTargets;
}

```

Třída **Area** definuje vlastnosti jednotlivých oblastí zobrazených v úloze. Atributy **PosX** a **PosY** uchovávají relativní pozice oblasti na canvasu a atributy **SizeX** a **SizeY** uchovávají relativní velikosti oblasti na canvasu. Relativní pozice a rozměr je udán jako poměr pozice pixelu (u pozice) nebo vzdálenosti dvou pixelů (u rozměru) vůči rozměrům obrázku na pozadí. Atribut **priority** je určen pouze pro úlohy, kde zcela záleží na pořadí a udává hodnotu pořadí vyplnění dané oblasti. Do konfiguračního souboru neukládán atributy **borderPtr**, **stackPnlPtr** a **update** jsou využívány pouze v rámci úlohy pro uchování odkazu na orámování oblasti, vnitřní panel oblasti a příznak úpravy oblasti v editačním módu. Seznam **items** obsahuje seznam položek, které při řešení úlohy patří do dané oblasti.

```

// třída definující oblast
public class Area
{
    // relativní pozice na Canvas souřadnice
    [XmlAttribute] public int PosX;
    [XmlAttribute] public int PosY;
    // relativní velikost na Canvas
    [XmlAttribute] public int SizeX;
    [XmlAttribute] public int SizeY;
    // pořadí vyplnění oblasti
    [XmlAttribute] public int priority;
    // orámování oblasti
    [XmlIgnore] public Border borderPtr;
    // vnitřní panel oblasti
    [XmlIgnore] public StackPanel stackPnlPtr;
    // aktualizace oblasti
    [XmlIgnore] public bool update;
    // seznam položek patřících do oblasti
    [XmlArray("Items"),
    XmlArrayItem("Item", typeof(Item))]
    public List<Item> items;
}

```

Třída **Item** reprezentuje další dodatečné informace objektu **Image**, který se umísťuje na **StackPanel** ve své rodičovské oblasti. Atribut **priority** je určen pouze pro úlohy typu, kdy částečně záleží na pořadí a udává hodnotu pořadí obrázku v oblasti. Atribut **fileName** obsahuje název souboru s obrázkem uloženým v úložišti. Do konfiguračního souboru neukládán atributy **extension**, **area**, **sound**, **update**

a `inAreaTarget` jsou využívány pouze v rámci úlohy pro uchování koncovky souboru s obrázkem, odkaz na rodičovskou oblast, kam obrázek patří a příznaky zvukové nápovědy obrázku, úpravy obrázku v editačním módu a správném umístění obrázku v režimu běhu.

```
// třída rozšiřující Image
public class Item
{
    // pořadí obrázku v oblasti
    [XmlAttribute] public int priority;
    // název souboru s obrázkem
    [XmlText] public string fileName;
    // koncovka souboru s obrázkem
    [XmlIgnore] public string extension;
    // zvuk k předmětu
    [XmlIgnore] public bool sound;
    // rodičovská oblast
    [XmlIgnore] public Area area;
    // aktualizace objektu
    [XmlIgnore] public bool update;
    // umístěn do své oblasti
    [XmlIgnore] public bool inAreaTarget;
}
```

4.2.2 Křížovky a osmisměrky

Třída `TaskCrosswordsVariant.cs` obsahuje definice jednotlivých položek zobrazených v maticích a rozměry matic. Dvourozměrné pole `items` uchovává položky zobrazené v hlavní matici a jednorozměrné pole `itemsWords` uchovává položky zobrazené v pravé matici (legenda). Atributy `rows` a `columns` uchovávají počet řádků a sloupců matice, atribut `right` uchovává počet slov legendy. Atribut `column` je určen pouze pro úlohy typu doplňovačky a uchovává index sloupce s tajenkou. Neukládané atributy `content` a `words` uchovávají odkazy na oblasti umístěné v buňkách matic.

```
// třída pro konfiguraci úloh křížovky a rébusy
public class TaskCrosswordsVariant
{
    // položky hlavní matice
    [XmlAttribute] public int rows;
    [XmlAttribute] public int columns;
    [XmlAttribute] public bool right;
    [XmlAttribute] public string content;
    [XmlAttribute] public string words;
    [XmlAttribute] public string solution;
    [XmlAttribute] public int priority;
    [XmlAttribute] public string fileName;
    [XmlAttribute] public string extension;
    [XmlAttribute] public bool sound;
    [XmlAttribute] public Area area;
    [XmlAttribute] public bool update;
    [XmlAttribute] public bool inAreaTarget;
}
```



```

// počet slov legendy
[XmlAttribute] public int right;
// index sloupce s tajenkou
[XmlAttribute] public int column;
// oblasti matice
[XmlIgnore] public List<List<Area>> content;
// oblasti legendy
[XmlIgnore] public List<Area> words;
}

```

Třída **Area** definuje vlastnosti jednotlivých oblastí, umístěných v buňkách matic. Atributy **PosX** a **PosY** uchovávají indexy pro sloupec a řádek buňky v matici. Atributy **borderPtr** a **viewPtr** uchovávají odkazy na orámování buňky a vnitřní obsah buňky. Atribut **item** obsahuje odkaz na položku umístěnou v buňce. Příznaky **right**, **ok** a **selected** jsou určeny pro umístění buňky v seznamu slov, identifikaci vyřešeného slova v buňce z pravé matice a označení buňky s písmenem v režimu hry.

```

// třída definující oblast
public class Area
{
    // pozice v matici
    public int PosX;
    public int PosY;
    // orámování oblasti
    public Border borderPtr;
    // vnitřek oblasti
    public Viewbox viewPtr;
    // obsah buňky
    public Item item;
    // legenda
    public bool right;
    // vyřešené slovo
    public bool ok;
    // označení oblasti
    public bool selected;
}

```

Třída **Item** definuje obsah jedné buňky matice. Její atribut **text** udržuje textový obsah buňky. Neukládaný atribut **target** udržuje odkaz na buňku v matici a neukládaný příznak **isPlaced** je určen pro úlohu vyhledávání v textu, kde určuje, je-li písmeno v buňce obsaženo v některém z ukrytých slov.

```

// obsah jedné buňky matice
public class Item
{
    // textový obsah
    [XmlText] public String text;
    // buňka matice
    [XmlIgnore] public Area target;
    // součást slova
    [XmlIgnore] public bool isPlaced;
}

```

4.2.3 Spojovačky a obrázky

Třída `OrderingConfig.cs` obsahuje definice jednotlivých bodů na canvasu určených k označení. Seznam `points` uchovává definice jednotlivých zobrazených bodů. Atribut `format` je určený pro úlohu typu posloupnosti, kde určuje počet cifer generovaných čísel.

```
// třída pro konfiguraci úloh spojovačky
public class OrderingConfig
{
    // seznam bodů
    [XmlAttribute("Points"), XmlArrayItem("Point", typeof(Area))]
    public List<Area> points;
    // počet cifer
    [XmlAttribute] public int format;
}
```

Třída `Area` definuje vlastnosti jednotlivých bodů, zobrazených v úloze. Atributy `PosX` a `PosY` uchovávají relativní pozice bodu na canvasu a atributy `SizeX` a `SizeY` uchovávají relativní velikosti bodu na canvasu. Neukládaný atribut `priority` je určen k uchování pořadí daného bodu v posloupnosti. Další do konfiguračního souboru neukládané atributy `borderPtr`, `stackpanelPtr`, `update` a `selected` jsou využívány pouze v rámci úlohy pro uchování odkazu na orámování bodu, vnitřní panel bodu, příznak úpravy bodu v editačním módu a správně označeného bodu v režimu běhu.

```
// třída definující oblast
public class Area
{
    // relativní pozice na Canvas souřadnice
    [XmlAttribute] public int PosX;
    [XmlAttribute] public int PosY;
    // relativní velikost na Canvas
    [XmlAttribute] public int SizeX;
    [XmlAttribute] public int SizeY;
    // pořadí bodu v posloupnosti
    [XmlAttribute] public int priority;
    // orámování oblasti
    [XmlAttribute] public Border borderPtr;
    // vnitřní panel oblasti
    [XmlAttribute] public StackPanel stackPnlPtr;
    // aktualizace oblasti
    [XmlAttribute] public bool update;
    // již označený bod
    [XmlAttribute] public bool selected;
}
```

4.2.4 Paměť

Třída `MemoryConfig.cs` obsahuje definice jednotlivých symbolů a rozměry matic. Seznam `items` uchovává symboly pro aktuální zadání. Atributy `rows`

a **columns** uchovávají počet řádků a sloupců matice, atribut **starts** uchovává počet symbolů ve spodní matici. Neukládané atributy **areas** a **starters** uchovávají odkazy na oblasti umístěné v buňkách matic.

```
// třída pro konfiguraci úloh pamět'
public class TaskCrosswordsVariant
{
    // symboly zadání
    [XmlArray("Items"),
    XmlArrayItem("Item", typeof(List<Item>))]
    public List<List<Item>> items;
    // textové zadání úlohy
    [XmlElement("Solution")] public string solution;
    // rozměry matice
    [XmlAttribute] public int rows;
    [XmlAttribute] public int columns;
    // počet sloupců dolního Grid
    [XmlAttribute] public int starts;
    // oblasti matice
    [XmlIgnore] public List<List<Area>> areas;
    // oblasti dolního Grid
    [XmlIgnore] public List<Area> starters;
}
```

Třída **Area** definuje vlastnosti jednotlivých oblastí, umístěných v buňkách matic. Atributy **PosX** a **PosY** uchovávají indexy pro sloupec a řádek buňky v matici. Atributy **borderPtr** a **viewPtr** uchovávají odkazy na orámování buňky a vnitřní obsah buňky. Atribut **item** obsahuje odkaz na položku umístěnou v buňce a atribut **solve** využívá pouze typ úlohy s umístěním, kde se zde ukládá položka, která má být do této oblasti umístěna. Příznak **start** je určen pro umístění buňky v dolní matici.

```
// třída definující oblast
public class Area
{
    // pozice v matici
    public int PosX;
    public int PosY;
    // orámování oblasti
    public Border borderPtr;
    // vnitřek oblasti
    public Viewbox viewPtr;
    // obsah buňky
    public Item item;
    // správný obsah buňky
    public Item solve;
    // dolní matice
    public bool start;
}
```

Třída **Item** definuje obsah buňky matice. Její atribut **fileName** udržuje název souboru s obrázkem symbolu. Neukládané atributy **extension** a **target** udržují koncovku souboru s obrázkem symbolu a odkaz na buňku v matici. Neukládané

příznaky **update** a **inTarget** uchovávají hodnoty pro pohybování s obrázkem a správné umístění obrázku v režimu běhu.

```
// třída rozšiřující Image
public class Item
{
    // název souboru s obrázkem
    [XmlText] public string fileName;
    // koncovka souboru s obrázkem
    [XmlIgnore] public string extension;
    // rodičovská oblast
    [XmlIgnore] public Area target;
    // aktualizace objektu
    [XmlIgnore] public bool update;
    // umístěn do své oblasti
    [XmlIgnore] public bool inTarget;
}
```

4.2.5 Odpovědi a dvojice

Třída **QuestionsConfig.cs** popisuje možnosti a zadání úlohy. Seznam **answers** uchovává jednotlivé možnosti pro výběr. Atribut **task** uchovává textové zadání úlohy. Neukládané atributy **left** a **right** uchovávají odkazy na oblasti umístěné v buňkách matic.

```
// třída pro konfiguraci úloh texty a slova
public class QuestionsConfig
{
    // seznam možností
    [XmlArray("Answers"),
    XmlArrayItem("Answer", typeof(Answer))]
    public List<Answer> answers;
    // oblasti s možnostmi
    [XmlIgnore] public List<Area> left;
    [XmlIgnore] public List<Area> right;
    // textové zadání úlohy
    [XmlAttribute] public String task;
}
```

Třída **Answer** definuje jednu položku odpovědi nebo dvojici možností. Její jediný atribut **items** uchovává seznam položek, které jsou součástí odpovědi. V případě typu úlohy jedna odpověď obsahuje seznam jednu položku, v případě možností obsahuje dvě položky, které patří k sobě.

```
// třída definující odpověď
public class Answer
{
    // seznam položek odpovědi (možnosti)
    [XmlArray("Pairs"), XmlArrayItem("Pair", typeof(Item))]
    public List<Item> items;
}
```

Třída **Item** definuje obsah buňky matice. Její atribut **text** udržuje textový obsah buňky. Příznak **ok** je určen pouze pro typ úlohy jedna odpověď a určuje správnost možnosti. Neukládané atributy **target** a **order** udržují odkaz na buňku v matici a index položky v seznamu možností.

```
// obsah jedné buňky matice
public class Item
{
    // textový obsah
    [XmlText] public String text;
    // správná odpověď
    [XmlAttribute] public bool ok;
    // buňka matice
    [XmlIgnore] public Area target;
    // index položky
    [XmlIgnore] public int order;
}
```

Třída **Area** definuje vlastnosti jednotlivých oblastí, umístěných v buňkách matic. Příznak **left** a atribut **PosY** určují umístění oblasti do levé nebo pravé matice a index řádku buňky v matici. Atributy **borderPtr** a **viewPtr** uchovávají odkazy na orámování buňky a vnitřní obsah buňky. Atribut **item** obsahuje odkaz na položku umístěnou v buňce. Příznaky **selected** a **ok** jsou určeny pro aktuální označení oblasti s možností a správně označené možnosti nebo dvojice.

