

4.4. Spoločenská hra pre tablet

Spoločenskou hrou rozumieme hru pre dvoch a viac hráčov. Hráči sú pri hre fyzicky prítomní na jednom mieste, súperia proti sebe individuálne alebo v tímoch (hrajú sa spolu, relaxujú, zabávajú sa). Existujú rôzne typy spoločenských hier: doskové, kartové, slovné, papierové, konštrukčné. Pri niektorých hrách potrebujeme viacero pomôcok (hrací plán, hracie kocky, figúrky), pri iných menej (hracie karty, papier a pero) alebo dokonca žiadne. Pri spoločenskej hre sa nám môže zísť aj tablet (napr. na cestách). Dokáže ľahko zastúpiť hraciu kocku, kresliace plátno, zápisník, stopky a pod. Niektoré spoločenské hry môžu byť však na použitie tabletu a jeho špecifických vlastnostiach aj založené. V takomto prípade hru riadi mobilná aplikácia: automaticky zobrazuje úlohy a vyhodnocuje úspešnosť. Zmysluplná interakcia s tabletom prispieva k dynamike a atraktivnosti hry. Zaujímavým spostením býva tiež multimediálny výstup (zvukové a grafické efekty).

Otázky na zamyslenie

Ktorú spoločenskú hru ste sa hrávali/sa hrávate najčastejšie doma/v škole/na výletoch?

Uveďte príklad doskovej, kartovej, slovnej, papierovej a konštrukčnej hry.

Ktoré z vymenovaných spoločenských hier má zmysel hrať s využitím tabletu?

Použili ste už pri spoločenskej hre tablet?

Kľúčové slová

spoločenská hra, ovládanie hry pomocou akcelerometra, webová databáza TinyWebDB, tímový projekt

Čo sa naučíme a čo si precvičíme

- vytvoríme spoločenskú hru pre tablet,
- údaje pre hru získame zo vzdialenej databázy, s ktorou budeme komunikovať pomocou komponentu *TinyWebDB*,
- priebeh hry budeme riadiť nakláňaním tabletu vpred a vzad (pomôže nám akcelerometer).
- pri tvorbe aplikácie budeme pracovať v tímoch a naučíme sa, ako je možné tímový projekt realizovať v App Inventore,
- hotovú aplikáciu otestujeme v praxi (zahráme sa).

Príprava na výučbu

Prerekvizity: senzor zrýchlenia (etuda 2.1), komponent *Notifier* (etuda 2.2), *Sound* (etuda 2.3), *TinyDB* (etuda 2.4), *Screen* (etuda 2.5), *Spinner* (etuda 2.6), zoznamy (etuda 2.8).

Súbory s obrázkami, ktoré sú použité vo vzorových riešeniach, sa nachádzajú v archíve **media.zip**.

Prílohou kapitoly sú riešenia Úloh 1 až 3 implementované v samostatných projektoch (

- **pmz_4_4_uloha1_Screen1_R.aia**, **pmz_4_4_uloha1_Screen2_R.aia**,

- **pmz_4_4_uloha1_R.aia**
- **pmz_4_4_uloha2_R.aia, pmz_4_4_uloha3_R.aia.**

Prikladáme tiež vzorové riešenie tímového projektu (Úloha 4 a Úloha 5):

V projekte **pmz_karticky_Screen1_R.aia** je implementované riešenie problému komunikácie s webovou databázou. Projekt **pmz_4_4_karticky_Screen2_R.aia** obsahuje riešenie problému riadenia hry. Projekt **pmz_4_4_karticky_R.aia** predstavuje hotovú aplikáciu – spoločenskú hru **Kartičky**.

Žiaci si vyskúšajú prácu tímového vývojára. Budú pracovať s nástrojom, ktorý umožňuje zlúčiť dva alebo viaceré projekty prostredia App Inventor do jedného celku. Nástroj **App Inventor Merger**, ktorý v tejto kapitole použijeme, je k dispozícii zdarma na stiahnutie na stránke <http://appinventor.mit.edu/explore/resources/ai2-project-merger.html>. Uvádzame ho aj v prílohe kapitoly.

AI2MergerApp.jar je desktopová aplikácia napísaná v Jave, je preto potrebné, aby bola pred jej spustením v počítači nainštalovaná podpora pre Javu (*JRE, Java Runtime Environment*). Príslušný inštalačný súbor nájdete napr. tu:

<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

Akú zaujímavú aplikáciu môžeme vytvoriť?

Naprogramujeme tabletovú verziu spoločenskej hry známej pod názvami *Šarády*, *Aktivitty* alebo *Kartičky*. Ide o *slovnú hru* pre 2 hráčov. Jeden hráč má k dispozícii kartičky so slovami a opisuje ich druhému hráčovi, ktorý má slovo uhádnuť. Hra má časový limit, cieľom oboch hráčov je, aby hádajúci hráč uhádol čo najviac slov. Hra je zaujímavejšia, keď proti sebe súperia viaceré dvojice hráčov. Pravidlá pre opisovanie slov môžu byť rôzne a ovplyvňujú obtiažnosť a zábavnosť hry: hráč nesmie použiť v slovnom opise základ slova, musí hovoriť v cudzom jazyku, musí pri opisovaní spievať, veršovať, imitovať zvuky a pod. Použitie tabletu zefektívni organizačnú stránku hry: nebudeme musieť vopred pripravovať žiadne papierové kartičky ani s nimi pri hre manipulovať, nemusíme merať čas ani počítať skóre. Na výber bude viacero kategórií slov, databázu slov bude možné upravovať a rozširovať.

