



Středoškolská technika 2014

Setkání a prezentace prací středoškolských studentů na ČVUT

W2Go Maturita Challenge accepted

Pavel Fojtík, Jana Prášková

Purkyňovo gymnázium, Strážnice

Masarykova 379, Strážnice

W2Go Maturita
Challenge accepted

Obsah

1	DIDAKTICKÉ HLEDISKO	8
1.1	ČASOVÁ VARIABILITA	8
1.2	OTÁZKA IMAGE	8
1.3	CENA PRODUKTU	9
1.4	EFEKTIVITA UČENÍ	9
1.5	TEST EFEKTIVITY UČENÍ.....	9
1.6	PODMĚTY Z DALŠÍ TESTŮ	10
1.7	VYUŽITÍ APLIKACE VE VÝUCE	11
1.8	ALTERNATIVY	11
2	OS ANDROID	12
2.1	SMARTPHONE JAKO TECHNOLOGIE KTERÁ DOBÝVÁ SVĚT	12
2.2	OPERAČNÍ SYSTÉMY PRO MOBILNÍ ZAŘÍZENÍ.....	12
2.3	ANDROID - NÁSTROJ KE ZKROCENÍ SMARTPHONE.....	13
2.4	STRUČNÁ HISTORIE OS ANDROID.....	14
3	PROGRAMOVÉ ŘEŠENÍ	14
3.1	PRÁCE S AKTIVITAMI.....	14
3.2	PRÁCE SE ZÁMĚRY	16
3.3	ZÁPISY DO MANIFESTU	16
3.4	TVORBA GRAFICKÉHO ROZHRANÍ.....	16
3.5	LOGIKA APLIKACE	17
3.6	REALIZACE PROGRAMU	17

Seznam obrázků

Obr.1.	Učení se z tabletu	8
Obr.2.	Studentka soustředěná na procvičování slovíček	10
Obr.3.	Využití ve výuce	11
Obr. 4.	Životní cyklus activity.....	15
Obr. 5.	GUI výběr módu	17
Obr. 6.	XML deklarace radibuttonu.....	18
Obr. 7.	GUI zobrazení slovíčka.....	18
Obr. 8.	GUI aktivita nastavení.....	18
Obr. 9.	GUI výběr témat	18

Úvod

Mobilní technologie nás v posledních letech stále více ovlivňuje a stává se nedílnou součástí našeho života. Narážíme na ni ve všech oblastech života a naprostá většina studentů této technologii již zcela podlehla. Bez mobilního telefonu či tabletu nejsou někteří studenti schopni udělat již ani krok a vypěstovali si na těchto zařízeních závislost. Nelze však nepoznamenat, že tyto mobilní technologie nám ulehčují život v mnoha oblastech. Ve spojení s mobilním internetem jsou především chytré telefony neuvěřitelně silným nástrojem moderní doby. Rozhodl jsem se tedy věnovat vývoji aplikace pro chytré mobilní zařízení, která by mě a mnoha studentům pomohla si život opět o něco ulehčit.

Jako platformu jsem si zvolil OS Android, který se v posledních letech stal leadrem trhu a je obsažen v různých modifikacích na většině prodávaných zařízeních. Při programování aplikace na Android jsem také mohl využít svých dosavadních znalostí jazyka Java a XML, proto byla tato platforma pro mne jasnou volbou.

Samotná aplikace pak dokáže navýšit čas, který student stráví učením. Studenti často tráví svůj čas čekáním a cestováním. Proč tedy tento čas nevyužít ke studiu, když je to při tom tak snadné. Stačí mít při sobě svůj mobilní telefon.

Jako student maturitního ročníku vím, že jeden z nejdůležitějších a nejpotřebnějších předmětů vůbec je výuka anglického jazyka. Proto jsem se svůj čas rozhodl věnovat napsání programu, který by mne učil právě anglické slovíčka, na které si jinak nedokážu já ani spousta dalších studentů nalézt čas.

Slovíčka jsou rozdělena do dvaceti pěti celků, a to podle maturitních témat. Studenti, nejen maturitních ročníků, tak mají skvělou příležitost, jak se připravovat k maturitě vždy a všude, co jen to baterie telefonu dovolí.

Vývoj programu ovlivňovali i testovací uživatelé, od kterých jsme dostávali zpětnou vazbu a systém tak uzpůsobili jejich představám

