



Pedagogická
fakulta
Faculty
of Education

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta
Katedra informatiky

Bakalářská práce

Grafické programování mobilních aplikací v prostředí MIT App Inventor 2

Vypracovala: Ema Kučerová
Vedoucí práce: doc. PaedDr. Jiří Vaníček, Ph.D.

České Budějovice 2018

Prohlašuji, že svoji bakalářskou práci jsem vypracoval/a samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě Pedagogickou fakultou elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

Datum

Podpis studenta

Chtěla bych poděkovat vedoucímu práce panu doc. PaedDr. Jiřímu Vaníčkovi, Ph.D. za cenné rady, trpělivost a odborné vedení bakalářské práce. Dále bych chtěla poděkovat panu Mgr. Vladimíru Příbylovi za ochotu ověřit praktickou část mé práce.

Název práce: Grafické programování mobilních aplikací v prostředí MIT App Inventor 2

Autor: Ema Kučerová

Katedra: Katedra Informatiky

Vedoucí diplomové práce: doc. PaedDr. Jiří Vaníček, Ph. D.

E-mail vedoucího: vanicek@pf.jcu.cz

ANOTACE

Tato práce se zabývá grafickým vývojovým prostředím MIT App Inventor 2, které je určeno pro vytváření mobilních aplikací. MIT App Inventor je založen na podobném principu jako Scratch, jenž je určen k výuce základů programování, kde se kód sestavuje pomocí bloků s příkazy, které do sebe zapadají podobně jako puzzle. Porovnávají se zde obě prostředí, jak dalece je možno se zkušenostmi se Scratchem přejít do MIT App Inventoru. Práce obsahuje průzkum tohoto prostředí z pohledu využití možností mobilních zařízení, vestavěných senzorů a způsobů komunikace s okolím a také analyzuje množství materiálů které jsou k dispozici k výuce v MIT App Inventoru. Hlavním cílem této práce je sada úloh s tutoriály. Sada obsahuje závěrečný projekt, který slouží jako ukázka možnosti tohoto prostředí a některých senzorů v aplikaci. Závěrečný projekt je doplněn dokumentací, jak aplikace funguje a jak byla vytvořena.

KLÍČOVÁ SLOVA

App Inventor, Scratch, mobilní aplikace, tutoriál, úlohy

Title: Graphical programming of mobile applications in MIT App Inventor environment

Author: Ema Kučerová

Department: Katedra Informatiky

Supervisit: doc. PaedDr. Jiří Vaníček, Ph. D.

Supervisor's e-mail address: vanicek@pf.jcu.cz

ANOTATION

The work deals with graphic development environment MIT App Inventor 2, based on similar principle like Scratch is. Scratch is designed to teach the basics of programming where the code is assembled using blocks of commands that match each other similar to a puzzle. Both environments are compared, how far is possible to continue in MIT App Inventor with experiences in Scratch. The work includes exploring this environment from point of view of the use of mobile devices, embedded sensors and ways of communicating with the environment, and also analyzes the amount of materials available for learning in MIT App Inventor. The main purpose of this work is set of tasks with tutorials containing sample tasks with a final project to show the possibilities of this environment and some of sensors in the application. Part of the final project is documentation how the application works and how was it created.

KEYWORDS

App Inventor, Scratch, mobile app, tutorial, tasks

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Fakulta pedagogická

Akademický rok: 2016/2017

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Ema KUČEROVÁ

Osobní číslo: P15669

Studijní program: B7507 Specializace v pedagogice

Studijní obor: Informační technologie a e-learning

**Název tématu: Grafické programování mobilních aplikací v prostředí MIT
App Inventor**

Zadávací katedra: Katedra informatiky

Z á s a d y p r o v y p r a c o v á n í :

MIT App Inventor 2(<http://appinventor.mit.edu/explore/>) je online vývojové prostředí, které je vyvíjeno na Massachusetts Institute of Technology. V tomto prostředí se vytvářejí mobilní aplikace pro chytré telefony s operačním systémem Android. MIT App inventor 2 (dále též MITI) je založen na stejném principu jako vývojové prostředí Scratch, též vyvíjeného na MIT. Programy pro mobilní aplikace se sestavují z předpřipravených bloků, není tedy podmínkou znát syntaxi programovacího jazyka. MIT App inventor 2 umožňuje okamžitý náhled na vyvíjenou aplikaci.