Ako budeme postupovať pri tvorbe aplikácie?

V našom projekte sa dajú ľahko identifikovať dva problémy, ktoré je možné vyriešiť nezávisle. Budeme preto pracovať v dvojčlenných tímoch:

Prvý člen tímu bude zodpovedný za to, aby si aplikácia po každom spustení z webovej databázy stiahla aktuálne dostupné kategórie slov a z kategórie, ktorú si hráč vyberie, uložila všetky slová na hádanie do lokálnej databázy.

Pripravené dáta použije **druhý člen tímu** vo svojej časti riešenia. Jeho úlohou bude zabezpečiť riadenie priebehu hry, jej riadne ukončenie a vyhodnotenie. Hra bude prebiehať takto: Hráč, ktorý má hádať, drží tablet v rukách tak, aby naň videl hádajúci hráč (t. j. zvislo, displejom smerom k spoluhráčovi). Ďalšie slovo sa na tablete zobrazí po uhádnutí slova (háďajúci hráč nakloní tablet na chvíľu vpred) alebo vtedy, keď sa háďajúci hráč vzdá a rozhodne sa radšej pre

háďanie ďalšieho slova (vtedy tablet nakloní vzad). Po skončení časového limitu sa zobrazí počet úspešne uhádnutých slov.

Prvý člen tímu bude pracovať na hlavnej obrazovke aplikácie (komponent *Screen1*). Druhý člen tímu vytvorí druhú obrazovku aplikácie (komponent *Screen2*). Z prvej obrazovky sa bude otvárať druhá obrazovka. Z druhej obrazovky sa bude dať vrátiť na prvú (hlavnú) obrazovku. Čiastkové projekty členov tímu sa v záverečnej fáze spoja do jedného spoločného projektu pomocou nástroja **App Inventor Merger**.

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

- vizuálne komponenty a správcov rozvrhnutia,
- *AccelerometerSensor*,
- *Notifier*,
- *Sound*,
- *Spinner*,
- lokálnu databázu *TinyDB*.

Dáta hry chceme uložiť vo vzdialenej databáze. Mobilná aplikácia k nim bude pristupovať prostredníctvom webovej služby, s ktorou budeme v App Inventore komunikovať pomocou komponentu *TinyWebDB* z kategórie *Storage*.

Najprv si vyskúšame, ako budeme zlučovať viaceré projekty do jedného:

Úloha 1

Vytvorte aplikáciu s dvomi obrazovkami. Na prvej obrazovke nech je obrázok psa, na druhej obrazovke nech je obrázok mačky. Keď na obrazovke so psom stlačíme tlačidlo, otvorí sa druhá obrazovka, na ktorej sa objaví alebo ozve správa pre mačku. Po návrate na hlavnú obrazovku uvidíme alebo si vypočujeme, čo na oplátku mačka odkazuje psovi. Správy môžete generovať aj náhodne. Pracujte v dvojčlenom tíme. Každý člen tímu nech je zodpovedný za implementáciu jednej z obrazoviek. Na záver svoje projekty zlúčte do jedného a aplikáciu dokončite a otestujte.

Vysvetlíme si

V praxi programátori väčšinou pracujú v tímoch. Aj v App Inventore je možné, aby jednotlivé obrazovky aplikácie vyvíjali nezávisle dvaja alebo viacerí programátori. Jednotlivé projekty sa v záverečnej fáze dajú pomocou jednoduchého desktopového nástroja zlúčiť do finálneho produktu.

Členovia tímu sa musia na začiatku **dohodnúť na konvenciách**, ktoré budú dodržiavať pri pomenovaní obrazoviek, zdieľaných komponentov (takým je napr. lokálna databáza *TinyDB*) a súborov médií, aby sa mohli v zdrojovom kóde svojich projektov na ne odvolávať. **Hlavná obrazovka aplikácie** sa nedá v App Inventore premenovať, preto sa **vždy musí volať *Screen1***.

Uvažujme dvoch vývojárov, ktorých úlohou je naprogramovať aplikáciu s dvoma obrazovkami. Prvý z nich bude pracovať vo svojom projekte na hlavnej obrazovke aplikácie s názvom

Screen1. Druhý z nich si vo vlastnom projekte **pridá novú obrazovku** a pomenuje ju napr. *Screen2*. Navrhne jej dizajn a naprogramuje jej funkcionality. Svoju hlavnú obrazovku s názvom *Screen1* môže ignorovať, nanajvýš na ňu umiestni pomocné tlačidlo na otvorenie druhej obrazovky, aby sa k nej dostal pri testovaní svojej časti aplikácie. Táto pomocná obrazovka sa pri zlučovaní projektov vynechá, nie je podstatné, čo obsahuje.

