

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



Bakalářská práce

Programy pro výuku dětí s poruchou zraku

Kateřina Štromerová

Vedoucí práce: Ing. Petr Felkel, Ph.D.

Studijní program: Softwarové technologie a management, Bakalářský

Obor: Web a multimedia

27. května 2011

Poděkování

Ráda bych poděkovala panu Ing. Petru Felkelovi, PhD. za trpělivé vedení mé bakalářské práce. Děkuji také Speciálně pedagogickému centru při MŠ Horáčkova a především paní Mgr. Haně Červenákové za velmi přínosnou spolupráci během celého projektu. Mé díky patří také autorům her ze společnosti 7-128 Software za poskytnutí dvou jejich her pro účely rešerše bez nároku na finanční odměnu. Děkuji panu Ing. Tomáši Pastorkovi, autorovi GamLab Studia, za jeho ochotné konzultace při řešení problémů, které se během práce v GamLab Studiu vyskytly. Děkuji také Grantové agentuře ČVUT za poskytnutí potřebných prostředků (číslo grantu SGS10/084/OHK3/1T/13). V neposlední řadě děkuji také panu Bohumilu Tománkovi za hladkou a příjemnou spolupráci na společném projektu.

Prohlášení

Prohlašuji, že jsem práci vypracovala samostatně a použila jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 21. 5. 2011

.....

Abstract

Graphics for some therapy methods for visually impaired children exist currently only in paper form. The aim of this thesis was to digitize these documents and to create electronic version of the exercises. The result is a working program that implements some types of exercises and a collection of images that this program uses. The application was created in GamLab Studio development environment. The program was tested in a special nursery school and is now ready to be used.

Abstrakt

Grafické podklady pro některé metody terapie zrakově postižených dětí existují v současné době pouze v papírové podobě. Cílem této práce byla digitalizace těchto podkladů a vytvoření elektronické verze cvičení. Výstupem je funkční program implementující některé typy těchto úloh a sada obrázků, které tento program využívá. Aplikace byla vytvořena ve vývojovém prostředí GamLab Studio. Program byl otestován ve speciální mateřské škole a je připraven k použití.

Obsah

1	Úvod	1
2	Popis problému, specifikace cíle	3
2.1	Oční poruchy a metody jejich nápravy	3
2.2	Vztah problematiky zrakových postižení k této práci	5
3	Analýza problému a návrh řešení	7
3.1	Katalog požadavků	12
3.1.1	Funkční požadavky	12
3.1.2	Nefunkční požadavky	13
4	Přehled existujících programů	15
4.1	Priory Woods School and Arts College	15
4.2	Inclusive TLC	16
4.3	7-128 Software	18
4.4	Kritéria pro programy	19
5	Grafické podklady	21
5.1	Vektorová grafika	21
5.2	Prostředí Inkscape	21
5.3	Vlastní tvorba grafických podkladů	23
6	Vývojové prostředí GamLab Studio	27
7	Vlastní řešení	29
7.1	Hlavní menu	29
7.2	Ovládání	29
7.3	Matematika 1	30
7.4	Matematika 2	30
7.5	Najdi dvojici 1	32
7.6	Barvy	35
8	Testování	37
8.1	Příprava testu	37
8.2	Průběh testování	38
8.3	Výsledky testování	39

8.4 Diskuze řešení	41
9 Diskuze	43
10 Závěr	47
10.1 Možnosti dalšího vývoje	47
A Žádost pro rodiče	51
B Speciálně pedagogické centrum při MŠ Horáčkova	53
C Skriptovací jazyk Boo	55
D Obsah přiloženého CD	57

Seznam obrázků

3.1	Kniha Rozvoj zrakového vnímání, 2. díl [3]	7
3.2	Příklad úlohy na levo-pravou orientaci	8
3.3	Příklad úlohy na horno-dolní orientaci	9
3.4	Příklad úlohy na rozpoznání odlišného tvaru	9
3.5	Příklad úlohy na rozpoznání správného obrysu	9
3.6	Příklad úlohy Najdi dvojici	9
3.7	Příklad úlohy Přiřaď správnou barvu	10
3.8	Příklad úlohy Urči barvu obrázku	10
3.9	Příklad úlohy na označení zadaného počtu obrázků	10
3.10	Příklad úlohy na určování správného počtu detailů	11
3.11	Příklad úlohy na logické spojování	11
3.12	Příklad doplňovací úlohy	12
3.13	Náhled návrhové prezentace	12
4.1	Snímek obrazovky ze hry Bugz	16
4.2	Snímky obrazovek ze hry Shapes	16
4.3	Snímek obrazovky ze hry Alphabet paint	17
4.4	Snímek obrazovky ze hry 1-2-3 paint	18
4.5	Snímky obrazovky ze hry Orchestra	19
4.6	Snímek obrazovky ze hry Edge	19
5.1	Prostředí editoru Inkscape	22
5.2	Příklady vytvořených obrázků	24
5.3	Příklad rozdělení obrázku beruska.png	25
6.1	Prostředí GamLab	27
7.1	Tři varianty šipky	30
7.2	Snímky obrazovky ze hry Matematika 1	31
7.3	Snímky obrazovky ze hry Matematika 2	31
7.4	Snímky obrazovky ze hry Najdi dvojici 1	33
7.5	Vytvoření obrázku z vygenerované varianty	34
7.6	Snímky obrazovky ze hry Barvy	35
9.1	Nutný posun objektů ve scéně	44

Seznam tabulek

4.1	Porovnání nalezených řešení dle definovaných kritérií	20
8.1	Tabulka odpovědí dětí.	41

Kapitola 1

Úvod

Některé terapeutické metody pro nápravu zrakových postižení u dětí předškolního věku používají v rámci cvičení různé grafické úlohy. V současné době tyto úlohy existují většinou pouze v papírové podobě. Vzhledem k dnešní rozšířenosti informačních technologií vzniká snaha řešit i tato cvičení elektronickou formou, což by přineslo řadu výhod a nových možností oproti současnému stavu, například větší variabilitu úloh, doplnění o zvukové a animační efekty apod. Z tohoto důvodu byla navázána spolupráce se Speciálně pedagogickým centrem při MŠ Horáčkova (dále v textu jen SPC). Toto centrum se věnuje především specializovanému poradenství a terapeutickým službám pro děti se speciálně vzdělávacími potřebami (vadami zraku, kombinovanými vadami a vadami řeči). V rámci své terapeutické činnosti hojně využívá tištěné grafické úlohy a jeho pracovníkům by velmi vyhovovalo mít k dispozici elektronickou verzi těchto cvičení.

Cílem této práce je tedy vytvořit elektronické řešení, které by nahrazovalo papírové cvičební pomůcky. Řešení se skládá z přípravy grafických materiálů a naprogramování funkční aplikace, která by tyto materiály využívala. Důraz je kladen na co nejvyšší variabilitu jednotlivých úloh a jejich maximální možnou automatizaci.

Součástí práce je také průzkum existujících programů, které by mohly tvořit alternativu k výslednému řešení. Průzkum má na základě definovaných kritérií zhodnotit, nakolik by se jednotlivá nalezená řešení dala použít jako cvičení pro zrakově postižené děti.

V závěrečné fázi práce na projektu je provedeno testování s cílovou skupinou - dětmi z mateřské školy Horáčkova. Cílem uživatelských testů je získat informace o vhodnosti a použitelnosti vytvořené aplikace.

Tento projekt je realizován ve spolupráci s dalším studentem Fakulty elektrotechnické, Bohumilem Tománkem.

Základním požadavkem ze strany fakulty je využití vývojového prostředí GamLab, jehož vytvoření bylo předmětem diplomové práce Ing. Tomáše Pastorka.[2] Tento projekt byl podpořen Grantovou agenturou Českého vysokého učení technického v Praze, číslo grantu SGS10/084/OHK3/1T/13.

Kapitola 2

Popis problému, specifikace cíle

V rámci této kapitoly jsou pro zajímavost stručně popsány nejdůležitější zrakové poruchy, pro jejichž léčbu se používají grafické úlohy. Dále jsou zde zmíněny metody nápravy těchto poruch, které tato cvičení využívají. V závěru kapitoly je představena vazba mezi popsanou problematikou a cílem této bakalářské práce.

2.1 Oční poruchy a metody jejich nápravy

Binokulární vidění a jeho poruchy

Binokulární vidění (z lat. binis = dvojitý, oculus = oko) je vidění za použití obou očí současně, v důsledku čehož je člověk schopen vidět oběma očima pozorovaný předmět jednoduchý, nezdvojený. Binokulární vidění není vrozené, ale vyvíjí se do jednoho roku věku dítěte a upevňuje se zhruba do šesti let, kdy dochází k jeho stabilizaci. V případě přerušení normálního vývoje binokulárního vidění vzniká šilhání a tupozrakost - poruchy binokulárního vidění, které spolu úzce souvisejí. Pro úspěšné vyléčení těchto poruch je důležité především jejich včasné objevení a okamžitý začátek léčby.

Šilhavost (strabismus)

Šilhavost je porucha zraku, při níž je porušeno rovnovážné postavení očí a jejich spolupráce. Oči nehledí rovnoběžně, ale jedno se odchyluje. Jednou z příčin šilhání je například velký dioptrický rozdíl mezi oběma očima. V takovém případě dítě stáčí oko, jehož ostrost je snížena. Rozlišují se dva základní typy šilhání. O tzv. sbíhavém šilhání hovoříme tehdy, stáčí-li dítě postižené oko do vnitřního očního koutku, u šilhání rozbíhavého je oko stáčeno do koutku zevního.

Obecně platí, že čím dříve je porucha podchycena a léčena, tím lepší jsou vyhlídky na nápravu. Šance na vyléčení po šestém roce života dítěte konzervativní metodou je velmi nízká.

Mezi tzv. konzervativní metody léčení patří optické pomůcky jako brýle. Používá se tzv. okluzor (např. náplast, brýle s okluzorem z umělé hmoty, neprůhledná kontaktní čočka), aby se nerozvinula tupozrakost. Tato metoda je vhodná, pokud je důvodem šilhání dioptrická porucha. Pokud je léčba těmito prostředky nedostatečná, je možné problém řešit chirurgickou

cestou. U starších lidí se však většinou jedná o nápravu pouze estetickou, neboť neléčená šilhavost může způsobit tupozrakost jednoho oka a ostrost vidění zůstává omezená.

Tupozrakost (amblyopie)

Tupozrakost patří mezi funkční, tedy ne dioptrické, vady zraku. V tomto případě je podstatně snížena zraková ostrost z důvodu nečinnosti oka. Ostrost se může pohybovat od lehce snížené až po praktickou slepotu. Trvá-li tato nečinnost delší dobu, zraková ostrost potlačovaného oka se ještě více sníží, což může vést až těžké tupozrakosti. Takto postižený člověk pak přestává tupozraké oko používat, v důsledku čehož se naruší jeho prostorové vidění.

Podobně jako u šilhavosti je i v případě této poruchy důležité její včasné odhalení, neboť s přibývajícím věkem klesá možnost obnovení funkce vidění.

Podstatou léčby tupozrakosti je obnova aktivity oka, které je docíleno pomocí okluzoru. Tím se zakryje zdravé oko, dítě je pak nuceno používat oko slabé a tedy jej trénovat. Délka a frekvence nošení okluzoru se zkracuje, resp. snižuje současně se zlepšováním stavu.

Důsledky poruch binokulárního vidění

- Porucha vnímání prostoru: dítě nedokáže vnímat prostor trojrozměrně, nedokáže se v něm rychle orientovat
- Porucha vizuálně motorické koordinace: pro dítě je obtížné odhadovat vzdálenost předmětů, má zhoršenou koordinaci končetin
- Porucha analytický - syntetické činnosti: dítě má potíže při čtení a psaní
- Porucha zrakové ostrosti: pro dítě je obtížné vyhledat stejné tvary mezi podobnými, postřehnout jemnější rozdíly tvarů
- Porucha vnímání barev: dítě vnímá barvy matně, méně jasně, nejčastěji je narušeno vnímání červené a zelené
- Narušení zrakových představ: dítě je kvůli nedostatečně kvalitnímu zrakovému vnímání omezeno ve vytváření pojmů

Terapeutické metody k nápravě poruch binokulárního vidění

Terapeutických metod používaných k nápravě poruch binokulárního vidění existuje více, než je v následujícím textu zmíněno. Zde jsou popsány jen ty metody, které souvisí s předmětem této bakalářské práce, tedy s grafickými úlohami.

Okluzivní terapie Okluzivní terapie využívá při léčbě okluzor, kterým se zakrývá zdravé oko, v důsledku čehož je namáháno oko slabé. Postupným cvičením je donuceno k činnosti a tím se jeho stav zlepšuje. Nevýhodou je estetická stránka této léčby. V kolektivu jiných dětí se vyskytuje možnost traumatizace postiženého jeho sociálním okolím a následně i k poruchám jeho chování.

Pleoptická terapie Tato cvičení mají za úkol napomoci správnému monokulárnímu vidění dětí s tupozrakostí. Při této terapii je zdravé oko dítěte zcela zakryto okluzorem. Cvičí

se kromě zrakové ostrosti také koordinace zraku a motorické činnosti. V rámci pleoptické terapie je často využíváno grafických podkladů. Existuje mnoho různých cvičení, například hledání odlišností mezi téměř stejnými obrázky, hledání obrázku podle vzoru, hledání dvojic stejných obrázků, hledání horno-dolních nebo levo-pravých orientací obrázků, různé výtvarné činnosti atp.

