

České vysoké učení technické v Praze

Fakulta elektrotechnická

BAKALÁŘSKÁ PRÁCE

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačové grafiky a interakce

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Jan Teplý**

Studijní program: Softwarové technologie a management
Obor: Web a multimedia

Název tématu: **Interaktivní učebnice pro ANDROID**

Pokyny pro vypracování:

Analyzujte a porovnejte existující aplikace pro interaktivní výuku na mobilních zařízeních s různým operačním systémem. Aplikujte nabyté poznatky v rámci společného projektu tvorby interaktivní učebnice pro střední školy, která se bude skládat z jednotlivých interaktivních úloh demonstrujících vybraná témata z biologie.

Klasifikujte vám přidělené úlohy dle požadavků na uživatelské rozhraní. Navrhněte a implementujte tyto úlohy, primárně se soustředte na intuitivní uživatelské rozhraní a na znovu-použitelnost vámi vytvořené implementace. Úlohy zakomponujte do společně vytvářené výukové aplikace. Implementaci proveďte pro systém ANDROID 4 a vyšší, pro návrh uživatelského rozhraní předpokládejte velikost zařízení minimálně 9". Finální aplikaci otestujte s cílovou skupinou a zhodnoťte použitelnost vámi vytvořených částí aplikace.

Tématická náplň úloh bude přesně specifikována odborníky z Přírodovědecké Fakulty UK. Podklady pro grafickou úpravu a skupiny testerů dodá vedoucí práce.

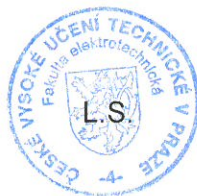
Seznam odborné literatury:

Satya Komatineni, Dave MacLean, Pro Android 4, Apress, 2012.

Vedoucí: Ing. David Sedláček, Ph.D.

Platnost zadání: do konce letního semestru 2014/2015


prof. Ing. Jiří Žára, CSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 21. 2. 2014

ABSTRAKT

Práce se zabývá postupem tvorby aplikace pro podporu výuky biologie pro tablety s operačním systémem Android. Popisuje návrh architektury, řešení implementace zadaných cvičení pro studenty a testování aplikace s uživateli. Zpracovávaná část aplikace obsahuje několik typů úloh, které budou žáci plnit při postupném přibližování se od pohledu na celý svět až na úroveň atomu. Tato část je součástí větší aplikace, na které spolupracuje několik dalších lidí.

KLÍČOVÁ SLOVA

Android, Tablet, Aplikace, Podpora výuky, Interaktivní učebnice

ABSTRACT

This bachelor thesis discuss about process of developing application for support in teaching of biology for tablets with Android operating system. Work describes architecture design, solution of implementation of given exercises for students and usability testing of application with users. The processed part of the application contains several types of task that students will perform during zooming through from view of the whole world to level of the atom. This section is part of larger application, on which works team of other people.

KEYWORD

Android, Tablet, Aplikation, Teaching support, Interactive schoolbook,

TEPLÝ, Jan. *Interaktivní učebnice pro ANDROID*. Praha: České vysoké učení technické v Praze, Fakulta elektrotechnická. Bakalářská práce. Vedoucí práce: Ing. David Sedláček

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma *Interaktivní učebnice pro ANDROID* jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Praze dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Davidovi Sedláčkovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Praze dne

.....

(podpis autora)

OBSAH

Úvod	7
1 Operační systém android	8
1.1 Architektura.....	8
1.2 Stavební prvky aplikace	9
1.2.1 Activity.....	10
1.2.2 Resources	11
1.2.3 Fragment.....	12
1.3 Vývoj pro Android	13
1.3.1 Android Software Development Kit (SDK).....	13
2 Aplikace pro podporu výuky	14
2.1 Příklady aplikací na Google Play	14
2.2 Závěry z analýzy	18
3 Architektura části aplikace	19
3.1 ZoomActivity	19
3.2 ZoomFragment	19
4 Implementace	21
4.1 ZoomActivity	21
4.1.1 Příprava ZoomFragmentů	22
4.1.2 Detekce gest	22
4.1.3 Ukládání stavu aplikace	23
4.2 ZoomFragment	24

4.2.1	Rekurzivní nastavení aktivity prvků UI.....	24
4.2.2	Inicializace nápovědy	25
4.2.3	Nastavení stavu fragmentu.....	26
4.2.4	Zobrazení zprávy uživateli.....	26
4.3	OpenQuestionFragment	26
4.3.1	TextView a EditText.....	27
4.3.2	ImageView	28
4.4	NumberQuestionFragment.....	28
4.5	ChoiceQuestionFragment	29
4.5.1	RadioButton a CheckBox	30
4.6	DragDropQuestionFragment	30
4.6.1	TouchableSpan a LinkTouchMovementMethod	31
4.6.2	Drag and Drop	32
4.6.3	AspectRatioImageView	33
4.6.4	Barevná mapa	33
4.7	ClickQuestionFragment	34
4.7.1	Detailní popis skládání obrázku a kontrola.....	35
4.7.2	LayerDrawable.....	36
4.8	StateClickQuestionFragment	36
4.8.1	TransitionDrawable	37
4.9	ChemicalFormulaQuestionFragment.....	38
4.9.1	Animace	39
4.10	CrossWordsQuestionFragment.....	40
4.10.1	Generování křížovky.....	41
4.10.2	Problém českého písmena CH a posouvání po polích	42

4.10.3	Problémy s backspace	42
4.11	VideoFragment.....	43
5	Testování s uživateli	44
5.1	Testování v laboratorních podmínkách.....	44
5.1.1	Pre-test dotazník.....	44
5.1.2	Průběh testování	45
5.1.3	Výsledky testování	45
5.1.4	Post-test dotazník	47
5.1.5	Video z testování.....	48
5.2	Testování ve výuce.....	48
5.3	Testování aplikace Vinařická horka v terénu.....	50
5.4	Závěry z testování	51
5.5	Závěr.....	52
	Literatura	53
	Seznam Obrázků	55
	Seznam Tabulek	57
	Seznam zkratk a pojmů	58
	Přílohy	59
A	Testování	59
A.1	Výsledky post-test dotazníku k celé aplikaci	59
A.2	Poznámky z testování Vinařická horka.....	64
B	Obsah CD	66

ÚVOD

Tato práce se zabývá postupem tvorby aplikace pro podporu výuky biologie pro tablety s operačním systémem Android. Tato aplikace vzniká jako součást projektu Střední průmyslové školy stavební a Obchodní akademie v Kladně. Jedním z cílů projektu je zatraktivnění výuky přírodovědných a technických předmětů na školách ve Středočeském kraji. Toho se snaží dosáhnout nákupem nového vybavení laboratoří a také tabletů využívaných při výuce.

Aplikace, tvorba jejíž části je popsána v této práci, se zabývá fotosyntézou a obsahuje několik sekcí s úkoly, sekci s pokusem měření koncentrace CO_2 a O_2 a průvodce pozorováním mikroskopem. Zde je popsána část aplikace nazvaná „Od pralesa k atomu“. Pracovně je nazvána „zoomy“. Student při plnění úkolů prochází pomocí gesta pinch problematikou od vnějšku až na úroveň atomu. Tento modul je tvořen tak, aby šel snadno použit i v jiné aplikaci nebo byl snadno modifikován o nové úkoly.

Grafické a obsahové podklady byly zprostředkovány vedoucím práce. Stejně tak testování na střední škole v Kladně. Na dalších částech pracovali Josef Veselý, Jiří Rychlovský a Ing. David Sedláček. Autor práce je autorem tříd, které lze v projektu nalézt v balíčku `cz.scientica.kladnobiologie01.zoom`. U tříd jejichž základ je přejat z webových zdrojů je tato skutečnost uvedena v komentáři na začátku souboru.

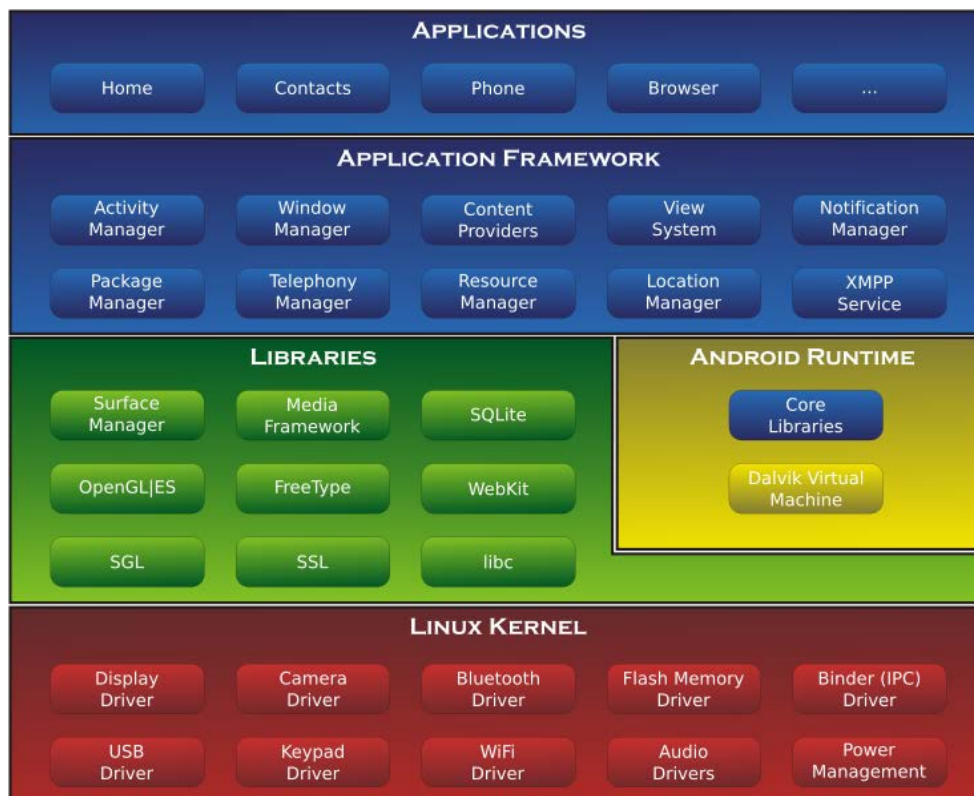
1 OPERAČNÍ SYSTÉM ANDROID

Aplikace, kterou se zabývá tato práce je vytvářena pro operační systém Android ve verzi 4.0 a vyšší. Android je operační systém postavený na Linuxovém jádře. Od roku 2007 je platforma Android vyvíjena společenstvím Open Handset Alliance v čele se společnostmi jako je Google, HTC či Samsung společně s dalšími výrobci mobilních zařízení a softwaru.

Android je v roce 2013 nejrozšířenějším operačním systémem pro chytré telefony a tablety. [1]

1.1 Architektura

Operační systém je rozdělen do 5 vrstev (viz obr. 1.1), které mají rozdílný účel, nemusí však být od sebe přímo oddělené [2].



Obr. 1.1: Vrstvy operačního systému Android.

Jádro operačního systému

Základem systému je Linuxové jádro ve verzi 2.6. Systém využívá podpory správy sítí, paměti a procesů nebo zabudované ovladače hardware. Jádro odděluje hardware zařízení od dalších softwarových vrstev operačního systému.

Knihovny

Druhou vrstvou systému jsou knihovny poskytující služby prostřednictvím Application Framework. Knihovny jsou psané v jazyce C nebo C++. Jako příklad lze uvést knihovny SQLite pro práci s databází nebo Media Libraries pro práci s video, audio a obrazovými formáty.

Android Runtime

Android Runtime je vrstva obsahující virtuální stroj Dalvik. Dalvik Virtual Machine nahrazuje kvůli licencím a lepší optimalizaci pro běh na mobilních zařízeních Java Virtual Machine. Tato vrstva obsahuje další knihovny Javy, knihovny pro uživatelské rozhraní a knihovny Apache pro práci se sítí. Každá spuštěná aplikace běží ve vlastním procesu s vlastní instancí DVM.

Application Framework

Nejdůležitější vrstvou pro vývojáře je Application Framework. Zpřístupňuje množství služeb, které lze využít v aplikaci. Například prvky potřebné k sestavení uživatelského rozhraní, resource manager pracujících se zdroji dat aplikace jako jsou obrázky či řetězce nebo activity manager řídící životní cyklus aplikace.

Aplikace

Nejvyšší vrstvu systému tvoří aplikace. Ať už základní SMS klient, mapy nebo kontakty tak i další aplikace stažené s Google Play. Aplikace jsou velice důležitou součástí ekosystému Android.

1.2 Stavební prvky aplikace

Základními stavebními prvky aplikace pro operační systém android jsou Activity reprezentující obrazovku, Services umožňující provádět akce na pozadí, Content

providers poskytující přístup k datům a Broadcast receivers reagující na příchozí oznámení [2].

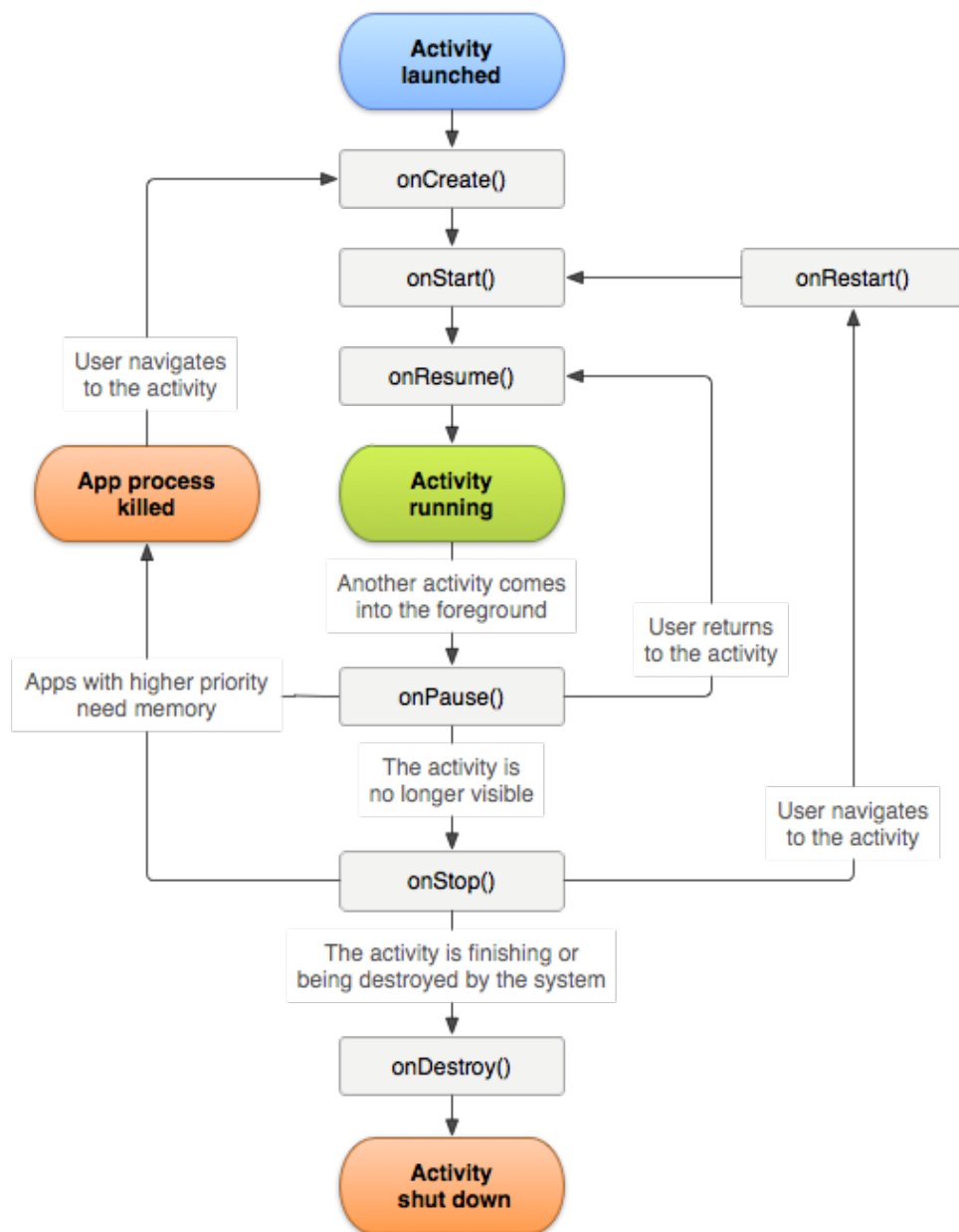
Všechny tyto komponenty musí být definovány v souboru AndroidManifest.xml, uloženém v kořenovém adresáři projektu. V tomto projektu bylo použito zejména Activity. Dalšími stavebními prvky aplikace jsou prvky View, ze kterých se skládá uživatelské rozhraní a Resources obsahující obrázky, řetězce či další data využívaná v aplikaci.