```
// třída definující oblast
public class Area
{
    // pozice v matici
    public int PosY;
    // orámování oblasti
    public Border borderPtr;
    // vnitřek oblasti
    public Viewbox viewPtr;
    // obsah buňky
    public Item item;
    // levý Grid
    public bool left;
    // správně označené
    public bool ok;
    // aktuální označení oblasti
    public bool selected;
}
```

5 Prvotní testování

Úlohy byly předběžně otestovány na několika (zhruba pěti) uživateli ve věku od 5 do 80 let. Tak velký věkový rozptyl byl zvolen z důvodu pozdějšího využití úloh (od dětí z Jedličkova ústavu až po pečovatelské domy pro starší občany). Postřehy od uživatelů lze rozdělit do dvou základních skupin.

První skupinou jsou připomínky k použitému frameworku. Do této skupiny patří názory:

- Poněkud malá tlačítka v horní části pro ovládání aplikace.
- Ovládací tlačítka na tabletu o úhlopříčce 10 palců zabírají (i přes svou malou velikost) zbytečně mnoho z pracovní plochy.

Jedná se o poznatky adresované spíše na vedoucího práce (tvůrce použitého frameworku) nikoli na tvůrce úloh a proto zde nebudou více rozebírány. Druhou skupinou jsou připomínky týkající se samotných úloh, jejich nejen grafického řešení, ale i ovládání. Jednotlivé poznatky a jejich rozbor bude popsán v následujících podkapitolách. Některé dále uvedené připomínky uživatelů mohou vycházet pouze z omezeného počtu demonstračních zadání (pro účely této práce) a proto jsou v podstatě pouze zmíněny a tedy nejsou ve skutečnosti zakomponovány.

5.1 Posloupnosti akcí

První připomínkou je nelibost/zbytečnost v zaznění zvukové interakce i v případě správně vykonaného tahu. V případě správně i nesprávně vykonaného tahu (pokud má předmět přiřazen vlastní zvuk), zazní stejný zvukový záznam se správným umístěním souboru. To může být občas pro uživatele matoucí a proto se při správně provedeném tahu zvuk nebude přehrávat, proběhne pouze grafická interakce v podobě zezelenání pozadí oblasti.

Další připomínkou je spíš návrh, kdy po vložení předmětu do oblasti, se všechny předměty v oblasti nezobrazí vedle sebe ale přes sebe (obrázky předmětů se tedy překrývají). Toto by bylo možná vhodné pro úlohy typu, kdy částečně záleží na pořadí. Jednotlivé položky by se poté překrývaly (například části oblečení). Nevýhodou pak bude, že si řešitel po poskládání více položek do jedné oblasti nemusí uvědomit plnou posloupnost jednotlivých položek. Tato připomínka nebude tudíž do této sady úloh zakomponována.

Poslední připomínkou, z důvodu zapomnětlivosti některých řešitelů, je vhodnost neustálého zobrazení zadání úlohy v textové formě v její horní části. Tato připomínka je zřejmě velmi důležitá a tudíž byla zakomponována do všech skupin úloh. Aby byla tato funkcionálníta zakomponována jednotně do všech úloh (někde se nehodí přímé zobrazení

textu se zadáním), bylo na ovládací panel přidáno tlačítko „Přehrát“, po jehož stisknutí se znovu přehraje zvukové zadání aktuální úlohy.

5.2 Křížovky a osmisměrky

První připomínka se vztahuje k legendě úloh. Při hodně velké množině slov (na pravé straně) se slova zobrazují velmi malá a proto se těžko vybírají (hlavně dotykem prstu). V rámci tvorby úloh pomocí editačního módu je počet slov omezen na 20. V případě manuální tvorby konfiguračního souboru však tento počet není nijak omezen. Je na uvážení autora úloh, aby vybral vhodný počet slov jednotlivých úloh z hlediska této připomínky.

Další připomínkou, nebo spíše návrhem, je u osmisměrek po ukončení úlohy znatelně barevně označit výslednou tajenku. Tato připomínka je určitě vhodná a byla tedy zakomponována do úlohy osmisměrky.

Poslední připomínka se vztahuje k výběru hledaného v osmisměrce z legendy. Písmena v matici se musí označit jednotlivě jedno po druhém a uživatelsky přívětivější by bylo například po kliknutí na první písmeno vést čáru k poslednímu písmenu. Toto by bylo určitě vhodné zakomponovat do řešení sady úloh, ale z hlediska časové náročnosti doplnění této funkcionality tak nebude nyní uskutečněno.

5.3 Spojovačky a obrázky

První připomínkou k vylepšení úlohy posloupnosti je návrh negenerovat náhodně pouze čísla v oblastech, ale i oblasti náhodně rozmísťovat po ploše. Toto řešení by přispělo k větší originalitě zadání, ale tím, že se čísla generují náhodně z celkem obsáhlého intervalu hodnot, je velmi nepravděpodobné, že by si uživatel jejich řešení zautomatizoval, proto nebude tato připomínka v současné verzi zakomponována.

Dalším návrhem je podle obtížnosti úlohy omezit počet nutných bodů pro spojení pro vytvoření obrázku (třeba každý druhý nebo třetí). Z hlediska úlohy posloupnosti je toto nezařaditelné. Hledat každý x-tý bod v posloupnosti je pro uživatele těžší, než najít ten aktuální. Z hlediska kreslení se jedná o zjednodušení, ale úloha by částečně ztratila svůj smysl, neboť by dokreslený obrázek nemusel vůbec odpovídat zamýšlené skutečnosti. Tato připomínka nebude zakomponována.

Poslední připomínka je podle obtížnosti úlohy nastavit současně velikost všech bodů na obrázku. Při větším počtu bodů je určitě vhodné nějak měnit velikosti všech bodů najednou a ne jeden po druhém, proto byla v editačním módu přidána možnost změny velikosti všech bodů současně pro každé zadání úlohy.

5.4 Paměť

První připomínka je místo klikacího tlačítka ke spuštění úlohy spustit úlohu automaticky po uplynutí nějakého časového limitu. Určitě je vhodné nastavit časový limit pro zapamatování, proto byla v editačním módu přidána možnost nastavení limitu pro každé zadání úlohy.

Dalším návrhem je náhodně měnit barvy použitých symbolů a zvýšit tak jedinečnost jednotlivých zadání. Vzhledem k tomu, že lze vytvořit v podstatě nespočetně sad symbolů, ze kterých se generují zadání těchto úloh a když se dále uváží, že ne každý obrázek (již vytvořený) lze jednoduše modifikovat, tak tento návrh nebude zakomponován.

Poslední připomínka požaduje možnost několika úrovní úlohy s různými počty symbolů. Tohoto je možné dosáhnout vytvořením více zadání se stejnou sadou symbolů a nastavením jiného počtu sloupců ve spodní matici, tudíž tato připomínka nebude nijak jinak samostatně zakomponována.

5.5 Odpovědi a dvojice

První připomínka se vztahuje k sloupcům se slovy. Při poněkud velké množině slov se slova zobrazují velmi malá a proto se těžko vybírají (zejména dotykem prstu). V rámci tvorby úloh pomocí editačního módu je počet slov omezen na 10. V případě manuální tvorby konfiguračního souboru tento počet není nijak omezen. Je na uvážení autora úloh, aby vybral vhodný počet slov jednotlivých úloh z hlediska této připomínky.

Další připomínkou u úlohy dvojice je znázornění již přiřazených dvojic, aby řešitel věděl i zpětně, co k čemu patří. Tato připomínka je určitě vhodná a byla zakomponována do úlohy dvojice (u již označených dvojic se, v současné verzi, zobrazí v závorce číslo položky).

Posledním návrhem je opačný způsob této sady úloh (tzv. zadání je text a odpovědi jsou obrázky). K tomu by bylo zapotřebí vytvořit úplně nový typ úlohy, což nyní nebude zakomponováno ve výsledném řešení.

6 Popis editace úloh

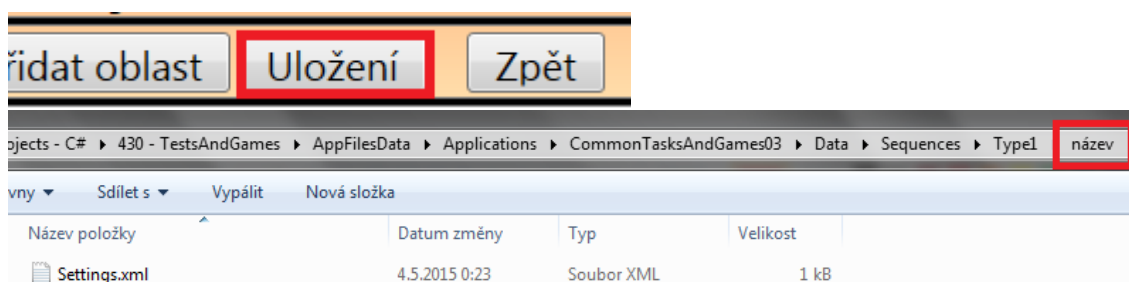
Obecná funkcionálnita editace všech úloh zahrnuje přidávání a odebrání jednotlivých zadání. Oba tyto úkony lze vykonat dvěma způsoby.

Prvním způsobem je v rámci jednoduchého editačního módu úloh. Pro výběr přidání zadání je třeba na ovládacím panelu stisknout tlačítko „Přidat zadání“. Po stisku tohoto tlačítka se zobrazí dialog, do kterého je třeba zadat název přidávaného zadání. Není možné přidat zadání s názvem, který už existuje pro danou úlohu. V případě nekorektního zadání názvu nebude zadání vytvořeno a uživatel o tom bude vyrozuměn v informačním dialogu.



Obrázek 5 - Přidání nového zadání

Po úspěšném vytvoření zadání je vytvořen adresář pro toto nové zadání a úlohu je po opětovném přepnutí do konfiguračního módu možné začít skutečně vytvářet. Průběh vytváření jednotlivých úloh pomocí programu je popsán v jednotlivých podkapitolách. Po dokončení vytváření úlohy je potřeba zmáčknout tlačítko „Uložit“, které uloží data nově vytvořené úlohy do adresáře s vytvořeným zadáním. Každý adresář obsahuje konfigurační XML soubor s daty definujícími vzhled a obsah úlohy. Případné další soubory jsou uloženy také v adresáři se zadáním.



Obrázek 6 - Uložení zadání

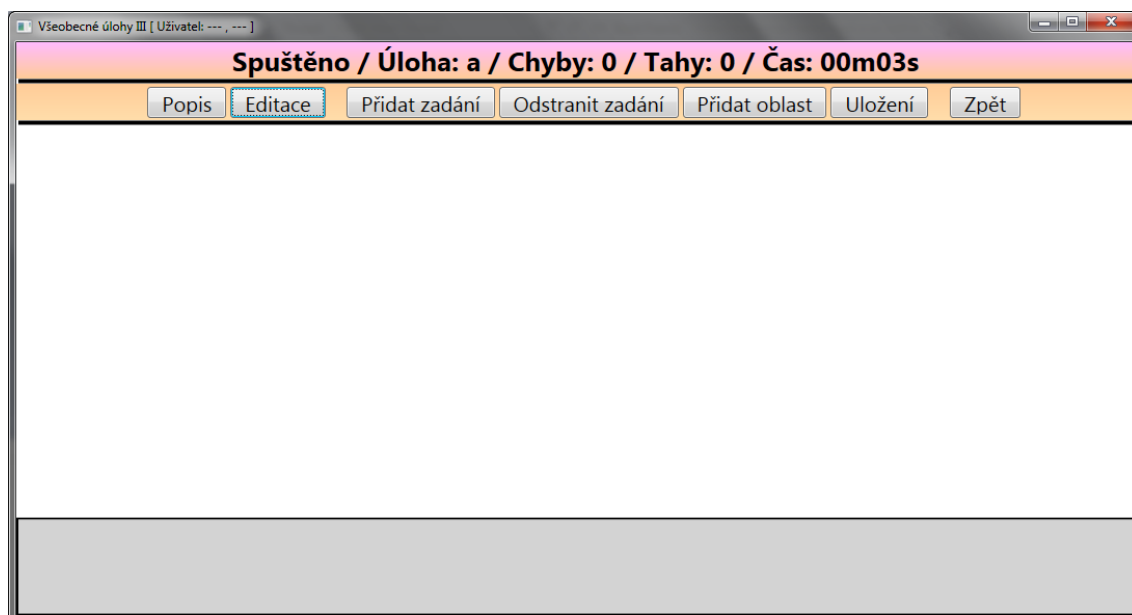
Pokud chceme odebrat některé zadání z úlohy je potřeba při jejím spuštění se přepnout do jejího konfiguračního módu a zde stisknout tlačítko „Odebrat zadání“. Po potvrzení odebrání je aktuální zadání odebráno a dále se vygeneruje (tedy zobrazí) zadání nové (existuje-li nějaké).

Druhým způsobem je vytvoření zadání manuálně. U manuálního vytváření zadání je nejprve potřeba v úložišti vytvořit adresář pro zadání úlohy. Do adresáře je

poté potřeba přidat konfigurační soubor, jehož obsah se vytvoří taktéž manuálně. Další potřebná data, která úloha využívá je potřeba do adresáře rovněž nakopírovat. Manuální odstranění zadání se uskuteční pouhým smazáním celého adresáře se zadáním z úložiště.

6.1 Posloupnosti akcí

Na začátku úloha obsahuje startovní prázdnou oblast a bílé pozadí. Prvním krokem v tvorbě všech tří typů úloh v této skupině je vložení obrázku na pozadí a vložení obrázků představujících předměty do úlohy. Vložení obrázku na pozadí se uskuteční uchopením a přetažením obrázku do aplikace (například z běžného průzkumníka Windows). Pokud v adresáři, odkud se přetahuje obrázek zadání, existuje zvukový soubor se stejným názvem (jako obrázek se zadáním), bude zvukový záznam použit jako zvukové zadání úlohy.

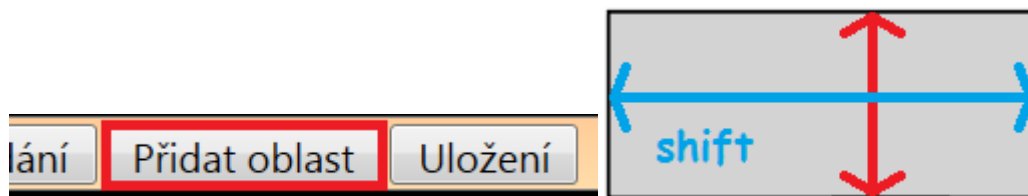


Obrázek 7 - Prázdná úloha

Vložení obrázků představujících předměty do úlohy se vykoná uchopením a přetažením obrázků do startovního dolního rámečku (na obrázku zobrazen šedou barvou). Pokud v adresáři, odkud se přetahuje obrázek předmětu, existuje zvukový soubor se stejným názvem (jako přetahovaný obrázek), bude zvukový záznam použit jako zvuková nápověda při umístění tohoto předmětu.

Dalším krokem tvorby zadání úlohy je vytvoření jednotlivých oblastí, kam se budou předměty při řešení úlohy rozmisťovat. Vytvoření oblasti se uskuteční stisknutím tlačítka „Přidat oblast“. Po přidání oblasti se musí oblast umístit na své místo (uchopením a přetažením myši) a nastavit její požadovanou velikost. Změna velikosti oblasti se uskuteční pomocí kolečka myši. Upravovanou oblast je potřeba vybrat stisknutím levého tlačítka myši a za jeho současného držení lze měnit její velikost

pomocí rolování kolečka. Při současném stisknutí klávesy shift se mění horizontální rozměr oblasti. Bez držení klávesy shift se mění vertikální rozměr oblasti. Pro odebrání oblasti z canvasu je potřeba na oblast kliknout a současně držet klávesu Ctrl.



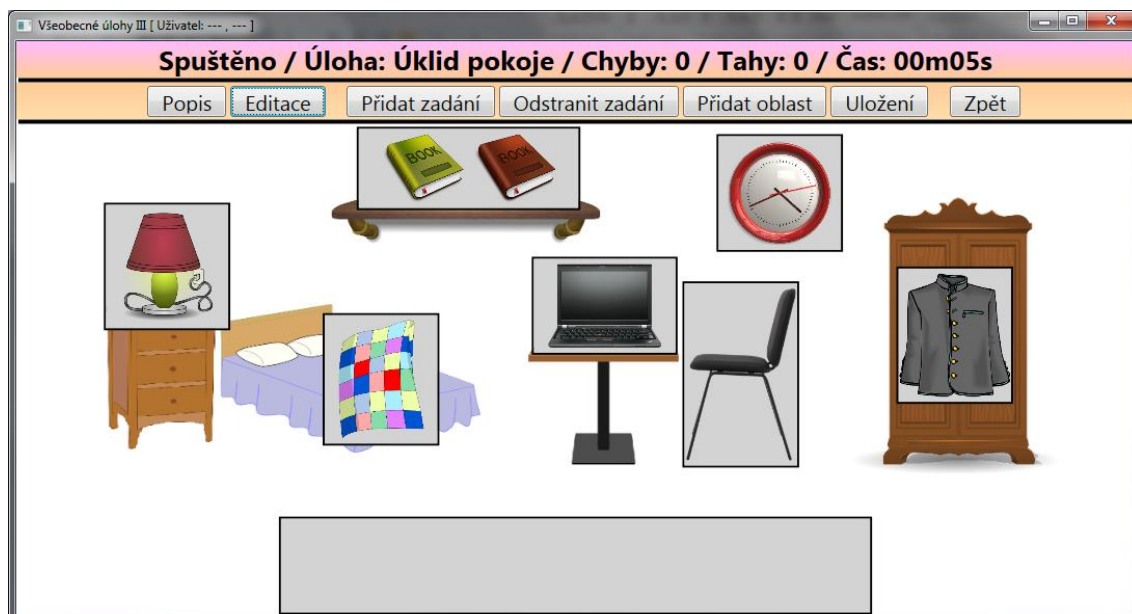
Obrázek 8 - Přidání oblasti a její úpravy

U prvních dvou typů úloh stačí přidat potřebný počet oblastí a ty rozmístit po obrázku zadání (pozadí). U posledního typu, kdy záleží kompletně na posloupnosti umístěných předmětů, se musí dát pozor na pořadí přidávání oblastí do úlohy. Oblast, která bude přidána na plochu jako první, bude při řešení potřeba vyplnit předměty také jako první.

V případě editace existující úlohy třetího typu bude přidaná oblast umístěna v pořadí až na konci, je-li třeba ji umístit někam doprostřed pořadí, musí se postupně posunovat předměty v oblastech vždy o jednu dál. Například existuje úloha s procesem nakupování. Nejprve je potřeba vzít si košík, poté vložit položky nákupu do košíku a nakonec zaplatit. Nyní chceme tuto úlohu rozšířit, a to tak, že budeme chtít, aby se nejprve vložilo do košíku pečivo a poté mléčné výrobky. K tomu je tedy potřeba přidat do úlohy čtvrtou oblast, která ale bude v pořadí vyplňování oblastí na posledním místě. Musí se do ní tedy přemístit proces placení a oblast, kde bylo původně placení bude nyní obsahovat položky mléčných výrobků.

Posledním krokem je rozmístění obrázků předmětů do připravených oblastí, kam patří. U prvního a posledního typu úlohy nezáleží na pořadí rozmístění předmětů do těchto oblastí. U druhého typu úlohy, kdy částečně záleží na pořadí, se musí dát pozor na pořadí vložení obrázku do jednotlivých oblastí. Pořadí vkládání předmětů do oblastí odpovídá pozdějšímu postupu řešení této úlohy uživatelem. První předmět v seznamu musí být umístěn do oblasti samozřejmě jako první.

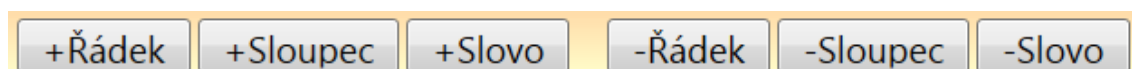
V případě editace existující úlohy druhého typu je třeba si dát pozor na pořadí položek v jednotlivých oblastech – jsou zde rozmístěny náhodně, jako při spuštění ve startovní oblasti. Přeskupení se vykoná vyzvednutím předmětu a jeho opětovným upuštěním nad tuto oblast. Upuštěný předmět bude nyní až úplně na konci seznamu všech předmětů v dané oblasti.



Obrázek 9 - Kompletní úloha ze skupiny Posloupnosti akcí

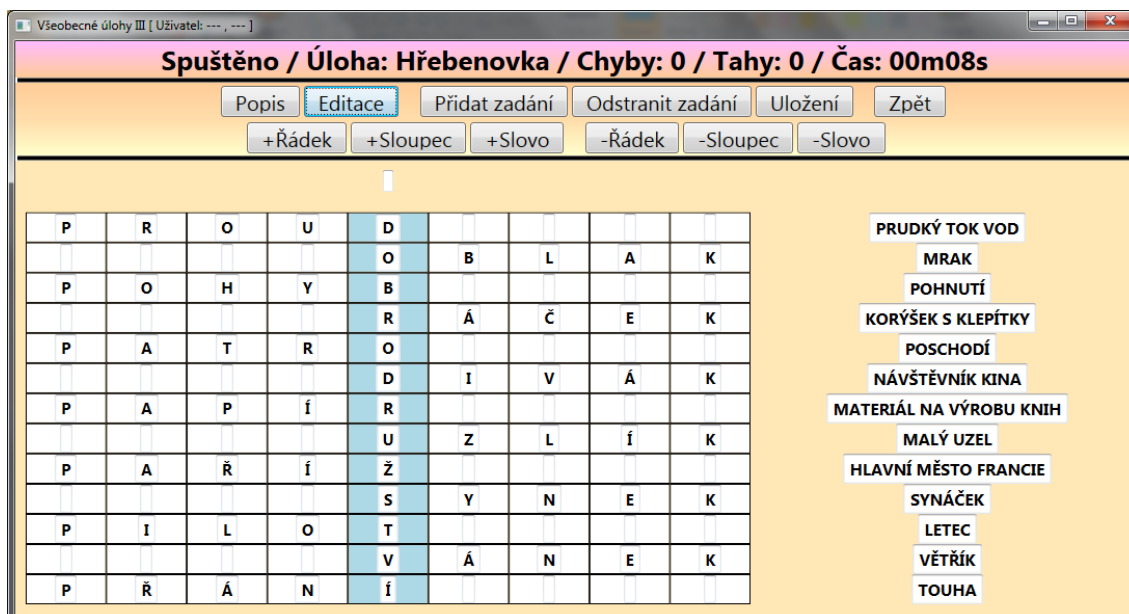
6.2 Křížovky a osmisměrky