1 Didaktické hledisko

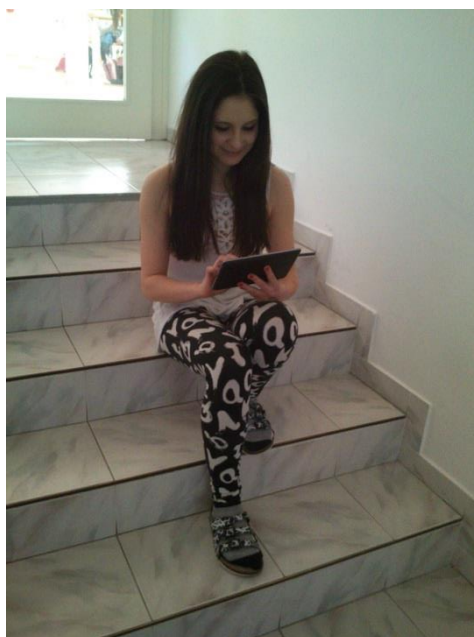
Učební pomůcky již snad od počátků vzdělávání usnadňují studentům výuku. Postupně se vyvíjejí a přicházejí nové, rychlejší způsoby. Asi nejnovějším trendem na poli výukových programů jsou mobilní aplikace. Tyto aplikace umožňují to, co žádný jiný systém prozatím nedovoluje, a to nosit si veškeré didaktické pomůcky v kapse a sáhnout po nich kdykoliv je zapotřebí.

1.1 Časová variabilita

Hlavní předností našeho výukového programu je skutečnost, že po něm můžeme sáhnout prakticky kdykoliv. Pokud si studenti navyknou náš výukový program používat například během cestování veřejnou dopravou, znamená to pro ně obrovský nárůst doby strávené učením. Testované subjekty potvrdily, že po týdnu používání naší aplikace, již automaticky při vstupu do vlaku zapínali výukový program a procházeli si nahraná slovíčka. Učení jim připadalo mnohem rychlejší a pohodlnější než v případě papírové verze

1.2 Otázka image

Asi nekladnější ohlasy získala aplikace v oblasti image samotných testovaných subjektů. Zatím co studenti, kteří se učili v dopravních prostředcích z papírové verze, byli často terčem posměchu nebo byli označováni za šprty, tak v případě učení se pomocí naší aplikaci docházelo buď to k neutrálnímu nebo naopak kladnému hodnocení vykonávané aktivity.



Obr.1. Učení se z tabletu

Testované subjekty uváděly velké změny v oblasti vnímání jejich snažení kolektivem. Část lidí často ani nezpozorovala, že by se testované subjekty učily, část pak obdivovala nový způsob učení a označovala jej jako „cool způsob, jak se šprtat“.

Fakt, že učení se z mobilních zařízení je považováno za něco moderního, za něco nadčasového dodává studentům pocit, že to co dělají je pro ně opravdu dobré a dochází tak ještě k častějšímu používání programu, protože když studenti i jejich okolí tento typ výuky vnímá pozitivně, nedochází frustraci a pocitům zbytečnosti a učení se stává mnohdy i zábavným, nebo něčím, co začneme brát zcela automaticky, bez jakéhokoliv odporu.

1.3 Cena produktu

Zajímavou zpětnou vazbu jsme dostali v případě ceny produktu. Ačkoliv náklady na pořízení elektronické verze slovíček jsou nižší, než na verzi papírovou, a taktéž elektronické verzi nehrozí zničení či opotřebení, studenti do ankety uváděli, že kupovat si program pro učení slovíček jim přijde zbytečné a raději by sahalo po papírové verzi.

Usoudili jsme, že se jedná o zažitý stereotypy a fakt, že studenti jsou zvyklí používat programy, které jsou zdarma nebo získat bezplatné studentské licence, ale nikoliv si za elektronickou podobu čehokoliv platit. Doufáme, že tato skutečnost nebude příliš bránit distribuci aplikace.

1.4 Efektivita učení

Na efektivitu učení pomocí takového programu byli rozporuplné názory, které byly hodně subjektivního charakteru. Rozhodli jsme se tedy realizovat několik testů, které nám měli poukázat na efektivitu učení pomocí mobilních aplikací.

1.5 Test efektivity učení

Tento test byl proveden celkem na dvaceti studentech maturitního ročníku. Polovina z nich dostala slovíčka v papírové verzi, druhá polovina pak dostala slovíčka v podobě našeho programu. Na naučení slovíček dostali časový limit 7 minut a měli se naučit celkem 30 slovíček.

Aby test byl, co nejvíce objektivní nešlo o česko-anglická slovíčka, ale o slovíčka česko-litevské. Litevština je jazyk, se kterým nikdo z testovaných subjektů nepřišel do styku. Slovíčka byly přeloženy pomocí google translátor.

Po uplynutí času byl ihned subjektům rozdán test, který se dělil na dvě části. První část kontrolovala překlad slovíček z češtiny do litevštiny, část druhá pak testovala slovíčka v převráceném pořadí.

Za 24 hodin došlo k zopakování testování, avšak již bez předchozího procvičování slovíček. Byla tedy testována jak krátkodobá tak i dlouhodobá paměť.

Výsledky testu byli poměrně překvapivé. Subjekty, které se slovíčka učili z mobilní aplikace, měli přibližně o 14% - 19% lepší výsledky při překladu slovíček z češtiny do litevštiny v případě testování krátkodobé paměti. Naopak při překladu z litevštiny do češtiny byli o 8% - 13% horší než subjekty které měli k dispozici papírovou formu

Snažili jsme se přijít na to, co způsobuje tak razantní rozdíl. Po konzultaci se studentem psychologie nám bylo doporučeno zkusit zobrazovat české slovíčko rovnou s překladem. Mnělo by tak teoreticky dojít k lepšímu propojení slovíček, protože je mozek vidí současně a nikoliv odděleně.