2.2 Vztah problematiky zrakových postižení k této práci

V současnosti používané grafické podklady pro pleoptická cvičení existují pouze v papírové formě, což přináší hned několik nevýhod. Tou hlavní je bezesporu nedostatečná variabilita úloh. Vzhledem k opakujícím se podkladům pro tyto úlohy je výrazně snížena zajímavost a zábavnost cvičení, která je zejména pro děti předškolního věku velmi důležitá. Děti si navíc po několikátém opakování stejného úkolu zapamatují procvičovaný problém a terapie tak pozbývá na účinnosti. Například u úloh výběrového typu si zapamatují pozici řešení. Další nevýhody vyplývají ze všeobecných nedostatků papírových řešení vůči elektronickým. Papírové řešení je náročnější na skladovací prostor, složitěji se distribuuje, aktualizuje a kopíruje.

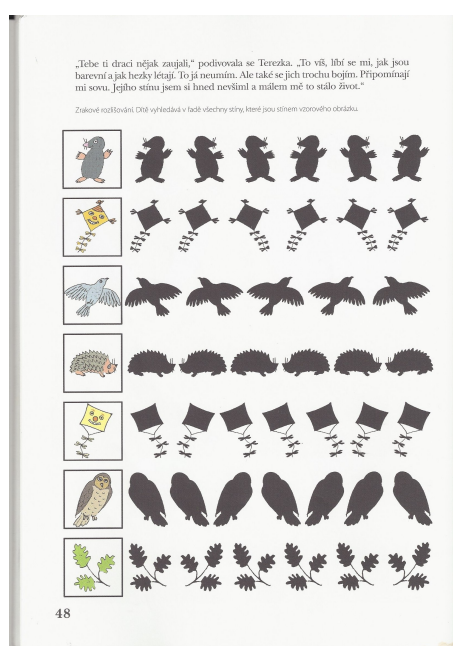
Cílem této bakalářské práce je digitalizace stávajících papírových řešení a jejich rozšíření o nové funkce. Výstupem by pak měl být funkční program implementující některé typy těchto úloh. Toto řešení by nejen eliminovalo výše zmíněné nevýhody, ale zároveň poskytovalo řadu nových výhod - umožňovalo by dětem například seznámit se s počítačem a jeho ovládáním, trénovalo by jejich jemnou motoriku a díky doprovodným zvukovým a animačním efektům zvyšovalo zábavnost a poutavost celých cvičení.

Kapitola 3

Analýza problému a návrh řešení

Vzhledem k tomu, že cílem této práce je vytvoření počítačových alternativ již existujících cvičení v papírové podobě, analýza problému spočívá zejména v analýze, posuzování a klasifikaci papírových předloh a podkladů. K dispozici byly tyto materiály:

- Kniha Rozvoj zrakového vnímání, 2. díl [3], Obrázek 3.1
- Publikace Náprava poruch binokulárního vidění [6]
- Semestrální projekt Bohumila Tománka [1]



Obrázek 3.1: Kniha Rozvoj zrakového vnímání, 2. díl [3]

Dalším zdrojem podnětů byl výstup rešerše, která je součástí této práce. Všechny jmenované podklady byly použity pro rozlišení typů her podle funkcionality a tvořily výchozí

bod pro určování základních požadavků. Na bázi těchto podkladů byla sestavena následující klasifikace her podle typu. U typů her, které byly zpracovány v rámci tohoto projektu, je uvedeno jméno studenta, který příslušnou hru vytvořil.

1. Poznávачky

- Levo-pravá, horno-dolní orientace **Bohumil Tománek**
- Který obrázek nepatří mezi ostatní?
- Najdi správný obrys/siluetu **Bohumil Tománek**
- Najdi stejnou dvojici **Bohumil Tománek, Kateřina Štomerová**
- Najdi tvar v obrázku

2. Hry s barvami

- Přiřaď k předmětu správnou barvu **Kateřina Štomerová**
- Urči barvu obrázku

3. Matematické hry

- Označ zadaný počet obrázků **Kateřina Štomerová**
- Urči obrázek s nejmenším/největším počtem teček, koleček apod.

4. Logické hry

- Najdi logickou dvojici **Bohumil Tománek**

5. Doplnovačky

- Dopln správnou část **Bohumil Tománek**
- Dopln chybějící části

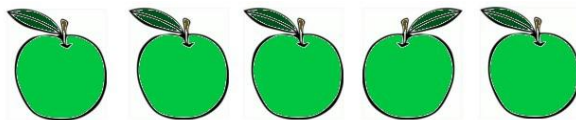
Následuje podrobný popis jednotlivých typů her podle výše uvedeného seznamu.

Poznávачky

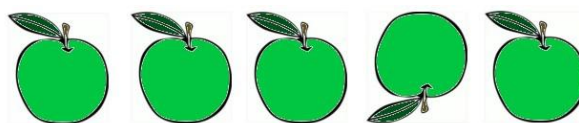
Podstatou těchto úloh je rozpoznat obrázek, který je buď shodný se vzorem, nebo se naopak odlišuje od ostatních. Obrázky se používají jak v barevné, tak obrysové nebo stínové verzi.

Levo-pravá, horno-dolní orientace

Úkolem v tomto cvičení je poznat ten obrázek v řadě, který je buď vertikálně nebo horizontálně překlopený.



Obrázek 3.2: Příklad úlohy na levo-pravou orientaci



Obrázek 3.3: Příklad úlohy na horní-dolní orientaci

Který obrázek nepatří mezi ostatní?

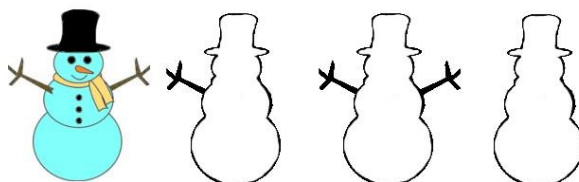
Dítě má v této úloze poznat obrázek, který se od ostatních odlišuje, i když je tvarem velmi podobný. Rozdíl může být také pouze v detailech.



Obrázek 3.4: Příklad úlohy na rozpoznání odlišného tvaru

Najdi správný obrys/siluetu

Zde je úkolem najít správný obrys nebo siluetu ke vzorovému barevnému obrázku.



Obrázek 3.5: Příklad úlohy na rozpoznání správného obrysu

Najdi stejnou dvojici

V této úloze má dítě za úkol označit dva takové obrázky, které jsou shodné. Rozložení všech obrázků může být dvojího typu: v řadě a nahodilě. Všechny obrázky jsou si buď tvarově podobné, nebo se liší v detailech.



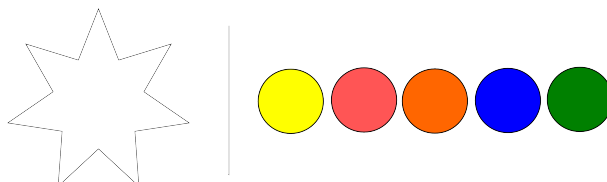
Obrázek 3.6: Příklad úlohy Najdi dvojici

Hry s barvami

Hry s barvami se zaměřují na trénink rozlišování barev. Pomocí těchto cvičení se napomáhá zjemnění barvocitu očí dítěte.

Přiřaď k obrázku správnou barvu

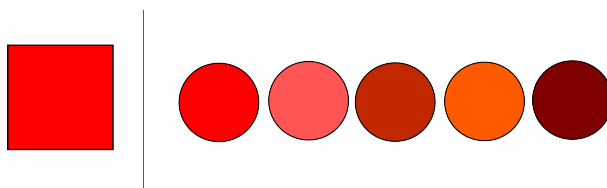
Úkolem této hry je vybarvit obrázek v obrysové formě správnou barvou. Dítě takto trénuje také spojení zrakového vjemu s pojmenováním objektu, např. tráva - zelená.



Obrázek 3.7: Příklad úlohy Přiřaď správnou barvu

Urči barvu obrázku

V této úloze má dítě za úkol určit, jakou barvu má barevný obrázek. K dispozici je paleta barev podobného odstínu.



Obrázek 3.8: Příklad úlohy Urči barvu obrázku

Matematické hry

Matematické hry spojují trénink očí s jednoduchým počítáním. U některých úloh je třeba, aby se dítě zaměřilo na detaily obrázků, neboť ty má za úkol spočítat.

Označ zadaný počet obrázků

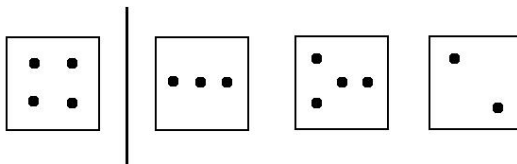
V této matematické hře má dítě za úkol vybarvit nebo jiným způsobem označit tolik obrázků, kolik udává číslo.



Obrázek 3.9: Příklad úlohy na označení zadaného počtu obrázků

Urči obrázek s nejmenším/největším/příslušným počtem teček, koleček apod.

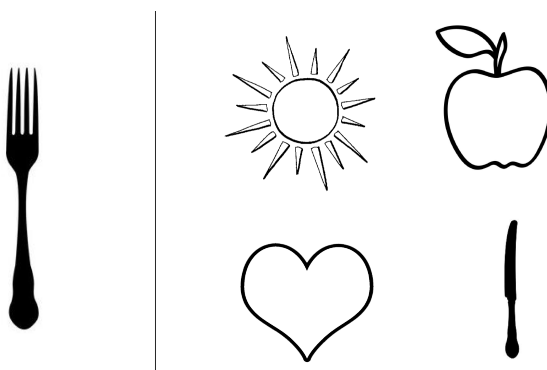
Zde je cílem určit mezi ostatními ten obrázek, který má buď nejméně nebo nejvíce detailů, popř. právě tolik, kolik jich má vzorový obrázek.



Obrázek 3.10: Příklad úlohy na určování správného počtu detailů

Logické hry

Hra Najdi logickou dvojici není založená na hledání shodných tvarů, ale na spojování obrázků, které mají mezi sebou logickou souvislost.



Obrázek 3.11: Příklad úlohy na logické spojování

Doplňovačky

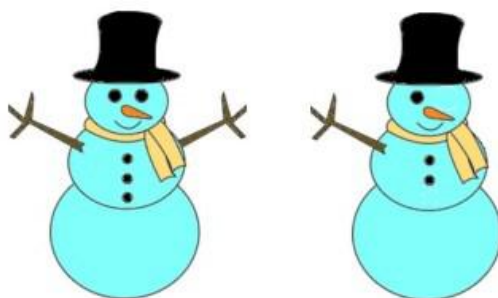
V doplňovacích hrách má dítě za úkol hledat buď chybějící části obrázku nebo jednu část, která v obrázku chybí.

Doplň správnou část

V této hře chybí ve vzorovém obrázku vždy jedna část, dítě má vybrat z palety nabídnutých možností tu správnou.

Doplň chybějící části

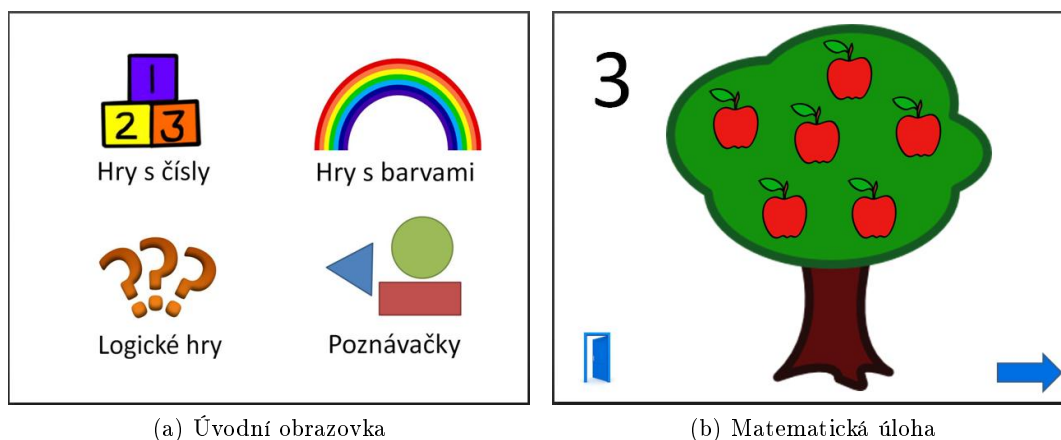
Klasická doplňovačka, kdy se dítě podle vzorového obrázku snaží najít ty části, které v druhém obrázku chybí.



Obrázek 3.12: Příklad doplňovací úlohy

V další fázi analýzy byla navázána úzká spolupráce s Mgr. Hanou Červenákovou, pedagožkou SPC. Její podněty a návrhy daly základ požadavkům na finální program.

Na základě předpřipravené analýzy a úvodního setkání s Mgr. Červenákovou byla připravena interaktivní prezentace - návrhový model práce, Obrázek 3.13. Prezentace byla vytvořena v programu Microsoft Office Powerpoint 2007 a simulovala rámcovou funkčnost budoucího programu. Představovala především grafický návrh. Tento model byl paní Červenákové odprezentován, její připomínky zapracovány a vzniknul definitivní návrhový model s katalogem požadavků, který byl použit pro samotnou implementaci.



Obrázek 3.13: Náhled návrhové prezentace

3.1 Katalog požadavků

3.1.1 Funkční požadavky

1. Všechny úlohy budou součástí jedné aplikace
2. Aplikace bude obsahovat hlavní menu, kde bude možné vybrat typ hry

3. Různé hry ve stejné kategorii se budou spouštět náhodně za sebou
4. Všechny hry budou mít stejné navigační ovládání
 - Každá hra bude obsahovat tlačítko pro návrat do hlavního menu.
 - Každá hra bude obsahovat tlačítko na vyhodnocení a posun na další úlohu (ve stejné kategorii)
5. Obohacení her o zvukové efekty

3.1.2 Nefunkční požadavky

1. Jednoduché obrázky
2. Silné kontury u obrázků
3. Barevnost a kontrast obrázků
4. Logické uspořádání
 - Ne příliš obrázků na ploše
 - Mezery mezi obrázky
5. Ovládání myši + velký ukazatel myši
6. Snadná rozšiřitelnost o nové obrázky
7. Implementace v prostředí GamLab + jazyk Boo
8. Maximální možná automatizace generování obrázků.
9. Snadná rozšiřitelnost o nové obrázky.