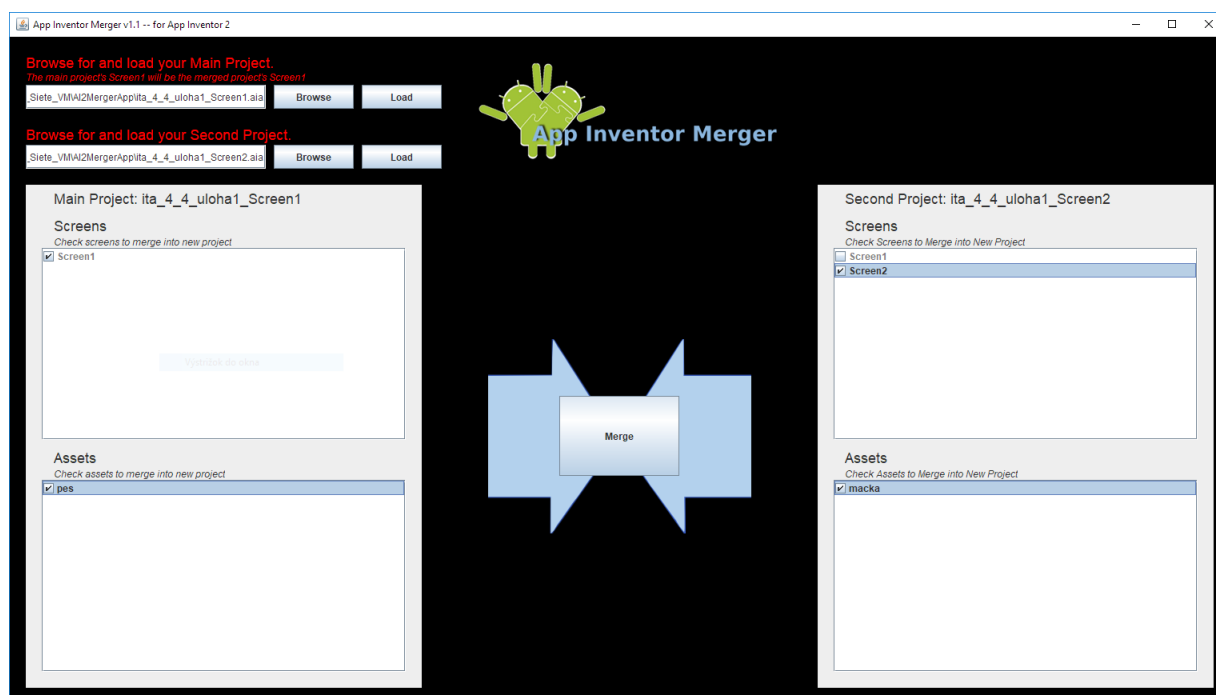
Po dokončení čiastkových riešení každý z vývojárov exportuje svoj projekt z cloudového úložiska do počítača. Po spustení nástroja **App Inventor Merger** (súbor aplikácie má názov *AI2MergerApp.jar*) sa zobrazí okno (Obr. 4.4.1), v ktorom vyberáme projekty na zlúčenie. Po načítaní projektov (prvý projekt sa pokladá za hlavný projekt s obrazovkou *Screen1*) uvidíme zoznam komponentov a mediálnych súborov, ktoré ich tvoria. V zoznamoch pre jednotlivé projekty zvolíme, čo má byť súčasťou cieľového projektu po zlúčení. Niekedy totiž používame rovnaké obrázky a zvuky na viacerých obrazovkách. Alebo zatiaľ nechceme do výsledného projektu zahrnúť všetky obrazovky, na ktorých v projekte pracujeme.

Zlúčenie zrealizujeme kliknutím na tlačidlo *Merge (Zlúčiť)*. Vznikne nový projektový súbor *.aia*, ktorý v dialógovom okne vhodne pomenujeme a uložíme na disk. Následne ho importujeme do prostredia App Inventor (je jedno, ktorý z vývojárov), kde s ním môžeme ďalej pracovať.

Dokumentáciu k nástroju **App Inventor Merger** a súbor *AI2MergerApp.jar* na stiahnutie sa nachádza tu: <http://appinventor.mit.edu/explore/resources/ai2-project-merger.html>.

AI2MergerApp.jar je desktopová aplikácia napísaná v Jave, je preto potrebné, aby bola pred jej spustením v počítači nainštalovaná podpora pre Javu (*JRE, Java Runtime Environment*). Príslušný inštalčný súbor je na stiahnutie napr. tu:

<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

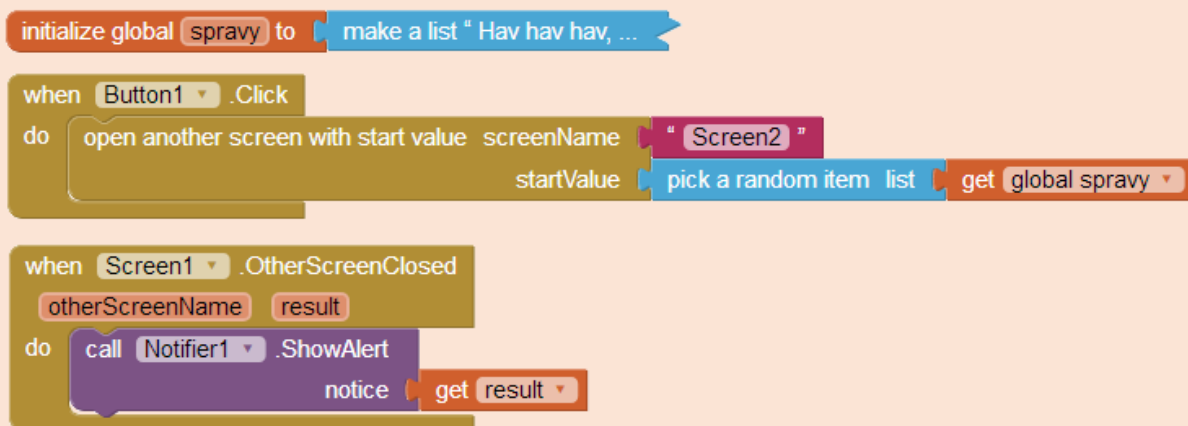


Obr. 4.4.1 Zlučovanie projektov v okne nástroja *App Inventor Merger*

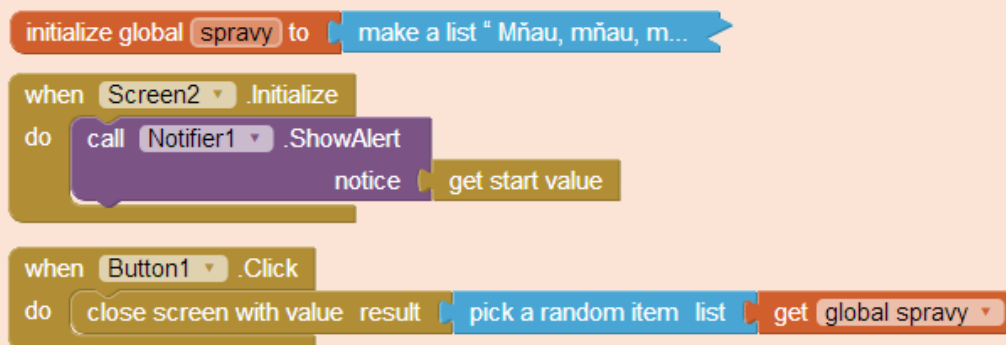
Pomoc k riešeniu úlohy

Zadanie úlohy je jednoduché. Žiaci si pri tvorbe aplikácie zopakujú, ako sa tvorí aplikácia s viacerými obrazovkami a zoznámia sa s novým nástrojom na zlučovanie projektov. Na zobrazenie obrázku psa a mačky môžeme použiť komponent *Image* s kategórie *User Interface*. Správy, ktoré si mačka a pes z obrazoviek budú posilať, môžeme zobrazovať napr. pomocou komponentu *Notifier*:

V projekte **pmz_4_4_uloha1_Screen1_R.aia** stačí naprogramovať otvorenie obrazovky s názvom *Screen2* (náhodne zvolenú správu pre mačku jej odovzdáme v parametri *startValue*) a reakciu na zatvorenie druhej obrazovky (tá v tomto projekte samozrejme zatiaľ neexistuje, pracuje na nej druhý člen tímu vo svojom projekte):



V projekte **pmz_4_4_uloha1_Screen2_R.aia** pri otvorení (inicializácii) obrazovky zobrazíme správu od psa. Pri zatváraní obrazovky zabezpečíme odovzdanie náhodne zvolenej správy od mačky obrazovke, ktorá obrazovku s mačkou otvorila (využijeme nato parameter *result*):



Obr. 4.4.1 ukazuje okno aplikácie **App Inventor Merger** pred zlúčením dvoch vyššie spomenutých projektov do jedného (pomenovali sme ho **pmz_4_4_uloha1_R.aia**). V tomto projekte už nebude potrebné robiť žiadne úpravy, aplikáciu môžeme otestovať na tablete.

