4.3. Pomocník pri učení sa cudzieho jazyka

Ľudia sa vždy učili a vždy budú učiť cudzie jazyky. Dôvodov na učenie sa cudzích jazykov je veľmi veľa. Nejde len o možnosť získavať informácie z originálnych zdrojov o veciach, ktoré nás zaujímajú, príp. o kultivovanie vlastných vyjadrovacích schopností. Prostredie, v ktorom žijeme, študujeme, pracujeme a komunikujeme (osobne alebo online), je alebo sa prirodzene stáva multikultúrnym. Niekto sa učí jazyky rád a ide mu to ľahko, iný potrebuje viac času a dlhodobý tréning. Keď to myslíme s učením sa cudzieho jazyka vážne, chceme mať bohatú slovnú zásobu a správne používať gramatiku. Chceme rozumieť hovorenej reči aj písanému textu, plynulo rozprávať a správne vyslovovať, byť schopní napísať bez chýb vlastný text. Na trénovanie jazykových zručností máme okrem vyučovania v škole či v jazykovom kurze k dispozícii mnoho ďalších, digitálnych pomocníkov.

Otázky na zamyslenie

V ktorých povolaniach je nevyhnutné ovládať cudzí jazyk?

Budú sa musieť ľudia učiť cudzie jazyky aj v budúcnosti?

Ktoré cudzie jazyky sa učíte alebo chcete naučiť vy?

Venujete sa učeniu sa cudzieho jazyka cieľavedome aj mimo povinného vyučovania? Čo robíte preto, aby sa ste sa cudzí jazyk naučili lepšie a rýchlejšie?

Kľúčové slová

webová služba, strojový preklad, YandexTranslate, syntéza a rozpoznávanie reči, lokálna databáza, viac obrazoviek

Čo sa naučíme a čo si precvičíme

- dozvieme sa, čo je webová služba,
- spoznáme webovú službu zameranú na strojový preklad,
- použijeme komponent *YandexTranslate* z kategórie *Media*,
- budeme pracovať so zoznamami a lokálnou databázou,
- vhodne využijeme syntézu a rozpoznávanie reči,
- vytvoríme aplikáciu s viacerými obrazovkami.

Príprava na výučbu

Prerekvizity: senzor zrýchlenia (etuda 2.1), komponent *TinyDB* (etuda 2.4), viac obrazoviek (etuda 2.5), komponenty *TextToSpeech* a *SpeechRecognizer* (etuda 2.6), zoznamy a komponent *ListView* (etuda 2.8)

Súbory ikon pre tlačidlá a obrázky vlajok, ktoré sú použité vo vzorovej aplikácii, sa nachádzajú v archíve **media.zip**.

Prílohou kapitoly sú tiež dve verzie riešenia. V projekte **pmz_prekladac_ver1_R.aia** sú implementované požiadavky z úloh 1, 2, 3. V projekte **pmz_prekladac_ver2_R.aia** je riešenie rozšírené o ďalšiu obrazovku a prácu s lokálnou databázou.

Akú zaujímavú aplikáciu môžeme vytvoriť?

V aplikačnom obchode *Google Play* ľahko vyhľadáme rôzne mobilné aplikácie na podporu učenia sa cudzieho jazyka pre deti aj dospelých - slovníky, interaktívne cvičenia zamerané na slovnú zásobu, gramatiku, počúvanie, čítanie, hovorenie, podcasty, videá, hry pre jedného i viac hráčov. Mnohé populárne jazykové aplikácie integrujú viacero spôsobov učenia sa do jedného celku.

V našom prípade pôjde o aplikáciu, ktorá nám umožní:

- prekladať slová alebo slovné spojenia zo slovenčiny do anglického jazyka a naopak,
- zvoliť si medzi textovým alebo hlasovým ovládaním (slovo na preklad budeme môcť aj vysloviť, nielen napísať),
- zvoliť si medzi textovým a zvukovým výstupom prekladu (použijeme syntézu reči).
- zaznamenávať si vety s výskytom kľúčového slova do lokálnej databázy.

Príklady viet, v ktorých sa slovo používa v konkrétnom kontexte, sú veľmi užitočnou pomôckou pri trénovaní jazykových zručností (rozširovanie slovnej zásoby, zdokonaľovanie výslovnosti, plynulosť a gramatická správnosť vo vyjadrovaní).

Ako budeme postupovať pri tvorba aplikácie?