1.2.1 Activity

Activity by měla odpovídat jedné obrazovce aplikace. Zobrazuje grafické rozhraní a zpracovává uživatelské akce. Vytvoření aktivity je náročné na prostředky systému proto Android obsahuje ActivityManager. ActivityManager vytváří, ruší a zpravuje celý životní cyklus aplikace. Spuštěné aktivity jsou uchovány v zásobníku, ze kterého jsou postupně vyvolávány.

Activity se může nacházet v následujících stavech [3]. Při přechodu mezi stavy jsou postupně volány metody tak, jak popisuje schéma na obrázku 1.2.

- Activity starts – počátek, kdy je Activity inicializována.
- Activity is running – Activity je zobrazena na displeji a může mít interakci s uživatelem. V jediném okamžiku může být právě jedna Activity v tomto stavu.
- Process is killed – Activity Manager zrušil Activity z důvodu nedostatku paměti. K této akci může dojít, pokud Activity není viditelné. Další možnost není tak obvyklá – Activity je viditelné, ale uživatel s ní nemůže navázat interakci (nastává například při dialogových hláškách).
- Activity is shut down – Activity Manager ukončil Activity a ta již nevyužívá žádnou paměť.



Obr 1.2: Životní cyklus Activity.

1.2.2 Resources

Sobory jako obrázky a řetězce jsou na Androidu ukládány odděleně od kódu ve složce res. To umožňuje tyto soubory snadno měnit nebo podmíněně vybírat podle modifikátoru v názvu podsložky. Zdroje obsahují následující typy dat podle názvu složek [4].

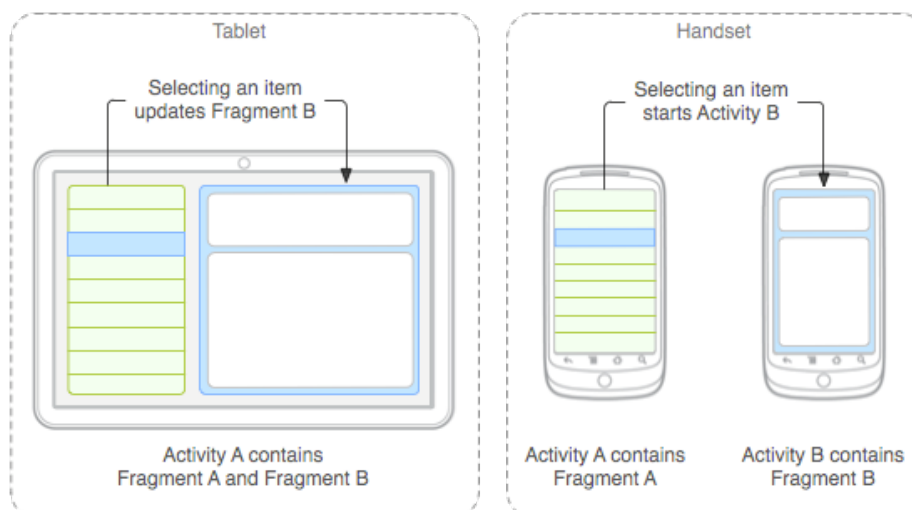
- Drawable – obsahuje bitmapové obrázky, nebo XML soubory převeditelné na Drawable. K souborům se přistupuje pomocí identifikátoru R.drawable.nazev.

- Layout – obsahuje XML soubory s popisem uživatelského rozhraní. K souborům se přistupuje pomocí identifikátoru `R.layout.nazev`.
- Menu – obsahuje XML soubory s definicemi menu. K souborům se přistupuje pomocí identifikátoru `R.menu.nazev`.
- Raw – může obsahovat jakékoliv soubory, které lze načíst pomocí `InputStream`. K souborům se přistupuje pomocí identifikátoru `R.raw.nazev`.
- Values – tato složka obsahuje XML soubory obsahující jednoduché řetězce, čísla, hodnoty barev nebo například styly. Identifikátor se takových dat má tvar `R.typ_dat.name`, například `R.string.error_message`.

Každá složka může mít dále v názvu modifikátor, podle kterého lze například určit layout pro různé orientace zařízení nebo řetězce pro různé lokalizace. Prioritně jsou data vybírána ze složek s nejvíce specifickými modifikátory.

1.2.3 Fragment

Rozšířením operačního systému na tablety vznikla potřeba definovat různá sestavení aplikace pro různé velikosti displeje. Proto byl do Android API přidán Fragment. Fragment je mezistupněm mezi Activity a View. Lze s ním pracovat jako s View s přidávat ho do Layoutu Activity. Fragment však může obsahovat vlastní logiku pracující s jeho obsahem. Příklad použití fragmentu lze vidět na obrázku 1.3 [5].



Obr. 1.3: Příklad použití fragmentu na různých velikostech displeje.

Fragmenty mají svůj životní cyklus podobný jako Activity. Volání metod životního cyklu Fragmentu je závislé na volání metod Activity, do které je Fragment přidán.

1.3 Vývoj pro Android

Aplikace byla vyvíjena v IDE Eclipse, pro které je vydána oficiální podpora a SDK. Nově je pro vývoj možné využít také Android Studio, které je dostupné verzi Early Access Preview. Lze ale využít i jiný IDE [2].

1.3.1 Android Software Development Kit (SDK)

Nástroje potřebné pro vývoj aplikací pro platformu Android obsahuje SDK, který je dostupný pro Linux, Windows a Mac OS. Sadu SDK je možné stáhnout v několika konfiguracích.

SDK v základní konfiguraci obsahuje například nástroje pro debugging, emulátor, nástroj umožňující instalaci přímo do zařízení a zejména platformy skládající se z knihoven API a jiných zdrojů podle verze platformy [2].

2 APLIKACE PRO PODPORU VÝUKY

Interaktivní učebnice na tabletech jsou dalším logickým krokem po prezentacích a interaktivních tabulích používaných ve výuce na mnohých školách. Prostřednictvím tabletu se může žák dostat snadno k informacím, které se v klasických učebnicích nesyadno zobrazují. Mohou to být například 3D vizualizace či animace a videa.

Dalším prostředkem výuky na tabletu je možnost interaktivních cvičení a automatickou kontrolou zadaných odpovědí. To umožňuje studentům pracovat samostatně či ve skupinách bez kontroly učitele či hledání odpovědí na zadní straně učebnice.

Další velice užitečnou vlastností tabletů je možnost přehrávání zvuku. Toho lze v kombinaci se sluchátky velice dobře využít při výuce cizích jazyků. Například pro poslech delších textů, rozhovorů nebo předčítání slovíček kvůli výslovnosti.

Jako motivace ve výukových aplikacích může sloužit forma hry a sociální funkce. Student tak například ztrácí životy při špatných odpovědích, nebo naopak při správných získává zkušenosti potřebné k otevření dalších částí aplikace. Díky systému životů se nedá aplikaci snadno obelstít náhodným klikáním a student je tak donucen při svých odpovědích opravdu přemýšlet. Pokud se mu to i tak nepovede, při dalším opakování si již bude pamatovat, jak měl odpovědět a naučí se tak danou věc. Výsledky také u některých aplikací lze sdílet s přáteli a ostatními studenty a tak soupeřit o lepší výsledky v pořadí.

Takovéto aplikace nemusí žáci používat jen ve škole, ale díky automatické kontrole mohou pohodlně studovat i z domu.

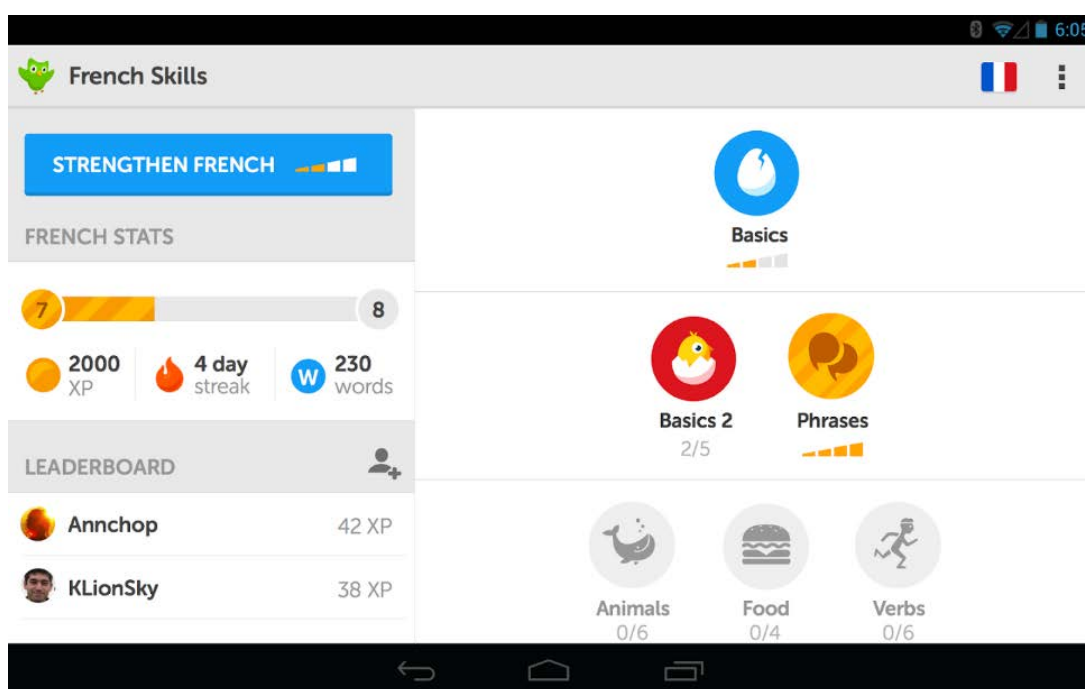
2.1 Příklady aplikací na Google Play

Na Google Play lze takových aplikací najít velké množství. Velká část z nich cílí na menší děti a také na výuku cizích jazyků. Zde předkládáme několik příkladů těchto aplikací.

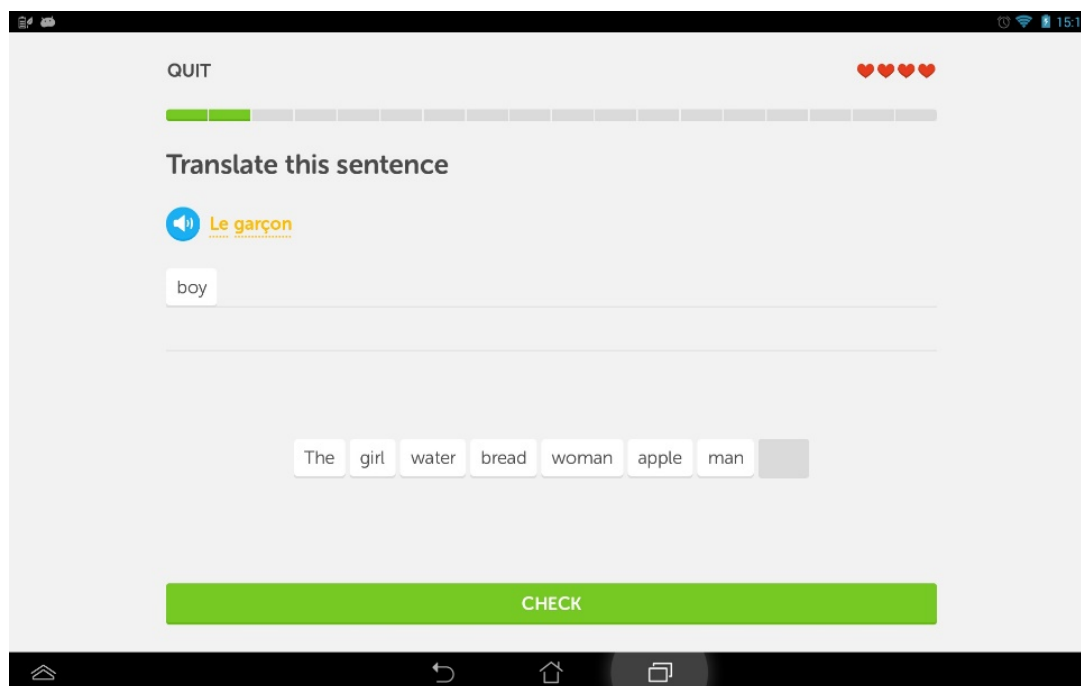
Duolingo: Learn Languages Free

Duolingo je aplikace pro učení cizích jazyků[1]. Lokalizace do češtiny není dostupná,

proto je potřeba umět anglicky či některý z nově přidaných jazyků. Aplikace podporuje výuku španělštiny, francouzštiny, němčiny, portugalštiny, italštiny a angličtiny. Duolingo využívá k výuce různá doplňovací, poslechová a překládací cvičení. Podporuje také rozpoznávání řeči za účelem naučení správně výslovnosti. Celá výuka má nádech hry, protože je využito systému životů, sbírání zkušeností i žebříčků mezi přáteli. To uživatele motivuje k dalšímu učení (viz obrázky 2.1 a 2.2). Aplikace také každý den připomíná, aby uživatel splnil další cvičení.



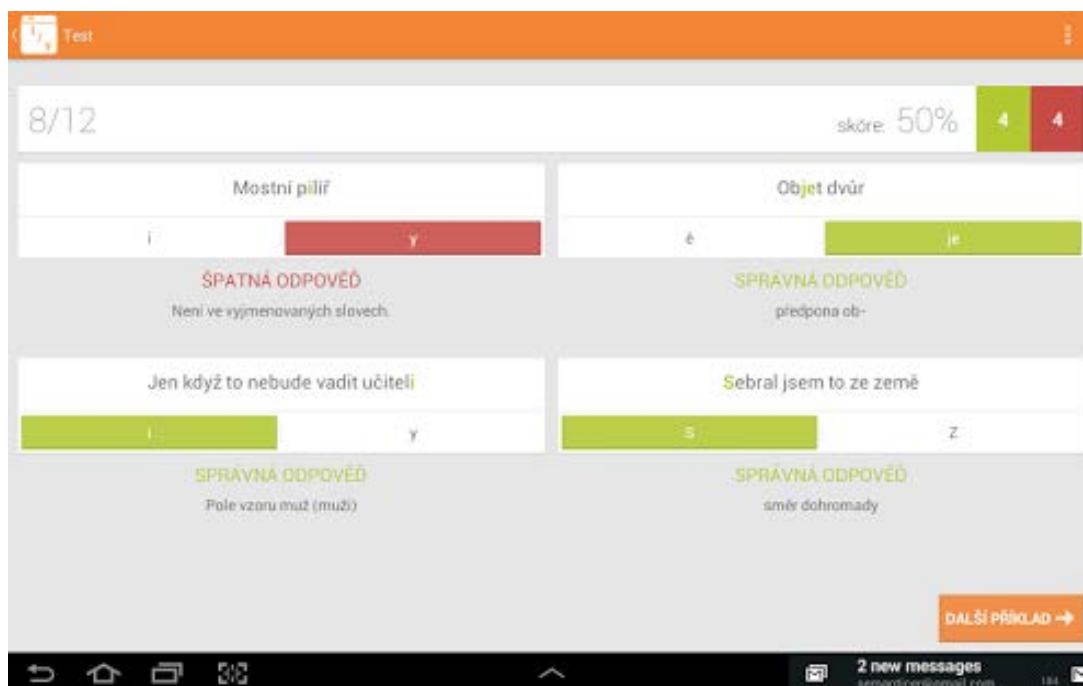
Obr. 2.1: Hlavní menu aplikace Duolingo. V levé části vidíme počítadlo zkušeností a tabulku s pořadím mezi přáteli. V pravé pak výběr mezi tematickými okruhy.