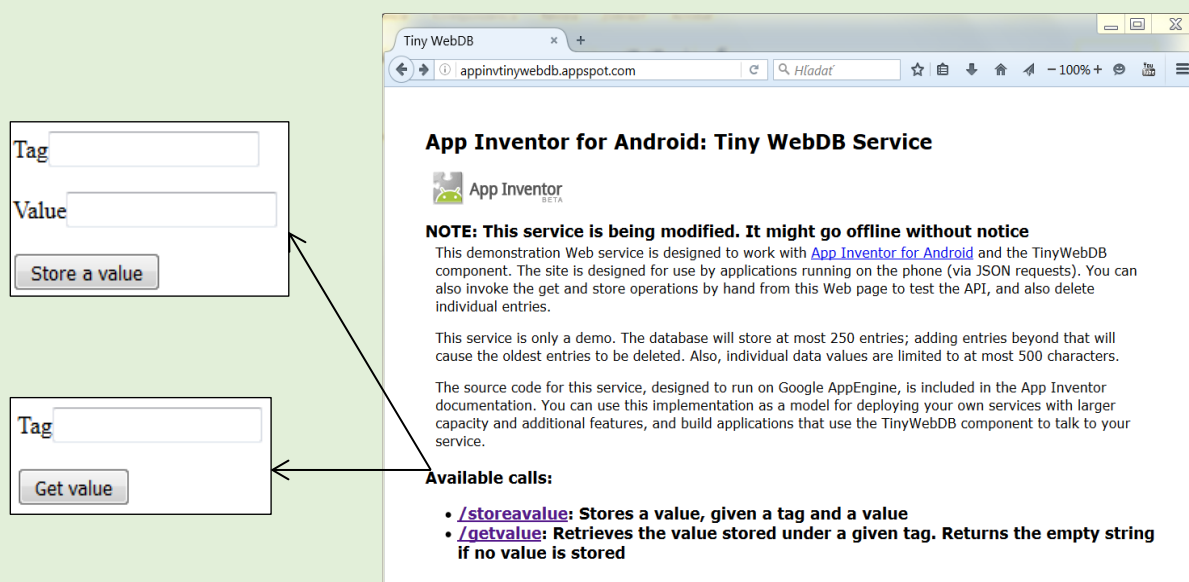
Úloha 2

Vyberte si kategóriu slov na hádanie (napr. Povolania, Hudobné skupiny, Rozprávkové bytosti a pod.). Zapište si na papier zoznam slov, ktoré chcete do tejto kategórie zaradiť. Uložte tieto údaje do vzdialenej databázy s využitím databázového komponentu *TinyWebDB*. Potom vytvorte aplikáciu, v ktorej údaje uložené v databáze načítate a zobrazíte.

Vysvetlíme si

S komponentom *TinyDB*, ktorý reprezentuje lokálne databázové úložisko aplikácie, sme už pracovali viackrát. Ak chceme, aby bola databáza umiestnená na serveri a mohli ju tak zdieľať viacerí používatelia (presnejšie inštancie rovnakej aplikácie nainštalované v mobilných zariadeniach rôznych používateľov), môžeme použiť komponent *TinyWebDB*.

Po vložení komponentu *TinyWebDB* do aplikácie zistíme, že má vo vlastnosti `ServiceURL` prednastavenú adresu `http://appinvtinywebdb.appspot.com`. Ide o demo webovej služby s obmedzeniami na počet a veľkosť záznamov (Obr. 4.4.2). Pre naše účely nám zatiaľ takéto riešenie postačí. Musíme si však uvedomiť, že rovnakú databázu, s ktorou budeme pracovať, používajú aj iní vývojári. Nie je preto vylúčené, že naše údaje niekto získa alebo prepíše. Stačí, že sa náhodou rozhodne použiť pri ukladaní hodnoty rovnaký *tag* (kľúč). Aby sa minimalizovalo toto riziko, zvyknú sa ako tagy voliť reťazce v tvare obrátenej doménovej adresy, napr.: `"sk.mojaskola.mojemenokarticky.povolania"`.



Obr. 4.4.2 Webová služba *Tiny WebDB* s formulármi pre uloženie a získanie údajov

Návod na vytvorenie vlastnej webovej služby typu *Tiny WebDB* (bez spomínaných obmedzení demoverzie) nájdeme na: <http://ai2.appinventor.mit.edu/reference/other/tinywebdb.html>).

Údaje sa do databázy dajú ukladať aj manuálne, s využitím formulára vo webovom rozhraní služby (Obr. 4.4.2). V našom prípade zvolíme vhodný *Tag* a do políčka *Value* zapíšeme zoznam slov oddelených čiarkami uzavretý v hranatých zátvorkách. Ak chceme overiť uloženie hodnoty v databáze, môžeme to urobiť zadaním tagu do vstupného poľa druhého formulára. Ak v databáze so zadaným tagom nie je asociovaná žiadna hodnota, vráti sa ako výsledok dopytu prázdny reťazec.

V App Inventore máme k dispozícii dve metódy a tri udalostné bloky súvisiace s komponentom *TinyWebDB*:



Ak poznáme tag, vieme z databázy získať hodnotu, ktorá je s ním asociovaná. Do databázy môžeme samozrejme ukladať ďalšie záznamy. V aplikácii sa dá zareagovať na úspešné získanie aj na uloženie hodnoty. V prípade, že komunikácia s webovou službou zlyhala, vznikne udalosť `TinyWebDB.WebServiceError`.

Pomoc k riešeniu úlohy

Na uloženie testovacích záznamov do databázy môžu žiaci využiť webové rozhranie služby *Tiny WebDB* (<http://appinvtinywebdb.appspot.com/>).

Po uložení záznamu, napr.:

Tag

Value

Store a value

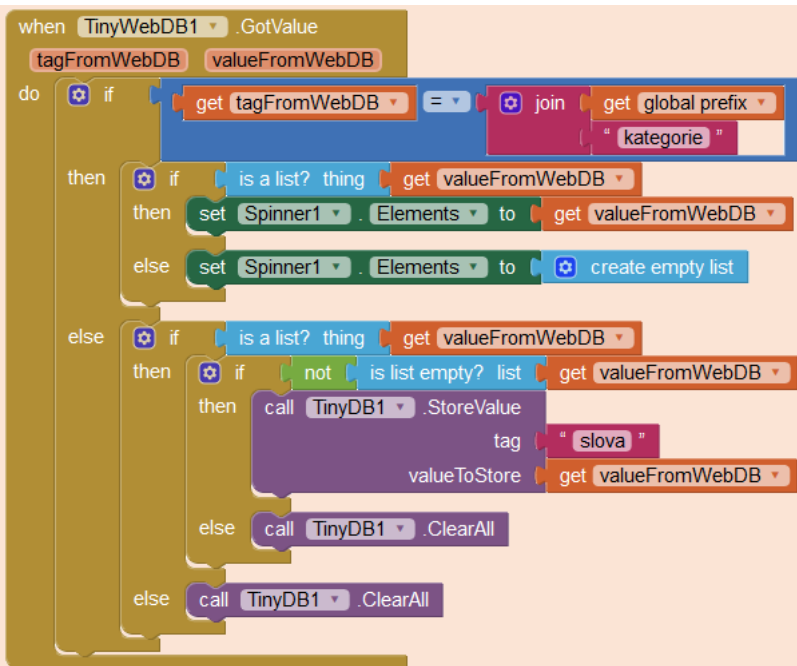
uvidíme správu servera o výsledku operácie, napr.