Tento návrh jsme testovali a došli jsme ke dvěma závěrům. Zobrazení slovíčka i jeho překladu současně má skutečně vliv na zmenšení rozdílu výsledků při testování překladu do nebo z litevštiny, ale

celková úspěšnost pak klesá a jen mírně (0%-5%) zlepšuje efektivitu v případě použití mobilní verze oproti verzi papírové. Procento lepší úspěšnosti navíc ještě o něco více kleslo v případě dlouhodobé paměti.

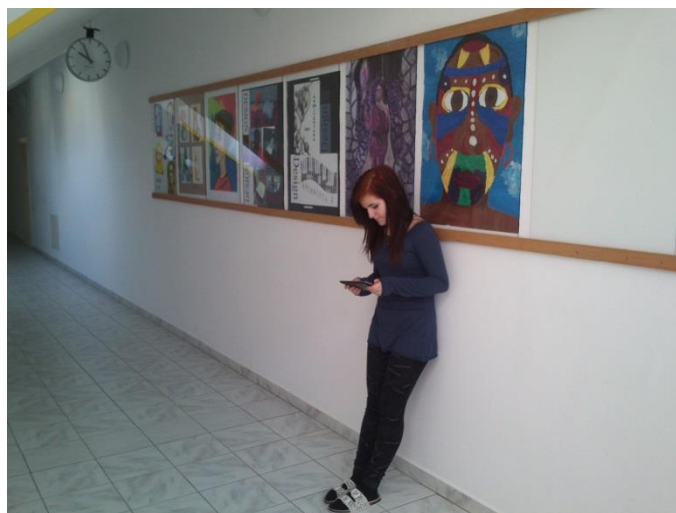
Nakonec jsme tuto problematiku vyřešili možností vybrat si, zda se bude nejdříve zobrazovat české slovíčko nebo jeho příslušný překlad. Střídáním těchto módů lze docílit asi největší efektivity učení.

Mnohem zajímavější byli výsledky dlouhodobější paměti. Po 24 hodinách se čísla změnily. V případě překladu z češtiny do litevštiny naše aplikace dosahovala téměř o 32%-37% lepších výsledků než v případě papírových podkladů. V případě převráceného překladu byla naše aplikace srovnatelná s papírovou verzí.

Z dlouhodobého hlediska se tedy zdá výhodnější používat elektronickou formu. Domníváme se, že lepší výsledky oproti papírové verzi má tato forma učení slovíček proto, že jak studenti uváděli, je mnohem snadnější udržet pozornost v případě elektronické verze, než v případě papíru kde vždy dochází k vnímání vícero slovíček. Zároveň si není potřeba hlídat aktuální pozici, což má zřejmě taktéž příznivý dopad na učení.

1.6 Podměty z další testů

Dalšími typy testů jsme též zjistili zajímavé skutečnosti. Jednalo se sice již o méně objektivní testy, ale i subjektivní názor měl pro nás jistou váhu. V prvních verzích aplikace obsahovala tlačítka pro zobrazení překladu nebo pro načtení dalšího slovíčka, či pro opětovné zařazení slovíčka do fronty.



Obr.2. Studentka soustředěná na procvičování slovíček

Tyto tlačítka rozptylovaly pozornost testovaných subjektů, kteří pak podávali horší výkony. Samotní studenti pak dodávali, že by přijali hladké uživatelské prostředí bez jakýchkoliv rozptylovacích prvků. V další verzi jsme tedy tyto prvky odstranili a nahradili je gesty, které se člověk sice musí naučit, ale pak umožňují mnohem plynulejší procházení slovíček.

Dalším velmi subjektivní poznatek je spojen s pozadím. To nejdříve bylo bílé. Bílá barva má na display velký jas a rychle unavuje oči. Studenti uváděli, že modrá barva pozadí je mnohem příjemnější a slovíčka se jim učí líp.

Několik málo testovaných studentů uvedlo, že v případě používání aplikace s modrým pozadím omezili během učení chatování na facebooku. Tato informace by stála jistě za podrobnější prozkoumání, protože jestli podobné barevné tónování, jako je použito na facebooku, snižuje potřebu používání této sociální sítě, jednalo by se o velmi cenný poznatek, který by pomohl zvýšit efektivitu některých úkonů.

1.7 Využití aplikace ve výuce

Naše aplikace může najít své uplatnění i ve výuce. Testovací režim nabízí poměrně snadný způsob, jak zjistit úroveň znalostí studentů. Bylo uvažováno i o modifikaci aplikace, která by kantorům umožňovala přes učitelský tablet, či mobil zadávat testy a přijímat výsledky od studentů. Tato forma je obzvláště vhodná do učeben vybavených tablety.