Na začátku úloha obsahuje dvě prázdné matice. Všechny typy úloh mají stejnou základní funkcionalitu na úpravu velikosti matice a počtu slov. Pomocí tlačítek „+Sloupec“ a „+Řádek“ se rozměry matice zvětšují, pomocí tlačítek „-Sloupec“ a „-Řádek“ se naopak zmenšují. Minimální rozměr matice je 1x1, maximální pak 20x20. Pomocí tlačítek „+Slovo“ a „-Slovo“ se mění počet slov v pravém seznamu. Minimální počet je 1 slovo a maximální pak 20 slov. Dále je možné u všech variant úloh nastavit textový doprovod v podobě pomocné věty k tajence.



Obrázek 10 - Nastavení velikosti matic

Velikost matice pro křížovky se nastaví tak, aby se do ní daly vyplnit všechny požadované odpovědi. Současně s přidáváním řádků matice se přidávají pole ve sloupci nápověd na pravé straně (položky legendy). Po roztažení matice je umožněno vkládat písmena do všech políček, avšak políčka, která nejsou vyplněna písmeny, nebudou při skutečném spuštění řešení úlohy vykreslena. Z každého políčka se uloží pouze jedno písmeno, není možné jich zde uložit více. Do pravé matice se napíše jednotlivá slova legendy k vyplnění do křížovky. Pro vybrání sloupce ve kterém se nachází tajenka stačí kliknout na libovolnou buňku v požadovaném sloupci matice. Pozadí sloupce s tajenkou se tak obarví modře.



Obrázek 11 - Kompletní úloha typu Křížovky

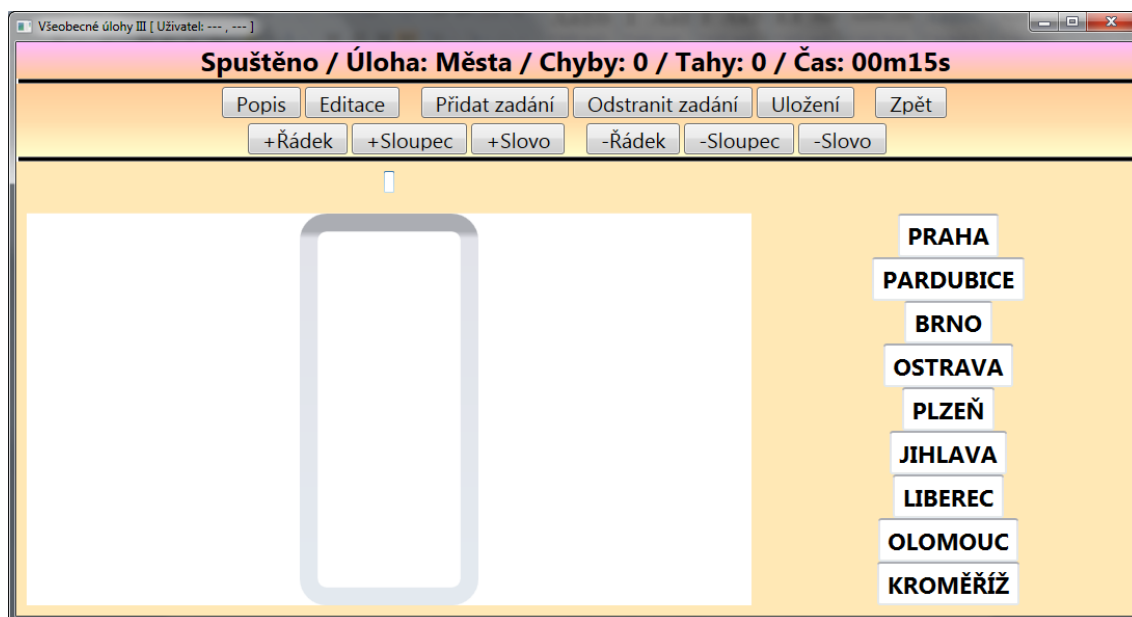
U úlohy typu osmisměrky se nastaví velikost matice na potřebný počet sloupečků a řádků. Velikost pravé matice se nastaví na potřebný počet slov. Do jednotlivých políček se vloží potřebná písmena a do pravé matice se zapíše slova určená k vyhledávání.



Obrázek 12 - Kompletní úloha typu Osmisměrky

Pro vyhledávání v textu není třeba nastavovat velikost matice. To je potřeba pouze v případě, kdy má být matice velká a počet slov je malý. Obecně, pokud je počet polí matice menší než dvojnásobek písmen v zadaných slovech, tak se matice automaticky roztáhne na takovou velikost, aby počet jejích polí byl větší nebo roven dvojnásobku písmen v zadaných slovech. V případě většího počtu polí matice zůstanou její rozměry nezměněny. Zde není umožněno vyplňovat písmena políček matice,

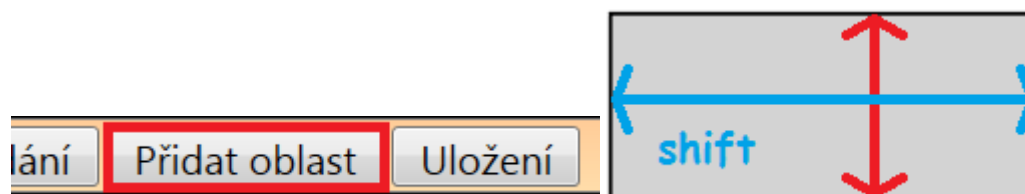
protože ta jsou generována náhodně až při vytváření zadání úlohy. Dále je potřeba vyplnit do pravé matice (spíše sloupečku) slova, která budou náhodně ukryta do textu.



Obrázek 13 - Kompletní úloha typu Vyhledávání v textu

6.3 Spojovačky a obrázky

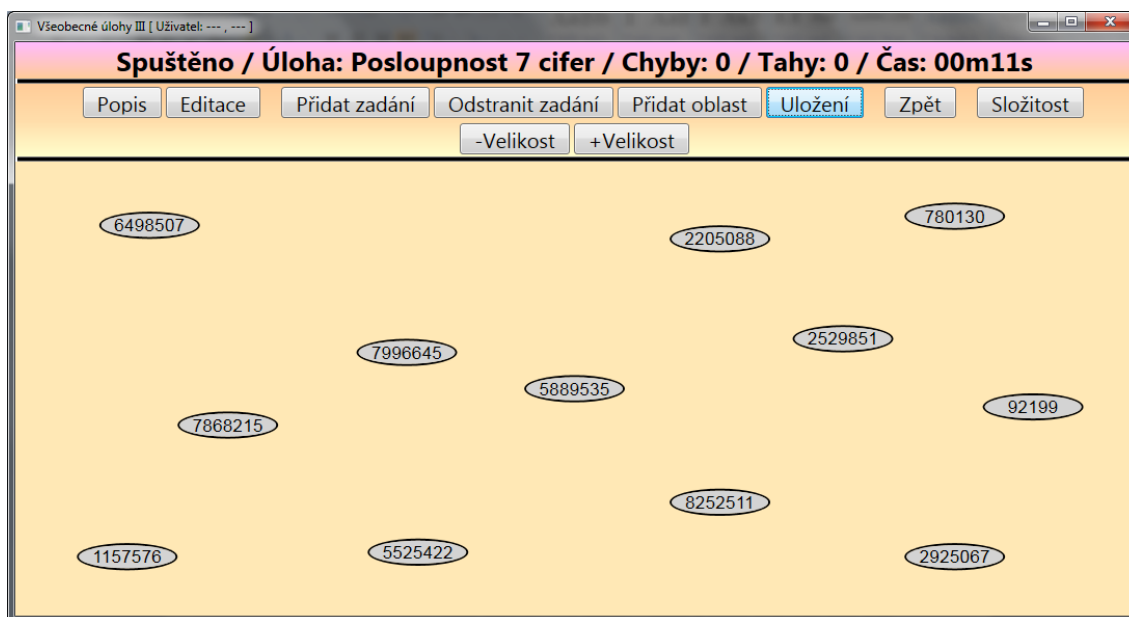
Společnou částí editace těchto úloh je přidávání oblastí určených k označení kliknutím myši nebo dotykem prstu v případě dotykových obrazovek. Přidání oblasti se uskuteční stisknutím tlačítka „Přidat oblast“. Po přidání každé nové oblasti je potřeba tuto oblast umístit na požadované místo a nastavit její velikost. Změna velikosti oblasti se uskuteční pomocí kolečka myši. Upravovanou oblast je potřeba vybrat stisknutím levého tlačítka myši a za jeho současného držení měnit velikost pomocí rolování kolečka myši. Při současném stisknutí klávesy shift se mění horizontální rozměr oblasti. Bez držení klávesy shift se mění vertikální rozměr oblasti. Pro odebrání oblasti z plochy je potřeba na oblast kliknout za současného držení klávesy Ctrl. Dále je možné po přidání všech oblastí měnit velikost všech oblastí najednou pomocí tlačítek „+Velikost“ a „-Velikost“.



Obrázek 14 - Přidání oblasti a její úpravy

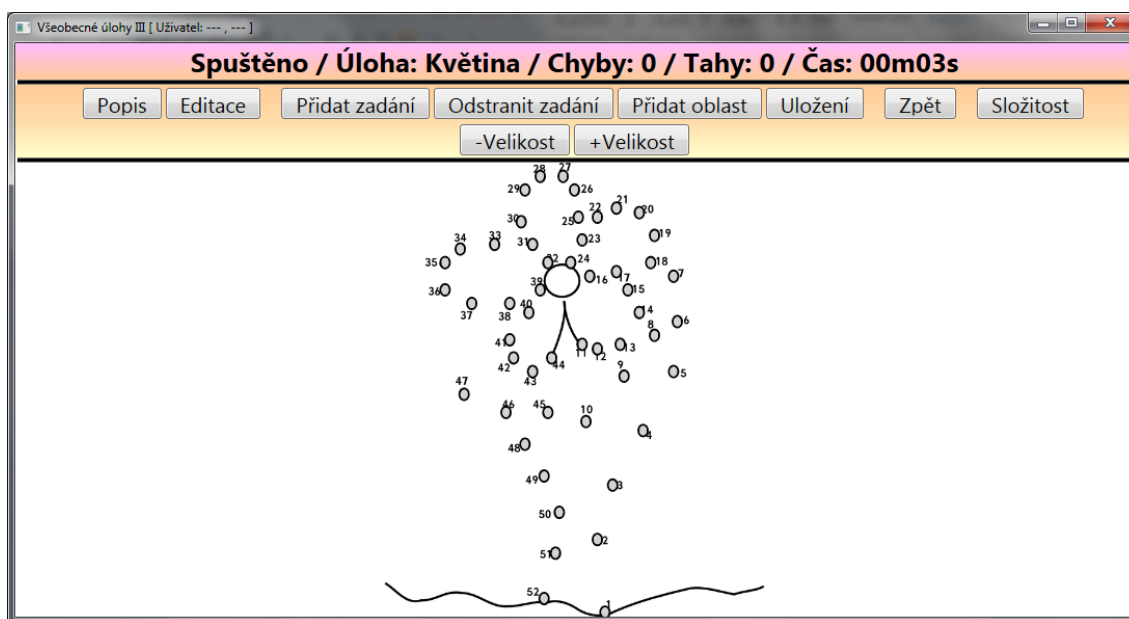
U posloupností se přidá a rozmístí určitý počet oblastí v závislosti na složitosti úlohy. Velikost každé oblasti je potřeba přizpůsobit velikosti čísel, která jsou náhodně generována. Rozsah generovaných čísel je 1 až 10^N , přičemž N je počet cifer. Počet

cifer se nastavuje v dialogu, který se zobrazí po stisknutí tlačítka Složitost. Počet cifer je možné nastavit v rozmezí 3 až 9.



Obrázek 15 - Kompletní úloha typu Posloupnosti

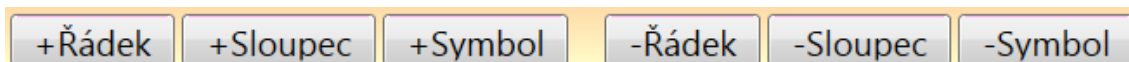
U kreslení (pomocí spojování čísel) je nejprve potřeba vložit obrázek na pozadí. Vložení obrázku na pozadí se uskuteční uchopením a přetažením obrázku do aplikace (například z běžného průzkumníka Windows). Dále je potřeba rozmístit oblasti pro označení v určitém pořadí. Oblast, která bude přidána na plochu jako první se musí přidat na bod s označením 1 a bude muset být při řešení označena jako první. Při editaci existující úlohy budou další oblasti přidány na konec. Je-li třeba přidat označení bodu někam doprostřed, musí se přidat oblast na konec a posunout všechny oblasti o jednu zpět.



Obrázek 16 - Kompletní úloha typu Kreslení

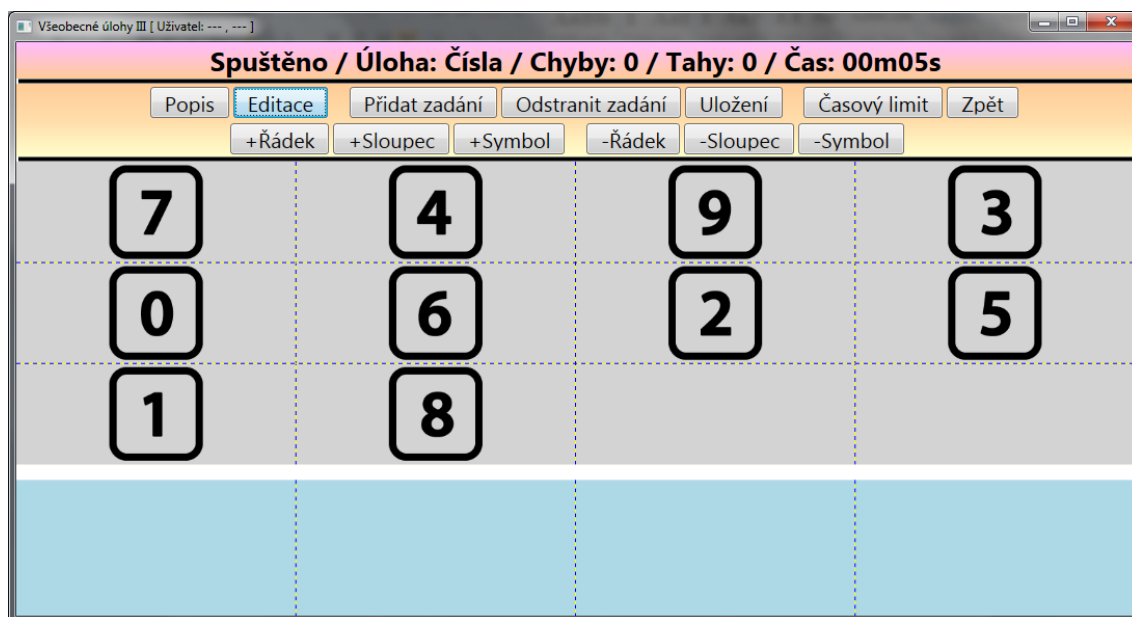
6.4 Paměť

Na začátku úloha obsahuje dvě matice – horní a dolní. Oba typy úloh mají stejnou funkcionalitu pro tvorbu zadání. Pomocí tlačítek „+Sloupec“ a „+Řádek“ se rozměry matice zvětšují, pomocí tlačítek „-Sloupec“ a „-Řádek“ se naopak zmenšují. Minimální rozměr matice je 1x1, maximální pak 6x6. Pomocí tlačítek „+Symbol“ a „-Symbol“ se mění počet rámečků ve spodní matici. Minimum je počet 1 symbol a maximum pak 10. Po stisknutí tlačítka „Časový limit“ se zobrazí dialog s nastavením časového limitu pro zapamatování umístění symbolů.



Obrázek 17 - Nastavení velikosti matic

Po nastavení rozměrů obou matic je potřeba vložit do úloh obrázky symbolů. Vložení obrázků symbolů do úlohy se provede uchopením a přetažením obrázků do horní matice (na obrázku s šedým pozadím). Při spuštění obou typů úloh bude ze sady vložených obrázků náhodně vybráno tolik symbolů, kolik je sloupečků ve spodní matici. U úlohy typu bez umístění bude po uplynutí časového limitu zapojena do úlohy celá sada symbolů vložených při editaci na horní mřížku.

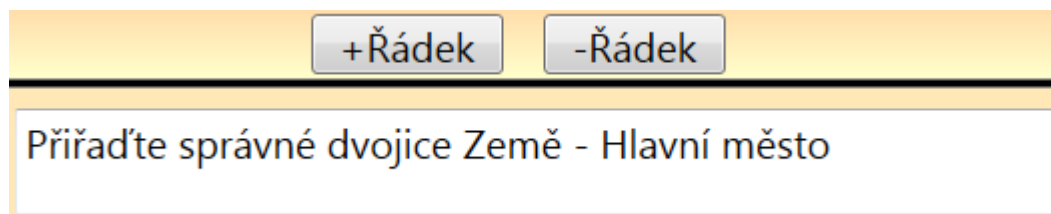


Obrázek 18 - Kompletní úloha ze skupiny Paměť

6.5 Odpovědi a dvojice

Oba typy úloh mají stejnou funkcionalitu pro úpravu velikosti počtu možností. Pomocí tlačítka „+Řádek“ se přidává položka nebo dvojice možností v závislosti na typu úlohy. Pomocí tlačítka „-Řádek“ se naopak odebírají. Minimální počet položek je 1, maximální pak 10.

Dalším společným prvkem editace úloh je vyplnění zadání. Povinné je vyplnit textové zadání do textového pole umístěného v horní části úlohy a dobrovolné je umístit k zadání doprovodný obrázek. Vložení obrázku na pozadí se provede uchopením a přetažením obrázku do aplikace. Pokud v adresáři, odkud se přetahuje obrázek zadání, existuje zvukový soubor se stejným názvem, bude zvukový záznam použit jako zvukové zadání úlohy.



Obrázek 19 - Nastavení počtu odpovědí a vyplnění zadání

U úlohy typu dvojice se po přidání řádku přidá vždy 1 možnost do obou matic. Přidané řádky tvoří dohromady dvojici, která k sobě patří a při řešení úlohy je potřeba tyto dvě možnosti k sobě přiřadit. Hodnoty přidané dvojice se vyplní do textových polí – jedna možnost do levé matice a druhá možnost do pravé matice. Pořadí možností je po spuštění úlohy náhodně proházeno v obou maticích.



Obrázek 20 - Kompletní úloha typu Dvojice

Úloha typu jedna odpověď obsahuje pouze jeden sloupec s možnými odpověďmi, do kterého se přidávají jednotlivé položky/možnosti. Po přidání možností se vyplní do textových polí možné odpovědi. Správná odpověď se vybere kliknutím na její buňku. Pozadí této buňky poté zežlutne. Pořadí odpovědí je po spuštění úlohy náhodně proházeno.

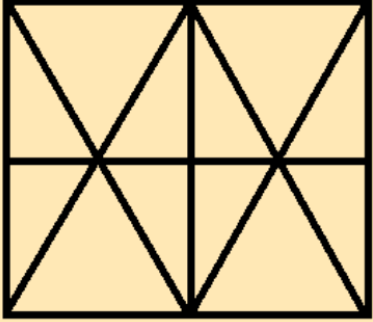
Všeobecné úlohy III [Uživatel: ---, ---]

Spuštěno / Úloha: Počet obrazců / Chyby: 0 / Tahy: 0 / Čas: 00m04s

Popis Editace Přidat zadání Odstranit zadání Uložení Zpět

+Řádek -Řádek

Kolik je na obrázku trojúhelníků?



22
28
30
18

Obrázek 21 - Kompletní úloha typu Jedna odpověď

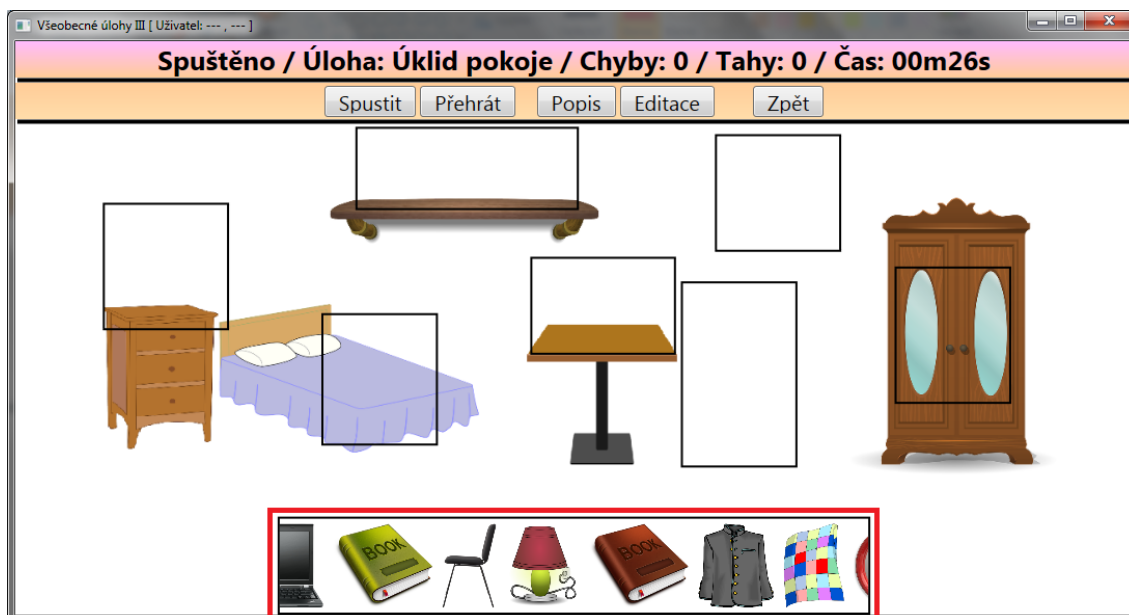
7 Popis ovládání

Následuje popis posloupnosti řešení jednotlivých úloh nyní z pohledu řešitele. V každé podkapitole je umístěna tabulka s přehledem defaultních zvukových záznamů pro daný typ úlohy.

7.1 Posloupnosti akcí

U všech typů úloh jsou na začátku ve startovní oblasti (ve spodní části obrazovky, na obrázku zvýrazněno červeným ohrazením) umístěny všechny obrázky předmětů a to v náhodném pořadí. Při vyzvednutí a následném navrácení obrázku do startovní oblasti je obrázek umístěn do pozice úplně napravo (tedy na konec seznamu obrázků v této oblasti). Pokud jakákoliv oblast obsahuje více obrázků, než je schopna zobrazit, jsou tyto obrázky skryty až za okrajem této oblasti. Při postupném odebrání zobrazených obrázků ze startovní oblasti se obrázky postupně sesouvají k jejímu středu a je možné s nimi poté manipulovat (uchopit je).

V případě, že je potřeba pracovat s obrázkem, který momentálně není vidět, je nutno vyzvednout některé obrázky, které právě vidět jsou a opět je upustit na startovní oblast. Zbylé schované obrázky se budou postupně zobrazovat (sesouvat ke středu oblasti) a naopak nebude vidět obrázek, který se přesunul na konec seznamu. Toto posouvání obrázků ve startovní oblasti není charakterizováno jako chybný tah.



Obrázek 22 - Počáteční stav úlohy ze skupiny Posloupnosti akcí

Rozmístění předmětů se uskuteční přetažením jednotlivých obrázků ze startovní oblasti do příslušných cílových oblastí. Přetažení obrázku lze provést pomocí držení levého tlačítka (při přesunu obrázku) myši nebo položením a přidržením prstu, v případě dotykové obrazovky. Pokud předmět do oblasti patří, tak při uvolnění tlačítka

myši nebo zvednutí prstu se cílová oblast podbarví zeleně a obrázek zůstane umístěn v této oblasti.

Při chybném umístění obrázku se cílová oblast podbarví červeně, obrázek se vrátí do startovní oblasti a zvýší se počítadlo chyb o jednu. Neúspěšný úkon doprovází zvuková nápověda se správným umístěním předmětu. Pokud není zvuková nápověda nastavena, bude přehrán defaultní zvukový soubor oznamující chybné umístění.



Obrázek 23 – Grafické reakce úloh ze skupiny Posloupnosti akcí

	Defaultní zvuk
Zadání	Rozmístíte předměty do vyznačených oblastí.
Nesprávný tah	Sem tento předmět nepatří.
Úspěšné ukončení	Všechny předměty byly správně umístěny.

Tabulka 1 - Defaultní zvuky úlohy typu Nezáleží na pořadí

	Defaultní zvuk
Zadání	Rozmístíte předměty do vyznačených oblastí. Do každé oblasti musejí být předměty vloženy v určitém pořadí.
Nesprávný tah	Předmět vložen v nesprávném pořadí nebo sem tento předmět nepatří.
Úspěšné ukončení	Všechny předměty byly správně umístěny.

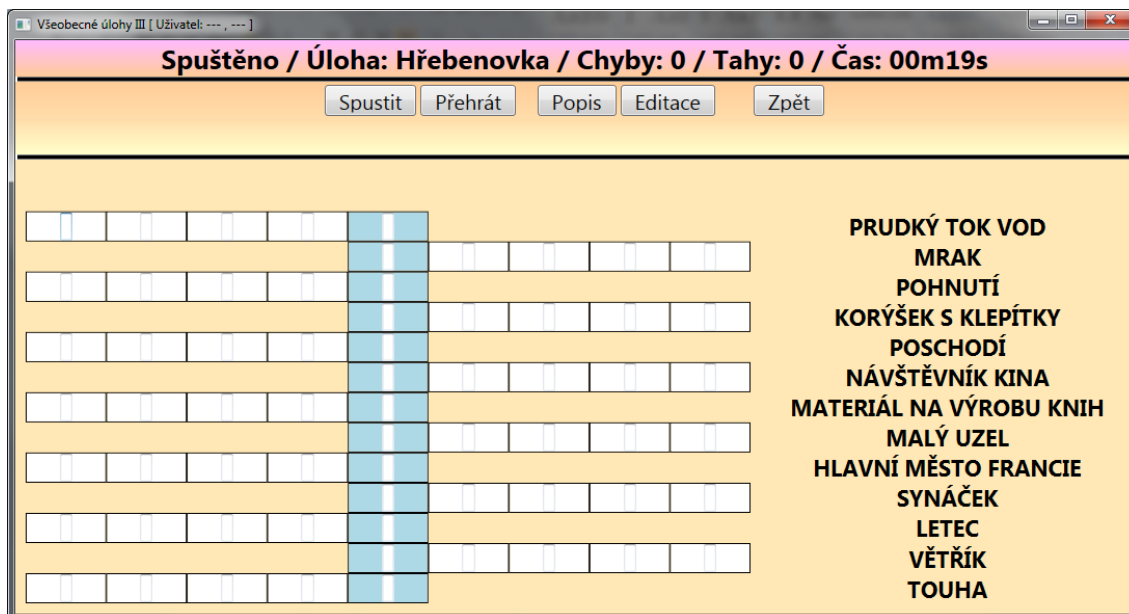
Tabulka 2 - Defaultní zvuky úlohy typu Částečně záleží na pořadí

	Defaultní zvuk
Zadání	Rozmístíte předměty do vyznačených oblastí. Předměty musejí být vkládány do oblastí v určitém pořadí.
Nesprávný tah	Předmět musí být vložen do oblasti v určitém pořadí nebo sem tento předmět nepatří.
Úspěšné ukončení	Všechny předměty byly správně umístěny.

Tabulka 3 - Defaultní zvuky úlohy typu Zcela záleží na pořadí

7.2 Křížovky a osmisměrky

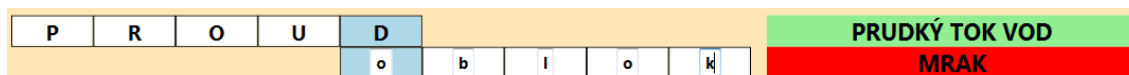
U úlohy typu doplňovačky je v pravém sloupci zobrazen seznam nápověd (legenda ke křížovce). Ke každé nápovědě je potřeba vyplnit správné slovo. Nápověda a k ní přiřazené slovo jsou vždy zobrazeny na stejném řádku.



Obrázek 24 - Počáteční stav úlohy typu Křížovky

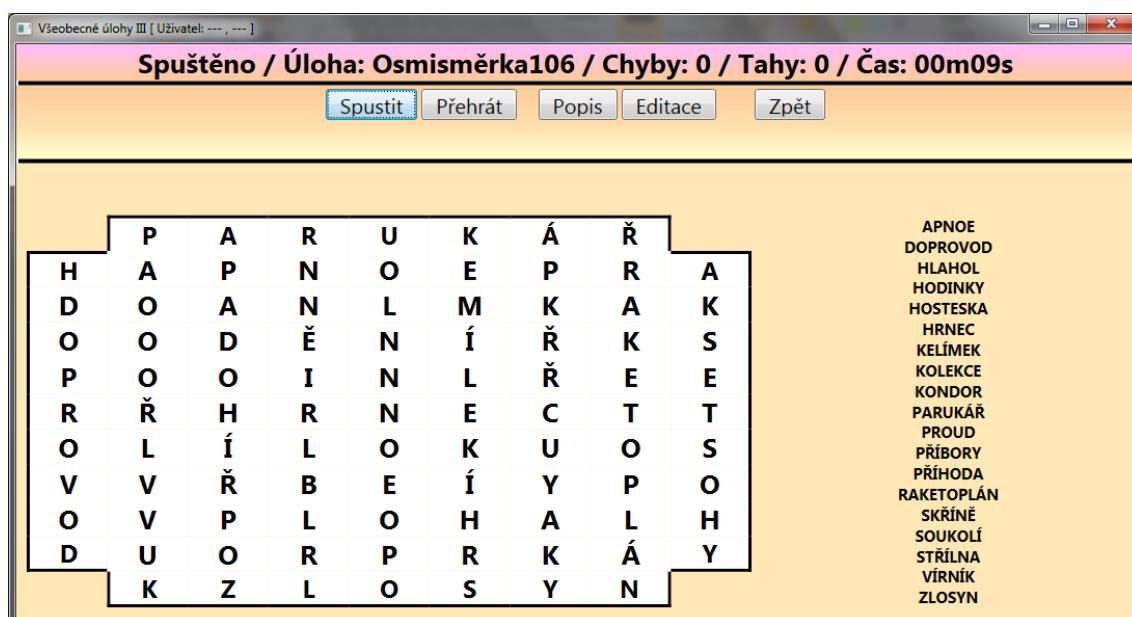
Vyplnění slova se uskuteční postupným klikáním do jednotlivých buněk slova a vepsáním vždy jednoho písmene do každé z těchto buněk. Potvrzení vyplnění se odsouhlasí kliknutím na příslušné slovo legendy (napravo). Při správném vyplnění slova se pozadí slova v legendě zbarví dozelena, zvýší se počítadlo tahů a je přehráno zvukové zadání informující o správně vyplněném slově.

V případě nesprávně vyplněného slova se pozadí nápovědy zbarví červeně, současně s počítadlem uskutečněných tahů se rovněž zvýší i počítadlo chyb a je přehráno zvukové zadání informující o nesprávně vyplněném slově. Poté musí řešitel slovo opravit a může se pokusit znovu potvrdit správnost vyplněného slova (klikem na něho).



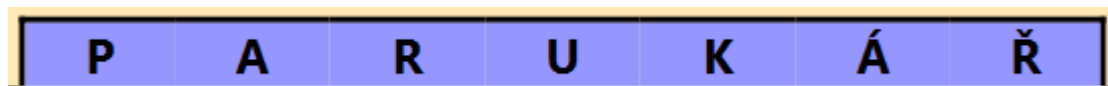
Obrázek 25 - Grafické reakce úlohy typu Křížovky

U úloh typu osmisměrky a vyhledávání v textu je postup řešení velmi obdobný. V pravém sloupci je zobrazen seznam slov, která je potřeba vyhledat a označit. Obsah matice u vyhledávání v textu je při každém spuštění náhodně vygenerován a poté jsou do matice umístěna slova v náhodném pořadí.



Obrázek 26 - Počáteční stav úlohy typu Osmisměrky

Označit slova lze postupným klikáním na buňky (písmena) v matici. Odoznačení písmene se provede opětovným kliknutím na buňku s tímto písmenem. Potvrzení nalezení slova se dosáhne kliknutím na nalezené slovo v pravém sloupci. Při správném nalezení se pozadí slova v pravém sloupci zbarví dozelená, zvýší se počítadlo tahů a je přehráno zvukové zadání informující o správně najitém slově.



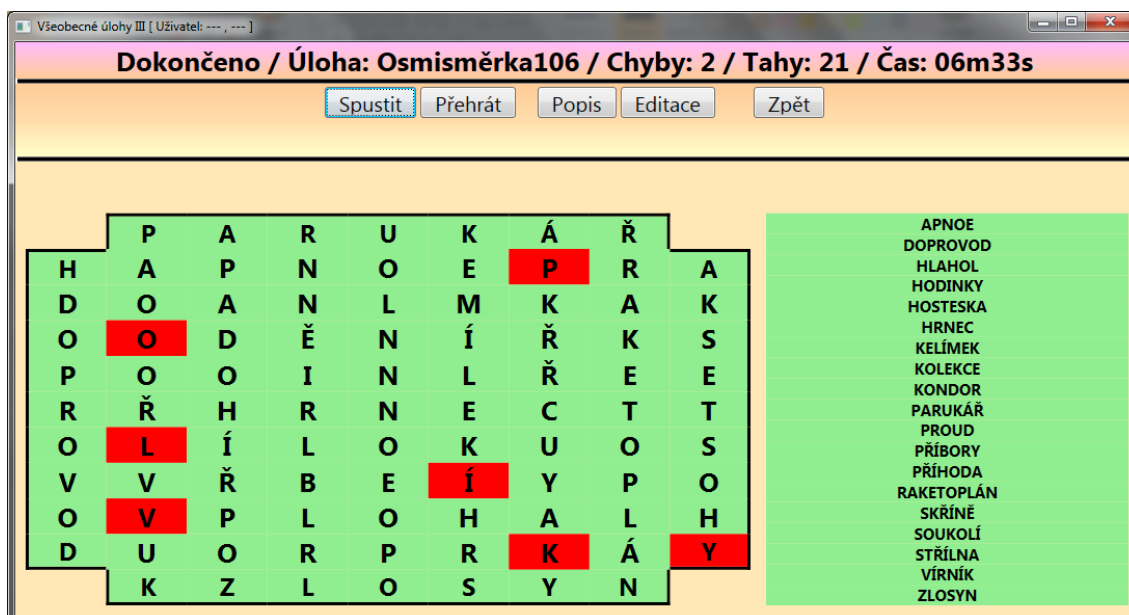
Obrázek 27 - Vybrání slova z písmen matice

V případě nesprávně najitého slova se pozadí slova v pravém sloupci zbarví červeně, současně s počítadlem tahů se zvýší rovněž počítadlo chyb a je přehráno zvukové zadání informující o nesprávně najitém slově. Poté musí řešitel vyhledané slovo opravit a může se pokusit znovu potvrdit správnost najitého slova.



Obrázek 28 - Grafické reakce úlohy typu Osmisměrky

U úlohy typu osmisměrka je po úspěšném vyhledání všech slov znázorněna tajenka začervenaním pozadí zbylých písmen.



Obrázek 29 - Zobrazení tajenky u úlohy typu Osmisměrky

	Defaultní zvuk
Zadání	Doplňte správná slova.
Správný tah	Správná odpověď.
Nesprávný tah	To není správná odpověď.