V našom projekte budeme musieť postupne vyriešiť rôzne podproblémy:

- navrhneme vzhľad aplikácie
  - zabezpečíme, aby texty zobrazené v rozhraní korešpondovali so smerom prekladu
  - smer prekladu bude možné ľahko zmeniť, napr. potrasením tabletu
- umožníme používateľovi zadať vstupné slovo
  - napísaním do textového poľa alebo
  - rozpoznaním hovorenej reči
- poskytneme používateľovo preklad
  - s využitím webovej služby zameranej na strojový preklad
  - výstup prekladu okrem zobrazenia aplikácia vysloví nahlas
- rozšírime aplikáciu o ďalšiu obrazovku určenú na prácu s databázou viet
- navrhneme a implementujeme ďalšie rozšírenia aplikácie

Niektoré komponenty, ktoré budeme potrebovať, sme už použili v jednoduchých etudách či iných projektoch:

- vizuálne komponenty a správcov rozvrhnutia,
- *AccelerometerSensor*,
- *TextToSpeech*,
- *SpeechRecognizer*,
- lokálnu databázu *TinyDB*.

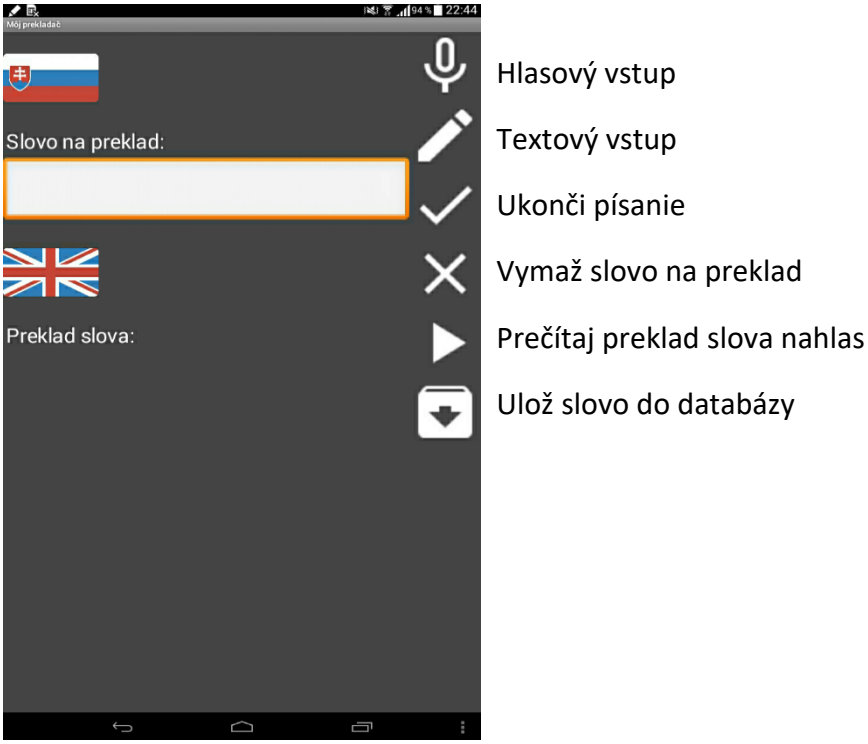
Na preklad slova využijeme webovú službu, s ktorou budeme komunikovať pomocou komponentu *YandexTranslate* z kategórie *Media*.

Úloha 1

Navrhnite vzhľad hlavnej obrazovky aplikácie, na ktorej budeme zadávať slová na preklad, získavať výsledok prekladu (a neskôr tiež umožníme otvorenie ďalšej obrazovky súvisiacej s databázou).

Ovládacie prvky aplikácie by mali byť rozmiestnené prakticky. Profesionálny dojem dosiahneme použitím vhodnej sady ikon. Ak chceme meniť smer prekladu (napr. potrasením tabletu), musíme aktuálnej situácii prispôsobiť aj vzhľad aplikácie.

Obr. 4.3.1 obsahuje príklad rozloženia potrebných vizuálnych komponentov. V ľavej časti sú použité komponenty `Image`, `Label` a `Text`, v pravej časti tlačidlá s obrázkami nastavenými vo vlastnosti `Button.Image`.

