



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Výuková aplikace Drá ek II - zásuvné moduly II
Student:	Michal Bureš
Vedoucí:	Ing. Ji í Chludil
Studijní program:	Informatika
Studijní obor:	Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2016/17

Pokyny pro vypracování

Drá ek II je dotyková vzd lávací aplikace pro OS Android pro žáky prvního stupn základní školy, mí ena p edevším na d ti s poruchami u ení. Aplikace navazuje na desktopovou aplikaci Drá ek, která byla p edm tem bakalá ských prací z minulých let.

1. Analyzujte požadavky pedagog základních škol týkající se obsahu a forem jednotlivých cvi ení.
2. Analyzujte možnosti vytvá ení jednotlivých zadání pro cvi ení a jejich nastavení.
3. Navrhn te architekturu zásuvných modul pro podporu graficky orientovaných cvi ení a komunikaci s jádrem aplikace.
4. Navrhn te architekturu editoru pro vytvá ení cvi ení pro jednotlivé moduly.
5. Implementujte alespo 5 zásuvných modul pro vybrané typy cvi ení v etn p íslušného GUI a jejich komunikaci s jádrem aplikace.
6. Implementujte editor pro vytvá ení zadání pro jednotlivá cvi ení všech 5 zásuvných modul .
7. Hotové ešení podrobte vhodným test m a vyhodno te výsledky testování.

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 4. ledna 2016

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

Výuková aplikace Dráček II - zásuvné moduly II

Michal Bureš

Vedoucí práce: Ing. Jiří Chludil

16. května 2016

Poděkování

Rád bych poděkoval vedoucímu mé práce panu Ing. Jiřímu Chludilovi za cenné rady a připomínky při tvorbě této práce.

Dále bych rád poděkoval kolegům z týmu Dráček, kteří se se mnou na tomto projektu podíleli. Jmenovitě Mirku Mazlovi, Patriku Pavelcovi, Ondřeji Filipovi a Karlu Kovařovicovi.

A rád bych také poděkoval všem, kteří mi pomáhali při vývoji modulů a při hledání chyb ve výsledném textu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 16. května 2016

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2016 Michal Bureš. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Bureš, Michal. *Výuková aplikace Dráček II - zásuvné moduly II*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Tato práce obsahuje analýzu, návrh, implementaci a testování šesti graficky orientovaných modulů pro aplikaci Dráček II: dva moduly, co děti učí rozpoznávat barvy a tvary, modul pro rozlišování inverzních figur, modul umožňující skládat obrázek z dílků, modul, co testuje souvislosti mezi obrázky a modul pro rozlišování figury a pozadí. Každý modul obsahuje, jak samostatné cvičení, tak i editor pro tvorbu nových cvičení. Dráček II je vzdělávací aplikace určená pro mobilní zařízení na platformě Android, která se zaměřuje na výuku dětí se specifickými poruchami učení.

Klíčová slova vzdělávací aplikace, poruchy učení, zraková percepce, moduly, cvičení, Dráček, hry, děti, Android

Abstract

This thesis contains of analysis, design, implementation and testing of six graphically oriented modules for application Dragon II: two modules that teaches children to recognize colors and shapes, a module for differentiating inverse figures, a module allowing to compose the image from pieces, a module that tests the connection between some images and a module for differentiating between figure and background. Each module contains exercise and editor for creating new exercises. Dragon II is an educational application designed for mobile devices on the Android platform that focuses on children with learning disabilities.

Keywords learning application, learning disabilities, visual perception, modules, exercises, Dragon, games, children, Android

Obsah

Úvod	1
O projektu Dráček	1
Interpretace zadání	1
1 Analýza	3
1.1 Termíny	3
1.2 Specifické poruchy učení	3
1.3 Výběr jednotlivých modulů	7
1.4 Analýza modulů Dráčka I	8
1.5 Požadavky jádra aplikace	11
1.6 Společné požadavky všech modulů	13
1.7 Modul Vybarvování	14
1.8 Modul Ukazování	14
1.9 Moduly Skládání	15
1.10 Modul Obrácení	15
1.11 Modul Rozlišování	15
1.12 Modul Figury a pozadí	16
1.13 Případy užití	16
1.14 Technologie	17
2 Návrh	19
2.1 Návrh knihovny	19
2.2 Výpočet skóre	20
2.3 Návrh modulu Vybarvování	21
2.4 Návrh modulu Ukazování	23
2.5 Návrh modulu Obrácení	25
2.6 Návrh modulu Rozlišování	27
2.7 Návrh modulu Skládání	29
2.8 Návrh modulu Figury a pozadí	31

3 Implementace	33
3.1 Komunikace s jádrem aplikace	33
3.2 Instalační příručka	34
3.3 Grafika	35
3.4 Nedostatky	36
3.5 Uživatelská příručka	36
4 Testování	41
4.1 Manuální testování	41
4.2 Unit testy	41
4.3 Uživatelské testování	41
4.4 Testovací scénář editorů	41
4.5 Výsledky testování	46
Závěr	47
Splnění cílů	47
Osobní přínos	48
Budoucnost projektu	48
Literatura	49
A Seznam použitých zkratk	51
B Obsah příloženého CD	53

Seznam obrázků

1.1	Dráček I: Modul Vybarvování	9
1.2	Dráček I: Modul Ukazování	10
1.3	Dráček I: Modul Pozadí	11
1.4	Dráček I: Modul Obrácení	12
1.5	Dráček I: Modul Ukazování	12
2.1	Grafický návrh editoru	20
2.2	Modul Vybarvování	22
2.3	Editor modulu Vybarvování	23
2.4	Modul Ukazování	24
2.5	Modul Obrácení	26
2.6	Modul Rozlišování	28
2.7	Editor modulu Rozlišování	29
2.8	Modul Skládání	30
2.9	Modul Figury a pozadí	32
3.1	Schéma aplikace Dráček	34
3.2	Grafický návrhu modulu Vybarvování	35

Seznam tabulek

1.1 Seznam modulů	8
-----------------------------	---

Úvod

O projektu Dráček

Projekt Dráček začal vznikat na počátku roku 2012, kdy ředitelka Základní školy Smečno požádala fakultu informačních technologií ČVUT o spolupráci při vývoji výukového softwaru pro děti s poruchami učení. Na tento popud vznikl projekt Dráček I pod vedením Ing. Jiřího Chludila. Tento projekt byl úspěšně dokončen v roce 2013 a vznikla desktopová aplikace Dráček, která je úspěšně používána na ZŠ Smečno.

V roce 2015 Ing. Jiří Chludil založil projekt Dráček II, který navazoval na projekt Dráček I. Úkolem Dráčka II bylo předělat Dráčka I pro mobilní zařízení na platformě Android.

Na projektu Dráček II pracoval tým lidí, kde každý člen měl za úkol vytvoření různých částí:

- Miroslav Mazel - textově orientované moduly [1]
- Ondřej Filip - serverová část [2]
- Patrik Pavelec - jádro aplikace
- Michal Bureš - graficky orientované moduly

Během práce se k týmu připojil Karel Kovařovic, který se zaměřil na návrh grafické stránky některých modulů [3].

Interpretace zadání

1. Analyzujte požadavky pedagogů základních škol týkající se obsahu a forem jednotlivých cvičení.

V tomto bodě budu analyzovat specifické poruchy učení, zrakovou percepci, dále se budu věnovat jednotlivým cvičením, které testují poruchy učení. Také

bude muset vybrat cvičení, která jsou vhodná pro vypracování.

2. Analyzujte možnosti vytváření jednotlivých zadání pro cvičení a jejich nastavení.

V této části budu analyzovat, jaké jsou možnosti pro ukládání zadání v jednotlivých cvičení.

3. Navrhněte architekturu zásuvných modulů pro podporu graficky orientovaných cvičení a komunikaci s jádrem aplikace.

V tomto bodě navrhnu architekturu modulu pro každé cvičení, které jsem vybral v bodu č.1. Další částí, kterou navrhnu je společné rozhraní pro komunikaci s jádrem aplikace.

4. Navrhněte architekturu editoru pro vytváření cvičení pro jednotlivé moduly.

V této části navrhnu architekturu editoru pro každý modul.

5. Implementujte alespoň 5 zásuvných modulů pro vybrané typy cvičení včetně příslušného GUI a jejich komunikaci s jádrem aplikace.

V této části implementuji 5 zásuvných modulů, které budou splňovat požadavky jádra aplikace a vytvořím jednoduché GUI rozhraní pro každý modul.

6. Implementuje editor pro vytváření zadání pro jednotlivá cvičení všech 5 zásuvných modulů.

V tomto bodě implementuji editor pro každý modul. Editor bude sloužit k vytváření a editaci zadání pro příslušný modul.

7. Hotové řešení podrobte vhodným testům a vyhodnoťte výsledky testování.

Všechny moduly budu v průběhu vývoje podrobovat testování za účelem nalezení chyb.

Analýza

V úvodu kapitoly analýza jsou definovány pojmy specifické poruchy učení, zraková percepce a dále jsou zde analyzována jednotlivá cvičení na rozvoj zrakového percepce. Poté následuje analýza požadavků jádra aplikace a na závěr kapitoly jsou sepsány všechny funkční a nefunkční požadavky každého modulu a analýza technologií.

1.1 Termíny

V následujícím textu jsou uvedeny některé termíny, které se v práci často objevují a jejichž význam nemusí být na první pohled úplně zřejmý.

- **Modul** je samostatná aplikace, která obsahuje hru a editor.
- **Hra** je část modulu, která žákovi umožňuje vypracovat jednotlivá cvičení.
- **Editor** je část modulu, která se věnuje vytváření a editaci zadání.
- **Jádro aplikace** je hlavní část aplikace Dráček II, která spravuje všechny moduly a výsledky cvičení. Pomocí jádra aplikace se spouští jednotlivé hry a editory.

1.2 Specifické poruchy učení

V následujícím textu jsou definovány pojmy specifické poruchy učení a zraková percepce. Dále jsou zde rozebrány problémy, které poruchy způsobují a uvedena cvičení na rozvoj jednotlivých poruch.

1.2.1 Definice specifické poruchy učení

„Poruchy učení jsou souhrnným označením různorodé skupiny poruch, které se projevují zřetelnými obtížemi při nabývání a užívání takových dovedností, jako je mluvení, porozumění mluvené řeči, čtení, psaní, matematické usuzování nebo počítání. Tyto poruchy jsou vlastní postiženému jedinci a předpokládají dysfunkci centrálního nervového systému. I když se porucha učení může vyskytnout souběžně s jinými formami postižení (jako např. smyslové vady, mentální retardace, sociální a emocionální poruchy) nebo souběžně s jinými vlivy prostředí (např. kulturní zvláštnosti, nedostatečná nebo nevhodná výuka, psychogenní činitelé), není přímým následkem takových vlivů.“ [4]

Tyto poruchy často vznikají drobným poškozením už v době před narozením nebo v raném věku dítěte, kdy je mozková tkáň snadno zranitelná. Svou mnohdy významnou roli zde hraje i dědičnost, případně kombinace dědičnosti a výše uvedených faktorů. Nejedná se tedy o problematiku získanou z vnějších příčin, kdy obdobné obtíže také mohou vznikat např. použitím nesprávných didaktických postupů, neurotizací dítěte, dlouhodobou absencí, nižší sociokulturní úrovní nebo odlišným jazykovým prostředím rodiny dítěte.

Faktem je i to, že intelektuální schopnosti takto postižených dětí jsou většinou průměrné až nadprůměrné. Jejich porucha není tedy způsobena sníženými intelektovými schopnostmi, ale vyplývá z jejich příčin. Bývají porušeny funkce, které jsou potřebné pro učení, čtení, psaní, počítání a dalších základních dovedností. Jedná se zejména o funkce percepční, kdy je porušeno především smyslové vnímání (zrakové, sluchové). Dále to mohou být poruchy funkce kognitivní (poznávací), funkce motorické a motorické koordinace. A nakonec i porucha senzomotorických funkcí, kdy se jedná o propojení percepčních, poznávacích a motorických funkcí. Proto jsou specifické poruchy učení vnímány jako funkční poruchy, protože není poškozen tělesný orgán jako takový, ale je narušena funkce centrální nervové soustavy [5].

1.2.2 Zraková percepce

Percepce neboli vnímání je schopnost a dovednost smyslů pociťovat a uvědomovat si vnější i vnitřní podněty. Je to kognitivní proces organizování a interpretování sensorických informací z jednotlivých smyslů. V současné době rozeznáváme základních pět smyslů. Mezi základní smysly patří sluch, hmat, čich, chuť a zrak. Dalšími smysly mohou být smysl pro teplo, chlad, vibrace ale i vnímání gravitačního pole země a mnoho dalších. Mnoho našich vjemů vzniká za účasti několika smyslů, ale nejpodstatnější část informací získáváme pomocí zrakového vnímání. Zrakem vnímáme až osmdesát procent všech vnějších informací. Proto je nutno zrakovému vnímání věnovat maximální pozornost. [6]

Zraková percepce se rozvíjí už od narození dítěte. Nejdříve jsou vnímány světlo a tma, později obrysy jednotlivých předmětů. S přibývajícím věkem se vnímání zpřesňuje směrem k větší diferenciaci tvarů. Zraková percepce je

jedním z důležitých činitelů při nábviku čtení, psaní, matematiky (převážně aritmetiky a geometrie) a dalších základních dovedností.