The server will send this to the component:

```
["STORED", "sk.mojaskola.mojemeno.karticky.povolania", "[u\u010dite\u013e, vedec, mur\u00e1r, lek\u00e1r]"]
```

[Return to TinyWebDB Main Page](#)

V testovacej aplikácii (**pmz_4_4_uloha2_R.aia**) pripravíme jednoduchý formulár so vstupným poľom na zadanie tagu a tlačidlom na spustenie operácie načítania. Získané údaje (zoznam slov) zobrazíme v komponente *ListView* alebo *Spinner*. Musíme rozlišovať medzi dopytom na zoznam kategórií a zoznamom slov pre zvolenú kategóriu. Hodnota asociovaná s tagom má byť v oboch prípadoch zoznam, ktorý nemá byť prázdny:



Na túto úlohu nadväzuje **Úloha 4**.

Úloha 3

Na detegovanie pohybu tabletu pri nakláňaní tabletu zo zvislej polohy vpred alebo vzad, resp. z okrajových polôh naspäť do zvislej polohy využijeme v našej hre senzor zrýchlenia, t. j. komponent *AccelerometerSensor*. Vytvorte jednoduchú aplikáciu na vypisovanie aktuálnych hodnôt, ktoré akcelerometer meria. **Uvažujte o tom, ako rozpoznať, aký pohyb používateľ s tabletom práve urobil.**

Poznámka

V aplikácii najprv nastavte napevno orientáciu obrazovky na *Landscape*. Komponentu *AccelerometerSensor* zmeňte v režime *Design* vlastnosť *Legacy Mode* na `true`. Vypnete tým dodatočnú úpravu hodnôt poskytovaných aplikácii systémom Android, ktorú autori App Inventoru robia preto, aby zabezpečili rovnaké fungovanie aplikácie pre tablety aj smartfóny. Telefóny mávajú totiž východiskovú orientáciu nastavenú na *Portrait*.

Zrealizujte niekoľko experimentov s nakláňaním tabletu. Sledujte ako sa menia hodnoty parametrov `xAccel`, `yAccel`, `zAccel`, ku ktorým máme prístup v udalostnom bloku `when AccelerometerSensor1.AccelerationChanged`:

Experimenty s nakláňaním tabletu				
tablet je zvislo, držíme ho displejom otočeným od seba	zo zvislej polohy nakláňame tablet vpred až do vodorovnej polohy, v ktorej je tablet obrátený displejom nadol	z vodorovnej polohy s displejom natočeným nadol vraciame tablet späť do zvislej polohy	zo zvislej polohy nakláňame tablet vzad až do vodorovnej polohy, v ktorej je tablet obrátený displejom nahor	z vodorovnej polohy s displejom otočeným nahor vraciame tablet späť do zvislej polohy

Pomoc k riešeniu úlohy

Pri nakláňaní tabletu zo zvislej polohy smerom dozadu alebo dopredu nás bude zaujímať hodnota zrýchlenia v smere osi z . V hraničných polohách má parameter `zAccel` hodnotu väčšiu ako 9, resp. menšiu ako -9.

Umiestnenie tabletu do zvislej polohy je možné detegovať na základe hodnoty zrýchlenia v smere osi y . V tejto polohe má parameter `yAccel` hodnotu väčšiu ako 9.

Dôležité je tiež uvedomiť si, že po rozpoznaní náklonu vpred alebo vzad a pri návrate do zvislej polohy je potrebné si túto skutočnosť ihneď zapamätať (v logickej premennej) a to z dvoch dôvodov:

1. aby sa zabránilo opakovanému detegovaniu náklonu vpred/vzad predtým, ako používateľ vráti tablet znovu do zvislej polohy,
2. aby bolo možné správne rozlíšiť smer pohybu zo zvislej polohy vpred/vzad a naopak z hraničných polôh naspäť do zvislej polohy.

Aplikáciu použiteľnú na realizovanie experimentov (sledovanie nameraných hodnôt) žiaci vytvoria veľmi rýchlo (**pmz_4_4_uloha3_R.aia**). Na túto úlohu nadväzuje **Úloha 5**.

Naprogramujeme aplikáciu **Kartičky** z úvodu kapitoly:

Úloha 4 (pre prvého člena tímu)

Navrhnete dizajn hlavnej obrazovky. Umožnite používateľovi vybrať zo zoznamu kategóriu slov na hádanie. Pre zvolenú kategóriu načítajte zoznam slov z webovej databázy. Uložte ho do lokálnej databázy aplikácie, ktorú budú zdieľať obe obrazovky. Pripravte tiež tlačidlo na spustenie hry. Dbajte však na to, aby bolo prístupné len v prípade, že slová na hádanie sú k dispozícii.