Obr.3. Využití ve výuce

1.8 Alternativy

V současné době již vzniká jedna alternativa aplikace. Nese název W2Go Top 3000. Tato aplikace má databázi tří tisíc nejpoužívanějších anglických slov, které je možné procházet obdobným způsobem jako u popisované aplikace.

Není nutné se však omezovat jen na anglický jazyk. Do budoucna je taktéž plánované aplikaci vydat v mutacích pro německý a ruský jazyk.

Bylo uvažováno též nad alternativou, která by umožňovala procvičování a testování matematických a fyzikálních vzorečků.

2 OS Android

Svět informačních technologií se v posledních letech rozvíjí až neuvěřitelným tempem. Dá se předpokládat, že tento trend bude i nadále pokračovat a řada z nás bude s tímto světem stále více propojena. Je tomu již pár let, co se na trhu objevily smartphony (chytré telefony). Tato zařízení posunula naše možnosti neuvěřitelným způsobem vpřed. Využívá je stále více uživatelů. Cílem této části je seznámit Vás s právě těmito chytrými zařízeními a především s jejich softwarovým řešením, konkrétně s operačním systémem Android a umožnit Vám nahlédnout na vývoji aplikací pod OS Android.

2.1 Smartphone jako technologie která dobývá svět

Smartphone neboli „chytrý telefon“ je nový typ mobilního telefonu, který používá mnoho nových pokročilých funkcí, které u běžných telefonů nenaleznete, a to hlavně z důvodu jejich ať už hardwarové nebo softwarové náročnosti. Na rozdíl od klasických mobilních telefonů je smartphone vybaven operačním systémem a chová se tedy jako malý počítač. Jeho velikost není větší než velikost klasického mobilního telefonu, a také proto je označován jako Personal Pocket Computer (PPC) neboli malý kapesní počítač.

Vůbec první smartphone by vyroben v roce 1992 společností IBM a nesl jméno Simson. Dalším mezníkem na poli chytrých telefonů byl smartphone od společnosti Nokia, a to model Nokia 9210, který byl prvním smartphonem, vyrobený s barevným displejem a byl to první smartphone s tzv. „otevřeným“ operačním systémem.[11] Od té doby šel vývoj značně kupředu až do podoby, jakou známe dnes.

V dnešní době již všechny smartphony vlastní operační systém. Díky operačnímu systému je možné do smartphonu nahrát různé aplikace a využívat tak potenciálu těchto telefonů nejen k obyčejné komunikaci. Smartphony nabízejí také personalizaci a zároveň umožňují vytvořit vlastní spustitelnou aplikaci a tím podporují růst skupiny vývojářů mobilních aplikací.

2.2 Operační systémy pro mobilní zařízení

Trh dnes nabízí spoustu operačních systémů, a to jak známých a rozšířených, tak i průkopníků v různých specializovaných oblastech. Níže jsou uvedeny nejnámější z nich.

Stručný přehled OS pro smartphony:

- Android – Platforma na linuxovém jádru určená jak pro mobilní zařízení, tak pro tablety. Tuto platformu vyvíjí společnost Google.
- iOS – Platforma od Applu, na které běží iPhone, iPad, iPod. Platforma je dostupná jak pro Tablet PC, tak i mobilní a kapesní zařízení.

- Windows Phone 7 – Platforma do Microsoftu. Značně ztrácí na své popularitě a je stále méně distribuován.
- BADA – OS od společnosti Samsung, který je postaven stejně jako Android na linuxovém jádře, kombinuje dohromady grafické prostředí z Androidu a iOS. Na trhu je zatím jen pár telefonů s OS BADA.
- Symbian - Dříve velmi populární. Osazován hlavně do mobilních telefonů značky Nokia. S nástupem iOS či Androidu nastal velký úpadek této platformy a stále ubývá zařízení používajících Symbian.
- MAEMO (MeeGo) – Opět linuxové jádro (distribuce Debian). Vývoj: Nokia.
- Web OS – Dříve Palm OS. Podpora od společnosti HP. V budoucnu by měli Web OS obsahovat dokonce i tablety.
- RIM – Blackberry OS je populární hlavně v Americe. Slouží především manažerům nebo pro firemní účely.

2.3 Android - nástroj ke zkrocení smartphone

Android je jedním z výše uvedených operačních systémů, který si na trhu vybojoval v posledních letech významnou pozici. Jedná se o software s otevřeným zdrojovým kódem (open source platforma), což umožňuje při dodržení daných podmínek systém zdarma užívat, přetvářet, popřípadě za splnění jistých podmínek distribuovat.