Studentka prozkoumá prostředí MIT App Inventor 2 a vyzkouší různé možnosti a funkce této aplikace pro tvorbu jednoduchých mobilních aplikací. Porovná prostředí MIT App Inventor 2 s prostředím Scratch, zjistí, nakolik je programování v obou prostředích kompatibilní a nakolik a za jakých podmínek je možné po zvládnutí programování ve Scratch přejít na MIT App inventor 2. Porovná tato prostředí z pohledu úvodní orientace s cílem posoudit, jak dalece je možno pomocí předchozích zkušeností ze Scratch přejít do prostředí MIT App Inventor 2, v oblasti způsobu práce, komfortu prostředí, snadnosti zápisu a editace algoritmů, práce s daty a grafikou. Prozkoumá prostředí MIT App Inventoru 2 z pohledu využití možností mobilních zařízení, vestavěných senzorů a způsobů komunikace s okolím.

Studentka zpracuje a analyzuje přehled a množství studijních materiálů, které jsou k výuce v MIT App Inventor 2 k dispozici především u nás a na Slovensku a také v angličtině. Porovná tyto materiály z pohledu praktičnosti, srozumitelnosti a způsobu, jakým se pomocí nich programování vyučuje, jak kvalitně pedagogicky jsou materiály zpracovány.

Dalším cílem bakalářské práce bude vypracovat sérii vlastních úloh pro zvládnutí základního programování v prostředí MIT App Inventoru 2. Tyto příklady budou sloužit k vysvětlení, jak v tomto prostředí pracovat, jak programovat a k ukázce jaké funkce se dají do mobilní aplikace vložit.

Studentka zpracuje metodickou příručku s praktickými radami pro uživatele, které získá z literatury nebo vlastní zkušeností. Příručka bude vypracována tak, aby byla srozumitelná i pro začátečníky v programování, kteří ovládají programování ve Scratch. Tyto úlohy sestaví do příručky, která povede uživatele po zvláštностech programování v MIT App Inventoru 2. V příručce budou řešeny způsoby přenesení aplikace do zařízení, prohlížení vyvíjené aplikace v simulátoru, testování mobilní aplikace a nastavení takového mobilního zařízení pro běh vytvořené aplikace. Bude zde vysvětleno, jak vytvářet mobilní aplikace v jednotlivých krocích.

Studentka dále vytvoří jeden vzorový programovací projekt, ve kterém ukáže možnosti tohoto prostředí především v ovládání některých senzorů v mobilní aplikaci. V dokumentaci bude uvedeno, jak aplikace funguje, jakým způsobem byla vytvořena. Tento projekt se stane součástí a vyvrcholením příručky.

Studentka vyzkouší úlohy, vytvořené k příručce, ve výuce podle možností se studenty střední nebo vysoké školy.

Příloha zadání bakalářské práce

Seznam odborné literatury:

1. WOLBER, David, Hal ABELSON, Ellen SPERTUS a Liz LOONEY. **App Inventor 2: create your own Android apps. Second edition.** ISBN 978-1-491-90684-2.
2. BEER, Paula a Carl SIMMONS. **Hello app inventor!: Android programming for kids and the rest of us.** ISBN 9781617291432.
3. KREJSA, Jan. **Výuka základů programování v prostředí Scratch.** České Budějovice, 2012. Diplomová práce. Jihočeská univerzita v Českých Budějovicích. Fakulta pedagogická. Vedoucí práce doc. PaedDr. Jiří Vaníček, Ph.D.
4. ŠNAJDER, L'ubomír.: **Metodika výučby programovania mobilných zariadení.** Zborník konferencie DidInfo2015. Banská Bystrica: Univerzita Mateja Bela, 2015. - ISBN 978-80-557-0852-2. - s. 18-27
5. Tomcsányiová Monika a Peter Tomcsányi.: **Vyučovanie programovania vývojom androidových aplikácií v App Inventore.** Zborník konferencie DidInfo2015. Banská Bystrica: Univerzita Mateja Bela, 2015. - ISBN 978-80-557-0852-2. - s. 160-163
6. **Vývojové prostredie MIT App Inventor 2.** [online]. Dostupné na: <http://ai2.appinventor.mit.edu/>
7. WOLBER, David. **Learn to build Android apps.** [online]. Dostupné na: <http://www.appinventor.org>
8. ŠNAJDER, L'ubomír. **Výučba programovania mobilných aplikácií v App Inventor 2.** [online]. Dostupné na: <http://ics.upjs.sk/snajder/ai2/>