Kapitola 4

Přehled existujících programů

Tato kapitola je výsledkem rešerše existujících dostupných softwarových řešení her pro zrakově postižené děti. Následuje podrobný popis jednotlivých nalezených řešení, jejichž vlastnosti jsou v závěru kapitoly uspořádány do přehledné tabulky.

4.1 Priory Woods School and Arts College

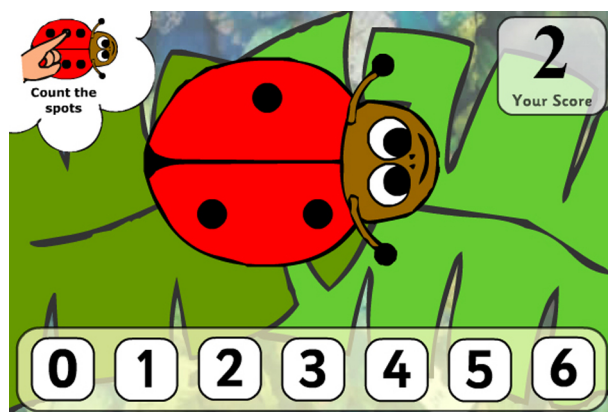
Webové stránky: <http://www.priorywoods.middlesbrough.sch.uk/>

Priory Woods School and Arts College je speciální škola se zaměřením na vzdělávání žáků s vážnými poruchami učení a speciálními potřebami. Je součástí projektu European Comenius založeného Evropskou Unií, který je zaměřen na kreativní umělecké, sportovní a jiné venkovní aktivity.

Priory Woods poskytuje volně na svých webových stránkách několik her, které vyhovují většině definovaných požadavků. Pro představu jsem vybrala dvě z nich.

Bugz - Brouci

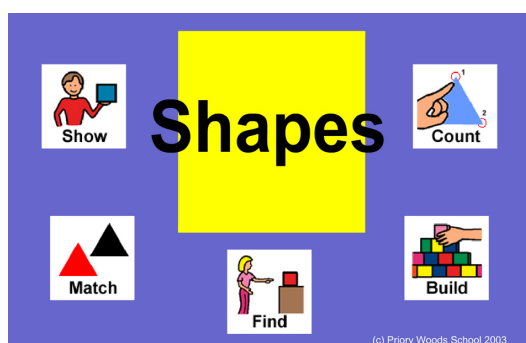
V této jednoduché hře má hráč za úkol nakrmit ještěrku Larryho beruškami se správným počtem teček. Po spuštění hry je nejprve zobrazena úvodní obrazovka. Kliknutím kdekoliv na obrazovce se zobrazí instrukce ke hře, dalším kliknutím je spuštěna hra samotná. Na obrazovku přileze beruška s určitým počtem teček a v dolní části obrazovky se objeví panel s čísly od 1 do 6. Hráč má za úkol kliknout na číslo, které odpovídá počtu teček na krovkách berušky. V případě úspěchu se zobrazí animace s Larrym, který berušku sní, v opačném případě je hráč vyzván k opětovnému pokusu.



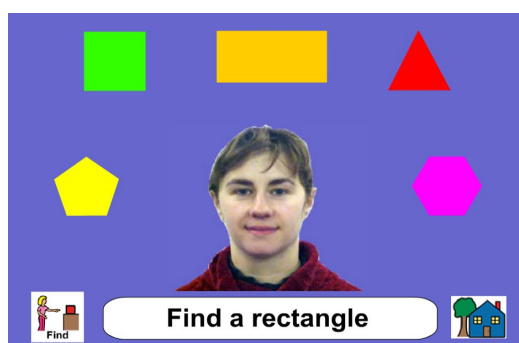
Obrázek 4.1: Snímek obrazovky ze hry Bugz

Shapes - Tvary

Tato hra je poněkud rozsáhlejší, program nabízí několik možností her se základními tvary, jako například spojování stejných tvarů, určování správného tvaru podle zadání nebo počítání rohů jednotlivých tvarů. Hra je doprovázena slovními instrukcemi. Obrázky jsou velmi jednoduché, taktéž ovládání.



(a) Hlavní menu



(b) Úloha Find - Najdi

Obrázek 4.2: Snímky obrazovek ze hry Shapes

4.2 Inclusive TLC

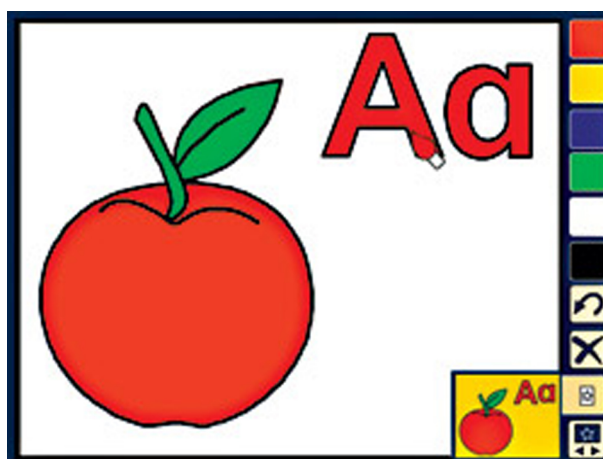
Webové stránky: <http://www.inclusivetlc.com/>

Inclusive TLC (Technology - Learning - Communication) je součástí společnosti Inclusive Technology Ltd., která je největším dodavatelem vzdělávacích kompenzačních technologií ve Velké Británii. Specializuje se na vzdělávací software, spínače a další zařízení pro přístup k počítači a jednoduché komunikační pomůcky a podpůrné technologie pro studenty s těžkou poruchou učení a tělesným nebo smyslovým postižením.

Licence na jejich software je ovšem zpoplatněna vysokou částkou, nebyla tedy možnost hry vyzkoušet. Údaje o hrách jsou čerpány přímo z webových stránek Inclusive TLC, rovněž snímky obrazovek jsou odtud převzaty.

Alphabet paint

Hra Alphabet paint je založena na principu klasických omalovánek. Úkolem hráče je vybarvit obrázek správnou barvou podle reality. Název předmětu na obrázku vždy začíná jedním písmenem z abecedy. Hráč má tedy možnost procvičit si kromě motorických dovedností i své schopnosti didaktické, zároveň se učí přiřazovat správné barvy k doprovodnému obrázku. Jeho úkony doprovází zvuky a animace, takže je hra nejen výchovná a vzdělávací, ale také zábavná. Obrázky jsou kontrastní a velmi jednoduché, hra je tedy vhodná pro zrakově postižené osoby.



Obrázek 4.3: Snímek obrazovky ze hry Alphabet paint

1-2-3 Paint

Tato hra funguje obdobně jako Alphabet paint ovšem s čísly. Obrázky jsou vysoce kontrastní, jednoduché, tedy velmi vhodné pro zrakově postižené děti. Úkoly jsou rozděleny do čtyř úrovní obtížnosti. Hra je bohatě doprovázena zvuky, které zvyšují hráčovu motivaci. Hráč je odměněn v případě, že vybarví obrázek podle vzoru správně.



Obrázek 4.4: Snímek obrazovky ze hry 1-2-3 paint

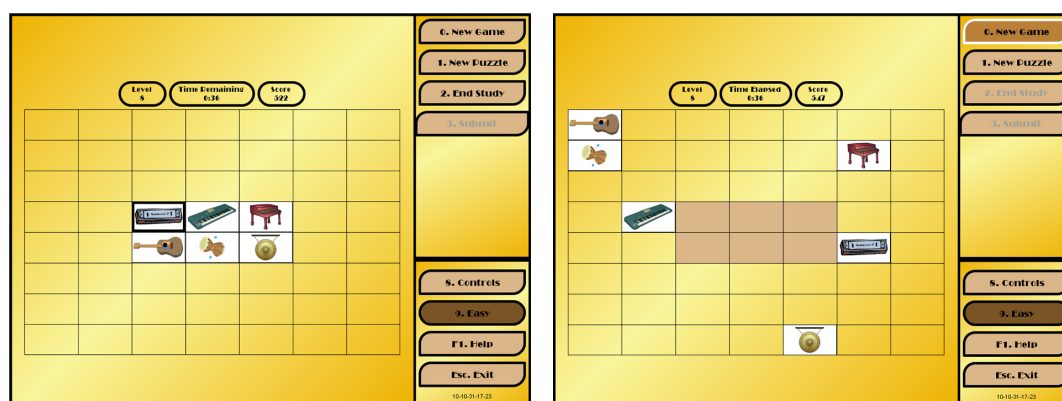
4.3 7-128 Software

Webové stránky: www.7128.com

7-128 Software je skupina zabývající se tvorbou jednoduchých her. Jejich hry jsou velmi dobře použitelné i pro osoby s různými postiženími, od úplně nevidomých přes zrakově či pohybově postižené až k osobám s postižením sluchovým. Svoji tvorbu nazývají souhrnně Simply Entertainment, což samo o sobě naznačuje herní jednoduchost, která je v tomto případě kompenzována zábavou spíše intelektuální. 7-128 Software poskytují své hry za finanční poplatek, nicméně byli velice ochotní uvolnit dvě ze svých her pro účely této bakalářské práce bez nároku na finanční odměnu. Následuje podrobný popis vybraných her.

The Orchestra Game

Hra s názvem Orchestra trénuje jednak hráčovu paměť, jednak jeho jemné motorické dovednosti. Na herní ploše je na začátku hry umístěno několik obrázků hudebních nástrojů, viz Obrázek 4.5a, z nichž každý vydává nějaký konkrétní zvuk. Hráč dostane nějaký čas na zapamatování si pořadí hudebních nástrojů, úkolem je pak seřadit hudební nástroje tak, jak byly umístěny v první scéně, Obrázek 4.5b. Umisťování jednotlivých obrázků s nástroji funguje na principu Drag&Drop.



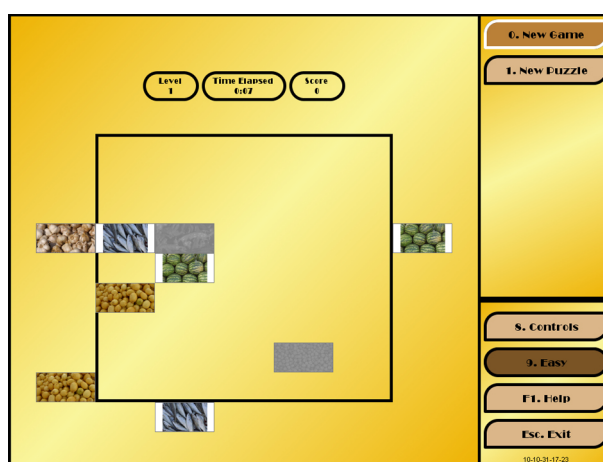
(a) Vzorové seřazení hudebních nástrojů

(b) Úloha: seřadit nástroje

Obrázek 4.5: Snímky obrazovky ze hry Orchestra

The Edge Game

Úkolem hráče ve hře Edge je přiřazovat pomocí Drag&Drop k sobě dvojice obdélníků se stejným vzorem. Hrací plocha je ohraničena černou konturou, uvnitř které se postupně objevují vzorované obdélníky. Vně hrací plochy se na různých místech objeví jejich shodné dvojice. Čas od času se na hrací ploše zobrazí obdélníky šedé, se kterými se nedá pohybovat a hráč se jim musí obratně vyhnout. Hra je rozdělena do deseti úrovní podle obtížnosti.



Obrázek 4.6: Snímek obrazovky ze hry Edge

4.4 Kritéria pro programy

Ve spolupráci s SPC byly stanoveny požadavky na funkcionalitu, ovládání a vizuální stránku počítačových her.

- Jednoduché obrázky

- Silné kontury u obrázků (alespoň 5 px)
- Barevné a kontrastní obrázky
- Logické uspořádání
 - Ne příliš obrázků na ploše
 - Mezery mezi obrázky
- Propojení zraku a sluchu
- Ovládání myši + velký ukazatel myši

Následující tabulka přehledně zobrazuje, nakolik nalezená řešení vyhovují stanoveným požadavkům. Některé údaje nebylo možné ověřit, neboť hry nebyly k dispozici na vyzkoušení.

	Jednoduchost	Kontury	Barevnost, kontrast	Logické usp.		Zvuky	Ovl. myši	Kurzor
				Málo obr.	Mezery			
Bugz	✓	✓	✓	✓	✗	✓	✓	✗
Shapes	✓	✗	✓	✓	✓	✓	✓	✗
Alphabet	✓	✓	✓	✓	✓	✓	✓	?
1-2-3	✓	✓	✓	?	?	✓	✓	?
Orchestra	✓	✓	✓	✓	✗	✓	✓	✗
Edge	✓	✗	✓	✗	✓	✗	✓	✗

Tabulka 4.1: Porovnání nalezených řešení dle definovaných kritérií

Kapitola 5

Grafické podklady

Veškerý obrazový materiál byl vytvořen pomocí open source vektorového editoru Inkscape, z něž byly následně exportovány konečné rastrové obrázky ve formátu PNG. Použití rastrových obrázků vyžaduje prostředí GamLab, v němž byl výstupní program této práce implementován. V následujících několika odstavcích je stručně popsán princip vektorové grafiky a prostředí programu Inkscape.

5.1 Vektorová grafika

Existují dvě základní metody reprezentace obrazové informace v počítači: rastrová a vektorová. Rastrová grafika popisuje celý obrázek pomocí pole jednotlivých pixelů a jejich hodnot a je používána zejména ve světě fotografie, zatímco v grafice vektorové je obrázek složen ze základních geometrických útvarů jako jsou body, přímky, křivky a mnohoúhelníky, tedy objekty vyjádřitelné matematickou rovnicí. Jednou z hlavních výhod vektorové grafiky je možnost libovolně zvětšovat či zmenšovat obrázek bez ztráty jeho kvality. Využívá se především v počítačové sazbě, tvorbě diagramů atd.

