

2. Tvorba jednoduchých aplikácií – programátorských etúd

Kľúčové slová

Softvérová aplikácia, životný cyklus vývoja aplikácie, udalosťami riadené programovanie, údajové typy, riadiace konštrukcie, senzory, komponenty, udalosti, metódy, vlastnosti

Čo sa naučíme a čo si precvičíme

- Analyzovať funkcionality softvérovej aplikácie:
 - na základe prečítania jej programového kódu
 - na základe jej spustenia
- Modifikovať, resp. rozšíriť softvérovú aplikáciu o nové funkcionality
- Prehliť teoretické poznatky o životnom cykle vývoja softvérovej aplikácie pre mobilné zariadenie a prakticky precvičiť vývoj udalosťami riadenej aplikácie
- Precvičiť základné údajové typy a riadiace konštrukcie prostredia Ai2
- Rozvíjať tvorivosť pri rozširovaní funkcionality aplikácie
- Vysvetliť význam využitia jednotlivých programátorských etúd v každodennom živote pre určité cieľové skupiny používateľov.

Príprava na výučbu

Okrem zabezpečenia Android mobilných zariadení pre žiakov budeme potrebovať pracovné listy, sebahodnotiace karty a súbory so zdrojovými kódmi aplikácií – programátorských etúd (AIA súbory). Učiteľ bude pre demonštráciu tvorby aplikácie, či komentovanie jej funkcionalít potrebovať dataprojektor, vývojový počítač a mobilné zariadenie vzájomne prepojené pomocou programu TeamViewer.

Odporúčaný priebeh výučby

V tejto kapitole sú predstavené vybrané možnosti Ai2 na zbierke 12 skupín jednoduchých aplikácií – programátorských etúd. Každá etuda pokrýva vybranú úzku skupinu možnosti Ai2 a tiež námety na objavovanie ďalších príbuzných možnosti Ai2 a námety na ďalšie zmysluplné rozšírenie etudy. Táto zbierka programátorských etúd predstavuje základné stavebné kamene využiteľné pri vyvíjaní väčších a komplexnejších projektov.

Výber etúd a ich poradie zaradenia do výučby necháme na učiteľa. Pomôckou mu bude tabuľka mapujúca prvky učiva jednotlivými aplikáciami uvedenými v tejto kapitole a tiež ďalšie štyri kapitoly popisujúce vývoj 12 náročnejších užitočných aplikácií. Vo výučbe môže učiteľ použiť tradičný prístup založený na tom, že demonštruje riešenie programovania jednotlivých etúd a ich rozširovanie buď urobí sám alebo ich nechá urobiť svojim žiakom. Odporúčame však, aby učiteľ poskytol žiakom programátorské etudy, ktorí by ich mali aktívne a samostatne skúmať a rozširovať. Učiteľ tu plní rolu usmerňovateľa a konzultanta svojim žiakom.

2.0. Zoznam programátorských etúd

Zbierka 12 programátorských etúd pokrýva vybrané funkcionality Ai2. Jednotlivé etudy sú rozpracované v rôznych verziách náročnosti, resp. s rôznym rozsahom rozšírenia (v zátvorkách sú žltým podfarbením zvýraznené v etudách použité komponenty Ai2):

- 2.1. **Kresliaci editor** (2 aplikácie)
(**Canvas.TouchDown**, **TouchUp**; **Screen.Title**, **AppName**, **Icon**, **BackgroundImage**, **BackgroundColor**)
- 2.2. **Hra Postreh** (3 aplikácie)
(**Clock.Timer**, **TimerInterval**, **TimerEnabled**; **Ball.TouchDown**, **MoveTo**, **Enabled**; **Button.Click**; **Label.text**; **HorizontalArrangement**; **Sound.Play**; **Screen.Initialize**; **Canvas.Width**, **Height**; globálna premenná – inicializácia, **set**, **get**; **close application**; **random integer from X to Y**)
- 2.3. **Hra Guľka** (5 aplikácii)
(**OrientationSensor.Enabled**; **Ball.X**, **Y**, **PaintColor**, **Visible**, **CollidedWith**; **Clock.Now**, **Duration**; **TinyDB.GetValue**, **StoreValue**; **IF**)
- 2.4. **Kalkulačka** (4 aplikácie)
(**TextBox.Text**; **Slider.PositionChanged**; **Notifier.ShowAlert**, **ShowTextDialog**, **ShowMessageDialog**, **AfterTextInput**; lokálna premenná, vlastná funkcia s parametrom; **IF-THEN-ELSE**; **join**)
- 2.5. **Zbierka vtipov** (3 aplikácie)
(**Screen.OtherScreenClosed**; **open another screen**, **close screen**, **close screen with value**; **Canvas.Flung**)
- 2.6. **Čítačka QR kódu** (3 aplikácie)
(**BarcodeScanner.DoScan**, **AfterScan**, **UseExternalScanner**; **TextToSpeech.Speak**, **Country**, **Language**, **SpeechRate**, **Pitch**; **Spinner.AfterSelecting**; **SpeechRecognizer.GetText**, **AfterGettingText**)
- 2.7. **Asistent pri cvičení** (4 aplikácie)
(**ProximitySensor.ProximityChanged**, **Enabled**; **Sound.Vibrate**; **CheckBox.Checked**; **Pedometer.Start**, **Resume**, **Reset Pause**, **Save**, **Stop**, **WalkStep**, **WalkSteps**, **Distance**, **ElapsedTime**, **StrideLength**, **StopDetectionTimeout**), **AccelerometerSensor.Enabled**; **TableArrangement**)
- 2.8. **Generátor náhodných viet** (4 aplikácie)
(**make a list**, **select list item**, **length of list**, **remove list item**, **insert list item**, **create empty list**, **add items to list**; **ListView.Elements**; **FOR EACH NUMBER**; **FOR EACH ITEM**)
- 2.9. **Zobrazovač aktuálnej polohy** (2 aplikácie)
(**LocationSensor.LocationChanged**, **latitude**, **longitude**, **altitude**, **speed**, **StatusChanged**,

status, LatitudeFromAddress, LongitudeFromAddress, ProviderName, HasAccuracy, Accuracy, CurrentAddress, TimeInterval, DistanceInterval, Enabled; **Map**)

2.10. Asistent aktuálnej polohy (3 aplikácie)

(**ActivityStarter.Action**, **DataUri**, **Extras**, **StartActivity**; **ListPicker.Elements**, **Selection**, **AfterPicking**, **BeforePicking**, **TinyDB.GetTags**, **ClearAll**; **HorizontalArrangement.Visible**; pick a random item)

2.11. Hlasovanie na internete (2 aplikácie)

(**FirestoreDB.StoreValue**, **GetValue**, **GetValue**, **DataChanged**, **tag**, **value**, **FirestoreURL**; index in list; max)

2.12. Komunikačný asistent (2 aplikácie)

(**Texting.MessageReceived**, **number**, **messageText**, **SendMessage**, **PhoneNumber**, **Message**, **ReceivingEnabled**; **PhoneCall.MakePhoneCall**, **PhoneNumber**)

Pre úplnosť a orientáciu čitateľa uvádzame aj prehľad ďalších komponentov Ai2, ktoré nie sú pokryté uvedenými etudami:

Skupina	Komponenty
User Interface	DataPicker, PasswordTextBox, TimePicker, WebView
Layout	-
Media	Camcorder, ImagePicker, Player, SoundRecorder, VideoPlayer, Yandex Translate
Drawing and Animation	ImageSprite
Maps	Circle, FeatureCollection, LineString, Marker, Polygon, Rectangle
Sensors	GyroscopeSensor, NearField
Social	ContactPicker, EmailPicker, PhoneNumberPicker, Sharing, Twitter
Storage	File, Fusion Tables, TinyWebDB
Connectivity	BluetoothClient, BluetoothServer, Web
LEGO® MINDSTORMS®	NxtDirectCommands, NxtColorSensor, NxtLightSensor, NxtSoundSensor, NxtTouchSensor, NxtUltrasonicSensor, NxtDrive Ev3Motors, Ev3ColorSensor, Ev3GyroSensor, Ev3TouchSensor, Ev3UltrasonicSensor, Ev3Sound, Ev3UI, Ev3Commands
Experimental	CloudDB

Niektoré z týchto komponentov (nepoužitých v etudách) sú vysvetlené a použité pri projektoch v kapitolách 3 až 6.

Pokrytie vybraných komponentov Ai2 problémami v programátorských etudách

názov problému	názov súboru s kódom	Použ. rozhranie		Mmédiá		Senzory					Komunik.		Pamäť		Jazykové koncepty																					
		Screen, Canvas, Ball, Image	Multiple Screens	Button, TextBox, Label, CheckBox	H/V/T Arrangements	ListView, ListPicker	Slider, Spinner	Notifier	Sound	TextToSpeech	SpeechRecognizer	Clock	AccelerometerSensor	LocationSensor	Map	OrientationSensor	BarcodeScanner	ProximitySensor	Pedometer		ActivityStarter	Texting	PhoneCall	TinyDB	FireBase	Cyklus	Vetvenie, podmienky	matematické operácie a funkcie	Globálne premenné (číslo, farby)	Zoznamy	Procedúry, funkcie	lokálne premenné				
Kresliaci editor (1.1, 1.2)	pmz_2_1_platno	■																																	2	
Kresliaci editor (1.3)	pmz_2_1_platno_R	■																																		2
Hra Postreh (2.1)	pmz_2_2_hra_postreh1	■																																		4
Hra Postreh (2.2)	pmz_2_2_hra_postreh2	■																																		7
Hra Postreh (2.3)	pmz_2_2_hra_postreh2_R	■																																		7
Hra Gulka (3.1)	pmz_2_3_hra_gulka1	■																																		4
Hra Gulka (3.2)	pmz_2_3_hra_gulka1_R	■																																		3
Stopky (3.3)	pmz_2_3_stopky	■																																		4
Počítadlo (3.4)	pmz_2_3_pocitadlo	■																																		5
Hra Gulka (3.5)	pmz_2_3_hra_gulka2_R	■																																		9
Kalkulačka BMI (4.1)	pmz_2_4_kalkulacka1a	■																																		5
Kalkulačka BMI (4.2)	pmz_2_4_kalkulacka1b	■																																		6
Kalkulačka BMI (4.3)	pmz_2_4_kalkulacka2_R	■																																		7
Kalkulačka BMI (4.4)	pmz_2_4_kalkulacka3	■																																		6
Zbierka vtipov (5.1)	pmz_2_5_vtipy1	■																																		5
Zbierka vtipov (5.2)	pmz_2_5_vtipy1_R	■																																		5
Zbierka vtipov (5.3)	pmz_2_5_vtipy2	■																																		5
Čítačka QR kódu (6.1)	pmz_2_6_QR_kod1	■																																		3
Čítačka QR kódu (6.2)	pmz_2_6_QR_kod2	■																																		5
Čítačka QR kódu (6.3)	pmz_2_6_QR_kod2_R	■																																		6
Asistent cvikov (7.1)	pmz_2_7_cviky1	■																																		4
Asistent cvikov (7.2)	pmz_2_7_cviky1_R	■																																		9
Asistent cvikov (7.3)	pmz_2_7_krokomer1	■																																		5
Asistent cvikov (7.4)	pmz_2_7_krokomer2	■																																		7
Generátor viet (8.1)	pmz_2_8_vety	■																																		5
Generátor viet (8.2)	pmz_2_8_vety_R	■																																		6
Spracovanie zoznamov (8.3)	pmz_2_8_zoznam_cisel	■																																		10
Spracovanie zoznamov (8.4)	pmz_2_8_zoznam_cisel_R	■																																		10
Zobrazovač GPS polohy (9.1)	pmz_2_9_gps	■																																		4
Zobrazovač GPS polohy (9.2)	pmz_2_9_gps_R	■																																		6
Asistent GPS polohy (10.1)	pmz_2_10_astarter_mapy	■																																		8
Asistent GPS polohy (10.2)	pmz_2_10_astarter_mapy_R	■																																		14
Spúšťač externých apiiek (10.3)	pmz_2_10_astarter_rozne	■																																		5
Webový kliker (11.1)	pmz_2_11_kliker	■																																		4
Webové hlasovanie (11.2)	pmz_2_11_hlasovanie_R	■																																		7
Hlasná čítačka SMS (12.1)	pmz_2_12_sms_nahlas	■																																		3
Asistent SMS (12.2)	pmz_2_12_sms_telefon_R	■																																		10

20 3 31 21 4 4 2 7 7 3 5 5 4 2 3 3 3 2 3 2 1 3 2 3 16 22 14 8 8 6

2.1. Kresliaci editor

Prvky nového učiva

Komponenty	Udalosti (premenné)	Metódy	Vlastnosti
Canvas	TouchDown TouchUp	-	PaintColor
Screen	-	-	Title AppName Icon BackgroundImage BackgroundColor

Jazykové konštrukcie	Iné prvky jazyka
-	-

Pracovné súbory

Meno súboru	Použitie súboru
pmz_2_1_platno_PL.docx	Pracovný list – Programujeme etudu 2.1 Kresliaci editor
pmz_2_1_platno_SK.docx	Sebahodnotiaca karta – Programujeme etudu 2.1 Kresliaci editor
pmz_2_1_platno.aia	Pracovný súbor pre úlohu 2
pmz_2_1_platno_R.aia	Súbor s riešením úlohy 3

Komentár k priebehu a výsledkom výučby

Hlavným cieľom tejto etudy je, aby žiak bol schopný rozšíriť svoju prvú aplikáciu vytvorenú v 1. kapitole na kresliacu aplikáciu využívajúcu ďalšie udalosti (TouchDown, TouchUp) a vlastnosti (PaintColor) komponentu Canvas. Zároveň táto aplikácia bude mať svoj titulný pás, nastavenú farbu a obrázok pozadia, vlastný názov a ikonu po inštalácii na mobilné zariadenie.

Žiaci riešia **úlohy z pracovného listu**, ktorý obsahuje:

Úloha 1 – žiaci skúmajú zdrojový kód aplikácie `pmz_2_1_platno.aia` a diskutujú nové funkcionality aplikácie (udalosti TouchDown, TouchUp komponentu Canvas).

Úloha 2 – žiaci importujú a inštalujú aplikáciu `pmz_2_1_platno.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (TouchDown, TouchUp).

Úloha 3 – žiaci rozširujú aplikáciu `pmz_2_1_platno.aia` o ďalšie funkcionality (vlastnosti Title, AppName, Icon, BackgroundImage, BackgroundColor komponentu Screen). Výsledný kód uložia do súboru `pmz_2_1_platno_R.aia`.

Na záver žiaci vyplnia **sebahodnotiacu kartu**, aby si uvedomili a zafixovali nové učivo. Sebahodnotiacu kartu môže učiteľ použiť na stručnú sumarizáciu nového učiva.

Pracovný list – Programujeme etudu 2.1 Kresliaci editor

Úloha 1: V aplikácii **Kreslička** z prvej kapitoly sme doplnili spracovanie ďalších dvoch udalostí **Canvas.TouchDown** a **Canvas.TouchUp**. V dvojiciach si preštudujte uvedený zdrojový kód a prediskutujte nové funkcionality aplikácie.

Úloha 2: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_1_platno.aia`. Zostavte z neho inštalateľný balík a nainštalujte ho na mobilné zariadenie. Po jeho spustení preskúmajte správanie sa aplikácie pri spracovaní nových udalostí **Canvas.TouchDown** a **Canvas.TouchUp** a svoje zistenia zapíšte do posledných dvoch riadkov tabuľky.

Komponent	Udalosť	Akcia
Canvas (Plátno)	Touched (Dotyk)	Na plátno sa vykreslí kruh so stredom v mieste dotyku a polomerom 10
Canvas (Plátno)	Dragged (Ťahanie)	Na plátno sa vykreslí úsečka od predchádzajúcej do aktuálnej pozície
AccelerometerSensor (Senzor zrýchlenia)	Shaking (Zatrasenie)	Zmaže sa obsah plátna
Canvas (Plátno)	TouchDown (.....)
Canvas (Plátno)	TouchUp (.....)

Úloha 3: V aplikácii `pmz_2_1_platno.aia` urobte zmeny uvedené v prvom stĺpci tabuľky a do druhého stĺpca na základe vlastného experimentovania uveďte aký efekt spôsobili tieto zmeny. Pri tvorbe ikon odporúčame použiť rozmer 48x48 a formát PNG (transparentný). Upravený kód uložte do súboru `pmz_2_1_platno_R.aia`

Zmena	Efekt
V komponente Screen nastaviť vlastnosť:	
a. Title na hodnotu Kreslička2	a.
b. AppName na hodnotu Kreslička2	b.
c. Icon na hodnotu farbicka_ikona.png	c.
d. BackgroundImage na hodnotu mriezka_16x9.png	d.
e. BackgroundColor na hodnotu yellow	e.

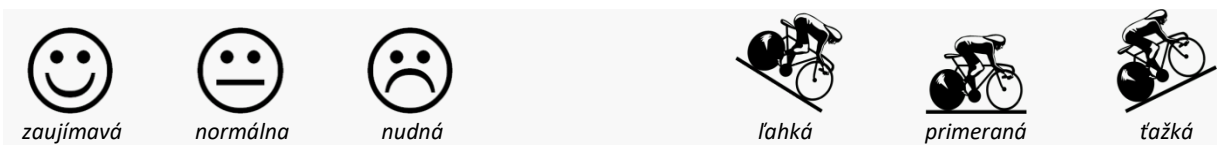
Sebahodnotiaca karta – Programujeme etudu 2.1 Kresliaci editor

Zapísaním symbolu ✓ na príslušné miesta tabuliek vyjadrite, do akej miery ovládáte uvedené prvky učiva.

Z uvedeného učiva nasledovným pojmom / princípom / tvrdeniam:	rozumiem	čiastočne rozumiem	vôbec nerozumiem
Udalosť Canvas.TouchDown sa vyvolá po priložení prstu (pera) na plátno			
Udalosť Canvas.TouchUp sa vyvolá po zdvihnutí prstu (pera) z plátna			
Vlastnosť Canvas.PaintColor odpovedá farbe, ktorou sa bude kresliť na plátno			
Vlastnosť Screen.Title odpovedá textu v titulnom páse aplikácie			
Vlastnosť Screen.AppName odpovedá názvu aplikácie po jej nainštalovaní na mobilné zariadenie			
Vlastnosť Screen.Icon odpovedá obrázku ikony, ktorá bude reprezentovať aplikáciu nainštalovanú na mobilnom zariadení			
Vlastnosť Screen.BackgroundImage odpovedá obrázku pozadia obrazovky			
Vlastnosť Screen.BackgroundColor odpovedá farbe pozadia obrazovky			

Z uvedeného učiva viem vykonať nasledovné činnosti:	samostatne	s malou pomocou	s veľkou pomocou
Nastaviť vlastnosť Canvas.PaintColor na danú farbu			
Nastaviť vlastnosť Screen.Title na daný text			
Nastaviť vlastnosť Screen.AppName na daný názov aplikácie			
Nastaviť vlastnosť Screen.Icon na daný obrázok ikony aplikácie			
Nastaviť vlastnosť Screen.BackgroundImage na daný obrázok pozadia obrazovky			
Nastaviť vlastnosť Screen.BackgroundColor na danú farbu pozadia obrazovky			
Vytvoriť kresliacu aplikáciu využívajúcu udalosti komponentu Canvas (TouchUp , TouchDown , Touched , Dragged)			

Aká bola pre vás táto etuda? Zaujímavá? Ľahká? Zafarbite/zakrúžkujte niektorú z uvedených možností:



Uveďte, čo by ste ešte doplnili do tejto mobilnej aplikácie, aby bola viac zaujímavá a využiteľná v praxi:

2.2. Hra Postreh

Prvky nového učiva

Komponenty	Udalosti (premenné)	Metódy	Vlastnosti
Clock	Timer		TimerInterval TimerEnabled
Ball	TouchDown	MoveTo	Enabled
Button	Click		
Label			Text
HorizontalArrangement			
Sound		Play	
Screen	Initialize		
Canvas			Width Height

Jazykové konštrukcie	Iné prvky jazyka
globálna premenná (inicializácia, get, set)	príkaz close application funkcia random integer from X to Y

Pracovné súbory

Meno súboru	Použitie súboru
pmz_2_2_hra_postreh_PL.docx	Pracovný list – Programujeme etudu 2.2 Hra Postreh
pmz_2_2_hra_postreh_R.docx	Riešenie – Programujeme etudu 2.2 Hra Postreh
pmz_2_2_hra_postreh_SK.docx	Sebahodnotiaca karta – Programujeme etudu 2.2 Hra Postreh
pmz_2_2_hra_postreh1.aia	Pracovný súbor pre úlohu 1
pmz_2_2_hra_postreh2.aia	Pracovný súbor pre úlohu 2
pmz_2_2_hra_postreh2_R.aia	Súbor s riešením úlohy 3

Komentár k priebehu a výsledkom výučby

Hlavným cieľom tejto etudy je, aby žiak bol schopný vytvoriť aplikáciu využívajúcu časovač (Clock.Timer) a generátor náhodných čísel na pohyb objektu (Ball) po plátne. Zároveň táto aplikácia reaguje na dotyk používateľa (Ball.TouchDown) zvukovým efektom (Sound) a počítaním dotykov (globálna premenná) a ich zobrazením (Label).

Žiaci riešia **úlohy z pracovného listu**, ktorý obsahuje:

Úloha 1 – žiaci importujú a inštalujú aplikáciu pmz_2_2_hra_postreh1.aia, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (Clock, Ball, Sound, random integer from X to Y).

Úloha 2 – žiaci importujú a inštalujú aplikáciu `pmz_2_2_hra_postreh2.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (Label, Button, HorizontalArrangement, Screen.Initialize, close application, globálna premenná).

Úloha 3 – žiaci rozširujú aplikáciu `pmz_2_2_hra_postreh2.aia` o ďalšie funkcionality (Ball.Enabled a Clock.TimerEnabled). Výsledný kód uložia do súboru `pmz_2_2_hra_postreh2_R.aia`.

Na záver žiaci vyplnia **sebahodnotiacu kartu**, aby si uvedomili a zafixovali nové učivo. Sebahodnotiacu kartu môže učiteľ použiť na stručnú sumarizáciu nového učiva.

Pracovný list – Programujeme etudu 2.2 Hra Postreh

Úloha 1: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_2_hra_postreh1.aia`. Po jeho nainštalovaní a spustení na mobilnom preskúmajte správanie sa tejto aplikácie. Svoje zistenia zapíšte do voľných políčok tabuliek.

(Poznámka: Pri skúmaní správania aplikácie odporúčame použiť referenčné materiály uvedené na stránkach: <http://ai2.appinventor.mit.edu/reference/components/> a <https://developer.android.com/guide/topics/media/media-formats>)

```

when Clock1.Timer
do
  call Ball1.MoveTo
    x random integer from 0 to Canvas1.Width
    y random integer from 0 to Canvas1.Height
  
```

```

when Ball1.TouchDown
  x y
do
  call Sound1.Play
  
```

Otázka	Odpoveď
a. V ktorej skupine komponentov je uvedený komponent Ball ?	a.
b. Ktoré vlastnosti má komponent Ball ?	b.
c. Do ktorého komponentu môžeme vložiť komponent Ball ?	c.
d. Na čo sa dá využiť komponent Ball ?	d.
e. Ktoré zvukové formáty vie prehrávať komponent Sound ?	e.
f. Čo predstavujú vlastnosti Canvas.Width a Canvas.Height ?	f.
g. Čo vracia funkcia random integer from X to Y ?	g.
h. Ako sa zmení správanie aplikácie (udalosti Clock.Timer) ak zmeníme v komponente Clock vlastnosti TimerInterval a TimerEnabled ?	h.

Komponent	Udalosť	Akcia
Ball (.....)	TouchDown (.....)	Sound.Play
Clock (.....)	Timer (.....)	Ball.MoveTo

Úloha 2: Preskúmajte používateľské rozhranie a správanie aplikácie so zdrojovým kódom uloženým v súbore `pmz_2_2_hra_postreh2.aia`. Svoje zistenia zapíšte do voľných políčok tabuliek.

```

initialize global dotykov to 0

when Clock1.Timer
do
  call Ball1.MoveTo
  random integer from 0 to Canvas1.Width
  random integer from 0 to Canvas1.Height

when Ball1.Touched
do
  call Sound1.Play
  set global dotykov to get global dotykov + 1
  set Label_Dotykov.Text to get global dotykov

when Screen1.Initialize
do
  set global dotykov to 0
  set Label_Dotykov.Text to get global dotykov

when Button_Start.Click
do
  set global dotykov to 0
  set Label_Dotykov.Text to get global dotykov

when Button_Koniec.Click
do
  close application
  
```

Otázka	Odpoveď
a. Ktoré vizuálne a nevizuálne komponenty tvoria používateľské rozhranie aplikácie?	a.
b. Do ktorého komponentu je vložený komponent Ball (lopta)?	b.
c. Na čo slúži komponent Button (tlačidlo)?	c.
d. Na čo slúži komponent Label (popisok)?	d.
e. Na čo slúži komponent HorizontalArrangement ?	e.
f. V ktorej situácii sa spracuje udalosť Screen.Initialize ?	f.
g. Čo urobí príkaz close application v rámci udalosti Button_Koniec.Click ?	g.
h. V zdrojovom kóde aplikácie vyznačte časti, v ktorých sa pracuje s globálnou premennou dotykov .	h.
i. Čo sa stane s hodnotou premennej dotykov , ak sa viackrát rýchlo dotkneme lopty na danom mieste?	i.
j. Ktoré multimediálne súbory sú použité v aplikácii a akým spôsobom?	j.

Komponent	Udalosť	Akcia
Ball	TouchDown	Sound.Play
Clock	Timer	Ball.MoveTo = náhodná pozícia na plátne
Button Koniec	Click
Button Štart	Click
Screen	Initialize

Úloha 3: Vylepšite aplikáciu `pmz_2_2_hra_postreh2.aia` zmenou nastavovania vlastnosti **Ball.Enabled** a **Clock.TimerEnabled** na hodnoty **true** a **false**. Výsledný kód uložte do súboru `pmz_2_2_hra_postreh2_R.aia`.

Riešenie – Programujeme etudu 2.2 Hra Postreh

Úloha 3: Vylepšite aplikáciu `pmz_2_2_hra_postreh2.aia` zmenou nastavovania vlastnosti `Ball.Enabled` a `Clock.TimerEnabled` na hodnoty `true` a `false`. Výsledný kód uložte do súboru `pmz_2_2_hra_postreh2_R.aia`.

V riešení upravíme spracovanie udalosti nasledovne:

Komponent	Udalosť	Akcia
Ball	TouchDown	Zahraj zvuk Zvýš hodnotu premennej <i>dotykov</i> o 1 Zobraz v komponente Label_Dotykov hodnotu premennej <i>dotykov</i> V komponente Ball vypni vlastnosť Enabled
Clock	Timer	Premiestni komponent Ball na náhodnú pozíciu na plátne V komponente Ball zapni vlastnosť Enabled
Button Koniec	Click	Uzavri aplikáciu
Button Štart	Click	Nastav hodnotu premennej <i>dotykov</i> na 0 Zobraz v komponente Label_Dotykov hodnotu premennej <i>dotykov</i> V komponente Clock zapni vlastnosť TimerEnabled V komponente Ball zapni vlastnosť Enabled
Screen	Initialize	Nastav hodnotu premennej <i>dotykov</i> na 0 Zobraz v komponente Label_Dotykov hodnotu premennej <i>dotykov</i> V komponente Clock vypni vlastnosť TimerEnabled V komponente Ball vypni vlastnosť Enabled

```

initialize global dotykov to 0

when Clock1.Timer
do
  call Ball1.MoveTo
  x random integer from 0 to Canvas1.Width
  y random integer from 0 to Canvas1.Height
  set Ball1.Enabled to true

when Button_Koniec.Click
do
  close application

when Ball1.Touched
x y
do
  call Sound1.Play
  set global dotykov to get global dotykov + 1
  set Label_Dotykov.Text to get global dotykov
  set Ball1.Enabled to false

when Screen1.Initialize
do
  set global dotykov to 0
  set Label_Dotykov.Text to get global dotykov
  set Clock1.TimerEnabled to false
  set Ball1.Enabled to false

when Button_Start.Click
do
  set global dotykov to 0
  set Label_Dotykov.Text to get global dotykov
  set Clock1.TimerEnabled to true
  set Ball1.Enabled to true
  
```

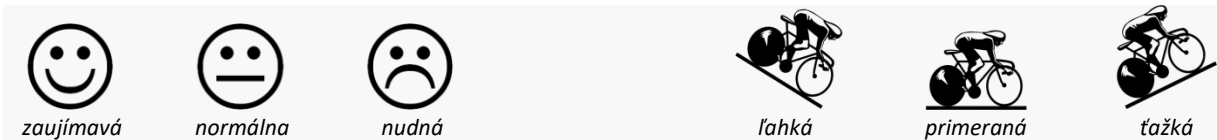
Sebahodnotiaca karta – Programujeme etudu 2.2 Hra Postreh

Zapísaním symbolu ✓ na príslušné miesta tabuliek vyjadrite, do akej miery ovládate uvedené prvky učiva.