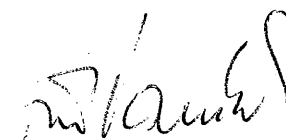
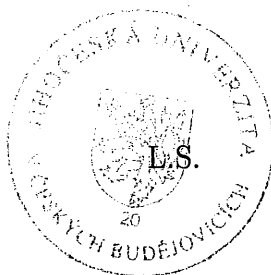
Rozsah grafických prací: **CD ROM**
Rozsah pracovní zprávy: **40**
Forma zpracování bakalářské práce: **tištěná**
Seznam odborné literatury: **viz příloha**

Vedoucí bakalářské práce: **doc. PaedDr. Jiří Vaníček, Ph.D.**
Katedra informatiky

Datum zadání bakalářské práce: **24. dubna 2017**
Termín odevzdání bakalářské práce: **30. dubna 2018**



Mgr. Michal Vančura, Ph.D.
děkan



doc. PaedDr. Jiří Vaníček, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 24. dubna 2017

Obsah

Obsah	6
1. Úvod.....	8
1.1. Cíle práce	9
1.2. Metody práce	9
2. MIT App Inventor 2	11
3. Porovnání prostředí Scratch a App Inventor 2	13
3.1. Porovnání prostředí	15
4. Dostupné studijní materiály k App Inventor 2.....	17
4.1. App Inventor 2: Create your own apps.....	17
4.2. App Inventor: Vytvořte si vlastní aplikaci pro Android	17
4.3. Hello App Inventor! Android Programming for Kids and the Rest of Us.....	18
4.4. Educational App Development Toolkit for Teachers and Learners	19
4.5. Appinventor.mit.edu/explore	19
4.6. Appinventor.org	20
4.7. Výučba programovania mobilných aplikácií v App Inventore 2	20
5. Vývoj mobilních aplikací v MIT App Inventor 2	21
5.1. Základní požadavky	21
5.2. Přihlášení do aplikace App Inventor 2.....	21
5.3. Seznámení s prostředím App Inventor 2	22
5.4. Testování a ladění aplikací	24
5.5. Přenos vytvořených aplikací do zařízení.....	26
6. Sada úloh s tutoriály pro MIT App Inventor	27
6.1. Seznam úloh.....	28
6.1.1. Hello World	29
6.1.2. Virtuální kostka	36
6.1.3. Pohlednice	42

6.1.4. Průvodce	52
6.1.5. Kde jsi?.....	65
6.1.6. Závěrečný projekt – Fit App.....	72
7. Ověření sady úloh ve výuce	74
8. Závěr.....	75
9. Literatura a ostatní použité zdroje	77
Přílohy	

1. Úvod

Téma bakalářské práce Grafické programování mobilních aplikací v prostředí MIT App Inventor 2 jsem si vybrala z toho důvodu, že mne zaujalo, jakým způsobem se dají vytvářet aplikace, s porovnáním mé předchozí zkušenosti vývoje mobilní aplikace v Android Studiu. Líbí se mi možnost jednoduchého využití různých senzorů, rychlý export aplikace do telefonu využitím QR kódu a také to, že není nezbytné znát syntaxi programovacího kódu.