Cvičení na rozvoj zrakové percepce můžeme rozdělit do následujících oblastí. Rozdělení do oblastí má význam spíše teoretický, protože oblasti spolu velmi úzce souvisí: [7]

- Zraková analýza a syntéza
- Zraková paměť
- Rozlišování figura - pozadí
- Rozlišování reverzních figur
- Rozlišování barev a tvarů

1.2.3 Zraková analýza a syntéza

Vnímání celku předchází vnímání dílčích částí. Pro předškolní věk je charakteristické vnímání se zaměřením spíše na celek než na jednotlivé části. Pro čtení, psaní (opisování, přepisování), počítání a dalších dovedností je však důležité vnímání celku i jeho jednotlivostí. Vnímání částí sehrává výraznou roli při odlišení detailů, je významné pro tzv. technické myšlení a matematiku. Ve školním věku se postupně dítě učí rozpoznávat dílčí části.

Oslabení zrakové analýzy a syntézy může mít u dítěte ve školním věku za následek pomalejší osvojování písmen, jejich zapamatování a časté záměny písmen. Dále může způsobovat obtíže v matematice, hlavně pak v aritmetice nebo geometrii a může negativně ovlivňovat práci s mapou [8].

Při cvičení této schopnosti se typicky používají cvičení:

- Skládání rozstříhaných obrázků.
- Skládání písmen z jednotlivých částí.
- Dokreslování nekompletních obrázků a písmen.

1.2.4 Zraková paměť

Zraková paměť je druh krátkodobé paměti, která zpracovává údaje, které přijímáme pomocí zraku.

Oslabení zrakové paměti může mít u dítěte ve školním věku za následek obtíže v zapamatování si tvarů písmen, což způsobuje problémy se čtením a psaním. Obecně se dá říct, že porucha zrakové paměti velmi ovlivňuje schopnosti dítěte se učit.

Pro trénování zrakové paměti se používají cvičení:

- Děti sledují vybrané předměty a po zakrytí těchto předmětů je jmenují.

- Dětem jsou na krátkou chvíli ukázány písmena nebo slabiky. Následně je musí zopakovat.
- Pexeso.

1.2.5 Rozlišování figura - pozadí

Při vnímání zrakových podnětů dochází zároveň k jejich uspořádávání. Celek je rozčleněn na objekt, který nás zajímá přednostně (figura) a jeho okolí (pozadí). Jde tedy o schopnost soustředit se na potřebný zrakový podnět a rozlišit, co je pro nás důležité (figura) a co nepodstatné (pozadí).

Oslabení figury a pozadí může mít u dítěte školního věku za následek ztíženou orientaci v ploše při prohlížení složitějších obrázků s více prvky nebo také potíže při zaměření pozornosti na jednotlivé prvky a jejich splýnutí s pozadím. Dalším následkem často bývá oslabené čtení [8].

Pro trénování rozlišování figur a pozadí se používají cvičení:

- Vyhledávání předmětů na pozadí.
- Vyhledávání písmen na pozadí.
- Určení více písmen, které jsou napsány přes sebe.

1.2.6 Rozlišování reverzních figur

Rozlišování reverzních (inverzních) figur je důležitá vlastnost, která se uplatňuje hlavně při čtení a psaní. Inverze můžeme rozdělit na statické a kinetické. Statické inverze způsobují záměnu figur či písmen odlišných podle roviny horizontální (6-9) nebo vertikální (b-d). Kinetické inverze se projevují záměnou písmen i celých slabik (jen-nej, lokomotiva-lomokotiva) [7].

Oslabení zrakového rozlišování může mít u dítěte školního věku za následek záměny písmen lišících se v detailech (např. záměny písmen m-n, k-h, záměny číslic 3-9, 4-7), záměny operačních znaků v matematice, obtížné rozlišení horní-dolní postavení (záměny písmen a číslic lišících se polohou nahoře-dole např. t-j, b-p, 6-9), pomalejší osvojování písmen a problémy se zapamatováním jednotlivých písmen. Dále také zvyšuje chybovost při čtení [8].

Pro trénování rozlišování figur a pozadí se dají použít cvičení:

- Vybarvi všechna „a“ jednou barvou a písmena „e“ odlišnou barvou.
- Hledání tvarů podle vzorů.
- Hledání stejných tvarů.
- Podtrhni tvar, který je stejný jako první tvar na řádku.
- Hledání stranově převrácených tvarů nebo obrázků.

1.2.7 Rozlišování barev a tvarů

Funkci rozlišování barev získává člověk již v předškolním věku. V předškolním věku je dítě schopno rozeznávat tři až čtyři základní barvy, a to červenou, zelenou, žlutou a modrou. S přibývajícím věkem se barevná citlivost zdokonaluje a dítě se učí rozeznávat i doplňkové barvy a jemnější barevné odstíny [9].

Pro trénování rozlišování barev a tvarů se používají cvičení:

- Spojování zrakového vjemu s pojmenováním a s charakteristickým předmětem (zelená jako tráva, žlutá jako sluníčko atd.).
- Rozlišování dvou a více barev. Při cvičení se vždy začíná u základních barev a postupně se přidávají i další barvy.
- Třídění předmětů podle barev, složitější je třídění podle barev a současně podle tvarů (např. červené kolečka, zelené čtverce, atd.).

1.3 Výběr jednotlivých modulů

Jednotlivé moduly byly vybrány tak, aby otestovaly základní poruchy zrakové percepce. Dalším faktorem, který byl zvážen při výběru modulů, bylo zachování modulů z Dráčka I i v nové verzi Dráčka pro mobilní zařízení. Tyto moduly jsou úspěšně používány na ZŠ Smečno, a proto byly by škoda, kdyby tyto moduly nebyly i v nové verzi Dráčka.

Modul Vybarvování si dává za cíl testovat děti v rozeznávání barev a rozeznávání tvarů. Modul připraví obrázek rozdělený na určité tvary (předměty) a úkolem bude obrázek správně obarvit podle legendy.

Modul Ukazování si dává za cíl testovat děti v rozeznávání tvarů. Modul zobrazí obrázek, na kterém bude žák postupně ukazovat jednotlivé tvary (předměty) obsažené v obrázku podle legendy.

Modul Skládání testuje zrakovou analýzu a syntézu. Tento modul „rozstříhá“ obrázek na několik částí, následně je promíchá a úkolem žáka bude složit obrázek zpátky do původní podoby.

Modul Rozlišování je vytvořen jako modul, který lze použít pro testování více poruch. Modul zobrazí skupinu obrázků a úkolem žáka bude vybrat některé obrázky podle zadání úkolu. Modul lze například použít pro rozeznávání barev (úkolem bude vybrat určitou barvu), tvarů (úkolem bude vybrat určité tvary), rozeznávání nějakých souvislostí (úkolem může být vybrat obrázky, které nepředstavují dopravní prostředky, atd.) a mnoha dalších.

Tabulka 1.1: Seznam modulů a poruch, které testují

Modul Vybarvování	Rozlišování barev a tvarů
Modul Ukazování	Rozlišování tvarů
Modul Skládání	Zraková analýza a syntéza
Modul Rozlišování	Záleží na typu obrázků
Modul Obrácení	Rozlišování reverzních figur
Modul Figury a pozadí	Rozlišování figury a pozadí

Modul Obrácení se zabývá testováním schopnosti rozlišování reverzních figur. V modulu bude zobrazena skupina obrázků a úkolem žáka bude vybrat obrázek, který je stranově převrácený nebo případně pootočený.

Modul Figury a pozadí si klade za úkol testovat žáky v rozlišení figury a pozadí. Úkolem pro tento typ cvičení je podívat se na obrázky (jeden nebo více) v jedné skupině obrázků a najít obrázek ve druhé skupině, který skrývá všechny obrázky z první skupiny. Obrázky ve druhé skupině budou upraveny otočením nebo překrytím, aby nalezení žákovi komplikovaly.

1.3.1 Shrnutí

V tabulce 1.1 na stránce 8 jsou shrnuty vybrané moduly a poruchy, na které se zaměřují.

1.4 Analýza modulů Dráčka I

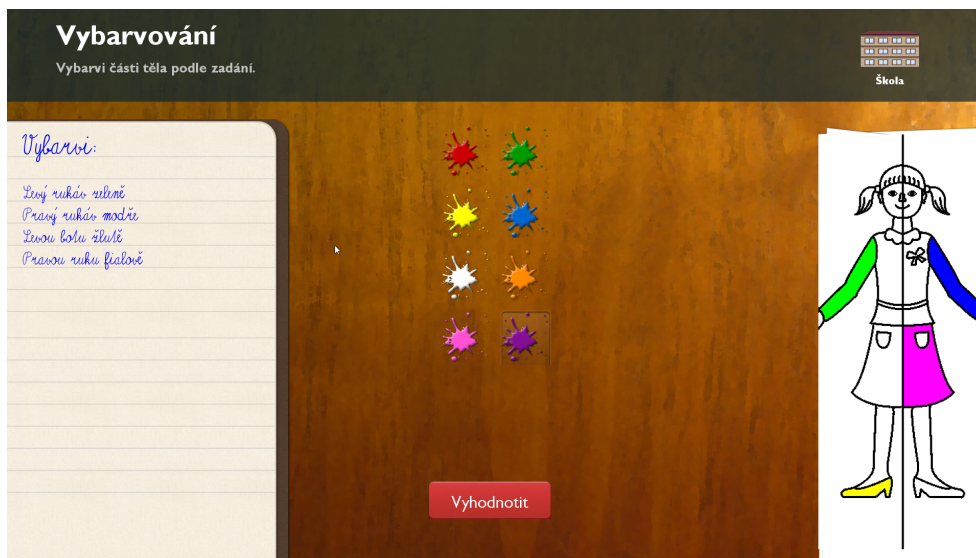
V této části kapitoly jsou analyzovány moduly z Dráčka I [10]. Vzhledem k tomu, že výběr modulů je podobný modulům Dráčka I bude analýza zaměřena převážně na části, které by šly v nové verzi Dráčka vylepšit.

1.4.1 Modul Vybarvování

Modul Vybarvování má dvě omezení, které by bylo vhodné v nové verzi Dráčka odstranit. Prvním omezením je skutečnost, že každé cvičení může obsahovat pouze jeden obrázek. Toto omezení učitelé komplikuje práci při vytváření nových zadání, jelikož je nucen pro každý další obrázek vytvářet nové zadání. Proto v nové verzi bude každé cvičení obsahovat libovolný počet obrázků, což práci učitelům výrazně ulehčí. Druhým omezením je nutnost zadání konkrétní barvy při určování oblastí (tvarů). V nové verzi tedy přibude možnost náhodné barvy, která při spuštění cvičení bude barvu náhodně generovat.

Shrnutí

- Modul bude obsahovat více obrázků v jednom cvičení.



Obrázek 1.1: Ukázka modulu Vybarvování z Dráčka I

- Při označování oblastí (tvarů) obrázku bude možnost zvolit náhodnou barvu.

1.4.2 Modul Ukazování

I modul Ukazování obsahuje omezení pouze jednoho obrázku ve cvičení, proto i v tomto modulu přibude možnost mít ve cvičení více obrázků.

Shrnutí

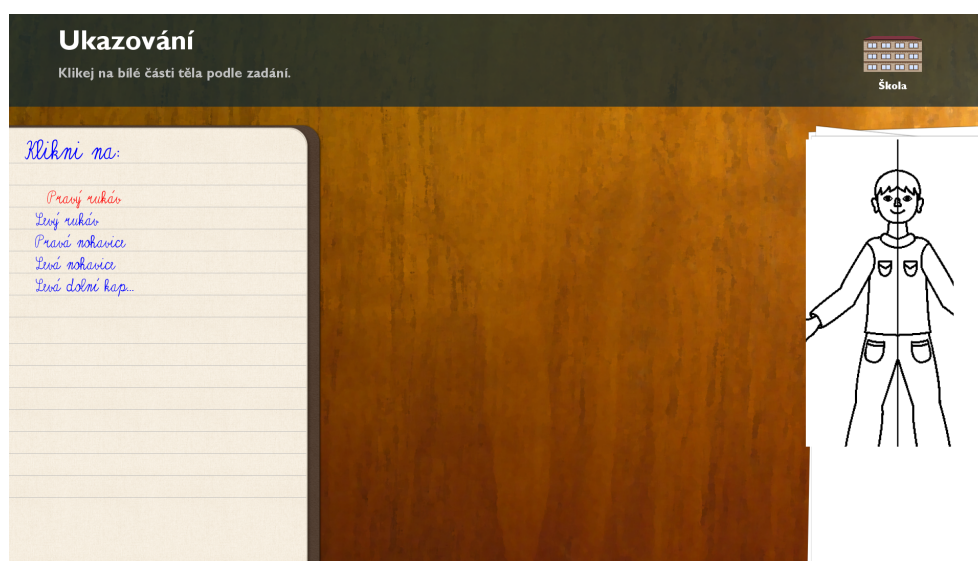
- Modul bude obsahovat více obrázků v jednom cvičení.