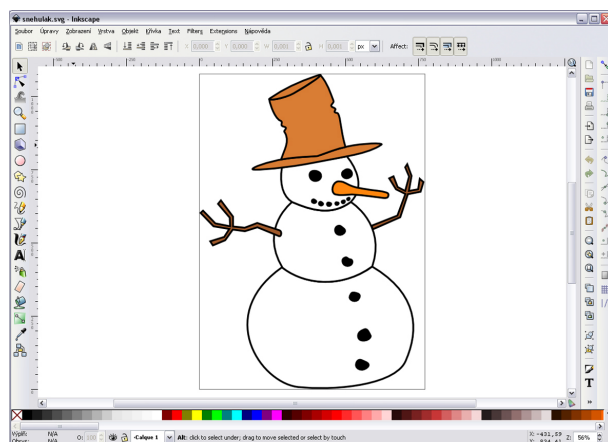
Scalable Vector Graphics (SVG)

Scalable Vector Graphics je formát založený na XML popisující dvojrozměrnou vektorovou grafiku. Tato specifikace je otevřeným standardem vyvíjeným World Wide Web Consortiumem (W3C). Podporuje křivky, základní tvary, text, animace aj.

5.2 Prostředí Inkscape

Inkscape vznikl v roce 2003 jako odnož projektu Sodipodi. Je to multiplatformní aplikace, která používá jako svůj nativní formát standardizovaný SVG. Jejím cílem je stát se grafickým nástrojem s plnohodnotnou podporou standardů SVG, XML a CSS. Poskytuje mnoho užitečných prvků pro práci s vektorovou grafikou, jako například vytváření objektů a manipulaci s nimi, nástroje pro práci s křivkami, s textem apod. Podporuje práci s vrstvami, editaci meta-dat, komplexní operace s křivkami, export do mnoha vektorových formátů a do bitmapového formátu PNG podporujícího alpha průhlednost a mnoho dalšího. V rámci

tvorby obrazové kolekce pro účely této práce bylo použito několik nástrojů, jejichž stručný popis následuje.



Obrázek 5.1: Prostředí editoru Inkscape

Tvorba objektů

Tvorba Béziových křivek a přímých čar Pomocí tohoto nástroje lze vytvářet Béziové křivky. Nástroj umožňuje přichycení křivek k přednastaveným úhlům. Takto vytvořené křivky lze následně upravovat pomocí jejich řídicích bodů.

Tvorba obdélníků a čtverců Nástroj umožňuje vytvářet obdélníky a čtverce, poskytuje také možnost upravovat takto vytvořeným objektům míru zaoblení rohů.

Tvorba kruhů, elips a oblouků Použitím tohoto nástroje lze vytvořit kruhy, elipsy a oblouky. Co se týče tvorby oblouků, dává Inkscape na výběr buď klasický oblouk nebo segment s dvěma poloměry.

Tvorba a úprava textových objektů Do dokumentu aplikace Inkscape je také možné vkládat textové objekty. Text lze upravovat klasickým způsobem, od velikosti přes font a barvu až po styl a zarovnání. Lze nastavit horní nebo dolní index, libovolné mezery mezi jednotlivými písmeny, slovy a řádky nebo vertikální orientaci textu.

Trasování bitmapy Jedním z důležitých nástrojů aplikace Inkscape je Trasování bitmapy. Pomocí tohoto nástroje lze převést rastrový obrázek do vektorů. Nástroj pracuje ve dvou režimech - jednoduchý sken a vícenásobné skenování. Skenování v jednoduchém režimu vytvoří jednu křivku. Tento režim poskytuje tři metody skenování:

- **Odřízutí jasu:** Vyhodnoceno je zastoupení červené, modré a zelené (popř. odstín šedi) v jednotlivém pixelu a na základě toho je pixel prezentován buď černou nebo bílou barvou.

- **Detekce hran:** Tato metoda používá algoritmus na detekování hran pro rychlé nalezení izochar stejného kontrastu.
- **Kvantizace barvy:** Vyhledávány jsou hrany, kde se mění barva.

Vícenásobné skenování vytváří skupinu křivek, ne jen jednu, jako u jednoduchého skenu. Bitmapa je skenována několikrát, pro každý sken je vytvořena jedna křivka. Počet skenování je možné upravit, přednastavená hodnota čítá osm skenování. Vícenásobné skenování poskytuje rovněž tři metody:

- **Kroky jasu:** Jednotlivé skeny jsou vytvářeny na základě jasového prahu, který je z každým skenem zvyšován.
- **Barvy:** Počet barev v obrázku je redukován na tolik, kolik je nastavený počet skenů. Použije se sada barev sestavená pomocí algoritmu kvantizace barev.
- **Šedé:** Pracuje stejně jako Barvy, ovšem výsledek je převeden do stupňů šedi.

Rozdíl mezi těmito dvěma režimy, jednoduchým a vícenásobným skenováním, je tedy v počtu výsledných křivek, s čímž úzce souvisí kvalita výsledného vektorového obrázku. Jednoduché skenování umožňuje vytvořit pouze černobílé obrázky, kdežto vícenásobné umí pracovat s barvami.

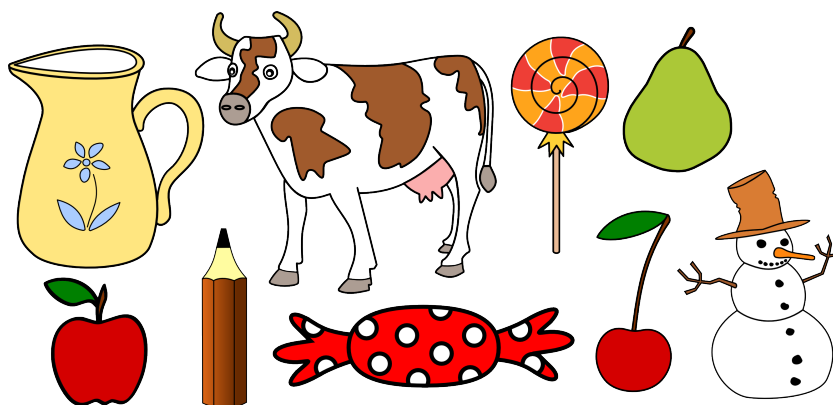
Tvorba stylů Inkscape umožňuje nastavit objektům barvu výplně (jednou barvou, přechodem či vzorem), styl, barvu a sílu obrysu s mnoha možnostmi jeho zakončení (šipka, čtvereček, tečka atd.), průhlednost a poskytuje také řadu přednastavených filtrů. Styl objektu lze také upravovat pomocí masek.

5.3 Vlastní tvorba grafických podkladů

Ve spolupráci s Mgr. Hanou Červenákovou, pedagožkou SPC, byla stanovena základní kritéria pro tvorbu grafických podkladů. Na základě těchto kritérií byla následně vytvořena kolekce obrázků. První část obrázků byla paní Červenákovou zkontrolována a následně námi upravena podle jejích připomínek. Všechny grafické materiály byly vytvořeny mj. z důvodu pozdější upravitelnosti v programu Inkscape, z něhož byly exportovány hotové rastrové obrázky ve formátu PNG. Všechny obrázky byly uloženy ve třech variantách - barevná, obrys a silueta. Tvorba všech těchto variant nalezne opodstatnění u jednotlivých typů her.

Kritéria pro tvorbu obrázků

- Silné kontury (alespoň 5 px), nejlépe černé (kontrastní k pozadí)
- Jednoduchost, vynechat detaily
- Barevné a kontrastní obrázky



Obrázek 5.2: Příklady vytvořených obrázků

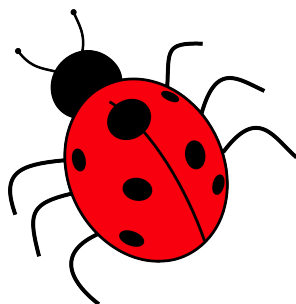
Část obrázků bylo vytvořeno podle mých vlastních nápadů, pro část byly použity předlohy, podle kterých byly obrázky namalovány. Předlohy byly čerpány ze serveru clker.com[7] (ve vektorové nebo rastrové formě) a z knihy Rozvoj zrakového vnímání pro děti od 4 do 6 let[3], jež je používána v SPC (pouze rastrové). Některé z obrázků byly podle předloh obkresleny a upraveny (zjednodušeny, pozměněny apod.), u některých rastrových byl využit nástroj Trasování bitmapy, výsledky byly opět upraveny do požadované podoby podle kritérií. Byla sestavena kolekce čítající přes osmdesát obrázků.

Specifické požadavky na organizaci grafických podkladů

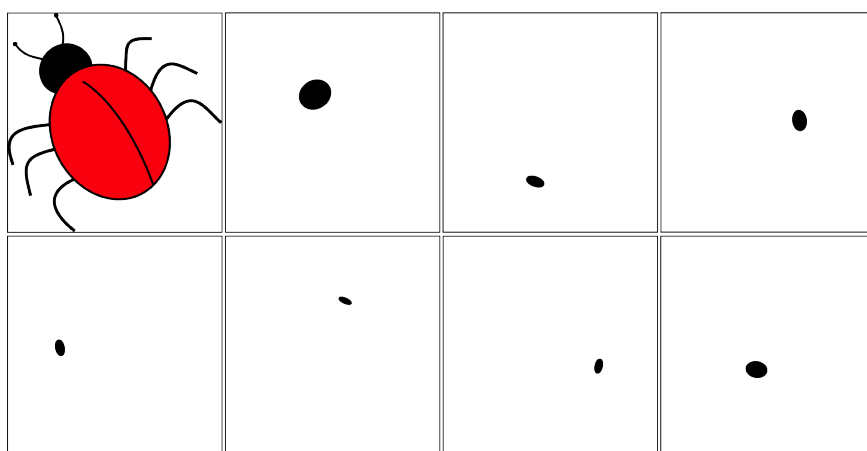
Řešení hry Najdi dvojici 1 vyžaduje, aby byly obrázky ukládány v určité formě. Týká se to názvu souboru a rozměrů obrázku. Každý obrázek je rozdělen na neměnný základ a volitelné detaily, viz Obrázek 5.3. Každá jednotlivá část tvoří jeden soubor o rozměrech základního obrázku. Toto je velmi důležité, neboť výsledný obrázek ve hře je vytvořen skládáním jednotlivých detailů na základ. Takto je uchována informace o pozici detailu na obrázku. V případě vektorového obrázku je možné detaily vytvořit v Inkscapu a to tak, že je v nastavení velikosti dokumentu vybráno „Resize page to drawing or selection”. Postupně je pak třeba umazávat části obrázku a exportovat zvlášť každý detail jako stránku (v dialogu exportu bitmapy je třeba vybrat možnost oblasti exportu „Stránka”). Uloží se tak ve velikosti původního obrázku. Pokud je k dispozici obrázek rastrový, je možné využít jakýkoliv grafický editor, který umí pracovat s průhledností, např. Gimp nebo Adobe Photoshop. V tomto případě je potřeba gumou umazat vše kromě každého jednoho detailu a ty pak následně uložit každý zvlášť. Toto rozdělení na části je třeba provést pro barevný obrázek i jeho obrysovou verzi.

Co se týče specifikace názvů, bylo zavedeno přiřazení „-0” pro základ a „-x” pro každý detail, kde x vyjadřuje číslo od jedné do počtu detailů. Tyto přípony je třeba vložit za název obrázku. Pro lepší představu zde uvedu příklad. Výchozím obrázkem je obrázek s názvem „beruska.png”. Barevný základ, tedy tělo berušky, je pojmenováno „beruska-0.png”, jednotlivé tečky pak nesou názvy „beruska-1.png”, „beruska-2.png” atd. To samé je třeba udělat pro obrys. Základ bude tedy „beruska-obrys-0.png” a jednotlivé tečky analogicky jako

u barevné verze. Je třeba dodržet stejné číslování barevných detailů s číslováním detailů obrysových.



(a) Původní obrázek



(b) Základ obrázku a jednotlivé detaily

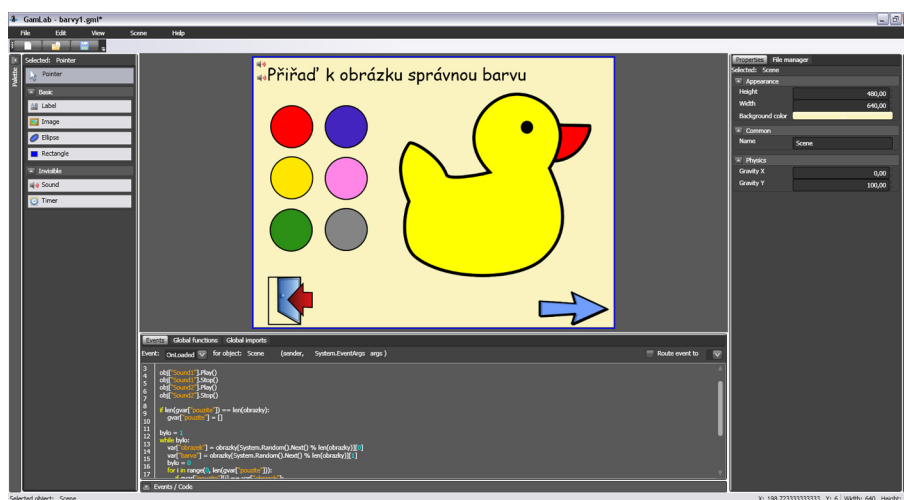
Obrázek 5.3: Příklad rozdělení obrázku beruska.png

Kapitola 6

Vývojové prostředí GamLab Studio

Pro účely této práce bylo ze strany fakulty předepsáno vývojové prostředí GamLab Studio. Toto prostředí bylo vyvinuto v rámci diplomové práce Ing. Tomáše Pastorka s názvem "Prostředí pro tvorbu jednoduchých her" podpořené Grantovou agenturou Českého vysokého učení technického v Praze. Bylo vytvořeno v jazyku C# pro platformu .NET, je tedy určeno výhradně pro použití v operačním systému Microsoft Windows.