Z uvedeného učiva nasledovným pojmom / princípom / tvrdeniam:	rozumiem	častočne rozumiem	vôbec nerozumiem
Komponent Clock je nevizuálnym komponentom súvisiaci s meraním a vypisovaním času, dátumu a s časovými intervalmi			
Udalosť Clock.Timer sa opakovane vyvoláva po uplynutí časového intervalu zadaného vo vlastnosti Clock.TimerInterval			
Udalosť Clock.Timer sa nevyvolá, ak je vypnutá vlastnosť Clock.TimerEnabled (t. j. je nastavená na hodnotu false)			
Komponent Ball je vizuálnym komponentom často využívaným na animácie, ktorý je umiestnený na komponente Canvas			
Udalosť Ball.TouchDown sa vyvolá po dotyku prstu (pera) na komponent Ball			
Komponent Ball je neaktívnym, ak je vypnutá jeho vlastnosť Ball.Enabled (t. j. je nastavená na hodnotu false)			
Komponent Button je vizuálnym komponentom využívaným hlavne na spúšťanie rôznych aktivít pomocou udalosti Button.Click			
Komponent Label je vizuálnym komponentom využívaným na výpis hodnôt vlastností komponentov, či premenných. Tieto hodnoty sú uložené vo vlastnosti Label.Text			
Udalosť Screen.Initialize sa vyvolá hneď po otvorení aplikácie			
Komponent HorizontalArrangement sa využíva ako kontajner na vodorovné umiestnenie rôznych komponentov vedľa seba			
Príkaz close application ukončí beh aplikácie			
Komponent Sound je nevizuálnym komponentom, ktorý sa používa na prehrávanie krátkych zvukov uložených v rôznych zvukových formátoch, napr. vzorky (flac, wav, ogg, mp3, 3gp, mp4, aac, mkv), skladby (mid, xmf, mxmf), zvonenia (rtttl, rtx, ota, imy). Na prehrávanie zvuku sa používa metóda Sound.Play			
Vlastnosti Canvas.Width a Canvas.Height predstavujú šírku a výšku komponentu Canvas			
Funkcia random integer from X to Y vráti náhodné celé číslo z intervalu <X,Y>			
Pri výpočtoch môžeme využívať premenné . Hodnota globálnej premennej sa inicializuje pomocou špeciálneho bloku			
Priradenie hodnoty výrazu do premennej sa realizuje pomocou set bloku			
Hodnota premennej vo výraze je reprezentovaná get blokom			

Z uvedeného učiva viem vykonať nasledovné činnosti:	samostatne	s malou pomocou	s veľkou pomocou
Vytvoriť používateľské rozhranie aplikácie, ktoré obsahuje plátno s loptou a tiež vodorovne vedľa seba zarovnané tlačidlá a popisok			
Pre komponent Sound nastaviť zvukový súbor a použiť metódu Sound.Play			
Použiť udalosť Clock.Timer na spúšťanie určitých aktivít v pravidelných intervaloch			
Použiť zapínanie a vypínanie vlastností Clock.TimerEnabled a Ball.Enabled pre rôzne stavy bežiacej aplikácie			
Použiť udalosť Ball.TouchDown pre zachytenie dotyku prstu (pera) na komponente Ball			
Použiť udalosť Screen.Initialize pre počiatočné nastavenia aplikácie			
Použiť príkaz close application na ukončenie behu aplikácie			
Použiť premenné vo výpočtoch (inicializáciu , bloky set a get)			
Použiť generovanie náhodných celých čísel vo výpočtoch			

Aká bola pre vás táto etuda? Zaujímavá? Ľahká? Zafarbite/zakrúžkujte niektorú z uvedených možností:



Uvedte, čo by ste ešte doplnili do tejto mobilnej aplikácie, aby bola viac zaujímavá a využiteľná v praxi:

2.3. Hra Guľka

Prvky nového učiva

Komponenty	Udalosti (premenné)	Metódy	Vlastnosti
OrientationSensor	OrientationChanged (roll, pitch)		Enabled
Ball	CollidedWith		X Y Visible PaintColor
Clock		Now Duration	
TinyDB		GetValue StoreValue	

Jazykové konštrukcie	Iné prvky jazyka
Príkaz vetvenia IF	-

Pracovné súbory

Meno súboru	Použitie súboru
pmz_2_3_hra_gulka_PL.docx	Pracovný list – Programujeme etudu 2.3 Hra Guľka
pmz_2_3_hra_gulka_R.docx	Riešenie – Programujeme etudu 2.3 Hra Guľka
pmz_2_3_hra_gulka_SK.docx	Sebahodnotiaca karta – Programujeme etudu 2.3 Hra Guľka
pmz_2_3_hra_gulka1.aia	Pracovný súbor pre úlohu 1
pmz_2_3_stopky.aia	Pracovný súbor pre úlohu 3
pmz_2_3_pocitadlo.aia	Pracovný súbor pre úlohu 4
pmz_2_3_hra_gulka1_R.aia	Súbor s riešením úlohy 2
pmz_2_3_hra_gulka2_R.aia	Súbor s riešením úlohy 5

Komentár k priebehu a výsledkom výučby

Hlavným cieľom tejto etudy je, aby žiak bol schopný vytvoriť aplikáciu využívajúcu na pohyb lopty nakláňanie mobilného zariadenia (OrientationSensor.OrientationChanged), kolíziu s inou loptou (Ball.CollidedWith), meranie uplynulého času (Clock.Now, Clock.Duration) a zaznamenanie najlepšieho času hry do mobilného zariadenia (TinyDB.StoreValue, TinyDB.GetValue).

Žiaci riešia **úlohy z pracovného listu**, ktorý obsahuje:

Úloha 1 – žiaci importujú a inštalujú aplikáciu pmz_2_3_hra_gulka1.aia, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (OrientationSensor.OrientationChanged, Roll, Pitch, Ball.CollidedWith).

Úloha 2 – žiaci rozširujú aplikáciu `pmz_2_3_hra_gulka1.aia` o ďalšie funkcionality (`OrientationSensor.Enabled`, `Ball.Visible`). Výsledný kód uložia do súboru `pmz_2_3_hra_gulka1_R.aia`.

Úloha 3 – žiaci importujú a inštalujú aplikáciu `pmz_2_3_stopky.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (`Clock.Now`, `Duration`).

Úloha 4 – žiaci importujú a inštalujú aplikáciu `pmz_2_3_pocitadlo.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (`TinyDB.StoreValue`, `GetValue`).

Úloha 5 – žiaci rozširujú aplikáciu `pmz_2_3_hra_gulka1.aia` o ďalšie funkcionality uvedené v úlohách 3 a 4. Výsledný kód uložia do súboru `pmz_2_3_hra_gulka2_R.aia`.

Na záver žiaci vyplnia **sebahodnotiacu kartu**, aby si uvedomili a zafixovali nové učivo. Sebahodnotiacu kartu môže učiteľ použiť na stručnú sumarizáciu nového učiva.

Pracovný list – Programujeme etudu 2.3 Hra Gulka

Úloha 1: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_3_hra_gulka1.aia`. Po jeho nainštalovaní a spustení na mobilnom preskúmajte správanie sa tejto aplikácie. Svoje zistenia zapíšte do voľných políчков tabuliek.

```
when OrientationSensor1 .OrientationChanged
  azimuth pitch roll
do
  set Ball_Gulka . X to Ball_Gulka . X - get roll
  set Ball_Gulka . Y to Ball_Gulka . Y - get pitch
```

```
when Ball_Gulka .CollidedWith
  other
do
  call Sound1 .Play
```

Otázka	Odpoveď
a. V ktorej skupine komponentov je uvedený komponent OrientationSensor ?	a.
b. Aký význam majú roll a pitch ako parametre udalosti OrientationSensor.OrientationChanged ?	b.
c. Aký význam má udalosť Ball.CollidedWith ?	c.
d. Ako sa správajú komponenty Ball_Gulka a Ball_Jama ?	d. Ball_Gulka Ball_Jama

Komponent	Udalosť	Akcia
OrientationSensor (.....)	OrientationChanged (.....)	set Ball_Gulka.X set Ball_Gulka.Y
Ball (.....)	CollidedWith (.....)	Sound.Play

Úloha 2: Upravte aplikáciu `pmz_2_3_hra_gulka1.aia`, aby mala nasledovné funkcionality:

- Pri kolízii lopty Gulka s iným komponentom (napr. loptou Jama) sa **zmení farba lopty Jama** na červenú (vlastnosť `PaintColor`), **schová sa lopta Gulka** (vlastnosť `Visible`) a **vypne sa komponent OrientationSensor** (vlastnosť `Enabled`)

- Doplní sa tlačidlo ŠTART, ktoré **nastaví y-súradnicu lopty Jama** na dolný okraj (vlastnosť `Y`), **nastaví farbu lopty Jama** na sivú farbu, **ukáže loptu Gulka** a **zapne komponent OrientationSensor**

Preskúmajte rozdiel medzi vlastnosťami **Enabled** a **Visible** komponentu **Ball**.

Výsledný kód uložte do súboru `pmz_2_3_hra_gulka1_R.aia`.

Úloha 3: Preskúmajte používateľské rozhranie a správanie aplikácie so zdrojovým kódom uloženým v súbore pmz_2_3_stopky.aia. Svoje zistenia zapíšte do voľných políčok tabuliek.

```

initialize global cas_start to 0

when Button_Start .Click
do set global cas_start to call Clock1 .Now

when Button_Stop .Click
do set Label_Trvanie .Text to call Clock1 .Duration
start get global cas_start
end call Clock1 .Now
    
```

Otázka	Odpoveď
a. Aký význam majú funkcie Clock.Now a Clock.Duration ?	a.
b. V akých časových jednotkách vracia výsledok funkcia Clock.Duration ?	b. Clock.Now Clock.Duration.....
c. V akých situáciách by ste využili tento programový kód?	c.

Komponent	Udalosť	Akcia
Button_Start	Click
Button_Stop	Click

Úloha 4: Preskúmajte používateľské rozhranie a správanie aplikácie so zdrojovým kódom uloženým v súbore pmz_2_3_pocitadlo.aia. Svoje zistenia zapíšte do voľných políчков tabuliek.

```

when Screen1 Initialize
do
  set Label_Poradie . Text to call TinyDB1 .GetValue
  tag "poradie"
  valueIfTagNotThere 0

when Button_Koniec .Click
do
  call TinyDB1 .StoreValue
  tag "poradie"
  valueToStore Label_Poradie . Text + 1
close application
  
```

Otázka	Odpoveď
a. Ako sa správa aplikácia po viacnásobnom spustení a ukončení?	a.
b. Akú máte skúsenosť s (nerelačnou) databázou typu tag:value , v ktorej sú údaje uložené ako dvojice klúč:hodnota ? (Poznámka: táto databáza sa označuje rôzne – slovník, asociatívne pole, mapa...)	b.
c. Čo je klúčom a čo hodnotou našej databázy v uvedenom programovom kóde?	c. klúč hodnota
d. Na čo slúži metóda TinyDB.GetValue ?	d.
e. Na čo slúži metóda TinyDB.StoreValue ?	e.
f. Aký význam má ValueIfTagNotThere ?	f.

Komponent	Udalosť	Akcia
Screen	Initialize
Button Koniec	Click

Úloha 5: Za pomoci programových kódov uvedených v predchádzajúcich úlohách vytvorte **hru Gulka**, ktorá bude mať nasledovné funkcionality:

- Nakláňaním mobilného zariadenia sa snažíme dostať malú guľku (loptu) do väčšej kruhovej jamky
- Ak dostaneme guľku do jamky, guľka sa schová, jamka sa zafarbí na červeno a hra končí
- Po skončení hry sa zaznamená čas trvania hry do databázy, ale len vtedy ak je v hre dosiahnutý čas menší ako čas predtým uložený do databázy

Výsledný kód uložte do súboru pmz_2_3_hra_gulka2_R.aia.

Riešenie – Programujeme etudu 2.3 Hra Gulka

Úloha 2: Upravte aplikáciu `pmz_2_3_hra_gulka1.aia`, aby mala nasledovné funkcionality:

- Pri kolízii lopty Gulka s iným komponentom (napr. loptou Jama) sa **zmení farba lopty Jama** na červenú (vlastnosť `PaintColor`), **schová sa lopta Gulka** (vlastnosť `Visible`) a **vypne sa komponent OrientationSensor** (vlastnosť `Enabled`)
- Doplní sa tlačidlo ŠTART, ktoré **nastaví y-súradnicu lopty Jama** na dolný okraj (vlastnosť `Y`), **nastaví farbu lopty Jama** na sivú farbu, **ukáže loptu Gulka** a **zapne komponent OrientationSensor**.

Zdrojový kód výslednej aplikácie `pmz_2_3_hra_gulka1_R.aia`:

```

when OrientationSensor1 .OrientationChanged
  azimuth pitch roll
do
  set Ball_Gulka . X to Ball_Gulka . X - get roll
  set Ball_Gulka . Y to Ball_Gulka . Y - get pitch

when Ball_Gulka .CollidedWith
  other
do
  set Ball_Jama . PaintColor to red
  set Ball_Gulka . Visible to false
  set OrientationSensor1 . Enabled to false

when Button_Start .Click
do
  set Ball_Gulka . Y to Canvas1 . Height - Ball_Gulka . Radius
  set Ball_Jama . PaintColor to gray
  set Ball_Gulka . Visible to true
  set OrientationSensor1 . Enabled to true
  
```

Úloha 5: Za pomoci programových kódov uvedených v predchádzajúcich úlohách vytvorte **hru Guľka**, ktorá bude mať nasledovné funkcionality:

- Nakláňaním mobilného zariadenia sa snažíme dostať malú guľku (loptu) do väčšej kruhovej jamky
- Ak dostaneme guľku do jamky, guľka sa schová, jamka sa zafarbí na červeno a hra končí
- Po skončení hry sa zaznamená čas trvania hry do databázy, ale len vtedy ak je v hre dosiahnutý čas menší ako čas predtým uložený do databázy

Výsledný kód uložte do súboru `pmz_2_3_hra_gulka2_R.aia`.

Zdrojový kód výslednej aplikácie `pmz_2_3_hra_gulka2_R.aia`:

```

when OrientationSensor1 .OrientationChanged
  azimuth pitch roll
do
  set Ball_Gulka . X to Ball_Gulka . X - get roll
  set Ball_Gulka . Y to Ball_Gulka . Y - get pitch

when Ball_Gulka .CollidedWith
  other
do
  set Ball_Jama . PaintColor to red
  set Ball_Gulka . Visible to false
  set OrientationSensor1 . Enabled to false
  set Label_Trvanie . Text to call Clock1 .Duration / (1000)
  start get global cas_start
  end call Clock1 .Now
  if Label_Trvanie . Text < Label_Rekord . Text
  then
    set Label_Rekord . Text to Label_Trvanie . Text
    call TinyDB1 .StoreValue
    tag "trvanie"
    valueToStore Label_Trvanie . Text

when Button_Start .Click
do call Start

when Screen1 .Initialize
do call Start

initialize global cas_start to 0

to Start
do
  set Ball_Gulka . Y to Canvas1 . Height - Ball_Gulka . Radius
  set Ball_Jama . PaintColor to gray
  set Ball_Gulka . Visible to true
  set OrientationSensor1 . Enabled to true
  set global cas_start to call Clock1 .Now
  set Label_Rekord . Text to call TinyDB1 .GetValue
  tag "trvanie"
  valueIfTagNotThere "100"
  
```







Sebahodnotiaca karta – Programujeme etudu 2.3 Hra Guľka

Zapísaním symbolu ✓ na príslušné miesta tabuliek vyjadrite, do akej miery ovládáte uvedené prvky učiva.

Z uvedeného učiva nasledovným pojmom / princípom / tvrdeniam:	rozumiem	čistočne rozumiem	vôbec nerozumiem
Komponent OrientationSensor je nevizuálnym komponentom súvisiaci s naklonením mobilného zariadenia			
Komponent OrientationSensor je neaktívnym, ak je vypnutá jeho vlastnosť OrientationSensor.Enabled (t. j. je nastavená na hodnotu false)			
Udalosť Ball.CollidedWith sa vyvolá pri kolízii (zrážke) komponentu Ball s iným animačným komponentom na plátne			
Vlastnosti Ball.X a Ball.Y predstavujú súradnice komponentu Ball na rodičovskom komponente Canvas (ľavý horný bod plátna má súradnice (0,0), súradnica y narastá zhora nadol)			
Komponent Ball je viditeľným, resp. neviditeľným, ak je jeho vlastnosť Ball.Visible nastavená na hodnotu true, resp. false			
Vlastnosť Ball.PaintColor predstavuje farbu komponentu Ball			
Rozdiel medzi vlastnosťami Ball.Enable a Ball.Visible je v tom, že prvá znamená, že komponent Ball je povolený a reaguje na udalosti a druhá znamená, že komponent Ball je viditeľný			
Metóda Clock.Now vracia aktuálny čas (vo vlastnom formáte)			
Metóda Clock.Duration vracia čas v milisekundách uplynutý medzi časmi uvedenými v parametroch start a end			
Komponent TinyDB slúži na uloženie údajov do zariadenia, ktoré sú reprezentované ako dvojica kľúč:hodnota			
Na ukladanie a načítanie údajov z databázy sa používajú metódy GetValue a StoreValue			

Z uvedeného učiva viem vykonať nasledovné činnosti:	samostatne	s malou pomocou	s veľkou pomocou
Vytvorí hru, ktorá využíva:			
• nakláňanie zariadenia (komponent OrientationSensor)			
• kolíziu lôpt (udalosť Ball.CollidedWith)			
• meranie uplynutého času (metóda Clock.Duration)			
• ukladanie hodnôt do/z databázy (komponent TinyDB)			

Aká bola pre vás táto etuda? Zaujímavá? Ľahká? Zafarbite/zakrúžkujte niektorú z uvedených možností:

					
zaujímavá	normálna	nudná	ľahká	primeraná	ťažká

Uvedte, čo by ste ešte doplnili do tejto mobilnej aplikácie, aby bola viac zaujímavá a využiteľná v praxi:

2.4. Kalkulačka

Prvky nového učiva

Komponenty	Udalosti (premenné)	Metódy	Vlastnosti
TextBox			Text
Notifier	AfterTextInput (response)	ShowAlert ShowTextDialog ShowMessageDialog	-
Slider	PositionChanged (minValue, maxValue, thumbPosition)		-

Jazykové konštrukcie	Iné prvky jazyka
lokálna premenná (inicializácia, get, set) IF-THEN-ELSE (s výstupom, vnorený) vlastná funkcia s parametrom	join

Pracovné súbory

Meno súboru	Použitie súboru
pmz_2_4_kalkulacka_PL.docx	Pracovný list – Programujeme etudu 2.4 Kalkulačka BMI
pmz_2_4_kalkulacka_R.docx	Riešenie – Programujeme etudu 2.4 Kalkulačka BMI
pmz_2_4_kalkulacka_SK.docx	Sebahodnotiaca karta – Programujeme etudu 2.4 Kalkulačka BMI
pmz_2_4_kalkulacka1a.aia	Pracovný súbor pre úlohu 1
pmz_2_4_kalkulacka1b.aia	Pracovný súbor pre úlohu 2
pmz_2_4_kalkulacka2_R.aia	Súbor s riešením úlohy 3
pmz_2_4_kalkulacka3.aia	Pracovný súbor pre úlohu 4

Komentár k priebehu a výsledkom výučby

Hlavným cieľom tejto etudy je, aby žiak bol schopný vytvoriť aplikáciu na výpočet určitého matematického vzťahu (BMI index), a to jednak využitím vstupného a výstupného poľa (TextBox, Label) a posúvača (Slider), jednak využitím vyskakovacích okien (Notifier). Zároveň táto aplikácia využíva lokálne premenné a vnorený príkaz IF-THEN-ELSE aj s výstupom.

Žiaci riešia **úlohy z pracovného listu**, ktorý obsahuje:

Úloha 1 – žiaci importujú a inštalujú aplikáciu `pmz_2_4_kalkulacka1a.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (TextBox, lokálne premenné, vnorený príkaz IF-THEN-ELSE aj s výstupom, matematické výrazy s operátormi /, ^).

Úloha 2 – žiaci importujú a inštalujú aplikáciu `pmz_2_4_kalkulacka1b.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (vlastná funkcia s parametrom).

Úloha 3 – žiaci rozširujú aplikáciu `pmz_2_4_kalkulacka1b.aia` o ďalšie funkcionality (Slider). Výsledný kód uložia do súboru `pmz_2_4_kalkulacka2_R.aia`.

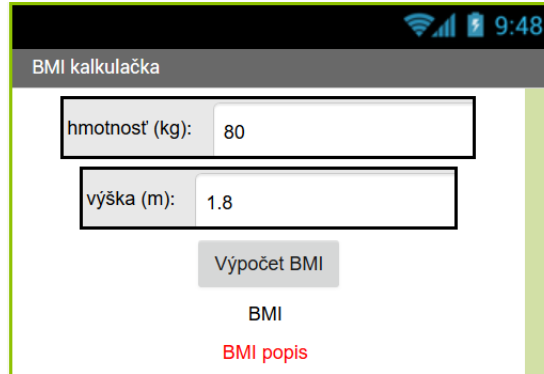
Úloha 4 – žiaci importujú a inštalujú aplikáciu `pmz_2_4_kalkulacka3.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (Notifier.ShowAlert, ShowTextDialog, ShowMessageDialog, join).

Na záver žiaci vyplnia **sebahodnotiacu kartu**, aby si uvedomili a zafixovali nové učivo. Sebahodnotiacu kartu môže učiteľ použiť na stručnú sumarizáciu nového učiva.

Pracovný list – Programujeme etudu 2.4 Kalkulačka BMI

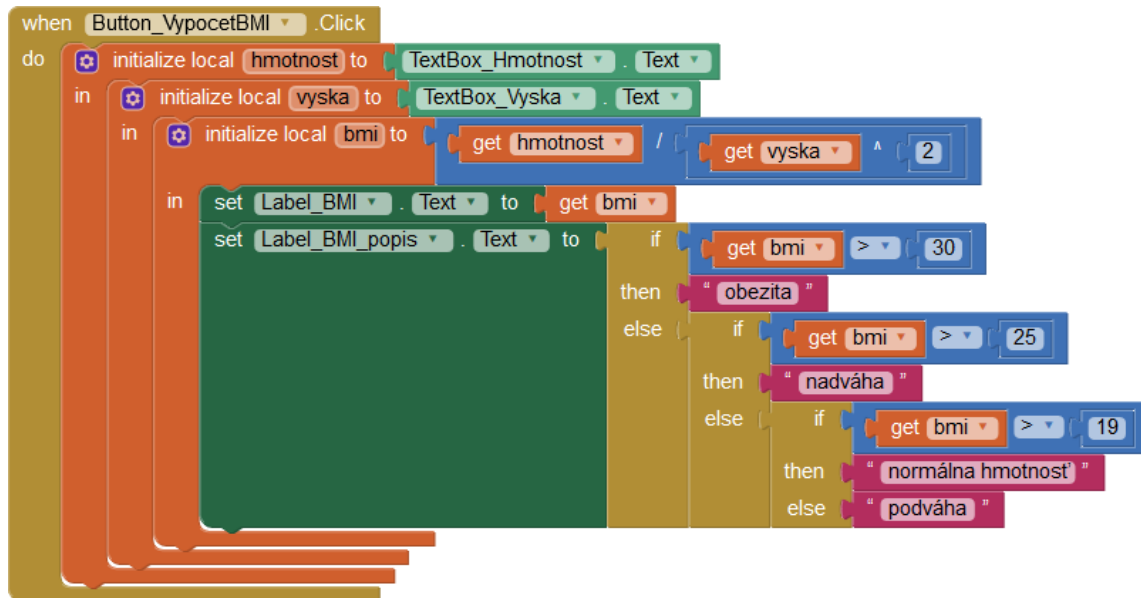
Úloha 1: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore pmz_2_4_kalkulacka.aia. Po nainštalovaní a spustení aplikácie na mobilnom zariadení preskúmajte jej správanie. Svoje zistenia zapíšte do voľných políček tabuliek.

Používateľské rozhranie:



Otázka	Odpoveď
a. Ktoré komponenty v tejto aplikácii zabezpečujú načítanie vstupov ?	a.
b. Ktoré komponenty v tejto aplikácii zabezpečujú spracovanie vstupov ?	b.
c. Ktoré komponenty v tejto aplikácii zabezpečujú výpis výstupov ?	c.
d. Na výpočet čoho je vhodná táto aplikácia?	d.

Zdrojový kód:



Otázka	Odpoveď
e. Ktoré globálne premenné a ktoré lokálne premenné sú použité v zdrojovom kóde?	e. globálne lokálne
f. Uveďte matematický výraz , ktorý je priradený do premennej bmi .	f.
g. Uveďte výsledok , ktorý bude uložený v komponente Label_BMI_popis pre nasledovné hodnoty bmi :	g. bmi = 24 Label_BMI_popis = bmi = 19 Label_BMI_popis = bmi = 31 Label_BMI_popis = bmi = 18 Label_BMI_popis =

Úloha 2: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_4_kalkulackalb.aia` a preskúmajte ho. Svoje zistenia zapíšte do voľných políчков tabuľky.

Zdrojový kód:

```

when Button_VypocetBMI .Click
do
  initialize local hmotnost to TextBox_Hmotnost .Text
  in initialize local vyska to TextBox_Vyska .Text
  in initialize local bmi to (get hmotnost) / ((get vyska) ^ 2)
  in set Label_BMI .Text to (get bmi)
  set Label_BMI_popis .Text to call BMI_popis
  bmi (get bmi)
  
```

```

to BMI_popis bmi
result
  if (get bmi) > 30
  then "obezita"
  else
    if (get bmi) > 25
    then "nadváha"
    else
      if (get bmi) > 19
      then "normálna hmotnosť"
      else "podváha"
  
```

Otázka	Odpoveď
a. V čom sa líši zdrojový kód aplikácie <code>pmz_2_4_kalkulackalb</code> od kódu <code>pmz_2_4_kalkulackala</code> ?	a.
b. Aké výhody prináša používanie vlastných funkcií ?	b.

Úloha 3: Rozšírite aplikáciu `pmz_2_4_kalkulackalb.aia`, aby sa vstupné hodnoty mohli zadať nielen pomocou komponentov `TextBox`, ale aj pomocou komponentov `Slider` (posúvačov). Výsledný kód uložte do súboru `pmz_2_4_kalkulacka2_R.aia`.