1.4.3 Modul Pozadí

Hlavním omezením modulu Pozadí je nemožnost použít všechny typy překrytí najednou. Modul Pozadí dovoluje pouze následující kombinace překrytí:

- Překrytí čarami pouze buď zleva doprava, shora dolů anebo obráceně
- Překrytí čarami zleva doprava i zprava doleva, shora dolů
- Překrytí jednoho objektu druhým (případně více) bez pootočení
- Překrytí jednoho objektu druhým (případně více) s možností rotace pro všechny objekty

1. ANALÝZA



Obrázek 1.2: Ukázka modulu Ukazování z Dráčka I

Toto omezení není příliš důležité, ale přeci jenom učitel by mohl chtít použít jiné kombinace překrytí obrázků, než nabízí původní modul Pozadí.

Shrnutí

- Při vytváření zadání bude možnost kombinovat všechny možnosti překrytí obrázků.

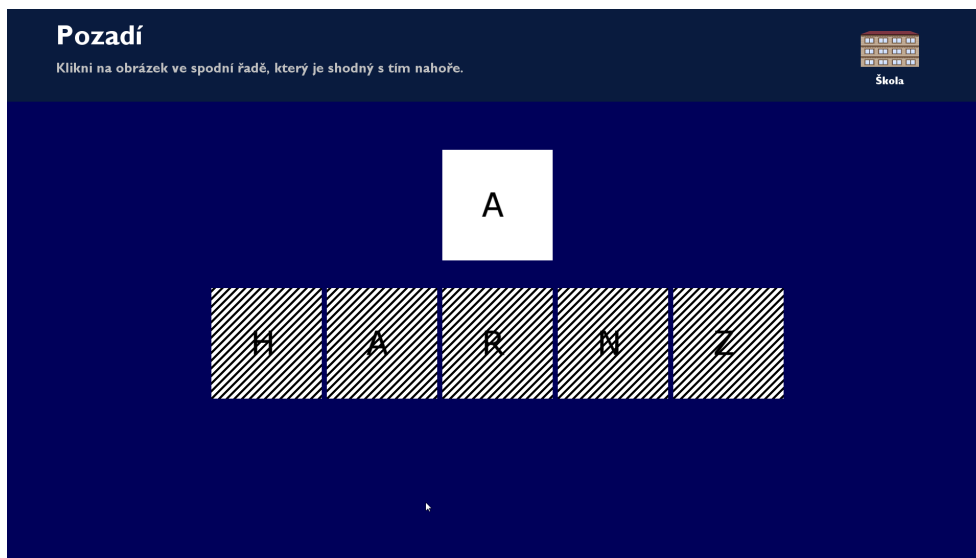
1.4.4 Modul Obrácení

Modul Obrácení by se dal vylepšit ve dvou aspektech. Prvním aspektem je problém s přesně stanoveným počtem obrázků, který je v každém cvičení nastaven na osm. Tato limitace bude v nové verzi dráčka odstraněna a učitel si sám bude moct v editoru nastavit, z kolika obrázků žák bude vybírat jeden odlišný.

Další aspekt, který by se dal vylepšit souvisí s vytvářením nových zadání. Učitel při tvorbě zadání musí načíst, jak obrázek normální, tak obrázek, který žák bude vybírat jako odlišný. Z toho plyne, že učitel si musí obrázek sám upravit někde mimo modul v jiné aplikaci. V nové verzi Dráčka se tedy počítá s úpravou obrázku přímo v modulu.

Shrnutí

- Učitel bude mít možnost si vybrat libovolný počet obrázků ve cvičení.



Obrázek 1.3: Ukázka modulu Pozadí z Dráčka I

- Učitel nebude muset zadávat dva obrázky při tvorbě cvičení. Zadávat bude pouze jeden obrázek a způsob odlišení jednoho obrázku.

1.4.5 Modul Rozlišování

Hlavním omezením modulu Rozlišování je stejně jako u předchozího modulu přesně stanovený počet obrázků v každém cvičení, který je v tomto případě také nastaven na osm.

Shrnutí

- Učitel bude mít možnost si vybrat libovolný počet obrázků ve cvičení.

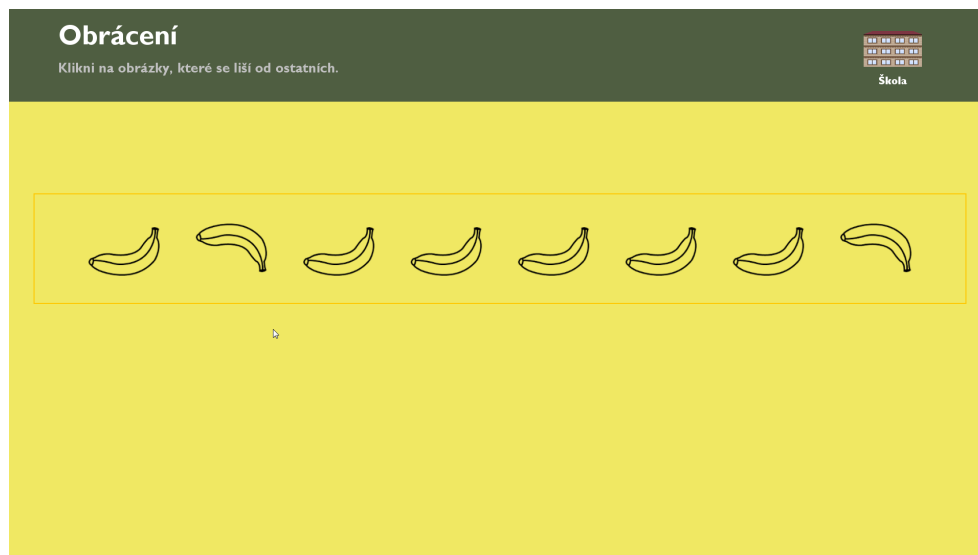
1.5 Požadavky jádra aplikace

V následujícím textu jsou analyzovány požadavky jádra aplikace.

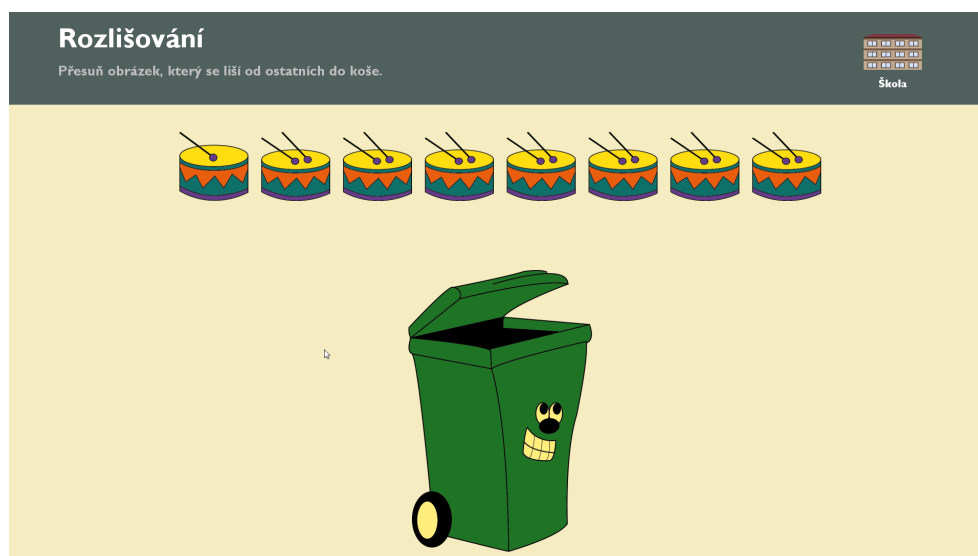
1.5.1 Zadání

Jádro aplikace požaduje, aby každý modul obsahoval editor pro tvorbu a editaci zadání. Vzhledem k tomu, že jádro bude zadání pouze ukládat je struktura a formát zadání ponechán čistě na jednotlivých modulech. Jádro pouze požaduje, aby zadání bylo v textovém formátu. Analýza možností, jak ukládat zadání lze nalézt v sekci Formát zadání na stránce 17.

1. ANALÝZA



Obrázek 1.4: Ukázka modulu Obrácení z Dráčka I



Obrázek 1.5: Ukázka modulu Rozlišování z Dráčka I

1.5.2 Výsledky cvičení

Jádro aplikace má dva požadavky ohledně výsledků cvičení. Prvním požadavkem je struktura výsledků, která je jádrem přesně definována. Struktura a formát výsledků je uveden v sekci Formát výsledků na stránce 33.

Druhý požadavek se týká jednotlivých výsledků cvičení. Každý modul musí umět změřit čas cvičení a spočítat procentuální hodnocení. Moduly si také mohou v případě potřeby přidat další specifické výsledky, které by mohli být pro učitele užitečné.

1.5.3 Android API verze

Jádro aplikace pro svoji funkčnost potřebuje API verze 21 a vyšší. Aby moduly mohly fungovat tam, kde jádro je tedy nutné, aby i všechny moduly fungovaly alespoň na API verze 21.

1.6 Společné požadavky všech modulů

Na základě analýzy poruch učení a existujících implementací byly sepsány funkční a nefunkční požadavky, které musí splňovat každý modul.

1.6.1 Společné funkční požadavky

- **FR1** - Cvičení bude obsahovat libovolný počet úkolů.
- **FR2** - Modul bude umět spočítat procentuální hodnocení cvičení a bude měřit čas potřebný pro vykonání cvičení.
- **FR3** - Žák bude moci ukončit cvičení předčasně. Výsledky v případě předčasného ukončení se nemusejí ukládat.
- **FR4** - Editor umožní vytvořit nové zadání.
- **FR5** - Editor bude umět upravovat zadání.
- **FR6** - Editor umožní učiteli přidat libovolný počet úkolů do každého cvičení.
- **FR7** - Editor bude umět smazat úkol ze cvičení.
- **FR8** - Editor bude umět upravovat jednotlivé úkoly ve cvičení.
- **FR9** - Editor bude umět nastavit jméno každého úkolu.

1.6.2 Společné nefunkční požadavky

- **N1** - Hru budou používat převážně žáci prvního stupně základní školy, a proto by prostředí mělo být přívětivé a působit nerušivě.
- **N2** - Moduly budou v českém jazyce.
- **N3** - Moduly budou implementovány pro Platformu Android, verze API 21, Android 5.0 (LOLLIPOP).

1.7 Modul Vybarvování

V této části kapitoly jsou sepsány požadavky modulu Vybarvování.

- **FRV1** - Modul musí umožnit uživateli vybrat barvu. Barva se bude vybírat z předem definované množiny základních barev.
- **FRV2** - Modul bude zobrazovat obrázek. Hranice oblastí bude reprezentovat černá barva a pozadí obrázku bude bílé.
- **FRV3** - Modul uživateli zobrazí seznam oblastí a příslušejících barev.
- **FRV4** - Modul při kliknutí na obrázek obarví vybranou oblast zvolenou barvou. Pokud je oblast už vybarvená, bude oblast přebarvená novou barvou. Kliknutí na hranice oblasti nebude mít žádný efekt.
- **FRV5** - Editor umožní uživateli načíst nový obrázek.
- **FRV6** - Editor umožní uživateli označit jednotlivé oblasti obrázku a zadat jejich jméno a barvu.

1.8 Modul Ukazování

V této části kapitoly jsou sepsány požadavky modulu Ukazování.

- **FRU1** - Modul bude zobrazovat obrázek. Hranice oblastí bude reprezentovat černá barva a pozadí obrázku bude bílé.
- **FRU2** - Modul uživateli zobrazí seznam oblastí, které se budou ukazovat na obrázku.
- **FRU3** - Modul bude označovat oblasti, které už jsou hotové.
- **FRU4** - Editor umožní uživateli načíst nový obrázek.
- **FRU5** - Editor umožní uživateli označit jednotlivé oblasti obrázku a zadat jejich jméno.

1.9 Moduly Skládání

V této části kapitoly jsou sepsány požadavky modulu Skládání.

- **FRS1** - Modul „rozstříhá“ obrázek na dílky a jednotlivé dílky prohází.
- **FRS2** - Modul umožní uživateli přesouvat jednotlivé dílky.
- **FRS3** - Editor modulu umožní uživateli zadat nový obrázek.
- **FRS4** - Editor umožní učiteli vybrat, na kolik dílků se obrázek ve cvičení „rozstříhá“.

1.10 Modul Obrácení

V této části kapitoly jsou sepsány požadavky modulu Obrácení.