GamLab Studio umožňuje tvořit jednoduché hry s 2D grafikou, základními fyzikálními simulacemi a zvuky. Aby bylo možné tyto zvuky přehrávat, je třeba mít nainstalovaný Windows Media Player 11 (Windows 7 jsou tímto programem již vybaveny).



Obrázek 6.1: Prostředí GamLab

Editor scény

Základním prvkem prostředí je editor scén, tedy plocha, do které je možné systémem Drag&Drop vkládat různé objekty, jako jsou obrázky, základní geometrické tvary (elipsy, obdélníky), zvuky, text apod. S těmito objekty se pak dají provádět různé operace, jako například mazání, kopírování nebo transformace. Kliknutím na objekt ve scéně se tento

stává aktivním, zobrazí se panel jeho vlastností a editor kódu jeho událostí. Editor scény se nachází na obr. 6.1 nahoře uprostřed.

Paleta objektů

Paleta objektů obsahuje všechny typy objektů, které prostředí GamLab poskytuje. Objekty jsou graficky rozděleny do dvou kategorií: základní a neviditelné. Mezi základní objekty patří textová popiska, obrázek, elipsa a obdélník. Zvuk a časovač pak patří do kategorie objektů neviditelných. Paleta objektů je umístěna na obr. 6.1 vlevo.

Panel vlastností

Každému objektu se dají nastavit různé vlastnosti a tím přizpůsobit jeho chování. Prostedí umožňuje simulovat nad objekty jednoduché fyzikální jevy. Pro účely těchto simulací mají základní objekty fyzikální vlastnosti, jako např. váha, odrazivost, tření apod. Všechny tyto vlastnosti se dají nastavit v panelu vlastností, obr. 6.1 vpravo, levá záložka.

Správce souborů

Do každé vytvářené aplikace je možné naimportovat různé typy souborů, konkrétně obrázky a zvuky. GamLab podporuje pouze rastrové obrázky, podpora vektorové grafiky zatím není implementovaná. Obrázkové formáty, které se dají v GamLabu použít, jsou BMP, JPG a také PNG a GIF, které podporují průhlednost. Ze zvukových formátů jsou to pak MP3, WAV a WMA. Při importu obrázků a zvuků se jejich kopie uloží do hlavního souboru aplikace, nevytváří se odkaz do souborového systému. Panel Správce souborů je v pravé záložce pravého panelu.

Editor kódu

Aplikace vytvořená v tomto prostředí přijímá vstup od uživatele a na jeho základě generuje události (např. kliknutí na objekt, stlačení klávesy atd.). Pomocí editoru kódu lze k jednotlivým událostem napsat obslužné rutiny ve skriptovacím jazyce Boo, které jsou volány vždy při vygenerování příslušné události. Editor se nachází na obr. 6.1 uprostřed dole.

Kapitola 7

Vlastní řešení

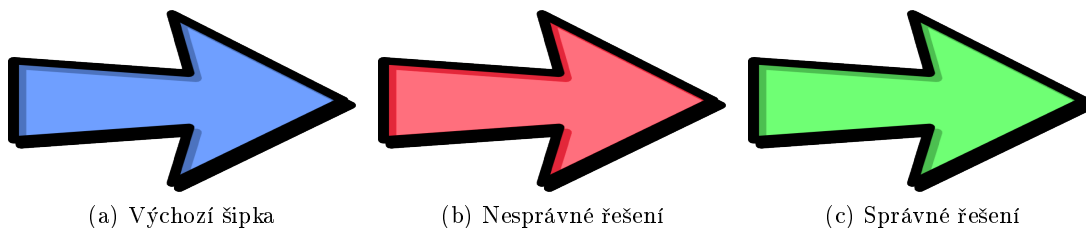
V rámci této práce byly vytvořeny čtyři hry - Matematika 1, Matematika 2, Barvy a Najdi dvojici 1 - a hlavní menu.

7.1 Hlavní menu

Po spuštění programu je načtena obrazovka s hlavním menu, které slouží jako navigace celým programem, a přehrána doprovodná melodie[5]. V horní části obrazovky se nachází uvítací hláška, pod kterou je umístěno pět obrázků s titulkem. Každý titulek odpovídá jednomu konkrétnímu typu úloh: Poznávačky, Hry s barvami, Hry s čísly, Logické hry a Doplnovačky. Kromě položky Poznávačky vyvolá kliknutí na typ hry načtení příslušné úlohy. Po kliknutí na Poznávačky se otevře podmenu, kde je možné vybrat ze tří variant: Levo-pravé/horno-dolní, Najdi obrys/siluetu, Najdi dvojici. Kliknutím na jednotlivý typ hry je spuštěna příslušná scéna. Po kliknutí na obrázek dveří v levém dolním rohu v podmenu Poznávaček je načteno hlavní menu, kliknutím na dveře v hlavním menu je program ukončen.

7.2 Ovládání

Jedním z požadavků na program bylo stejné navigační ovládání her. Ve spodní části obrazovky každé hry se tedy nachází dva obrázky - ovládací prvky: šipka, zajišťující kontrolu správnosti úlohy a předchod na novou úlohu ve stejné kategorii, a dveře - návrat do hlavního menu (v případě her v kategorii Poznávačky vrací dveře do podmenu). Kontrola úlohy je provedena po kliknutí na šipku. V případě, že je úloha vyřešena špatně, je přehrán nepříjemný zvuk[4] a úloha je resetována. Výjimku tvoří hra Barvy, kde je přehrán pouze nepříjemný zvuk. Při správném řešení se přehraje zvuk veselý[9] a je načtena nová úloha. Obrázek šipky byl vytvořen ve třech variantách - modré, zelené a červené, viz Obrázek 7.1. Modrá varianta šipky je výchozí. Po kliknutí na šipku se mění její barva. Šipka zčervená v případě, že je úloha vyřešena nesprávně. Pokud však hráč vyřešil úlohu správně, šipka zezelená. Následně je zobrazena další úloha.



Obrázek 7.1: Tři varianty šipky

7.3 Matematika 1

Popis úlohy

Cílem hry Matematika 1 je vybarvit zadaný počet obrysů. Obrazovka hry je pomyslně rozdělena do dvou částí: v levé je umístěno číslo udávající počet obrysů, které mají být vybarveny, vpravo se pak nachází 8 obrysů. Kliknutím na obrys se tento vybarví, opětovným kliknutím se opět zprůhlední. Lze vybarvit i více obrysů, než je zadané číslo, hráč tedy musí správně spočítat, kolik jich vybarvil. Rozložení obrazovky viz Obrázek 7.2.

Popis řešení

Základním stavebním kamenem této hry je množina dvojic obrázků. Dvojice je tvořena barevným obrázkem a jeho obrysem. Konkrétní obrázek, který je pak ve scéně použit, je vybrán z množiny náhodně. Použitím dlaždičkovacího algoritmu, Ukázka kódu 1, je vyplněna plocha osmi obrysy tak, aby byla obrázky co nejvíce pokryta. Algoritmus rozděluje obrázky na základě jejich rozměrů do tří kategorií: přibližný čtverec, obrázek na šířku, obrázek na výšku. Podle kategorie vypočítá algoritmus finální velikost obrázku a určí vhodné rozložení. Vygeneruje se náhodné číslo v zadaném intervalu, které udává počet obrázků k vybarvení, a zobrazí se na scéně. V obsluze události onClick je zkontrolován stav obrázku a provedena jeho výměna vždy za druhou možnost (barevný - obrys). Na základě tohoto se upraví vnitřní čítač. Při vyhodnocení je porovnávána hodnota čítače s uloženým náhodně generovaným číslem.

7.4 Matematika 2

Popis úlohy

Úkolem této hry je shodit ze stromu zadaný počet kusů ovoce. Na obrazovce se nachází strom, na kterém visí 7 kusů ovoce. Kliknutím na ovoce vybraný kus za zvukového doprovodu spadne. Lze shodit i více kusů ovoce, než udává zadané číslo.



(a) Obrazovka hry

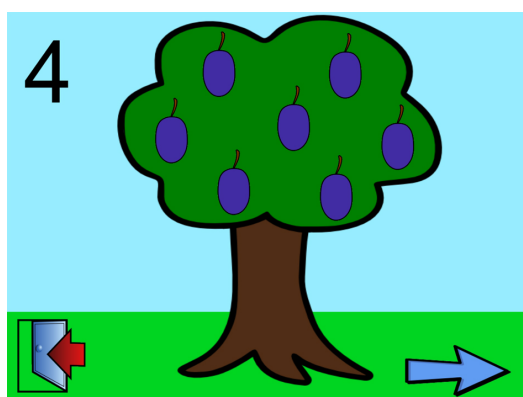


(b) Nesprávné řešení

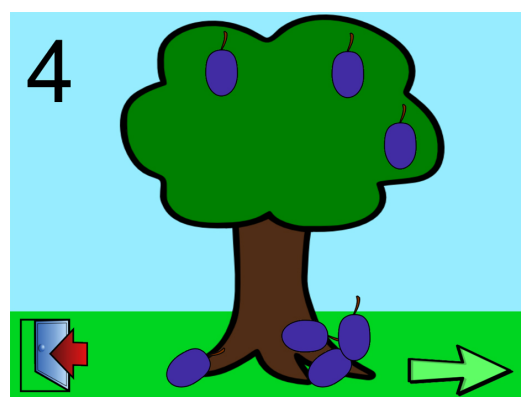


(c) Správné řešení

Obrázek 7.2: Snímky obrazovky ze hry Matematika 1



(a) Obrazovka hry



(b) Správné řešení

Obrázek 7.3: Snímky obrazovky ze hry Matematika 2

Popis řešení

Scéna této hry je tvořena statickým obrázkem stromu a dvěma obdélníky, tvořícími trávu.

Ukázka kódu 1 Dlaždičkovací algoritmus

```

if ratio > 0.8 and ratio < 1.8:
    vRadku = 3
    vyska = (obj["Scene"].Height - yoff - yoff2 - mezera*2) / 3
    sirka = vyska * ratio
    if sirka > ((obj["Scene"].Width - xoff - xoff2 - mezera*2) / 3):
        sirka = ((obj["Scene"].Width - xoff - xoff2 - mezera*2) / 3)
        vyska = sirka / ratio
    xmezera = (obj["Scene"].Width - xoff - xoff2 - sirka*3) / 2
    ymezera = (obj["Scene"].Height - yoff - yoff2 - vyska*3) / 2
// obrazek na vysku
elif ratio < 1:
    vRadku = 4
    vyska = (obj["Scene"].Height - yoff - yoff2 - mezera) / 2
    sirka = vyska * ratio
    if sirka > ((obj["Scene"].Width - xoff - xoff2 - mezera*3) / 4):
        sirka = ((obj["Scene"].Width - xoff - xoff2 - mezera*3) / 4)
        vyska = sirka / ratio
    xmezera = (obj["Scene"].Width - xoff - xoff2 - sirka*4) / 3
    ymezera = mezera
// obrazek na sirku
else:
    vRadku = 2
    sirka = (obj["Scene"].Width - xoff - xoff2 - mezera) / 2
    vyska = sirka/ratio
    if vyska > ((obj["Scene"].Height - yoff - yoff2 - mezera*3) / 4):
        vyska = (obj["Scene"].Height - yoff - yoff2 - mezera*3) / 4
        sirka = vyska * ratio
    xmezera = mezera
    ymezera = (obj["Scene"].Height - yoff - yoff2 - vyska*4) / 3

```

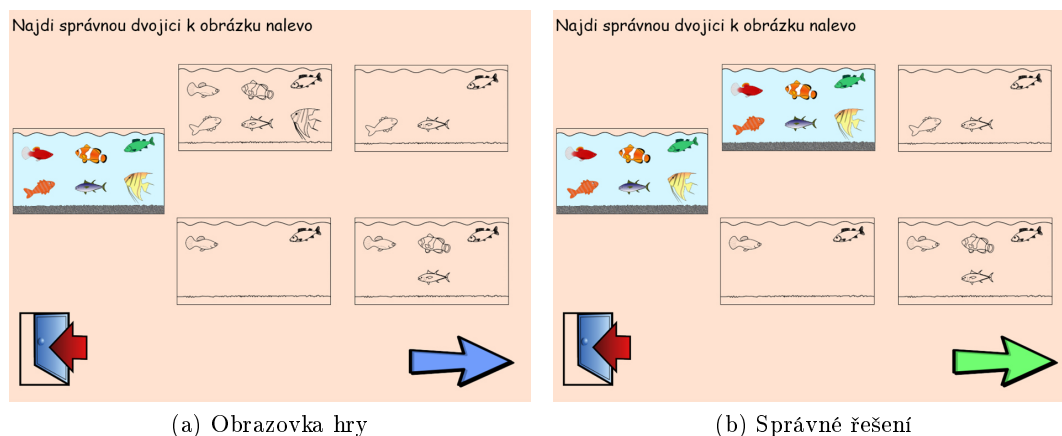
V oblasti koruny stromu jsou pak umístěny objekty typu Image, do kterých je náhodně vybírán konkrétní druh ovoce. V levém horním rohu se nachází náhodně generované číslo udávající počet kusů ovoce ke shození. Po kliknutí na jednotlivé obrázky ovoce se upraví vnitřní čítač, obrázkům je nastaven StaticState na false, čímž se spustí fyzikální simulace pádu, a přehraje se zvuk padajícího ovoce. Tráva je tvořena dvěma na sebe navazujícími obdélníky, z nichž vrchní má vlastnost CollisionEnabled nastavenou na false, tedy nekoliduje s žádnými objekty. To, že ovoce prvním obdélníkem "propadne", vytváří prostorový efekt scény.