Úspěšné ukončení	Úloha byla správně vyplněna.

Tabulka 4 - Defaultní zvuky úlohy typu Křížovky

	Defaultní zvuk
Zadání	Vyhledejte a označte slova ze seznamu.
Správný tah	Správně označené slovo.
Nesprávný tah	Toto slovo není správně označené.
Úspěšné ukončení	Všechna slova byla nalezena.

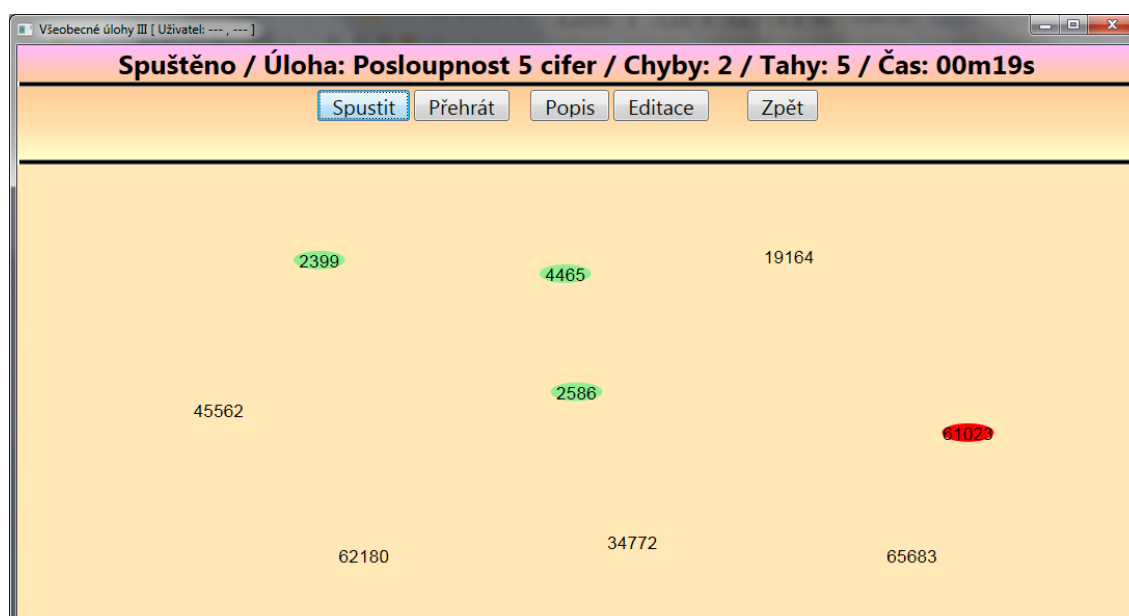
Tabulka 5 - Defaultní zvuky úlohy typu Osmisměrky

	Defaultní zvuk
Zadání	Vyhledejte a označte slova ze seznamu v souvislém textu.
Správný tah	Správně označené slovo.
Nesprávný tah	Toto slovo není správně označené.
Úspěšné ukončení	Všechna slova byla nalezena.

Tabulka 6 - Defaultní zvuky úlohy typu Vyhledávání v textu

7.3 Spojovačky a obrázky

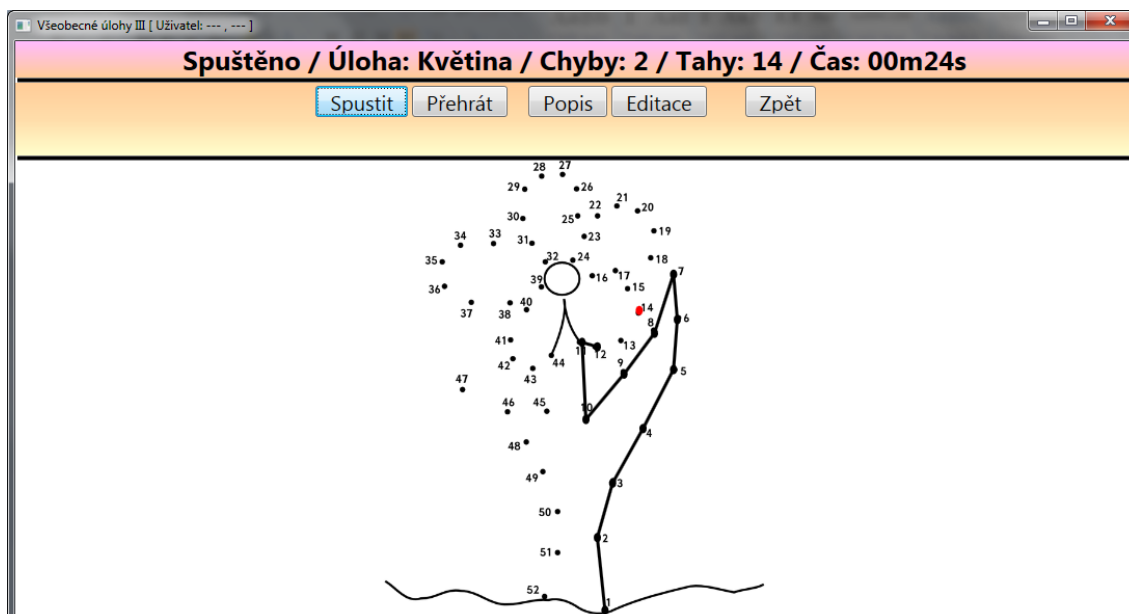
Jak u řazení, tak u kreslení jsou po ploše rozmístěny body s čísly. Je třeba tyto všechny body označit od nejmenšího čísla po největší. U řazení jsou hodnoty čísel při každém spuštění stejného zadání generovány náhodně, zůstávají pouze rozmístění vybraných bodů. Zadání s kreslením obrázků jsou neměnné.



Obrázek 30 - Grafické reakce úlohy typu Posloupnosti

Označit lze bod pomocí stisku levého tlačítka myši nebo ťuknutím prstu, v případě dotykové obrazovky. Pokud je bod označen ve správném pořadí, jeho pozadí se zbarví (dozelená u řazení, dočerna u kreslení), zvýší se počítadlo tahů a je přehráno zvukové zadání informující o správně označeném bodě. V případě úlohy kreslení je správně označený bod doplněn rovněž dokreslením spojnice mezi dvěma posledními body.

Pokud není bod označen ve správném pořadí, jeho pozadí se zbarví dočervena, současně s počítadlem tahů se zvýší také počítadlo chyb a je přehráno zvukové zadání informující o nesprávně označeném bodě. Poté musí řešitel označit jiný bod, o kterém si myslí, že je ten pravý v pořadí.



Obrázek 31 - Grafické reakce úlohy typu Kreslení

	Defaultní zvuk
Zadání	Označte čísla od nejmenšího po největší.
Správný tah	Číslo označeno ve správném pořadí.
Nesprávný tah	Toto číslo není ve správném pořadí.
Úspěšné ukončení	Všechna čísla byla označena ve správném pořadí.

Tabulka 7 - Defaultní zvuky úlohy typu Posloupnosti

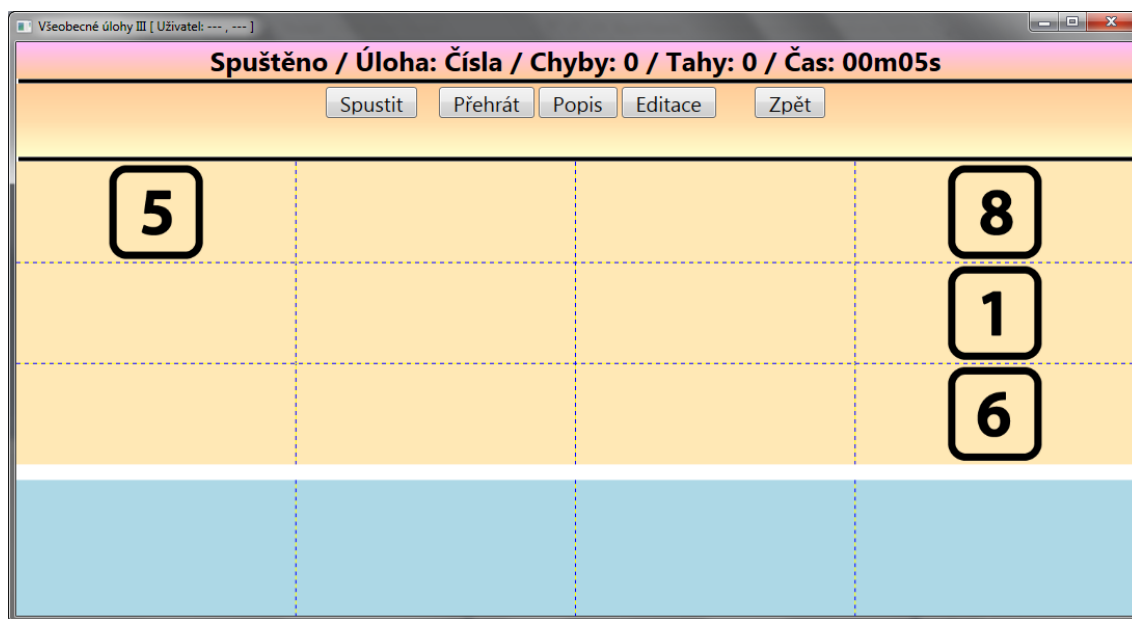
	Defaultní zvuk
Zadání	Označte body ve správném pořadí a dokreslete obrázek.
Správný tah	Správně.
Nesprávný tah	Špatně.
Úspěšné ukončení	Všechny body byly označeny ve správném pořadí.

Tabulka 8 - Defaultní zvuky úlohy typu Kreslení

7.4 Paměť

U obou typů těchto úloh jsou při spuštění zobrazeny dvě matice (horní a dolní). Horní sestává z několika rámečků v několika sloupcích a řádcích. Dolní matice sestává z několika sloupců a vždy pouze jednoho řádku. Zobrazené symboly jsou vybrány náhodně ze sady dostupných symbolů, aby se vždy obměňovali při spuštění stejného zadání. U zadání jsou neměnné pouze rozměry obou matic. Do úlohy vstupuje vždy tolik symbolů, kolik obsahuje spodní matice sloupečků.

U obou úloh je dále nastavený časový limit pro zapamatování zobrazených symbolů. U úloh typu s umístěním jsou symboly zobrazeny v horní matici a je třeba zapamatovat si jejich umístění. Po uplynutí časového limitu jsou symboly přeházeny a zobrazeny v dolní matici. Poté je potřeba rozmístit zapamatované symboly na původní místa do horní matice. Rozmístění zapamatovaných symbolů se dosáhne přetažením jednotlivých obrázků z jejich aktuálních rámečků v dolní matici do příslušných původních rámečků v horní matici.

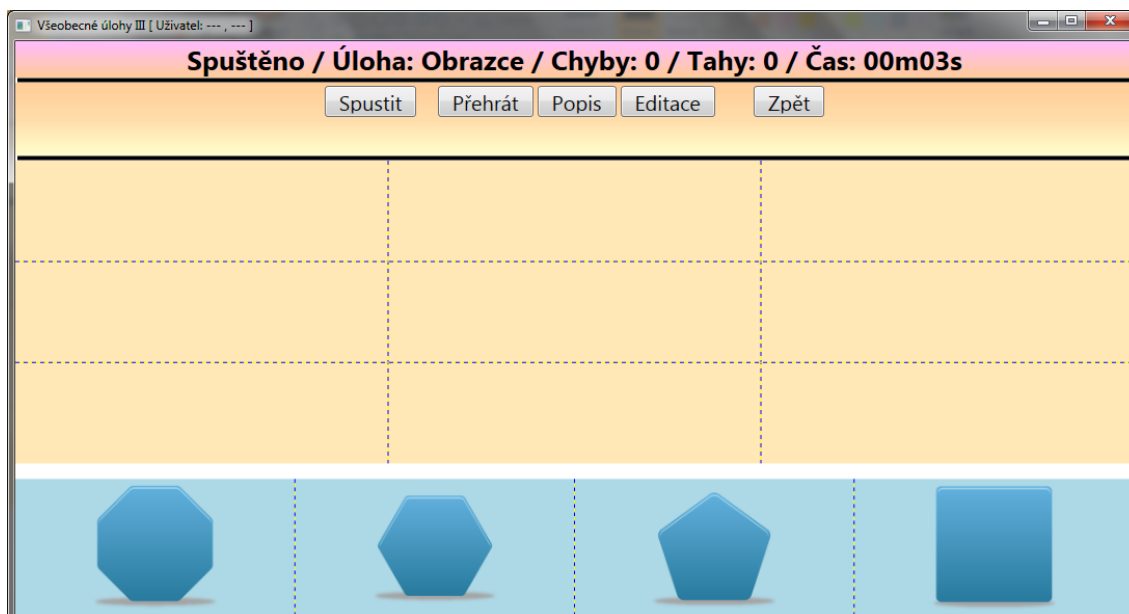


Obrázek 32 - Počáteční stav úlohy typu S umístěním

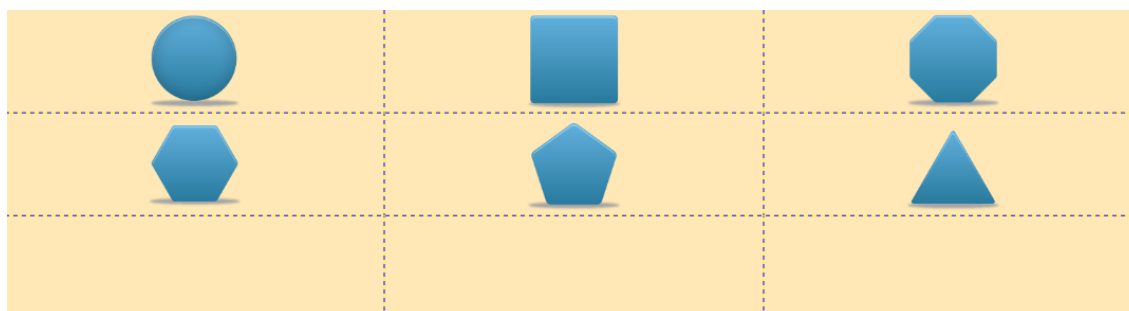


Obrázek 33 - Po uplynutí časového limitu

U úloh typu bez umístění jsou symboly zobrazeny v dolní matici a není třeba si pamatovat jejich umístění. Po uplynutí časového limitu jsou v horní matici zobrazeny nejen zapamatované symboly (z dolní matice), ale i několik dalších symbolů a je potřeba v horní matici vybrat pouze ty zapamatované symboly. Výběr zapamatovaných symbolů se dosáhne přetažením jednotlivých obrázků z jejich aktuálních rámečků v horní matici do libovolného rámečku v dolní matici.



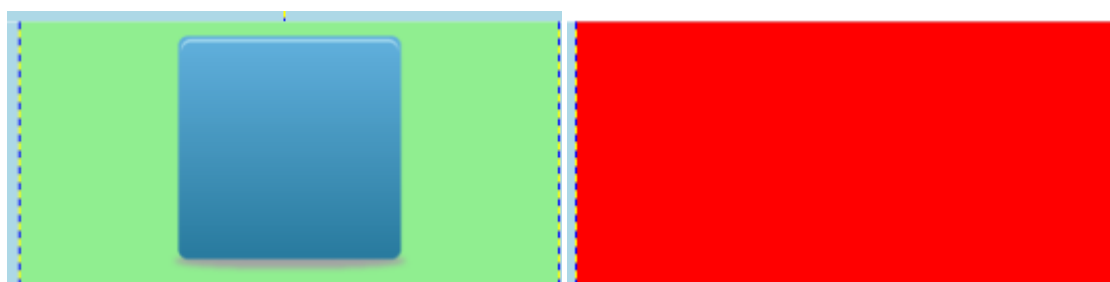
Obrázek 34 - Počáteční stav úlohy typu Bez umístění



Obrázek 35 - Po uplynutí časového limitu

Přetažení obrázku lze uskutečnit pomocí držení levého tlačítka myši nebo položením a přidržením prstu, v případě dotykové obrazovky. Pokud symbol do cílového rámečku patří, tak při uvolnění tlačítka myši nebo zvednutí prstu se cílový rámeček podbarví zeleně, zvýší se počítadlo tahů, obrázek zůstane umístěn v panelu a je přehráno zvukové zadání informující o správně umístěném symbolu.

Při chybném umístění se cílový rámeček podbarví červeně, obrázek se vrátí do aktuálního rámečku, současně s počítadlem tahů se zvýší počítadlo chyb a je přehráno zvukové zadání informující o nesprávně umístěném symbolu.



Obrázek 36 - Grafické reakce úloh ze skupiny Paměť

	Defaultní zvuk
Zadání	Zapamatujte si symboly i s jejich umístěním a počkejte, než se symboly přesunou do spodní matice.
Spuštění řešení	Umístěte zapamatované symboly na svá místa.
Správný tah	Symbol byl správně umístěn.
Nesprávný tah	Sem tento symbol nepatří.
Úspěšné ukončení	Všechny symboly byly správně umístěny.

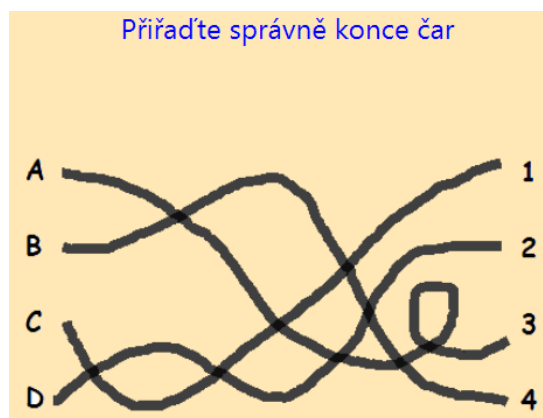
Tabulka 9 - Defaultní zvuky úlohy typu S umístěním

	Defaultní zvuk
Zadání	Zapamatujte si symboly a počkejte, než se zobrazí všechny symboly v horní matici.
Spuštění řešení	Vyberte zapamatované symboly a umístěte je do modře vyznačených polí.
Správný tah	Symbol byl správně umístěn.
Nesprávný tah	Sem tento symbol nepatří.
Úspěšné ukončení	Všechny symboly byly správně vybrány.

Tabulka 10 - Defaultní zvuky úlohy typu Bez umístění

7.5 Odpovědi a dvojice

U úlohy typu možnosti jsou v pravém sloupci zobrazeny položky pro odpověď na otázku, která je zobrazena v textové formě na panelu nalevo od seznamu těchto odpovědí. Na panelu se u některých úloh může také vyskytovat pomocný obrázek pro osvětlení zadání. U úlohy typu dvojice je rozmístění komponent stejné, pouze je nalevo od panelu zobrazen ještě jeden seznam s možnostmi. Možnosti v obou sloupcích jsou u obou typů úloh generovány náhodně (samozřejmě ze zadaného seznamu).

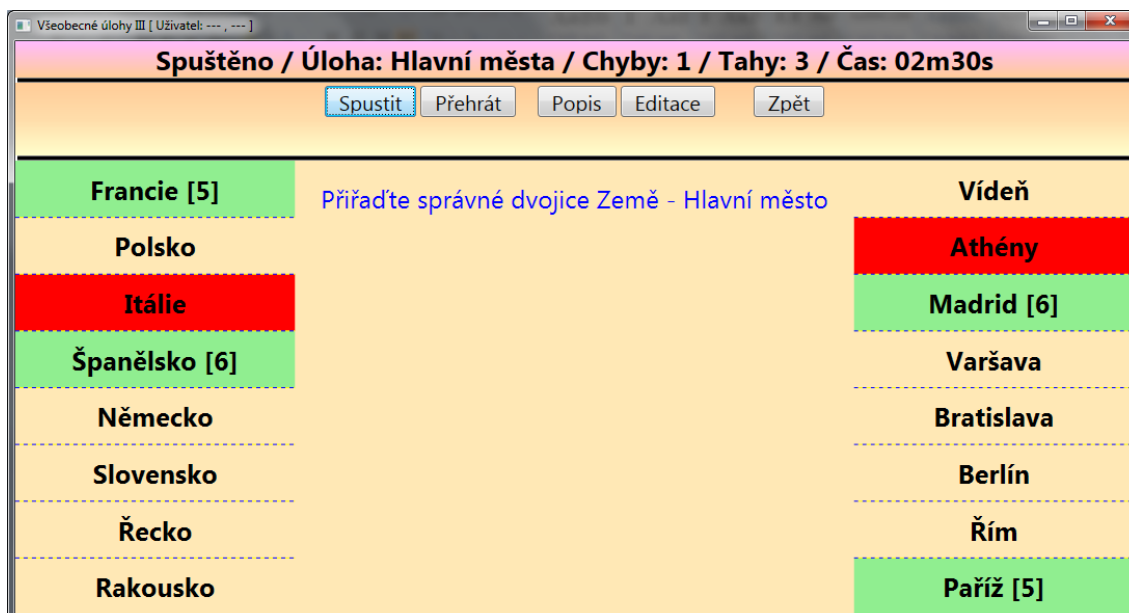


Obrázek 37 - Pomocný obrázek



Obrázek 38 - Počáteční stav úlohy typu Dvojice

U dvojic lze vybrat pouze jednu položku z jednoho sloupce a k ní označit pouze jednu položku z druhého sloupce. V případě opětovného kliknutí do stejného sloupce bude přeznačena vybraná položka. Vybrání položek se uskuteční pomocí stisknutí levého tlačítka myši nebo ťuknutím prstu, v případě dotykové obrazovky, a pozadí první vybrané možnosti se zbarví modře. Po označení správné dvojice se u obou položek ukáže index těchto položek z editačního módu pro přehled, které dvojice k sobě takto patří.



Obrázek 39 - Grafické reakce úlohy typu Dvojice

Pokud je dvojice správně přiřazena, po označení druhé možnosti z dvojice se pozadí obou možností podbarví zeleně, zvýší se počítadlo tahů a je přehráno zvukové zadání informující o správně přiřazené dvojici.

Při nesprávném přiřazení dvojice se pozadí obou možností podbarví červeně, současně s počítadlem tahů se zvýší počítadlo chyb a je přehráno zvukové zadání informující o nesprávně přiřazené dvojici.

Postup u úlohy s možnostmi je v podstatě totožný, jen se neoznačují všechny možnosti v rámci jednoho zadání, ale je potřeba označit pouze správnou odpověď k otázce.

	Defaultní zvuk
Zadání	Spojte dvojice, které k sobě patří.
Správný tah	Správně přiřazená dvojice.
Nesprávný tah	Vybrané položky netvoří dvojici.
Úspěšné ukončení	Všechny dvojice byly správně přiřazeny.

Tabulka 11 - Defaultní zvuky úlohy typu Dvojice

	Defaultní zvuk
Zadání	Označte správnou odpověď.
Správný tah	Správná odpověď.
Nesprávný tah	To není správná odpověď.

Tabulka 12 - Defaultní zvuky úlohy typu Jedna odpověď

Závěr

Soubor vytvořených úloh současně s tímto textem splňuje všechny požadavky zadání této práce. Úlohy jsou rozděleny do 5 skupin podle jejich zaměření a rovněž způsobu řešení uživatelem. Každá ze skupin obsahuje 2-3 různé varianty od každé úlohy. Všechny úlohy byly vytvořeny nejen s ohledem na jejich jednoduchou modifikaci, tedy doplněné o vlastní zadání, ale samozřejmě i snadné ovládání. Ovládání úloh sestává vždy z jednoduché posloupnosti dotyků.

Úlohy vykazují vhodné interaktivní chování ve formě zvukových a grafických výstupů. Všechny zvukové reakce jednotlivých úloh je možné nahradit vlastními zvukovými soubory, ať už se jedná o defaultní zvuky pro všechna zadání nebo u některých úloh, kde je to umožněno, přímo pro každé zadání, nebo dokonce dílčí akci. Grafické reakce jsou založeny na bázi podbarvení aktuálně používaných prvků v průběhu řešení úlohy. Barvy podbarvení byly zvoleny z hlediska běžných konvencí pro správně vykonané kroky zeleně a pro nesprávné naopak červeně.

Pro každou skupinu úloh byla předpřipravena ukázková zadání i s namluvenými zvukovými reakcemi. K jejich tvorbě byly použity volně dostupné obrázky a textové materiály [15] [16] [17] [18].

Srozumitelnost a snadnost ovládání úloh bylo otestováno na uživateliích ve velmi širokém věkovém rozmezí 5 až 80 let. Účastníci testování byli kromě pacientů i pracovníci dozoru. Jejich připomínky k ovládání úloh byly vzaty v úvahu a některé z nich byly přímo nyní ještě zakomponovány v rámci této práce. Některé připomínky nebyly vyhodnoceny jako vhodné k jejich začlenění do práce a některé pro časově náročnější tvorbu, byly určeny k zapracování do budoucna jako další rozšíření (vylepšení) této sady úloh.

Mezi možná rozšíření sady úloh lze tedy zařadit vylepšení ovládání zmíněné v předchozím odstavci. Konkrétně se z připomínek v průběhu testování jedná o výběr slov v matici písmen. Další možné rozšíření sady úloh je vytvoření úplného editoru úloh (úplně nová samostatná aplikace nebo značné rozšíření existujícího konfiguračního módu) pro jednodušší automatickou tvorbu konfiguračních souborů a zdrojových dat zadání.

Zdroje

- [1] PREISS, Marek a Jaro KŘIVOHLAVÝ. *Trénování paměti a poznávacích schopností*. Vyd. 1. Praha: Grada, 2009, 205 s. Psychologie pro každého. ISBN 978-80-247-2738-7.
- [2] KLUCKÁ, Jana a Pavla VOLFOVÁ. *Kognitivní trénink v praxi*. Vyd. 1. Praha: Grada, 2009, 150 s. Psyché (Grada). ISBN 978-80-247-2608-3.
- [3] DOSTÁLOVÁ, Květoslava. Paměť. In: *E-learningová podpora mezioborové integrace výuky tématu vědomí na UP Olomouc* [online]. 2013 [cit. 2015-05-04]. Dostupné z: <http://pfyziolffup.upol.cz/castwiki/?p=4495>
- [4] Paměť a její funkce. In: *Studium psychologie: Psychologie pro každého* [online]. 2015 [cit. 2015-05-04]. Dostupné z: <http://www.studium-psychologie.cz/obecna-psychologie/8-pamet.html>
- [5] Vnímání a pozornost. In: *Studijni-svet.cz: Skupina webů pro chytré studium* [online]. 2013 [cit. 2015-05-04]. Dostupné z: <http://www.studijni-svet.cz/vnimani-a-pozornost/>
- [6] PETERKOVÁ, Michaela. Pozornost: Výběr jen toho skutečně důležitého. In: *Psychotesty a testy osobnosti online* [online]. 2008 [cit. 2015-05-04]. Dostupné z: <http://www.psychotesty.psyx.cz/texty/pozornost.htm>
- [7] SDRUŽENÍ CEREBRUM. Afázie. In: *Poranenimozku.cz* [online]. 2013 [cit. 2015-05-04]. Dostupné z: <http://www.poranenimozku.cz/nasledky-a-rehabilitace/rec-a-komunikace/afazie.html>
- [8] SDRUŽENÍ CEREBRUM. Poruchy prostorové orientace. In: *Poranenimozku.cz* [online]. 2013 [cit. 2015-05-04]. Dostupné z: <http://www.poranenimozku.cz/nasledky-a-rehabilitace/kognitivni-funkce/poruchy-prostorove-orientace.html>
- [9] Myšlení, myšlenkové operace, řešení problémů. In: *Studium psychologie: Psychologie pro každého* [online]. 2015 [cit. 2015-05-04]. Dostupné z: <http://www.studium-psychologie.cz/obecna-psychologie/11-mysleni-myslenkove-operace.html>
- [10] GERONTOLOGICKÉ CENTRUM. *Vzpominkovi.cz* [online]. © 2004-2013 [cit. 2015-05-04]. Dostupné z: <http://www.vzpominkovi.cz/laicka-verejnost/>
- [11] SAMCO. *Samco home* [online]. 2015 [cit. 2015-05-04]. Dostupné z: <http://www.neurop.de/>
- [12] ALPELEPHANT, s.r.o. *Brain Jogging* [online]. © 2015 [cit. 2015-05-04]. Dostupné z: <http://www.brainjogging.cz/>

- [13] ČÁPKA, David. Úvod do C# a .NET frameworku. In: *ITnetwork.cz: Sociální síť pro IT profesionály* [online]. 2012 [cit. 2015-05-04]. Dostupné z: <http://www.itnetwork.cz/c-sharp-tutorial-uvod-do-jazyka-a-dot-net-framework>
- [14] ČÁPKA, David. Úvod do WPF (Windows Presentation Foundation). In: *ITnetwork.cz: Sociální síť pro IT profesionály* [online]. 2013 [cit. 2015-05-04]. Dostupné z: <http://www.itnetwork.cz/c-sharp-tutorial-wpf-uvod-a-prvni-formularova-aplikace>
- [15] BRAXMEIER & STEINBERGER GBR. *Pixabay: Obrázky zdarma* [online]. © 2015 [cit. 2015-05-04]. Dostupné z: <http://pixabay.com/>
- [16] *Pngimg.com: PNG images and cliparts for web design* [online]. © 2013-2015 [cit. 2015-05-04]. Dostupné z: <http://pngimg.com/>
- [17] BÖHMOVÁ, Iva a Zdeněk BÖHM. *Lusk.cz* [online]. 2000 [cit. 2015-05-04]. Dostupné z: <https://www.lusk.cz/index.php>
- [18] *Omalovanky.name: Omalovánky k vytisknutí zdarma* [online]. 2014 [cit. 2015-05-04]. Dostupné z: <http://k-vytisknuti.omalovanky.name/>

Příloha A – Ukázka konfigurace úlohy posloupnosti akcí