- **FRS1** - Modul zobrazí obrázky, z kterých bude jeden odlišný. Pozice odlišného obrázku bude náhodná.
- **FRS2** - Modul zobrazí koš.
- **FRS3** - Uživatel bude moci přesouvat obrázky do koše.
- **FRS4** - Editor modulu umožní uživateli zadat nový obrázek.
- **FRS5** - Editor umožní učiteli zvolit způsob odlišení obrázku. Odlišení může být provedeno stranovou inverzí nebo otočením.
- **FRS6** - Editor umožní učiteli zvolit, z kolika obrázků se bude odlišný obrázek vybírat.

1.11 Modul Rozlišování

V této části kapitoly jsou sepsány požadavky modulu Rozlišování.

- **FRR1** - Modul zobrazí skupinu obrázků.
- **FRR2** - Modul zobrazí koš.
- **FRR3** - Modul umožní uživateli přesouvat obrázky do koše.
- **FRR4** - Modul zobrazí popis úkolu, který bude určovat, jaké obrázky mají být přesunuty do koše.
- **FRR5** - Editor umožní uživateli načíst nové obrázky.
- **FRR6** - Učitel bude moci označit libovolný počet obrázků jako ty, které mezi ostatní nepatří.
- **FRR7** - Učitel bude moci zadat popis úkolu.

1.12 Modul Figury a pozadí

V této části kapitoly jsou sepsány funkční požadavky modulu Figury a pozadí.

- **FRF1** - Modul zobrazí dvě řady obrázků. První řada bude obsahovat obrázky (tvary), které žák hledá. Ve druhé řadě je jednak jeden obrázek, který obsahuje všechny obrázky (tvary) z první řady a další obrázky, která žákovi hledání komplikují.
- **FRF3** - Hledaný obrázek se bude označovat kliknutím.
- **FRF4** - Editor umožní uživateli načíst nové obrázky.
- **FRF5** - Editor umožní uživateli zvolit typ překrytí obrázků. Typ překrytí může být překreslení čarami nebo pootočení.

1.13 Případy užití

Ačkoli každý modul má jiný průběh cvičení, dají se případy užití definovat všeobecně pro všechny moduly. Všechny moduly mají dva stejné aktéry, a to žáka a učitele.

1.13.1 Žák

- **UC1** - Vykonání cvičení.

1.13.2 Učitel

- **UC3** - Vytvoření nového zadání.
- **UC2** - Editace stávajícího zadání.

1.13.3 Hlavní scénář UC1

- Scénář začíná v jádru aplikace, kde si žák zvolí cvičení, které chce vypracovat.
- Jádro aplikace spustí cvičení v příslušném modulu.
- Žák vykoná zvolené cvičení.
- Modul vypočte výsledné hodnocení a odešle výsledky jádru aplikace.

1.13.4 Hlavní scénář UC2

1. Scénář začíná v jádru aplikace, kde se učitel rozhodne vytvořit nové zadání pro konkrétní modul.
2. Jádro aplikace spustí editor příslušného modulu a zobrazí zadání cvičení.
3. Učitel může editovat jednotlivé úkoly, mazat je nebo přidávat nové.
4. Učitel uloží zadání a zadání se odešle jádru aplikace. Případně může vytváření cvičení zrušit.

1.13.5 Hlavní scénář UC3

1. Scénář začíná v jádru aplikace, kde si učitel vybere zadání, které chce upravit.
2. Jádro aplikace spustí editor příslušného modulu.
3. Učitel může editovat jednotlivé úkoly, mazat je nebo přidávat nové úkoly.
4. Učitel uloží zadání a zadání se odešle jádru aplikace. Případně může zrušit všechny změny.

1.14 Technologie

V sekci technologie je uvedena analýza technologií, které jsou vhodné pro vývoj modulů.

1.14.1 Formát zadání

Jelikož jádro požaduje zadání v textovém formátu byly pro výběr zvoleny dva nejpoužívanější textové formáty pro ukládání dat a to XML a JSON. Pro výběr formátu zadání budou použity následující hodnotící kritéria: podpora Android SDK, rychlost interpretace, velikost a osobní znalost použití formátu. Maximální ohodnocení pro každé kritérium je pět.

XML XML je textově založený formát používaný pro přenos a ukládání dat. Základním stavební blokem XML je element, který se skládá z počáteční značky, koncové značky a dalších vnořených elementů [11]. Pro práci s formátem XML jsou v Android SDK připraveny potřebné nástroje.

JSON JSON je také textově založený formát pro přenos dat. JSON ukládá data jako dvojici klíč - hodnota a organizuje je do objektů a datových polí [12]. I pro zpracování formátu JSON je v Android SDK připravena potřebná podpora. V porovnání s formátem XML používá JSON méně řídicích znaků a jeho velikost pro stejná data je tedy o něco menší. I co se týče rychlosti interpretace je JSON o něco rychlejší.

	XML	JSON
Podpora Android SDK	5	5
Rychlost interpretace	4	5
Znalost formátu	4	1
Velikost	4	5
Výsledek	17	16

Vyhodnocení Jako nejlepší formát pro ukládání výsledků byl vybrán formát **XML**.

1.14.2 Jazyk

Pro vývoj na platformě Android lze primárně používat dva programovací jazyky Javu nebo C/C++.

Java Java je nejrozšířenější a doporučený jazyk pro vývoj aplikací na platformě Android. Android Java API přibližně kopíruje funkce Javy. Lze říct, že obsahuje většinu funkcí Javy 7, ale žádnou z funkcí Javy 8. Jedná se o staticky typovaný jazyk, který používá garbage collector pro správu paměti.

C/C++ Android NDK je sada nástrojů, která umožňuje napsat části Android aplikace pomocí jazyka C nebo C++. Typicky se používá pouze v případě, kdy je hlavní prioritou aplikace výkon, jako např. herní enginy, simulace chování nebo zpracování signálu. Použití C/C++ v běžných aplikacích nemá ve většině případů za následek zlepšení výkonu, ale vždy výrazně zvyšuje složitost implementace. [13].

Vyhodnocení Vzhledem k tomu, že mám větší zkušenosti s programovacím jazykem Java a u žádného modulu nejsou prováděny žádné náročné výpočetní operace, a tedy není hlavní prioritou výkon, tak byl pro vývoj všech modulů vybrán jazyk **Java**.

Návrh

V této kapitole je nejprve popsán návrh společné knihovny, která bude využívána ve všech modulech. Další část kapitoly se poté věnuje návrhu jednotlivých modulů a k nim příslušným editorům.

2.1 Návrh knihovny

První část, která je společná pro všechny moduly, je komunikace s jádrem aplikace. Tato část tedy bude obsažena ve společné knihovně **coreLibrary**, kterou poté budou všechny vytvořené moduly využívat. Součástí knihovny budou také některé konstanty, které jsou použity pro komunikaci s jádrem aplikace.

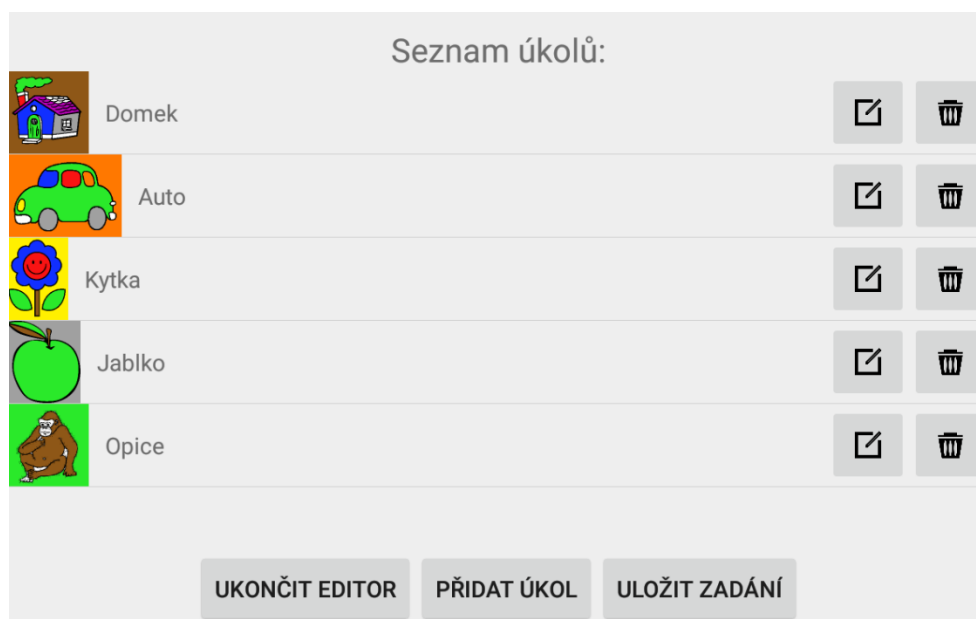
Další část, která je používána ve všech modulech je načítání obrázků a práce se souborovým systémem, a proto tato funkcionality bude také obsažena ve společné knihovně. Knihovna také bude obsahovat některé obrázky a popisky, které jsou společné pro všechny moduly.

2.1.1 Úkoly

Dalším prvkem, který obsahují všechny moduly je skutečnost, že každé cvičení může obsahovat více úkolů. Z tohoto důvodu budou vytvořeny dvě třídy, které se budou starat o jejich správu. Třída **ExerciseManager** se bude věnovat úkolům ve hrách modulů. Hlavní funkcí této třídy bude procházet jednotlivé úkoly ve cvičení a pro každý úkol ukládat výsledek.

Třída **EditorActivity** se pak bude věnovat správě úkolů v jednotlivých editorech modulů. Tato třída bude sloužit jako jakýsi kontejner pro všechny úkoly a umožní úkoly mazat, upravovat a vytvářet nové. Při vytváření a upravování úkolů se z **EditorActivity** zavolá třída **SingleEditorActivity**, která bude vytvářet úkol podle specifikace jednotlivého modulu. Grafický návrh obrazovky spravující jednotlivé úkoly v editoru je na obrázku 2.1.

2. NÁVRH



Obrázek 2.1: Grafický návrh editoru

2.1.2 Obrázky

Každé cvičení ve všech modulech může obsahovat poměrně mnoho obrázků a vzhledem tomu, že mobilní zařízení mají omezené paměťové prostředky, není vhodné, aby se všechny obrázky načítali do paměti při spuštění cvičení. Každý obrázek tedy bude mít svůj unikátní identifikátor a načten bude až v případě potřeby.

Dalším problémem, který je potřeba vyřešit, je velikost obrázků. Při vytváření cvičení má učitel možnost načítat libovolný obrázek, který může být i poměrně veliký, což by mohlo způsobit problémy jednak se zobrazováním obrázků, kvůli nedostatku paměti a také by mohlo výrazně zpomalovat aplikaci při ukládání a načítání cvičení. Proto po načtení obrázku v editoru bude velikost obrázku zmenšena na hodnotu, která zajistí, jak rozumnou kvalitu obrázku, tak i rozumnou velikost.

2.2 Výpočet skóre

Procentuální výsledek celého cvičení bude vypočten jako aritmetický průměr výsledků jednotlivých úkolů.

2.3 Návrh modulu Vybarvování

Hlavním bodem, který je potřeba navrhnout, je způsob vybarvování obrázku. V úvahu přicházejí dva způsoby. Prvním způsobem, který lze použít je obrázek rozstříhat na jednotlivé oblasti a pro každou oblast si na disk uložit kopii obrázku, a to pro všechny možné barvy. Při vybarvování obrázku by se pak tyto oblasti překrývali přes sebe. Tento způsob poskytuje určitou výhodu, neboť každá oblast obrázků by se obarvovala pouze jednou, ale na druhou stranu má tento přístup velké paměťové nároky, protože obrázek by musel být uložen mnohokrát.

Druhým způsob, který přichází v úvahu, je obrázek vybarvovat přímo. Pro vybarvování lze použít algoritmus Semínkové vybarvování (Flood Fill algorithm). Složitost tohoto algoritmu je závislá na počtu vybarvovaných pixelů, ale jelikož obrázky budou mít omezenou velikost, není tato složitost takovou překážkou. Kvůli komplikovanosti a paměťové náročnosti prvního způsobu byl pro vybarvování obrázků vybrán algoritmus Semínkové vybarvování.

Druhý problém, který bylo potřeba vyřešit, je formát načítaného obrázku. Jelikož učitel může v editoru načítat libovolný obrázek, musí proběhnout kontrola, jestli obrázek obsahuje pouze černé a bílé pixely. Z toho vzniká problém, jak se zachovat v případě jiných barev než černé a bílé. Problém lze vyřešit tak, že pixely, které nejsou bílé, budou obarveny na černo, a tedy reprezentovat okraje oblastí.

Dalším problémem, se kterým se bylo potřeba vypořádat, je odlišení hranic oblastí s pozadím obrázku. Prvním způsobem řešení je považovat černou barvu vždy za hranice oblastí. Tento způsob by ale znemožnil vybarvovat oblasti černou barvou, neboť při vybarvení černou barvou by už nebylo možné rozlišit, co jsou hranice a co je pozadí. Druhou možností je vždy při načtení obrázku projít všechny pixely a u každého si zapamatovat, jestli se jedná o hranice oblastí nebo nikoliv. Tento způsob je sice časově náročnější, ale pokud se přidá ke kontrole obrázku při načítání, nebude to znamenat žádné další požadavky na výkon.