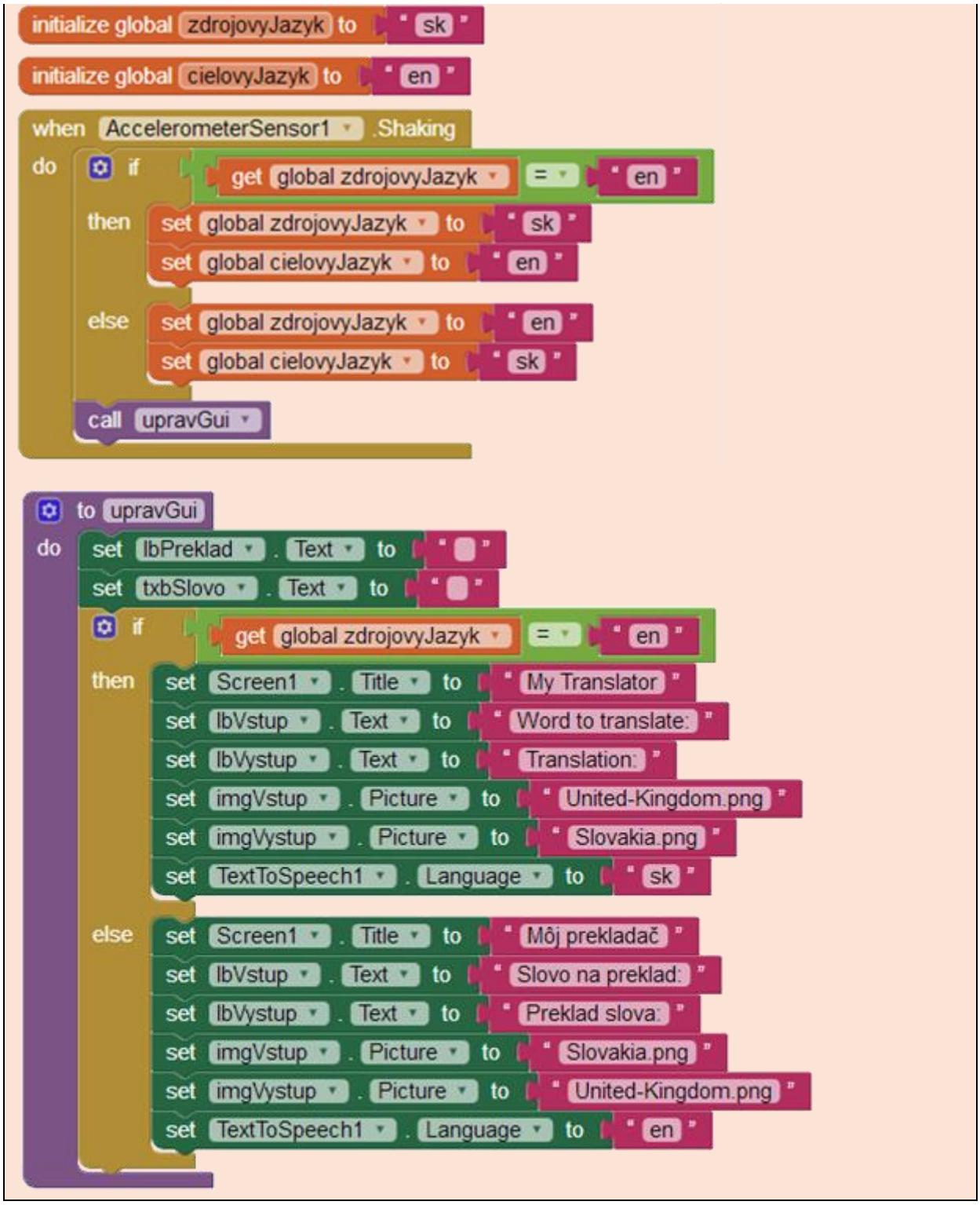


Obr. 4.3.1 Návrh hlavnej obrazovky aplikácie

Pomoc k riešeniu úlohy

Aby sme dosiahli umiestnenie tlačidiel pri pravom okraji obrazovky, použili sme kombináciu dvoch správcov rozvrhnutia. Komponent `HorizontalArrangement` obsahuje 2 komponenty typu `VerticalArrangement`. Vo vlastnostiach obrazovky nezabudnime upraviť titulok aplikácie, nastaviť orientáciu displeja na výšku (*Portrait*), farbu pozadia prispôsobiť použitým obrázkom.

V reakcii na udalosť `AccelerometerSensor.Shaking` aktualizujeme obsah globálnych premenných `zdrojovyJazyk` a `cielovyJazyk`. Výmenu textov a obrázkov v rozhraní aplikácie môžeme naprogramovať ako samostatnú procedúru `upravGui`. Nastavenie vlastnosti `TextToSpeech.Language` musí v našom prípade zodpovedať cieľovému jazyku. Syntéza reči sa totiž využíva na prečítanie výsledku prekladu:

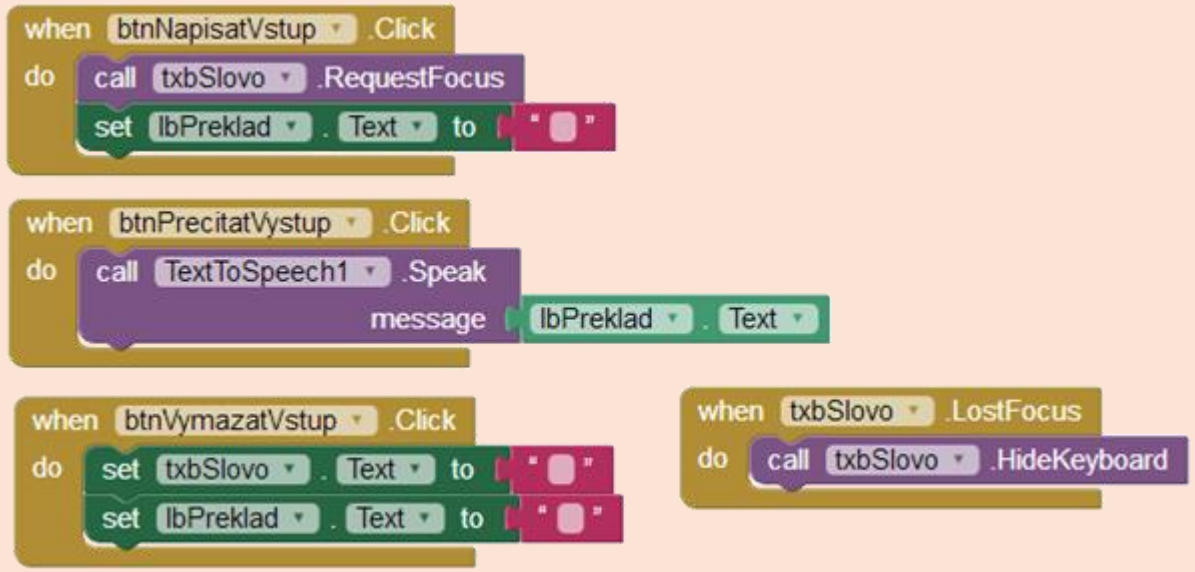


Úloha 2

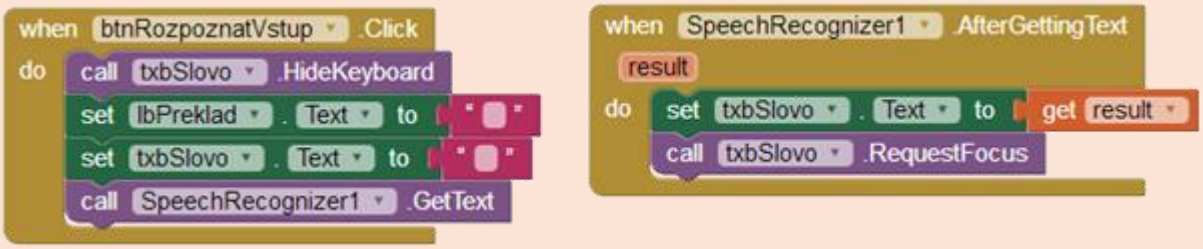
Používateľ bude aplikáciu ovládať pomocou tlačidiel. Môže sa rozhodnúť pre hlasový vstup alebo písanie na klávesnici. V používateľskom rozhraní aplikácie sú aj tlačidlá na potvrdenie vstupu (neskôr ním vyvoláme zobrazenie prekladu), vymazanie vstupného textového poľa a prečítanie prekladu zadaného slova nahlas. Naprogramujte reakcie na udalosti vznikajúce pri stláčaní jednotlivých tlačidiel. Buďte dôslední pri testovaní používateľského rozhrania aplikácie.

Pomoc k riešeniu úlohy

Pre zlepšenie používateľského komfortu môžeme pri zvolení textového vstupu zamerať textové pole `txbSlovo`. Ak toto vstupné pole stratí zameranie (teda keď nastane udalosť `txbSlovo.LostFocus`), môžeme zobrazenú klávesnicu schovať. Podobne aj v prípade voľby hlasového vstupu. Aby sme otestovali syntetizátor reči, môžeme namiesto prázdneho reťazca nastaviť do komponentu `lbPreklad` dočasne konkrétny text:



Po rozpoznaní rečového vstupu môžeme opäť zamerať textové pole so vstupom, používateľ ho možno bude chcieť editovať:



Úloha 3

Doprogramujte hlavnú funkcionalitu aplikácie, ktorou je získanie prekladu zadaného slova. Výstup prekladu sa zobrazí na obrazovke. Po stlačení tlačidla ho aplikácia prečíta nahlas.

Vysvetlíme si

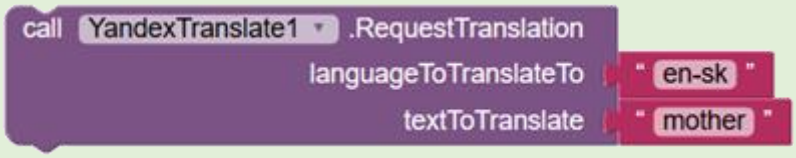
Na prekladanie slov zo zdrojového do cieľového jazyka použijeme komponent **YandexTranslate** z kategórie *Media*. Komponent potrebuje pre svoju funkčnosť internetové pripojenie, keďže pri každej žiadosti o preklad komunikuje s webovou službou Yandex (http://api.yandex.com/translate/).