(Odporúčanie: V režime Designer rozšírite rozhranie aplikácie o dva komponenty `Slider`, v ktorých nastavíme dolnú a hornú hranicu povolených hodnôt. V režime Blocks doplníme dve udalosti `Slider.PositionChanged`, v ktorých nastavíme textové polia na aktuálnu hodnotu posúvača v parametri `thumbPosition` danej udalosti.)

Úloha 4: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore pmz_2_4_kalkulacka3.aia a preskúmajte používateľské rozhranie a správanie tejto aplikácie. Svoje zistenia zapíšte do voľných políček tabuľky.

initialize global hmotnost to 0 initialize global vyska to 0 initialize global bmi to 0

```

when Button_VypocetBMI .Click
do
  call Notifier_Vyska .ShowTextDialog
  message "Zadaj výšku v m: "
  title "Výška "
  cancelable false
  call Notifier_Hmotnost .ShowTextDialog
  message "Zadaj hmotnost' v kg: "
  title "Hmotnost' "
  cancelable false
  
```

```

when Notifier_Hmotnost .AfterTextInput
response
do
  set global hmotnost to get response
  call Notifier_Hmotnost .ShowAlert
  notice get global hmotnost
  
```

```

when Notifier_Vyska .AfterTextInput
response
do
  set global vyska to get response
  set global bmi to (get global hmotnost / (get global vyska ^ 2))
  call Notifier_Vyska .ShowMessageDialog
  message join "BMI: "
  get global bmi
  "<br> Hodnotenie: "
  call vysledok
  bmi get global bmi
  title "Výsledok: "
  buttonText "Zavri okno "
  
```

```

to vysledok bmi
result
  if get bmi > 30
  then "obezita"
  else
    if get bmi > 25
    then "nadváha"
    else
      if get bmi > 19
      then "normálna hmotnost'"
      else "podváha"
  
```

Otázka	Odpoveď
a. Vysvetlite funkcionalitu použitých metód komponentu Notifier :	a. Notifier.ShowDialog Notifier.ShowAlert Notifier.ShowMessageDialog
b. Ako by sa zmenil výpočet programu, ak by sme v udalosti Button_VypocetBMI.Click prehodili volania oboch uvedených metód?	b.
c. Ktorá z metód komponentu Notifier vyvolá udalosť Notifier.AfterTextInput ?	c.
d. Ktoré z metód komponentu Notifier by ste použili v uvedených situáciách :	d. len na výpis údajov na načítanie údajov

Riešenie – Programujeme etudu 2.4 Kalkulačka BMI

Úloha 3: Rozšírte aplikáciu `pmz_2_4_kalkulacka1b.aia`, aby sa vstupné hodnoty mohli zadať nielen pomocou komponentov `TextBox`, ale aj pomocou komponentov `Slider` (posúvačov). Výsledný kód uložte do súboru `pmz_2_4_kalkulacka2_R.aia`.

(Odporúčanie: V režime Designer rozšírime rozhranie aplikácie o dva komponenty `Slider`, v ktorých nastavíme dolnú a hornú hranicu povolených hodnôt. V režime doplníme dve udalosti `Slider.PositionChanged`, v ktorých nastavíme textové polia na aktuálnu hodnotu posúvača v parametri `thumbPosition` danej udalosti.)

Zdrojový kód výslednej aplikácie `pmz_2_4_kalkulacka2_R.aia`:

```

when Button_VypocetBMI .Click
do
  initialize local hmotnost to TextBox_Hmotnost .Text
  in initialize local vyska to TextBox_Vyska .Text
  in initialize local bmi to (get hmotnost / (get vyska ^ 2))
  in set Label_BMI .Text to get bmi
  set Label_BMI_popis .Text to call BMI_popis bmi
end

to BMI_popis bmi
result
  if get bmi > 30
  then "obezita"
  else
    if get bmi > 25
    then "nadváha"
    else
      if get bmi > 19
      then "normálna hmotnosť"
      else "podváha"
    end
  end
end

when Slider_Hmotnost .PositionChanged
thumbPosition
do set TextBox_Hmotnost .Text to get thumbPosition

when Slider_Vyska .PositionChanged
thumbPosition
do set TextBox_Vyska .Text to get thumbPosition
  
```

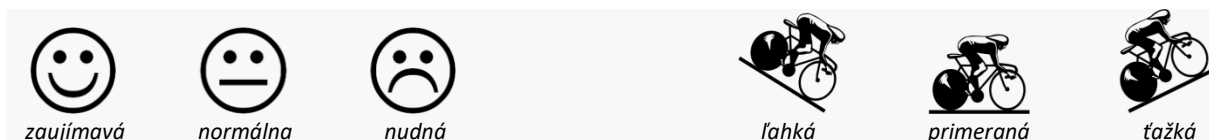
Sebahodnotiaca karta – Programujeme etudu 2.4 Kalkulačka BMI

Zapísaním symbolu ✓ na miesta tabuliek vyjadrite, do akej miery ovládáte uvedené prvky učiva.

Z uvedeného učiva nasledovným pojmom / princípom / tvrdeniam:	rozumiem	častočne rozumiem	vôbec nerozumiem
Vo výpočte môžeme použiť lokálne premenné , ktoré sú dostupné len vo vymedzenej časti programu			
Príkaz IF-THEN-ELSE môžeme viackrát vnoriť do seba			
Príkaz IF-THEN-ELSE môže mať aj výstup z vetiev THEN a ELSE			
Vo výrazoch na výpočet mocniny sa používa blok ^			
Vlastné funkcie (procedúry s výstupom) sa používajú na sprehľadnenie výpočtu, na opätovné použitie v programe, pri tímovej tvorbe programov			
Komponent Slider je vizuálnym komponentom súvisiaci so vstupom údajov podľa polohy posúvača			
Udalosť Slider.PositionChanged sa vyvolá zmenou polohy bežca posúvača medzi nastavenými krajnými pozíciami minValue a maxValue , pričom aktuálna hodnota posúvača je uložená v jej parametri thumbPosition			
Komponent Notifier je vizuálnym komponentom umožňujúcim vstupy a výstupy údajov pomocou dialógových okien			
Metóda Notifier.ShowAlert slúži na krátkodobý výpis správy v okne			
Metóda Notifier.ShowMessageDialog slúži na výpis správy v okne ukončený stlačením tlačidla okna			
Metóda Notifier.ShowTextDialog slúži na zadanie vstupných údajov pomocou dialógového okna			
Udalosť Notifier.AfterTextInput sa vyvolá po spustení metódy Notifier.ShowTextDialog , pričom aktuálne zadaný vstup je uložený v jej parametri response			
V metóde Notifier.ShowMessageDialog v parametri message môžeme na viacriadkový výpis použiť HTML značku
			

Z uvedeného učiva viem vykonať nasledovné činnosti:	samostatne	s malou pomocou	s veľkou pomocou
Vytvoriť výpočtovú aplikáciu využívajúcu na vstupy a výstupy: <ul style="list-style-type: none"> komponenty TextBox a Label komponent Notifier 			

Aká bola pre vás táto etuda? Zaujímavá? Ľahká? Zafarbite/zakrúžkujte niektorú z uvedených možností:



Uvedte, čo by ste ešte doplnili do tejto mobilnej aplikácie, aby bola viac zaujímavá a využiteľná v praxi:

2.5. Zbierka vtipov

Prvky nového učiva

Komponenty	Udalosti (premenné)	Metódy	Vlastnosti
Screen	OtherScreenClosed (result)		
Canvas	Flung (xvel)		

Jazykové konštrukcie	Iné prvky jazyka
	open another screen close screen close screen with value

Pracovné súbory

Meno súboru	Použitie súboru
pmz_2_5_zbierka_vtipov_PL.docx	Pracovný list – Programujeme etudu 2.5 Zbierka vtipov
pmz_2_5_zbierka_vtipov_R.docx	Riešenie – Programujeme etudu 2.5 Zbierka vtipov
pmz_2_5_zbierka_vtipov_SK.docx	Sebahodnotiaca karta – Programujeme etudu 2.5 Zbierka vtipov
pmz_2_5_vtipy1.aia	Pracovný súbor pre úlohu 1
pmz_2_5_vtipy1_R.aia	Súbor s riešením úlohy 2
pmz_2_5_vtipy2.aia	Pracovný súbor pre úlohu 3

Komentár k priebehu a výsledkom výučby

Hlavným cieľom tejto etudy je, aby žiak bol schopný vytvoriť aplikáciu využívajúcu viacero obrazoviek s využitím udalostí Button.Click a Canvas.Flung na otváranie obrazoviek.

Žiaci riešia **úlohy z pracovného listu**, ktorý obsahuje:

Úloha 1 – žiaci importujú a inštalujú aplikáciu `pmz_2_5_vtipy1.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (open another screen, Canvas.Flung).

Úloha 2 – žiaci rozširujú aplikáciu `pmz_2_5_vtipy1.aia` o ďalšie funkcionality (pridanie ďalšej obrazovky, doplnenie obrázkov do komponentov Canvas). Výsledný kód uložia do súboru `pmz_2_5_vtipy1_R.aia`.

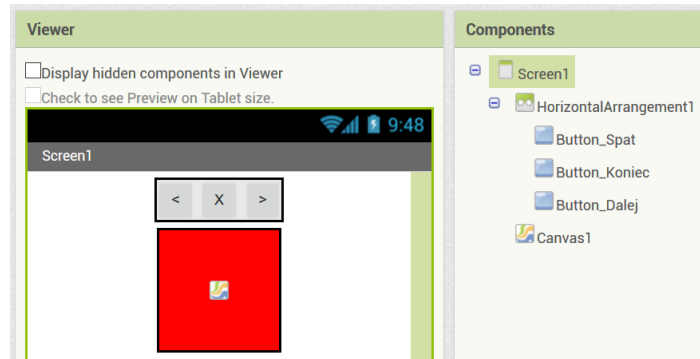
Úloha 3 – žiaci importujú a inštalujú aplikáciu `pmz_2_5_vtipy2.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (s hlavnou obrazovkou a uzatváraním obrazoviek – close screen (with value)).

Na záver žiaci vyplnia **sebahodnotiacu kartu**, aby si uvedomili a zafixovali nové učivo. Sebahodnotiacu kartu môže učiteľ použiť na stručnú sumarizáciu nového učiva.

Pracovný list – Programujeme etudu 2.5 Zbierka vtipov

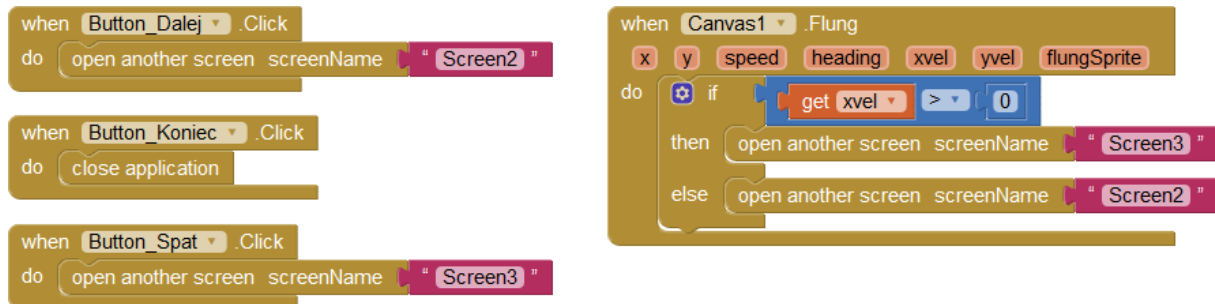
Úloha 1: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_5_vtipy1.aia`. Po nainštalovaní a spustení aplikácie na mobilnom zariadení preskúmajte jej správanie. Svoje zistenia zapíšte do voľných políček tabuliek.

Používateľské rozhranie obrazovky Screen1:



Otázka	Odpoveď
a. Čo sa stane po kliknutí na tlačidlá označené symbolmi „<“ a „>“?	a.
b. Čo sa stane, ak potiahneme prstom vľavo (resp. vpravo) po farebnom plátne?	b.
c. Koľko obrazoviek obsahuje táto aplikácia?	c.
d. Má každá obrazovka svoje multimediálne súbory ?	d. ÁNO / NIE
e. Má každá obrazovka svoje programové kódy ?	e. ÁNO / NIE
f. Uveďte aspoň jeden vlastný námet na aplikáciu s viacerými obrazovkami:	f.

Zdrojový kód obrazovky Screen1:



Otázka	Odpoveď
g. Aké použitie má príkaz open another screen ?	g.
h. Akou činnosťou používateľa sa spúšťa udalosť Canvas.Flung ?	h.
i. Čo predstavuje parameter xvel udalosti Canvas.Flung ?	i.

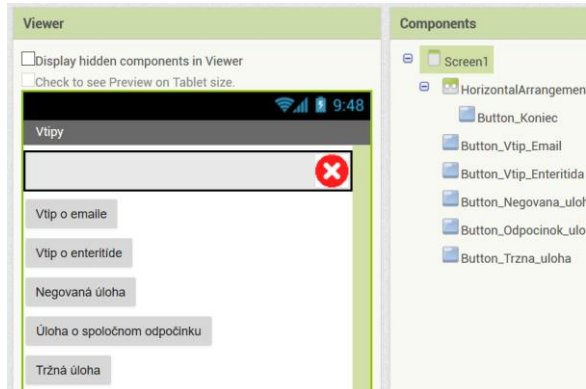
Úloha 2: Upravte aplikáciu `pmz_2_5_vtipy1.aia`, aby tlačidlá namiesto symbolov „<“, „X“ a „>“ boli reprezentované vhodnými obrázkami a komponenty **Canvas** namiesto rôznych farieb pozadia boli reprezentované vhodnými obrázkami s vtipmi či vtipnými zadaniami úloh. Aplikáciu rozšírite aspoň o jednu stranu s vtipom či vtipným zadaním úlohy. Výsledný kód uložte do súboru `pmz_2_5_vtipy1_R.aia`.

(Odporúčanie: Pri kopírovaní programového kódu z jednej obrazovky do druhej odporúčame použiť **schránku** (ikona modrozeleného ruksaku vpravo hore). Na kopírovanie objektov do ruksaku a z ruksaku používame kontextovú ponuku vyvolanú stlačením pravého gombíka myši. Obsah ruksaku zobrazíme stlačením ľavého gombíka myši.)

Úloha 3: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_5_vtipy2.aia` a preskúmajte používateľské rozhranie a správanie tejto aplikácie. Svoje zistenia zapíšte do voľných políчков tabuľky.

Používateľské rozhranie a zdrojový kód:

Obrazovka Screen1



```

when Button_Koniec .Click
do close application

when Screen1 .OtherScreenClosed
otherScreenName result
do if not is empty get result
then open another screen screenName get result

when Button_Vtip_Email .Click
do open another screen screenName "Screen2"

when Button_Vtip_Enteritida .Click
do open another screen screenName "Screen3"

when Button_Negovana_uloha .Click
do open another screen screenName "Screen4"

when Button_Odpocinok_uloha .Click
do open another screen screenName "Screen5"

when Button_Trzna_uloha .Click
do open another screen screenName "Screen6"

```

Obrazovka Screen2



```

when Canvas1 .Flung
x y speed heading xvel yvel flungSprite
do if get xvel > 0
then close screen with value result "Screen6"
else close screen with value result "Screen3"

when Button_Domov .Click
do close screen

when Button_Koniec .Click
do close application

```

Otázka	Odpoveď
a. Koľko obrazoviek obsahuje táto aplikácia?	a.
b. Ako sa líši prvá obrazovka od ostatných?	b.
c. Akú funkcionálnosť zastupuje udalosť <code>Screen.OtherScreenClosed</code> ?	c.
d. Prečo je v tejto udalosti uvedená podmienka?	d.
e. Prečo je na Screen2 v udalosti <code>Canvas.Flung</code> použitý príkaz <code>close screen</code> a nie <code>open screen</code> ?	e.
f. V čom je lepšie riešenie aplikácie <code>vtipy2</code> od aplikácie <code>vtipy1</code> ?	f.
g. Uveďte ďalší vlastný námet na aplikáciu s viacerými obrazovkami:	g.

Riešenie – Programujeme etudu 2.5 Zbierka vtipov

Úloha 2: Upravte aplikáciu `pmz_2_5_vtipy1.aia`, aby tlačidlá namiesto symbolov „<“, „X“ a „>“ boli reprezentované vhodnými obrázkami a komponenty Canvas namiesto rôznych farieb pozadia boli reprezentované vhodnými obrázkami s vtipmi či vtipnými zadaniami úloh. Aplikáciu rozšírite aspoň o jednu stranu s vtipom či vtipným zadaním úlohy. Výsledný kód uložte do súboru `pmz_2_5_vtipy1_R.aia`.

(Odporúčanie: Pri kopírovaní programového kódu z jednej obrazovky do druhej odporúčame použiť **schránku** (ikona modrozeleného ruksaku vpravo hore). Na kopírovanie objektov do ruksaku a z ruksaku používame kontextovú ponuku vyvolanú stlačením pravého gombíka myši. Obsah ruksaku zobrazíme stlačením ľavého gombíka myši.)

Používateľské rozhranie a zdrojový kód výslednej aplikácie `pmz_2_5_vtipy1_R.aia` – obrazovka Screen4:



```
when Button_Spat.Click
do open another screen screenName "Screen3"
```

```
when Button_Dalej.Click
do open another screen screenName "Screen1"
```

```
when Canvas1.Flung
do if (get xvel > 0)
then open another screen screenName "Screen3"
else open another screen screenName "Screen1"
```

```
when Button_Koniec.Click
do close application
```

Okrem pridanej obrazovky Screen4 sme upravili programové kódy

- na obrazovke Screen1:
v udalosti **Button_Spat.Click** aj v udalosti **Canvas.Flung** vo vetve **then** sme zmenili hodnotu `screenName` na **Screen4**
- na obrazovke Screen3:
v udalosti **Button_Dalej.Click** aj v udalosti **Canvas.Flung** vo vetve **else** sme zmenili hodnotu `screenName` na **Screen4**

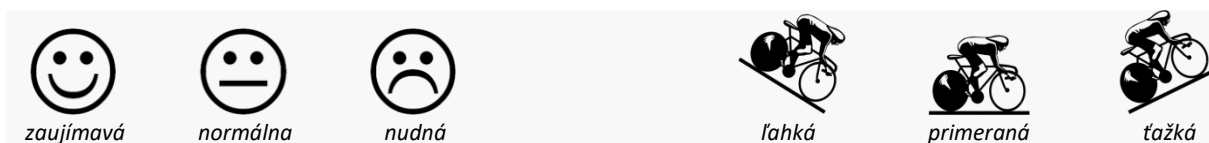
Sebahodnotiaca karta – Programujeme etudu 2.5 Zbierka vtipov

Zapísaním symbolu ✓ na príslušné miesta tabuliek vyjadrite, do akej miery ovládáte uvedené prvky učiva.

Z uvedeného učiva nasledovným pojmom / princípom / tvrdeniam:	rozumiem	čiastočne rozumiem	vôbec nerozumiem
V aplikácii s viacerými obrazovkami má každá obrazovka svoj vlastný programový kód			
V aplikácii s viacerými obrazovkami sú nahrané multimediálne súbory prístupné pre každú obrazovku			
Viacere obrazovky je vhodné použiť v aplikáciách, ktoré potrebujú zobraziť viacero údajov s rôznym rozmiestnením a účelom (napr. Úvod, Pomoc, Nastavenia, Výsledky)			
Na otvorenie obrazovky sa používa príkaz open another screen so zadaným menom obrazovky			
Na uzavretie obrazovky sa používa príkaz close another screen so zadaným výsledkom (napr. menom obrazovky), resp. príkaz close screen , ak ide o aktuálnu obrazovku			
Udalosť Screen.OtherScreenClosed sa vyvolá po uzavretí nejakej obrazovky, pričom v jej parametri result je uložený výsledok (napr. meno nasledovnej obrazovky), resp. result je prázdny reťazec po uzavretí obrazovky príkazom close screen			
Na otváranie obrazoviek môžeme použiť rôzne komponenty, napr. Button , Canvas			
Udalosť Canvas.Flunge sa vyvolá pomocou dotykového gesta, napr. potiahnutím vpravo, čo indikuje kladná hodnota jej parametri xvel			

Z uvedeného učiva viem vykonať nasledovné činnosti:	samostatne	s malou pomocou	s veľkou pomocou
Vytvoriť aplikáciu s viacerými obrazovkami: <ul style="list-style-type: none"> s rovnakým rozmiestnením obsahu bez uzatvárania obrazoviek 			
<ul style="list-style-type: none"> s rôznym rozmiestnením obsahu obrazoviek aj s uzatváraním obrazoviek 			

Aká bola pre vás táto etuda? Zaujímavá? Ľahká? Zafarbite/zakrúžkujte niektorú z uvedených možností:



Uvedte, čo by ste ešte doplnili do tejto mobilnej aplikácie, aby bola viac zaujímavá a využiteľná v praxi:

2.6. Čítačka QR kódu

Prvky nového učiva

Komponenty	Udalosti (premenné)	Metódy	Vlastnosti
BarcodeScanner	AfterScan (result)	DoScan	UseExternalScanner
TextToSpeech		Speak	Country Language SpeechRate Pitch
Spinner	AfterSelecting (selection)		
SpeechRecognizer	AfterGettingText (result)	GetText	

Jazykové konštrukcie	Iné prvky jazyka
-	-

Pracovné súbory

Meno súboru	Použitie súboru
pmz_2_6_citacka_QR_kodov_PL.docx	Pracovný list – Programujeme etudu 2.6 Čítačka QR kódu
pmz_2_6_citacka_QR_kodov_R.docx	Riešenie – Programujeme etudu 2.6 Čítačka QR kódu
pmz_2_6_citacka_QR_kodov_SK.docx	Sebahodnotiaca karta – Programujeme etudu 2.6 Čítačka QR kódu
pmz_2_6_QR_kod1.aia	Pracovný súbor pre úlohu 1
pmz_2_6_QR_kod2.aia	Pracovný súbor pre úlohu 2
pmz_2_6_QR_kod2_R.aia	Súbor s riešením úlohy 3

Komentár k priebehu a výsledkom výučby

Hlavným cieľom tejto etudy je, aby žiak bol schopný vytvoriť aplikáciu na čítanie čiarového kódu (BarcodeScanner) využívajúcu analýzu reči (TextToSpeech), syntézu reči (SpeechRecognizer) a tiež rozbaľovací zoznam (Spinner) .

Žiaci riešia **úlohy z pracovného listu**, ktorý obsahuje:

Úloha 1 – žiaci importujú a inštalujú aplikáciu `pmz_2_6_QR_kod1.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (BarcodeScanner, TextToSpeech).

Úloha 2 – žiaci importujú a inštalujú aplikáciu `pmz_2_6_QR_kod2.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (SpeechRecognizer, Spinner, TextBox).

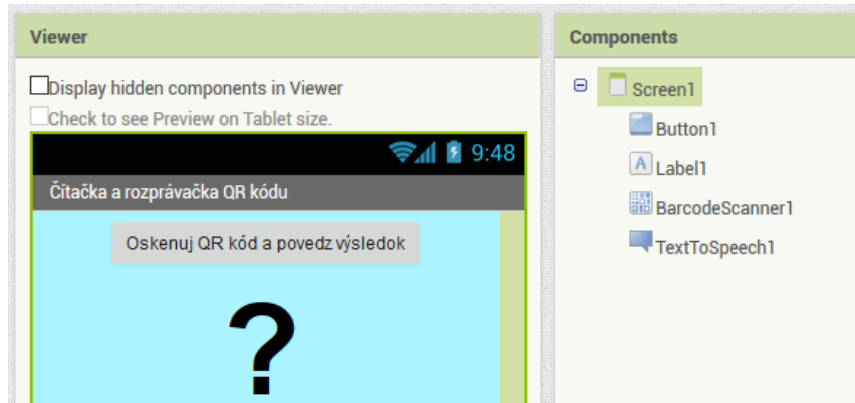
Úloha 3 – žiaci rozširujú aplikáciu `pmz_2_6_QR_kod2.aia` o ďalšie funkcionality (TextToSpeech.Country, Language SpeechRate, Pitch). Výsledný kód uložia do súboru `pmz_2_6_QR_kod2_R.aia`.

Na záver žiaci vyplnia **sebahodnotiacu kartu**, aby si uvedomili a zafixovali nové učivo. Sebahodnotiacu kartu môže učiteľ použiť na stručnú sumarizáciu nového učiva.

Pracovný list – Programujeme etudu 2.6 Čítačka QR kódu

Úloha 1: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_6_QR_kod1.aia`. Po nainštalovaní a spustení aplikácie na mobilnom zariadení preskúmajte jej správanie. Svoje zistenia zapíšte do voľných políčok tabuliek.

Používateľské rozhranie:



Otázka	Odpoveď
a. V ktorej skupine komponentov je uvedený komponent BarcodeScanner ?	a.
b. V ktorej skupine komponentov je uvedený komponent TextToSpeech ?	b.
c. Aký význam má komponent BarcodeScanner ?	c.
d. Aký význam má komponent TextToSpeech ?	d.
e. Pomocou ktorej aplikácie v mobilnom zariadení sa skenujú čiarové kódy?	e.

Zdrojový kód:

```

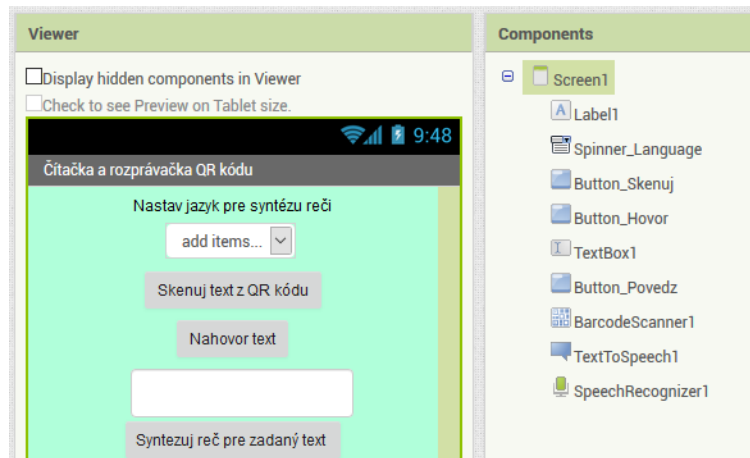
when Button1 .Click
do call BarcodeScanner1 .DoScan

when BarcodeScanner1 .AfterScan
result
do set Label1 .Text to get result
call TextToSpeech1 .Speak
message get result
    
```

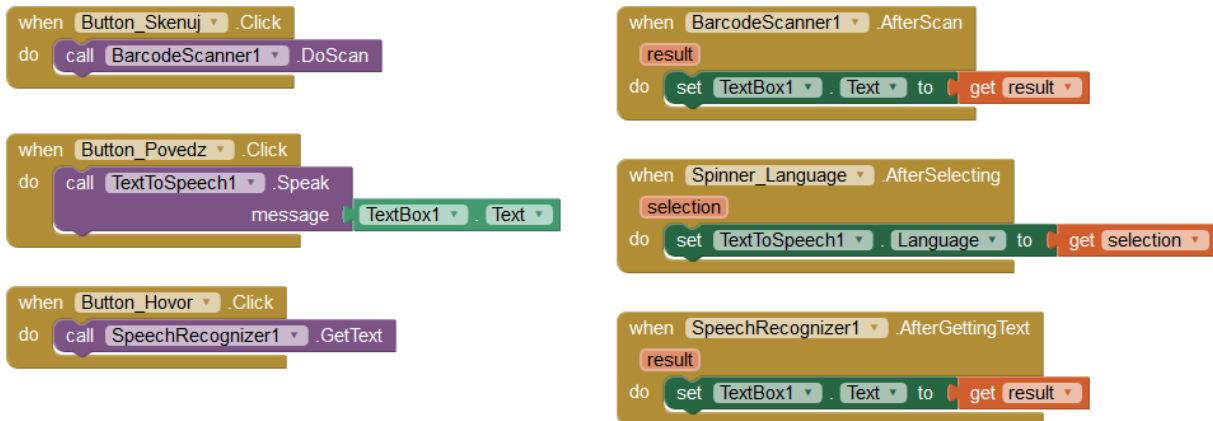
Otázka	Odpoveď
f. Čo sa stane po spustení metódy BarcodeScanner.DoScan ?	f.
g. Čo vyvolalo udalosť BarcodeScanner.AfterScan a kedy?	g.
h. Aký význam v udalosti BarcodeScanner.AfterScan má parameter result ?	h.
i. Čo sa stane, ak v komponente BarcodeScanner zaškrtneme vlastnosť UseExternalScanner ?	i.
j. Čo robí metóda TextToSpeech.Speak ?	j.