V dnešní době jsou mobilní operační systémy Android těmi nejvyužívanějšími. Podle webového portálu Gartner.com OS Android je využíván více než 80 % mobilních zařízení. Většina z nás téměř každý den využívá mobilní aplikace na chytrých telefonech, od elektronického bankovníctví, přes hry, po ovládání různých zařízení. Využití mobilních aplikací je velice rozšířené. Pokud máme nápad na aplikaci, máme tyto možnosti – nedělat s tím nic, zaplatit programátorovi za zhotovení aplikace, či si aplikaci naprogramovat sami, což není zrovna jednoduché. Ve spolupráci Massachusettského technologického institutu a společnosti Google bylo vyvinuto grafické vývojové prostředí – nyní již ve druhé verzi – MIT App Inventor 2 (dále AI2).

Mobilní aplikace se v tomto prostředí sestavují pomocí bloků s příkazy, které do sebe zapadají podobně jako puzzle. Programovat je zde tedy možné bez znalosti syntaxe programovacího jazyka. AI2 umožňuje okamžitý náhled na aplikaci a nahrání aplikace do telefonu pomocí QR kódu, či přes WI-FI připojení. Dále nabízí využití různých senzorů, např. senzor polohy zařízení, GPS, využití dotykové obrazovky, Bluetooth připojení aj. Dají se zde tedy vytvořit zajímavé aplikace za krátkou dobu.

1.1. Cíle práce

Porovnávání prostředí Scratch a MIT App Inventor 2

V mé práci porovnávám programovací prostředí Scratch a MIT App Inventor 2, nakolik je programování v obou prostředích kompatibilní a nakolik a za jakých podmínek je možné po zvládnutí programování ve Scratch přejít na MIT App Inventor 2. Porovnávám tato prostředí v oblasti způsobu práce, komfortu prostředí, snadnosti zápisu a editace algoritmů, práce s daty a grafikou.

Analýza dostupných studijních materiálů

Analyzuji dostupné studijní materiály a tutoriály, které jsou k dispozici v českém, slovenském a anglickém jazyce.

Sada úloh s tutoriály

Zpracovala jsem sadu úloh s tutoriály, která vede uživatele vývojem mobilní aplikace v prostředí MIT App Inventor 2. Tyto úlohy slouží k vysvětlení, jak v tomto prostředí pracovat, jak vytvářet mobilní aplikace a k ukázce jaké funkce se dají do mobilní aplikace vložit.

Závěrečný projekt

V závěrečném projektu jsou ukázány možnosti prostředí App Inventor 2 na obsáhlejší mobilní aplikaci, která je doplněna dokumentací, kde je popsáno, jak byla aplikace vytvořena a jak funguje.

1.2. Metody práce

Aby bylo možné porovnat programovací prostředí Scratch a MIT App Inventor 2 a posoudit, nakolik a za jakých podmínek je možné přejít z prostředí Scratch do AI2, prostudovala jsem nejdříve vývojové prostředí Scratch. Zapsala jsem si proto kurz Didaktika programování, kde se pracovalo primárně v prostředí Scratch. Tak jsem mohla pokračovat v MIT App Inventor 2 jako uživatel, který se zatím orientuje pouze ve Scratchi.

V rámci přípravy k mé bakalářské práci jsem využila knihy App Inventor od Davida Wolbera, kterou jsem se inspirovala při vymýšlení mobilních aplikací do sady úloh a ze které jsem se sama učila v App Inventor 2 programovat.

Analyzovala jsem studijní materiály a tutoriály, které jsou k dispozici v českém, slovenském a anglickém jazyce, z pohledu praktičnosti, srozumitelnosti a způsobu, jakým se pomocí nich programování vyučuje, jak kvalitně pedagogicky jsou materiály zpracovány. Analyzovala jsem materiály, které byly k dispozici v plném rozsahu.

Po získání přehledu o AI2 jsem vymyslela vlastní aplikace, které jsem zpracovala do tutoriálů – podrobné a srozumitelné návody krok za krokem, které budou obsahovat názorný kód. Tyto tutoriály jsem doplnila ilustračními obrázky, screenshoty.