```
// konfigurace úlohy částečně záleží na pořadí
<SequencesConfig>
  // startovní oblast a její atributy
  <AreaStart PosX="50" PosY="90" SizeX="100" SizeY="20"
priority="-1"> // oblasti nemají stanovenou prioritu (-1)
  <Items /> // do startovní oblasti nepatří žádné předměty
</AreaStart>
  // cílové oblasti
  <AreaTargets>
    // cílová oblast a její atributy
    <AreaTarget PosX="25" PosY="11" SizeX="13" SizeY="12"
priority="-1"> // oblasti nemají stanovenou prioritu (-1)
    // seznam předmětů patřících do této oblasti
    <Items>
      // předmět s pořadím (priority = 0)
      // soubor obrázku je Item04381
      <Item priority="0">Item04381</Item>
    </Items>
  </AreaTarget>
    // další cílová oblast a její atributy (toto se opakuje pro
každou oblast)
    <AreaTarget PosX="75" PosY="11" SizeX="13" SizeY="12"
priority="-1">
    <Items>
      <Item priority="0">Item08203</Item>
    </Items>
  </AreaTarget>
    <AreaTarget PosX="75" PosY="30" SizeX="36" SizeY="20"
priority="-1">
    <Items>
      // jednotlivé předměty v pořadí, ve kterém je nutno je
do oblasti vložit (priority = 0, 1, 2)
      <Item priority="0">Item02406</Item>
      <Item priority="1">Item09106</Item>
      <Item priority="2">Item06212</Item>
    </Items>
  </AreaTarget>
    <AreaTarget PosX="25" PosY="29" SizeX="25" SizeY="20"
priority="-1">
    <Items>
      <Item priority="0">Item09684</Item>
      <Item priority="1">Item09161</Item>
    </Items>
  </AreaTarget>
    <AreaTarget PosX="75" PosY="50" SizeX="19" SizeY="16"
priority="-1">
    <Items>
      <Item priority="0">Item09085</Item>
      <Item priority="1">Item00684</Item>
    </Items>
  </AreaTarget>
</AreaTargets>
</SequencesConfig>
```

Příloha B – Ukázka konfigurace úlohy křížovky a rébusy