Procentuální hodnocení jednoho úkolu bude vypočteno jako počet správně vybarvených oblastí dělený celkovým počtem oblastí, které se mají vybarvit.

2.3.1 Chování modulu

- Kliknutí mimo obrázek -> žádná reakce
- Kliknutí na hranice oblastí -> žádná reakce
- Kliknutí na aktuálně zvolenou barvu -> žádná reakce
- Kliknutí na aktuálně nezvolenou barvu -> zvýraznění zvolené barvy, zrušení zvýraznění předešlé barvy
- Kliknutí na oblast obrázku -> vybarvení oblasti zvolenou barvou

2. NÁVRH



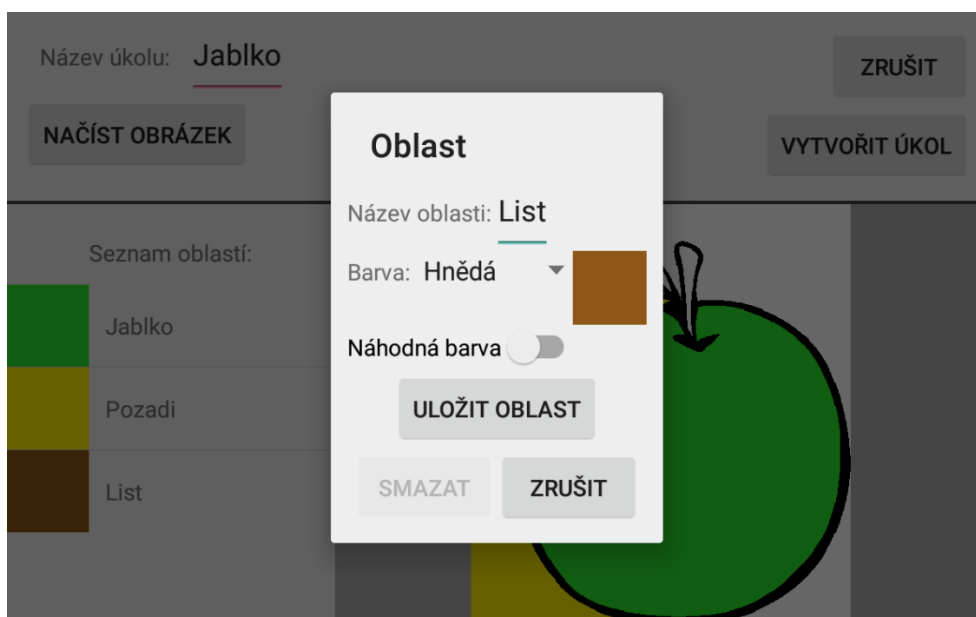
Obrázek 2.2: Ukázka grafického návrhu modulu Vybarvování

- Kliknutí na tlačítko vyhodnotit v neposledním úkolu -> vyhodnocení, přechod na další úkol
- Kliknutí na tlačítko vyhodnotit v posledním úkolu -> vyhodnocení, ukončení modulu

2.3.2 Editor

Vzhledem k tomu, že jádro požaduje editor zadání, je potřeba navrhnout jednoduchý editor, který učitelé umožní vytvořit nová zadání. Editor bude používán pouze učitelé, takže požadavky na grafiku nejsou tak náročné. V editoru bude možné vytvářet úkoly podle konceptu modulu Vybarvování. Učitel nejprve vybere obrázek, u kterého chce určit jednotlivé oblasti. Poté kliknutím do obrázku vybere oblast, kterou chce, aby žák ve cvičení vybarvoval. U každé oblasti poté nastaví barvu a slovní popis. Po vytvoření oblasti se oblast přidá do seznamu všech oblastí a oblast obrázku se obarví příslušnou barvou. Pokud učitel bude chtít u nějaké oblasti změnit barvu nebo popis, případně ji smazat, bude tak moci učinit pomocí zvolení příslušné oblasti v seznamu oblastí nebo kliknutím do obrázku na příslušnou oblast.

Každé zadání úkolu ve cvičení pro modul Vybarvování bude obsahovat následující položky:



Obrázek 2.3: Ukázka grafického návrhu editoru pro modul Vybarvování

- Identifikátor úkolu
- Název úkolu
- Seznam oblastí
 - Identifikátor oblasti
 - Bod, kterým učitel oblast určil
 - Barvu oblasti
 - Textový popis oblasti

2.4 Návrh modulu Ukazování

Vzhledem k tomu, že tento modul je velice podobný předchozímu modulu, bude využito stejných prostředků jako v modulu Vybarvování. Rozdíl nastává ve chvíli, kdy je potřeba vybrat, jakým způsobem ověřit, že žák klikl na správnou oblast obrázku. Jako první způsob se nabízí spustit vyhodnocovací algoritmus při každém kliknutí na obrázek. Vzhledem k tomu, že žák může klikat na obrázek velice často, není tento způsob ideální.

Jako druhý způsob se nabízí spustit algoritmus z modulu Vybarvování (Semínková vybarvování) na všechny body, které učitel zadal při vytváření cvičení



Obrázek 2.4: Ukázka grafického návrhu modulu Ukazování

a u každého pixelu uložit identifikátor oblasti. V tomto případě se sice prodlouží doba načítání cvičení, ale vyhodnocení správnosti kliknutí bude v konstantním čase. Vzhledem k tomu, že pro uživatele modulu je důležitější rychlé vyhodnocení správnosti, bude použit druhý způsob.

Při průběhu cvičení se žákovi bude zobrazovat, na které oblasti už klikl a které mu ještě zbývají. Po vybarvení všech oblastí modul sám přejde na další úkol.

Procentuální hodnocení úkolu bude vypočteno jako počet správně ukázaných oblastí dělený celkovým počtem oblastí, které se mají ukázat.

2.4.1 Chování modulu

- Kliknutí mimo obrázek -> žádná reakce
- Kliknutí na hranice oblastí -> žádná reakce
- Kliknutí na nesprávnou oblast v obrázku podle legendy -> žádná reakce, přičtení chyby
- Kliknutí na správnou oblast podle legendy -> přeškrtnutí oblasti v legendě

- Kliknutí na poslední správnou oblast podle legendy v neposledním úkolu -> přechod na další úkol
- Kliknutí na poslední správnou oblast podle legendy v posledním úkolu -> vyhodnocení, ukončení modulu
- Kliknutí na tlačítko ukončit cvičení -> ukončení modulu

2.4.2 Editor

Formát editoru bude velice podobný editoru v modulu Vybarvování. Nejprve učitel vybere obrázek. Dále pomocí kliknutí na obrázek postupně určí jednotlivé oblasti a pojmenuje je tak, aby žák mohl rozpoznat, o kterou část se jedná. Při vybrání oblasti se oblast vybarví náhodnou barvou, aby učitel věděl, že danou oblast už vybral.

Každé zadání úkolu ve cvičení pro modul Ukazování bude obsahovat následující položky:

- Identifikátor úkolu
- Název úkolu
- Seznam oblastí
 - Identifikátor oblasti
 - Bod, kterým učitel oblast určil
 - Textový popis oblasti

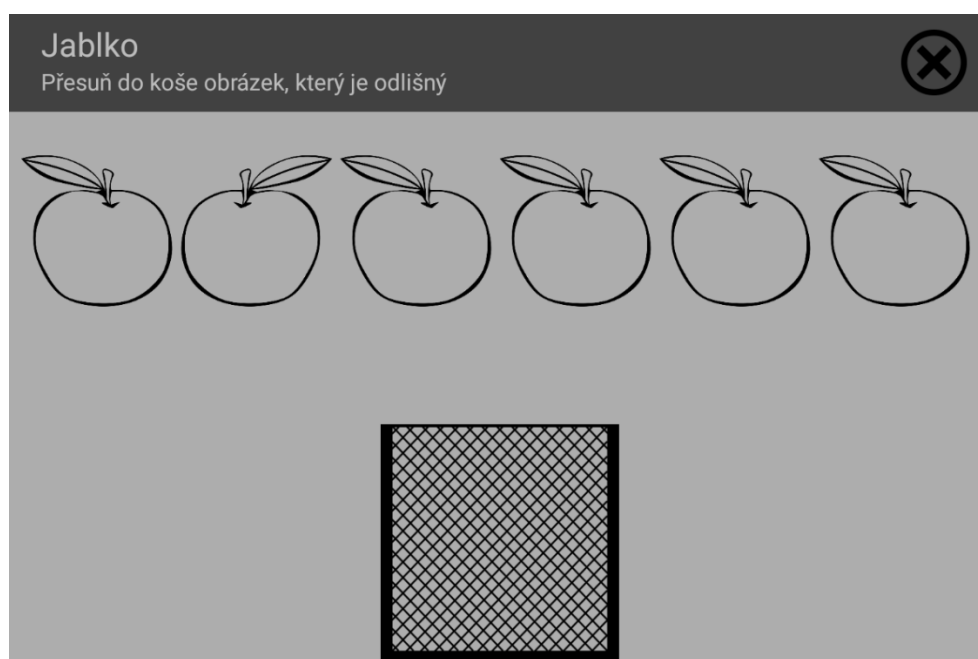
2.5 Návrh modulu Obrácení

V modulu Obrácení je nejprve nutné navrhnout, jakým způsobem se obrázky budou stranově převracet a otáčet. K tomu se výborně hodí třída *Matrix* a její metoda *postRotate()*, která dokáže obrázek otočit o libovolný počet stupňů a metoda *postScale()*, která dokáže obrázek stranově převrátit.

Dále je potřeba navrhnout, jakým způsobem se modul zachová při neúspěšném přesunutí obrázku do koše. První možnost by byla nechat obrázek vždy na místě, kam jej žák přesunul. Druhá možnost by mohla být obrázek vrátit vždy na původní místo. Vzhledem k tomu, že v první možnosti může nastat situace, kdy odlišný obrázek není viditelný, bude výhodnější použít druhý způsob.

Procentuální hodnocení v modulu Obrácení bude nejprve nastaveno na sto procent. Kdykoliv žák přesune špatný obrázek do koše, bude od hodnocení odečteno přesně tolik procent, aby při přesunu všech špatných obrázků do koše byl výsledek nula procent.

2. NÁVRH



Obrázek 2.5: Ukázka grafického návrhu modulu Obrácení

2.5.1 Chování modulu

- Kliknutí na koš nebo na pozadí modulu -> žádná reakce
- Přetáhnutí správného obrázku do koše v posledním úkolu -> vyhodnocení a ukončení modulu
- Přetáhnutí správného obrázku v neposledním úkolu -> přechod na další cvičení
- Přetáhnutí špatného obrázku do koše -> vrácení obrázku na původní místo a přičtení chyby
- Přetáhnutí správného i špatného obrázku mimo koš -> vrácení obrázku na původní místo, nezapočítání chyby
- Kliknutí na tlačítko ukončit cvičení -> ukončení modulu

2.5.2 Editor

Pro tento modul je také potřeba vytvořit jednoduchý editor, který učitelům umožní vytvořit nové zadání nebo upravit stávající zadání.

Učitel nejprve načte potřebný obrázek. Obrázek musí být čtvercového charakteru, jinak by výsledné cvičení působilo nepřehledně a nevhledně. Dále

určí, kolik obrázků bude ve výsledném cvičení a jakým způsobem se jeden z obrázků bude lišit. Editor nebude kontrolovat, jestli se upravený obrázek nebude shodovat s obrázkem původním. Tento stav způsobí, že žák ve cvičení uvidí všechny obrázky stejné a nebude schopen rozlišit, který je ten odlišný. Kontrola této správnosti zadání zůstane na učiteli.

Každé zadání úkolu ve cvičení pro modul Obrácení bude obsahovat následující položky:

- Identifikátor úkolu
- Název úkolu
- Počet obrázků v úkolu
- Typ odlišení obrázku

2.6 Návrh modulu Rozlišování

Vzhledem k tomu, že tento modul je velice podobný modulu předchozímu, bude využito stejných prostředků jako v modulu předchozím.

Jediným rozdílem je, že žák bude obrázků vybírat více a ne pouze jeden. Pokud žák přesune obrázek do koše ve chvíli, kdy ještě zbývají další obrázky, které patří do koše, obrázek pouze zmizí. Pokud přesune poslední obrázek, přejde se na další úkol.

Hodnocení modulu Rozlišování bude stejné jako v předchozím modulu.