Úloha 2: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_6_QR_kod2.aia` a preskúmajte používateľské rozhranie a správanie tejto aplikácie. Svoje zistenia zapíšte do voľných políček tabuľky.

Používateľské rozhranie:



Zdrojový kód:



Otázka	Odpoveď
a. Na čo slúži komponent Spinner ?	a.
b. Aký význam v komponente Spinner majú uvedené vlastností?	b. Prompt ElementsFromStrings Selection
c. Kedy sa vyvolá udalosť Spinner.AfterSelecting ?	c.
d. Aký význam má jej parameter selection ?	d.
e. Aký význam v komponente TextToSpeech má vlastnosť TextToSpeech.Language ?	e.
f. Na čo slúži komponent SpeechRecognizer ?	f.
g. Kedy sa vyvolá udalosť SpeechRecognizer.AfterGettingText ?	g.
h. Aký význam má jej parameter result ?	h.

Úloha 3: Doplňte aplikáciu `pmz_2_6_QR_kod2.aia` o ďalšie tri komponenty **Spinner** na nastavenie **rýchlosti reči**, **výšku hlasu** a **krajiny**, v ktorých nastavíte aspoň 1 ďalší jazyk. Výsledný kód uložte do súboru `pmz_2_6_QR_kod2_R.aia`.

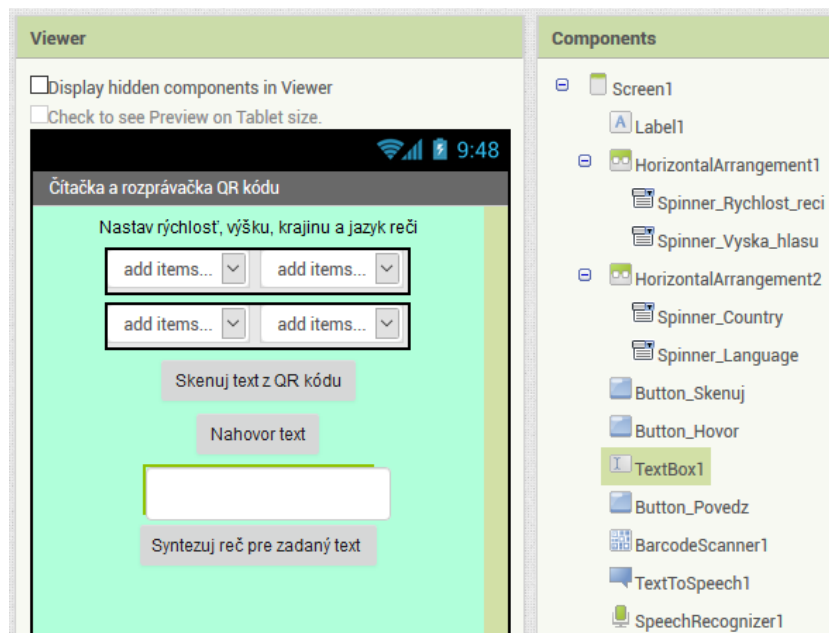
(Odporúčanie: Pre nastavenie krajiny (Country) sa používajú trojpísmenové skratky, napr. SVK, USA, GBR, CZE a pre nastavenie jazyka (Language) sa používajú dvojpísmenové skratky, napr. sk, en, cz. Ďalšie kódy krajín a jazykov sa dajú nájsť na <https://docs.thunkable.com/thunkable-classic-android/create/components/voice/text-to-speech> či <https://cloud.google.com/speech-to-text/docs/languages>)

Riešenie – Programujeme etudu 2.6 Čítačka QR kódu

Úloha 3: Doplňte aplikáciu `pmz_2_6_QR_kod2.aia` o ďalšie tri komponenty **Spinner** na nastavenie **rýchlosti reči, výšku hlasu a krajiny**, v ktorých nastavíte aspoň 1 ďalší jazyk. Výsledný kód uložte do súboru `pmz_2_6_QR_kod2_R.aia`.

(Odporúčanie: Pre nastavenie krajiny (Country) sa používajú trojpísmenové skratky, napr. SVK, USA, GBR, CZE a pre nastavenie jazyka (Language) sa používajú dvojpísmenové skratky, napr. sk, en, cz. Ďalšie kódy krajín a jazykov sa dajú nájsť na <https://docs.thunkable.com/thunkable-classic-android/create/components/voice/text-to-speech> či <https://cloud.google.com/speech-to-text/docs/languages>)

Používateľské rozhranie a zdrojový kód výslednej aplikácie `pmz_2_6_QR_kod2_R.aia`:



<pre> when Button_Skenuj .Click do call BarcodeScanner1 .DoScan </pre>	<pre> when Spinner_Vyska_hlasu .AfterSelecting selection do set TextToSpeech1 .Pitch to get selection </pre>
<pre> when Button_Hovor .Click do call SpeechRecognizer1 .GetText </pre>	<pre> when Spinner_Language .AfterSelecting selection do set TextToSpeech1 .Language to get selection </pre>
<pre> when Button_Povedz .Click do call TextToSpeech1 .Speak message TextBox1 .Text </pre>	<pre> when BarcodeScanner1 .AfterScan result do set TextBox1 .Text to get result </pre>
<pre> when Spinner_Rychlost_reci .AfterSelecting selection do set TextToSpeech1 .SpeechRate to get selection </pre>	<pre> when SpeechRecognizer1 .AfterGettingText result do set TextBox1 .Text to get result </pre>
<pre> when Spinner_Country .AfterSelecting selection do set TextToSpeech1 .Country to get selection </pre>	

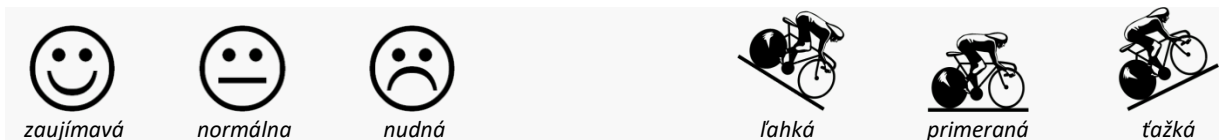
Sebahodnotiaca karta – Programujeme etudu 2.6 Čítačka QR kódu

Zapísaním symbolu ✓ na príslušné miesta tabuliek vyjadrite, do akej miery ovládáte uvedené prvky učiva.

Z uvedeného učiva nasledovným pojmom / princípom / tvrdeniam:	rozumiem	čiastočne rozumiem	vôbec nerozumiem
Nevizuálny komponent BarcodeScanner slúži na skenovanie čiarových kódov			
Pomocou metódy BarcodeScanner.DoScan sa spustí interný alebo externý skener čiarových kódov, čo závisí od zaškrtnutia vlastnosti BarcodeScanner.UseExternalScanner			
Po ukončení skenovania sa vyvolá udalosť BarcodeScanner.AfterScan , ktorá výsledok skenovania má uložený v parametri result			
Nevizuálny komponent TextToSpeech slúži na rečovú syntézu			
Metóda TextToSpeech.Speak syntetizuje reč pre text zadaný v parametri message			
Parametre syntezovanej reči (krajina, jazyk, rýchlosť reči, výška hlasu) sa dajú v komponente TextToSpeech nastaviť pomocou vlastností Country, Language, SpeechRate, Pitch			
Vizuálny komponent Spinner sa používa na vstup údajov z vymenovaného zoznamu prvkov, ktoré sú uvedené (oddelené čiarkou) vo vlastnosti ElementsFromString			
Po výbere hodnoty v komponente Spinner sa vyvolá udalosť Spinner.AfterSelecting , ktorá vybranú hodnotu má uloženú v parametri selection			
Nevizuálny komponent SpeechRecognizer slúži na analýzu reči			
Po rozpoznaní reči sa vyvolá udalosť SpeechRecognizer.AfterGettingText , ktorá má výsledok rozpoznávania uložený v parametri result			

Z uvedeného učiva viem vykonať nasledovné činnosti:	samostatne	s malou pomocou	s veľkou pomocou
Vytvorí aplikáciu na čítanie čiarových kódov, ktorá využíva aj:			
• syntézu reči			
• analýzu reči			
• výberový zoznam (komponent Spinner)			

Aká bola pre vás táto etuda? Zaujímavá? Ľahká? Zafarbite/zakrúžkujte niektorú z uvedených možností:



Uvedte, čo by ste ešte doplnili do tejto mobilnej aplikácie, aby bola viac zaujímavá a využiteľná v praxi:

2.7. Asistent pri cvičení

Prvky nového učiva

Komponenty	Udalosti (premenné)	Metódy	Vlastnosti
ProximitySensor	ProximityChanged (distance)		Enabled
Sound		Vibrate	
CheckBox			Checked
Pedometer	WalkStep (walkSteps, distance)	Start Resume Reset Pause Save	WalkSteps Distance ElapsedTime StrideLength StopDetectionTimeout
AccelerometerSensor			Enabled
TableArrangement			

Jazykové konštrukcie	Iné prvky jazyka
-	-

Pracovné súbory

Meno súboru	Použitie súboru
pmz_2_07_asistent_pri_cviceni_PL.docx	Pracovný list – Programujeme etudu 2.7 Asistent pri cvičení
pmz_2_07_asistent_pri_cviceni_R.docx	Riešenie – Programujeme etudu 2.7 Asistent pri cvičení
pmz_2_07_asistent_pri_cviceni_SK.docx	Sebahodnotiaci karta – Programujeme etudu 2.7 Asistent pri cvičení
pmz_2_7_cviky1.aia	Pracovný súbor pre úlohu 1
pmz_2_7_cviky1_R.aia	Súbor s riešením úlohy 2
pmz_2_7_krokomer1.aia	Pracovný súbor pre úlohu 3
pmz_2_7_krokomer2.aia	Pracovný súbor pre úlohu 4

Komentár k priebehu a výsledkom výučby

Hlavným cieľom tejto etudy je, aby žiak bol schopný vytvoriť aplikáciu použiteľnú ako asistenta pri fyzickom cvičení využívajúcu senzor priblíženia (ProximitySensor) a virtuálny senzor krokomer (Pedometer), vibrácie zariadenia (Sound.Vibrate), výberové políčko (CheckBox) a tabuľkové rozmiestnenie komponentov na obrazovke (TableArrangement).

Žiaci riešia **úlohy z pracovného listu**, ktorý obsahuje:

Úloha 1 – žiaci importujú a inštalujú aplikáciu `pmz_2_7_cviky1.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (ProximitySensor, Sound.Vibrate).

Úloha 2 – žiaci rozširujú aplikáciu `pmz_2_7_cviky1.aia` o ďalšie funkcionality (`ProximitySensor.Enabled`, `AccelerometerSensor.Enabled`, `CheckBox`). Výsledný kód uložia do súboru `pmz_2_7_cviky1_R.aia`.

Úloha 3 – žiaci importujú a inštalujú aplikáciu `pmz_2_7_krokomer1.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (`Pedometer.Reset`, `Start`, `TableArrangement`).

Úloha 4 – žiaci importujú a inštalujú aplikáciu `pmz_2_7_krokomer2.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (`Pedometer.Pause`, `Resume`, `Save`).

Na záver žiaci vyplnia **sebahodnotiacu kartu**, aby si uvedomili a zafixovali nové učivo. Sebahodnotiacu kartu môže učiteľ použiť na stručnú sumarizáciu nového učiva.

Pracovný list – Programujeme etudu 2.7 Asistent pri cvičení

Úloha 1: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_7_cviky1.aia`. Po nainštalovaní a spustení aplikácie na mobilnom zariadení preskúmajte jej správanie. Svoje zistenia zapíšte do voľných políčok tabuliek.

Používateľské rozhranie:



Otázka	Odpoveď
a. V ktorej skupine komponentov je uvedený komponent ProximitySensor ?	a.
b. Na čo sa využíva komponent ProximitySensor ?	b.
c. Na aký účel by ste využili túto aplikáciu?	c.

Zdrojový kód:

```

when ProximitySensor1 .ProximityChanged
  distance
do
  if
  then
    set Screen1 .BackgroundColor to
    call Sound1 .Play
    call Sound1 .Vibrate
      milliseconds 100
  else
    set Screen1 .BackgroundColor to
  
```

```

when AccelerometerSensor1 .Shaking
do
  call Sound2 .Play
  
```

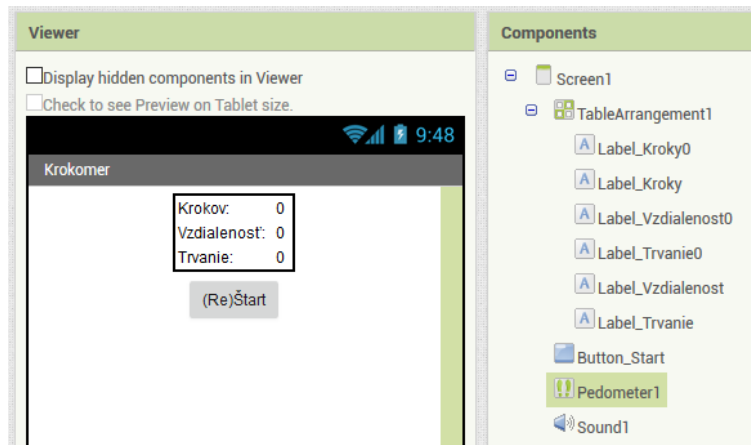
Otázka	Odpoveď
d. Akou činnosťou sa vyvolá udalosť ProximitySensor.ProximityChanged ?	d.
e. Aký význam má parameter distance udalosti ProximitySensor.ProximityChanged ?	e.
f. Akú funkčnosť predstavuje metóda Sound.Vibrate ?	f.

Úloha 2: Doplňte aplikáciu `pmz_2_7_cviky1.aia`, aby zaznamenávala, resetovala a vypisovala počty priblížení a zatrasení mobilného zariadenia a tiež aby umožňovala zapínať a vypínať senzory priblíženia a zrýchlenia. Výsledný kód uložte do súboru `pmz_2_7_cviky1_R.aia`.

(Odporúčanie: Na zapnutie a vypnutie senzorov priblíženia a zrýchlenia použite komponent **CheckBox**, pomocou ktorého sa dajú nastaviť vlastnosti **ProximitySensor.Enabled** a **AccelerometerSensor.Enabled**. Po zaškrtnutí a odškrtnutí komponentu **CheckBox** sa vyvolá udalosť **CheckBox.Changed**, stav komponentu **CheckBox** reprezentuje jeho vlastnosť **Checked**.)

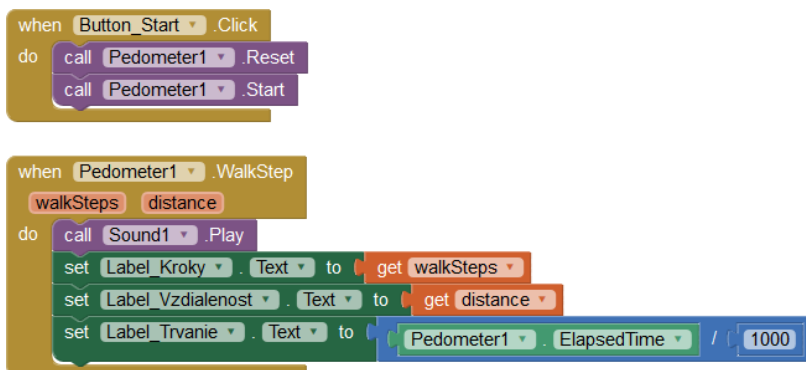
Úloha 3: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_7_krokomer1.aia` a preskúmajte používateľské rozhranie a správanie tejto aplikácie. Svoje zistenia zapíšte do voľných políček tabuľky.

Používateľské rozhranie:



Otázka	Odpoveď
a. V ktorej skupine komponentov je uvedený komponent Pedometer ?	a.
b. Na čo sa využíva komponent Pedometer ?	b.
c. Aké hodnoty a aký význam v komponente Pedometer majú uvedené vlastnosti?	c. <code>StrideLength</code> <code>StopDetectionTimeout</code>

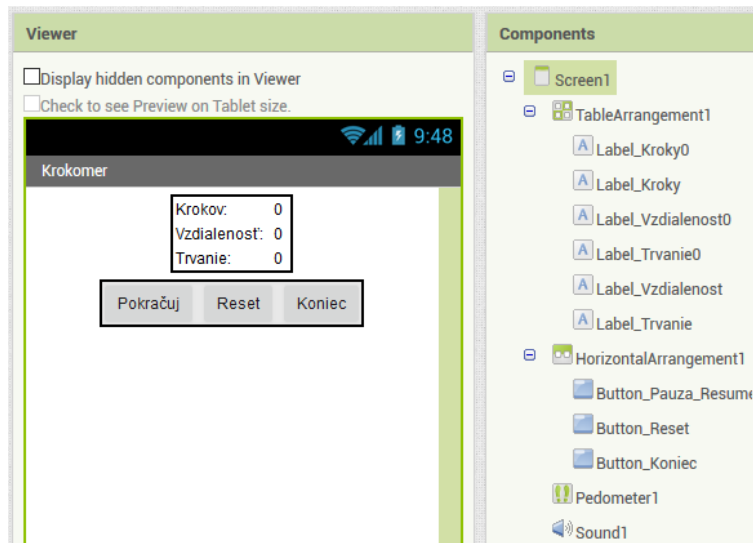
Zdrojový kód:



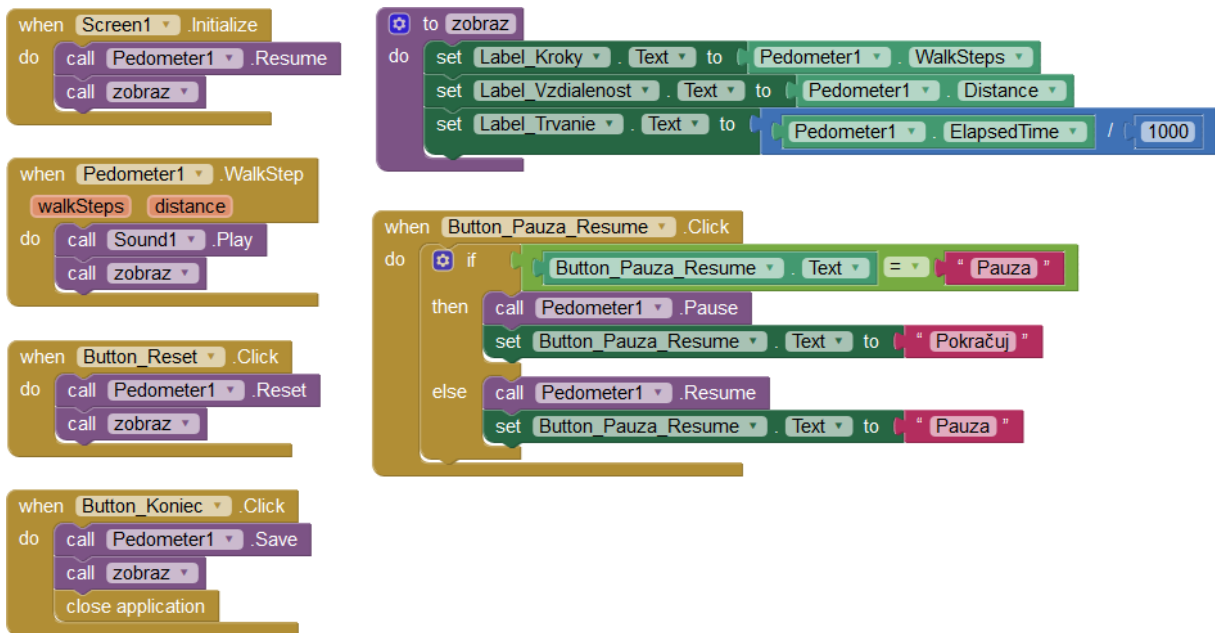
Otázka	Odpoveď
d. Čo robí metóda Pedometer.Reset ?	d.
e. Ako by sa zmenilo správanie programu, ak by sa vynechala metóda Pedometer.Reset ?	e.
f. Čo robí metóda Pedometer.Start ?	f.
g. Akou činnosťou sa vyvolá udalosť Pedometer.WalkStep ?	g.
h. Aký význam v udalosti Pedometer.WalkStep má jej parameter walkSteps ?	h.
i. Aký význam v udalosti Pedometer.WalkStep má jej parameter distance ?	i.
j. Ako by sa zmenilo správanie programu, ak by sa v udalosti Pedometer.WalkStep namiesto parametra walkSteps použila vlastnosť Pedometer.WalkSteps ?	j.
k. Aký význam má vlastnosť Pedometer.ElapsedTime ?	k.

Úloha 4: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_7_krokomer2.aia` a preskúmajte používateľské rozhranie a správanie tejto aplikácie. Svoje zistenia zapíšte do voľných políček tabuľky.

Používateľské rozhranie:



Zdrojový kód:



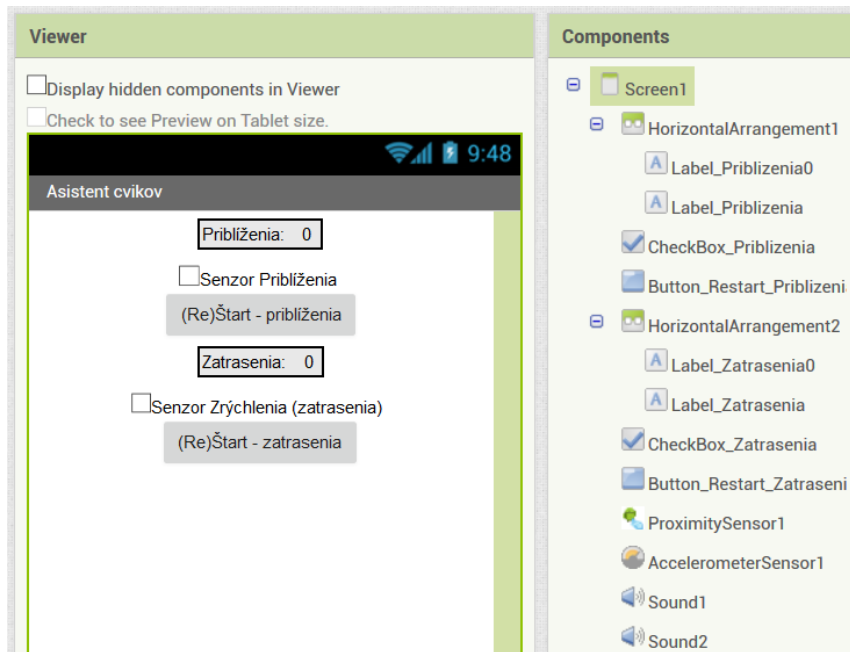
Otázka	Odpoveď
a. Aký význam má metóda Pedometer.Save v súvislosti s opätovným spustením aplikácie?	a.
b. Prečo v programe chýba metóda Pedometer.Start ? Ktorá metóda ju zastupuje?	b.
c. Čo sa deje po stlačení tlačidla Button_Pauza_Resume ?	c.
d. Uvedte, ktoré ďalšie užitočné funkcionality by sa dali doplniť do tejto aplikácie?	d.

Riešenie – Programujeme etudu 2.7 Asistent pri cvičení

Úloha 2: Doplňte aplikáciu `pmz_2_7_cviky1.aia`, aby zaznamenávala, resetovala a vypisovala počty priblížení a zatrasení mobilného zariadenia a tiež aby umožňovala zapínať a vypínať senzory priblíženia a zrýchlenia. Výsledný kód uložte do súboru `pmz_2_7_cviky1_R.aia`.

(Odporúčanie: Na zapnutie a vypnutie senzorov priblíženia a zrýchlenia použite komponent **CheckBox**, pomocou ktorého sa dajú nastaviť vlastnosti **ProximitySensor.Enabled** a **AccelerometerSensor.Enabled**. Po zaškrtnutí a odškrtnutí komponentu **CheckBox** sa vyvolá udalosť **CheckBox.Changed**, stav komponentu **CheckBox** reprezentuje jeho vlastnosť **Checked**.)

Používateľské rozhranie a zdrojový kód výslednej aplikácie `pmz_2_7_cviky1_R.aia`:



```
initialize global priblizeni to 0
```

```
when Screen1.Initialize
do
  set ProximitySensor1.Enabled to false
  set AccelerometerSensor1.Enabled to false
  set CheckBox_Priblizenia.Checked to false
  set CheckBox_Zatrasenia.Checked to false
```

```
when ProximitySensor1.ProximityChanged
  distance
do
  if get distance = 0
  then
    call Sound1.Play
    set global priblizeni to get global priblizeni + 1
    set Label_Priblizenia.Text to get global priblizeni
```

```
when Button_Restart_Priblizenia.Click
do
  set global priblizeni to 0
  set Label_Priblizenia.Text to get global priblizeni
```

```
when CheckBox_Priblizenia.Changed
do
  if CheckBox_Priblizenia.Checked
  then
    set ProximitySensor1.Enabled to true
  else
    set ProximitySensor1.Enabled to false
```

```
initialize global zatraseni to 0
```

```
when AccelerometerSensor1.Shaking
do
  call Sound2.Play
  set global zatraseni to get global zatraseni + 1
  set Label_Zatrasenia.Text to get global zatraseni
```

```
when Button_Restart_Zatrasenia.Click
do
  set global zatraseni to 0
  set Label_Zatrasenia.Text to get global zatraseni
```

```
when CheckBox_Zatrasenia.Changed
do
  if CheckBox_Zatrasenia.Checked
  then
    set AccelerometerSensor1.Enabled to true
  else
    set AccelerometerSensor1.Enabled to false
```

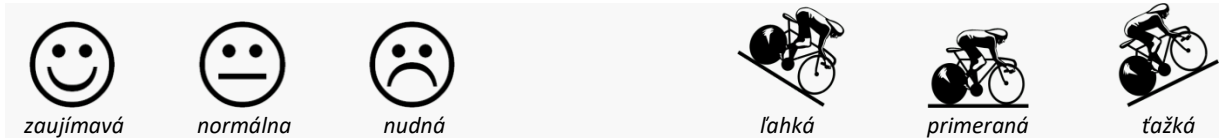
Sebahodnotiaca karta – Programujeme etudu 2.7 Asistent pri cvičení

Zapísaním symbolu ✓ na príslušné miesta tabuliek vyjadrite, do akej miery ovládate uvedené prvky učiva.