```
// konfigurace úlohy osmisměrka
<TaskCrosswordsVariant rows="11" columns="9" right="22"
column="0"> // rozměry plochy 11x9, legenda obsahuje 22 položek
  <Items> // výpis dvourozměrného pole plochy
    <Item> // první sloupec plochy
      <Item>P</Item> // buňky prvního sloupce a jejich písmena
      <Item>O</Item>
      <Item>D</Item>
      <Item>P</Item>
      <Item>U</Item>
      <Item>L</Item>
      <Item>T</Item>
      <Item>O</Item>
      <Item>V</Item>
      <Item>K</Item>
      <Item>A</Item>
    </Item>
    // postupně jsou takto popsány všechny sloupce plochy
  <ItemWords> // slova legendy
    <ItemWord>ACETON</ItemWord> // jednotlivá slova legendy
    <ItemWord>BEDNA</ItemWord>
    <ItemWord>BLAHO</ItemWord>
    <ItemWord>DÁVKA</ItemWord>
    <ItemWord>DELTA</ItemWord>
    <ItemWord>KLESTÍ</ItemWord>
    <ItemWord>KONKA</ItemWord>
    <ItemWord>OLIHĚŇ</ItemWord>
    <ItemWord>PAVEL</ItemWord>
    <ItemWord>PLZEŇ</ItemWord>
    <ItemWord>PODPULTOVKA</ItemWord>
    <ItemWord>POVEL</ItemWord>
    <ItemWord>PROTEKTOR</ItemWord>
    <ItemWord>PROUD</ItemWord>
    <ItemWord>PŘEDKLON</ItemWord>
    <ItemWord>RATLÍK</ItemWord>
    <ItemWord>ŠEVEL</ItemWord>
    <ItemWord>ŠPROT</ItemWord>
    <ItemWord>VIDLICE</ItemWord>
    <ItemWord>VÝDOBYTEK</ItemWord>
    <ItemWord>VÝSKOK</ItemWord>
    <ItemWord>VÝTLAK</ItemWord>
  </ItemWords>
  // pomocný text tajenky
  <Solution>Lidi myslí pouze, jen aby ospravedlnili své špatné
činy, a hovoří, jen aby maskovali své ...(tajenka).</Solution>
</TaskCrosswordsVariant>
```

Příloha C – Ukázka konfigurace úlohy spojovačky