7.5 Najdi dvojici 1

Popis úlohy

V této hře má hráč za úkol poznat obrázek shodný se vzorem. V levé části obrazovky se nachází vzorový barevný obrázek, v pravé části jsou pak umístěny čtyři obrysy téhož obrázku, tři z nich ovšem s drobnými změnami. Kliknutím na nesprávný obrys je přehrán nepříjemný zvuk, kliknutím na správný je tento vybarven.

Popis řešení



(a) Obrazovka hry

(b) Správné řešení

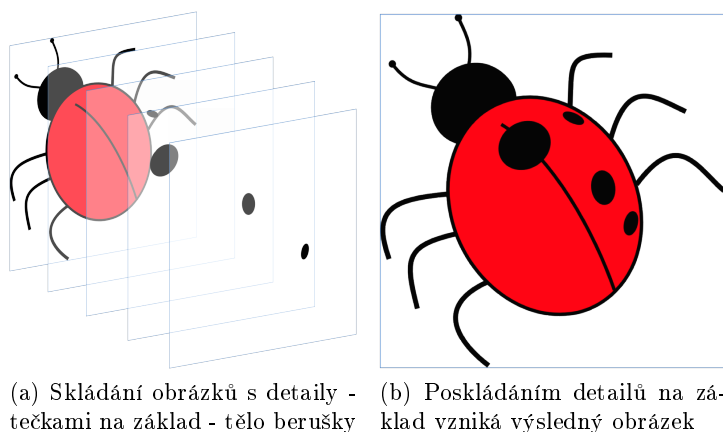
Obrázek 7.4: Snímky obrazovky ze hry Najdi dvojici 1

Řešení je založené na náhodném generování různých variant jednoho obrázku. Obrázky pro tuto hru jsou speciálně připravené, podrobně popsáno v kapitole 5.3. Pro každý obrázek existuje několik souborů s jeho částmi. Hlavní soubor obsahuje základní neměnnou část obrázku, ostatní soubory pak představují jednotlivé volitelné detaily, viz Obrázek 5.3b. Tyto soubory musejí mít stejné rozměry jako základní obrázek a jejich pozadí musí být transparentní. Všechny obrázky existují v barevné a obrysové variantě. Po spuštění scény se prohledají všechny přiložené soubory a vytvoří se kolekce obrázků. Každému obrázku je nalezena množina jeho detailů. Následně se vybere náhodný obrázek z kolekce, ke kterému se vygenerují příslušné čtyři různé podmnožiny detailů. Pomocí vrstvení detailů s průhledným pozadím se do scény vloží čtyři výsledné varianty obrázku v obrysové podobě. Ze čtyř variant je náhodně jedna vybrána a použita v barevné podobě jako vzorový obrázek.

Uvedu zde příklad na obrázku beruska.png, obr. 5.3a. Obrázky je třeba uložit podle návodu v kapitole 5.3 a všechny je nahrát do hry pomocí Správce souborů, kapitola 6. Program pak funguje následovně:

1. Prohledají se přiložené soubory, vytvoří se kolekce obrázků, která obsahuje mj. i položku beruska, ke každému obrázku se zároveň uloží počet jeho detailů.
2. Náhodně se vybere jeden obrázek z kolekce. Pro tento příklad předpokládejme, že je to právě beruska.
3. Připraví se pole `varianty[]` pro ukládání jednotlivých sad detailů.
4. Pomocí funkce `Varianta()`, viz Ukázka kódu 2, se náhodně vygeneruje podmnožina množiny detailů berušky - sada čísel od 1 do počtu detailů.
5. Funkce `JeUnikatni()`, Ukázka kódu 3, následně zkontroluje, jestli je varianta unikátní (porovná s obsahem pole `varianty[]`). Pokud není, proběhne znovu předchozí bod, pokud je, uloží se do pole `varianty[]`.
6. Z pole `varianty[]`, které teď obsahuje čtyři různé sady detailů, se jednotlivé sady přiřadí k jednotlivým objektům typu `Image` na scéně.

7. Vygeneruje se náhodné číslo od 1 do 4, které určí, která z variant se použije pro vzorový obrázek.
8. Pro každou variantu se vyberou takové obrázky detailů, jejichž přípony názvů odpovídají číslům obsaženým ve variantě.
9. Pro každou variantu je určena pozice ve scéně, na kterou se umístí obrázek základního tvaru (těla berušky) a následně postupně všechny obrázky detailu, Obrázek 7.5a. Tím se dosáhne efektu navrstvení obrázků a výsledek se pak jeví jako jediný obrázek, Obrázek 7.5b.



Obrázek 7.5: Vytvoření obrázku z vygenerované varianty

Ukázka kódu 2 Funkce generuje sadu detailů - jednu variantu

```
def Varianta(pocetCasti):
    varianta = []
    rand = System.Random().Next() % (pocetCasti + 1)
    for j in range(0, rand):
        bylo = 1
        while bylo:
            castecka = (System.Random().Next() % pocetCasti) + 1
            bylo = 0
            for k in range(0, j):
                if castecka == varianta[k]:
                    bylo = 1
                    break
            varianta.Add(castecka)
    return varianta.Sort()
```

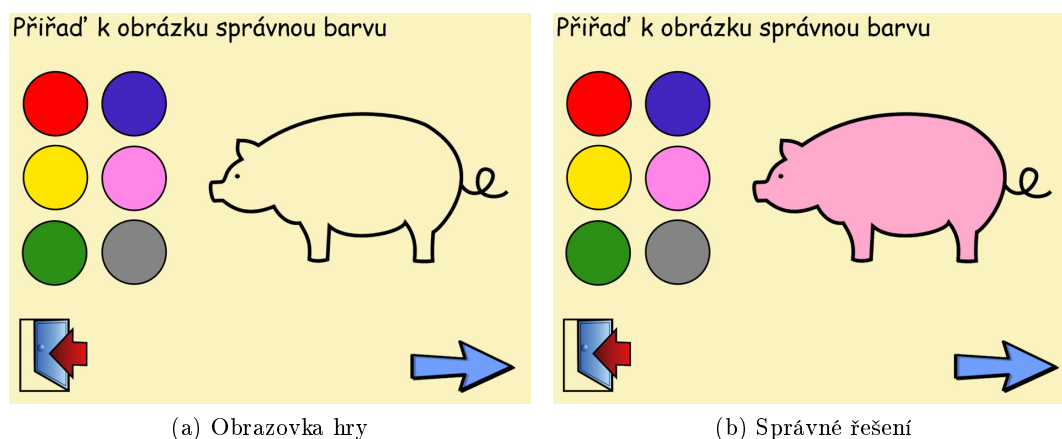
Ukázka kódu 3 Funkce kontrolující jedinečnost varianty ve hře Najdi dvojici 1

```
def JeUnikatni(varianty, varianta):
    for i in range(0, len(varianty)):
        if len(varianta) != len(varianty[i]):
            continue
        ruzne = 0
        for j in range(0, len(varianta)):
            if varianta[j] != varianty[i][j]:
                ruzne = 1
                break
        if ruzne == 0:
            return false
    return true
```

7.6 Barvy

Popis úlohy

Podstatou hry Barvy je přiřazování správné barvy (podle reálného světa) k obrázku. Na obrazovce je na výběr několik barev a vzorový obrys. Ten je třeba vybarvit kliknutím na správnou barvu. Princip ovládání je obdobný jako u výše zmíněných her.



Obrázek 7.6: Snímky obrazovky ze hry Barvy

Popis řešení

Obrázkové podklady jsou po spuštění scény načteny vždy jako dvojice obrázek - barva. V levé části scény je umístěna paleta barev - objekty typu Ellipse, každý je vyplněn jednou barvou. Vzorový obrázek je náhodně vybírán ze sady možností, avšak pro zajištění diverzity úloh se do globální proměnné ukládají již použité obrázky. Po vyčerpání všech možností je tato proměnná vymazána, Ukázka kódu 4. Výběr barvy se provádí kliknutím na elipsu příslušné barvy. Zde také následně proběhne vyhodnocení a v případě správného řešení je obrys vzorového obrázku nahrazen jeho barevnou variantou. V opačném případě je po kliknutí pouze přehráván negativní zvuk. Šipka opět zajišťuje v případě úspěchu přechod na další scénu, dveře návrat do hlavního menu.

Ukázka kódu 4 Tento kus kódu zajišťuje, aby se obrázky ve hře Barvy neopakovaly.

```
if len(gvar["pouziteBar"]) == len(obrazky):  
    gvar["pouziteBar"] = []  
  
bylo = 1  
while bylo:  
    nahoda = System.Random().Next() % len(obrazky)  
    var["obrazek"] = obrazky[nahoda][0]  
    var["barva"] = obrazky[nahoda][1]  
    if not gvar["pouziteBar"].Contains(var["obrazek"]):  
        bylo = 0  
gvar["pouziteBar"].Add(var["obrazek"])
```

Kapitola 8

Testování

V rámci vytvoření bezchybného a co nejlépe vyhovujícího programu byly naplánovány a následně provedeny uživatelské testy ve Speciálně pedagogickém centru. Otestováno bylo pět funkčních prototypů her, a to hry Matematika 1, Matematika 2, Barvy, Najdi siluetu a Najdi dvojici 1. Tato kapitola popisuje přípravu a průběh testování, v závěru je pak diskuze možných řešení nalezených nedostatků/chyb.

8.1 Příprava testu

Na základě požadavku SPC bylo nutné sepsat žádost o souhlas s účastí dítěte na tomto uživatelském testování pro rodiče dětí z MŠ Horáčkova. Bylo vytištěno 14 kopií, podepsány byly následně všechny.

Pro účely samotného testování byl sestaven testovací protokol, který měl posloužit jako vodítko při hledání a analyzování nedostatků a chyb softwaru. Protokol zohledňuje veškeré problémy, které se mohly vyskytnout. Zároveň obsahuje bod, který se věnuje zpětné vazbě od dětí, které se zúčastnily testu, a od pedagožky, jenž celému testování supervizovala. Pro účastníky testu byla v závěru připravena malá sladká odměna jako poděkování za pomoc. Následuje zmíněný testovací protokol.

Testovací protokol

- Programové chyby
 - Pády
 - Zaseknutí
 - Nestabilní chování
 - Možné problémy s kompatibilitou
 - Funkční chyby/nedostatky
- Ošetření nepovoleného uživatelského vstupu
- Nedostatky uživatelského prostředí

- Nevhodně zvolené ovládací prvky
- Nejasné ovládání
- Grafika
 - Grafická přitažlivost
 - Vhodnost obrazového materiálu
- Zpětná vazba
 - Děti
 - * Oblíbenost jednotlivých her
 - * Návrhy na vylepšení
 - Pedagog
 - * Návrhy na vylepšení
 - * Ostatní kritika

8.2 Průběh testování

Testy byly prováděny jednak na notebooku (AMD Turion 64 Mobile, 2GB RAM, NVidia GeForce Go 7300, Windows 7), jednak v prostorách SPC na místním počítači (Intel Pentium 4, 1 GB RAM, integrovaná grafická karta Intel, Windows XP Professional Service Pack 3). Z důvodu správné funkčnosti doprovodných zvuků v programu bylo nutné nainstalovat Windows Media Player 11. Pro testování nakonec mateřská škola uvolnila 7 dětí ve věku od tří do šesti let s různým typem a různou četností zrakových postižení. Z důvodu nízkého věku účastníků nebyl pro samotné testy připraven pevný scénář. Každé dítě si tedy mohlo vždy vybrat pořadí her, ve kterém je bude hrát. Délka hraní jednotlivých her pak záležela na tom, do jaké míry se dítěti hra líbila. Z vytvořených her byly v rámci tohoto testování k dispozici tyto hry: Matematika 1, Matematika 2, Barvy 1, Najdi dvojici 1 a Najdi siluetu.

Následuje seznam dětí, které se zúčastnily testování. Ve výše zmiňované žádosti pro rodiče bylo přislíbeno, že v rámci zpracování a zveřejnění výsledků testu nebudou uváděna žádná jména zúčastněných. Proto uvádím pouze pohlaví, věk a zrakové i jiné postižení.

Seznam účastníků:

1. Holčička, 5 let, sekundární glaucom (vysoký nitrooční tlak ve sklivci), mikroftalmos (malé oko), šedý zákal, chronický zánět oka
2. Chlapec, 6 let, porucha autistického spektra, pásmo slabozrakosti
3. Chlapec, 4 roky, šilhavost, astigmatismus -> problém s jemnou motorikou
4. Chlapec, 5 let, dalekozrakost, nystagmus (nepravidelné nekontrolované pohyby oka)
5. Holčička, 5 let, krátkozrakost
6. Chlapec, 3 roky, tupozrakost
7. Holčička, 5 let, tupozrakost

8.3 Výsledky testování

Provedené testy přinesly mnoho cenných poznatků, od funkčních chyb programu přes věcné připomínky supervizorky až k vyjádření samotných účastníků testu. Tyto poznatky jsou v následující části uvedeny podle struktury výše zmíněného testovacího protokolu.