Z uvedeného učiva nasledovným pojmom / princípom / tvrdeniam:	rozumiem	častočne rozumiem	vôbec nerozumiem
Nevizuálny komponent ProximitySensor sa používa na registrovanie objektu v blízkosti mobilného zariadenia			
Priblíženie objektu (ucha) k mobilnému zariadeniu (telefónu) vyvolá udalosť ProximitySensor.ProximityChanged , ktorá v parametri distance má uloženú jednu z dvoch hodnôt: blízko (0 cm) a ďaleko (napr. 8 cm)			
Vibrovanie zariadenia vyvoláme spustením metódy Sound.Vibrate			
Vizuálny komponent CheckBox sa používa na zaškrtnutie, resp. odškrtnutie možnosti			
Stav komponentu CheckBox reprezentuje jeho vlastnosť Checked			
Nevizuálny komponent Pedometer sa používa na meranie počtu krokov			
Chôdza s mobilným zariadením vyvolá udalosť Pedometer.WalkStep , ktorá v parametri walkSteps má uložený počet prejdenej krokov a v parametri distance prejdenu vzdialenosť			
Počet prejdenej krokov, resp. prejdenu vzdialenosť sú uložené vo vlastnostiach Pedometer.WalkSteps , resp. Pedometer.Distance			
Čas chôdze je uložený vo vlastnosti Pedometer.ElapsedTime			
Spustenie krokometra sa vykoná pomocou metódy Pedometer.Start , resp. Pedometer.Resume po pozastavení			
Resetovanie hodnôt počtu krokov, prejdenej vzdialenosti a času chôdze sa vykoná pomocou metódy Pedometer.Reset			
Pozastavenie merania počtu krokov a vzdialenosti sa vykoná pomocou metódy Pedometer.Pause			
Uloženie stavu komponentu Pedometer do mobilného zariadenia sa vykoná pomocou metódy Pedometer.Save			
Dĺžka kroku je uložená vo vlastnosti Pedometer.StrideLength (štandardne je nastavená hodnota 0.73 m)			
Čas nečinnosti, po ktorom sa zastaví meranie krokov, je uložený vo vlastnosti Pedometer.StopDetectionTimeout (štandardne je nastavená hodnota 2000 ms)			
Na usporiadanie vizuálnych komponentov do tabuľky sa používa komponent TableArrangement zo skupiny Layout			

Z uvedeného učiva viem vykonať nasledovné činnosti:	samostatne	s malou pomocou	s veľkou pomocou
Vytvoriť aplikáciu registrujúcu pohyb zariadenia využitím:			
• komponentu ProximitySensor			
• komponentu AccelerometerSensor			
• komponentu Pedometer			
• komponentu CheckBox			

Aká bola pre vás táto etuda? Zaujímavá? Ľahká? Zafarbite/zakrúžkujte niektorú z uvedených možností:



Uvedte, čo by ste ešte doplnili do tejto mobilnej aplikácie, aby bola viac zaujímavá a využiteľná v praxi:

2.8. Generátor náhodných viet

Prvky nového učiva

Komponenty	Udalosti (premenné)	Metódy	Vlastnosti
ListView			Elements

Jazykové konštrukcie	Iné prvky jazyka
list (make a list, select list item, length of list, remove list item, insert list item, create empty list, add items to list) FOR EACH NUMER FOR EACH ITEM	-

Pracovné súbory

Meno súboru	Použitie súboru
pmz_2_8_generator_nahodnych_viet_PL.docx	Pracovný list – Programujeme etudu 2.8 Generátor náhodných viet
pmz_2_8_generator_nahodnych_viet_R.docx	Riešenie – Programujeme etudu 2.8 Generátor náhodných viet
pmz_2_8_generator_nahodnych_viet_SK.docx	Sebahodnotiaca karta – Programujeme etudu 2.8 Generátor náhodných viet
pmz_2_8_vety.aia	Pracovný súbor pre úlohu 1
pmz_2_8_vety_R.aia	Súbor s riešením úlohy 2
pmz_2_8_zoznam_cisel.aia	Pracovný súbor pre úlohu 3
pmz_2_8_zoznam_cisel_R.aia	Súbor s riešením úlohy 4

Komentár k priebehu a výsledkom výučby

Hlavným cieľom tejto etudy je, aby žiak bol schopný vytvoriť aplikáciu na spracovanie skupín textových a číselných údajov využívajúcu údajovú štruktúru zoznam (List) s jej metódami (make, select, length, remove, insert, create empty, add), komponent ListView a príkazy opakovania (FOR EACH NUMER, FOR EACH ITEM).

Žiaci riešia **úlohy z pracovného listu**, ktorý obsahuje:

Úloha 1 – žiaci importujú a inštalujú aplikáciu `pmz_2_8_vety.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (make a list, select list item, length of list).

Úloha 2 – žiaci rozširujú aplikáciu `pmz_2_8_vety.aia` o ďalšie funkcionality (remove list item, insert list item). Výsledný kód uložia do súboru `pmz_2_8_vety_R.aia`.

Úloha 3 – žiaci importujú a inštalujú aplikáciu `pmz_2_8_zoznam_cisel.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (ListView.Elements, create empty list, add items to list, FOR EACH NUMBER, FOR EACH ITEM).

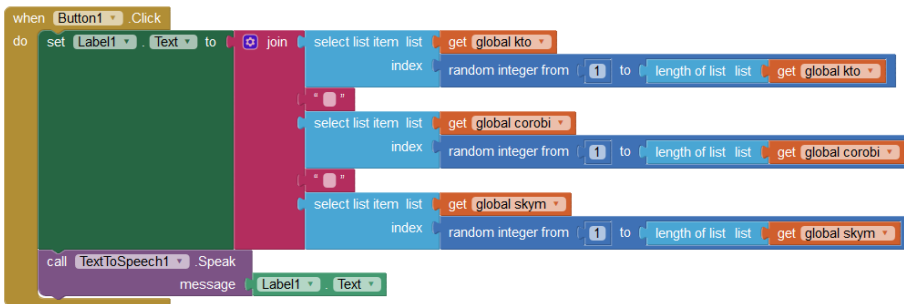
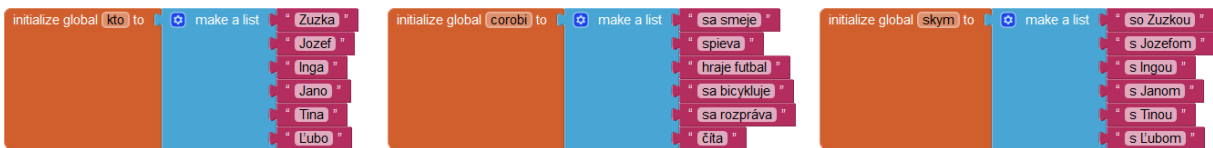
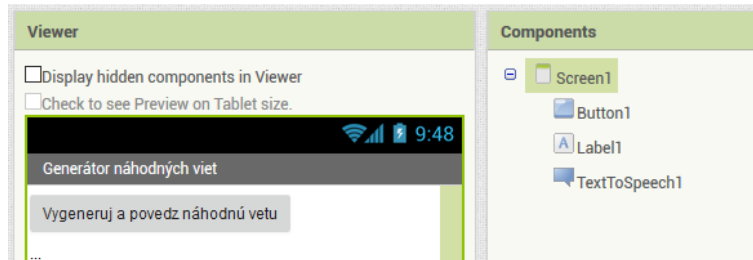
Úloha 4 – žiaci rozširujú aplikáciu `pmz_2_8_zoznam_cisel.aia` o ďalšie funkcionality (výpočet aritmetického priemeru prvkov zoznamu). Výsledný kód uložia do súboru `pmz_2_8_zoznam_cisel_R.aia`.

Na záver žiaci vyplnia **sebahodnotiacu kartu**, aby si uvedomili a zafixovali nové učivo. Sebahodnotiacu kartu môže učiteľ použiť na stručnú sumarizáciu nového učiva.

Pracovný list – Programujeme etudu 2.8 Generátor náhodných viet

Úloha 1: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_8_vety.aia`. Po nainštalovaní a spustení aplikácie na mobilnom zariadení preskúmajte jej správanie. Svoje zistenia zapíšte do voľných políček tabuľky.

Používateľské rozhranie a zdrojový kód aplikácie:



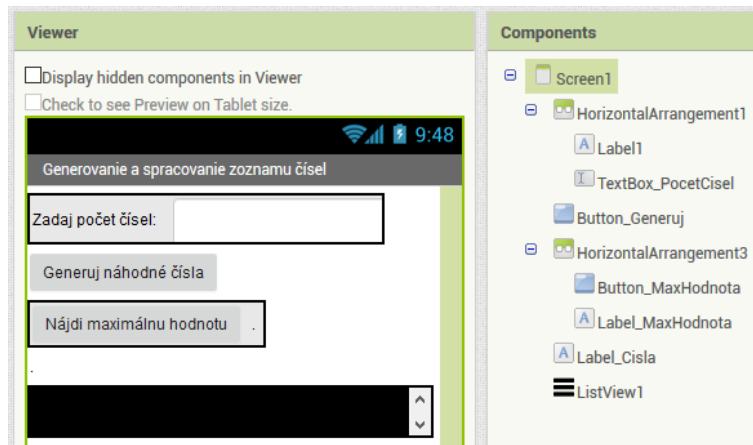
Otázka	Odpoveď
a. Čo robí daná aplikácia?	a.
b. Dáva rovnaké výsledky po opätovnom spustení?	b.
c. Prečo sú v zdrojovom kóde použité premenné typu zoznam ?	c.
d. Aký význam má funkcia make a list ?	d.
e. Aký význam má funkcia select list item ?	e.
f. Aký význam má funkcia length of list ?	f.

Úloha 2: Upravte aplikáciu `pmz_2_8_vety.aia`, aby sa negenerovali dve rovnaké krstné mená v podmete a predmete. Aplikáciu rozšírite o generovanie ďalších vetných členov, napr. prívlastok, príslovkové určenie miest či času. Výsledný kód uložte do súboru `pmz_2_8_vety_R.aia`.

(Odporúčanie: Pri riešení tohto problému môžeme vhodne použiť odobranie prvku zoznamu pomocou funkcie **remove list item** a vloženie prvku do zoznamu pomocou funkcie **insert list item**.)

Úloha 3: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore pmz_2_8_zoznam_cisel.aia a preskúmajte používateľské rozhranie a správanie tejto aplikácie. Svoje zistenia zapíšte do voľných políček tabuľky.

Používateľské rozhranie:



Otázka	Odpoveď
a. V ktorej skupine komponentov je uvedený komponent ListView ?	a.
b. Na čo sa využíva komponent ListView ?	b.

Zdrojový kód:

```

initialize global cisla to create empty list

when Button_Generuj .Click
do
  for each number from 1
  to TextBox_PocetCisel .Text
  by 1
  do
    add items to list list
    item 50 + 10 * random integer from 0 to 5
  call ZobrazZoznam

when Button_MaxHodnota .Click
do
  set Label_MaxHodnota .Text to call NajdiMaxHodnotu

to ZobrazZoznam
do
  set Label_Cisla .Text to get global cisla
  set ListView1 .Elements to get global cisla

to NajdiMaxHodnotu
result
  initialize local MaxHodnota to select list item list
  index 1
  in
  do
    for each item in list
    do
      if get item > get MaxHodnota
      then set MaxHodnota to get item
    result get MaxHodnota
  
```

Otázka	Odpoveď
c. Čo vracia funkcia create empty list ?	c.
d. Čo sa nastavuje pomocou vlastnosti ListView.Elements ?	d.
e. Čo vracia funkcia add items to list ?	e.
f. Ako sa líšia od seba cykly for each number a for each item in list ?	f. for each number for each item in list

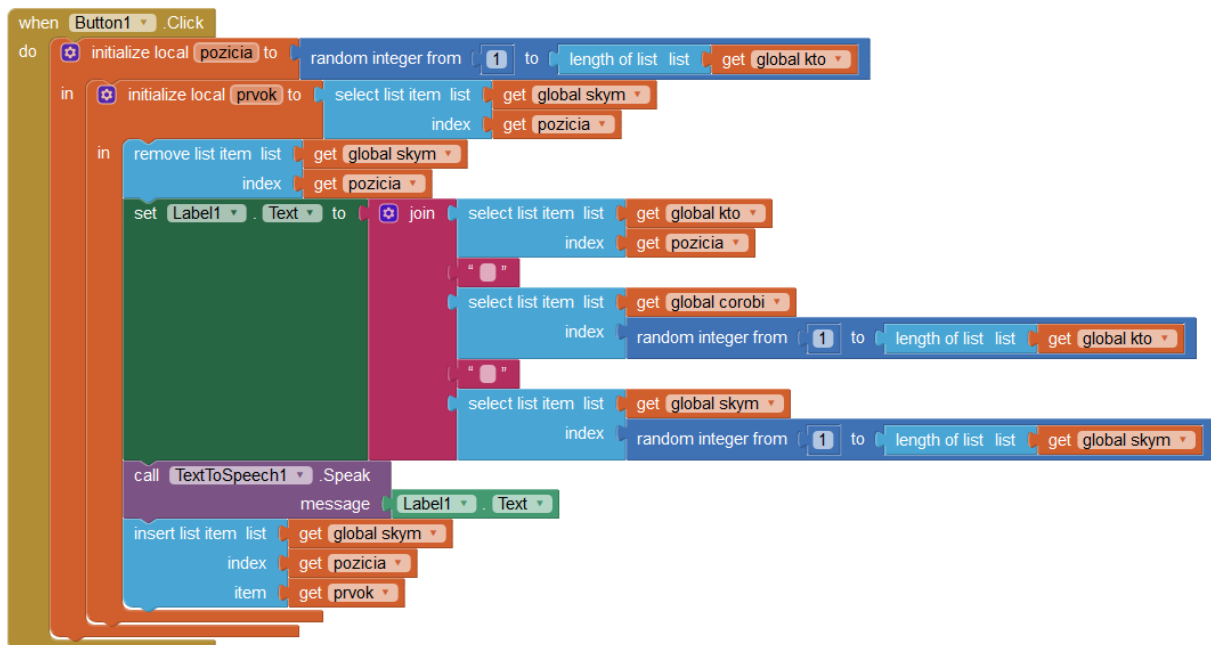
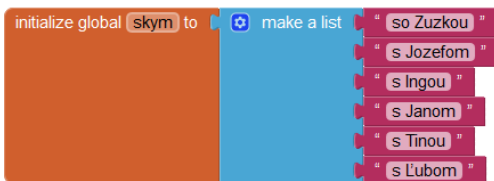
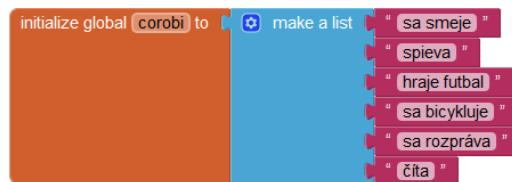
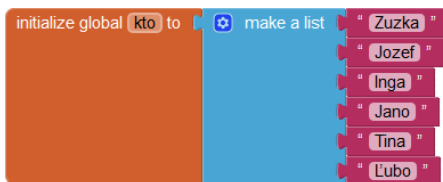
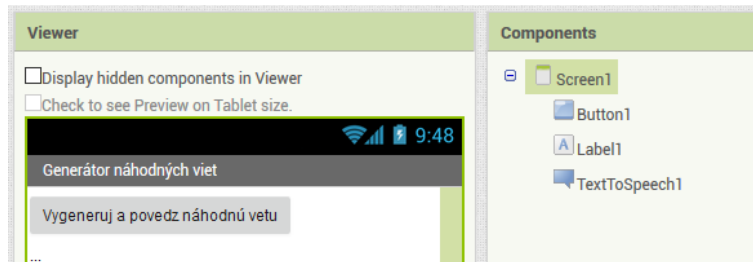
Úloha 4: Doplňte aplikáciu `pmz_2_8_zoznam_cisel.aia` o tlačidlo, ktoré by **vyprázdnilo zoznam** čísel a tlačidlo, ktoré by spustilo **výpočet priemernej hodnoty** zadaného zoznamu. Výsledný kód uložte do súboru `pmz_2_8_zoznam_cisel_R.aia`.

Riešenie – Programujeme etudu 2.8 Generátor náhodných viet

Úloha 2: Upravte aplikáciu `pmz_2_8_vety.aia`, aby sa negenerovali dve rovnaké krstné mená v podmete a predmete. Aplikáciu rozšírite o generovanie ďalších vetných členov, napr. prívlastok, príslovkové určenie miest či času. Výsledný kód uložte do súboru `pmz_2_8_vety_R.aia`.

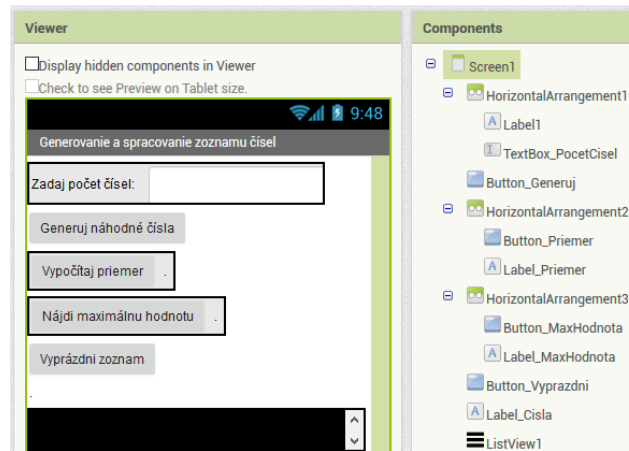
(Odporúčanie: Pri riešení tohto problému môžeme vhodne použiť odobranie prvku zo zoznamu pomocou funkcie **remove list item** a vloženie prvku do zoznamu pomocou funkcie **insert list item**.)

Používateľské rozhranie a zdrojový kód výslednej aplikácie `pmz_2_8_vety_R.aia`:



Úloha 4: Doplňte aplikáciu `pmz_2_8_zoznam_cisel.aia` o tlačidlo, ktoré by vyprázdnilo zoznam čísel a tlačidlo, ktoré by spustilo výpočet priemernej hodnoty zadaného zoznamu. Výsledný kód uložte do súboru `pmz_2_8_zoznam_cisel_R.aia`.

Používateľské rozhranie a zdrojový kód výslednej aplikácie `pmz_2_8_vety_R.aia`:



```

when Button_MaxHodnota.Click
do
  set Label_MaxHodnota.Text to call NajdiMaxHodnotu

when Button_Priemer.Click
do
  set Label_Priemer.Text to call VypocitajPriemer

when Button_Generuj.Click
do
  for each number from 1
  to TextBox_PocetCisel.Text
  by 1
  do
    add items to list list
    item 50 + 10 * random integer from 0 to 5
  call ZobrazZoznam

to ZobrazZoznam
do
  set Label_Cisla.Text to get global cisla
  set ListView1.Elements to get global cisla

initialize global cisla to create empty list

when Button_Vyprazdni.Click
do
  set global cisla to create empty list
  call ZobrazZoznam

to NajdiMaxHodnotu
result initialize local MaxHodnota to select list item list
index 1
in do
  for each item in list
  do
    if get item > get MaxHodnota
    then set MaxHodnota to get item
  result get MaxHodnota

to VypocitajPriemer
result initialize local sucet to 0
in do
  for each number from 1
  to length of list list
  by 1
  do
    set sucet to get sucet + select list item list
    index number
  result get sucet / length of list list
  
```

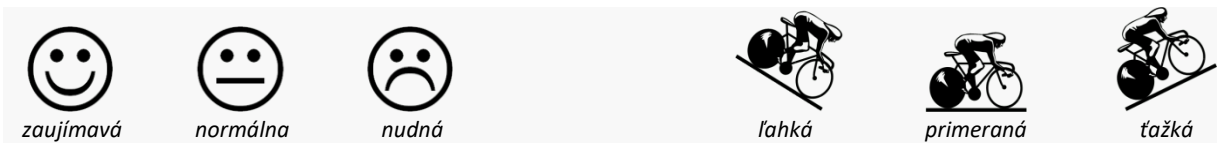

Sebahodnotiaca karta – Programujeme etudu 2.8 Generátor náhodných viet

Zapísaním symbolu ✓ na príslušné miesta tabuliek vyjadrite, do akej miery ovládate uvedené prvky učiva.

Z uvedeného učiva nasledovným pojmom / princípom / tvrdeniam:	rozumiem	častočne rozumiem	vôbec nerozumiem
Zoznam (List) je štruktúrovaný údajový typ na spracovanie skupiny údajov			
Zoznam sa vytvára pomocou funkcie make a list			
Prvok zoznamu sa sprístupňuje funkciou select list item			
Dĺžku zoznamu vracia funkcia length of list			
Prvok sa odoberá zo zoznamu pomocou funkcie remove list item			
Prvok sa vkladá do zoznamu pomocou funkcie insert list item			
Na zobrazenie prvkov zoznamu sa používa komponent ListView zo skupiny komponentov User Interface			
Zoznam sa zobrazí v komponente ListView nastavením jej vlastnosti ListView.Elements			
Prázdny zoznam sa vytvorí pomocou funkcie create empty list			
Prvok sa pridá na koniec zoznamu pomocou funkcie add items to list			
Na spracovanie zoznamu sa používajú cykly for each number a for each item			

Z uvedeného učiva viem vykonať nasledovné činnosti:	samostatne	s malou pomocou	s veľkou pomocou
Vytvoriť aplikáciu na spracovanie zoznamov využívajúcu:			
• výber prvkov zoznamu podľa ich indexu			
• postupný výber prvkov zoznamu			
• pridávanie a odstraňovanie prvkov zoznamu			
• funkciu length of list			
• komponent ListView na zobrazenie prvkov zoznamu			

Aká bola pre vás táto etuda? Zaujímavá? Ľahká? Zafarbite/zakrúžkujte niektorú z uvedených možností:



Uveďte, čo by ste ešte doplnili do tejto mobilnej aplikácie, aby bola viac zaujímavá a využiteľná v praxi:

2.9. Zobrazovač aktuálnej polohy

Prvky nového učiva

Komponenty	Udalosti (premenné)	Metódy	Vlastnosti
LocationSensor	LocationChanged (latitude, longitude, altitude, speed) StatusChanged (status)	LatitudeFromAddress LongitudeFromAddress	ProviderName HasAccuracy Accuracy CurrentAddress TimeInterval DistanceInterval Enabled
Map			

Jazykové konštrukcie	Iné prvky jazyka
-	-

Pracovné súbory

Meno súboru	Použitie súboru
pmz_2_9_zobrazovac_polohy_PL.docx	Pracovný list – Programujeme etudu 2.9 Zobrazovač aktuálnej polohy
pmz_2_9_zobrazovac_polohy_R.docx	Riešenie – Programujeme etudu 2.9 Zobrazovač aktuálnej polohy
pmz_2_9_zobrazovac_polohy_SK.docx	Sebahodnotiaca karta – Programujeme etudu 2.9 Zobrazovač aktuálnej polohy
pmz_2_9_gps.aia	Pracovný súbor pre úlohu 1
pmz_2_9_gps_R.aia	Súbor s riešením úlohy 2

Komentár k priebehu a výsledkom výučby

Hlavným cieľom tejto etudy je, aby žiak bol schopný vytvoriť aplikáciu na lokalizáciu zemepisnej polohy mobilného zariadenia (LocationSensor) a jej zobrazenie (Map, Label).

Žiaci riešia **úlohy z pracovného listu**, ktorý obsahuje:

Úloha 1 – žiaci importujú a inštalujú aplikáciu `pmz_2_9_gps.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (Map, LocationSensor.LocationChanged, latitude, longitude, CurrentAddress, LatitudeFromAddress, LongitudeFromAddress).

Úloha 2 – žiaci rozširujú aplikáciu `pmz_2_9_gps.aia` o ďalšie funkcionality (ProviderName, HasAccuracy, Accuracy, DistanceInterval, TimeInterval, altitude, speed, Enabled). Výsledný kód uložia do súboru `pmz_2_9_gps_R.aia`.

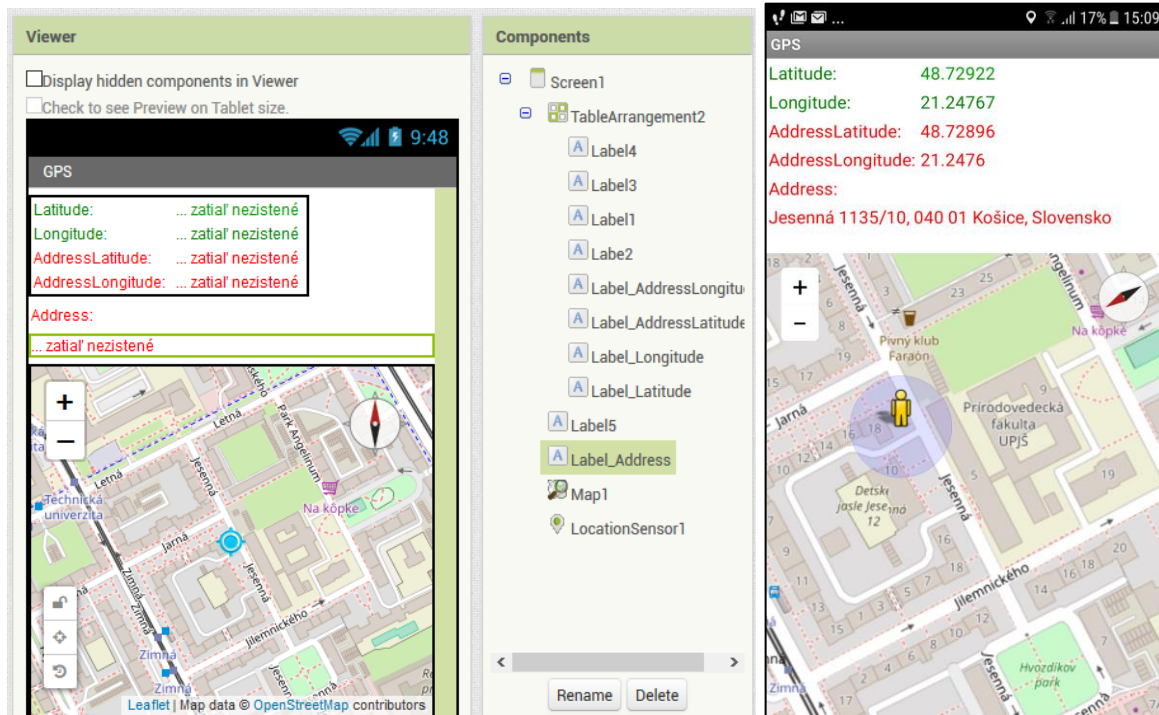
Na záver žiaci vyplnia **sebahodnotiacu kartu**, aby si uvedomili a zafixovali nové učivo. Sebahodnotiacu kartu môže učiteľ použiť na stručnú sumarizáciu nového učiva.

Pracovný list – Programujeme etudu 2.9 Zobrazovač aktuálnej polohy

Úloha 1: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_9_gps.aia`. Po nainštalovaní a spustení aplikácie na mobilnom zariadení preskúmajte jej správanie. Svoje zistenia zapíšte do voľných políčok tabuliek.

(Odporúčanie: V režime Designer v komponente **Map** nastavte svoju polohu zoomovaním a posúvaním mapy a tiež pomocou vlastnosti **ShowUser** a **CenterFromString**. Ak máte svoju polohu uprostred mapy, zafixujte toto zobrazenie v komponente Map stlačením tlačidla **Set initial map to current view**)

Používateľské rozhranie a kópia obrazovky bežajúcej aplikácie:



Otázka	Odpoveď
a. V ktorej skupine komponentov je uvedený komponent LocationSensor ?	a.
b. Na čo sa využíva komponent LocationSensor ?	b.
c. V ktorej skupine komponentov je uvedený komponent Map ?	c.
d. Na čo sa využíva komponent Map ?	d.
e. Ako sa líšia zemepisné súradnice vášho obľúbeného miesta merané našou aplikáciou a inou aplikáciou?	e. Naša apka: zem. šírka zem. dĺžka Iná apka: zem. šírka zem. dĺžka
f. Ak predpokladáme, že Zem je guľa s dĺžkou poludníka 40000 km. Aký dĺhý je úsek poludníka odpovedajúci nasledovným uhlom?	f. 1° ≈ m 0.001° ≈ m 0.1° ≈ m 0.0001° ≈ m 0.01° ≈ m 0.00001° ≈

Zdrojový kód:

```

when LocationSensor1 . LocationChanged
  latitude longitude altitude speed
do
  set Label_Latitude . Text to get latitude
  set Label_Longitude . Text to get longitude
  set Label_Address . Text to LocationSensor1 . CurrentAddress
  set Label_AddressLatitude . Text to call LocationSensor1 . LatitudeFromAddress
    locationName LocationSensor1 . CurrentAddress
  set Label_AddressLongitude . Text to call LocationSensor1 . LongitudeFromAddress
    locationName LocationSensor1 . CurrentAddress
  
```

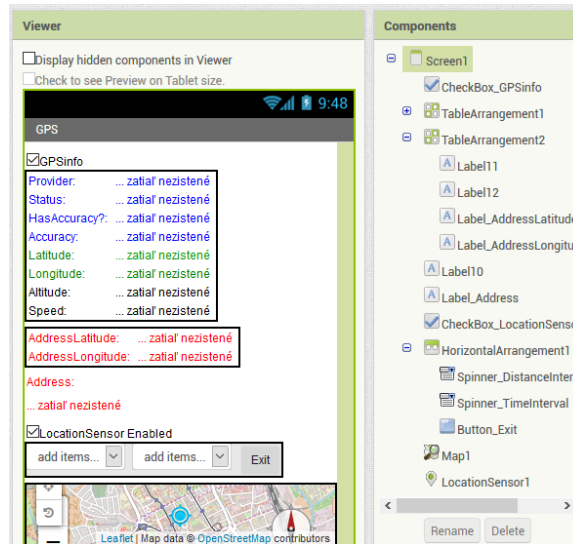
Otázka	Odpoveď
g. Akou aktivitou používateľa sa vyvolá udalosť LocationSensor.LocationChanged ?	g.
h. Aký význam v tejto udalosti majú jej parametre latitude a longitude ?	h. latitude longitude
i. Aký význam v komponente LocationSensor má vlastnosť LocationSensor.CurrentAddress ?	i.
j. Čo vrátia metódy LocationSensor.LatitudeFromAddress a LocationSensor.LongitudeFromAddress ?	j.