```
// konfigurace úlohy posloupnosti
// generovaná čísla budou max. 5 ciferná
<OrderingConfig format="5">
  <Points> // výčet jednotlivých bodů
    // bod dotyku s atributy umístění
    <Point PosX="50" PosY="50" SizeX="5" SizeY="5" />
    <Point PosX="55" PosY="83" SizeX="5" SizeY="5" />
    <Point PosX="80" PosY="86" SizeX="5" SizeY="5" />
    <Point PosX="31" PosY="86" SizeX="5" SizeY="5" />
    <Point PosX="18" PosY="54" SizeX="5" SizeY="5" />
    <Point PosX="85" PosY="59" SizeX="5" SizeY="5" />
    <Point PosX="27" PosY="21" SizeX="5" SizeY="5" />
    <Point PosX="49" PosY="24" SizeX="5" SizeY="5" />
    <Point PosX="69" PosY="20" SizeX="5" SizeY="5" />
  </Points>
</OrderingConfig>
```

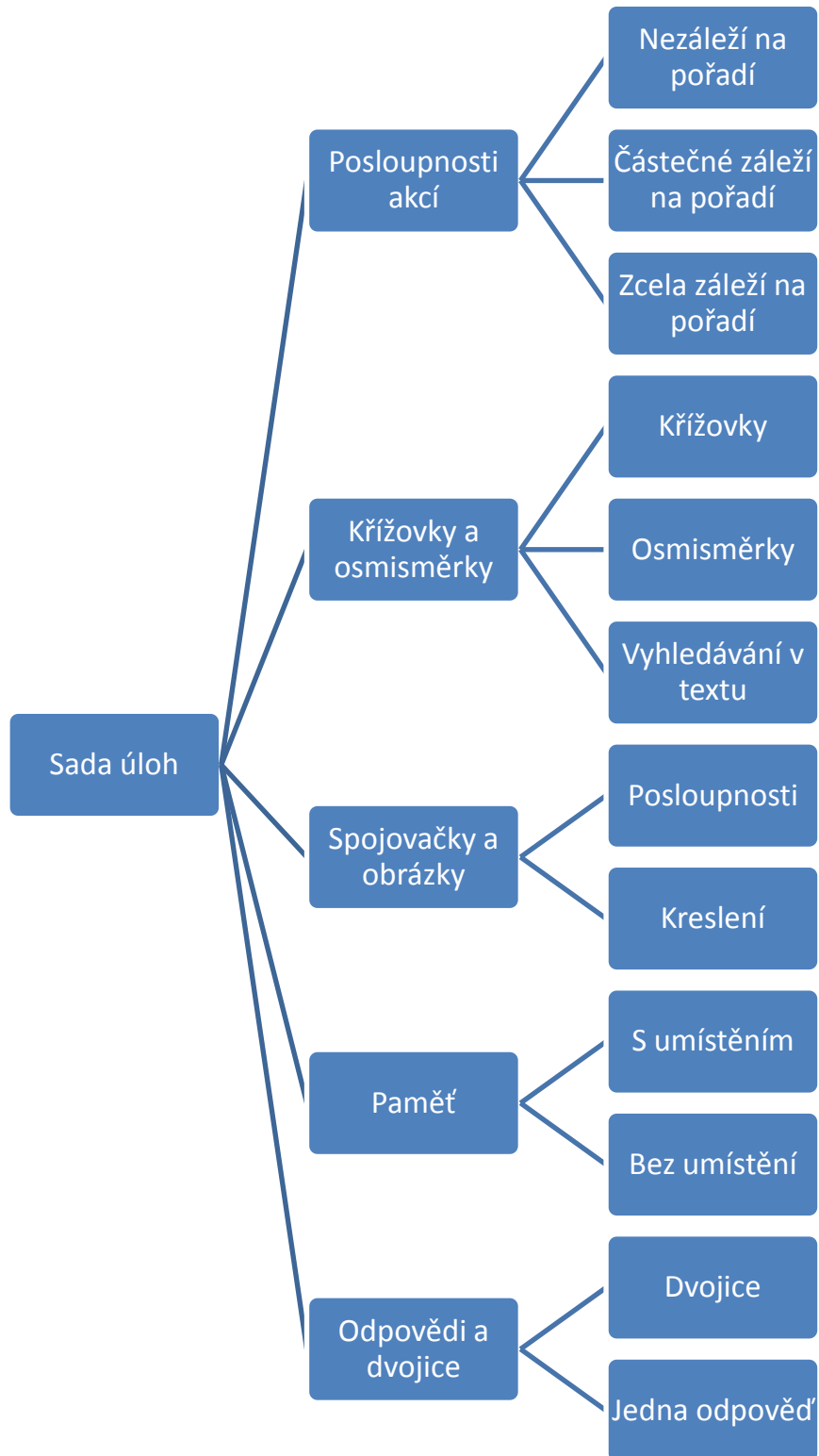
Příloha D – Ukázka konfigurace úlohy paměť