Obr. 2.2: Pro výuku aplikace využívá různé druhy cvičení. Na obrázku je zobrazeno překládání slovních spojení. Vpravo nahoře se nachází počítadlo životů, které jsou odečítány za špatné odpovědi.

Naučte se pravopis

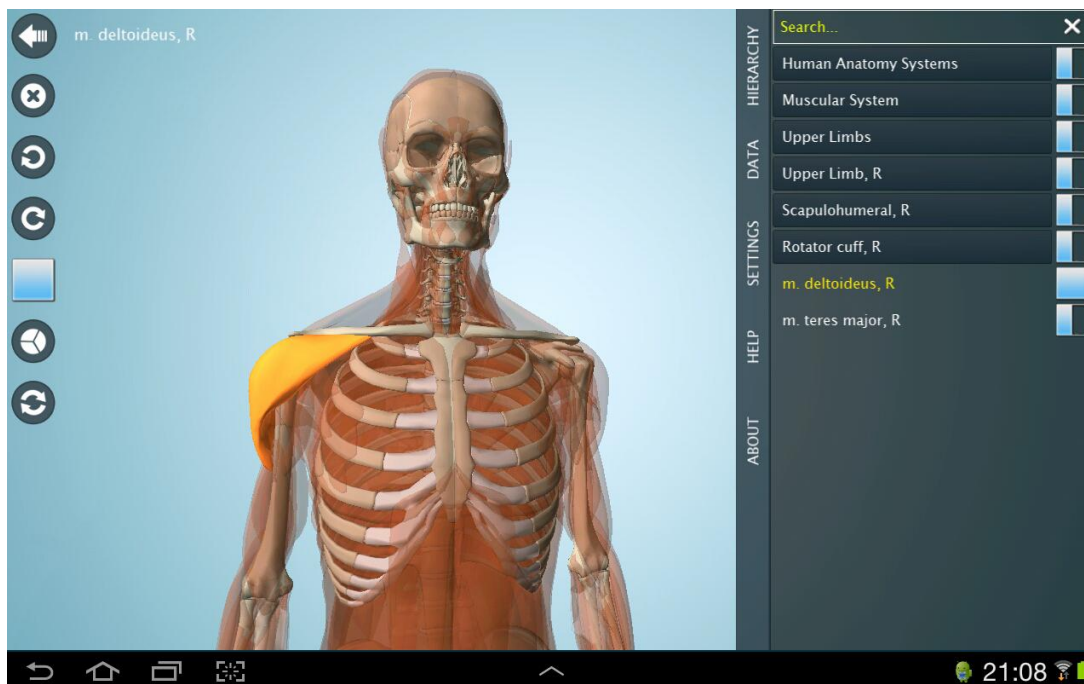
Česká aplikace Naučte se pravopis je určena k učení českého pravopisu za využití doplňovacích cvičení[7]. Uživatel si zvolí z jakých okruhů a kolik doplňovacích otázek chce připravit. Aplikace vygeneruje sadu otázek, které po vyplnění ihned kontroluje a zobrazí odůvodnění správné odpovědi (viz obrázek 2.3). Na konci testu se uživateli zobrazí podrobná statistika, toho jak se mu vedlo v jednotlivých okruzích. Uživatel sbírá za splněné testy ocenění a může také soupeřit v žebříčcích se svými přáteli. Pokud si neví uživatel rady, může nahlédnout do přiložených taháků na vyjmenovaná slova a vzory. Aplikace bohužel na použitém tabletu nefungovala, nebylo ji tedy možné více testovat.



Obr. 2.3: V doplňovacích cvičeních uživatel vybírá, jaké písmeno se má doplnit. Aplikace zadání ihned kontroluje a zobrazí také zdůvodnění správné odpovědi.

Anatomy 3D – Anatomica

Aplikace Anatomy 3D obsahuje vizualizaci lidského těla [8]. Uživatel může tělo prohlížet z různých stran, zobrazovat jen jednotlivé části nebo systémy organismu. Jednotivé části lze také zobrazit poloprůhledně (viz obrázek 2.4). Uživatel tak získá názornější pohled na anatomii nežli z obrázku v učebnici. Po označení některé z částí systému jsou navíc zobrazeny informace o tomto svalu, orgánu nebo této kosti.



Obr. 2.4: 3D vizualizaci lidského těla lze zobrazovat po jednotlivých částech. Lze také nastavit části těla poloprůhledné a vidět tak, co se nachází pod nimi.

2.2 Závěry z analýzy

Analýza tématu byla využita k získání přehledu v této kategorii aplikací a pro inspiraci při budoucí implementaci. Obsah aplikace, grafické elementy a způsob procházení byly předepsány přímo zadavatelem práce a poznatky z této analýzy na ně měli minimální vliv.

3 ARCHITEKTURA ČÁSTI APLIKACE

Architektura byla zvolena tak, aby byla tato část aplikace co nejvíce modulární a znovupoužitelná. Aplikace je to totiž součástí většího projektu a lze předpokládat, že podobný princip bude využit i v aplikaci na jiné učební téma.

Základní prvky architektury vychází z pokynů k psaní aplikací pro systém Android [9]. Aplikace využívá jednotlivých instancí třídy Activity ke správě zobrazovaného obsahu, tedy jednotlivých Fragmentů. Fragments definují jak úkol či jiná část aplikace vypadá a jak se chová.

3.1 ZoomActivity

Hlavním prvek je řídicí třída ZoomActivity, která se stará o vytváření a zobrazování fragmentů třídy ZoomFragment. V původním návrhu existovala ještě třída ZoomManager, která spravovala vytvořené fragmenty uložené v listu, starala o se o jejich spouštění ve správném pořadí a ukládání stavů fragmentů. Tato třída byla později integrována do BaseActivity a těchto služeb využívají i ostatní části aplikace.

3.2 ZoomFragment

Potomci třídy ZoomFragment představují jednotlivé úkoly pro studenta. Jsou navrženy tak, aby s nastavením jiných zdrojů mohli představovat jiné úlohy stejného typu. Mohou být proto v aplikaci použity vícekrát a není nikde duplikována celá třída jen kvůli jiným zdrojům dat.

Lze předpokládat, že podobný systém přibližování skrze vyučovanou problematiku bude využit i u jiných témat. Zde je možné využít stejné fragmenty jako u zoomů v aplikaci o fotosyntéze. Pokud se objeví nový typ úkolu, stačí ho napsat jako potomka třídy ZoomFragment a držet se stejné formy jako u ostatních fragmentů. ZoomFragment nejsou nijak závislé na samotném zoomování a lze je použít i v jiných částech aplikace. V části „podmínky“ je například využit ClickQuestionFragment.

Potomci třídy ZoomFragment

Jednotlivé typy úloh byly navrženy podle požadavků vytvořených pedagogy tak, aby co nejlépe sloužily ve výuce. Podle těchto požadavků byly určeny typy úloh, které se v aplikaci budou vyskytovat. Obsah a funkčnost těchto jednotlivých Fragmentů popisuje kapitola Implementace.

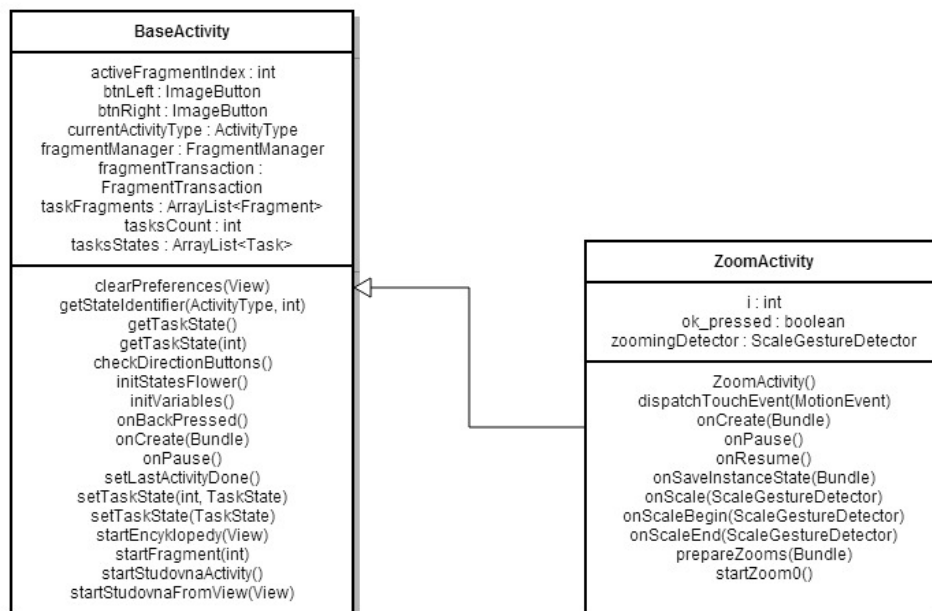
4 IMPLEMENTACE

Aplikace byla implementována v jazyce Java a využívá třídy z Android API. Skládá se z jednotlivých fragmentů, které jsou instancí ZoomFragment a hlavní třídy ZoomActivity, která má na starost správu Fragmentů, zobrazování Fragmentů a globální ukládání dat.

Tato kapitola se zabývá nejdříve implementací ZoomActivity a následně popisuje implementaci jednotlivých typů ZoomFragmentů.

4.1 ZoomActivity

Tato třída je hlavní aktivitou části aplikace zoomy. Dědí od základní BaseActivity, která má na starost propojení se zbytkem aplikace a správu připravených fragmentů a jejich stavů. BaseActivity je společným předkem všech Activity v aplikaci (viz obr. 4.1). ZoomActivity se stará konkrétně o část zoomy. Vytváří instance Fragment a plní je daty a také detekuje gesto pro přechod mezi fragmenty.



Obr. 4.1: Diagram tříd BaseActivity a ZoomActivity s výpisem metod

4.1.1 Příprava ZoomFragmentů

Příprava fragmentů probíhá tak, že je nejprve získána nová instance příslušného ZoomFragmentu, tomuto fragmentu jsou přidány příslušné argumenty definující jeho obsah a fragment je přidán do `ArrayList taskFragments` v `BaseActivity`. Z toho `ArrayListu` jsou fragmenty následně vybírány pro spuštění.

K přípravě fragmentů slouží metoda `prepareZooms()`. Do argumentů je například přidáno pozadí fragmentu, znění otázky, možné správné odpovědi a pomocné texty. Pokud má aplikace uložený předchozí stav fragmentu je nyní přidán do argumentů také. Následně je fragment vložen do příslušného `ArrayList` a je připraven k použití.

4.1.2 Detekce gest

Důležitým prvkem zoomů je zoomování. Tedy gesto pinch, které je využito k přechodu mezi fragmenty. Android API nabízí pro účely detekce gesta třídu `ScaleGestureDetector` a interface `ScaleGestureListener` [10]. Třída `ZoomActivity` přímo implementuje `ScaleGestureDetector` a jeho tři metody. Tento listener je zavěšen na `ScaleGestureDetector` a pokud je toto gesto detekováno, je nejdříve náležitě přiblížen nebo oddálen zobrazený fragment a poté je aktivní fragment nahrazen následujícím respektive předchozím fragment. Z důvodu použití `ScrollView` pro zobrazování fragmentů je nutné přepsat metodu `dispatchTouchEvent()`. Pokud `MotionEvent` obsahuje právě dva ukazatele na dotyky je `MotionEvent` poslána ke zpracování `ScaleGestureDetector`em. Pokud není gesto detekováno nebo má `MotionEvent` jiný počet pointerů než dva, je událost delegována na předka třídy `ZoomActivity`.

`ScaleGestureLisneren` naslouchá třem stavům gesta. Jeho začátek v metodě `onScaleBegin()`, jeho průběh v metodě `onScale()` a jeho zakončení v metodě `onScaleEnd()`. V metodě `onScaleBegin()` je nejprve nastaven příslušnému zoomovanému `View` pivot, kolem kterého je `View` následně zvětšováno.

Při volání metody `onScale()` je `View` škálováno druhou mocninou detekovaného faktoru. Druhá mocnina je použita, protože při použití samotného faktoru byla změna velikosti příliš malá. Tato metoda je volána průběžně při pohybu prstů po obrazovce a `View` je tak škálováno plynule.

Při ukončení gesta je zkontrolováno, zda je potvrzeno přečtení instrukcí k průchodu aplikací. Pokud je velikost `scaleFactor` větší než jedna znamená to, že má být zobrazen další úkol. Předtím je potřeba zkontrolovat, zda je to možné. Aby bylo možné pokračovat na další fragment, musí být aktuální úkol na fragmentu dokončený. Tuto kontrolu lze pouhým zakomentováním vypnout, čehož je využíváno pro zrychlení procházení při programování úkolů. Fragment také nesmí být posledním v řadě.

Je-li velikost `scaleFactor` větší než jedna má být zobrazen předcházející fragment. Pro postup aplikací zpět není žádné omezení na splnění úkolů. Důležité je však zkontrolovat, jestli fragment není první v pořadí.

Pokud nastane situace, že aktuální fragment a stav neprojde žádnou z kontrol, je velikost zvětšení fragmentu nastavena zpět na faktor 1, tedy původní velikost před škálováním.

Detekce gest způsobovala při testování problémy a právě při zoomování aplikace často padala. Nejčastěji to bylo způsobenou špatným počtem pointerů asociovaných s událostí. Tomu stavu je předcházeno kontrolou počtu pointerů.

Dalším problémem při implementaci zoomování byl výběr View, které má být škálováno. Nejprve byl zvětšován jen obrázek na pozadí, při tomto postupu byl ale obrázek nepřírozeně překrýván dalšími prvky uživatelského rozhraní nebo je sám překrýval. Dalším postupem bylo škálování kontejneru, ve kterém je zobrazován příslušný fragment. To mělo za následek, že se velikost po výměně fragmentu nevracela do původního stavu či se vracela dříve, než byl fragment nahrazen. Posledním a nakonec využitým řešením je škálování křenového layoutu jednotlivých fragmentů. Tyto layout jsou opatřeny id `zoom_fragment` a pro správnou funkci je nutné, aby toto id obsahovaly všechny nově vytvořené layouty fragmentů.

4.1.3 Ukládání stavu aplikace

Aby uživatelé nepřicházeli o svou rozdělanou práci na úkolech, je nutné ukládat stav jednotlivých fragmentů. Při výměně zobrazených fragmentů si ZoomFragmenty svůj stav ukládají sami do argumentů. Pokud je však Activity ZoomActivity ukončena nebo překryta jinou Activity, je nutné uložit stav centrálně. K tomu slouží metoda

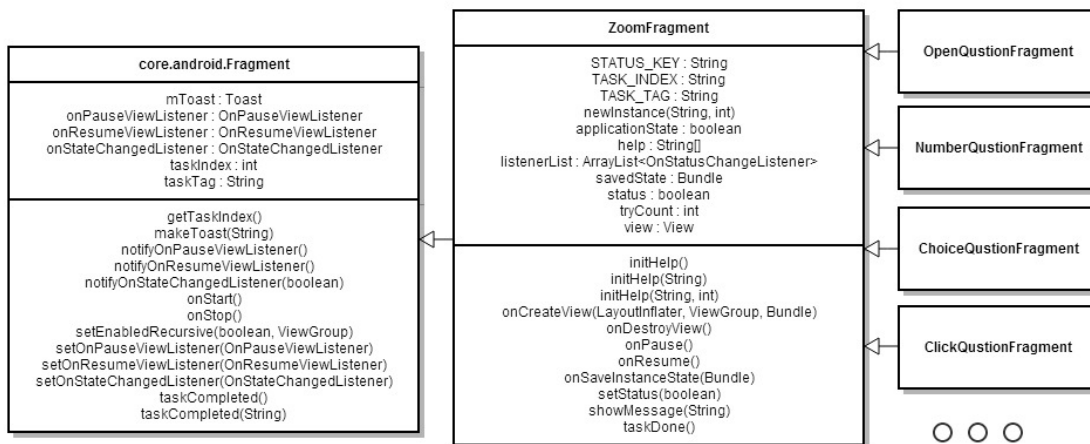
onSaveInstanceState() [3].

Postupně je u všech fragmentů vyvolána metoda `onSaveInstanceState()` a stav je uložen do `Bundle`, který je následně uložen do statické proměnné ve třídě `AppState`.