2.6.1 Chování modulu

- Kliknutí na koš nebo na pozadí modulu -> žádná reakce
- Přetáhnutí posledního správného obrázku do koše v posledním úkolu -> vyhodnocení, ukončení modulu
- Přetáhnutí posledního správného obrázku do koše v neposledním úkolu -> přechod na další úkol
- Přetáhnutí neposledního správného obrázku do koše -> zmizení obrázku
- Přetáhnutí špatného obrázku do koše -> vrácení obrázku na původní místo, přičtení chyby
- Přetáhnutí správného i špatného obrázku mimo koš -> vrácení obrázku na původní místo, nezapočítání chyby
- Kliknutí na tlačítko ukončit cvičení -> ukončení modulu



Obrázek 2.6: Ukázka grafického návrhu modulu Rozlišování

2.6.2 Editor

I pro tento modul bylo potřeba vytvořit jednoduchý editor pro vytváření jednotlivých úkolů. Tento editor bude mít stejné rozložení jako hra modulu. Editor připraví speciální pole, do kterých se jednotlivé obrázky budou načítat. Grafický návrh modulu Rozlišování je na obrázku 2.7.

Učitel nejprve načte všechny potřebné obrázky do připravených polí, případně si přidá další pole tlačítkem plus. Obrázky musejí být pouze čtvercového charakteru, jinak by výsledné cvičení působilo nepřehledně a nevzhledně. Dále učitel označí některé obrázky jaky ty, které žák bude hledat. Nakonec učitel zadá slovní popis úkolu.

Každý úkol ve cvičení pro modul Rozlišování bude obsahovat následující položky:

- Identifikátor úkolu
- Název úkolu
- Seznam obrázků
 - Identifikátor obrázku
 - Stav, jestli je obrázek odlišný nebo ne



Obrázek 2.7: Grafický návrh editoru pro modul Rozlišování

2.7 Návrh modulu Skládání

První bodem, na který je potřeba zaměřit pozornost, je způsob skládání obrázků z jednotlivých dílků. První způsob se podobá klasické hře puzzle. Ve cvičení se zobrazí jeden obrázek, který má promíchané dílky a úkolem žáka je přeskládat jednotlivé dílky tzv. „inside“. Druhým způsobem je zobrazení, jak obrázku s promíchanými dílky, tak dalšího speciálního prostoru, do kterého žák obrázek poskládá. Jelikož druhý způsob je více přehlednější, bude použit pro vývoj modulu.

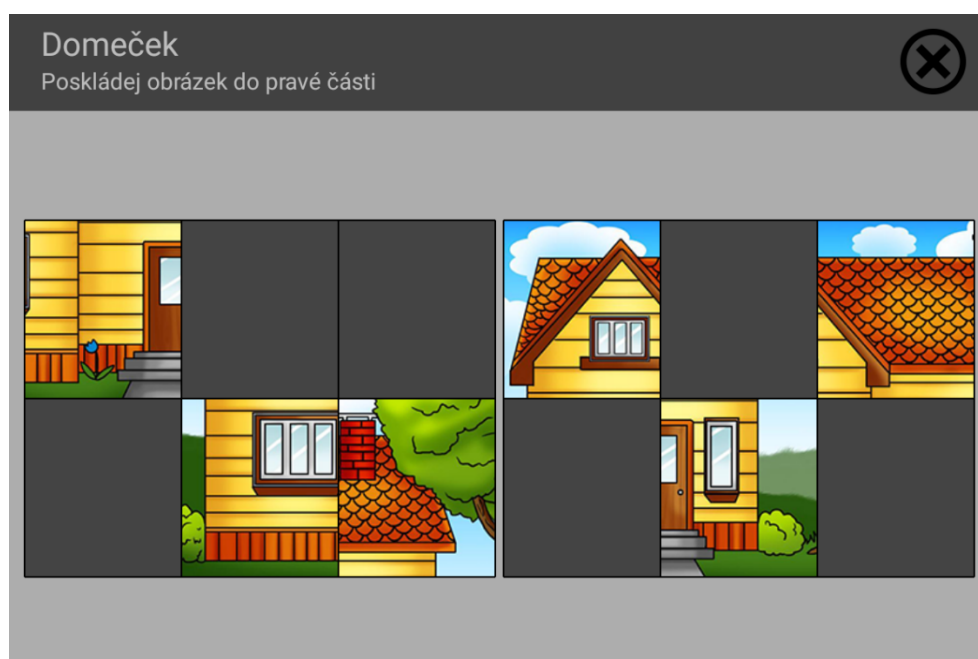
Dalším bodem, který stojí za pozornost, je rozlišení, jestli je dílek uvnitř nějakého jiného dílku. Pokud se obrázek nachází uvnitř jiného dílku alespoň na sedmdesát procent, bude to vyhodnoceno jako ano.

Procentuální hodnocení úkolu bude spočteno jako podíl minimálního počtu tahů a celkového počtu tahů žáka.

2.7.1 Chování modulu

- Kliknutí mimo obrázek -> žádná reakce
- Přesunutí všech dílků do správné pozice v posledním úkolu -> vyhodnocení a ukončení modulu

2. NÁVRH



Obrázek 2.8: Ukázka grafického návrhu modulu Skládání

- Přesunutí všech dílků do správné pozice v neposledním úkolu -> přechod na další úkol
- Přesun dílku s obrázkem na dílek s obrázkem -> prohození dílků s obrázkem
- Přesun dílku s obrázkem na dílek bez obrázku -> přesun dílku s obrázkem
- Přesun dílku s obrázkem mimo všechny ostatní dílky -> vrácení na předchozí pozici
- Kliknutí na tlačítko ukončit cvičení -> ukončení modulu

2.7.2 Editor

I pro tento modul je také potřeba vytvořit jednoduchý editor, který učitelům umožní vytvořit nové zadání nebo upravit stávající zadání.

Učitel nejprve načte potřebný obrázek. Dále určí, z kolik dílků se bude výsledný obrázek skládat a zadá název úkolu.

Editor nebude kontrolovat, jestli se nějaké dva dílky neshodují. Tento stav způsobí, že žák uvidí dva stejné dílky obrázku a nebude schopen rozlišit, který

patří na jakou pozici ve správném složení obrázků. Kontrola správnosti zadání zůstane na učiteli.

Každý úkol ve cvičení pro modul Skládání bude obsahovat následující položky:

- Identifikátor úkolu (zároveň bude reprezentovat identifikátor obrázku)
- Název úkolu
- Počet dílků, na který bude obrázek rozstříhán

2.8 Návrh modulu Figury a pozadí

V modulu Figury a pozadí jsou dva body, kterým je potřeba věnovat pozornost. Oba dva se týkají překrytí obrázku.

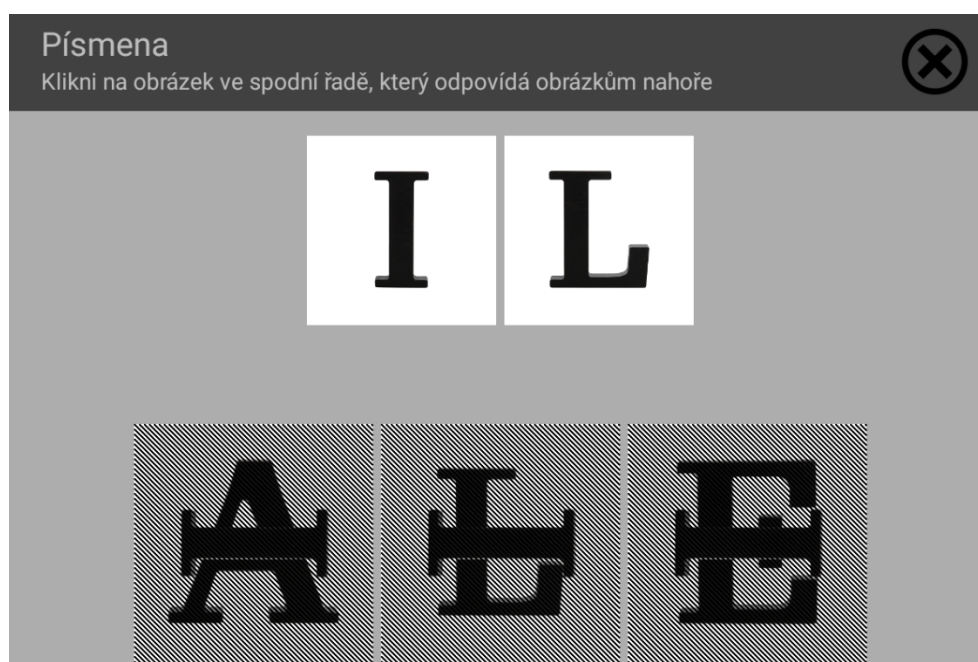
Prvním bodem je způsob překrytí obrázku čarami. K tomu se ideálně hodí třída *Canvas*. Tato třída v konstruktoru dostane obrázek a pomocí metody *drawLine()* vykreslí na obrázek potřebné čáry. Metodě je potřeba předat počáteční bod a koncový bod. Barva čar bude vždy černá.

Dalším bodem je způsob otočení obrázků přes sebe. K tomu je ideální třída *Canvas* a její metoda *drawBitmap()*. Nejprve se vytvoří nový obrázek s transparentním pozadím a poté pomocí metody *drawBitmap()* se na obrázek vykreslí další obrázky, které se nejdříve otočí pomocí třídy *Matrix* a její metody *postRotate()*.

Procentuální hodnocení v modulu Figury a pozadí bude nejprve nastaveno na sto procent. Kdykoliv žák klikne na špatný obrázek, bude od hodnocení odečteno přesně tolik procent, aby při kliknutí na všechny špatné obrázky byl výsledek nula procent.

2.8.1 Chování modulu

- Kliknutí na pozadí modulu -> žádná reakce
- Kliknutí na obrázek v horní řadě -> žádná reakce
- Kliknutí na správný obrázek v dolní řadě v posledním úkolu -> přechod na další úkol
- Kliknutí na správný obrázek v dolní řadě v posledním úkolu -> vyhodnocení a ukončení modulu
- Kliknutí na špatný obrázek v dolní řadě -> žádná viditelná reakce, přičtení chyby
- Kliknutí na tlačítko ukončit cvičení -> vyhodnocení a ukončení modulu



Obrázek 2.9: Grafický návrh modulu Figury a pozadí

2.8.2 Editor

I pro tento modul bylo potřeba vytvořit jednoduchý editor pro vytváření jednotlivých úkolů. Tento editor bude mít stejné rozložení jako hra tohoto modulu, přesněji zobrazí dvě řady obrázků. Horní řada bude sloužit pro načítání obrázků, který žák bude hledat a dolní řada bude složena pro načítání obrázků, které žákovi budou hledání komplikovat.

Učitel nejprve načte všechny potřebné obrázky do připravených polí, případně si přidá další pole tlačítkem plus a zvolí způsob překrytí obrázků.

Každý úkol ve cvičení pro modul Figury a pozadí bude obsahovat následující položky:

- Identifikátor úkolu
- Typ překrytí obrázků
- Seznam obrázků
 - Identifikátor obrázku

Implementace

V kapitole Implementace jsou nejprve zmíněny některé zajímavé části implementace, převážně tedy komunikace s jádrem aplikace. V další části kapitoly je pak uvedena uživatelská a instalační příručka.

3.1 Komunikace s jádrem aplikace

V této sekci je popsána komunikace modulu s jádrem aplikace a nároky, které by musel implementovat každý nový modul, aby byl schopen komunikovat s jádrem aplikace.

3.1.1 Schéma aplikace Dráček

Na schématu 3.1 jsou zobrazeny všechny jednotlivé části aplikace Dráček. Dále je na schématu vidět, jakým způsobem mezi sebou jednotlivé části komunikují.

3.1.2 Komunikace

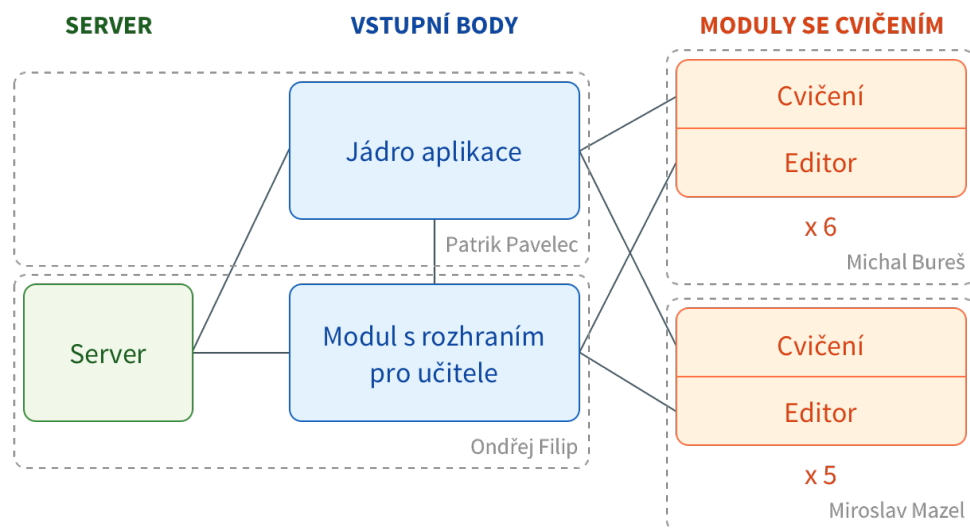
Jádro aplikace spouští jednotlivé moduly pomocí intentu. Intent je datová abstraktní struktura, která provádí nějakou operaci a skládá se ze dvou částí (akce a data, atd.). Akce specifikuje operaci (spuštění nové obrazovky, spuštění nové aplikace), kterou má daný intent provést. Jádro pomocí intentu pustí modul a předá potřebná data. Mezi data jádro ukládá zadání označené jménem **EXERCISE_FILE** a pole cest k obrázkům označené jménem **IMAGES**.