```
// konfigurace úlohy s umístěním
// matice má rozměry 3x4, dolní matice obsahuje 4 sloupce
// předměty se promíchají automaticky po 10 vteřinách
<MemoryConfig rows="3" columns="4" starts="4" timer="10">
  <Items> // výčet jednotlivých symbolů
    // název souboru s obrázkem symbolu
    <Item>Item00403</Item>
    <Item>Item02363</Item>
    <Item>Item06002</Item>
    <Item>Item00726</Item>
    <Item>Item02624</Item>
    <Item>Item08797</Item>
    <Item>Item02001</Item>
    <Item>Item04268</Item>
    <Item>Item02127</Item>
    <Item>Item04777</Item>
  </Items>
</MemoryConfig>
```

Příloha E – Ukázka konfigurace úlohy slova a texty

```
// konfigurace úlohy dvojice s atributem textového zadání
<QuestionsConfig task="Přiřadte správné dvojice Země - Hlavní
město">
  <Answers> // seznam možností
    <Answer> // jedna určitá možnost
      <Pairs> // seznam položek dvojice
        // určitá položka dvojice a její text
        // atribut ok není u této úlohy využit
        <Pair ok="false">Slovensko</Pair>
        <Pair ok="false">Bratislava</Pair>
      </Pairs>
    </Answer>
    <Answer>
      <Pairs>
        <Pair ok="false">Polsko</Pair>
        <Pair ok="false">Varšava</Pair>
      </Pairs>
    </Answer>
    <Answer>
      <Pairs>
        <Pair ok="false">Německo</Pair>
        <Pair ok="false">Berlín</Pair>
      </Pairs>
    </Answer>
    <Answer>
      <Pairs>
        <Pair ok="false">Rakousko</Pair>
        <Pair ok="false">Vídeň</Pair>
      </Pairs>
    </Answer>
    <Answer>
      <Pairs>
        <Pair ok="false">Itálie</Pair>
        <Pair ok="false">Řím</Pair>
      </Pairs>
    </Answer>
    // postupně jsou takto popsány všechny možnosti
  </Answers>
</QuestionsConfig>
```

Příloha F – Hierarchie vytvářené sady úloh



Příloha G – Obsah přiloženého CD

Přiložené CD obsahuje následující složky a soubory:

- Text – obsahuje zdrojový soubor tohoto textu ve formátu DOCX a výsledný needitovatelný soubor tohoto textu ve formátu PDF.
- Aplikace – zde se nachází spustitelná verze sady úloh a úložiště dat pro jednotlivé úlohy. Cesty k spouštěcímu souboru a zdrojovým datům jsou následující:
 - `.\Applications\WIN\CommonTestsAndGames\WIN03\CommonTestsAndGames\WIN03\bin\Debug\CommonTestsAndGames\WIN03.exe` je spouštěcí soubor aplikace
 - `.\AppFilesData\Applications\CommonTasksAndGames03\Data` obsahuje jednotlivé adresáře pro data jednotlivých skupin úloh

Aktuální verze zdrojového kódu aplikace je dostupná u vedoucího práce.