Mnohé webové a mobilné aplikácie, ktoré bežne používame, sú založené na spracúvaní a remixovaní výstupov získaných od iných **webových služieb**, napr. Google Maps, Facebook,

Twitter, YouTube, Flickr a pod.). Vývojári ich môžu využívať prostredníctvom verejného API (*Application Programming Interface,* rozhranie na programovanie aplikácií).

V dokumentácii si naštudujú, akým spôsobom je možné službe posielať **požiadavky** a v akom formáte sú **odpovede**, ktoré služba aplikácii vracia (napr. JSON, XML). Niektoré služby vyžadujú, aby vývojár najprv požiadal o pridelenie jedinečného kľúča (tzv. API key), na základe ktorého je možné aplikáciu identifikovať a sledovať, akým spôsobom službu využíva (počet prístupov, objem požadovaných dát a pod.).

App Inventor poskytuje špeciálny komponent, vďaka ktorému sa nemusíme zaoberať detailami komunikácie s webovou službou Yandex. Po vložení komponentu do aplikácie nenastavujeme žiadne vlastnosti. Požiadavku na preklad realizujeme zavolaním metódy, napr.:



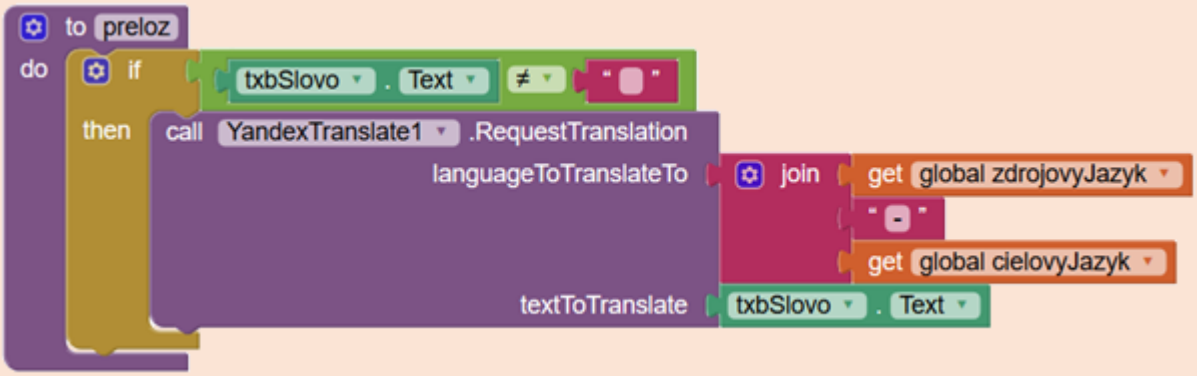
V parametroch metódy `RequestTranslation` určíme text na preloženie `textToTranslate` a jazyk, do ktorého chceme prekladať. V ukážke vyššie sme použili reťazec `"en-sk"`, čo znamená, že systém bude prekladať slovo `"mother"` z anglického do slovenského jazyka. Ak by sme prefix `"en-"`, vynechali a uviedli len kód cieľového jazyka, systém by sa najprv pokúsil zdrojový jazyk identifikovať sám. Dvojznakové kódy podporovaných jazykov nájdeme na stránke služby Yandex.

Preklad sa v aplikácii realizuje asynchrónne, na pozadí. Jeho ukončenie vygeneruje udalosť `YandexTranslate.GotTranslation`, na ktorú môžeme zareagovať. Výsledok prekladu získame z parametra `translation` (v našom príklade pôjde o reťazec `"matka"`).

Parameter `responseCode` obsahuje hodnotu, ktorá je dôležitá pre overenie úspešnosti prekladu. Ak má tento parameter hodnotu rôznu od 200, v komunikácii s webovou službou nastala chyba a preklad nie je k dispozícii. Význam kódov reprezentujúcich rôzne chybové stavy tiež nájdeme na stránke služby.

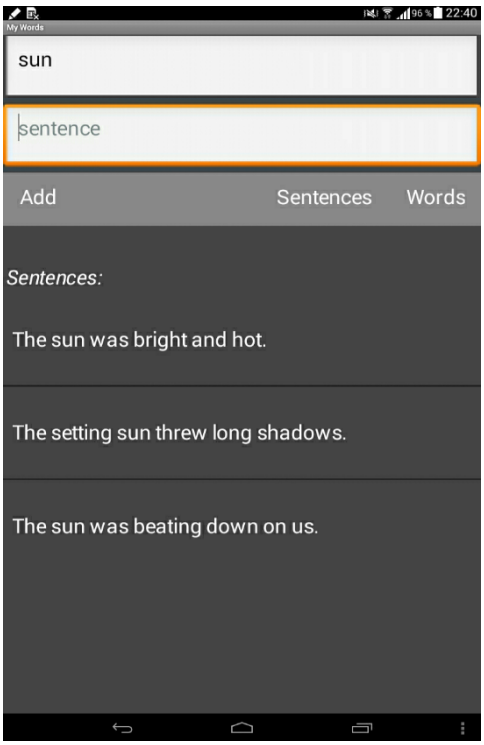
Pomoc k riešeniu úlohy

Reťazec `languageToTranslate` vytvoríme prečítaním obsahu globálnych premenných pomocou operácie spájania reťazcov:



Úloha 4

Doplňte do projektu ďalšiu, samostatnú obrazovku (komponent *Screen*), ktorá sa v aplikácii otvorí po stlačení tlačidla dostupného na úvodnej obrazovke. Druhá obrazovka má používateľovi umožniť pridávanie, vyhľadávanie a editovanie príkladov viet obsahujúcich kľúčové slovo.