Úloha 5 (pre druhého člena tímu)

Naprogramujte automatické zobrazovanie slov a počítanie správnych odpovedí. Hru ovládajte nakláňaním tabletu vpred a vzad. Po skončení časového limitu alebo uhádnutí všetkých slov hru ukončíte a oznámte výsledok. Po návrate na hlavnú obrazovku bude možné zvoliť inú kategóriu a spustiť ďalšie kolo hry.

Poznámka

V okamihu otvorenia druhej obrazovky už budú k dispozícii slová na hádanie. Pred spustením časomierky ale musíme mať istotu, že hráč drží tablet zvislo pred sebou tak, aby naň videl jeho spoluhráč. Až potom má zmysel hru odštartovať a zobrazíť prvé slovo.

Pomoc k riešeniu úloh

Členovia tímu by sa mali vopred dohodnúť na názve druhej obrazovky (prvá sa musí volať *Screen1*) a názve komponentu *TinyDB*, ku ktorému budú v cieľovom projekte pristupovať obe obrazovky. Tiež je vhodné dohodnúť sa na grafickom vzhľade aplikácie (pripraviť si napr. obrázok, ktorý bude použitý na pozadí na oboch obrazovkách a pod.).

Autor druhej obrazovky bude potrebovať pri testovaní dáta, ktoré zatiaľ nemá, naplnenie lokálnej databázy má totiž na starosti spolužiak. Riešením je ručné naplnenie globálneho zoznamu slov a dočasné vypnutie príkazu, ktorým sa neskôr tieto dáta získajú z lokálnej databázy.

Riešenia problémov z Úlohy 4 a 5 uvádzame v projektoch **pmz_4_4_karticky_Screen1_R.aia** a **pmz_4_4_karticky_Screen2_R.aia**.

Výsledný projekt s hotovou aplikáciou má názov **pmz_4_4_karticky_R.aia**. Vo vzorovom riešení po správnom umiestnení tabletu hru odštartujeme pomocou tlačidla (môže to urobiť aj hráč, ktorý sa na tablet díva).

Úloha 6

Projekty členov tímu zlúčte do jedného celku, aplikáciu dokončite a otestujte.

V diskusii so spolužiakmi navrhnete pravidlá turnaja v hre **Kartický**. Každý tím použije v súťaži vlastnú aplikáciu.

Poznámka

Aplikácie všetkých tímov by mali údaje zo vzdialenej databázy načítavať s použitím rovnakých tagov. Prefix identifikujúci projekt, napr. `"sk.mojaskola.karticky."`, je preto vhodné uložiť v globálnej premennej.

Pri dopytovaní sa na kategórie sa v aplikácii bude v tomto prípade používať tag:

```
sk.mojaskola.karticky.kategorie
```

Pri dopytovaní sa na slová z konkrétnej kategórie sa v aplikácii použije napr. tag:

```
sk.mojaskola.karticky.povolania
```

Údaj o časovom limite je tiež vhodné uložiť do globálnej premennej, keďže pri realizovaní súťaže tímov musí byť pre všetkých zúčastnených rovnaký.

Ako vylepšiť či rozšíriť našu aplikáciu?

Naším cieľom bolo vytvorenie jednoduchej, ale funkčnej a v praxi použiteľnej aplikácie. Môžete zvážiť ďalšie vylepšenia svojej verzie, napr.:

- upraviť grafické používateľské rozhranie aplikácie,
- zabezpečiť, aby sa hra dala hrať aj bez internetového pripojenia (aplikácia by mala mať v tomto prípade v lokálnej databáze všetky kategórie slov a realizovať synchronizáciu so vzdialenou databázou len na požiadanie),
- umožniť nastavovanie niektorých parametrov (napr. časového limitu, počtu kôl súťaže, náhodný výber kategórie, miešanie poradia slov a pod.),
- evidovať priebežné skóre pre viac tímov, ktoré sa pri hre striedajú,
- odpočítavať odštartovanie hry po rozpoznaní správnej polohy tabletu,
- signalizovať blížiaci sa koniec hry,
- umožniť editovanie dát vo vzdialenej databáze priamo v aplikácii atď.