Podobně jako se samotným systémem, který je postaven na linuxovém jádře, je to i s danými aplikacemi, jejichž vývojem se tato práce bude zabývat (Android aplikace jsou programovány v jazyce Java, jehož knihovny jsou taktéž distribuovány zdarma). Aplikace, na kterých uživatel pracuje, nepřistupují k jádru přímo, ale komunikují pomocí Android API. Samotným jádrem se není potřeba zabývat. Pozornost je dobré věnovat spíše systémovým vrstvám (OS Android je postaven na pěti vrstvách-Linux Kernel; Libraries; Android Runtime; Application Framework; Applications), které bude potřeba znát k vytvoření základních aplikací spustitelných na smartphonech. Nejdůležitější vrstvou pro vývojáře je vrstva s názvem Application Framework. Ta umožňuje přistupovat jak k hardwarovému vybavení přístroje, tak jako využít možnosti pracovat s obsahem jiných aplikací či ovládat celý životní cyklus aplikace. Důležité jsou i vrstvy Android Runtime a Applications. Android runtime obsahuje virtuální stroj DVM a základní knihovny Apache pro práci ze sítí. Poslední, nejvyšší vrstva zahrnuje již samotné aplikace, a to ať již pře-dinstalované, stažené z Google play nebo právě vývojáři vytvořené aplikace.

2.4 Stručná historie OS Android

Android je operační systém vytvořený, každému jistě známou, firmou Google. Jak bylo řečeno, je založen na open source platformě a již to naznačuje, že byl vybudován zcela od základů. V roce 2003 byla v Kalifornii založena společnost Android několika soukromníky, kteří začali vyvíjet mobilní aplikace. Rok 2005 byl pro firmu Android velkým mezníkem, jelikož se o něj začala zajímat společnost Google a nakonec došlo i k odkupu.[13] Firma Android se tak stala dceřinou společností Googlu a došlo k převzetí veškerých závazků i zaměstnanců.

K samotnému představení operačního systému Android došlo až v roce 2007, a to v den vzniku Open Handset Alliance, která sdružuje desítky firem pohybujících se na mobilním trhu, a to jak samotné operátory, tak i výrobce jednotlivých součástí pro smartphony. Cílem této aliance bylo vytvořit a udržet otevřené standardy pro chytrá zařízení.

V průběhu let si Android dobýval své místo na trhu a bojoval s konkurencí. První oficiální verze OS Android šla na trh 3. září 2008, a to v mobilním zařízení nesoucí název HTC Dream. V průběhu let pak vycházely další verze OS, nesoucí s sebou vylepšení, které uživatelé jistě ocení, přidání nových funkcí a vyladění chyb. Nejnovější verze androidu je Android 4.4 nesoucí název Kit Kat.

3 Programové řešení

Většina zdrojového kódu je psána v jazyce Java. Programování pro OS Android se vyznačuje jistými odlišnostmi od klasické syntaxe, tyto rozdíly však není potřeba zmiňovat. Pro tvorbu GUI jsem využíval vestavěného GUI editoru, jenž je součástí Android pluginu pro vývojové prostředí Eclipse v kombinaci s úpravou generovaného kódu v xml. Pro neměnné řetězce a hodnoty jsem taktéž upřednostňoval zápis v podobě xml. Pro zprovoznění některých specifických funkcí či například deklaraci užitých aktivit a záměrů bylo nutné provádět i zápisy do manifestu aplikace. Obrázky a grafika jsou pak ukládány podle standart pro Android resources. Pro ukládání pomocných informací, jako jsou například informace o licenci a dané kopii, je použito pouhé ukládání do souborů, jež je na rozdíl od klasických Java aplikací na Androidu dostatečně zabezpečeno proti čtení třetí stranou.

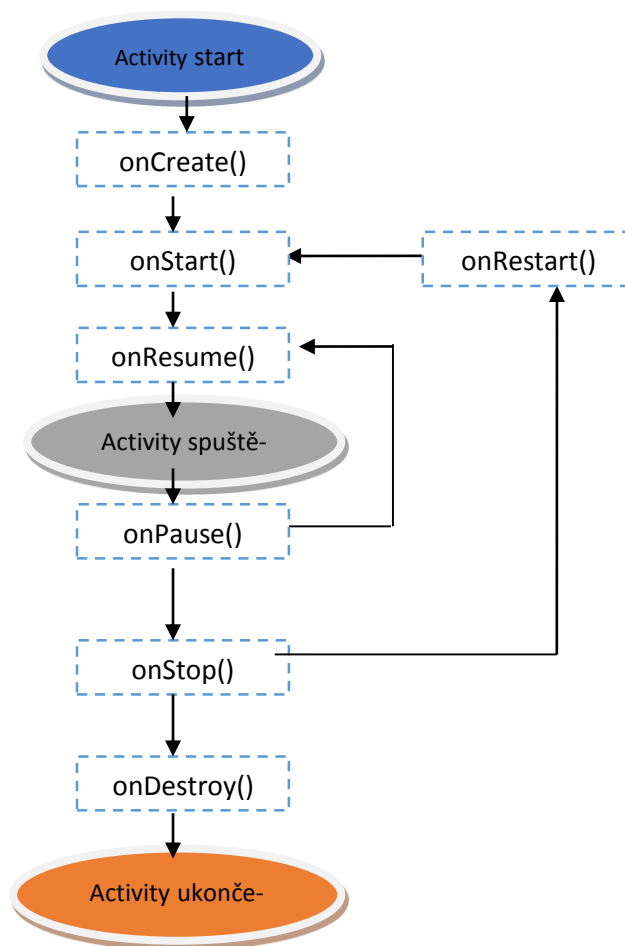
3.1 Práce s aktivitami

Jedna aktivita vždy znázorňuje jednu obrazovku aplikace. Tvoří ji uživatelské rozhraní (GUI) ukládané ve formě xml souboru a logika jazyka Java využívající knihovny dle příslušné verze OS Android.