Obr. 4.3.2 Zobrazenie príkladov viet pre zadané kľúčové slovo

Pri navrhovaní používateľského rozhrania aplikácie a programovaní zvážte, že:

- do databázy chceme zapisovať slová a vety v anglickom jazyku, rozhranie druhej obrazovky by preto malo byť v angličtine,

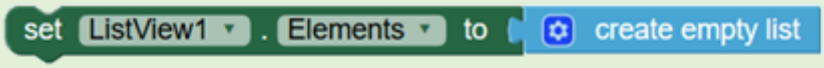
- pri inicializácii druhej obrazovky je vhodné pripraviť ako kľúčové slovo posledné anglické slovo, s ktorým sa pracovalo na úvodnej obrazovke,
- zoznamy slov a viet pre zvolené slovo môžeme zobrazovať v komponente *ListView*,
- komponent *Notifier* je užitočný pri zobrazovaní správ a upozornení pre používateľa.

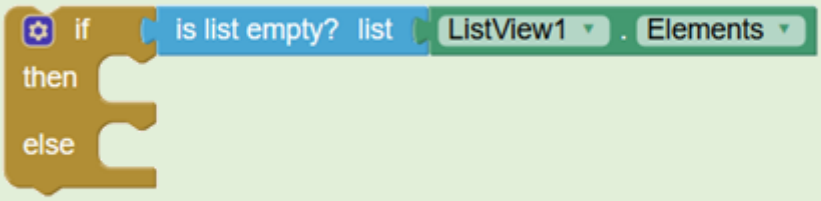
Pomôcky

Z jednej obrazovky môžeme otvoriť inú obrazovku a odovzdať jej hodnotu:

Ak sú údaje uložené v zozname, môžeme tento zoznam použiť ako zdroj dát pre komponent `ListView`.

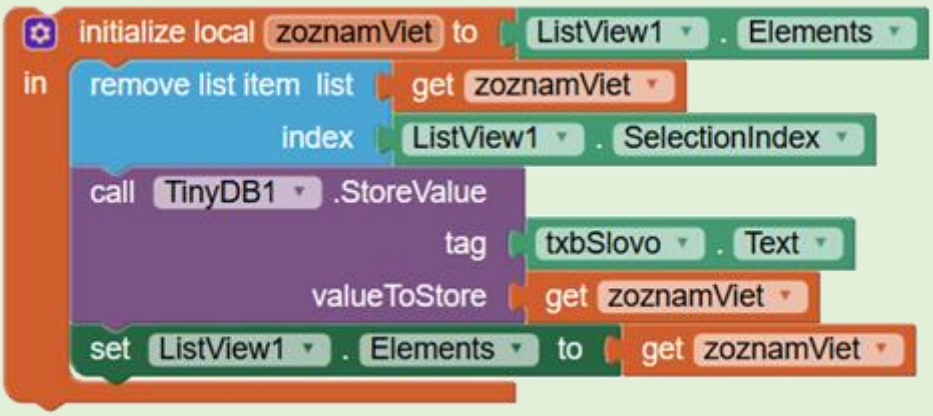
S vlastnosťou `ListView.Elements` pracujeme ako so zoznamom, môžeme overiť či je prázdny, pridávať a odoberať prvky. Po každej operácii však nesmieme zabudnúť na synchronizáciu s lokálnou databázou `TinyDB`:



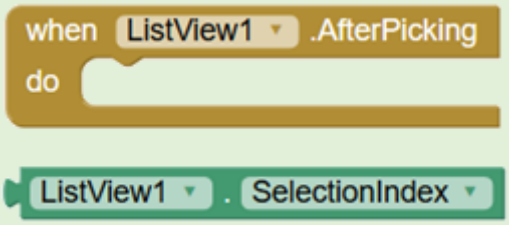


Pri získavaní zoznamu viet pre kľúčové slovo zadané v textovom poli sa nám zíde lokálna premenná:

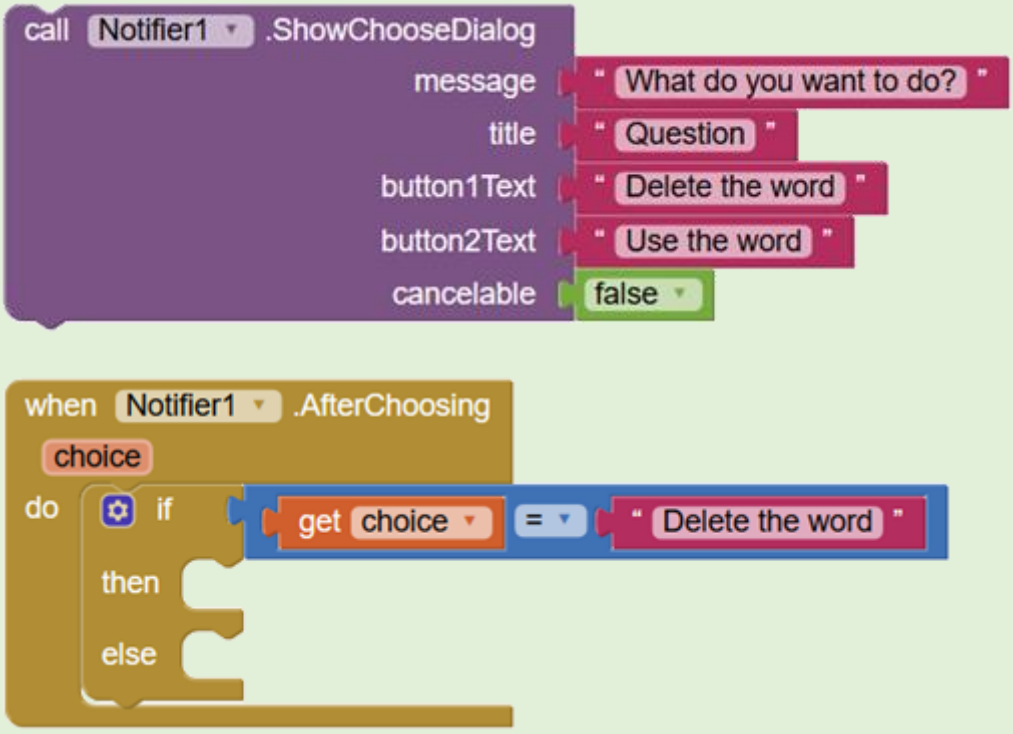
Podobne aj pri ukladaní aktualizovaného zoznamu viet do databázy (napr. po odstránení zvolenej vety) :



Ak chceme reagovať na výber slova zo zoznamu zobrazeného v komponente `ListView`, musíme naprogramovať reakciu na udalosť `ListView.AfterPicking`:



Ak chceme, aby používateľ potvrdil alebo zvolil zrealizovanie niektorej operácie (napr. vymazanie slova z databázy), zobrazíme dialógové okno a naprogramujeme reakciu na udalosť `Notifier.AfterChoosing`:



Pomoc k riešeniu úlohy

Úloha 4 je časovo náročná, pretože vyžaduje, aby si žiak dobre premyslel, akým spôsobom bude údaje z databázy zobrazovať a čo a ako umožní používateľovi s údajmi robiť.

Jedno z možných riešení uvádzame v priloženom projekte **pmz_prekladac_ver2_R.aia**. V tomto riešení sme použili jeden komponent `ListView`, v ktorom zobrazujeme slová uložené v databáze ako kľúče a po výbere konkrétneho slova aj vety, ktoré mu zodpovedajú. Pomocnú informáciu o aktuálnom zobrazení uchovávame v globálnej premennej `coVidim`.

Ako vylepšiť či rozšíriť našu aplikáciu?

Základnú verziu aplikácie môžeme vylepšiť rôznym spôsobom, prispôsobiť si ju vlastným potrebám, napr.:

- pridať podporu pre viaceré jazyky,
- porovnávať vlastnú výslovnosť v nahrávke (použiť komponent *SoundRecorder*) s výstupom syntetizátora reči,
- naplniť databázu konverzačnými frázami, umožniť vyhľadávanie a hlasový výstup,
- exportovať obsah databázy do textového súboru,
- využiť obsah databázy ako zdroj dát pre minihru zameranú na precvičovanie slovnej zásoby alebo gramatiky,
- umožniť ovládanie celej aplikácie hlasom,
- upraviť GUI aplikácie, pridať ďalšie obrazovky atď.