Při ukončování hry model musí odeslat výsledky cvičení, ty se označují jménem **RESULTS**. Přesný formát výsledků je uveden v 3.1. Při posílání zadání z editoru se také použije jméno **EXERCISE_FILE**. Spolu se zadáním se také posílají obrázky, které se musejí nastavit tak, aby je jádro dokázalo přečíst. Toho se dá docílit za pomoci funkce `setReadable(true,false)` použité na soubory s obrázky.

3. IMPLEMENTACE

Kód 3.1: Formát výsledků

```
<?xml version="1.0" encoding="UTF-8" ?>
<Results>
  <Time name="Celkovy cas">320s</Time>
  <Score name="Konecne score">95%</Score>
  <Specific name="specificky vysledek">
    hodnota specifického vysledku
  </Specific>
  <Specific name="specificky vysledek 2">
    hodnota specifického vysledku 2
  </Specific>
  <Specific name="specificky vysledek 3">
    hodnota specifického vysledku 3
  </Specific>
</Results>
```



Obrázek 3.1: Schéma aplikace Dráček

3.2 Instalační příručka

Pro správné nastavení projektu je potřeba nainstalovat:

- Java SDK verze alespoň 7. Java SDK je dostupné ze stránek Oracle.
- Android Studio, dostupné ze stránek Google.



Obrázek 3.2: Úkázka grafického návrhu modulu Vybarvování

Jelikož každý modul využívá společnou knihovna je potřeba ověřit, že knihovna je správně přidána do projektu. Ke správnému fungování projektu je potřeba ověřit, že soubor **settings.gradle** obsahuje následující dva příkazy, které určují, kde se knihovna nachází.

```
include ':corelibrary'
project(':corelibrary').projectDir =
    new File(settingsDir, 'coreLibraryPath')
```

Další podmínkou k funkčnosti projektu je přidání knihovny do závislostí modulu. To lze provést přidáním následujícího příkazu do sekce dependencies v souboru **build.gradle**.

```
compile project(':corelibrary')
```

3.3 Grafika

Později se k původnímu týmu připojil grafik, který navrhl grafické prostředí pro moduly Vybarvování, Rozlišování a modul Figury a pozadí. Úkázka výsledného grafické prostředí modulu Vybarvování je zobrazena na obrázku 3.2. Více informací a ukázky dalších modulů lze nalézt v práci [3].

3.4 Nedostatky

Moduly jsou v tuto chvíli funkční pro API 14 (ICE_CREAM_SANDWICH) až po API 22 (LOLLIPOP). Pro rozšíření na nejnovější verzi API 23 (MARSHMALLOW) je nutné změnit přístup k načítání obrázků ze zařízení. Pro načítání obrázků ve verzi API 23 se lehce změnila přístupová oprávnění.

3.5 Uživatelská příručka

V této sekci je sepsána uživatelská příručka pro všechny moduly a příslušející editory.

3.5.1 Instalace

Ve výsledné aplikaci se počítá s instalací přes jádro aplikace. Jádro aplikace se bude instalovat přes Google Play a z jádra aplikace se poté nainstalují všechny moduly.

Jelikož v tuto chvíli není jádro aplikace hotové, je potřeba instalovat každý modul zvlášť. Pro instalaci je nejdříve nutné povolit instalaci aplikací z neznámých zdrojů a poté pouze nainstalovat příslušné apk modulu. Moduly v tuto chvíli obsahují pouze jedno cvičení a editor, kterým lze vytvořit nové zadání a to si následně zahrát. Cvičení, která se v editoru vytvoří nejsou žádným způsobem ukládána.

3.5.2 Správa úkolů

Přidání nového úkolu - Přidání nové úkolu lze provést kliknutím na tlačítko *Vytvořit úkol* ve správci úkolů.

Smazání úkolu - Smazání úkolu se provede kliknutím na malý koš v příslušném úkolu.

Úprava úkolu - Každý úkol lze upravit stiskem tlačítka nalevo od koše v řádku s úkolem. Všechny změny se dají vrátit tlačítkem *Ukončit editor*.

Uložení úkolů - Úkoly se před ukončením editoru musejí uložit tlačítkem *Uložit úkoly*.

3.5.3 Modul Vybarvování

Změna barvy - Změna barvy se provede kliknutím na obrázek s příslušnou barvou v horní části aplikace. Tato barva bude následně použita pro vybarvování obrázku.

Vybarvení oblasti - Vybarvení oblasti se provede kliknutím do obrázku na příslušnou oblast.

Vyřešení úkolu - Po obarvení všech oblastí je potřeba kliknout na tlačítko *Vyhodnotit*.

Předčasné ukončení modulu - Modul lze ukončit kliknutím na černé X v pravém horním rohu obrazovky. Při předčasném ukončení cvičení nebude započteno žádné skóre.

3.5.4 Editor modulu Vybarvování

Načtení nového obrázku - Nový obrázek lze načíst pomocí tlačítka *Načíst obrázek*.

Přidání nové oblasti obrázku - Přidání oblasti se provede kliknutím do obrázku na příslušnou oblast. Při přidávání je potřeba zvolit barvu a zadat jméno příslušné oblasti. Jméno oblasti je povinný údaj.

Změna barvy - Změnu barvy lze provést dvojím způsobem. První možnost je kliknout do obrázku na příslušnou oblast a v dialogu zvolit požadovanou barvu. Druhá možnost je kliknout na oblast v seznamu jednotlivých oblastí a ve stejném dialogu zvolit příslušnou barvu.

Změna jména oblasti - Změnu jména lze provést dvojím způsobem. První možnost je kliknout do obrázku na příslušnou oblast a v dialogu změnit požadované jméno. Druhá možnost je kliknout na oblast v seznamu jednotlivých oblastí a ve stejném dialogu zvolit příslušnou barvu.

Smazat oblast - Smazat oblast lze provést dvojím způsobem. První možnost je kliknout do obrázku na příslušnou oblast a v dialogu stisknout tlačítko *Smazat*. Druhá možnost je kliknout na oblast v seznamu jednotlivých oblastí a ve stejném dialogu stisknout tlačítko *Smazat*.

3.5.5 Modul Ukazování

Ukázání oblasti - Ukazování oblasti se provede kliknutím do obrázku na příslušnou oblast.

Vyřešení úkolu - Pro vyřešení úkolu je nutné správně ukázat všechny požadované oblasti na obrázku.

Předčasné ukončení modulu - Modul lze ukončit kliknutím na černé X v pravém horním rohu obrazovky. Při předčasném ukončení cvičení nebude započteno žádné skóre.

3.5.6 Editor modulu Ukazování

Načtení nového obrázku - Nový obrázek lze načíst pomocí tlačítka *Načíst obrázek*.

Přidání nové oblasti obrázku - Přidání oblasti se provede kliknutím do obrázku na příslušnou oblast. Při přidávání je potřeba zadat jméno příslušné oblasti.

Změna jména oblasti - Změnu jména lze provést dvojím způsobem. První možnost je kliknout do obrázku na příslušnou oblast a v dialogu změnit požadované jméno. Druhá možnost je kliknout na oblast v seznamu jednotlivých oblastí a ve stejném dialogu zvolit příslušnou barvu.

Smazání oblasti - Smazat oblast lze provést dvojím způsobem. První možnost je kliknout do obrázku na příslušnou oblast a v dialogu stisknout tlačítko *Smazat*. Druhá možnost je kliknout na oblast v seznamu jednotlivých oblastí a ve stejném dialogu stisknout tlačítko *Smazat*.

3.5.7 Modul Obrácení

Vyřešení úkolu - Úkol se vyřeší přesunem odlišného obrázku do koše. Obrázek musí být v koši celou svoji plochou.

Předčasné ukončení modulu - Modul lze ukončit kliknutím na černé X v pravém horním rohu obrazovky. Při předčasném ukončení cvičení nebude započteno žádné skóre.

3.5.8 Editor modulu Obrácení

Načtení nového obrázku - Nový obrázek lze načíst pomocí tlačítka *Načíst obrázek*. Obrázek musí mít čtvercový tvar.

Úprava obrázku - Změnu typu odlišení obrázku lze provést v sekci *Úprava obrázku*. Stupně otočení jsou uvedeny ve směru hodinových ručiček.

Nastavení počtu obrázku ve cvičení - Počet obrázku musí být nastaven mezi třemi a patnácti. V tomto počtu je započten i obrázek, který bude zobrazován odlišně.

3.5.9 Modul Rozlišování

Vyřešení úkolu - Úkol se vyřeší přesunem všech správných obrázků do koše. Obrázek musí být v koši celou svoji plochou.

Předčasné ukončení modulu - Modul lze ukončit kliknutím na černé X v pravém horním rohu obrazovky. Při předčasném ukončení cvičení nebude započteno žádné skóre.

3.5.10 Editor modulu Rozlišování

Načtení nového obrázku - Obrázky se načítají do tmavě šedých polí uprostřed obrazovky. Kliknutím na pole se zobrazí dialog, který umožní obrázek načíst.

Nastavení obrázků, který do řady nepatří - Kliknutím na tmavě šedé pole se zobrazí dialog, který umožní obrázek nastavit jako odlišný.

Smazání obrázku - Kliknutím na tmavě šedé pole se zobrazí dialog, který umožní obrázek smazat. S obrázkem se smaže i příslušné pole.

Přidání pole pro obrázky - Přidání dalšího pole se provede stisknutím tlačítka *plus* v pravé části obrazovky.

3.5.11 Modul Skládání

Vyřešení úkolu - Pro vyřešení úkolu je potřeba složit všechny dílky obrázku do pravé části obrazovky a to tak, aby jako celek tvořili požadovaný obrázek.

Předčasné ukončení modulu - Modul lze ukončit kliknutím na černé X v pravém horním rohu obrazovky. Při předčasném ukončení cvičení nebude započteno žádné skóre.

3.5.12 Editor modulu Skládání

Načtení nového obrázku - Nový obrázek lze načíst pomocí tlačítka *Načíst obrázek*.

Změna počtu dílků zvoleného obrázku - Změnit počet dílků, na které bude obrázek ve cvičení rozdělen, lze změnou hodnot u vertikálního a horizontálního rozdělení. Výsledný počet dílků bude součin počtu vertikálních rozdělení a horizontálních rozdělení.

3.5.13 Modul Figury a pozadí

Určení správného obrázku - Správný obrázek se vybírá kliknutím na obrázek v dolní řadě obrázků. Při kliknutí na špatný obrázek se nic nestane. Pokud byl klik proveden na správný obrázek, modul přejde na další úkol ve cvičení.

Předčasné ukončení modulu - Modul lze ukončit kliknutím na černé X v pravém horním rohu obrazovky. Při předčasném ukončení cvičení nebude započteno žádné skóre.

3.5.14 Editor modulu Figury pozadí

Nastavení překrytí obrázku - Změnit překrytí lze stiskem tlačítka *Změna překrytí* v levém horním rohu obrazovky. Uvedené stupně pro otáčení obrázku vyjadřují otočení ve směru hodinových ručiček.

Načtení nového obrázku - Obrázky se načítají do tmavě šedých polí uprostřed obrazovky. Kliknutím na pole se zobrazí dialog, který umožní obrázek načíst.

Smazání obrázku - Kliknutím na tmavě šedé pole se zobrazí dialog, který umožní obrázek smazat. S obrázkem se smaže i příslušné pole.

Přidání pole pro obrázky - Přidání dalšího pole se provede stisknutím tlačítka *plus* v pravé části obrazovky.

Testování

4.1 Manuální testování

Po celou dobu implementace všech modulů byly prováděno manuální testování k ověření požadované funkcionality. Vzhledem k tomu, že moduly jsou hodně graficky orientované a výsledek všech operací je hned vidět, bylo v mnoha případech užitečnější správnou funkcionalitu ručně otestovat, než psát zbytečně komplikované testy.

4.2 Unit testy

Unit testy neboli jednotkové testy slouží k otestování jednotlivých tříd a funkcí. Unit testy se převážně použili pro testování knihovny, která obsahuje společné třídy a funkce pro všechny moduly. Testy byly pokryty převážně funkce pracující se souborovým systémem a obrázky.

4.3 Uživatelské testování

Uživatelské testování je považováno za nejdůležitější ze všech, protože dává reálnou představu o tom, jak aplikaci vidí budoucí uživatelé. Pro každý modul bylo připraveno speciální cvičení, které má za úkol ověřit funkcionalitu modulů. Uživatelské testování jsme si v týmu rozdělili na testování her a testování modulů. Mým úkolem bylo vytvořit testovací scénáře pro testování editorů. Testovací scénáře pro testování her lze nalézt v bakalářské práci [1].