Úloha 2: Rozšírite aplikáciu `pmz_2_9_gps.aia`, aby zobrazovala aj ďalšie vlastnosti komponentu LocationSensor, napr. **ProviderName**, **HasAccuracy**, **Accuracy**. Skúmajte ako sa hodnoty týchto vlastností menia v závislosti od spôsobu lokalizácie (Vysoká presnosť, Šetrenie batérie, Iba telefón). Doplnite tiež nastavenie vlastností **DistanceInterval** (s hodnotami 0, 1, 10, 100) a **TimeInterval** (s hodnotami 0, 1000, 10000, 60000, 300000). Zabezpečte tiež zapínanie/vypínanie komponentu LocationSensor a zobrazovanie/nezobrazovanie časti údajov. Výsledný kód uložte do súboru `pmz_2_9_gps_R.aia`.

Riešenie – Programujeme etudu 2.9 Zobrazovač aktuálnej polohy

Úloha 2: Rozšírite aplikáciu `pmz_2_9_gps.aia`, aby zobrazovala aj ďalšie vlastnosti komponentu `LocationSensor`, napr. **ProviderName**, **HasAccuracy**, **Accuracy**. Skúmajte ako sa hodnoty týchto vlastností menia v závislosti od spôsobu lokalizácie (Vysoká presnosť, Šetrenie batérie, Iba telefón). Doplňte tiež nastavenie vlastností **DistanceInterval** (s hodnotami 0, 1, 10, 100) a **TimeInterval** (s hodnotami 0, 1000, 10000, 60000, 300000). Zabezpečte tiež zapínanie/vypínanie komponentu `LocationSensor` a zobrazovanie/nezobrazovanie časti údajov. Výsledný kód uložte do súboru `pmz_2_9_gps_R.aia`.

Používateľské rozhranie a zdrojový kód výslednej aplikácie `pmz_2_9_gps_R.aia`:



```

when LocationSensor1 . LocationChanged
latitude longitude altitude speed
do
  set Label Provider . Text to LocationSensor1 . ProviderName
  set Label HasAccuracy . Text to LocationSensor1 . HasAccuracy
  set Label Accuracy . Text to LocationSensor1 . Accuracy
  set Label Latitude . Text to get latitude
  set Label Longitude . Text to get longitude
  set Label Altitude . Text to get altitude
  set Label Speed . Text to get speed
  set Label Address . Text to LocationSensor1 . CurrentAddress
  set Label AddressLatitude . Text to call LocationSensor1 . LatitudeFromAddress
  set Label AddressLongitude . Text to call LocationSensor1 . LongitudeFromAddress

when CheckBox_LocationSensorEnabled . Changed
do
  if CheckBox_LocationSensorEnabled . Checked
  then set LocationSensor1 . Enabled to true
  else set LocationSensor1 . Enabled to false

when LocationSensor1 . StatusChanged
provider status
do
  set Label Status . Text to get status
  set Label Provider . Text to LocationSensor1 . ProviderName
  set Label HasAccuracy . Text to LocationSensor1 . HasAccuracy
  set Label Accuracy . Text to LocationSensor1 . Accuracy

when CheckBox_GPSInfo . Changed
do
  if CheckBox_GPSInfo . Checked
  then set TableArrangement1 . Visible to true
  else set TableArrangement1 . Visible to false

when Spinner_DistanceInterval . AfterSelecting
selection
do set LocationSensor1 . DistanceInterval to get selection

when Spinner_TimeInterval . AfterSelecting
selection
do set LocationSensor1 . TimeInterval to get selection

when Button_Exit . Click
do close application
  
```

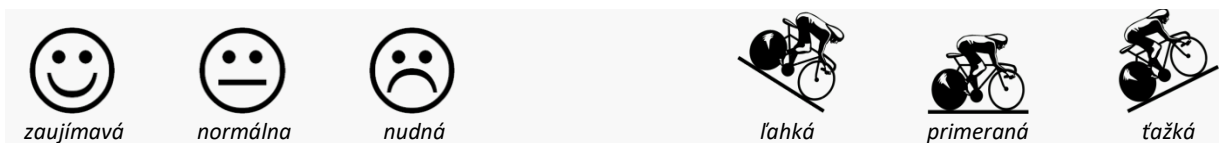
Sebahodnotiaca karta – Programujeme etudu 2.9 Zobrazovač aktuálnej polohy

Zapísaním symbolu ✓ na príslušné miesta tabuliek vyjadrite, do akej miery ovládate uvedené prvky učiva.

Z uvedeného učiva nasledovným pojmom / princípom / tvrdeniam:	rozumiem	čiastočne rozumiem	vôbec nerozumiem
Viditeľný komponent Map (uvedený v skupine Maps) sa používa na zobrazenie mapy vybraného miesta			
Nevizuálny komponent LocationSensor (uvedený v skupine Sensors) sa používa na určenie zemepisnej polohy zariadenia			
Zmena zemepisnej polohy zariadenia vyvolá udalosť LocationSensor.LocationChanged , ktorá má v parametroch latitude a longitude uloženú aktuálnu zemepisnú šírku a dĺžku			
Zapínanie a vypínanie komponentu LocationSensor umožňuje nastavenie vlastnosti LocationSensor.Enabled			
Pri zmene spôsobu lokalizácie a dostupnosti poskytovateľa GPS polohy sa vyvolá udalosť LocationSensor.StatusChanged			
Meno poskytovateľa GPS polohy (gps, network, passive) je uložené vo vlastnosti LocationSensor.ProviderName			
Presnosť určenia GPS polohy v metroch je uložená vo vlastnosti LocationSensor.Accuracy			
Ak má aktuálna pozícia zariadenia priradenú fyzickú adresu, tá bude uložená vo vlastnosti LocationSensor.CurrentAddress , inak je výsledkom správa "No address available"			
Zo zadanej CurrentAddress dostaneme zemepisnú pozíciu pomocou metód LocationSensor.LatitudeFromAddress a LocationSensor.LongitudeFromAddress			
Vyvolanie udalosti LocationSensor.LocationChanged určujú vlastnosti LocationSensor.DistanceInterval (prejdená vzdialenosť v metroch, napr. 5) a LocationSensor.TimeInterval (uplynutý čas v ms, napr. 10000)			

Z uvedeného učiva viem vykonať nasledovné činnosti:	samostatne	s malou pomocou	s veľkou pomocou
Vytvoriť aplikáciu zobrazujúcu aktuálnu GPS polohu zariadenia			
• bez komponentu Map			
• s komponentom Map			

Aká bola pre vás táto etuda? Zaujímavá? Ľahká? Zafarbite/zakrúžkujte niektorú z uvedených možností:



Uveďte, čo by ste ešte doplnili do tejto mobilnej aplikácie, aby bola viac zaujímavá a využiteľná v praxi:

2.10. Asistent aktuálnej polohy

Prvky nového učiva

Komponenty	Udalosti (premenné)	Metódy	Vlastnosti
ActivityStarter		StartActivity	Action DataUri Extras
ListPicker	BeforePicking AfterPicking		Elements Selection
TinyDB		GetTags ClearAll	
HorizontalArrangement			Visible

Jazykové konštrukcie	Iné prvky jazyka
-	pick a random item

Pracovné súbory

Meno súboru	Použitie súboru
pmz_2_10_asistent_polohy_PL.docx	Pracovný list – Programujeme etudu 2.10 Asistent aktuálnej polohy
pmz_2_10_asistent_polohy_R.docx	Riešenie – Programujeme etudu 2.10 Asistent aktuálnej polohy
pmz_2_10_asistent_polohy_SK.docx	Sebahodnotiaca karta – Programujeme etudu 2.10 Asistent aktuálnej polohy
pmz_2_10_astarter_mapy.aia	Pracovný súbor pre úlohu 1
pmz_2_10_astarter_mapy_R.aia	Súbor s riešením úlohy 2
pmz_2_10_astarter_rozne.aia	Pracovný súbor pre úlohu 3

Komentár k priebehu a výsledkom výučby

Hlavným cieľom tejto etudy je, aby žiak bol schopný vytvoriť aplikáciu spúšťajúcu iné externé aplikácie inštalované na mobilnom zariadení (zobrazovač máp, prehrávač videí, e-mailový klient, správca kontaktov) využívajúcu komponent ActivityStarter a iné komponenty (napr. LocationSensor, ListPicker, TinyDB).

Žiaci riešia **úlohy z pracovného listu**, ktorý obsahuje:

Úloha 1 – žiaci importujú a inštalujú aplikáciu `pmz_2_10_astarter_mapy.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (ActivityStarter.Action, DataUri; ListPicker).

Úloha 2 – žiaci rozširujú aplikáciu `pmz_2_10_astarter_mapy.aia` o ďalšie funkcionality (pridávanie/mazanie miest do/zo zoznamu a ich uchovávanie v lokálnej databáze TinyDB). Výsledný kód uložia do súboru `pmz_2_10_astarter_mapy_R.aia`.

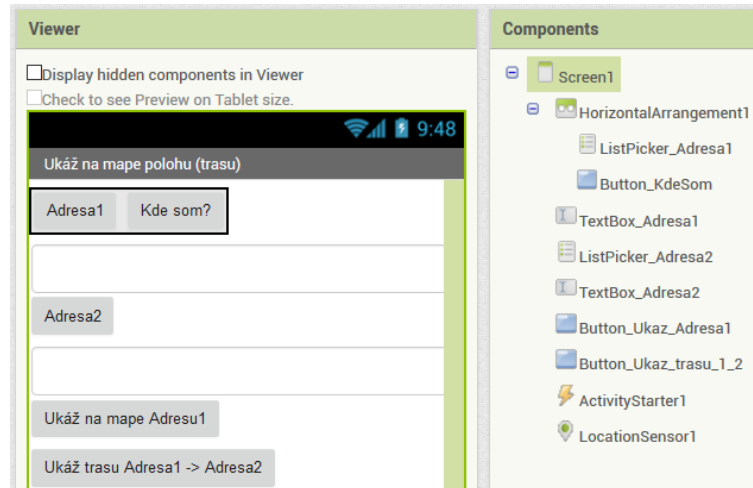
Úloha 3 – žiaci importujú a inštalujú aplikáciu `pmz_2_10_astarter_rozne.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (rôzne parametre Action, DataUri pri spúšťaní externých aplikácií – YouTube prehrávač, e-mailový klient, správca kontaktov).

Na záver žiaci vyplnia **sebahodnotiacu kartu**, aby si uvedomili a zafixovali nové učivo. Sebahodnotiacu kartu môže učiteľ použiť na stručnú sumarizáciu nového učiva.

Pracovný list – Programujeme etudu 2.10 Asistent aktuálnej polohy

Úloha 1: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_10_astarter_mapy.aia`. Po nainštalovaní a spustení aplikácie na mobilnom zariadení preskúmajte jej správanie. Svoje zistenia zapíšte do voľných políček tabuliek.

Používateľské rozhranie:



Otázka	Odpoveď
a. Do ktorej skupiny komponentov patrí komponent ActivityStarter ?	a.
b. Spustenie ktorých externých aplikácií spôsobil komponent ActivityStarter po stlačení tlačidiel Ukáž na mape ... a Ukáž trasu ... ?	b.
c. Do ktorej skupiny komponentov patrí komponent ListPicker ?	c.
d. Ako sa správajú dva komponenty ListPicker po stlačení tlačidiel Adresa1 a Adresa2 vo vzťahu k hodnotám dvoch komponentov TextBox ?	d.

Zdrojový kód:

```

initialize global miesta to make a list
  "Prírodovedecká fakulta UPJŠ Košice "
  "Fakulta prírodných vied UKF Nitra "
  "FMFI UK Mlynská dolina Bratislava "
  "Fakulta prírodných vied UMB Banská Bystrica "
  "Fakulta riadenia a informatiky Žilinská Univerzita "

when Screen1 Initialize
do
  set TextBox_Adresa1 .Text to "Prírodovedecká fakulta UPJŠ Košice "
  set TextBox_Adresa2 .Text to "Fakulta prírodných vied UKF Nitra "

when ListPicker_Adresa1 BeforePicking
do
  set ListPicker_Adresa1 .Elements to get global miesta

when ListPicker_Adresa1 AfterPicking
do
  set TextBox_Adresa1 .Text to ListPicker_Adresa1 .Selection

when ListPicker_Adresa2 BeforePicking
do
  set ListPicker_Adresa2 .Elements to get global miesta

when ListPicker_Adresa2 AfterPicking
do
  set TextBox_Adresa2 .Text to ListPicker_Adresa2 .Selection

when Button_Ukaz_Adresa1 Click
do
  set ActivityStarter1 .Action to "android.intent.action.VIEW "
  set ActivityStarter1 .DataUri to join "geo:0,0?q="
  TextBox_Adresa1 .Text
  call ActivityStarter1 .StartActivity

when Button_Ukaz_trasu_1_2 Click
do
  set ActivityStarter1 .Action to "android.intent.action.VIEW "
  set ActivityStarter1 .DataUri to join "http://maps.google.com/maps?saddr="
  TextBox_Adresa1 .Text
  "&daddr="
  TextBox_Adresa2 .Text
  call ActivityStarter1 .StartActivity

when Button_KdeSom Click
do
  set TextBox_Adresa1 .Text to join LocationSensor1 .Latitude
  " "
  LocationSensor1 .Longitude
  
```

Otázka	Odpoveď
e. Čím sa líšia udalosti ListPicker.BeforePicking a ListPicker.AfterPicking ?	e.
f. Aký význam majú uvedené vlastnosti komponentu ListPicker ?	f. Elements Selection
g. Ktoré vlastnosti komponentu ActivityStarter v tomto kóde sa nastavujú pred volaním metódy ActivityStarter.StartActivity ?	g.
h. Pre zobrazenie miesta, resp. trasy na mape sa vo vlastnosti ActivityStarter.DataUri používa schéma geo:0,0?q=adresa1 , resp. http://maps.google.com/maps?saddr=adresa1&daddr=adresa2 . Čo by sa stalo, ak by sme v parametroch adresa1 a adresa2 zadali GPS súradnice namiesto textovej adresy?	h.

Úloha 2: Rozšírte aplikáciu `pmz_2_10_astarter_mapy.aia` o ďalšie funkcionality, napr.:

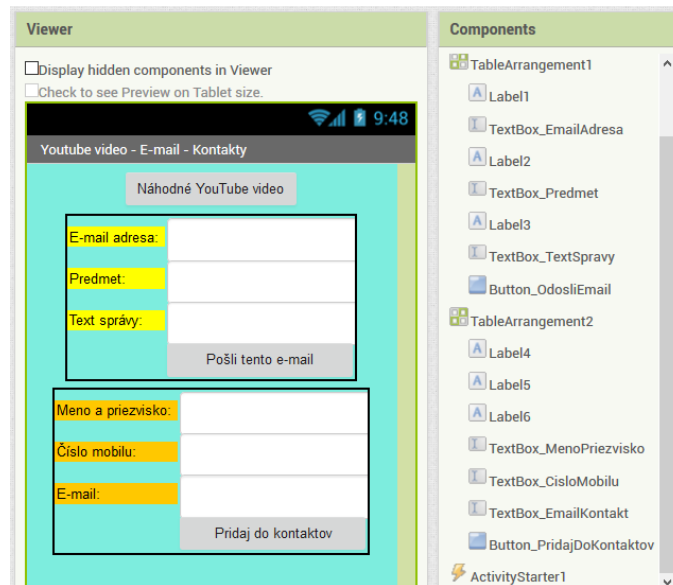
- zobrazenie textovej adresy aktuálnej polohy, ukončenie aplikácie,
- pridanie Adresy1 na koniec zoznamu adries, zmazanie Adresy1 zo zoznamu adries, zmazanie celého zoznamu,
- uloženie zoznamu adries do lokálnej databázy, zmazanie všetkých údajov z databázy.

Výsledný kód uložte do súboru `pmz_2_10_astarter_mapy_R.aia`.

(Poznámka: Inšpirácie pre ďalšie rozšírenia tejto a návrhy ďalších aplikácií využívajúce komponent `ActivityStarter` nájdete na webe, napr. <http://appinventor.mit.edu/explore/ai2/activity-starter.html>)

Úloha 3: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_10_astarter_rozne.aia` a preskúmajte používateľské rozhranie a správanie tejto aplikácie. Svoje zistenia zapíšte do voľných políчков tabuľky.

Používateľské rozhranie a zdrojový kód:



```

initialize global videa to make a list
  * YSkHlv7GhOM
  * Mv0SqWwy02Xs
  * jPoLeJJsbCw
  * qx8hrhBZJ98
  * vC9Qh709gas
  
```

```

when Button_YouTube Click
do
set ActivityStarter1 Action to android.intent.action.VIEW
set ActivityStarter1 DataUri to join vnd.youtube: pick a random item list get global videa
call ActivityStarter1 StartActivity
  
```

```

when Button_OdosliEmail Click
do
set ActivityStarter1 Action to android.intent.action.VIEW
set ActivityStarter1 DataUri to join mailto:
  TextBox_EmailAdresa Text
  ?subject=
  TextBox_Predmet Text
  &body=
  TextBox_TextSpravy Text
  \n\n zaslané z Ai2
call ActivityStarter1 StartActivity
  
```

```

when Button_PridajDoKontaktov Click
do
set ActivityStarter1 Action to android.intent.action.INSERT
set ActivityStarter1 DataUri to content://contacts/people/
set ActivityStarter1 Extras to
  make a list
  make a list phone
  make a list name
  make a list phone_type
  make a list mobile
  make a list email
  TextBox_CisloMobilu Text
  TextBox_MenoPriezvisko Text
  TextBox_EmailKontakt Text
call ActivityStarter1 StartActivity
  
```

Otázka	Odpoveď
a. Ktoré externé aplikácie spúšťa táto aplikácia?	a.
b. Ktoré spoločné a ktoré rozdielne nastavenia majú jednotlivé spúšťače externých aplikácií?	b. spoločné rozdielne

Riešenie – Programujeme etudu 2.10 Asistent aktuálnej polohy

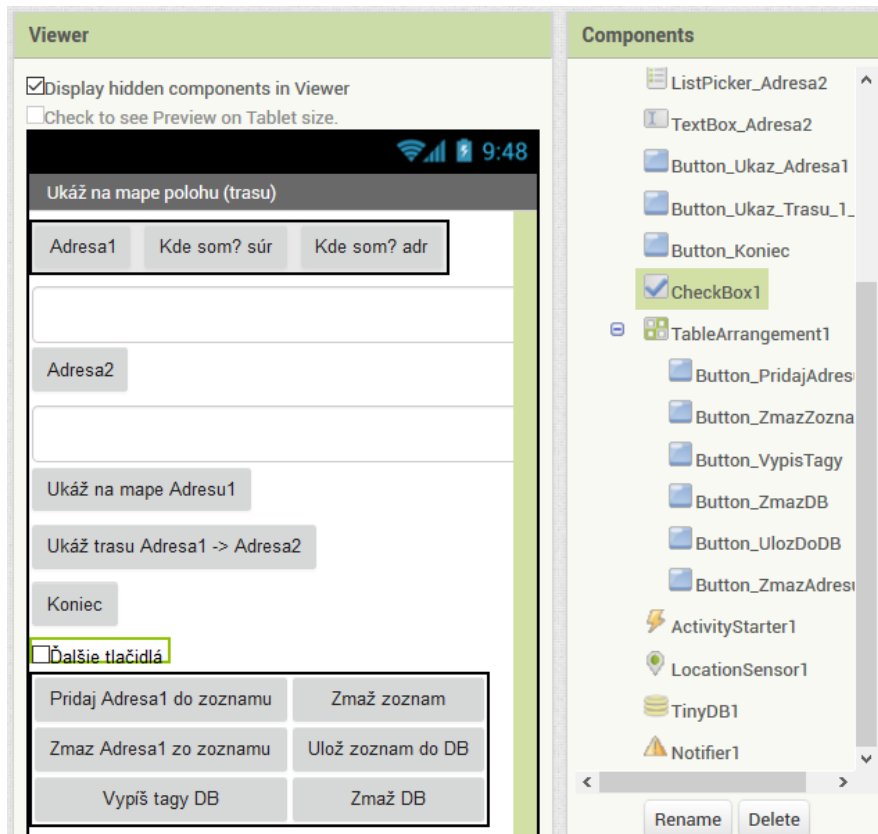
Úloha 2: Rozšírite aplikáciu `pmz_2_10_astarter_mapy.aia` o ďalšie funkcionality, napr.:

- zobrazenie textovej adresy aktuálnej polohy, ukončenie aplikácie,
- pridanie Adresy1 na koniec zoznamu adries, zmazanie Adresy1 zo zoznamu adries, zmazanie celého zoznamu,
- uloženie zoznamu adries do lokálnej databázy, zmazanie všetkých údajov z databázy.

Výsledný kód uložte do súboru `pmz_2_10_astarter_mapy_R.aia`.

(Poznámka: Inšpirácie pre ďalšie rozšírenia tejto a návrhy ďalších aplikácií využívajúce komponent ActivityStarter nájdete na webe, napr. <http://appinventor.mit.edu/explore/ai2/activity-starter.html>)

Používateľské rozhranie:



Pôvodnú aplikáciu sme rozšírili o:

- Button **Kde som? adr** – na zobrazenie textovej adresy aktuálnej polohy
- Button **Koniec** – na ukončenie behu aplikácie
- Skupinu 6 Buttonov zoskupenú v tabuľke pomocou komponentu TableArrangement, ktorý je celý zobrazený, resp. skrytý pomocou CheckBoxu **Ďalšie tlačidlá**:
 - Button **Pridaj Adresa1 do zoznamu** – na pridanie adresy na koniec zoznamu adries
 - Button **Zmaž Adresa1 zo zoznamu** – zmazanie Adresy1 zo zoznamu adries
 - Button **Zmaž zoznam** – na zmazanie celého zoznamu adries
 - Button **Ulož zoznam do DB** – na uloženie zoznamu adries do lokálnej databázy
 - Button **Vypíš tagy DB** – na kontrolný výpis tagov prvkov databázy (pomocou komponentu Notifier)
 - Button **Zmaž DB** – na zmazanie všetkých údajov z databázy

Zdrojový kód (1. časť):

V programe máme dve globálne premenné:

- **miesta** (typu zoznam) – na uloženie zoznamu adries
- **index** (typu celé číslo) – na uchovanie pozície vybraného prvku zoznamu, hlavne kvôli zmazaniu tohto prvku zo zoznamu

Po spustení aplikácie sa testuje či je lokálna databáza prázdna. Ak je prázdna, tak sa zoznam **miesta** naplní 5 prvkami a uloží do databázy. Ak nie je, tak sa naplní zoznam **miesta** hodnotami všetkých prvkov lokálnej databázy. Do TextBoxov **Adresa1** a **Adresa2** sa priradia zadané hodnoty.

```

initialize global index to 0
initialize global miesta to create empty list

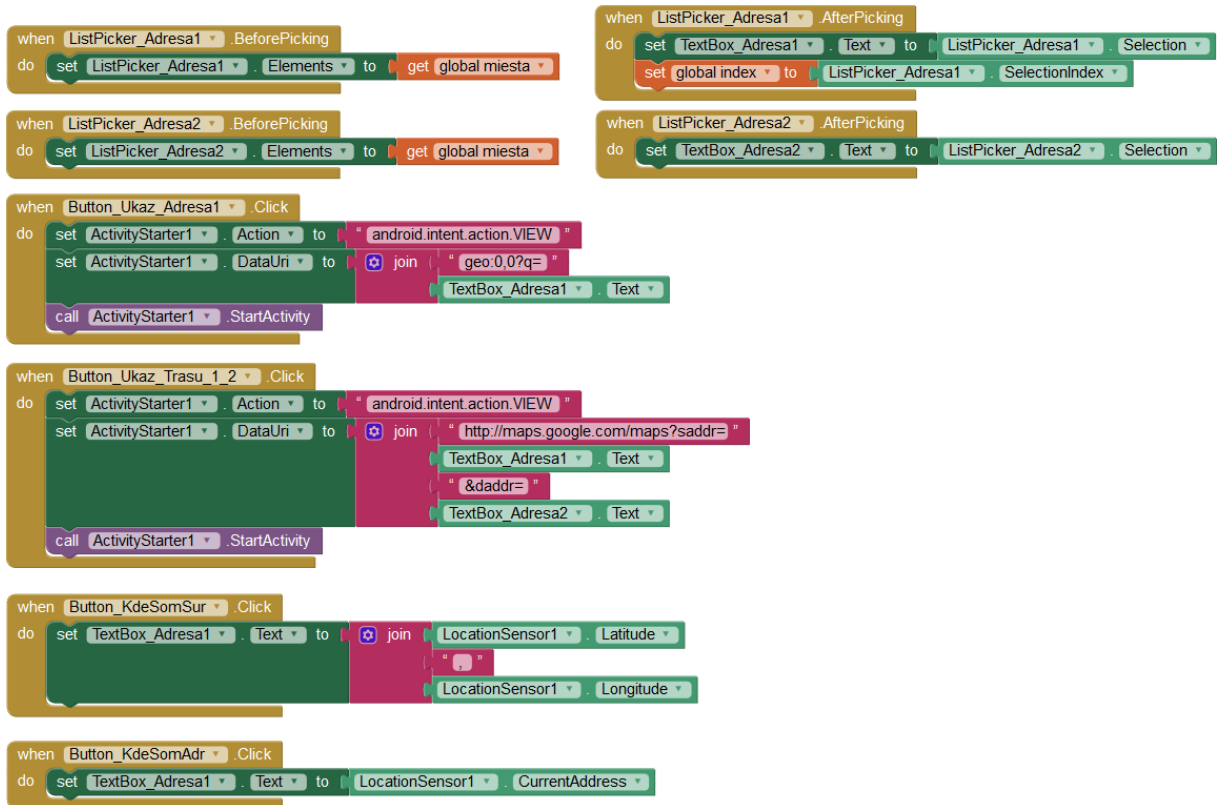
when Screen1.Initialize
do
  if length of list list call TinyDB1.GetTags = 0
  then
    set global miesta to make a list
    "Prírodovedecká fakulta UPJŠ Košice "
    "Fakulta prírodných vied UKF Nitra "
    "Fakulta matematiky fyziky a informatiky UK Mlyns... "
    "Fakulta prírodných vied UMB Banská Bystrica "
    "Fakulta riadenia a informatiky Žilinská Univerzita "
    call UlozZoznamDoDB
  else
    call NacitajZoznamZDB

  set TextBox_Adresa1.Text to call TinyDB1.GetValue
  tag "1"
  valueIfTagNotThere "Prírodovedecká fakulta UPJŠ Košice "
  set TextBox_Adresa2.Text to call TinyDB1.GetValue
  tag "2"
  valueIfTagNotThere "Fakulta prírodných vied UKF Nitra "

to NacitajZoznamZDB
do
  for each i from 1
  to length of list list call TinyDB1.GetTags
  by 1
  do
    add items to list list
    item get global miesta
    call TinyDB1.GetValue
    tag get i
    valueIfTagNotThere ""
  