Odporúčaný priebeh výučby

Činnosť učiteľa	Činnosť žiaka	Poznámky
1. hodina – Úvod, Úloha 1		
Uvedenie témy projektu: učiteľ moderuje diskusiu o učení sa cudzích jazykov a jazykových aplikáciách. Úloha 1: Učiteľ sformuluje základné požiadavky na aplikáciu. Učiteľ navrhne trasenie tabletom ako jednu z možností ako v aplikácii prepínať smer prekladu.	Žiaci reagujú na jeho otázky, diskutujú, hovoria o vlastných skúsenostiach. Žiaci navrhujú vzhľad úvodnej obrazovky. Dizajn aplikácie najskôr načrtnú na papier, pomenujú jednotlivé komponenty. Potom svoj návrh zrealizujú v App Inventore. Žiaci vložia do aplikácie komponent *AccelerometerSensor* a vyriešia problém s prispôsobením používateľského rozhrania.	Námety na otázky pre žiakov sme uviedli v úvode kapitoly. V diskusii by mala zaznieť aj informácia o strojovom preklade, dôvodoch jeho používania, jeho obmedzeniach. Učiteľ môže dať žiakom k dispozícii súbory s obrázkami vlajok a ikony pre tlačidlá (v závislosti od počtu hodín, ktorý má v pláne projektu venovať). Žiaci môžu navrhnúť aj iný spôsob, nemusia použiť senzor zrýchlenia.
2. hodina – Úloha 2 a Úloha 3		
Úloha 2: Učiteľ vyzve žiakov, aby vymenovali komponenty, ktoré sa používajú na syntézu reči a hlasové rozpoznávanie. Po spoločnom opakovaní nechá žiakov pracovať samostatne. Úloha 3: Učiteľ žiakom predstaví webovú službu *Yandex* a nový komponent, pomocou ktorého budú získavať preklad.	Žiaci programujú správanie aplikácie. Komunikujú spolu o riešeniach. Aplikáciu priebežne testujú. Žiaci dokončia aplikáciu, použijú komponent *YandexTranslate* v praxi.	Žiaci by mali základné riešenie dokončiť do 20 minút. Učiteľ by mal žiakom vo svojom výklade priblížiť aj to, ako webové služby fungujú a prečo sú užitočné. Žiaci tiež môžu využiť webovú stránku služby Yandex a experimentovať s prekladom medzi rôznymi jazykmi (https://translate.yandex.com/). K dispozícii je aj mobilná aplikácia na využívanie tejto služby.

3. hodina – Úloha 4, rozširujúce námety		
Úloha 4: Učiteľ motivuje žiakov, aby rozšírili svoju aplikáciu o ďalšie funkcionality, oceňuje tvorivé nápady žiakov. Povinnou súčasťou riešenia musí byť použitie lokálnej databázy. Učiteľ usmerní žiakov, na čo majú dať pri práci s databázou pozor, pripomenie im, ako sa tvoria aplikácie s viacerými obrazovkami.	Žiaci vyvíjajú aplikáciu už ako svoj individuálny projekt, mali preto navrhnúť a zrealizovať rôzne rozšírenia. Hotovú aplikáciu žiaci zverejnia v *Galérii* projektov, v popise aplikácie uvedú jej hlavné funkcionality.	Evidovanie príkladov viet pre kľúčové slová môžeme chápať ako jeden z námetov. Žiaci môžu uchovávať históriu prekladaných slov, slová či frázy triediť do tematických kategórií a pod. Lokálnu databázu by mohli využiť aj „pasívne“ ako vopred naplnený zdroj dát pre minihru alebo do nej uložiť frázy potrebné v rôznych komunikačných situáciách. V závere kapitoly sme uviedli niekoľko námetov na ďalšie vylepšenia a rozšírenia aplikácie. Žiakom môže učiteľ poskytnúť prehľad blokov, ktoré by mohli vo svojom riešení potrebovať (podobne ako v časti Pomôcky za Úlohou 4).
4. hodina – Úloha 4, rozširujúce námety		
Učiteľ podporuje žiakov pri práci, je v roli konzultanta. Učiteľ žiakom pripomenie, akým spôsobom bude prebiehať prezentácia a hodnotenie projektu.	Žiaci pokračujú v práci na svojej aplikácii.	Žiaci budú projekt pravdepodobne dokončovať doma. Zverejnenie výsledku práce podporuje budovanie identity vývojára. Zároveň poskytneme ostatným žiakom možnosť preskúmať riešenie, ktoré ich zaujalo.
5. hodina – Prezentácia a hodnotenie projektov		
Učiteľ hodnotí projekty žiakov. Po prezentácii žiaka položí žiakovi doplňujúcu otázku týkajúcu sa niektorého z riešených podproblémov. Učiteľ dá priestor na otázky aj ostatným žiakom.	Každý žiak ukáže stránku svojej aplikácie v *Galérii* a predstaví jej možnosti. Odpovedá na otázku učiteľa, spolužiakov. V závere žiaci hlasujú o najlepšej aplikácii, napr. s využitím systému Socrative.	Hodina venovaná vyhodnoteniu projektu nemusí nasledovať hneď po predchádzajúcich projektových hodinách. V záujme udržania motivácia však neodporúčame dlhšiu ako dvojtýždňovú prestávku.