Při novém startu aplikace je uložený stav obnoven a vložen do argumentů nově vytvořených fragmentů v metodě `prepareZooms()`.

4.2 ZoomFragment

Třída `ZoomFragment` je společným předkem fragmentů pro jednotlivé úkoly. Je potomkem třídy `Fragment` z balíčku `core.android`. Tento předek implementuje podporu listenerů a ukládá `taskTag` a `taskIndex` fragmentů. Tyto dvě proměnné se používají k identifikaci fragmentu. Tyto třídy obsahují důležité metody popsané v dalších podkapitolách (viz obr. 4.2).



Obr. 4.2: Diagram tříd týkajících se Fragmentů.

4.2.1 Rekurzivní nastavení aktivity prvků UI

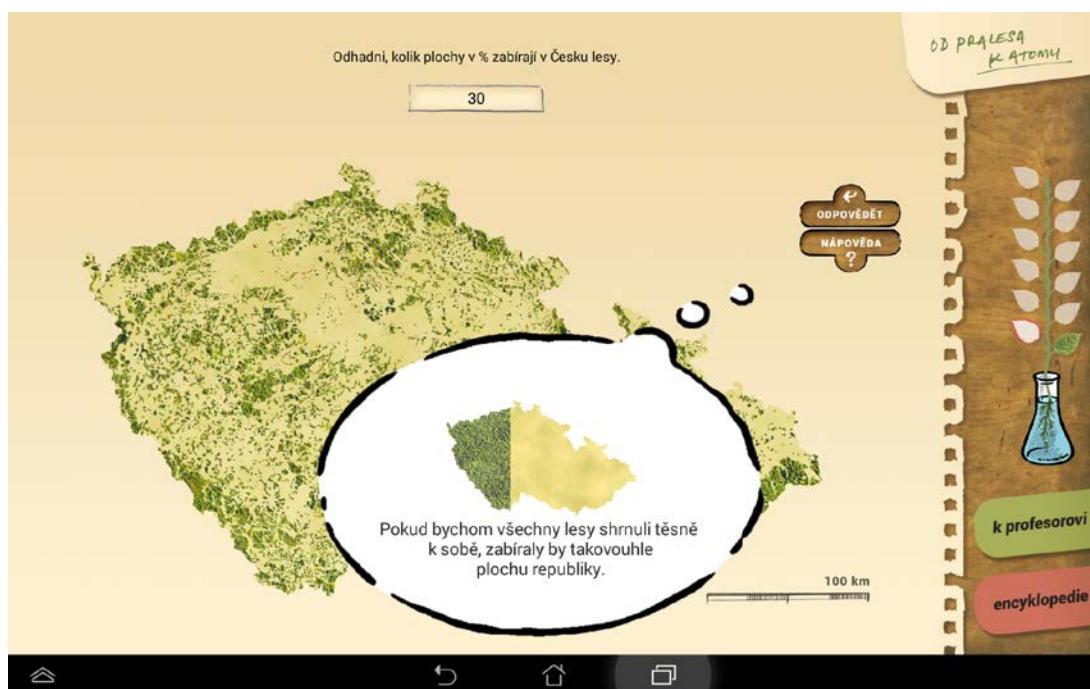
Užitečnou metodou ve třídě `Fragment` je `setEnabledRecursive()`. Tato metoda je použita na nastavení aktivity prvků uživatelské rozhraní fragmentu. Prochází do hloubky strom `View` obsažených v zadané `ViewGroup` a nastaví `setEnabled()` na zadanou hodnotu.

4.2.2 Inicializace nápovědy

Pokud si uživatel neví rady, má možnost zobrazit si nápovědu k úkolu. Nápovědu je možno zobrazovat ve všech fragmentech, proto byla její inicializace přesunuta do `ZoomFragmentu`. K inicializaci nápovědy je připravena třikrát přetížená metoda `initHelp()`.

Metoda `initHelp()` bez zadaných parametrů nejprve načte pomocné řetězce a případně id obrázku z argumentů a poté rozhodne, jak má být nápověda zobrazena.

Dále je volána metoda `initHelp()` s jedním nebo dvěma parametry. Prvním z nich je text nápovědy. Druhým argumentem je případný obrázek. Následující kód představuje metodu `initHelp()` s oběma parametry. Pokud je parametrem jen text, je nastaven jiný layout, který neobsahuje `ImageView`. Nejprve je zobrazeno tlačítko pro zobrazení nápovědy a nastaveno jeho chování po kliknutí. Následně je `ViewStub` s id `napoveda_bubble` nahrazeno připraveným layoutem. Do tohoto layoutu je nastaven zadaná text a obrázek. Po té je nápověda připravena k zobrazení (viz obr. 4.3).



Obr. 4.3: Bublina s nápovědou v provedení obrázku s textem.

4.2.3 Nastavení stavu fragmentu

Každý `ZoomFragment` může nabývat stavů z `Enum TaskState`. Jsou to stavy `active`, `done`, `active_done` a `undone`. Tento stav se ukládá v `BaseActivity` a je nutné ho průběžně aktualizovat. Stav je využíván v mnoha částech aplikace. Nejviditelnější je zobrazení stavu fragmentu jako lístečku na rostlině v pravé části layoutu aplikace. To pomáhá uživateli v lepší orientaci o jeho postupu v části aplikace.

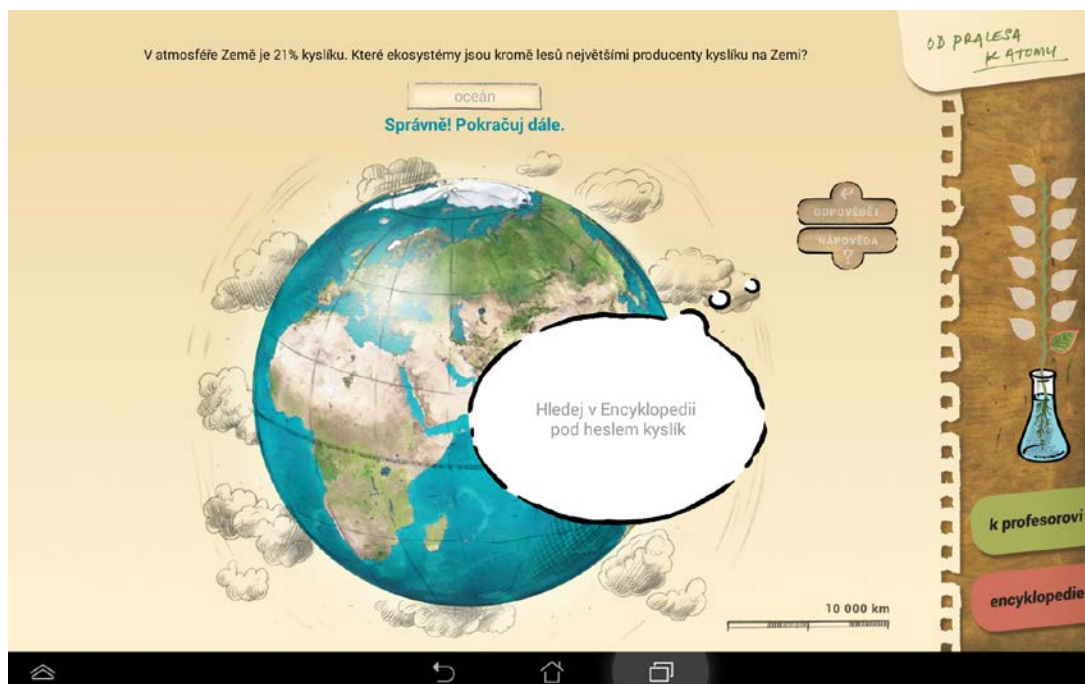
4.2.4 Zobrazení zprávy uživateli

Pro komunikaci s uživatelem slouží metoda `showMessage()`. Ta se využívá k zobrazení informací o stavu. Nejčastěji je to informace o tom jestli uživatel odpověděl správně nebo špatně. V této metodě je nastavena viditelnost příslušného `TextView` a spuštěna animace. Tato animace `TextView` nejprve postupně zobrazí a po 5 sekundách opět postupně skryje. Díky tomuto postupu je zpráva vždy zobrazena znovu a uživatel má tak přehled o tom zda kontrola jeho odpovědi proběhla.

4.3 OpenQuestionFragment

Jedním ze základních úkolů, které bude student v aplikaci plnit, je odpovídání na otevřenou otázku. Pro takový úkol je navržen `OpenQuestionFragment`.

Layout tohoto fragmentu obsahuje text otázky nebo i delšího sdělení, ze kterého otázka vychází. Pod ním následuje pole, kam má uživatel svou odpověď vyplnit. Odpověď je pro kontrolu nutno potvrdit tlačítkem odpovědět. Na pozadí celého fragment se nachází ilustrační obrázek. (Viz obr. 4.4)



Obr. 4.4: Grafické zpracování a rozmístění prvků pro OpenQuestionFragment.

Odpoověď uživatele je po potvrzení porovnána se zadanou sadou možných odpoovědí. Tato sada by měla být tvořena, jak s ohledem na možné odpoovědi na zadanou otázku, tak i s ohledem na možné překlepy či absenci diakritiky. Příkladem může být otázka „Jaký ekosystém produkuje po pralesích největší množství kyslíku?“. Odpoověď na tuto otázku může být „oceány“, „oceaný“, „oceán“ nebo „ocean“. Není-li řetězec zadaný uživatelem nalezen ve výčtu správných možností, je pole vymazáno a uživatel vyzván k novému pokusu. Je-li zadaná odpoověď správně, je odpoověď uložena a layout deaktivován.

4.3.1 TextView a EditText

Pro zobrazení textu v aplikacích pro Android slouží TextView [11]. Zobrazovanému textu lze nastavit základní vlastnosti jako písmo a velikost nebo pokročilejší dekorace s využitím Spanů. Obsahem TextView může být také html kód.

EditText je rozšířen TextView umožňující zadávání a editaci textu [11]. U EditTextu lze přímo specifikovat, jaký má být jeho obsah. Můžeme tedy nastavit, že má obsahovat pouze čísla. Toto nastavení také instruuje klávesnici, jaké rozložení kláves se zobrazí.

Text v poli lze také filtrovat jen na určitou délku nebo pouze velká písmena.

Možných nastavení je několik a lze vytvořit také vlastní. Typ vstupu do EditTextu lze omezit mnoha nastaveními typu InputType. Například omezit EditText jen na jeden řádek či skrývat znaky při zadávání hesla.

4.3.2 ImageView

ImageView je třída z Android API používaná k zobrazení obrázků v aplikaci [12] Stejně jako všem View, lze i ImageView nastavit background, tedy obrázek na pozadí. Ten automaticky vyplní celé View bez ohledu na velikost a poměr stran. ImageView však nabízí navíc možnost nastavit další Drawable jako obsah tohoto View.

Pro tento obraz lze již nastavit jednu z možností škálování

- `matrix` – škálování obrázku je provedeno nastavenou maticí.
- `fitXY` – velikost je nastavena tak aby obrázek pokryl celé View bez ohledu na poměr stran.
- `fitStart` – velikost obrázku je nastavena tak, aby se vešel celý do cílového ImageView. Škálování je provedeno s ohledem na poměr stran.
- `fitCenter` – obrázek je škálován stejně jako při použití `fitStart`. Výsledek je umístěn přímo do středu ImageView.
- `fitEnd` - obrázek je škálován stejně jako při použití `fitStart`. Výsledek je umístěn do pravého dolního rohu.
- `center` – obrázek je bez škálování umístěn do středu ImageView.
- `centerCrop` – obrázek je škálován tak, aby strany obrázku byly stejně velké nebo větší než strany ImageView.
- `centerInside` – obrázek je škálován tak, aby velikost jeho stran byla menší nebo stejná jako velikost stran ImageView.

Pro popis obsahu ImageView se používá `contentDescription`. Ten to řetězec slouží například nevidomým při použití služby TalkBack nebo podobných aplikací.

4.4 NumberQuestionFragment

NumberQuestionFragment je velice podobný předchozímu OpenQuestionFragment.

S tím rozdílem, že zde je otázka směřována na číselnou hodnotu.

Layout fragmentu je stejný. Jen textové pole je omezeno pouze na zadávání číselných hodnot. Jako argument je tomuto fragmentu zadán interval hodnot, které uznává jako správnou odpověď.

4.5 ChoiceQuestionFragment

ChoiceQuestionFragment je určený pro otázky s výběrem z několika možností. Správná může být jedna nebo i více možností. Aby uživatel rozpoznal, který typ otázky je aktuálně položen, liší se v použití RadioButtonů pro jednu správnou odpověď a CheckBoxů pro více možností.

Layout obsahuje kromě textu, obrázku na pozadí a tlačítek odpovědět a nápověda navíc LinearLayout, do kterého jsou programově přidány buďto CheckBoxy nebo skupina RadioButtonů (viz obr. 4.5). K těmto prvkům je přidán text odpovědí získaný z argumentů.



Obr. 4.5: Layout ChoiceQuestionFragment po splnění úkolu a deaktivaci prvků.

Kontrola správnosti odpovědí probíhá kontrolou zaškrtnutí polí na určených

indexech. Pokud zaškrtnutí odpovídá hodnotám získaných z argumentů je cvičení hodnoceno jako splněné.

U tohoto fragmentu je navíc možnost přidat i druhý obrázek pozadí, kterým je zaměněn původní obrázek po splnění cvičení. Tato možnost bude implementována do všech připravených fragmentů. Zda se má pozadí zaměnit či ne, rozhodne fragment podle toho, zda je druhé pozadí přidáno do argumentů.

4.5.1 RadioButton a CheckBox

RadioButtony a CheckBoxy jsou jedněmi ze základních formulářových prvků. Jejich účelem je označení či výběr nějaké možnosti. Rozdíl mezi nimi je v tom, že RadioButton může být v rámci layoutu nebo RadioGroup označen jen jeden. Zatímco CheckBoxů může být označeno více.

Grafické rozdělení je ustáleno tak, že RadioButtony jsou zobrazovány jako kruh, který při výběru obsahuje další tečku, naopak CheckBoxy jsou nejčastěji zobrazovány jako čtverec, který se označuje zaškrtnutím [14].

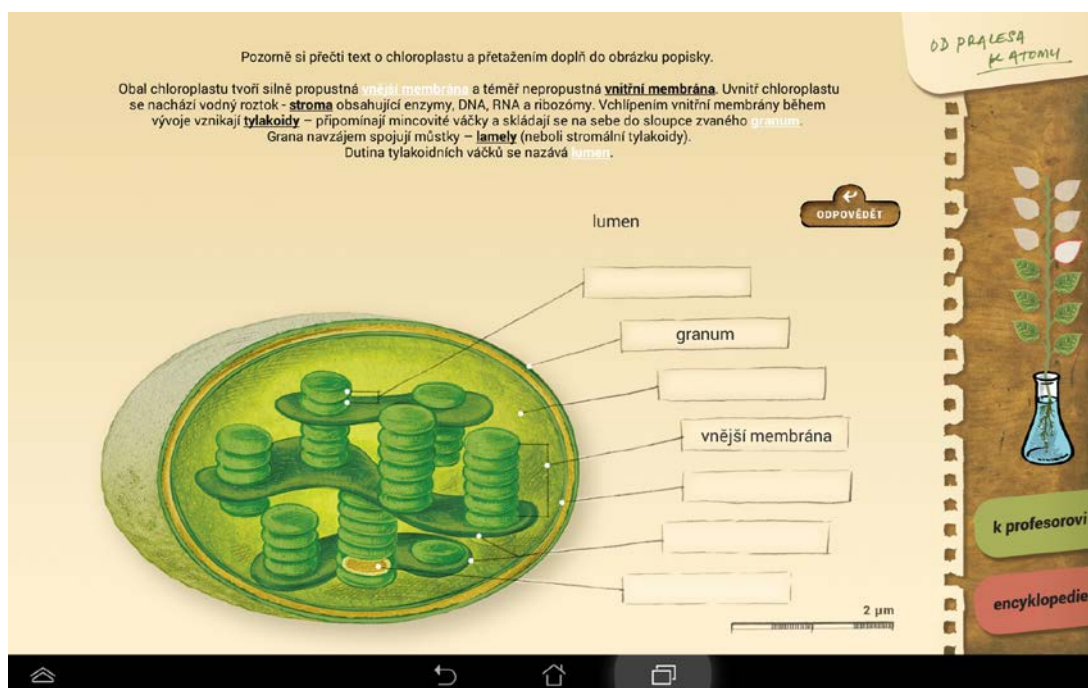
V Android API jsou obě třídy potomkem třídy CompoundButton [15]. Tato třída implementuje metody potřebné k zjištění stavu zaškrtnutí, takže je možné tento stav zjistit i bez znalosti toho, o kterou třídu se jedná. To v tomto fragmentu zjednodušuje implementaci kontroly správné odpovědi.