```

Zdrojový kód (2. časť):

Táto časť kódu je len minimálne upravená oproti kódu pôvodnej verzie. Do komponentov ListPicker sme pred výberom prvku (udalosť BeforePicking) priradili zoznam **miesta** a po výbere prvku zo zoznamu **Adresa1** a **Adresa2** (udalosť AfterPicking) sme vybranú hodnotu zobrazili v TextBoxe **Adresa1** a **Adresa2**. V prípade výberu prvku **Adresa1** zo zoznamu sme do premennej **index** uložili poradie tohto prvku v zozname adresies hlavne kvôli zmazaniu vybraného prvku.



```

when ListPicker_Adresa1 . BeforePicking
do set ListPicker_Adresa1 . Elements to get global miesta

when ListPicker_Adresa2 . BeforePicking
do set ListPicker_Adresa2 . Elements to get global miesta

when ListPicker_Adresa1 . AfterPicking
do set TextBox_Adresa1 . Text to ListPicker_Adresa1 . Selection
set global index to ListPicker_Adresa1 . SelectionIndex

when ListPicker_Adresa2 . AfterPicking
do set TextBox_Adresa2 . Text to ListPicker_Adresa2 . Selection

when Button_Ukaz_Adresa1 . Click
do set ActivityStarter1 . Action to " android.intent.action.VIEW "
set ActivityStarter1 . DataUri to join ( " geo:0,0?q="
TextBox_Adresa1 . Text )
call ActivityStarter1 . StartActivity

when Button_Ukaz_Trasu_1_2 . Click
do set ActivityStarter1 . Action to " android.intent.action.VIEW "
set ActivityStarter1 . DataUri to join ( " http://maps.google.com/maps?saddr="
TextBox_Adresa1 . Text
"&daddr="
TextBox_Adresa2 . Text )
call ActivityStarter1 . StartActivity

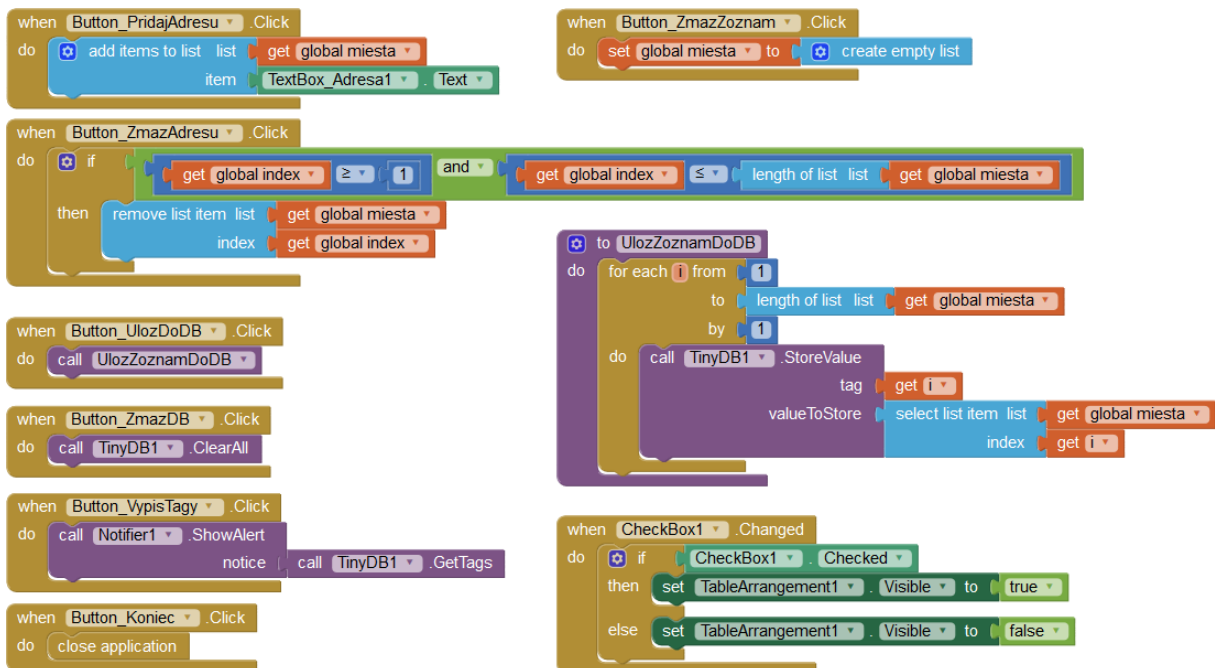
when Button_KdeSomSur . Click
do set TextBox_Adresa1 . Text to join ( LocationSensor1 . Latitude
" ."
LocationSensor1 . Longitude )

when Button_KdeSomAdr . Click
do set TextBox_Adresa1 . Text to LocationSensor1 . CurrentAddress
    
```

Zdrojový kód (3. časť):

V poslednej časti kódu sme doplnili:

- tlačidlo na pridanie adresy **Adresa1** na koniec zoznamu **miesta**,
- tlačidlo na zmazanie celého zoznamu **miesta**
- tlačidlo na zmazanie prvku s uvedeným jeho poradím v zozname (uloženého v premennej **index**)
- tlačidlo na uloženie všetkých prvkov zoznamu **miesta** do lokálnej databázy
- tlačidlo na zmazanie všetkých hodnôt v lokálnej databáze
- tlačidlo na kontrolný výpis tagov prvkov databázy (pomocou komponentu Notifier)
- tlačidlo na ukončenie behu aplikácie
- Checkbox na zobrazenie a skrytie komponentu TableArrangement so 6 tlačidlami



```

when Button_PridajAdresu Click
do
  add items to list list
  item get global miesta
  TextBox_Adresa1 Text

when Button_ZmazZoznam Click
do
  set global miesta to create empty list

when Button_ZmazAdresu Click
do
  if
    get global index >= 1 and get global index <= length of list list get global miesta
  then
    remove list item list
    index get global miesta
    get global index

when Button_UlozDoDB Click
do
  call UlozZoznamDoDB

when Button_ZmazDB Click
do
  call TinyDB1 ClearAll

when Button_VypisTagy Click
do
  call Notifier1 ShowAlert
  notice call TinyDB1 GetTags

when Button_Koniec Click
do
  close application

to UlozZoznamDoDB
do
  for each i from 1
  to length of list list get global miesta
  by 1
  do
    call TinyDB1 StoreValue
    tag get i
    valueToStore select list item list get global miesta
    index get i

when CheckBox1 Changed
do
  if
    CheckBox1 Checked
  then
    set TableArrangement1 Visible to true
  else
    set TableArrangement1 Visible to false
  
```

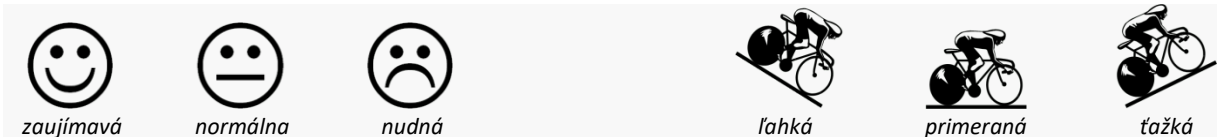
Sebahodnotiaca karta – Programujeme etudu 2.10 Asistent aktuálnej polohy

Zapísaním symbolu ✓ na príslušné miesta tabuliek vyjadrite, do akej miery ovládate uvedené prvky učiva.

Z uvedeného učiva nasledovným pojmom / princípom / tvrdeniam:	rozumiem	častočne rozumiem	vôbec nerozumiem
Nevizuálny komponent ActivityStarter (uvedený v skupine Connectivity) sa používa na spustenie externých aplikácií (napr. na zobrazenie mapy, spustenie videa, napísanie mailu, prídanie adresy do kontaktov)			
Pred samotným spustením externej aplikácie metódou ActivityStarter.StartActivity je potrebné nastaviť vlastnosti ActivityStarter.Action (napr. VIEW, INSERT) a ActivityStarter.DataUri na určenie typu (alebo konkrétnej) externej aplikácie a jej parametrov (napr. GPS poloha, video)			
Na výber prvku zo zoznamu sa používa komponent ListPicker (uvedený v skupine User Interface)			
Pred samotným výberom sa pomocou udalosti ListPicker.BeforePicking nastaví vlastnosť ListPicker.Elements na zoznam, z ktorého sa bude vyberať prvok			
Po výbere prvku sa vyvolá udalosť ListPicker.AfterPicking , ktorá vo vlastnosti ListPicker.Selection má uloženú hodnotu vybraného prvku a vo vlastnosti ListPicker.SelectionIndex index tohto prvku v zozname			

Z uvedeného učiva viem vykonať nasledovné činnosti:	samostatne	s malou pomocou	s veľkou pomocou
Vytvorí aplikáciu využívajúcu komponent ActivityStarter na spustenie <ul style="list-style-type: none"> • jednej externej aplikácie s jedným parametrom 			
<ul style="list-style-type: none"> • viacerých externých aplikácií alebo externej aplikácie s viacerými parametrami 			

Aká bola pre vás táto etuda? Zaujímavá? Ľahká? Zafarbite/zakrúžkujte niektorú z uvedených možností:



Uvedte, čo by ste ešte doplnili do tejto mobilnej aplikácie, aby bola viac zaujímavá a využiteľná v praxi:

2.11. Hlasovanie na internete

Prvky nového učiva

Komponenty	Udalosti (premenné)	Metódy	Vlastnosti
FireBaseDB	GotValue (tag, value) DataChanged (tag, value)	StoreValue GetValue	FirestoreURL

Jazykové konštrukcie	Iné prvky jazyka
list (index in list)	funkcia max

Pracovné súbory

Meno súboru	Použitie súboru
pmz_2_11_hlasovanie_na_internete_PL.docx	Pracovný list – Programujeme etudu 2.11 Hlasovanie na internete
pmz_2_11_hlasovanie_na_internete_R.docx	Riešenie – Programujeme etudu 2.11 Hlasovanie na internete
pmz_2_11_hlasovanie_na_internete_SK.docx	Sebahodnotiaci karta – Programujeme etudu 2.11 Hlasovanie na internete
pmz_2_11_kliker.aia	Pracovný súbor pre úlohu 1
pmz_2_11_hlasovanie_R.aia	Súbor s riešením úlohy 2

Komentár k priebehu a výsledkom výučby

Hlavným cieľom tejto etudy je, aby žiak bol schopný vytvoriť aplikáciu využívajúcu webovú databázu (Firebase).

Žiaci riešia **úlohy z pracovného listu**, ktorý obsahuje:

Úloha 1 – žiaci importujú a inštalujú aplikáciu `pmz_2_11_kliker.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (Firestore.StoreValue, GetValue, value, GotValue, DataChanged).

Úloha 2 – žiaci vytvárajú aplikáciu `pmz_2_11_hlasovanie_R.aia`, ktorá umožní webové hlasovanie s 3 možnosťami, početnosti ktorých sú uložené vo webovej databáze a s výpočtom víťaza hlasovania.

Na záver žiaci vyplnia **sebahodnotiacu kartu**, aby si uvedomili a zafixovali nové učivo. Sebahodnotiacu kartu môže učiteľ použiť na stručnú sumarizáciu nového učiva.

Pracovný list – Programujeme etudu 2.11 Hlasovanie na internete

Úloha 1: Vytvorte aplikáciu `pmz_2_11_kliker.aia`, ktorá bude zaznamenávať a vypisovať kliknutia viacerých používateľov pripojených na internet (napr. učiteľov na výlete, ktorí spočítavajú svojich žiakov vo viacerých skupinách, aby vedeli chatárovi nahlásiť záujem o dané jedlo alebo pitie).

Podme spoločne krok za krokom vyriešiť túto úlohu. Budeme postupovať nasledovne:

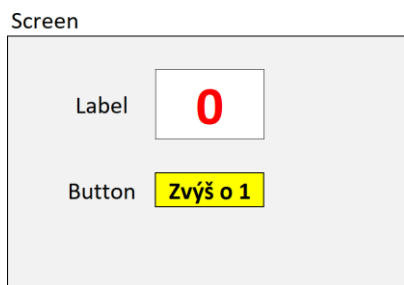
1. Navrhne funkcionality a používateľské rozhranie aplikácie
2. Navrhne štruktúru databázy
3. Na serveri Firebase pod svojim Google účtom zriadieme vlastnú databázu s navrhnutou štruktúrou
4. Naprogramujeme aplikáciu s navrhnutými funkcionalitami

1 Návrh funkcionality a používateľského rozhrania aplikácie

Používateľské rozhranie bude obsahovať len dva vizuálne komponenty:

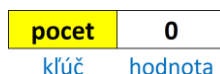
- tlačidlo na zvýšenie hodnoty spoločného počítadla o 1 (**Button**)
- popisok na výpis aktuálnej hodnoty spoločného počítadla (**Label**)

Na prácu s databázou použijeme nevizuálny komponent **Firestore** (uvedený v skupine Experimental)



2 Návrh štruktúry databázy

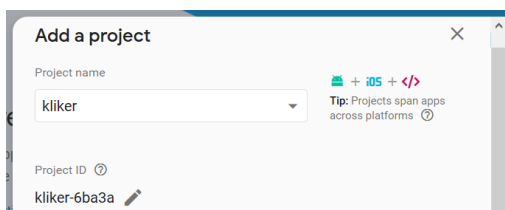
Naša databáza bude obsahovať len jeden kľúč **pocet**, v ktorom bude uložená aktuálna hodnota počtu žiakov (na začiatku 0).



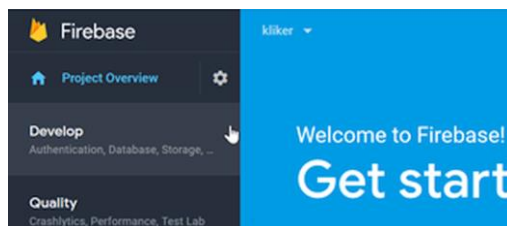
3 Zriadenie vlastnej databázy s navrhnutou štruktúrou na serveri Firebase

Navštívime webovú stránku <https://console.firebase.google.com/> a prihlásime sa na ňu svojim Google účtom.

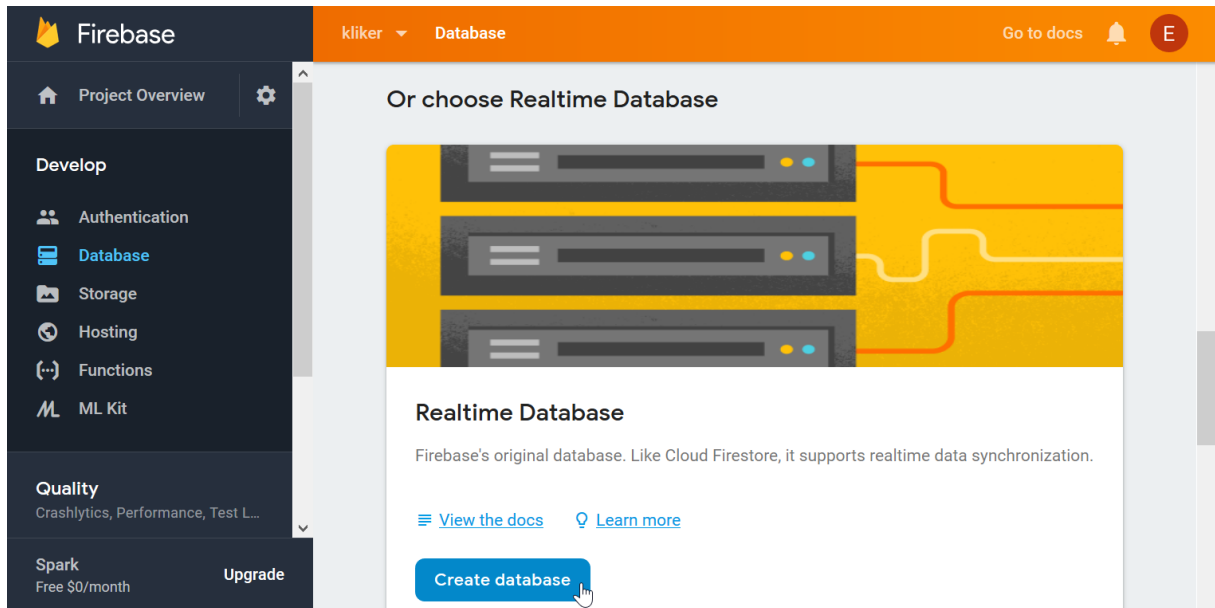
Stlačením tlačidla **Add project** na obrazovke vyvoláme okno, do ktorého uvedieme meno nášho projektu **kliker**. Nášmu projektu sa automaticky priradí **Project ID**, v našom prípade **kliker-6ba3a** (čo bude tiež menom našej vytváranej databázy).



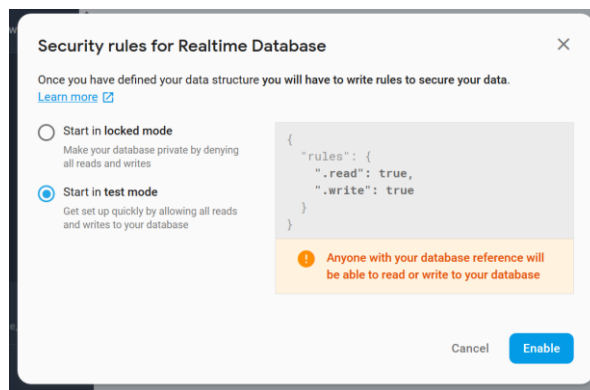
Po akceptovaní súhlasov bude v priebehu niekoľkých sekúnd vytvorený nový projekt, ktorý ďalej spravujeme.



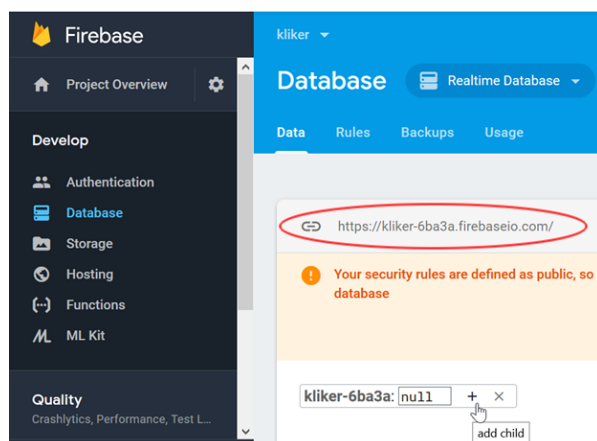
Teraz podme v tomto projekte vytvoriť novú databázu. Rozbalíme ponuku **Develop** v ľavej časti okna a vyberieme podponuku **Database**. Môžeme si vybrať jednu z dvoch druhov databáz: **Cloud Firestore** alebo **Realtime Database**. Pre naše účely nám poslúži tradičná praxou overená **Realtime Database**, ktorú vyberieme tlačidlom v dolnej časti okna.



Počas vytvárania databázy sme vyzvaní, aby sme si vybrali jeden z dvoch bezpečnostných režimov – **zamknutý** (locked mode) alebo **testovací** (test mode). My si vyberieme **testovací režim**, aby sme umožnili používateľom aplikácie čítať a zapisovať do našej databázy.



Po odkliknutí tlačidla **Enable** sa objaví obrazovka, v strede ktorej je uvedené URL našej databázy (zvýraznené červenou elipsou). V dolnej časti môžeme do našej databázy **kliker-6ba3a** pridávať kľúče s hodnotami.



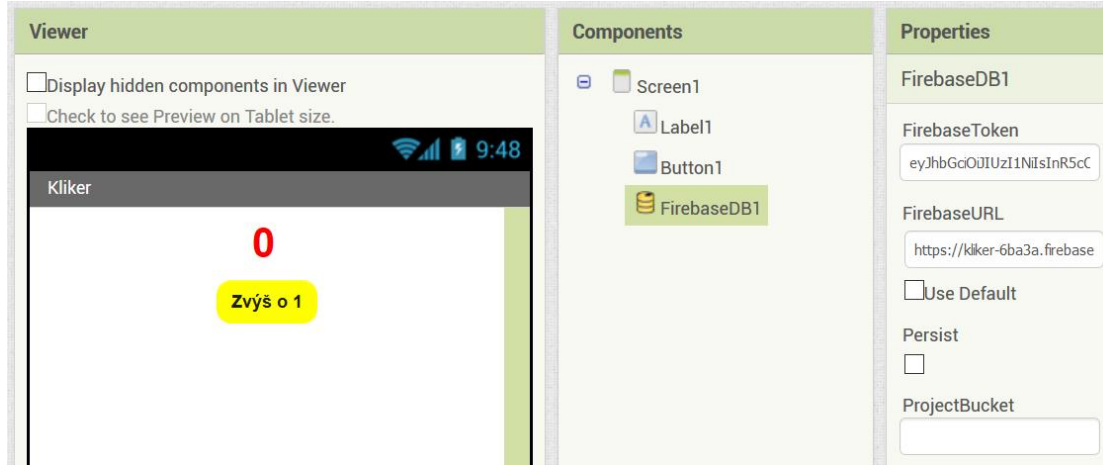
Pridaním kľúča **pocet** a jeho hodnoty **0** sme ukončili proces vytvárania štruktúry databázy a jej prvotného obsahu.

kliker-6ba3a
└── pocet: 0

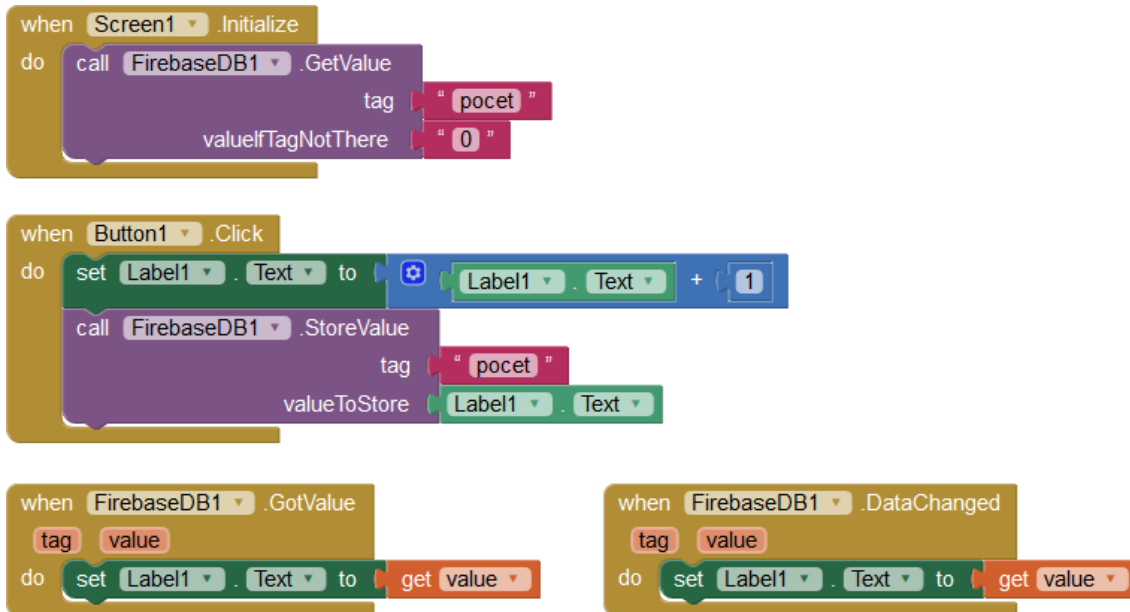
4 Naprogramovanie aplikácie s navrhnutými funkcionalitami

V režime **Designer** nastavíme komponentu **Firestore**:

- vlastnosť **FirestoreURL** z pôvodnej hodnoty „DEFAULT“ na hodnotu **https://kliker-6ba3a.firebaseio.com/**
- odznačíme vlastnosť **Use Default**
- vlastnosť **project Bucket** z pôvodnej hodnoty „pmz_2_11_kliker“ na prázdnu hodnotu.



V režime **Blocks** vytvoríme nasledovný zdrojový kód:



Po spustení aplikácie (vyvolaní udalosti **Screen.Initialize**) je zaslaná do databázy požiadavka na získanie hodnoty kľúča **pocet**. Ak databáza vie vrátiť hodnotu kľúča vyvolá sa udalosť **FirestoreDB.GotValue**, ktorá nastaví hodnotu popisku **Label1** na hodnotu získanú z databázy, ktorá je uložená v parametri **value** tejto udalosti.

Po stlačení tlačidla **Button1** sa zvýši hodnota popisku **Label1** o 1 a táto hodnota sa zapíše do databázy do kľúča **pocet**. Pri zmene kľúča **pocet** v databáze rôznymi používateľmi tejto aplikácie sa vyvolá udalosť **Firestore.DataChanged**, ktorá nastaví hodnotu popisku **Label1** na hodnotu získanú z databázy, ktorá je uložená v parametri **value** tejto udalosti.

Otázka	Odpoveď
i. Prečo metóda Firestore.GetValue neposkytne programu priamo hodnotu daného kľúča ako to robí lokálna databáza TinyDB , ale tá hodnota sa získa až po vyvolaní udalosti Firestore.GotValue ?	i.
j. Čo majú spoločné a čo rozdielne udalosti Firestore.GotValue a Firestore.DataChanged ?	j. spoločné rozdielne

Úloha 2: Vytvorte hlasovaciu aplikáciu `pmz_2_11_hlasovanie.aia`, ktorá umožní používateľom výber jednej z troch možností A, B, C. Bude ich tiež informovať, ktorá z možností hlasovania A, B, C je víťazom hlasovania.

(Odporúčanie: Pri spracovaní udalosti **Firestore.GotValue** a **Firestore.DataChanged** je potrebné pomocou podmienok rozlíšiť, ktorý kľúč (**tag**) sa načítal a ďalej spracovávať hodnotu (**value**) odpovedajúcu tomuto kľúču. Pri výpočte víťaza hlasovania sa dajú použiť funkcie na spracovanie zoznamu **select list item**, **index in list**, **make a list** a matematickú funkciu **max**.