Každá aktivita má svůj vlastní životní cyklus, může být například na popředí a mít uživatelský vstup, může být viditelná třeba jen částečně nebo může být celá skryta na pozadí. Pro přechod mezi jednotlivými aktivitami se používají takzvané callbacky, které určují, v jaké části životního cyklu se daná aktivita nachází.

Základní metody:

- `onCreate()` – Tato metoda je volána okamžitě při vytváření aktivity. Používali jsme ji například k odkazování a definování GUI, nastavování hodnot, jež mají být nastaveny ihned při vytváření aktivity, či například k volání metod, které prováděly potřebné operace po vytvoření aktivity. Jedná se o primární prostředek na práci s aktivitami (viz obrázek).
- `onStart()` – Tato metoda je volána, když se aktivita stane viditelnou pro uživatele.
- `onResume()` - Volána, když aktivita začne dostávat uživatelský vstup. V tento moment je tedy na vrcholu zásobníku a není nijak překryta jinou aktivitou.
- `onPause()` – Je volána těsně před zamrznutím aktivity.
- `onStop()` - Zavolána, když aktivita přestává být viditelná pro uživatele.
- `onDestroy()` - Volána před zrušením samotné instance aktivity.
- `onRestart()` - Tato metoda je volána těsně předtím než aktivita, která byla zastavena, je znovu vykreslena.



Obr. 4. Životní cyklus activity

3.2 Práce se záměry

Záměry jsou velmi mocným a esenciálním nástrojem, jenž skýtá neuvěřitelné možnosti. Ve většině případů jsou použity jako můstky mezi jednotlivými aktivitami. Záměr (intent) je jednoduchý objekt. Můžeme si jej představit jako jakýsi kontejner činnosti, která se má vykonat, parametru, nad kterým se má vykonat a aplikace, která má tuto činnost provést. Jedná se vlastně o jakousi abstrakci operace, která se má vykonat.

Záměry dělíme na explicitní a implicitní. Použití implicitních záměrů je velmi jednoduché. V přiložené aplikaci se používají implicitní záměry, například ke spuštění jisté aktivity a předání potřebných informací, akcí aktivitě, kterou chceme spustit. Použití explicitních záměrů je mírně složitější operace, která však má velmi široké využití. V případě explicitních záměrů neurčujeme aktivitu, které mají být data a akce předány, ale říkáme, o jaký typ aktivit se zajímáme. Je pak na uživateli, či systému jak s daným záměrem naloží. Takto je možné sdílet data a akce mezi aplikacemi, z různých zdrojů a samotným systémem.

3.3 Zápisy do manifestu

Při tvorbě samotného programu bylo potřeba zápisů i do manifestu vyvíjené aplikace. Jednak pro deklaraci aktivit samotných, tak i třeba pro přidávání různých záměrových filtrů, které určují, jaké záměry lze nad danou aktivitou explicitně spustit a jaká data jsou ještě přípustná. Dále se v manifestu uvádí například jméno a verze příslušné aplikace, či cesta k hlavní (spouštěcí aktivitě, která má obdobnou roli jako třída main při klasické Javovské syntaxi.

3.4 Tvorba grafického rozhraní

Pro tvorbu grafického rozhraní bylo užitých standardních knihoven, které poskytuje použité vývojové prostředí Eclipse. V tomto programu je dokonce i zabudován základní grafický editor, který nabízí možnost nahlédnout do kódu a kompletně jej ručně editovat. Jak již bylo zmíněno, OS Android používá pro svou grafiku zdrojové soubory ve formátu xml. Celý GUI bylo vyvíjeno tak aby bylo zcela intuitivní a uživatel měl možnost pomocí několika mála dotyků docílit přesně toho, čeho si žádá. Veškerá grafika, která byla v programu použita, byla vlastnoručně navržena přímo pro tento program. Při tvorbě programu bylo implementováno mnoho pokročilých metod jako například sledování souřadnic pohybu, sledování doteků a měření trvání doteků. Těchto metod bylo využito k co nejpříjemnějším a modernímu ovládní aplikace.