4.4 Testovací scénář editorů

Hlavním účelem testování editorů je vyhodnotit, zda jsou principy pro vytváření zadání pro jednotlivé moduly srozumitelné a snadno použitelné. Hlavní snaha bude objevit problémy s použitelností editorů.

4.4.1 Uvítání účastníků a vyplnění úvodního dotazníku

Každý z účastníků bude moderátorem uvítán a usazen v testovacím prostředí. Moderátor ponechá prostor pro vyplnění úvodního dotazníku. Dále moderátor vysvětlí účastníkům účel a důvod testování.

4.4.2 Testování

Moderátor bude postupně procházet jednotlivé moduly a u každého modulu nejdříve vysvětlí, co je cílem hry modulu, pro které se bude vytvářet nové zadání. Poté postupně projde testovací scénáře všech modulů. Po testu každého modulu ponechá čas na vyplnění koncového dotazníku pro modul. Moderátor bude u všech editorů zaznamenávat problémové body a také si sepíše otázky, na které se zeptá po skončení testování.

4.4.3 Shrnutí s účastníky

Nakonec moderátor položí dodatečné otázky, které ho napadly během testu (tykající se problémů, které nastaly).

4.4.4 Úvodní dotazník

1. Jaké předmět učíte?
2. Jaké ročníky dětí učíte?
3. Učil(a) jste někdy děti s poruchami učení?
4. Jaká je vaše praxe s prací na tabletu a počítači?
5. Jaké máte zkušenosti s operačním systémem Android?
6. Používal(a) jste někdy někdy tablet / počítač pro výuku?
7. Používal(a) jste někdy někdy tablet / počítač pro výuku dětí s poruchami učení?

4.4.5 Koncový dotazník

Tento dotazník bude použit po otestování každého modulu.

1. Podařilo se Vám vytvořit nové zadání?
2. Bylo pro Vás použití editoru srozumitelné (pokud ne, popište body, které byly nejasné)?
3. Chyběla Vám u modulu nějaká vlastnost / funkce?

4.4.6 Testovací scénář modulu Rozlišování

Úkolem je vytvořit nové zadání pro hru Rozlišování. Cílem hry Rozlišování je přesunout do koše obrázky, které se nehodí do řady obrázků.

1. Přidejte nový úkol
2. Načtěte čtyři nové obrázky
3. Dva obrázky označte jako odlišné (obrázky, které bude žák označovat jako ty, co nepatří do řady obrázků)
4. Zadejte název úkolu
5. Uložte úkol

4.4.7 Testovací scénář modulu Vybarvování

Úkolem je vytvořit nové zadání pro hru Vybarvování. Cílem hry Vybarvování je správně vybarvit obrázek podle legendy.

1. Přidejte nový úkol
2. Načtěte nový obrázek (obrázek musí být černobílý)
3. Nastavte tři nové oblasti obrázku. Zadejte u nich název oblasti a vyberte barvu.
4. Zadejte název úkolu
5. Uložte úkol

4.4.8 Testovací scénář modulu Ukazování

Úkolem je vytvořit nové zadání pro hru Ukazování. Cílem hry Ukazování je správně ukazovat na části obrázku podle legendy.

1. Přidejte nový úkol
2. Načtěte nový obrázek
3. Přidejte tři nové oblasti obrázku a zadejte u nich názvy oblastí
4. Zadejte název úkolu
5. Uložte úkol

4.4.9 Testovací scénář modulu Skládání

Úkolem je vytvořit nové zadání pro hru Skládání. Cílem hry Skládání je poskládat obrázek, který je rozstříhán na dílky a proházen.

1. Přidejte nový úkol
2. Načtěte nový obrázek
3. Nastavte cvičení tak, aby žák skládal obrázek z 9 částí
4. Zadejte název úkolu
5. Uložte úkol

4.4.10 Testovací scénář modulu Figury a pozadí

Úkolem je vytvořit nové zadání pro hru Figury a pozadí. Cílem hry Figury a pozadí je najít tvary (obrázky) v obrázku, který je nějak překryt.

1. Přidejte nový úkol
2. Načtěte do horní řady dva obrázky s bílým pozadím
3. Načtěte do dolní řady další dva obrázky
4. Nastavte překrytí obrázků na horizontální a vertikální
5. Zadejte název úkolu
6. Uložte úkol

4.4.11 Testovací scénář modulu Obrácení

Úkolem je vytvořit nové zadání pro hru Obrácení. Cílem hry Obrácení je vybrat jeden odlišný obrázek a přesunout odlišný obrázek do koše.

1. Přidejte nový úkol
2. Načtěte nový obrázek
3. Nastavte úpravu obrázku na zrcadlit vertikálně
4. Nastavte počet zobrazovaných obrázků na 8
5. Zadejte název úkolu
6. Uložte nový úkol

4.4.12 Testovací scénář modulu Doplnování do mezery

Úkolem je vytvořit nové zadání pro hru Doplnování do mezery. Cílem hry Doplnování do mezery je do slova doplnit správné písmeno.

1. Přidejte nové slovo s jednou mezerou, ve kterém bude žák doplňovat písmeno „s“
2. Přidejte nové slovo se dvěma mezerami, ve kterém bude žák doplňovat písmena „vz“
3. Uložte zadání

4.4.13 Testovací scénář modulu Hodiny

Úkolem je vytvořit nové zadání pro hru Hodiny. Cílem hry je nastavit analogové hodiny podle textového určení času.

1. Přidejte nový úkol
2. Nastavte počet otázek na 4
3. Nastavte, aby se popis na hodinách ukazoval každou hodinu
4. Přidejte další úkol
5. Nastavte počet otázek na 3
6. Nastavte pouze nastavování minut
7. Uložte zadání

4.4.14 Testovací scénář modulu Délka slabik

Úkolem je vytvořit nové zadání pro hru Délka slabik. Cílem hry Délka slabik je určit kolik má dané slov slabik.

1. Přidejte vzor, který reprezentuje slova skládající se z dvou krátkých slabik
2. Přidejte do vytvořeného vzoru dvě slova, která odpovídají vzoru
3. Přidejte vzor, který reprezentuje slova skládající se ze dvou slabik, první krátké a druhé dlouhé
4. Přidejte do vytvořeného vzoru dvě slova, která odpovídají vzoru
5. Uložte zadání

4.4.15 Testovací scénář modulu Čísla

Úkolem je vytvořit nové zadání pro hru Čísla. Cílem hry Čísla je vybrat z nabídky takové číslo, které po součtu se zobrazeným číslem dá číslo zobrazené v horní části obrazovky.

1. Přidejte nový úkol
2. Nastavte počet otázek na 6
3. Nastavte, aby suma čísel byla v rozmezí od 8 do 12
4. Přidejte další úkol
5. Nastavte počet otázek na 3
6. Nastavte, aby suma čísel byla v rozmezí od 7 do 11
7. Nastavte, aby se místo čísel zobrazovaly tečky
8. Uložte zadání

4.4.16 Testovací scénář modulu Slovní úlohy

Úkolem je vytvořit nové zadání pro hru Slovní úlohy. Cílem hry Slovní úlohy je vyřešit nějakou slovní úlohu zaměřenou na počty.

1. Zadejte zadání slovní úlohy „Tomáš měl koš s 5 jablky. Jana měla 2 koše, ale každý se 3 jablky“
2. Nastavte, aby se cvičení žáka zeptalo na celkový počet jablek.
3. Nastavte grafické zobrazování slovní úlohy
 - a) Nastavte, aby se u Tomáše zobrazovalo, kolik má jablek (5)
 - b) Nastavte, aby se u Jany zobrazovalo, kolik má jablek a to způsobem, aby bylo vidět, že Jana má dvakrát po třech jablkách.
4. Nastavte, aby se cvičení zeptalo na otázku „Kolik má Jana celkem jablek?“
5. Uložte zadání

4.5 Výsledky testování

Testování je naplánované s učiteli a dětmi ze ZŠ Smečno, ale kvůli nedostatku času se nestihlo zrealizovat do doby odevzdání práce. Pro testování je vše připraveno, jsou připraveny jak testovací scénáře, tak i cvičení, na kterých se testování uskuteční. Testování tedy proběhne až po odevzdání práce a výsledky testování budou dostupné při obhajobě práce.

Závěr

Splnění cílů

1. Analyzujte požadavky pedagogů základních škol týkající se obsahu a forem jednotlivých cvičení.

Nejprve jsem prostudoval specifické poruchy učení a zrakovou percepci. Dále jsem analyzoval jednotlivé typy zrakové percepce a cvičení, které žákům pomáhají zrakovou percepci rozvíjet.

2. Analyzujte možnosti vytváření jednotlivých zadání pro cvičení a jejich nastavení.

V této části jsem analyzoval, jaké jsou možnosti pro ukládání zadání.

3. Navrhněte architekturu zásuvných modulů pro podporu graficky orientovaných cvičení a komunikaci s jádrem aplikace.

V tomto bodě jsem navrhl architekturu pro jednotlivé moduly a navrhl knihovnu, které se stará o komunikaci s jádrem aplikace. Tuto knihovnu následně využívají všechny moduly.

4. Navrhněte architekturu editoru pro vytváření cvičení pro jednotlivé moduly.

V této části jsme navrhl architekturu pro všechny moduly.

5. Implementujte alespoň 5 zásuvných modulů pro vybrané typy cvičení včetně příslušného GUI a jejich komunikaci s jádrem aplikace.

Podle návrhů jsem implementoval šest zásuvných modulů, včetně k nim příslušejících GUI.

6. Implementuje editor pro vytváření zadání pro jednotlivá cvičení všech 5 zásuvných modulů.

Podle návrhů jsem implementoval šest modulů, která umožňují vytvářet a editovat jednotlivá zadání pro moduly.

7. Hotové řešení podrobte vhodným testům a vyhodnoťte výsledky testování.

Všechny moduly jsem podrobil ručnímu testování. Dále jsem vytvořil testovací scénáře pro uživatelské testování, které bohužel neproběhlo do doby odevzdání této práce.

Osobní přínos

Díky tomuto projektu jsem si poprvé mohl vyzkoušet práci v týmu. Poznal jsem, jaké jsou výhody a nevýhody spolupráce. Zjistil jsem, jak je důležitá komunikace v týmu. Poznal jsem i jednotlivé úskalí při vytváření nového softwaru. Určitě to pro mne byla cenná zkušenost to budoucna.

Budoucnost projektu

Budoucnost projektu vidím v dalším vytváření nových modulů, které se budou zaměřovat na další problémy spojené s poruchami učení.

Literatura

- [1] Mazel, M.: *Dragon II - Plugins I*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.
- [2] Ondřej, F.: *Výuková aplikace Dráček II – Serverová část a rozhraní pro učitele*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.
- [3] Kovařovic, K.: *Gamifikace a personalizace výukové aplikace Dráček*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.
- [4] Matejček, Z.: *Dyslexie – specifické poruchy čtení*. H & H, 1995, ISBN 808578727X.
- [5] Jucovičová, P.: *Specifické poruchy učení a chování*. D+H, 2014, ISBN 978-80-7290-657-4.
- [6] Kohoutek, R.: *Psychologie v teorii a praxi - Vnímání a jeho závady a poruchy [online]*. 6. prosince 2008, [cit. 2016-08-05]. Dostupné z: <http://rudolfkohoutek.blog.cz/0812/vnimani-a-jeho-zavady-a-poruchy>
- [7] Olga, Z.: *Specifické vývojové poruchy čtení, psaní a dalších školních dovedností*. Praha: Portál, 2003, ISBN 80-7178-800-7.
- [8] Jiřina, B.; Šmardová Vlasta: *Školní zralost*. EDIKA, 2010, ISBN 9788025125694.
- [9] Krejčíková, E.: *Vývojová psychologie pro učitele*. Praha: SPN, 1986, ISBN 17-258-85.
- [10] Tomšů, R.: *Výuková aplikace Dráček – Vývoj vybraných aplikačních zásuvných modulů*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2013.

LITERATURA

- [11] W3C: *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. 7. února 2013, [cit. 2016-10-05]. Dostupné z: <https://www.w3.org/TR/REC-xml/>
- [12] *Introducing JSON*. [cit. 2016-10-05]. Dostupné z: <http://www.json.org>
- [13] Google: *Android NDK*. [cit. 2016-09-05]. Dostupné z: <https://developer.android.com/tools/sdk/ndk/index.html>

Seznam použitých zkratk

GUI Graphical User Interface

XML Extensible Markup Language

JSON JavaScript Object Notation

API Application Programming Interface

SDK Software Development Kit

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
apk.....	adresář s instalačními balíčky apk
src	
_ impl.....	zdrojové kódy implementace
_ thesis	zdrojová forma práce ve formátu L ^A T _E X
text	text práce
_ BP_Bures_Michal_2016.pdf	text práce ve formátu PDF
_ user_guide.pdf	uživatelská příručka ve formátu PDF