Programové chyby

Zasekávání programu

Program tzv. "zamrznul". Stávalo se tak buď při přechodu na novou scénu nebo po nedokončeném načtení scény. V prvním případě byl úkol na aktuální scéně splněn a při přechodu na další scénu kliknutím na příslušnou šipku program tuto očekávanou operaci neprovedl a přestal reagovat na jakýkoliv uživatelský vstup. Musel být externě uzavřen a spuštěn znovu. Ve druhém případě se po kliknutí na šipku sice přechod na další scénu provedl, nicméně scéna se načetla jen částečně. V případě hry Barvu 1 se scéna načetla celá kromě hlavního obrázku (zobrazil se pouze objekt Image se zástupným obrázkem "No Image"), na další podněty uživatele již program neodpovídal, musel být uzavřen a opětovně spuštěn. V případě obou Matematických her a Poznáváčky se z nové scény načetl pouze nadpis a barva pozadí, tedy statické objekty (negerované). U hry Najdi siluetu podobné problémy nebyly zjištěny.

Možné problémy s kompatibilitou

V prostředí Windows 7 vykazuje program chybu, která se při spuštění ve Windows XP neprojeví. Jedná se o chybu "Nelze ovládat média, dokud není LoadedBehavior nebo UnloadedBehavior nastaveno na Manual." Tato chyba způsobí, že na objekty typu Sound nelze zavolat metodu Play(), Pause() nebo Stop(). Pravděpodobně se jedná o chybu v samotném prostředí GamLab.

Funkční nedostatky - časté opakování obrázků a čísel

Stávalo se, že se náhodným výběrem obrázků a náhodným generováním čísel hodnoty v obou případech často opakovaly. Dítě mělo tak za úkol několikrát za sebou například vybarvit tři zmrzliny. Tím hra pozbývala na různorodosti a zajímavosti. Je tedy potřeba zajistit, aby se za sebou jdoucí hodnoty lišily.

Ošetření nepovoleného uživatelského vstupu

Byly objeveny dvě situace, kdy je uživateli povoleno využívat program nesprávným způsobem. Ta první se týká hry Najdi siluetu, druhá pak hry Matematika 2. U hry Najdi siluetu je implementován čítač skóre, který započítává každou správně vyřešenou úlohu. Hodnota čítače se inkrementuje vždy po kliknutí na šipku (přechod na další úlohu), kdy se vyhodnotí, je-li současná úloha správně vyřešena. Problémem vyvstává v případě, že je na tuto šipku kliknuto vícekrát po sobě. Kliknutí na šipku vyvolá kontrolu správnosti řešení úlohy, případné zvýšení skóre, spuštění doprovodného zvuku a časovače, který v rámci své metody OnTick načte novou úlohu. Rychlým klikáním se tedy časovač spouští rychle za sebou, aniž

stihne "tiknout", nová úloha tedy není načtena. Čítač zaznamená tolik správných řešení, kolikrát bylo kliknuto, a není tak objektivní.

Co se týče chyby ve hře Matematika 2, jedná se zde o podobný problém jako v předchozím případě. Cílem hry je shodit ze stromu tolik kusů ovoce, kolik udává číslo v levé části obrazovky. Kliknutím na kontrolní šipku se vyhodnotí správnost úlohy a v pozitivním případě se načte nová scéna. Problém se zde vyskytuje u počítání shozeného ovoce. V programu totiž není ošetřena možnost, kdy dítě klikne na již shozené ovoce. V současné době se toto kliknutí započítá jako další shozený kus, výsledek tedy neodpovídá realitě a úloha je vyhodnocena nesprávně.

Nedostatky uživatelského prostředí

Ukazatel myši

Ukázalo se, že pro snazší orientaci dětí na obrazovce a jejich lepší vnímání ukazatele myši, je potřeba aplikovat dvě základní změny. První se týká vzhledu a velikosti kurzoru. Přednastavený ukazatel prostředí Windows XP je příliš malý a v porovnání s převážně světlým pozadím programu nedostatečně kontrastní. Druhým požadavkem je pak sledování stopy kurzoru, což dětem práci s myší velmi usnadňuje. Dále se u některých dětí vyskytl problém s rychlostí ukazatele. Některé děti měly problém s jemnou motorikou a manipulace s myší pro ně byla obtížná. V důsledku toho se pak stávalo, že dětem kurzor při trhaných rychlých pohybech nekontrolovaně "lítal" po obrazovce a děti byly naprosto dezorientované.

Poslední zjištěná možnost na vylepšení se týkala spíše funkčnosti ukazatele myši a byla inspirována řešením úlohy Najdi siluetu. Zde se kurzor při najetí na interaktivní prvek změnil ze šipky na velký, jinak barevný obrázek ruky, podobně jako při najetí na odkaz v běžných prohlížečích. Tato vlastnost kurzoru byla vyhodnocena jako velmi prospěšná pro orientaci dítěte v ovládání a vyvstal návrh řešit tímto způsobem znázornění interakce ve všech hrách.

Přesun na další úlohu - šipka

Ve stávající podobě her umožňuje šipka přesun na další úlohu pouze v případě, že je aktuální úloha správně vyřešena. Není tedy možné žádnou úlohu přeskóčit bez korektního vypracování. Byl vyřčen požadavek na úpravu funkce šipky takovou, aby bylo možné v případě potřeby aktuální úlohu přeskóčit.

Zadání úlohy ve hře Najdi siluetu

Ve hře Najdi siluetu se nahodile střídá hledání správné siluety a obrysu k hlavnímu obrázku. Slovní zadání je ovšem statické a nemění se podle příslušného typu úlohy.

Oddělení vzorového obrázku ve hře Najdi siluetu

Vzhledem k tomu, že děti s binokulárními zrakovými poruchami mají potíže s rozeznáváním tvarově podobných obrázků, tím spíš, pokud jsou uspořádány do řady, měli některé děti s těmito poruchami problém rozpoznat vzorový obrázek od nabídnutých možností. Pro zajištění jejich lepší orientace na obrazovce je třeba vzorový obrázek nějakým způsobem zvýraznit, například oddělit jej od ostatních výraznou čarou nebo jej uzavřít do výrazného ohraničení.

Grafika

Stávající grafika se ukázala být vyhovující. Nebyl objeven žádný nedostatek. Reakce dětí však v několika případech naznačovala, že některé úlohy nejsou příliš zábavné. Jednalo se hlavně o hru Matematika 1, která je ve své podstatě statická. Bylo navrženo obohatit grafiku o dynamiku (animace apod.).

Zpětná vazba

Zpětná vazba od dohlízející pedagožky je promítnuta do většiny výše zmíněných požadavků.

Dětem byly po provedení testu položeny tři otázky:

1. Jak se ti hry líbily? Která hra se ti líbila nejvíce/nejméně?
2. Která hra ti přišla nejjednodušší/nejtěžší?
3. Máš nějaký nápad, jak hry vylepšit?

Na tyto otázky odpovídaly děti velmi stručně, ke druhé otázce se všechny děti vyjádřily tak, že jim náročnost her připadala u všech stejná. Odpovědi na ostatní otázky jsou vypsány v tabulce 8.1.

Dítě	Oblíbená hra	Neoblíbená hra	Nápad na vylepšení
Dítě 1.	není	není	není
Dítě 2.	Najdi dvojici 1	Matematika 1	není
Dítě 3.	není	není	není
Dítě 4.	Barvy	není	U hry Barvy by si rád vybarvoval každou část obrázku zvlášť
Dítě 5.	není	není	není
Dítě 6.	Matematika 1 i 2, Barvy	není	není
Dítě 7.	není	není	není

Tabulka 8.1: Tabulka odpovědí dětí.

8.4 Diskuze řešení

Uživatelské studie odhalily několik nedostatků a zároveň přinesly mnoho nových návrhů. Mnohé z nich jsou velmi snadno zapracovatelné do stávajícího řešení, avšak některé vyžadují hlubší přepracování aplikací a několik z nich dokonce přesahuje rámec možností poskytovaných prostředím GamLab. Z toho důvodu bylo rozhodnuto některé připomínky a návrhy zapracovat a jiné nikoliv.

Zapracované připomínky a návrhy

Programové chyby

Co se týče problému s nestabilitou programu, byly změněny některé části kódu, které by mohly být příčinou tohoto chování. Po těchto úpravách se problém znovu nevyskytl.

Problém s kompatibilitou se nepodařilo při tvorbě her obejít, aniž by byla omezena funkcionalita.

V případech opakujících se obrázků a čísel se nabízejí tři možnosti řešení. Tou první je ukládat již použité obrázky nebo čísla do pole a po vybrání dalšího náhodného prvku zkontrolovat, zda-li se již v poli nenachází. Po zaplnění pole všemi dostupnými prvky by byl jeho obsah vymazán a pole by bylo opět připraveno k použití. U tohoto řešení se však stále vyskytuje možnost, že poslední použitý prvek by mohl být shodný s prvním prvkem nové sady. Toto by se však také dalo ošetřit. Druhá možnost, jak celý tento problém eliminovat, je nahodile vybranou sadu obrázků vygenerovat předem a tuto pak opětovně používat. Třetí způsob by byl řešen znovu pomocí pole, ovšem ukládat by se do něj mohlo jen posledních několik prvků. Pro střídání obrázků byla zvolena pro všechny hry první varianta, neboť zaručuje největší různorodost obrázků. Nemůže se totiž stát, že by se některý obrázek náhodným výběrem nepoužil, zároveň ale nevzniká zapamatovatelné pořadí po sobě jdoucích obrázků. V rámci zajištění neopakování čísel u početních her byla použita třetí z navrhovaných variant, tedy ukládání posledních několika čísel.

Ošetření nepovoleného uživatelského vstupu

Problém s čítačem skóre ve hře Najdi siluetu řeší kolega Bohumil Tománek, příslušná diskuze řešení zde tedy nebude podrobněji rozvinuta. Řešením by mohlo být vytvoření booleovské proměnné, jejíž hodnota se po kliknutí na šipku při správném vyřešení problému změní, opětovné klikání na šipku již nebude bráno v potaz. Při načtení nové scény se hodnota této proměnné vrátí opět do původního stavu.

Chyba ve hře Matematika 2, kde se vnitřní čítač inkrementuje i po kliknutí na již shozené ovoce, je poměrně triviálním problémem a byl vyřešen malou úpravou obslužné rutiny události. Bylo třeba doplnit jednu podmínku.

Nedostatky uživatelského prostředí

Všechny nedostatky uživatelského rozhraní byly vyřešeny pouze drobnými úpravami kódu.

Nezpracované připomínky

Požadavek na přitažlivější grafiku, tzn. jednoduché animování některých scén nebyl vyřešen, protože prostředí GamLab neumožňuje kromě jednoduchých fyzikálních simulací vytvářet animace. Dalším důvodem je nemožnost použít stejné animace pro obrázky různého typu. Není například možné animaci "příchod zvířátka na scénu" použít na obrázky náradí. Pro každý obrázek, nebo každou malou skupinu obrázků by proto musela existovat samostatná animace, co by výrazně zkomplikovalo rozšiřitelnost hry o nové obrázky.

Kapitola 9

Diskuze

Stěžejní část této práce souvisela s vývojem her v prostředí GamLab Studio, podrobněji o něm pojednává Kapitola 6. Práce v tomto prostředí přinesla žádané výsledky, nicméně bylo objeveno několik jeho nedostatků. V průběhu práce vyvstalo také několik návrhů na užitečná vylepšení. Tyto nedostatky a podněty jsou podrobněji popsány v následujícím textu. V závěru kapitoly jsou následně shrnuty nabyté zkušenosti s prostředím GamLab a diskutována vhodnost jeho použití v dalších projektech.

Nedostatky prostředí Gamlab

Editor kódu

Kód, který je zapsán do editoru kódu bez toho, aniž by byla vybrána událost, je po výběru konkrétní události nenávratně ztracen. Toto je velmi nepříjemná záležitost, mnohokrát nastala situace, že jsem omylem nevybrala událost a následně jsem pak přišla o napsaný kód.

Kolizní schránka

V dokumentaci GamLabu je uvedeno, že po přidání obrázku do scény je mu automaticky vytvořena tzv. polygonální kolizní obálka. Pro účely srážek by se obálka měla chovat jako okraj kresby obrázku. Toto však nefungovalo, objekty se srážely okraji celého obrázku, což při srážce nevytvořilo žádaný vizuální efekt - obrázky se nedotkly.

Nežádaná změna aktuálního adresáře

Po přidání souboru z jiného než aktuálního adresáře prostřednictvím Správce souborů dochází ke změně aktuálního adresáře. V důsledku toho přestane fungovat načítání scén ze souboru, neboť soubory jsou načítány vždy z aktuálního adresáře.

Nepřijímání souboru .gml jako parametru gamlab.exe

Program GamLab.exe nepřijímá soubor jako parametr příkazové řádky, proto ve Windows nefunguje otevírání .gml souborů dvojitém kliknutím myši.

Pád GamLabu po přenesení objektu z jedné instance GamLabu do druhé

Prostředí umožňuje copy-paste přenášení objektů mezi instancemi GamLabu. U objektu typu Image však dochází k pádu cílové instance v případě, že v ní přenášený obrázek nebyl předem přidán pomocí Správce souborů. Nejlepší by bylo v takovém případě automaticky

přenést soubor do cílové instance zároveň s objektem. Minimálně by však mělo být studio schopné tuto situaci ošetřit a reagovat upozorněním pro uživatele, aplikace by neměla havarovat.

Poškozený konfigurační soubor

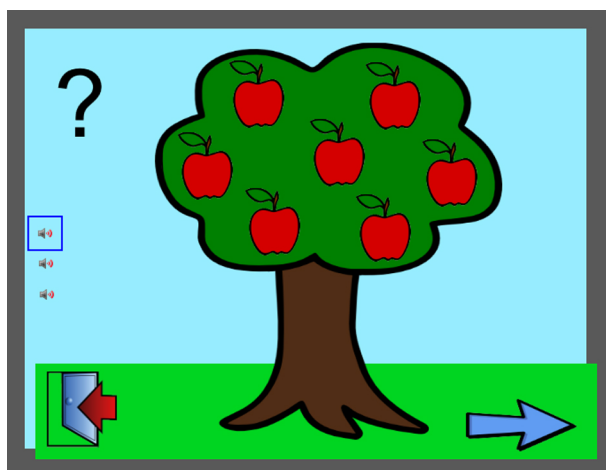
Stalo se, že GamLab studio nebylo možné spustit, protože při startu nastala fatální chyba. Nevyřešilo jej ani přinstalování aplikace. Jelikož je tento problém velice zásadní, byl okamžitě konzultován s autorem prostředí GamLab, Ing. Tomášem Pastorkem. Chyba byla v konfiguračním souboru, který se za nezjištěných okolností poškodil a bylo třeba jej vymazat. Tento problém se po celou dobu práce s GamLabem vyskytl dvakrát, nebylo však ani v jednom případě zjištěno, co jej zapříčinilo.