4.6 DragDropQuestionFragment

Pro přiřazování popisků k obrázkům či jako základ pro podobné úlohy s přetahováním textů je vytvořen DragDropQuestionFragment. Úkolem uživatele je přesunout hesla zvýrazněná v textu na správné místo v obrázku na pozadí. Nejčastěji k nějaké šipce nebo do připraveného rámečku (viz obr. 4.6).

Zvýraznění textu a možnost jeho přetahování je definována pomocí TouchableSpan. Span je prostředek Android API jak označit část textu a nastavit jí další vlastnosti. Jako například dekoraci nebo hypertextový odkaz. TouchableSpan dává textu vlastnost registrovat dotyky uživatele a vyvolat tak událost drag. Span je textu nastaven při parsování pomocí vodícího znaku @@. Podrobněji o tomto Spanu v další podkapitole.

Z přetahovaného textu je po puštění nad fragmentem vytvořeno TextView, se kterým je možné dále pohybovat a upravovat jeho umístění. Pro kontrolu správného umístění je použita černobílá mapa. Je vždy porovnána hodnota barvy pod středem popisku s hodnotou asociovanou s popiskem. Shodují-li se, je popisek umístěn na správném místě. Pokud se neshodují, je popisek zvýrazněn a uživatel je vyzván k jeho přemístění. Po správném umístění všech popisků je cvičení vyhodnoceno jako hotové a s popisy již nelze dále manipulovat.



Obr. 4.6: Layout DragDropQuestionFragment s několika umístěnými popisky.

4.6.1 TouchableSpan a LinkTouchMovementMethod

TouchableSpan je abstraktní třídou předepisující metodu onTouch pro zpracování dotyku. LinkTouchMovementMethod při kliknutí na TextView spočítá na jakou část textu a tedy i Span uživatel klikl. Pokud je příslušný Span typu TouchableSpan zavolá jeho onTouch() metodu.

Implementace metod TouchableSpan je provedena v jeho instanci ve fragmentu před nastavením Span částem textu. Po kliknutí na označené slovo je vyvolán sled akcí potřebných k vyvolání Drag a umožnění přemístění popisku do obrázku.

Díky tomu, že je třída TouchableSpan abstraktní, lze ji s rozdílnou implementací

použit i pro další úkoly využívající nějakého klikání do textu.

4.6.2 Drag and Drop

Android poskytuje vývojářům poměrně rozsáhlý Framework pro práce s drag and drop [10]. Tedy s přetahováním prvků a zpracováním přesouvaných dat či objektů. Framework obsahuje třídu definující nastávající události, posluchače těchto událostí a další pomocné třídy.

Zahájení přesouvání objektu probíhá po nějakém gestu uživatele. Nejčastěji při kliknutí a podržení nad objektem. Nejprve je nutné připravit data, která mají být přenášena. K tomu slouží třída `ClipData`. Také je třeba připravit stín objektu, který se uživateli při přetahování zobrazí pod prstem. Může to být přímo samotný objekt, jeho zmenšenina nebo nějaký úplně nový objekt. K vytvoření tohoto stínu slouží třída `DragShadowBuilder`. Ve své základní implementaci poskytnuté API je stín zobrazen přesně tak, jak vypadá `View`, které builder dostane. Má-li stín vypadat jinak, je nutné vytvořit potomka této třídy a příslušné metody vytvářející stín pozměnit.

Drag se odstartuje metodou třídy `View` `startDrag()`, která jako parametr dostane přesouvaná data a instanci `DragShadowBuilderu`, případně ještě lokální data a nastavení příznaků.

`View`, nad kterým mají být data odhozena, musí mít nastaven `OnDragListener`. V jeho jediné metodě `onDrag()` je podle typu události nutné rozhodnout co se má dělat. `DragEvent` generuje šest typů událostí.

- `ACTION_DRAG_STARTED` je vyvoláno je po té, co aplikace zavolá metodu `startDrag()` a získá stín.
- `ACTION_DRAG_ENTERED` je vyvoláno vstoupí-li stín do oblasti `View`. Chce-li `View` i dále získávat informace o událostech tohoto tažení je nutné, aby metoda `onDrag()` vracela hodnotu `true`.
- `ACTION_DRAG_LOCATION` poté co listener odpoví na událost `ACTION_DRAG_ENTERED` `true` začne získávat události typu `ACTION_DRAG_LOCATION` obsahující informace o umístění stínu nad `View`.
- `ACTION_DRAG_EXITED` značí, že stín opustil oblast `View`

- `ACTION_DROP` informuje o tom, že stín byl ukončen nad příslušným View. V tuto chvíli je potřeba zpracovat odhozená data. Pokud View zpracuje tuto událost, vrátí metoda `onDrag()` hodnotu `true`.
- `ACTION_DRAG_ENDED` je událost vyvolaná po zpracování `ACTION_DROP`. Pro zjištění, zda vše proběhlo v pořádku, slouží metoda `getResult()`, která vrátí stejnou hodnotu, jakou vrátila metoda `onDrag()` po zpracování události `ACTION_DROP`.

Pokud tedy nastane událost `ACTION_DROP` View zpracuje získaná data. V případě `DragDropQuestionFragmentu` to znamená, že na obrazovku umístí `TextView` s přesouvaným textem. Kontrola správného umístění probíhá až po potvrzení tlačítkem odpovědět.

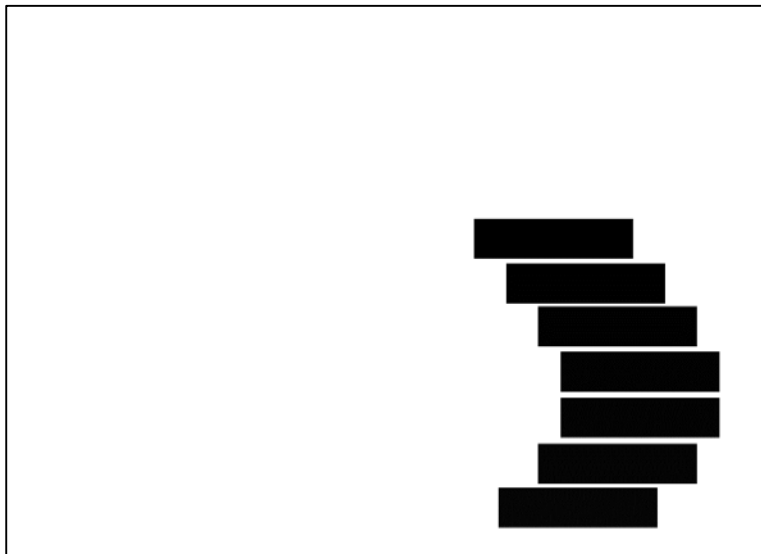
4.6.3 AspectRationImageView

Toto jednoduché, avšak velice důležité rozšíření základního `ImageView`, zajišťuje, aby byl obrázek vždy těsně obepnut hranicemi View. Je to velice důležité k zajištění toho, aby souřadnice dotyků nad `ImageView` a tedy i obrázkem bylo možné použít pro získání dat z jiného obrázku stejné velikosti.

Rozšíření funguje tak, že po vykreslení View na obrazovce je přepočítána jeho šířka tak, aby poměr stran View odpovídal poměru stran obrázku umístěného v něm.

4.6.4 Barevná mapa

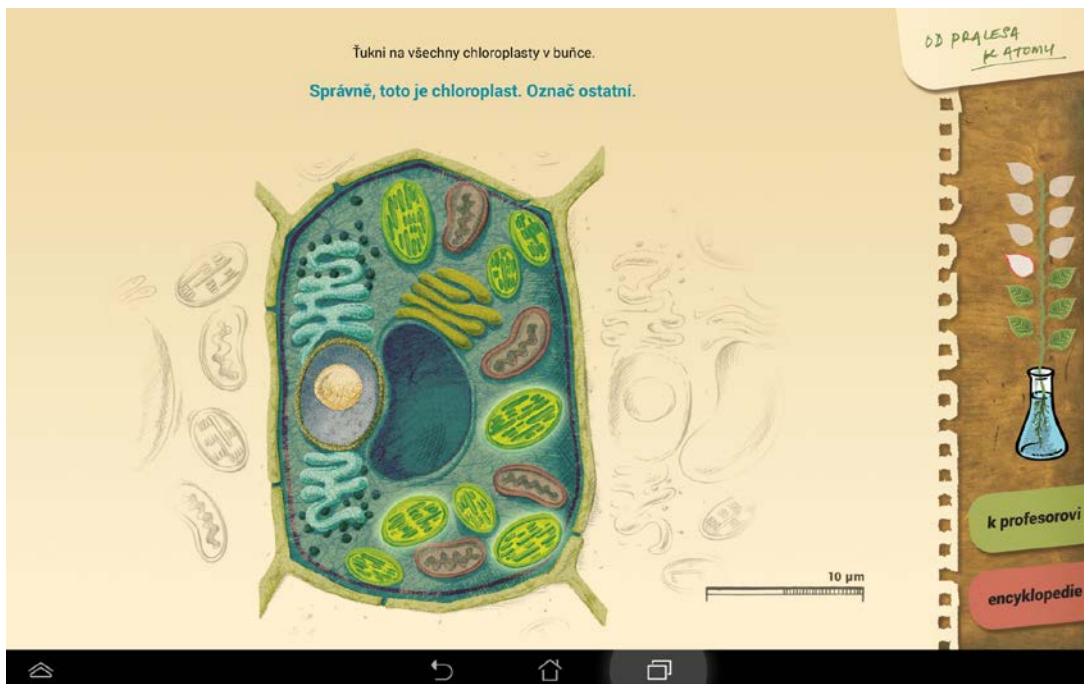
K rozhodnutí o správném umístění je využívána barevná mapa. Na bílém pozadí jsou příslušnou barvou, tam kde má být umístěn popisek, vyznačeny plochy (viz obr. 4.7). Každý popisek má svou hodnotu, která se musí shodovat s hodnotou červené barvy na pixelu pod ním. Implementace využívá jen červenou složku barvy, protože se nepředpokládá, že by uživatelé měli někdy umisťovat větší množství popisků. Rozšíření na kompletní kód barvy však není nijak problematické.



Obr. 4.7: Příklad barevné mapy pro DragDropQuestionFragment.

4.7 ClickQuestionFragment

Úkolem uživatele v případně ClickQuestionFragmentu je označit na obrázku zadané prvky (viz obr. 4.8). Pokud uživatel klikne na jiný prvek, může mu být zobrazen popis tohoto prvku. Tento fragment je také možné využít jen k zobrazování informací o označených částech obrázku. Při širším využití této možnosti by bylo dobré provést pár změn v označování prvků a odebrat kontrolu správnosti, aby se tak aplikace zrychlila.



Obr. 4.8: Obrázek na pozadí ClickQuestionFragment se zvýrazněnými prvky, na které bylo již správně kliknuto.

4.7.1 Detailní popis skládání obrázku a kontrola

Vyhodnocování kliknutí na obrázek probíhá pomocí mapy, podobné té u předchozího fragmentu. Prvky, na které je za úkol kliknout, mají hodnoty barvy od černé po počet prvků. Při 5 prvních je tedy barva v mapě posledního z nich RGB(4,4,4). Ostatní prvky obrázku mají svou barvu zadanou jako součást řetězce společně s názvem.

Po kliknutí na obrázek je tedy porovnána barva v mapě pod obrázek s barvou prvku, vybraného z listu všech prvků nacházejících se v obrázku. Pokud je to ten, který má být označen, je nastaveno, že byl nalezen a je zvýrazněn společně s dalšími již nalezenými prvky přidáním vrstvy do LayerDrawable na pozadí (viz obr. 4.5). Klikne-li uživatel na nějaký jiný prvek je zobrazeno hlášení, že to není ten správný prvek a je také zvýrazněn samostatně bez ostatních prvků.

Po kliknutí na poslední z prvků, které mají být označeny, je layout deaktivován a uživatel může pokračovat k dalšímu úkolu.

4.7.2 LayerDrawable

LayerDrawable definuje seznam jiných Drawable, které jsou následně vykresleny v pořadí, v jakém jsou zadány. Pozici obrázku lze upravit pomocí hodnot offsetů ze 4 stran [15].

LayerDrawable lze definovat jak v XML, tak přímo v kódu. Toho je využíváno při zvýrazňování objektů po kliknutí. Každý objekt na obrázku má připravenou svůj bitmapový obrázek ve stejném rozlišení, jaké má původní pozadí. Obrázky jsou následně v kódu skládány do vrstev nad sebe, takže se ve výsledku jeví jako zvýrazněné tak, jak je navrženo v grafice. Tento postup je jednodušší než přímo pozicovat menší obrázky na přesné místo, protože takto lze vše snadno připravit v grafickém editoru. Řešení je také nezávislé na velikosti zobrazení obrázku na tabletu.

4.8 StateClickQuestionFragment

Tento Fragment provede studenta třemi svými stavy, kde je jeho hlavním úkolem označit kliknutím objekt na obrázku (viz obr. 4.7). Například je tento fragment použit pro označení chybného popisku u obrázku. Fragment získá z argumentů tři obrázky na pozadí a tři texty, které postupně zobrazuje.



Obr. 4.9: Náhled úkolu, pro který je využít StateClickFragmentQuestion.

Úkolem uživatele je vždy kliknout. Nejprve kamkoliv na obrázek. Poté na konkrétní objekt, jehož správné označení je vyhodnoceno pomocí barevné mapy. V mapě jsou tentokrát jen dvě barvy. Bílá označuje plochu okolo objektu a černá označující oblast objektu kam by měl uživatel kliknout.

4.8.1 TransitionDrawable

TransitionDrawable slouží k vytvoření přechodu mezi dvěma obrázky. Důležitým natavením je metoda `setCrossFadeEnable()`. Pokud je cross-fade neaktivní první vrstva je vždy zobrazena neprůhledně. Aktivní cross-fade zajistí že, je první vrstva je zobrazena s opačnou hodnotou průhlednosti než má druhá vrstva [15].

Přechod je odstartován metodou `startTransition()` s délkou přechodu v milisekundách jako parametrem. Pomocí metody `reverseTransition()` lze spustit opačný proces přechodu.

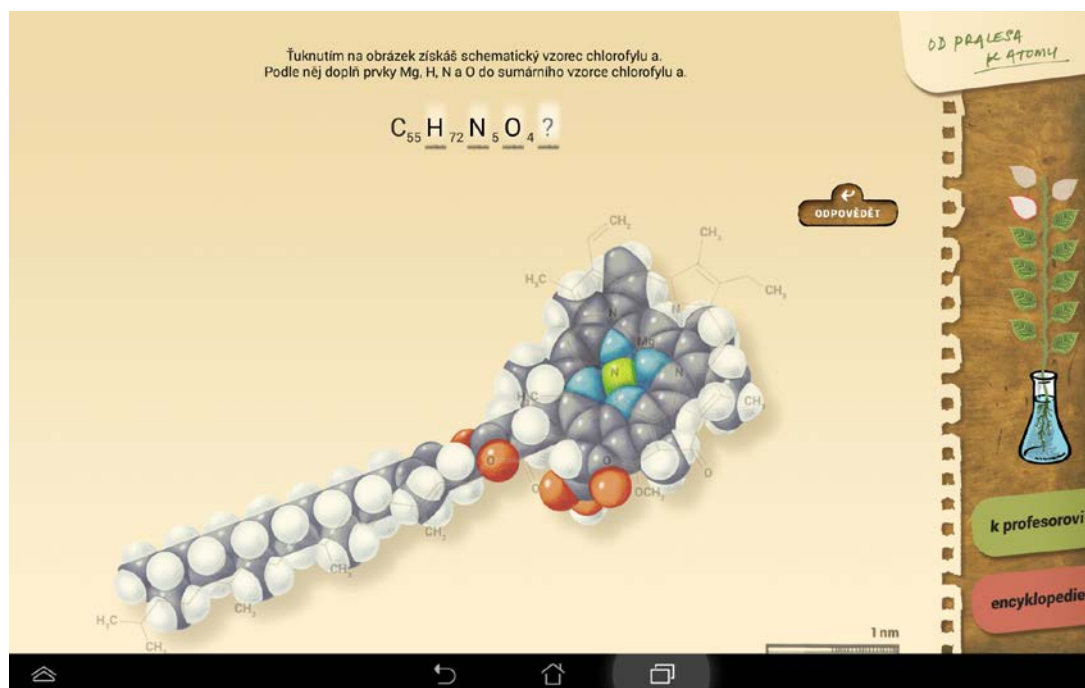
4.9 ChemicalFormulaQuestionFragment

ChemicalFormulaQuestionFragment představuje cvičení na doplnění značek prvků do chemického vzorce. Implementace nyní podporuje jednoduché chemické vzorce typu H_2O nebo H_2SO_4 .