Viac informácií o použití databázy Firestore pri programovaní v App Inventore nájdete na stránkach MIT, Experimental Components – App Inventor for Android:

<http://ai2.appinventor.mit.edu/reference/components/experimental.html#FirestoreDB>)

Riešenie – Programujeme etudu 2.11 Hlasovanie na internete

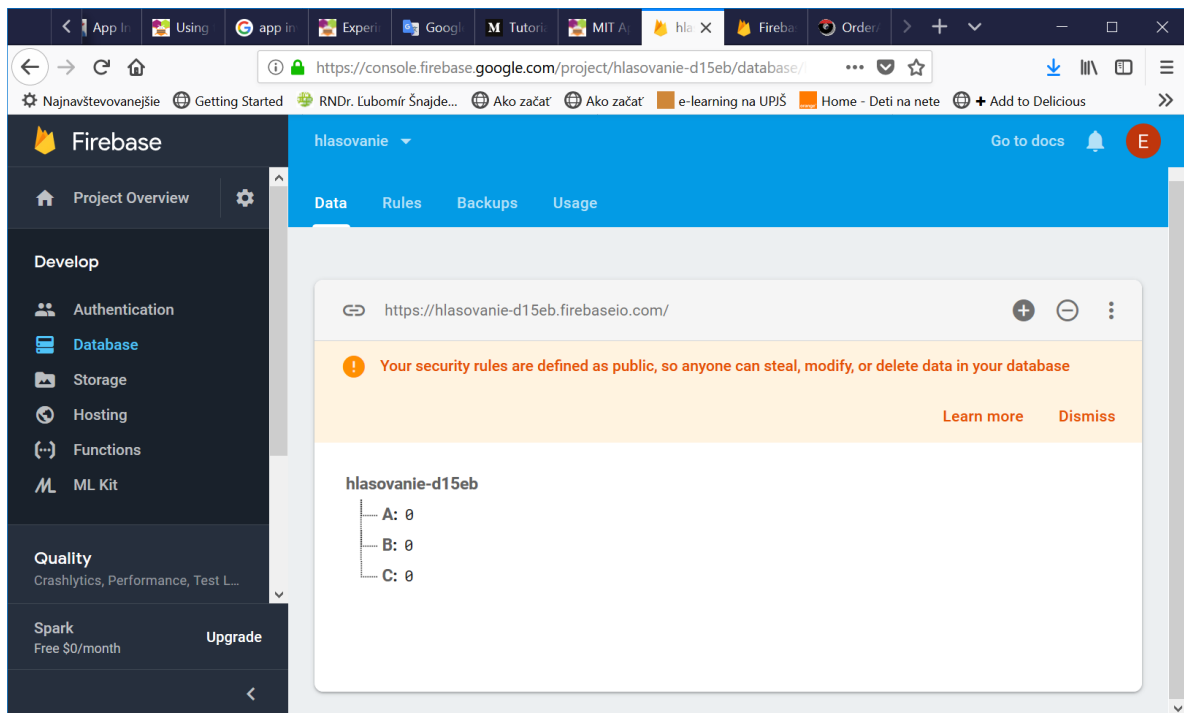
Úloha 2: Vytvorte hlasovaciu aplikáciu `pmz_2_11_hlasovanie.aia`, ktorá umožni používateľom výber jednej z troch možností A, B, C. Bude ich tiež informovať, ktorá z možností hlasovania A, B, C je víťazom hlasovania.

(Odporúčanie: Pri spracovaní udalosti **FirestoreGotValue** a **FirestoreDataChanged** je potrebné pomocou podmienok rozlíšiť, ktorý kľúč (**tag**) sa načítal a ďalej spracovávať hodnotu (**value**) odpovedajúcu tomuto kľúču. Pri výpočte víťaza hlasovania sa dajú použiť funkcie na spracovanie zoznamu **select list item**, **index in list**, **make a list** a matematickú funkciu **max**.

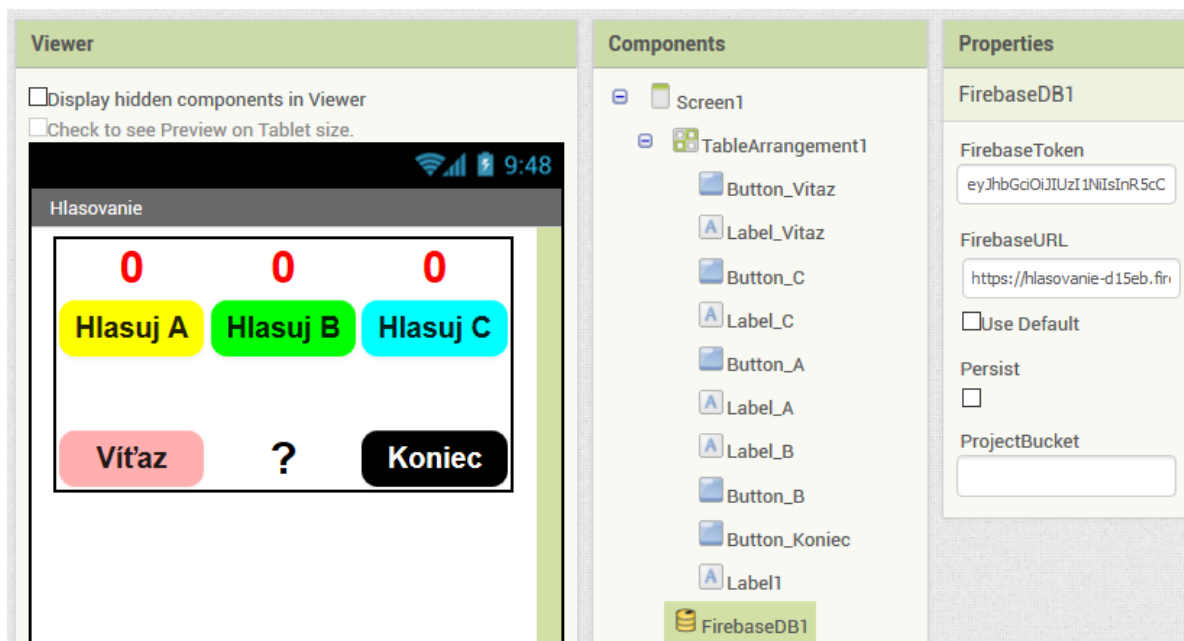
Viac informácií o použití databázy Firestore pri programovaní v App Inventore nájdete na stránkach MIT, Experimental Components – App Inventor for Android:

<http://ai2.appinventor.mit.edu/reference/components/experimental.html#FirestoreDB>

Štruktúra databázy (hlasovanie-d15eb) na serveri <https://hlasovanie-d15eb.firebaseio.com/>:



Používateľské rozhranie výslednej aplikácie `pmz_2_11_hlasovanie_R.aia`:



Zdrojový kód výslednej aplikácie pmz_2_11_hlasovanie_R.aia:

```

when Screen1.Initialize
do
  call FirebaseDB1.GetValue
  tag "A"
  valueIfTagNotThere "0"
  call FirebaseDB1.GetValue
  tag "B"
  valueIfTagNotThere "0"
  call FirebaseDB1.GetValue
  tag "C"
  valueIfTagNotThere "0"

when Button_A.Click
do
  set Label_A.Text to Label_A.Text + 1
  call FirebaseDB1.StoreValue
  tag "A"
  valueToStore Label_A.Text

when Button_B.Click
do
  set Label_B.Text to Label_B.Text + 1
  call FirebaseDB1.StoreValue
  tag "B"
  valueToStore Label_B.Text

when Button_C.Click
do
  set Label_C.Text to Label_C.Text + 1
  call FirebaseDB1.StoreValue
  tag "C"
  valueToStore Label_C.Text

when Button_Vitaz.Click
do
  set Label_Vitaz.Text to select list item list
  make a list
  "A"
  "B"
  "C"
  index
  index in list thing
  max
  Label_A.Text
  Label_B.Text
  Label_C.Text
  list
  make a list
  Label_A.Text
  Label_B.Text
  Label_C.Text

when FirebaseDB1.GetValue
tag value
do
  if get tag == "A"
  then set Label_A.Text to get value
  if get tag == "B"
  then set Label_B.Text to get value
  if get tag == "C"
  then set Label_C.Text to get value

when FirebaseDB1.DataChanged
tag value
do
  if get tag == "A"
  then set Label_A.Text to get value
  if get tag == "B"
  then set Label_B.Text to get value
  if get tag == "C"
  then set Label_C.Text to get value

when Button_Koniec.Click
do
  close application
  
```

Po spustení aplikácie (vyvolaní udalosti **Screen.Initialize**) je zaslaná do databázy požiadavka na získanie hodnôt kľúčov **A**, **B** aj **C**. Ak databáza vie vrátiť hodnotu kľúčov vyvolá sa udalosť **FirestoreDB.GetValue**, pri nej musíme pomocou podmienky rozlíšiť aký kľúč sa načítal a ďalej spracovávať hodnotu odpovedajúcu tomuto kľúču. Rovnako budeme postupovať aj pri spracovaní udalosti **Firestore.DataChanged**.

Pri výpočte víťaza hlasovania využijeme funkcie na spracovanie zoznamu. Pomocou funkcie **max** nájdeme maximálnu hodnotu z popiskov **Label_A**, **Label_B**, **Label_C**. Funkciou **index in list** nájdeme index maximálnej hodnoty. Bohužiaľ v prípade rovnosti sa nájde len index prvého výskytu. Napokon pomocou funkcie **select list item** priradíme nájdenej hodnote jednu z hodnôt **A**, **B**, **C**.

Odporúčame doriešiť aj prípad, ak je viacero maximálnych hodnôt, t. j. viacero víťazov hlasovania.

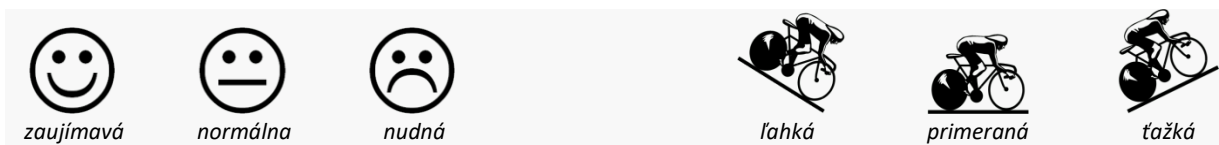
Sebahodnotiaca karta – Programujeme etudu 2.11 Hlasovanie na internete

Zapísaním symbolu ✓ na príslušné miesta tabuliek vyjadrite, do akej miery ovládate uvedené prvky učiva.

Z uvedeného učiva nasledovným pojmom / princípom / tvrdeniam:	rozumiem	čistočne rozumiem	vôbec nerozumiem
Pre uchovanie údajov prístupných pre viacerých používateľov sa používa webová databáza – komponent Firestore uvedený v skupine Experimental			
Pre vytvorenie vlastnej webovej databázy sa dá použiť server Firestore , na ktorom sa pridá nový projekt a v ňom nová databáza, napr. typu RealTime Database			
Pri vytváraní databázy treba nastaviť testovací mód , aby bola prístupná na čítanie aj zápis pre ostatných používateľov			
Poslednou fázou tvorby databázy je pridávanie kľúčov (tags) a nastavovanie ich hodnôt (values)			
Pri tvorbe používateľského rozhrania treba komponentu Firestore : 1. nastaviť vlastnosť FirestoreURL na hodnotu webovej adresy uvedenej na serveri Firestore v našom projekte 2. odškrtnúť vlastnosť Use Default 3. nastaviť prázdnu hodnotu na vlastnosť ProjectBucket			
Pre načítanie a zapísanie hodnoty daného kľúča do webovej databázy používame metódy Firestore.GetValue a Firestore.StoreValue .			
Po spustení metódy Firestore.GetValue sa vyvolá udalosť Firestore.GotValue , ktorá v parametri value má uloženú hodnotu daného kľúča			
Pre aktualizáciu údajov v mobilnej aplikácii sa využíva udalosť Firestore.Datachanged , ktorá sa vyvoláva pri každej zmene údajov databázy			

Z uvedeného učiva viem vykonať nasledovné činnosti:	samostatne	s malou pomocou	s veľkou pomocou
Vytvoriť na Firestore serveri nový projekt a webovú databázu s požadovanou štruktúrou a počiatočnými hodnotami			
Naprogramovať aplikáciu využívajúcu webovú databázu na Firestore serveri			

Aká bola pre vás táto etuda? Zaujímavá? Ľahká? Zafarbite/zakrúžkujte niektorú z uvedených možností:



Uvedte, čo by ste ešte doplnili do tejto mobilnej aplikácie, aby bola viac zaujímavá a využiteľná v praxi:

2.12. Komunikačný asistent

Prvky nového učiva

Komponenty	Udalosti (premenné)	Metódy	Vlastnosti
Texting	MessageReceived (number, messageText)	SendMessage	PhoneNumber Message ReceivingEnabled
PhoneCall		MakePhoneCall	PhoneNumber

Jazykové konštrukcie	Iné prvky jazyka
-	-

Pracovné súbory

Meno súboru	Použitie súboru
pmz_2_12_komunikacny_asistent_PL.docx	Pracovný list – Programujeme etudu 2.12 Komunikačný asistent
pmz_2_12_komunikacny_asistent_R.docx	Riešenie – Programujeme etudu 2.12 Komunikačný asistent
pmz_2_12_komunikacny_asistent_SK.docx	Sebahodnotiaca karta – Programujeme etudu 2.12 Komunikačný asistent
pmz_2_12_sms_nahlas.aia	Pracovný súbor pre úlohu 1
pmz_2_12_sms_telefon_R.aia	Súbor s riešením úlohy 2

Komentár k priebehu a výsledkom výučby

Hlavným cieľom tejto etudy je, aby žiak bol schopný vytvoriť aplikáciu využívajúcu komunikáciu prostredníctvom SMS (Texting) a hovorov (PhoneCall).

Žiaci riešia **úlohy z pracovného listu**, ktorý obsahuje:

Úloha 1 – žiaci importujú a inštalujú aplikáciu `pmz_2_12_sms_nahlas.aia`, ktorú ďalej skúmajú a svoje postrehy a závery diskusii zapisujú do tabuľky správania aplikácie (Texting.MessageReceived, number, messageText).

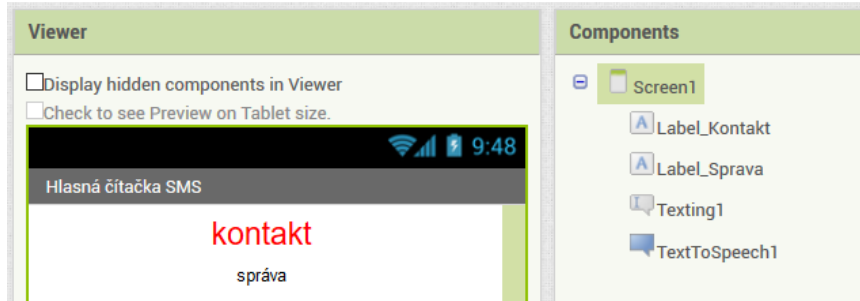
Úloha 2 – žiaci vytvárajú aplikáciu `pmz_2_12_sms_telefon_R.aia`, ktorá prečíta syntezátorom prijatú SMS správu, odosielateľovi odošle SMS správu alebo vytočí hovor.

Na záver žiaci vyplnia **sebahodnotiacu kartu**, aby si uvedomili a zafixovali nové učivo. Sebahodnotiacu kartu môže učiteľ použiť na stručnú sumarizáciu nového učiva.

Pracovný list – Programujeme etudu 2.12 Komunikačný asistent

Úloha 1: Do svojho Ai2 účtu importujte zdrojový kód aplikácie uložený v súbore `pmz_2_12_sms_nahlas.aia`. Po nainštalovaní a spustení aplikácie na mobilnom zariadení preskúmajte jej správanie. Svoje zistenia zapíšte do voľných políčk tabuliek.

Používateľské rozhranie:



Otázka	Odpoveď
a. V ktorej skupine komponentov sa nachádza komponent Texting ?	a.
b. Na čo slúži komponent Texting?	b.
c. Na ktorých mobilných zariadeniach sa dá použiť komponent Texting?	c.

Zdrojový kód:

```

when Texting1 .MessageReceived
  number  messageText
do
  set Label_Kontakt .Text to get number
  set Label_Sprava .Text to get messageText
  call TextToSpeech1 .Speak
    message (join ( " Došla Ti správa od "
                  get number
                  " Text správy: "
                  get messageText
    )
  )

```

Otázka	Odpoveď
d. Kedy sa vyvolá udalosť Texting.MessageReceived ?	d.
e. Čo predstavujú parametre number a messageText tejto udalosti?	e.
f. Čo robí daný program a na čo by dal použiť ?	f.

Úloha 2: Vytvorte aplikáciu `pmz_2_12_sms_telefon.aia`, ktorá:

- prečíta syntetickou rečou práve prijatú SMS správu,
- po zatrasení smartfónom sa vytočí telefónne spojenie s číslom uvedeným v došlej SMS správe,
- po priblížení ruky k hornej hrane smartfónu sa spustí analyzátor našej reči, ktorá sa opätovne spustí syntezátorom reči a pošle sa ako SMS správa pôvodnému odosielateľovi došlej SMS správy
- umožní prepínanie režimu prijímania správ 2 (Foreground) a 3 (Always)

(Odporúčanie: Na prepínanie režimov prijímania SMS správ použite vlastnosť **Texting.ReceivingEnabled**. Pri odosielaní SMS správy nastavte vlastnosti **Texting.PhoneNumber** a **Texting.Message** a spustite metódu **Texting.SendMessage**. Pred vytočením hovoru metódou **PhoneCall.MakePhoneCall** nastavte vlastnosť **PhoneCall.PhoneNumber**)

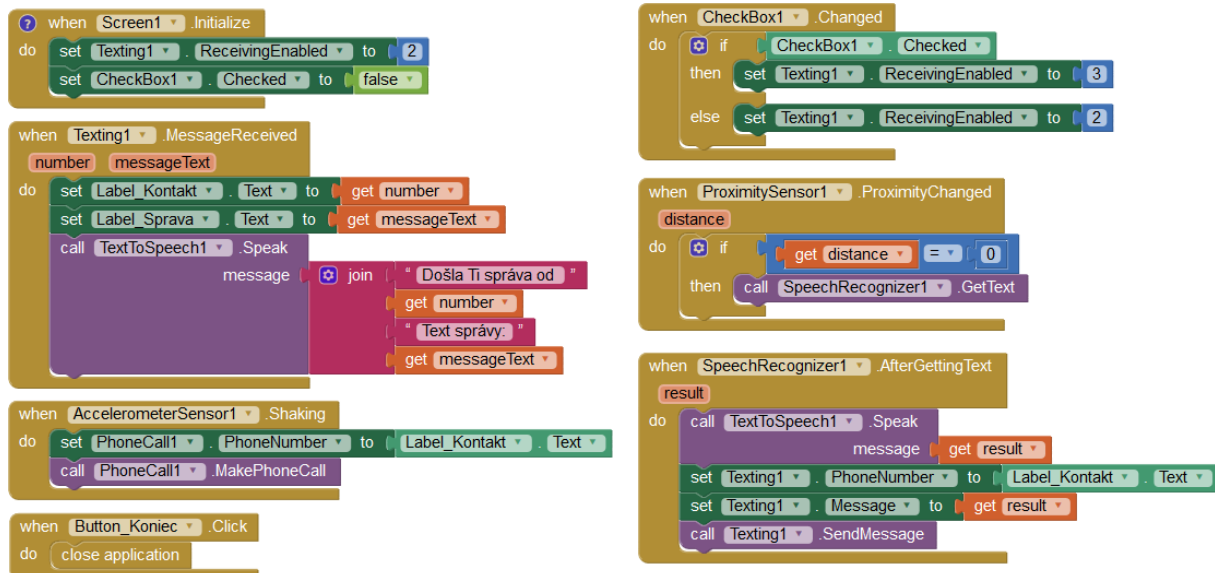
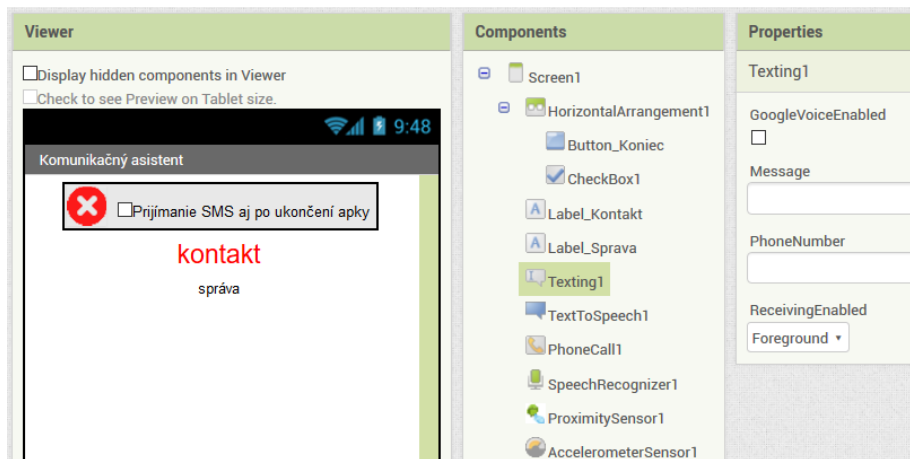
Riešenie – Programujeme etudu 2.12 Komunikačný asistent

Úloha 2: Vytvorte aplikáciu `pmz_2_12_sms_telefon.aia`, ktorá:

- prečíta syntetickou rečou práve prijatú SMS správu,
- po zatrasení smartfónom sa vytočí telefónne spojenie s číslom uvedeným v došlej SMS správe,
- po priblížení ruky k hornej hrane smartfónu sa spustí analyzátor našej reči, ktorá sa opätovne spustí syntezátorom reči a pošle sa ako SMS správa pôvodnému odosielateľovi došlej SMS správy
- umožní prepínanie režimu prijímania správ 2 (Foreground) a 3 (Always)

(Odporúčanie: Na prepínanie režimov prijímania SMS správ použite vlastnosť **Texting.ReceivingEnabled**. Pri odosielaní SMS správy nastavte vlastnosti **Texting.PhoneNumber** a **Texting.Message** a spustíte metódu **Texting.SendMessage**. Pred vytočením hovoru metódou **PhoneCall.MakePhoneCall** nastavte vlastnosť **PhoneCall.PhoneNumber**)

Používateľské rozhranie a zdrojový kód výslednej aplikácie `pmz_2_12_sms_telefon_R.aia`:



Pri spustení aplikácie (vyvolaním udalosti **Screen.Initialize**) sa nastaví režim prijímania SMS správ na hodnotu **2 (Foreground)**, t. j. len ak je táto aplikácia spustená. Pomocou **CheckBox**u prepíname režimy prijímania správ: **2 (Foreground)** a **3 (Always)**, t. j. aj po ukončení behu aplikácie na popredí. Zatrasením smartfónu (vyvolaním udalosti **AccelerationSensor.Shaking**) sa vytočí telefónne spojenie s číslom uvedeným v došlej SMS správe. Priblížením ruky k hornej hrane smartfónu (vyvolaním udalosti **ProximitySensor.ProximityChanged**) sa spustí metóda **SpeechRecognizer.GetText** na rozpoznávanie reči. Po úspešnom rozpoznaní reči sa vyvolá udalosť **SpeechRecognizer.AfterGettingText**, v rámci ktorej sa spustí metóda **TextToSpeech.Speak** na syntézu predtým nahovorenej reči a spustí sa metóda **Texting.SendMessage**, ktorá pošle sa SMS správa pôvodnému odosielateľovi.

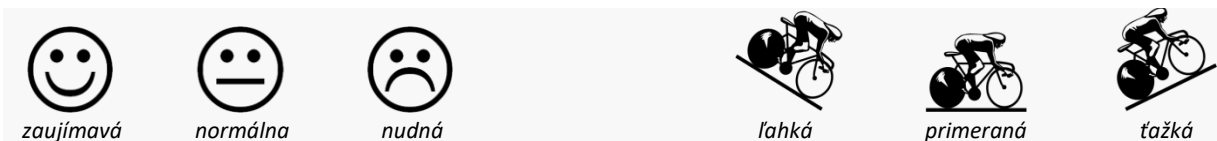
Sebahodnotiaca karta – Programujeme etudu 2.12 Komunikačný asistent

Zapísaním symbolu ✓ na príslušné miesta tabuliek vyjadrite, do akej miery ovládáte uvedené prvky učiva.

Z uvedeného učiva nasledovným pojmom / princípom / tvrdeniam:	rozumiem	častočne rozumiem	vôbec nerozumiem
Nevizuálny komponent Texting je uvedený v skupine Social a slúži na posielanie a prijímanie SMS správ			
Nevizuálny komponent PhoneCall je uvedený v skupine Social a slúži na vytáčanie a prijímanie telefónnych hovorov			
Aplikácie využívajúce komponenty Texting a PhoneCall fungujú na smartfónoch a iných mobilných zariadeniach schopných prijímať a posilať SMS a telefónne hovory			
Prijatím SMS správy sa vyvolá udalosť Texting.MessageReceived a v jej parametroch number a messageText sú uložené telefónne číslo odosielateľa a text prijatej SMS správy			
Na zaslanie SMS správy slúži metóda Texting.SendMessage , s nastavenými vlastnosťami Texting.PhoneNumber a Texting.Message			
Na prepínanie režimov prijímania SMS správ použite vlastnosť Texting.ReceivingEnabled , v režime 2 (Foreground) sa dajú prijímať správy len keď je spustená aplikácia a v režime 3 (Always) aj keď aplikácia nebeží na popredí			
Na vytočenie telefonického hovoru slúži metóda PhoneCall.MakePhoneCall s nastavenou vlastnosťou PhoneCall.PhoneNumber			

Z uvedeného učiva viem vykonať nasledovné činnosti:	samostatne	s malou pomocou	s veľkou pomocou
Vytvoriť komunikačnú aplikáciu využívajúcu:			
• komponent Texting			
• komponent PhoneCall			
• komponent TextToSpeech alebo SpeechRecognizer			
• komponent AccelerometerSensor alebo ProximitySensor			

Aká bola pre vás táto etuda? Zaujímavá? Ľahká? Zafarbite/zakrúžkujte niektorú z uvedených možností:



Uvedte, čo by ste ešte doplnili do tejto mobilnej aplikácie, aby bola viac zaujímavá a využiteľná v praxi:

2.13. Bibliografia

Beer, P., & Simmons, C. (2015). *Hello App Inventor! - Android programming for kids and the rest of us*. Manning Publications Co.

McGrath, M. (2014). *Building Android Apps In Easy Steps*. (Second Edition). In Easy Steps Limited.

Michaličková, V. (2016). *Programovanie mobilných aplikácií v prostredí MIT App Inventor 2*. Nitra: Univerzita Konštatína Filozofa v Nitre.

MIT. (2018). . Retrieved from: <http://ai2.appinventor.mit.edu/>

Wolber, D., Abelson, H., Spertus, E., & Liz, L. (2014). *App Inventor 2 - Create Your Own Android Apps*. O'Reilly.

2.14. Register pojmov

A

AccelerometerSensor, 2, 6, 42, 43, 44, 47, 49, 84

ActivityStarter, 3, 63, 65, 66, 68, 72

B

Ball, 2, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 22

BarcodeScanner, 2, 36, 38, 41

Button, 2, 8, 9, 11, 12, 13, 18, 19, 28, 31, 34, 35, 46, 68,
74

C

Canvas, 2, 5, 6, 7, 8, 10, 13, 22, 31, 32, 33, 34, 35

Clock, 2, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 22

F

FireBaseDB, 3, 73

H

HorizontalArrangement, 2, 3, 8, 9, 11, 13, 63

C

CheckBox, 2, 42, 43, 44, 47, 48, 49

L

Label, 2, 8, 9, 11, 12, 13, 23, 25, 30, 58, 74, 79

ListPicker, 3, 63, 65, 66, 70, 72

LocationSensor, 2, 58, 59, 60, 61, 62, 63

M

Map, 3, 58, 59, 62

N

Notifier, 2, 23, 24, 28, 30, 68, 71

O

OrientationSensor, 2, 15, 16, 17, 20, 22

P

Pedometer, 2, 42, 43, 45, 46, 48, 49

PhoneCall, 3, 81, 82, 83, 84

ProximitySensor, 2, 42, 43, 44, 47, 48, 49, 83, 84

S

Screen, 2, 5, 6, 7, 8, 9, 11, 12, 13, 14, 19, 31, 33, 35, 76,
79, 83

Slider, 2, 23, 24, 26, 29, 30

Sound, 2, 8, 10, 11, 13, 14, 17, 42, 44, 48

SpeechRecognizer, 2, 36, 39, 41, 83, 84

Spinner, 2, 36, 39, 40, 41

T

TableArrangement, 2, 42, 43, 48, 68, 71

TextBox, 2, 23, 26, 29, 30, 36, 65

Texting, 3, 81, 82, 83, 84

TextToSpeech, 2, 36, 37, 38, 39, 41, 83, 84

TinyDB, 2, 3, 15, 16, 19, 22, 63, 76