

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. Niektorí sa učia jazyky rád a ide mu to ľahko, iní potrebujú 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 tréning 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.

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 tréňovaní jazykových zručností (rozširovanie slovnej zásoby, zdokonaľovanie výslovnosti, plynulosť a gramatická správnosť vo vyjadrovaní).

Ako budeme postupovať pri tvorbe aplikácie?

V našom projekte budeme musieť postupne vyriešiť rôzne podproblémy:

- navrhne 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 používateľovi 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
- navrhne 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*.

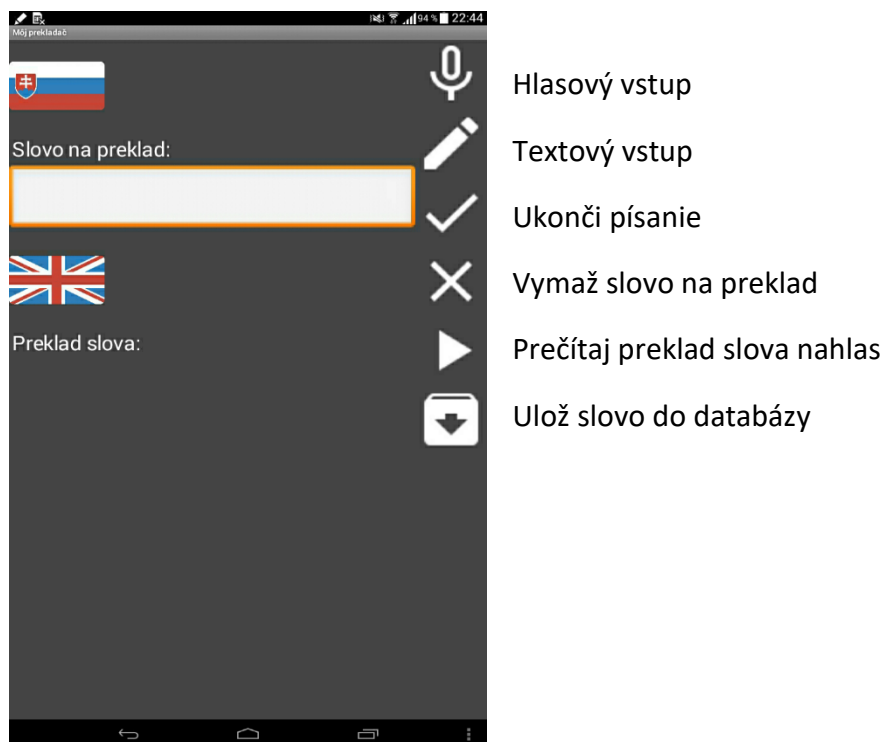
Na preklad slova využijeme webovú službu, s ktorou budeme komunikovať pomocou komponentu *YandexTranslate* z kategórie *Media*.

Úloha 1

Navrhnete 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 prispôbiť 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

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.

Úloha 3

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

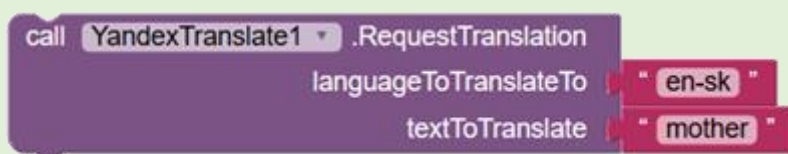
Vysvetlíme si

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 služby 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. Dvojnakové kódy podporovaných jazykov nájdeme na stránke služby Yandex.

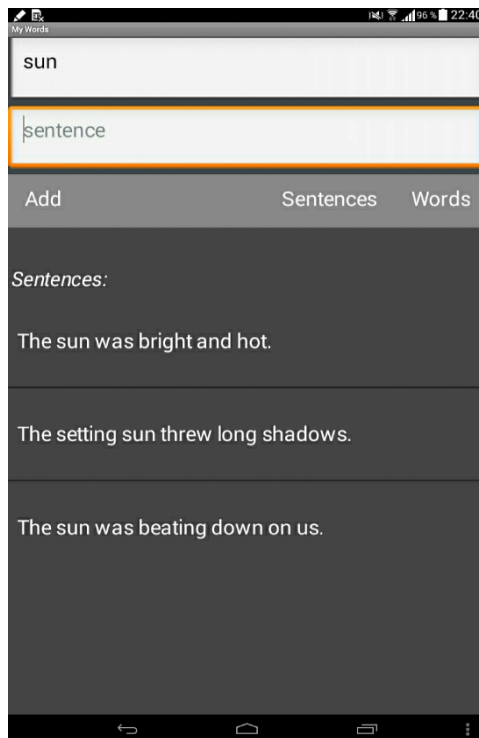
Preklad sa v aplikácii realizuje asynchrónne, na pozadí. Jeho ukončenie vygeneruje udalosť `YandexTranslate1.GotTranslation`, na ktorú môžeme zareagovať. Výsledok prekladu získame z parametra `translation` (v našom príklade pôjde o reťazec `"matka"`).



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.

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.



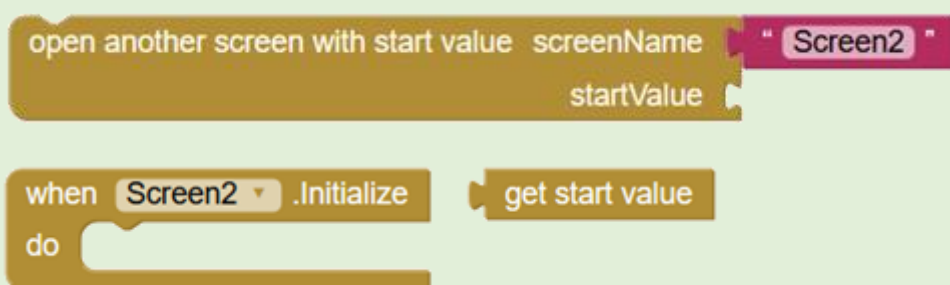
Obr. 4.3.2 Zobrazenie príkladov viet pre zadané kľúčové slovo

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`:

```
set ListView1 . Elements to create empty list

if is list empty? list ListView1 . Elements
then
else

set ListView1 . Elements to call TinyDB1 . GetTags
```

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

```
initialize local zoznamViet to call TinyDB1 . GetValue
tag txbSlovo . Text
valueIfTagNotThere ""
in
```

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

```
initialize local zoznamViet to ListView1 . Elements
in
remove list item list get zoznamViet
index ListView1 . SelectionIndex
call TinyDB1 . StoreValue
tag txbSlovo . Text
valueToStore get zoznamViet
set ListView1 . Elements to get zoznamViet
```

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


```
when ListView1 .AfterPicking
do
```

```
ListView1 . SelectionIndex
```

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`:

```
call Notifier1 .ShowChooseDialog
  message "What do you want to do? "
  title "Question "
  button1Text "Delete the word "
  button2Text "Use the word "
  cancelable false
```

```
when Notifier1 .AfterChoosing
  choice
do
  if
  then
  else
```

```
get choice = "Delete the word "
```

Ako vylepšiť či rozšíriť našu aplikáciu?

Základnú verziu aplikácie môžeme vylepšiť rôznym spôsobom, prispôbiť 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.