Chemická rovnice je zadána fragmentu v argumentech jako pole řetězců. Na sudých indexech a 0 jsou značky chemických prvků. Je-li značka v závorce, jedná se o značku, která má být doplněna uživatelem. Na lichých polích jsou násobnosti jednotlivých prvků ve vzorci. Z tohoto pole vygenerován layout za použití TextView s různým stylem pro značky a indexy ve vzorci a EditTextů pro zadávání chybějících značek.

Po potvrzení odpovědi je ze značek zadaných uživatelem vytvořen řetězec, který je porovnám s řetězcem vytvořeným při skládání vzorce. Shodují-li se, je vše v pořádku a úkol je splněn.

Důležitým prvkem tohoto fragmentu je možnost změnit po kliknutí obrázek na pozadí z obrázku molekuly na schématický vzorec a zpět (viz obr. 4.10). Toho je docíleno pomocí TouchListeneru navěšeného na ImageView s pozadím a AlphaAnimation



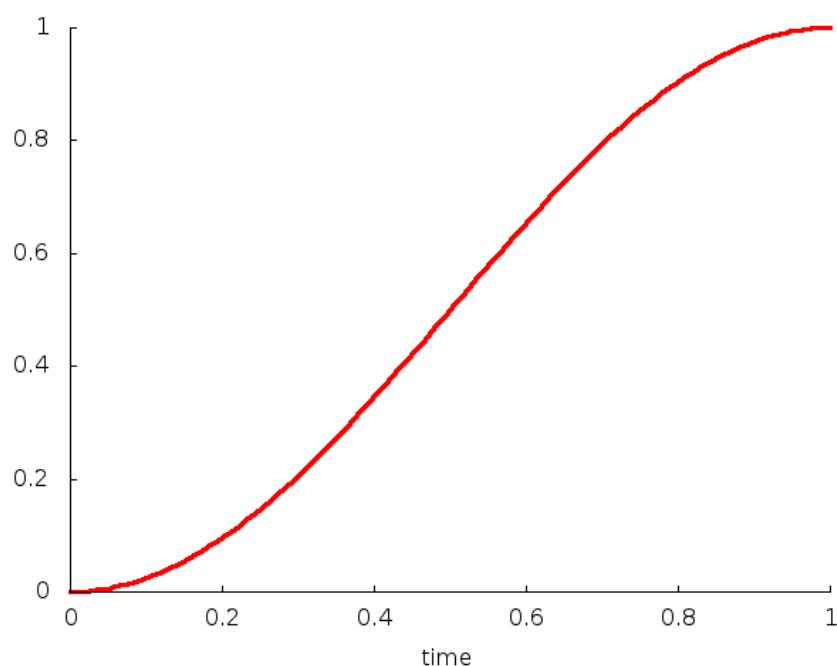
Obr. 4.10: Layout ChemicalQuestionFragment v průběhu animace změny pozadí.

4.9.1 Animace

Uživatel má možnost změnit kliknutím obrázků na pozadí. Efekt změny doprovází animace zvětšení a zmenšení obrázku a následný přechod pomocí `TransitionDrawable`. Animací je v nabídce Android API několik druhů a dají se definovat, jak přímo v kódu, tak pomocí XML souborů [16].

Tato kapitola se bude zabývat animacemi `View`. Animovat se dají také různé hodnoty pomocí `ValueAnimator` nebo `ObjectAnimator`. S `View` se dají animovat operace přesouvání, škálování, otáčení a průhlednost. Definice animací v XML obsahuje vždy počáteční a konečné hodnoty a krom dalších hodnot také čas, za který má animace proběhnout a čas, kdy má animace odstartovat. Všechny animace jsou prováděny současně a chceme-li aby tomu tak nebylo, je potřeba zadat čas, kdy má jaká animace proběhnout. Animace se dají také skládat do setů, které umožňují nastavit animacím některé společné hodnoty. Veškeré parametry animace se dají nastavit také v kódu. Nedoporučuje se to však, jelikož XML je přehlednější a variabilnější.

Animacím se dají nastavit také různé interpolátory, které definují průběh dopočítávání hodnot v animaci. Například hodnoty animace s nastaveným `AccelerateDecelerateInterpolator`, běží na začátku pomaleji a ke středu času zrychluje, od poloviny průběhu ke konci opět zpomaluje (viz obr. 4.11 [Zdroj]). Velice efektní je například `OverShootInterpolator`, který ke konci trochu přestřelí hodnoty a vrátí se zpět to nastavené konečné hodnoty.

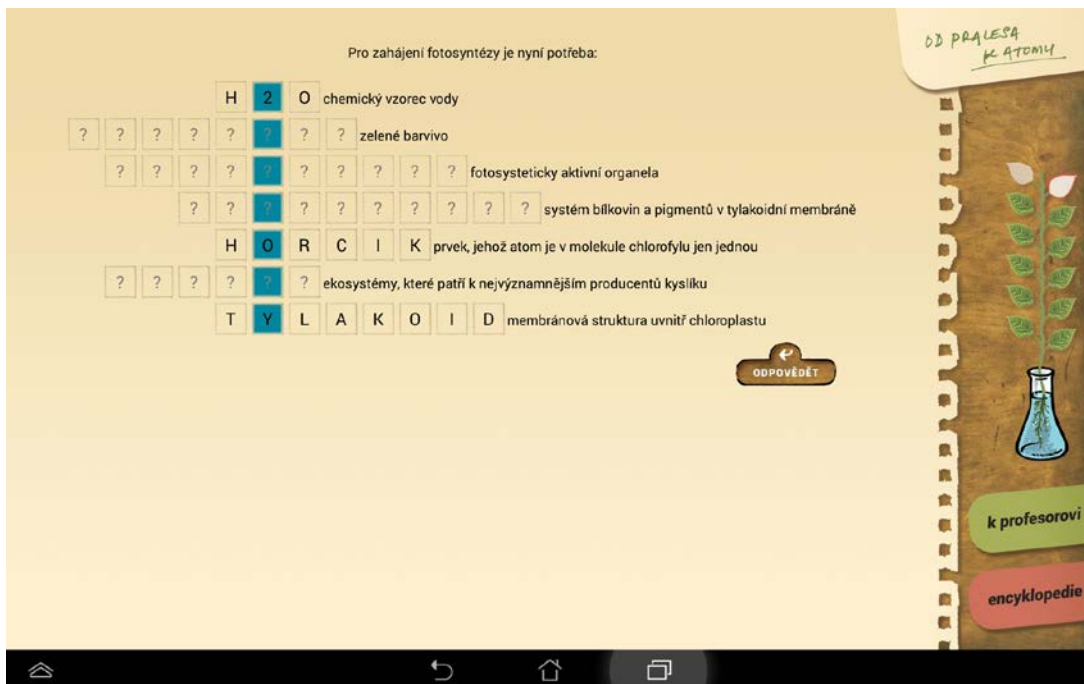


Obr. 4.11: Graf AccelerateDecelerateInterpolator [18]

Dalším důležitým prvkem animace je nastavení `fillAfter`. Pokud je `fillAfter` nastaveno na hodnoty `true`, zůstane View po animaci tak jak udávají konečné hodnoty animace. Naopak při nastavení na `false` se View vrátí do původního stavu. Existuje také nastavení `fillBefore`, které udává, zda se má View nastavit na počáteční hodnoty až v čase, kdy konkrétní animace započne, tedy až po uběhnutí času `startOffset`, nebo hned při odstartování celého bloku animací.

4.10 CrossWordsQuestionFragment

Zajímavým úkolem na implementaci je křížovka s tajenkou reprezentovaná `CrossWordsQuestionFragment` (viz obr. 4.12). Pro správnou funkčnost a vzhled křížovky bylo potřeba vyřešit hned několik věcí. Jako je samotné generování křížovky ze zdrojových textů a posouvání kurzoru mezi poli křížovky.



Obr. 4.12: Náhled layoutu CrossWordsQuestionFragment s několika vyplněnými slovy.

4.10.1 Generování křížovky

Fragmentu jsou při jeho vytváření argumentem předány dvě pole řetězců a jedno pole číselných hodnot. Tyto tři pole na stejných indexech obsahují informace k jednotlivým řádkům křížovky. Jeden řetězec představuje otázku, druhý odpověď, která má být zadána do křížovky, a číselná hodnota určuje, které písmeno ve slově doplňuje tajenku.

Po získání všech potřebných dat je nejprve nalezena velikost největšího ofsetu. Tedy největší počet písmen před tajenkou. Toto číslo je důležité pro správné zarovnání polí tajenky pod sebe.

Všechny pole křížovky jsou umístěny v `LinearLayout` s horizontální orientací, který představuje řádek křížovky a ten je umístěn v `LinearLayoutu` s vertikální orientací, který obaluje celou tabulku.

Pro každý řádek je tedy nyní vytvořen nový `LinearLayout`. Aby byla políčka tajenky správně zarovnána pod sebe, je nutné vypočítat levé odsazení řádku. K tomu slouží dříve získaná hodnota největšího ofsetu, od které je odečten ofset příslušného řádku a hodnota vynásobena šířkou jednoho pole křížovky.

Textovému poli je dále nastaveno několik parametrů omezujících možnosti zadávání. Například omezení na jeden řádek nebo filtr zapisující všechna písmena jako velká.

Další pole jsou již jen přidána přímo za první pole. Před přidáním je ještě nutné kontrolovat hodnotu ofsetu. Pokud se ofset shoduje s indexem pole, je mu nastaven jiný obrázek na pozadí zdůrazňující, že se jedná o písmeno z tajenky. Aby byl počet přidávaných polí správný, je nutné spočítat písmena ve slově s ohledem na české písmeno CH, které je zapsáno dvěma znaky.

4.10.2 Problém českého písmena CH a posouvání po polích

Problém s písmenem CH není jen připočítání polí pro zadání slova, ale zejména při zadávání znaků. Pokud by bylo každé písmeno tvořeno jen jedním znakem, stačí omezit počet znaků v poli na jeden a po naplnění posunout kurzor do dalšího pole. V případě, že v české abecedě máme písmeno CH, je nutné kontrolovat po každém novém písmenu, zda uživatel nechce napsat právě toto písmeno.

Pokud je pole prázdné a je do něj zadáno písmeno, je provedena kontrola, zda se nejedná o písmeno C, jakožto začátek CH. Pokud není zadáno písmeno C, kurzor se posune na další pole v pořadí. V případě, že je zadáno C, zůstává kurzor v poli.

Je-li další přidávané písmeno do pole H, tedy obsah pole se shoduje s písmenem CH, je kurzor posunut na další pole. Pokud je, ale písmeno jiné je toto písmeno vybráno a vloženo do následujícího pole tajenky. Pokud následující pole obsahuje nějaké písmeno je toto písmeno nahrazeno. Funguje to tak i v případě, že se uživatel vrátil a snaží se zadat další znak do již vyplněného pole.

Pokud se uživatel snaží přidat znaky k písmenu CH, toto písmeno se nijak nezmění. Je nutné ho nejprve smazat.

4.10.3 Problémy s backspace

Jedním z prostředků, jak zjednodušit zadávání slov do křížovky, je umožnit mazat text přes více EditTextů zároveň bez nutnosti umisťovat kliknutím kurzor do dalšího pole. Na zařízeních s Androidem k zadávání textu nejčastěji používá softwarová klávesnice, která

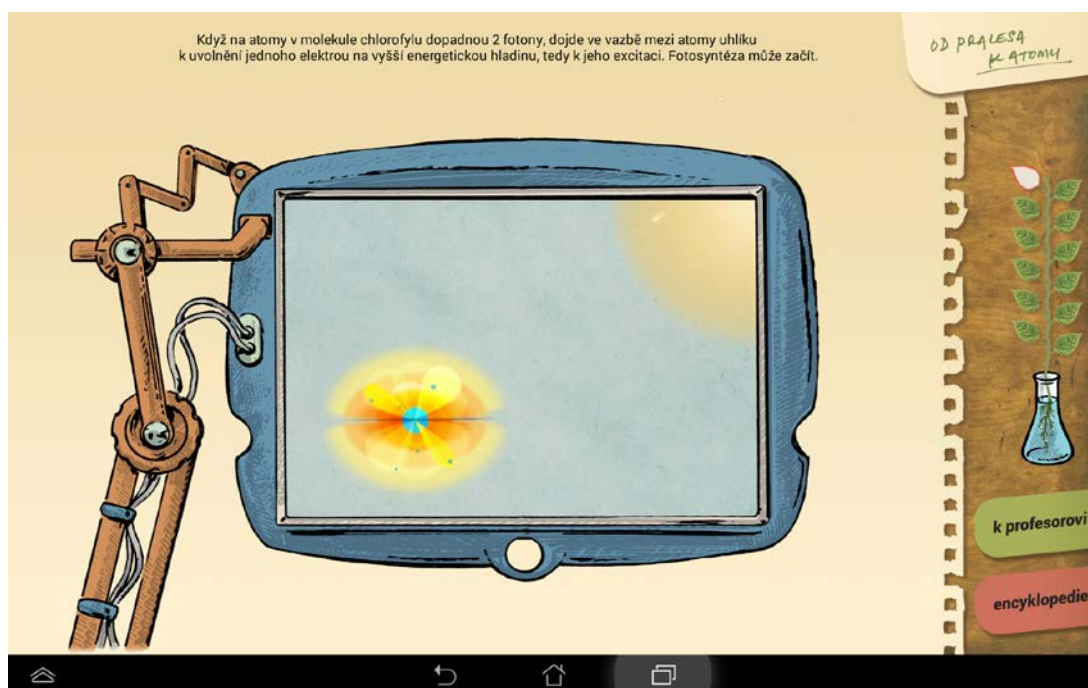
podle specifikace nemusí odesílat listenerům notifikace o stisknuté klávese [17].

Defaultní klávesnice tabletů od firmy Asus, na které je aplikace programována, tyto události však odesílá. Proto byl tento problém odhalen až později při vývoji aplikace. Zatím nebylo vypracováno jednoduché řešení jak tuto situaci řešit.

4.11 VideoFragment

VideoFragment slouží k přehrání doplňkového videa k probíranému tématu (viz obr. 4.13). Úkol je automaticky splněn po přehrání celého videa. Video je umístěno v k tomu určenému VideoView. Důležité je, aby video bylo v podporovaném formátu.

VideoView podporuje registraci různých listenerů stavu videa. V tomto fragmentu je použitý OnCompletionListener, který naslouchá ukončení videa.



Obr. 4.13: Náhled layoutu VideoFragmentu s rámečkem kolem videa.

5 TESTOVÁNÍ S UŽIVATELI

Testování s uživateli je důležitou součástí vývoje aplikace. Slouží k odhalení chyb v použitelnosti aplikace. Aplikace by se měla testovat v různých fázích vývoje, což pomůže odhalit chyby návrhu už v raném stádiu, kdy ještě nejsou velké změny tak složité a časově náročné. Vývoj aplikace by se měl řídit životním cyklem Návrh – Implementace – Testování. Kdy se vývoj po každém testování vrací k zapracování připomínek do návrhu a úpravě implementace. Po každém takové úpravě následuje opět testování, dokud aplikace není dostatečně použitelná.