Nedostatečné chybové hlášky

GamLab je opatřen chybovou konzolí, pomocí které lze sledovat chyby nastalé při běhu programu. Výpis chyb však není příliš podrobný a je mnohdy těžké podle něj příčinu chyby odhalit.

Posun výsledné scény oproti té v editoru scén

Pozice objektů vůči scéně není stejná v editoru scén a GamLab Playeru. Je proto nutné objekty ve scéně posunout, aby se ve spuštěné scéně nacházely na požadovaném místě, viz Obrázek 9.1. O kolik je třeba objekty posunout lze zjistit pouze opakovaným zkoušením.



Obrázek 9.1: Nutný posun objektů ve scéně

Dokumentace

Dokumentace prostředí GamLab je součástí diplomové práce Ing. Tomáše Pastorka [2], ovšem zahrnuje pouze základní ovládání prostředí. Některé funkce musely být vyhledány ve zdrojovém kódu programu. Dokumentace například neobsahuje informaci o tom, jak je možné procházet soubory nahrané pomocí Správce souborů.

Návrhy na vylepšení

Funkce Zpět (Undo) ve scéně

Velmi užitečnou funkcí editoru scény by byla možnost vrátit se o krok zpět v úpravách. Práce s editací objektů by tak byla velmi usnadněna.

Větší výběr základních objektů

GamLab poskytuje pouze čtyři základní objekty, které se dají vložit do scény. Bylo by užitečné, kdyby se daly použít i další objekty, jako například polygony, čáry, šipky a další základní tvary.

Shrnutí

Na základě vlastních zkušeností si myslím, že GamLab je pro vývoj jednoduchých her vyhovující. Poskytuje velmi užitečné funkce pro vývoj her, jako například vestavěné fyzikální simulace nebo grafický editor scény. Další velkou předností je využití skriptovacího jazyka Boo pro obsluhu událostí. Ačkoliv je tento jazyk sám o sobě velice jednoduchý, má možnost využít kompletní knihovnu objektů .NET, čímž se z něho stává velice silný nástroj.

Práce s tímto prostředím však není zatím uživatelsky pohodlná, na což mají vliv výše zmíněné nedostatky. Ty sice nemají vliv na výsledné hry, ovšem po jejich odstranění by byla práce pohodlnější a efektivnější. Vzhledem k tomu, že vyvstaly návrhy na vylepšení a je pravděpodobné, že při další práci by se objevilo mnoho jiných, a zároveň má-li být GamLab Studio dlouhodobě využíváno, jeho aktivní vývoj je pravděpodobně nevyhnutelný.

Kapitola 10

Závěr

Cílem této práce bylo vytvořit interaktivní hry pro slabozraké děti v předškolním věku. V rámci analýzy byly klasifikovány jednotlivé hry podle typu a následně rozděleny mezi oba řešitele. Na základě všech sesbíraných materiálů byl vytvořen katalog funkčních a nefunkčních požadavků. V úvodní fázi byla také provedena internetová rešerše existujících řešení. Pomocí tohoto průzkumu byly objeveny programy, jenž částečně vyhovovaly stanoveným požadavkům. Tyto programy pak sloužily jako jeden ze zdrojů inspirace pro vlastní řešení. Výsledky rešerše jsou prezentovány v Kapitole 5.

Podle kritérií sestavených ve spolupráci s Mgr. Hanou Červenákovou byla vytvořena kolekce grafických materiálů. Materiály byly tvořeny v prostředí open source grafického vektorového editoru Inkscape. Bylo vytvořeno přes osmdesát vyhovujících obrázků. Proces vytváření popisuje Kapitola 6.

Vzhledem k tomu, že bylo doporučeno program vytvořit v prostředí GamLab, bylo nutné si jej nastudovat společně se skriptovacím jazykem Boo, který GamLab využívá. Následovala implementace vlastního řešení na základě sestaveného katalogu požadavků a návrhové prezentace včetně připomínek.

Testování softwaru bylo provedeno v prostorách SPC, účastnilo jí 7 dětí přímo z místní mateřské školy. Testy proběhly za supervize Mgr. Hany Červenákové a odhalily několik nedostatků stávající aplikace a zároveň přinesly mnoho cenných návrhů na vylepšení. Chyby odhalené uživatelskými testy byly odstraněny a některé návrhy zapracovány. Detailní průběh uživatelské studie dokumentuje Kapitola 10.

Projekt byl úspěšně realizován, splnil všechny vytyčené cíle. Výsledný program je připraven k nasazení a použití.

10.1 Možnosti dalšího vývoje

V průběhu celé práce se objevovalo mnoho nápadů, ovšem ne všechny se podařilo realizovat. Z tohoto důvodu byla domluvena navazující spolupráce s SPC na dalším vývoji interaktivních úloh. První fáze návazné spolupráce bude spočívat v rozšiřování existujících her, další fáze pak ve vývoji her nových, což by mohlo být předmětem další bakalářské práce.

Literatura

- [1] Bohumil Tománek. Cvičební hry pro zrakově postižené děti v předškolním věku za použití počítače. Semestrální projekt, 2011.
- [2] Ing. Tomáš Pastorek. Prostředí pro tvorbu jednoduchých her. Diplomová práce, 2010.
- [3] Jiřina Bednářová. *Rozvoj zrakového vnímání pro děti od 4 do 6 let.* : Computer Press, 2009.
- [4] kantouth. *thefreesoundproject* [online]. Dostupné z: <<http://www.freesound.org/usersViewSingle.php?id=1752241>>.
- [5] KIDiddles.com. *Apples* [online]. Dostupné z: <<http://www.kididdles.com/lyrics/a049.html>>.
- [6] Mgr. Alena Keblová, Mgr. Lydie Lindáková, MUDr. Ivan Novák. *Náprava poruch binokulárního vidění.* : Septima, 1st edition, 2000.
- [7] Rolera LLC. *Clker.com, The online royalty free public domain clip art* [online]. 2011. Dostupné z: <<http://www.clker.com/>>.
- [8] SPC při MŠ Horáčkova. *Cíl, vznik, charakteristika SPC* [online]. [cit. 20.5.2011]. Dostupné z: <<http://www.mshorackova.cz/index.php/spc>>.
- [9] totya. *thefreesoundproject* [online]. Dostupné z: <<http://www.freesound.org/usersViewSingle.php?id=89453>>.

Příloha A

Žádost pro rodiče

Dobrý den,

jsme studenti 3. ročníku školy ČVUT – Fakulta elektrotechnická (www.fel.cvut.cz). Jako bakalářskou práci vytváříme **Interaktivní hry pro slabozraké děti v předškolním věku** za spolupráce s vaším SPC.

Chceme Vás požádat, jestli bychom si mohli vytvořené hry vyzkoušet s Vaším dítětem. Závěry budou použity k bakalářské práci bez údajů o dítěti. Pracovat s Vaším dítětem budeme v dopoledních hodinách v prostorách SPC za supervize Mgr. Hany Červenákové.

S pozdravem a poděkováním,

Bohumil Tománek
(obor Softwarové inženýrství)

Kateřina Štomerová
(obor Web a Multimedia)

.....
podpis

.....
podpis

Příloha B

Speciálně pedagogické centrum při MŠ Horáčkova

V rámci vytváření interaktivních her pro zrakově postižené děti byla navázána spolupráce se Speciálně pedagogickým centrem při MŠ Horáčkova na Pankráci (SPC) Pod vedením místní pedagožky Mgr. Hany Červenákové byla sestavena základní kritéria pro tvorbu jednotlivých her a grafických podkladů. Za její supervize probíhalo testování s koncovými uživateli výsledného programu - zrakově postiženými dětmi z MŠ Horáčkova. Tato kapitola představuje SPC a jeho základní činnost a cíle.

Vznik a charakteristika SPC

Speciálně pedagogické centrum vzniklo v roce 1992 jako součást speciální mateřské školy. Věnuje se především specializovanému poradenství, v rámci kterého poskytuje dětem se speciálně vzdělávacími potřebami (vadami zraku, kombinovanými vadami a vadami řeči) a jejich rodičům, ale také pedagogům, studentům a široké veřejnosti zájímající se o integraci podpůrné vzdělávací a terapeutické služby.

Náplň činnosti a cíle SPC

SPC si klade tři základní cíle:

- Integrate dětí nebo žáků se speciálními vzdělávacími potřebami do intaktní společnosti.
- Vychovat z dítěte nebo žáka se speciálními vzdělávacími potřebami samostatnou, soběstačnou osobnost, se základy společenského chování a s aktivním přístupem k vnímání okolí. Dále právoplatného člena společnosti, se stejnými právy a povinnostmi jako zdravý.
- Dosáhnout u dítěte nebo žáka se speciálními vzdělávacími potřebami co nejvyššího a všestranného rozvoje, vybavit ho potřebnými kompetencemi, vést ho k co největší samostatnosti[8].

Těchto cílů se snaží SPC dosáhnout svojí činností, do které patří především následující body:

- provádění komplexní pedagogické a psychologické diagnostiky dětí a žáků se speciálními vzdělávacími potřebami,
- rozvíjení a využívání kompenzačních a reedukačních smyslů,
- výcvik prostorové orientace,
- příprava a výuka čtení a psaní Braillova písma,
- nácvik práce s optickými pomůckami,
- rozvíjení komunikačních dovedností a sebeobsluhy,
- náprava grafomotorických obtíží,
- rozvíjející cvičení předcházející poruchám učení,
- logopedická náprava[8].

Příloha C

Skriptovací jazyk Boo

Skriptovací jazyk Boo je nový objektově orientovaný programovací jazyk, který klade důraz na rozšiřitelnost sama sebe a překladače. Jeho syntaxe je inspirována syntaxí jazyka Python. Zdrojový kód v jazyce Boo je překládán do mezikódu CLI (Common Language Infrastructure) podobně jako C#. CLI je standard definující spustitelný kód a runtime prostředí platformy .NET. Díky tomu je Boo porovnatelně rychlý s ostatními jazyky .NET a může využívat kompletní knihovnu tříd.

Následuje výčet základních rysů jazyka Boo.

Zjednodušení nejpoužívanějších konstrukcí

Jazyk Boo byl navržen tak, aby minimalizoval množství kódu potřebného k napsání programu. Z tohoto důvodu poskytuje i zkrácené (ale jasně srozumitelné) varianty nejčastěji používaných programových konstrukcí a nejběžnějších .NET tříd. Například .NET objekt HashTable je zjednodušen tak, že se s ním dá pracovat jako se standardním polem.

Automatická deklarace proměnných při přiřazení

V jazyku Boo není potřeba explicitně deklarovat proměnné - konstrukce pro deklaraci proměnných ani neexistuje. Kompilátor jazyka vytvoří proměnnou v okamžiku prvního přiřazení hodnoty do této proměnné.

Statické typování a automatické určení typu

Boo je staticky typový jazyk. To znamená, že typy proměnných se určí v době překladu a není tedy možné do jedné proměnné ukládat hodnoty různých typů. Explicitní deklarace typu proměnné je však volitelná: pokud není uvedena, překladač určí typ proměnné podle typu první hodnoty, která je do proměnné přiřazena. Dalším zajímavým črtem jazyka je tzv. "duck typing", možnost použít proměnné s dynamickým typem (typ se vyhodnocuje v runtime prostředí stejně jako u dynamicky typových jazyků).

Automatické přetypování

Jazyk boo umožňuje automatické přetypování, pokud jej ovšem lze provést. Kompilátor toto nepovažuje chybu a negeneruje při automatickém přetypování výstražná hlášení. Jazyk obsahuje také konstrukci pro manuální přetypování, kterou je možné použít pro větší přehlednost kódu.

Možnost práce s funkcemi jako s hodnotami

Boo poskytuje možnost pracovat s funkcemi jako s hodnotami, takže je lze ukládat do proměnných, používat jako argumenty funkcí a návratové hodnoty. To umožňuje velice snadné programování jakýchkoliv konstrukcí využívajících callback funkce.

Generátory

V Boo lze používat popisy množin libovolných elementů pomocí tzv. generátorů. To jsou výrazy nebo funkce, které jsou iterovatelné. V každé iteraci vrátí jeden prvek množiny - objekt, který vyhovuje definovaným kritériím generátoru. Generátor může popisovat např. množinu všech sudých čísel, ale i množinu objektů na základě složitějšího kritéria.

Ukázka kódu 5 Příklad kódu v jazyce Boo

```
def JeUnikatni(varianty, varianta):
    for i in range(0, len(varianty)):
        if len(varianta) != len(varianty[i]):
            continue
        ruzne = 0
        for j in range(0, len(varianta)):
            if varianta[j] != varianty[i][j]:
                ruzne = 1
                break
        if ruzne == 0:
            return false
    return true
```

Příloha D

Obsah příloženého CD

- Text: Adresář obsahující text BP ve formátu PDF a L^AT_EX
- Prezentace.pptx: Návrhová prezentace
- Obrazky: Adresář obsahující sestavenou kolekci obrázků
- readme.txt: Soubor s podrobným obsahem CD
- Aplikace: Adresář obsahující aplikaci a její zdrojové kódy