Odporúčaný priebeh výučby		
Činnosť učiteľa	Činnosť žiaka	Poznámky
<i>1. hodina – Úvod, motivácia, Úlohy 1, 2 a 3</i>		
<p>Uvedenie témy tímového projektu: učiteľ moderuje diskusiu o spoločenských hrách.</p> <p>Učiteľ predstaví žiakov koncept hry, ktorú budú vyvíjať, najlepšie tak, že si so žiakmi hru Kartičky zahrá tradičným spôsobom. Učiteľ bude v roli hádajúceho.</p> <p>Učiteľ vyzve žiakov, aby navrhovali kategórie slov na hádanie, zapisuje ich na tabuľu. Jednu z nich náhodne vyberie a nechá žiakov, aby pre ňu vymysleli slová.</p> <p>Po skončení jedného kola hry učiteľ spolu so žiakmi uvažuje, v čom by bol pre túto spoločenskú hru prínosom tablet.</p> <p>Úloha 1: Oboznámenie sa s nástrojom na zlučovanie projektov. Zlúčenie projektov zrealizuje učiteľ spolu so žiakmi.</p> <p>Úloha 2: Získanie skúsenosti s komponentom <i>TinyWebDB</i>.</p> <p>Úloha 3: Experimentovanie s náklonom tabletu.</p>	<p>Žiaci reagujú na jeho otázky, diskutujú, hovoria o vlastných skúsenostiach.</p> <p>Žiaci navrhujú rôzne kategórie slov na hádanie. Pre vybranú kategóriu každý zo žiakov vymyslí slovo a napíše ho na papier tak, aby to ostatní nevideli (= vyrobí kartičku). V priebehu hry žiaci opisujú slová, ktoré im ukazuje učiteľ.</p> <p>Úlohu riešia vo dvojiciach všetci žiaci (spolu pri jednom počítači).</p> <p>Úlohu rieši len prvý člen tímu.</p> <p>Úlohu rieši len druhý člen tímu.</p>	<p>Žiakom je vhodné zdôrazniť, že nás v tejto chvíli nezaujímajú počítačové hry pre viacerých hráčov. Tablet chápeme ako podporu pri hraní sa spoločenskej hry, ktorú by bolo možné zahrať sa aj bez neho.</p> <p>Jeden zo žiakov bude na tabuľu zapisovať skóre.</p> <p>Výsledkom diskusie by malo byť identifikovanie dvoch problémov, ktoré je možné riešiť nezávisle (komunikácia s databázou, riadenie hry náklonom tabletu) a nápad na vytvorenie tímov.</p> <p>Žiaci by úlohu mali vyriešiť do 15 min., učiteľ ich usmerňuje tak, aby napredovali a venovali pozornosť podstatným veciam.</p> <p>Žiakom pripravíme krátky návod, aby mohli pracovať samostatne, učiteľ nerobí frontálny výklad.</p>

2. hodina – Úloha 4 a Úloha 5		
<p>Učiteľ zopakuje požiadavky na aplikáciu, vyzve žiakov, aby sa rozdelili do tímov.</p> <p>Učiteľ odpovedá na doplňujúce otázky žiakov a nechá žiakov, aby začali pracovať na svojej časti vyvíjanej aplikácie.</p>	<p>Každý žiak je ako člen tímu zodpovedný za svoju časť riešenia, pracuje samostatne.</p>	<p>Učiteľ upriami pozornosť žiakov na to, aby sa najprv dohodli na pomenovaní druhej obrazovky a lokálnej databázy. Pripomenie tiež, že druhý člen tímu má začať svoju prácu pridaním novej obrazovky.</p>
3. hodina – Pokračovanie práce na projekte		
<p>Učiteľ podporuje žiakov pri práci, je v roli konzultanta.</p> <p>Učiteľ upozorní žiakov na to, že na ďalšej hodine overia funkčnosť svojich aplikácií v turnaji tímov.</p>	<p>Žiaci dokončujú svoje riešenia.</p> <p>V prípade, že jeden zo žiakov svoju prácu dokončil, pomôže druhému. V závere hodiny spolu svoje projekty zlúčia do jedného celku, otestujú aplikáciu na tablete.</p>	<p>Žiaci vyvíjajú aplikáciu ako tímový projekt, výsledné produkty by sa mali líšiť vo viacerých detailoch.</p> <p>Ak žiaci nepoužili na uloženie prefixu používaného pri tagoch a na uloženie časového limitu globálne premenné, mali by to v zdrojovom kóde opraviť.</p>
4. hodina – Vyhodnotenie projektu, turnaj tímov		
<p>Učiteľ každému žiakovi určí názov kategórie, pre ktorú má vymyslieť 20 slov.</p> <p>Spolu so žiakmi prediskutujú pravidlá turnaja (spôsob opisovania slov, výber kategórie).</p> <p>Učiteľ zapisuje výsledky jednotlivých súbojov do „pavúka“ na tabuľu.</p> <p>Cieľom spoločnej hry je zábavné ukončenie projektu a overenie funkčnosti riešení.</p>	<p>Každý žiak uloží do webovej databázy pre svoju kategóriu zoznam slov s využitím webového rozhrania služby <i>Tiny WebDB</i></p> <p>Ak v priebehu hry aplikácia znemožní tímu pokračovať alebo sa nespráva korektne, tím prehral. Dôvodom na prehru je tiež situácia, keď vysvitne, že žiak nenaplnil kategóriu slovami správne.</p>	<p>Prefix pre tagy a časový limit (napr. 120 sekúnd) určí žiakom učiteľ. Žiaci upravujú tieto hodnoty vo svojich aplikáciách.</p> <p>Žiaci nevedia názvy ostatných kategórií, pred začatím hry by sa preto slová nemali dozvedieť.</p> <p>V priebehu turnaja by mal učiteľ ustrážiť, aby žiadny žiak nehádal slová, ktoré sám do databázy uložil.</p> <p>Učiteľ môže všetkých, ktorým aplikácia v priebehu hry nezlyhala, odmeniť známku.</p>