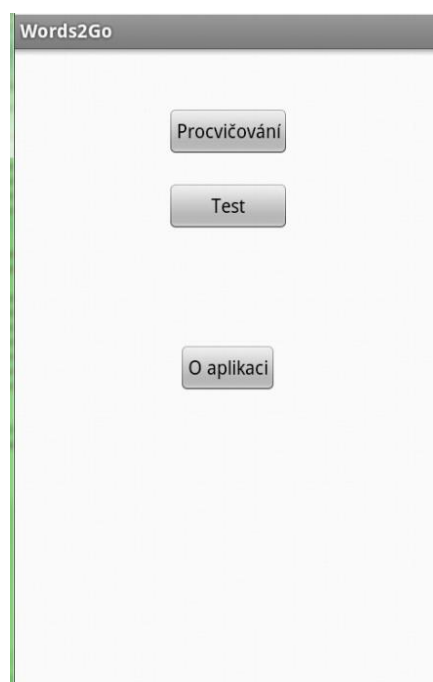
3.5 Logika aplikace

Logika aplikace je psána syntaxí jazyka Java. Knihovny používané OS Android jsou velmi podobné knihovnam klasickým, avšak jsou optimalizovány pro běh na mobilních zařízeních. OS Android nabízí i podporu klasických Java knihoven, ale jejich implementace především kvůli vysokému energetickému a výpočetnímu nároku není doporučována. Hlavními odlišnostmi od toho, co známe pod pojmem syntaxe jazyka Java, je absence třídy main. Taktéž používání konstruktorů je omezeno na minimum. Většina tříd jsou potomky třídy activity, jsou tedy samostatně spustitelné jako by se jednalo o třídu main. Při spouštění jiné třídy nevytváříme její instanci, ale vytvoříme záměr jejího spuštění, který je převzat příslušným launcherem přítomným na daném zařízení a následně je třída určena záměrem spuštěna a byla-li jí jinými záměry přidána příslušná data, jsou k dispozici v průběhu trvání celé životnosti aktivity, ovšem standardně jsou volána a převedena do lokálních proměnných již v metodě onCreate(); která je volána vždy při startu každé aktivity.

3.6 Realizace programu

Program má dva módy. První mód je určen k rychlému procvičování slovíček, druhý pak k jejich testování. Jelikož hodlám program distribuovat, rozhodl jsem si vytvořit vlastní systém licencování. V přiložené verzi programu je tato funkce vypnuta a po kliknutí na button aktivovat v úvodní aktivitě, která se zobrazí při prvním spuštění, jsou dostupné veškeré funkce programu. Další obrazovka pak nabízí výběr mezi zmiňovanými módy a nápovědu.

Po výběru jednoho z módů dojde k spuštění aktivity s výběrem příslušných témat. Tato aktivita je shodná pro oba módy, pouze předává informaci o módu v podobě záměru dále do dalších aktivit.



Obr. 5. GUI výběr módu

Témat je celkem 25, tak jako samotných maturitních témat. Pro implementaci výběru bylo použito prvku radiobutton. Každý jeden prvek pak po zaznamenání doteku volá přetěžovanou metodu onClick(), která je obsažena v defaultních knihovnách. Akci po doteku deklarujeme vlastní metodou ve zdrojovém kódu activity. Na kterou odkazujeme zadáním jejího názvu do xml deklarace daného prvku.

```
<RadioButton
    android:id="@+id/radio0"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/switch1"
    android:layout_marginTop="16dp"
    android:onClick="onRadBtnClick0"
    android:text="1) Personal history, family and friends" />

<RadioButton
    android:id="@+id/radio1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/radio0"
    android:onClick="onRadBtnClick1"
    android:text="2) Our town/village" />
```

Obr. 6. XML deklarace radiobuttonu

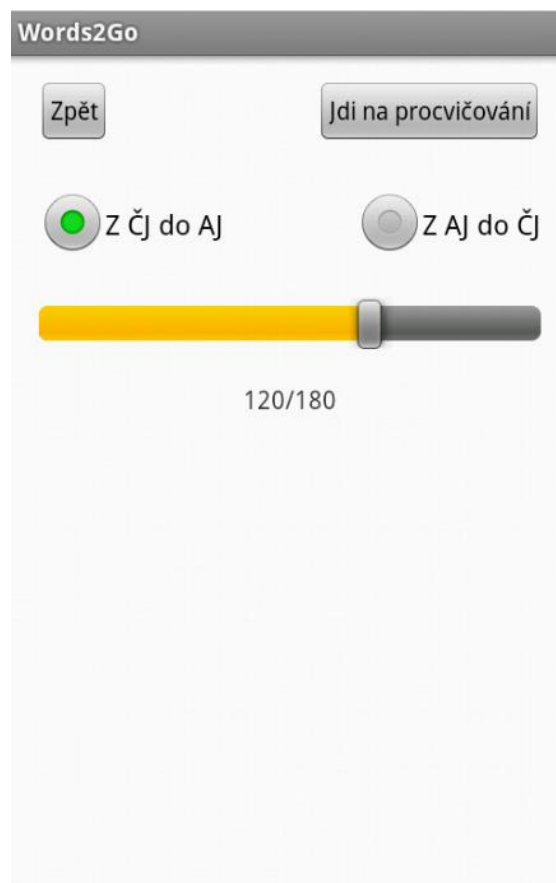
Po výběru příslušných témat a kliknutím na tlačítko další dojde k spuštění další activity již jsou předány záměry o příslušném módu a informace o vybraných tématech v podobě pole datového typu boolean [].