Aplikace pro výuku fotosyntézy byla nejprve v rané fázi testována s jednotlivými uživateli v laboratorním prostředí. Uživatelé se mohli ptát moderátora na správnou odpověď a sám moderátor je mohl lehce navést správně položenými otázkami na jejich postup. Později se přistoupilo k testování přímo ve výuce, kdy studenti ve skupinách bez přílišného vedení lektora plní zadané úlohy. Takové testování probíhá v přirozeném prostředí.

Testování této aplikace je specifické v tom, že se netestuje jen použitelnost samotné aplikace a jejího uživatelského prostředí, ale také pedagogická stránka výuky při, které je aplikace využita.

5.1 Testování v laboratorních podmínkách

První testování prototypu aplikace probíhalo se vzorkem osmi participantů z různých ročníků střední školy. Participantů nebyli vybíráni pomocí screeneru, ale byli přiděleni přímo školou.

5.1.1 Pre-test dotazník

Před testováním participantů zodpověděli několik pre-testových otázek. Výpis jejich odpovědí lze nalézt v tabulce 5.1.

Výsledky pre-test dotazníku								
Otázka	P1	P2	P3	P4	P5	P6	P7	P8
Věk	16	19	15	15	18	17	16	17
Pohlaví	muž	muž	žena	muž	muž	žena	muž	muž
Ročník	1.	3.	1.	1.	3.	3.	2.	2.
Má tablet nebo chytrý telefon	ano	ano	ano	ano	ano	ano	ano	ano
OS zařízení	iOS	android	android	android	android	iOS	jiný	symbian

Tab. 5.1: Odpovědi na otázky v pre-test dotazníku testování

Podle výsledků můžeme vidět, že všichni participanti mají zkušenost s chytrými telefony nebo tablety. V cílové skupině studentů se však může objevit student, který takové zařízení nevlastní nebo s ním má minimální zkušenosti a bylo by dobré aplikaci testovat i s takovými participanty.

5.1.2 Průběh testování

Participanti jednotlivě přicházeli k testovacímu tabletu, byli uvítáni moderátorem a byl jim vysvětlen další postup. Uživatelé byli instruováni, aby při práci s aplikací přemýšleli nahlas a pokud by nevěděli správnou odpověď na některou z otázek, moderátor byl připraven jim pomoci. Participantům bylo vysvětleno, že neprobíhá testování jich samotných a jejich znalostí, ale aplikace, jejího návrhu a implementace.

Následně bylo uživateli sděleno, kde má aplikaci spustit a v jakém pořadí bude spouštět testované sekce.

5.1.3 Výsledky testování

Testování probíhalo pro celou aplikaci, tedy i ostatní části, které jsou prací jiných programátorů. Následující nálezy se vztahují jen k části zoomy nebo obecně k celé aplikaci. Nálezy jsou rozděleny podle pořadí zoomu, na kterém k nálezu došlo a zobrazeny v tabulce 5.2. Ke každému nálezu je doplněna priorita napravení problému.

Priorita má tři stupně, nízká, střední a vysoká. Kromě priority je připojeno také možné řešení problému. Některé zjištěné problémy vedly k tomu, že fragmenty nebyly ve výsledné aplikaci vůbec použity a byly nahrazeny existujícími nebo úplně novými fragmenty. Spousta problémů nalezených při testování byla vyřešena nasazením finální grafiky, při jejímž vytváření byl brán ohled na výsledky testování.

Výsledky testování prototypu aplikace – část Zoomy		
Nález	Priorita	Návrh řešení
<i>Uživatelům není jasné používání gesta pinch</i>	<i>Vysoká</i>	<i>Přidat ukázkovou animaci</i>
1 – OpenQuestionFragment		
Nález	Priorita	Návrh řešení
<i>Uživatele mate zadání otázky</i>	<i>Střední</i>	<i>Upravit zadání otázky</i>
2 – NumberQuestionFragment		
Nález	Priorita	Návrh řešení
<i>Uživatel si nevšiml, že odpověděl správně</i>	<i>Střední</i>	<i>Zvýraznit hlášku o splnění</i>
3 - ChoiceQuestionFragment - jedna možnost		
<i>Bez nálezu</i>		
4 - CutQuestionFragment		
Nález	Priorita	Návrh řešení
<i>Řezání nereaguje nebo reaguje jen z jedné strany</i>	<i>Vysoká</i>	<i>Upravit logiku</i>
5 - ChoiceQuestionFragment - dvě možnosti		
<i>Bez nálezu</i>		
6 - ClickQuestionFragment		
Nález	Priorita	Návrh řešení
<i>Aplikace padá</i>	<i>Vysoká</i>	<i>Upravit logiku</i>
<i>Uživatel očekává reakci na chybné kliknutí</i>	<i>Střední</i>	<i>Přidat barevné rozlišení</i>

7 - DragDropQuestionFragment		
Nález	Priorita	Návrh řešení
<i>Aplikace padá</i>	<i>Vysoká</i>	<i>Upravit logiku</i>
<i>Malá tolerance umístění</i>	<i>Střední</i>	<i>Zvýšit toleranci</i>
8 - StateClickQuestionFragment		
Nález	Priorita	Návrh řešení
<i>Aplikace padá</i>	<i>Vysoká</i>	<i>Upravit logiku</i>
<i>Uživatelé neví, kam mají kliknout</i>	<i>Střední</i>	<i>Upravit zadání úkolu</i>
9 - ChemicalFormulaQuestionFragment		
Nález	Priorita	Návrh řešení
<i>Uživatelé mají problémy s klávesnicí</i>	<i>Nízká</i>	
10 - AtomQuestionFragment		
Nález	Priorita	Návrh řešení
<i>Zobrazuje se chybný toast</i>	<i>Nízká</i>	<i>Upravit zobrazování toastů</i>
<i>Animace fotonů uživatele mate</i>	<i>Nízká</i>	<i>Zlepšit animaci</i>

Tab. 5.2: Výsledky testování z části Zoomy s určenou prioritou a návrhem řešení

5.1.4 Post-test dotazník

Po skončení testu bylo participantovi položeno několik doplňujících otázek ohledně proběhnutého testování. Několik otázek se týkalo přímo jednotlivých částí. Další pak celé aplikace a jejího využití ve škole. Tabulka 5.3 představuje soupis odpovědí na otázky ohledně části zoomy. Odpovědi na otázky k celé aplikaci se nacházejí v příloze A.

Výsledky post-test dotazníku k části Zoomy								
Otázka	P1	P2	P3	P4	P5	P6	P7	P8
Při odpovídání na jednotlivé úkoly jsem si přešel/přišla efektivní?	<i>nevím</i>	<i>ano</i>	<i>ne</i>	<i>ano</i>	<i>ano</i>	<i>nevím</i>	<i>spíše ne</i>	<i>ano</i>
Úkoly mi přišly srozumitelné?	<i>ano</i>	<i>spíše ano</i>	<i>ano</i>	<i>ano</i>	<i>spíše ano</i>	<i>ano</i>	<i>ano</i>	<i>ano</i>
V této kapitole jsem se dozvěděl něco nového.	<i>spíše ne</i>	<i>ano</i>	<i>spíše ano</i>	<i>spíše ano</i>	<i>ne</i>	<i>ano</i>	<i>spíše ano</i>	<i>ano</i>
Co vám přišlo zajímavé?	<i>8. okno - obrázek</i>	<i>strom užívá 3 lidí, rozlišení chloroplast/ chlorofyl</i>		<i>9. okno</i>		<i>obrázky</i>		<i>pinch, interaktivita, rozříznutí listu</i>
Co se vám nelíbilo?	<i>4. okno - nefungovala interakce</i>	<i>přeříznutí listu</i>		<i>5. okno, složité</i>	<i>přeříznutí listu</i>	<i>7. okno</i>		<i>nic</i>
Problémy s řešením?	<i>4. okno, 7. okno - přiřazování</i>	<i>přiřazení popisků ke schématu, přeříznutí listu</i>		<i>5. okno, sinice</i>	<i>přeříznutí listu</i>	<i>7. okno</i>		<i>rovnice uhlovodíku (uznáno nesprávně)</i>

Tab. 5.3: Výsledky post-test dotazníku k části Zoomy

5.1.5 Video z testování

Při testování byly využity kamery GoPro k záznamu práce uživatele. Na záběrech však není mnoho vidět a zvuk je špatně slyšet. Záběry však posloužily k měření času, který uživatel strávil nad jednotlivými úkoly.

5.2 Testování ve výuce

Ve druhé fázi bylo testování prováděno přímo v učebně s celou třídou studentů. Testování

tedy probíhalo jako zkouška celého konceptu výuky s využitím tabletů.

Studentů byly tablety rozdány do dvoučlenných skupin. Byli upozorněni, že aplikace je stále ve vývoji a lze očekávat komplikace, na které se tímto testováním mají odhalit. Moderátoři a lektoři se pohybovali mezi skupinkami a sledovali, jak probíhá plnění jednotlivých úkolů. Testování odhalilo několik problémů. Například padání aplikace při zoomování, kvůli tomu, že oba studenti provedli gesto najednou. Bylo nalezeno i několik problémů v grafickém návrhu. Například špatný design check boxů. Více nálezů z prvního běhu testování se nachází v tabulce 5.3.

Po skončení testování byli studenti ještě dotázáni na několik doplňujících otázek, jejichž podnětné odpovědi byly zaznamenány do poznámek.

Toto testování proběhlo zatím ve dvou bězích. Při druhém testování nastalo několik komplikací nejprve s aktualizací na novou verzi aplikaci a poté byla nalezena v netestované narychlo vytvořené verzi závažná chyba, která znemožnila postup do další části aplikace. Otestována byla tedy jen část aplikace. Nálezy z tohoto testování byly pro zpřehlednění zapsány do tabulky 5.3.

Výsledky testování aplikace ve výuce – část Zoomy		
Nález	Priorita	Návrh řešení
<i>Aplikace při zoomování, při zoomování pokud ještě předchozí efekt nedoběhl a při dotecích více rukou najednou</i>	<i>Vysoká</i>	<i>Přidat výjimky</i>
<i>Na úplně prvním okně si nevšímají tlačítka OK, snaží se rovnou zoomovat</i>	<i>Střední</i>	<i>Dokud neodkliknou OK, nemělo by jít zoomovat. Tlačítko OK zvětšit a dát blíž k prstům.</i>
<i>Chybí blokování editace, pokud je okno již dokončené</i>	<i>Vysoká</i>	
<i>Chybí zakáz procházení bez hotového úkolu</i>	<i>Vysoká</i>	
<i>Klávesnice zakrývá nápovědu</i>	<i>Střední</i>	<i>Při zobrazení nápovědy skryt klávesnici</i>
<i>Participant si myslí si, že check box je radio button</i>	<i>Vysoká</i>	<i>Zaškrtnutí namísto tečky</i>

Tab. 5.4: Výsledky testování z části Zoomy s určenou prioritou a návrhem řešení.

5.3 Testování aplikace Vinařická horka v terénu

Součástí projektu aplikací pro podporu výuky je také aplikace Vinařická horka, která využívá principu geolokační hry v přírodně zajímavé oblasti poblíž Kladna. Testování takové aplikace je kvůli geolokaci nutné provádět přímo v terénu.

Velice důležitá je u této aplikace viditelnost na displeji. Lesklé displeje tabletu na slunci ztrácí čitelnost a malé rozdíly například v barvě grafiky jsou velice špatně viditelné. Takové problémy by při testování v laboratorních podmínkách byli velice těžko odhalitelné.

Dalším důležitým objektem testování je správná kalibrace GPS souřadnic. Při prvních testech vývojářského týmu se stávalo, že se některé lokace pro plnění úkolů nacházeli na nedostupných místech. Takové úkoly jsou pak nesplnitelné a výuku nelze dokončit. S GPS souvisí i zobrazení mapy, podle které se studenti po oblasti pohybují. Toto zobrazení musí být dostatečně přehledné, aby ve členitém terénu zbytečně nebloudili. Jedním z nálezů ohledně mapy bylo zjištění, že se na ní nenacházejí žádné jiné cesty než trasa, po které mají studenti přímo jít, což stěžovalo orientaci na některých

rozcestích či u nenápadných odboček.

Testování probíhalo s celou třídou studentů rozdělených do skupinek po 3-4 lidech. Moderátoři se pohybovali mezi skupinkami a dělali si poznámky, tak se postupně objevovaly problémy s aplikací. Součástí tohoto testování nebyly žádné papírové dotazníky pro jednotlivé participanty. Post a pre-test dotazování bylo prováděno s celou třídou najednou a pro participanty lehce zábavnou formou. Například byla položena otázka, jak se participantům líbilo grafické zpracování aplikace, a byli vyzváni, aby se podle své odpovědi rozmístili na ose od „líbilo“ po „nelíbilo“. Takové výsledky se poměrně špatně zaznamenávají, avšak přítomným dávají poměrně jasnou představu o názorech.

Studenti byli také hromadně dotazováni, se kterými úkoly měli problém, nebo které úkoly jim přišly zajímavé. Všechny nálezy byly zaznamenány do poznámek a využity při dalších úpravách návrhu či implementace. Tyto poznámky obsahuje příloha A.2

5.4 Závěry z testování

Při testování bylo nalezeno mnoho menších či větších chyb jak v návrhu, tak v implementaci aplikace. Tyto chyby a nedostatky byly opraveny a použitelnost aplikace tak byla zvýšena.

5.5 Závěr

Cílem této práce bylo prostudovat možnosti vývoje aplikací pro operační systém Android a navrhnout a implementovat část aplikace pro podporu výuky biologie. Součástí přípravy aplikace bylo také hodnocení existujících aplikací pro výuku na trhu. Výsledná aplikace byla testována s uživateli a následně upravena podle nálezů při testování.

Důležitou částí práce bylo studium samotného programování pro platformu Androidu, se kterou má autor minimální zkušenosti. Vyvinutá aplikace tedy slouží jako základní seznámení s platformou pro další aplikace vyvíjené v rámci projektu tabletových aplikací pro podporu výuky technických předmětů.

Podářilo se připravit pro různé typy úkolů znovupoužitelné implementace fragmentů, které lze snadno využít v jiných částech aplikace tak i aplikacích nových, které budou mít stejnou či velmi podobnou architekturu.

Aplikace byla otestována přímo ve výuce, tak jak má být v budoucnu využívána, a až na několik odstranitelných chyb funguje dobře a ve výuce studenty práce s tabletem baví a učí zároveň.

Vývoj aplikace bude dále pokračovat zejména ve směru optimalizace výkonu a dokončení dalších částí, které jsou stále ve vývoji. Fragmenty vytvořené pro tuto část budou buď přímo, nebo po úpravách, využity pro další podobné aplikace.

LITERATURA

- [1] Android tops 81 percent of smartphone market share in Q3. engadget. [online]. 31.10.2013 [cit. 2014-05-17]. Dostupné z: <http://www.engadget.com/2013/10/31/strategy-analytics-q3-2013-phone-share/>
- [2] Android (operační systém) – Wikipedie. Wikipedie. [online]. 13.4.2014 [cit. 2014-05-17]. Dostupné z: [http://cs.wikipedia.org/wiki/Android_\(opera%C4%8Dn%C3%AD_syst%C3%A9m\)](http://cs.wikipedia.org/wiki/Android_(opera%C4%8Dn%C3%AD_syst%C3%A9m))
- [3] Activities | Android developers. Android developers. [online]. [cit. 2014-05-17]. Dostupné z: <http://developer.android.com/guide/components/activities.html>
- [4] Vytvíjíme pro Android: Suroviny, intenty a jednotky | Zdroják. Zdroják. [online]. 29.6.2012 [cit. 2014-05-17]. Dostupné z: <http://www.zdrojak.cz/clanky/vyvijime-pro-android-suroviny-intenty-a-jednotky/>
- [5] Fragments | Android developers. Android developers. [online]. [cit. 2014-05-17]. Dostupné z: <http://developer.android.com/guide/components/fragments.html>
- [6] Duolingo: Learn Languages Free – Aplikace pro Android ve službě Google Play. Google Play. [online]. 21.4.2014 [cit. 2014-05-12]. Dostupné z: <https://play.google.com/store/apps/details?id=com.duolingo>
- [7] Nauč se pravopis – Aplikace pro Android ve službě Google Play. Google Play. [online]. 23.12.2013 [cit. 2014-05-12]. Dostupné z: <https://play.google.com/store/apps/details?id=nauc.se.cesky&hl=cs>
- [8] Anatomy 3D – Anatronica – Aplikace pro Android ve službě Google Play. Google Play. [online]. 2.10.2013 [cit. 2014-05-12]. Dostupné z: <https://play.google.com/store/apps/details?id=com.GoodwillEnterpriseDevelopment.Anatronica>
- [9] Fundamentals | Android developers. Android developers. [online]. [cit. 2014-05-17]. Dostupné z: <http://developer.android.com/guide/components/fundamentals.html>
- [10] Draging and Scaling | Android developers. Android developers. [online]. [cit. 2014-05-17]. Dostupné z: <http://developer.android.com/training/gestures/scale.html>
- [11] Text Fields | Android developers. Android developers. [online]. [cit. 2014-05-17].