V další aktivitě dojde k dotazu na počet slovíček daných témat a následnému výpočtu celkově dostupných slovíček pro vybrané témata. Pomocí implementovaného seekbaru pak lze jednoduše pomocí posunu nastavit, kolik slovíček má být procvičováno, nebo testováno.

V následující aktivitě pak již dojde ke spuštění zvoleného módu. Uprostřed se zobrazí slovíčko dle příslušného systému řazení (buď anglicky, nebo česky). V pravém horním rohu se pak zobrazuje pro informaci počítadlo. Při prvním spuštění této activity se zobrazí i nápověda, jak aplikaci ovládat. Při tvorbě ovládacích prvků byl brán potaz hlavně na rychlost a efektivitu. Krátkým dotekem se zobrazené slovíčko přeloží, dlouhým podržením se zobrazí další slovíčko. Rychlým přetáhnutím zleva doprava se načte další slovíčko, rychlým přetáhnutím zprava doleva se slovíčko přidá do seznamu slovíček pro opětovné opakování. Po přidání slovíčka do seznamu pro opětovné opakování se objeví v levém horním rohu počítadlo slovíček v tomto seznamu. Dojde-li k procvičení všech slovíček, seznam slovíček k opětovnému opakování se přesype do defaultního seznamu (počítadlo v pravém rohu) a uživatel pokračuje v procvičování těchto slovíček, dokud nevyprázdní oba seznamy.



Obr. 9. GUI výběr témat



Obr. 8. GUI aktivita nastavení



Obr. 7. GUI zobrazení slovíčka

Závěr

Vytvořený program se nad míru očekávání povedl a skvěle slouží k procvičování a rychlému učení anglických slovíček. Implementované metody jsou stabilní a do další verze je plánováno pouze vylepšení grafické části programu a zvýšení počtu slovíček. Je plánovaná i německá mutace této aplikace. Studenti tuto aplikaci přijímají velmi kladně a berou ji jako módní trend. Zpětná vazba nám dala velmi cenné informace, které jistě povedou k zlepšení celého projektu. Zcela jistě bude pokračovat další vývoj. Tento projekt nám ukázal, že mobilní výukové programy mají velký potenciál. Rádi bychom se tedy věnovali vývoji i dalších výukových programů a navázali tak na tento úspěšný projekt.

Použitá literatura a zdroje

SKALKOVÁ, Jarmila. *Obecná didaktika: vyučovací proces, učivo a jeho výběr, metody, organizační formy vyučování*. 2., rozš. a aktualiz. vyd., [V nakl. Grada] vyd. 1. Grada, 2007, 322 s. ISBN 978-80-247-1821-7.

ŠVAŘÍČEK, Roman a Klára ŠEĎOVÁ. *Kvalitativní výzkum v pedagogických vědách: vyučovací proces, učivo a jeho výběr, metody, organizační formy vyučování*. Vyd. 1. Praha: Grada, 2007, 322 s. ISBN 978-80-7367-313-0.

MUŽÍK, Jaroslav a Klára ŠEĎOVÁ. *Didaktika profesního vzdělávání dospělých: vyučovací proces, učivo a jeho výběr, metody, organizační formy vyučování*. 1. vyd. Plzeň: Fraus, 2005, 377 s. ISBN 80-723-8220-9.

KALHOUS, Zdeněk a Otto OBST. *Školní didaktika: vyučovací proces, učivo a jeho výběr, metody, organizační formy vyučování*. Vyd. 2. Plzeň: Portál, 2009, 447 s. ISBN 978-807-3675-714.

CHODĚRA, Radomír a Otto OBST. *Didaktika cizích jazyků: úvod do vědního oboru*. Vyd. 2. Praha: Academia, 2013, 447 s. ISBN 978-80-200-2274-5.

PODLAHOVÁ, Libuše a Otto OBST. *Didaktika pro vysokoškolské učitele: úvod do vědního oboru*. Vyd. 1. Praha: Grada, 2012, 209 s. Pedagogika (Grada). ISBN 978-802-4742-175.

KISZKA, Bogdan a Otto OBST. *1001 tipů a triků pro jazyk Java: úvod do vědního oboru*. Vyd. 1. Praha: Computer Press, 2009, 542 s. Pedagogika (Grada). ISBN 978-80-251-2467-3.

HENNEY, Kevlin a Otto OBST. *97 klíčových znalostí programátora: úvod do vědního oboru*. Vyd. 1. Brno: Computer Press, 2010, 240 s. Pedagogika (Grada). ISBN 978-80-251-3145-9.