- Dostupné z: <http://developer.android.com/guide/topics/ui/controls/text.html>
- [12] Image View | Android developers. Android developers. [online]. 13.5.2014 [cit. 2014-05-17]. Dostupné z: <http://developer.android.com/reference/android/widget/ImageView.html>
- [13] Switches | Android developers. Android developers. [online]. [cit. 2014-05-17]. Dostupné z: <http://developer.android.com/design/building-blocks/switches.html>
- [14] Compound Button | Android developers. Android developers. [online]. 13.5.2014 [cit. 2014-05-17]. Dostupné z: <http://developer.android.com/reference/android/widget/CompoundButton.html>
- [15] Drawable Resources | Android developers. Android developers. [online]. [cit. 2014-05-17]. Dostupné z: <http://developer.android.com/guide/topics/resources/drawable-resource.html>
- [16] Animation | Android developers. Android developers. [online]. [cit. 2014-05-17]. Dostupné z: <http://developer.android.com/guide/topics/resources/animation-resource.html>
- [17] Handling Keyboard Actions | Android developers. Android developers. [online]. [cit. 2014-05-17]. Dostupné z: <http://developer.android.com/training/keyboard-input/commands.html>
- [18] Android Animations Tutorial 5: More on Interpolators | Cogito Learning. Cogito Learning. [online]. 24.10.2013 [cit. 2014-05-17]. Dostupné z: <http://cogitolearning.co.uk/?p=1078>

SEZNAM OBRÁZKŮ

Obr. 1.1: Vrstvy operačního systému Android.....	8
Obr 1.2: Životní cyklus Activity.....	11
Obr. 1.3: Příklad použití fragmentu na různých velikostech displeje.....	12
Obr. 2.1: Hlavní menu aplikace Duolingo. V levé části vidíme počítadlo zkušeností a tabulku s pořadím mezi přáteli. V pravé pak výběr mezi tematickými okruhy.	15
Obr. 2.2: Pro výuku aplikace využívá různé druhy cvičení. Na obrázku je zobrazeno překládání slovních spojení. Vpravo nahoře se nachází počítadlo životů, které jsou odečítány za špatné odpovědi.....	16
Obr. 2.3: V doplňovacích cvičeních uživatel vybírá, jaké písmeno se má doplnit. Aplikace zadání ihned kontroluje a zobrazí také zdůvodnění správné odpovědi.....	17
Obr. 2.4: 3D vizualizaci lidského těla lze zobrazovat po jednotlivých částech. Lze také nastavit části těla poloprůhledné a vidět tak, co se nachází pod nimi.....	18
Obr. 4.1: Diagram tříd BaseActivity a ZoomActivity s výpisem metod	21
Obr. 4.2: Diagram tříd týkajících se Fragmentů.	24
Obr. 4.3: Bublina s nápovědou v provedení obrázku s textem.	25
Obr. 4.4: Grafické zpracování a rozmístění prvků pro OpenQuestionFragment.....	27
Obr. 4.5: Layout ChoiceQuestionFragment po splnění úkolu a deaktivaci prvků.	29
Obr. 4.6: Layout DragDropQuestionFragment s několika umístěnými popisky.....	31
Obr. 4.7: Příklad barevné mapy pro DragDropQuestionFragment.....	34
Obr. 4.8: Obrázek na pozadí ClickQuestionFragment se zvýrazněnými prvky, na které bylo již správně kliknuto.....	35
Obr. 4.9: Náhled úkolu, pro který je využit StateClickFragmentQuestion.....	37
Obr. 4.10: Layout ChemicalQuestionFragment v průběhu animace změny pozadí	38

Obr. 4.11: Graf AccelerateDecelerateInterpolator [18]	40
Obr. 4.12: Náhled layoutu CrossWordsQuestionFragment s několika vyplněnými slovy.	41
Obr. 4.13: Náhled layoutu VideoFragmentu s rámečkem kolem videa.	43

SEZNAM TABULEK

Tab. 5.1: Odpovědi na otázky v pre-test dotazníku testování.....	45
Tab. 5.2: Výsledky testování z části Zoomy s určenou prioritou a návrhem řešení	47
Tab. 5.3: Výsledky post-test dotazníku k části Zoomy.....	48
Tab. 5.4: Výsledky testování z části Zoomy s určenou prioritou a návrhem řešení.	50

SEZNAM ZKRATEK A POJMŮ

XML - Extensible Markup Language

API - Application Programming Interface

IDE Integrated Development Environment

OS - Operační Systém

UI - User Interface

Zoomy – jednotlivé obrazovky části „Od pralesa k atomu“

Zoomování – pohyb mezi úkoly pomocí gesta pinch

PŘÍLOHY

A TESTOVÁNÍ

A.1 Výsledky post-test dotazníku k celé aplikaci

Otázky

1. Chtěl bych podobné učebnice v jiných hodinách
2. Dovedu si představit písemku prostřednictvím této aplikace
3. Co si myslím o používání tabletů ve výuce
4. Způsob přihlašování do aplikace
5. Jak zamezit zneužití osobních údajů
6. Jak zamezit používání jiných aplikací
7. Jakým způsobem odnést vyplněné úkoly
8. Ocenil bys používání vlastního tabletu?
9. Jakou jinou elektronickou výukovou aplikaci si někdy použil?
10. nejdůležitější vlastnosti výukové aplikace?
11. Z jakého důvodu sis vybral OS na svém zařízení?
12. Líbí se ti připravovaná grafická podoba?
13. Poznámky

Odpovědi

Participant 1

1. spíše ne
2. spíše ne
3. bylo by to dobré, ale s mírou
4. uživatelské jméno, heslo
5. uživatelské jméno, heslo
6. dělo by se to často - blokovat přístup!
7. na tabletu
8. preferoval by školní
9. něco na anglický jazyk na webu

10. přehlednost; co nejmenší změny od sešitu
11. bez důvodu
12. ano
13. -

Participant 2

1. nevím
2. spíše ne
3. posuzovat předmět od předmětu, technické problémy učitelů, děti s počítačem, dobrá názornost, bojí se ztráty času kvůli použití tabletů, aby to nebylo zbytečně vynucené
4. školní login, příjmení
5. není co zneužít pokud nikdo nesdělí své heslo
6. neví jak technicky, ale ošetřit proti použití taháků
7. stáhnout ze školního serveru, nebrání se odeslání na mail
8. nemá, problém s vlastními aplikacemi -> každý by si dělal, co chce
9. autoškola (otázka na mobilu) - 3 sekce - učení, procvičení, test
10. obrázky, názornost, bojí se malé informační hodnoty
11. nechce apple
12. spíše ano
13. malý kontrast rámečků u sinic; neví, že je encyklopedie tlačítko

Participant 3

1. ano
2. spíše ano
3. ANO; zvyšuje pestrost předmětu; zamezit používání jiných aplikací
4. otisk prstů
5. zamknout při odložení
6. prostě vypnout
7. na flashce (email je náročný)
8. bylo by to dobré, ale odvádělo by to pozornost. Takže ne!
9. na angličtinu (mluvené slovo atd.)
10. dobrá grafika, zajímavé na pohled
11. bez důvodu, líbil se mi telefon
12. ano
13. -

Participant 4

1. nevím
2. nevím
3. zábavné, pohodlnější, netřeba učebnic a sešitů, více to rozvíjí mozek
4. heslo, gesto odemykání klávesnice
5. posláni nového heslo na mail, zpráva na mobil pokud se někdo připojí k účtu
6. zablokování internetu, zablokování aplikací, zabránit stažení aplikací
7. dropbox, apod.
8. ano
9. angličtina
10. nevím
11. je lepší, iOS je složitější
12. ano
13. pěkné, "školní", našel encyklopedii, neví co je "do laboratoře", zvýraznit políčko pro odpověď

Participant 5

1. ano
2. ano
3. dobrý nápad; zneužitelné k nevěnování se výuce
4. čip; školní login
5. čipem; nesděloval bych své heslo
6. nedal bych jim tablety; omezení funkcí
7. odeslat na mail/FB
8. nechtěl bych ho s sebou nosit
9. autoškoly; cady
10. přehledná; kvalitní obsah; více grafů/obrázků než textu
11. nemám rád Apple
12. ano
13. vizuálně pěkné

Participant 6

1. ano
2. ano
3. dobrý nápad, oživení, zábavnější, ne v matice, ano v chemii a fyzice
4. jméno a heslo

5. složité heslo, nepůjčovat
6. učitelský zámek (jako rodičovský), blokovat obchod a internet
7. vytisknout, do prezentace na mail
8. ocenila by vlastní tablet (chce iPod) nebo telefony
9. žádné
10. vzhled, jednoduché ovládání, rychlé
11. používá iOS, rychlé a přehledné
12. ano
13. zajímavé, pěkné, lupínky nemají čísla, chybí tlačítko zkontrolovat

Participant 7

1. spíše ano (jako oživení výuky)
2. spíše ne
3. výuka by mohla být zajímavější
4. to je jedno
5. nepřihlašovat přes FB nebo email
6. nedát možnost si jiné aplikace nainstalovat
7. vytisknout?
8. kdyby ta aplikace byla zábavná
9. na výuku cizích jazyků
10. grafika; jasné zadání
11. ne android
12. ano - hezký vědec
13. úvodní obrazovka by nemusely být moc interaktivní; aby se z toho dobře učilo

Participant 8

1. ano
2. spíše ne
3. má to své výhody, ale přímý zážitek je lepší (pitva žáby)
4. v případě anonymních výsledků login a heslo, jinak stačí jméno
5. loginem a heslem
6. zablokovat spuštění jiných aplikací
7. email
8. raději použít svůj
9. jazyky, matika
10. přehledné, výhodnější oproti knížkám (interaktivita)

11. -

12. ano

13. chce kliknout na "pozici v úloze", ví, že encyklopedie je tlačítko

A.2 Poznámky z testování Vinařická horka

aktivita	poznámky z testování - 23.4
geolokace a hlavní rozcestník aplikace	V některých místech je tezke se trefit na POI. hlavne kdyz jich je vice blizko u sebe (posledni dva body se překrývají dost)
1) SAD	Udělat obrázek o něco větší a a starou fotku více kontrastní oproti fotce dnešního sadu. Ohraničit obrázek na sluníčku není ani moc videt ze tam nějaký je. Dialogové otázky se zavřou i když jsou špatně a uživatel si myslí ze to má dobře. Označit hotové "lidi". Vytvořit grafické dialogové okno, aby mělo stejnou podobu, jako ta ostatní (teď je to prostě okno, které vyskočí, když se klikne na postavu).
	Fotka původního stromu překrývá tu právě vyfocenou
	špatné zadání!!!: má to být jen ta druhá věta. První zbyla omylem z předchozího úkolu
2) ŽIVOČICHOVÉ	prosím opravit text: "Vymyslete alespoň pět..."
	Fotky se nevejdou do rámečku. Zobrazí se jen asi 3 a víc není videt a nejde ani rolovat nikam. Taky by měly jít nějakým způsobem odstranit, když se nepovedou.
	Nejdou vidět dropzony. ->skladají kolečka různé přímo na obrázky a podobně. Na správná místa by mělo doskakovat samo! Tlačítko až pod scrollem
3) BÍLÁ STRÁŇ	
	Nakonci sipka překryva text. Otázky jsou nejasně formulovane. když odpoví špatně, měla by se tečka v kolečku nějak odlišit.
	vyřešit fotky v rámečku
4) VŘESOVISTĚ	dole je nějaká modrá čára...
	grafika by měla být stejná - chce to nějak sjednotit s tím čtvercem na bílé straně
	Chyba textu!! nemá tam být "hořec", ale "koniklec"!!
6) LES	Upravená otázka - potřeba změnit v app
	nepíše to "Správně" ani nic jiného
	otázka z vřesoviste
7) GEOLOGIE	nemá být Splněno, ale vyhodnotit
	zajistit, aby zůstávali správné odpovědi, pokud je jednou označí a dají vyhodnotit
	po doplnění to padá
	V první otázce nemá být okno na psaní. Jen se mají zamyslet. V dalších třech má být Vyhodnotit, a ne Splněno. Doplnit ke každému text: Nevíte-li si rady, nebo je-li mlha, poradí vám informační tabule. Přidat další možnosti doplnění i s překlepy. Špatně tlačítko. ----- Přeformulovala jsem problematickou otázku na elektrárny. Prosím změňte.dik.j
8) VÝHLED	mohlo by doskakovat. Skladnější a obsahově bohatší by byly obdélníčky. grafici by mohli zkusit zkontrastnit ty černobílé: na sluníčku není fakt nic vidět.
	Podle některých zbytečně věc. -> ZRUŠIT A SPOJIT s 8
9) SVÁČA	

10) POLE	ještě jsem nezmáčkla vyhodnotit a už mi to odpovídá Správně/špatně
	nepíše mi to, jestli Správně/špatně
11) MUFLONI	
	Nemá tam být Splněno, ale Vyhodnotit a k tomu odpověď viz googledisk
	13tka skoci driv nez 12 ->spojit
12) LOM	horni index u pi2. Studenti potřebujou udělat si nákres a poznámky. Strašně by se hodilo vložit sem nějaké Kreslení.
	potřeba zpřeházet
	ZRUŠIT a SPOJIT s 12
13) závěr	
	nemají vymýšlet živočichy, jen zapsat, co slyší.
5) Poslouchej	když se má scrolovat dolů, mohla by tam být nějaká upozorňující šipka
Obecné poznámky:	ikonka Splněno anebo Vyhodnotit by mohla být v pravém horním rohu, nad třemi hlavními ikonami: byla by větší a tím pádem by se zvýraznily její barevné rozdíly
	všechny vstupní okna by mohly být větší.
	Zrušit bod SVÁČA a umístit jako poslední úkol na předchozím VÝHLEDU
	Budou hlášky "Správně!" a "Chyba, zkuste to znovu." nějak graficky zpracované? Byla bych pro.
	všechny přetahovací věci by mohly na předpokládaná místa samy doskakovat (přicucnout se)
	Vtipný návrh jednoho studenta: u úkolů "Prodiskutujte...." by mohlo být nahrávání jejich diskuse - JR: realizoval bych klidně i fiktivně, budou houby vědět, že se to neukládá
	zkouší náhodné odpovědi, jen aby bylo splněno, kolikrát ani nečetli samotné texty a jen klikali na náhodu - přidat nějaké hodnocení/kontrolu?
	padalo na konci splnění tasků,
	komplikace s GPSkou - občas "poskakovala"/překrytí bodů v lomu a nad lomem/některým nalezen bod někde ve křoví__JM:s tím my asi nic nenadelame, nekdy funguje dobre, jindy hur. Domluv druzicim!
	dropzony málo kontrastní - snažili se většinou přetahovat na "napevno zadané texty"
	přáli by si, aby na mapě byla šipka ve směru natočení (jakým směrem jdou)
	po 4.tasku se ztratili snad všichni a lezli bůhví kudy__JM:dobře jim tak!
	Když přecházím na další stránku, mělo by to automaticky skočit na vršek stránky. Teď se musím na vršek pracně dorolovat a někdo si ani nevšimne, že není na začátku otázky.
	Na úvod by se udělalo okno na procvičení práce s tabletem a úkoly.

B OBSAH CD

- Zdroj - složka obsahuje zdrojové kódy aplikace.
- Aplikace- instalační soubory aplikace ve verzi pro výuku a pro testování a návod k instalaci aplikace na tablet.
- Dokumenty – složka obsahující bakalářskou práci ve formátu PDF a DOCX