Aplikace jsem navrhovala tak, aby se daly využít zkušenosti z předchozích úloh a zároveň přibylo něco nového. Mým cílem bylo, aby se využívané funkce a senzory telefonu v aplikacích neopakovaly a využilo se tak co nejvíce neviditelných komponent, ale aplikace stále dávala smysl.

Sada úloh je vytvořena tak, aby uživatele naučila orientovat se v prostředí AI2 a jak v něm vytvářet mobilní aplikace, neslouží k tomu, aby uživatele naučila programovat. Předpokládá se, že uživatel již zkušenost s programováním má, například z prostředí Scratch.

Ověřování probíhalo ve výuce na Střední škole strojní a elektrotechnické ve Velešíně pod vedením kmenového učitele Mgr. Vladimíra Příbyla. Ze zpětné vazby, kterou mi Mgr. Příbyl poskytl zodpovězením sady mých následných otázek jsem zpracovala výstup z ověření. Výstup z ověření se nachází v závěru bakalářské práce.

2. MIT App Inventor 2

Webová aplikace MIT App Inventor 2 (<http://appinventor.mit.edu/explore/>) je online vývojové prostředí pro tvorbu mobilních aplikací, které bylo vyvinuto na Massachusetts Institute of Technology (dále MIT) ve spolupráci se společností Google. První beta verze byla spuštěna v roce 2010 která byla v roce 2013 byla nahrazena novější verzí.

Služba je zdarma dostupná na webovém odkazu <http://ai2.appinventor.mit.edu/>. Aplikace je cloudově řešena, k vývoji mobilních aplikací pomocí MIT App Inventor 2 (dále též AI2) tak není potřeba instalace softwaru a programování probíhá v rámci webového prohlížeče. Je tedy nutné připojení k internetu. Přihlašování do AI2 probíhá pomocí účtu Google.

Počet uživatelů AI2 od vydání aplikace neustále roste. V roce 2015 čítal počet registrovaných uživatelů téměř 3 miliony s více jak 100 tisíci aktivními uživateli týdně a s více jak 7 milióny vytvořenými aplikacemi. [3] V roce 2018 tato čísla výrazně stoupla na 6 milionů registrovaných uživatelů z 195 různých zemí, z nichž je 400 tisíc měsíčně aktivních a bylo vytvořeno přes 22 milionu aplikací. [6]

V tomto prostředí se vytvářejí mobilní aplikace pro chytré telefony a tablety s operačním systémem Android, které mohou využívat různé senzory a funkce mobilního zařízení. MIT App Inventor 2 je založen na stejném principu jako vývojové prostředí Scratch, tedy přetahováním a skládáním předpřipravených bloků zapadajících do sebe podobně, jako puzzle. Scratch je prostředí určené pro výuku programování pro děti, ve kterém se dají vytvářet například animace a hry, též vyvíjeno na MIT. Není zde tedy podmínkou znát syntaxi programovacího jazyka, správnost kódu se pozná tak, že do sebe bloky zapadají. MIT App Inventor 2 umožňuje okamžitý náhled na vyvíjenou aplikaci na mobilním zařízení či emulátoru. Emulátor simuluje operační systém Android.

Aplikace App Inventor 2 je dostupná v několika cizích jazycích, například v anglickém, španělském, francouzském, český jazyk mezi nimi bohužel není.

Využití MIT App Inventor 2

MIT App Inventor je vhodný pro všechny začátečníky, pro děti i dospělé, kteří si chtějí jednoduše vytvořit mobilní aplikaci, naučit se základy programování. AI2

je určen pro výuku algoritmizace a základů programování např. ve školách. Při výuce může žák poměrně rychle navrhnout, vytvořit a otestovat vlastní funkční aplikaci a získá praktické zkušenosti s programováním mobilních aplikací. App Inventor 2 může být také využit například učitelem, který si chce vytvořit svou vlastní pomůcku do výuky ve formě aplikace, protože aplikací, které se dají využít k výuce s využitím mobilních zařízení, tedy m-learningu, je malé množství. Tato forma výuky může být zpestřením a motivací pro žáky.

3. Porovnání prostředí Scratch a App Inventor 2

V této kapitole porovnávám vývojová prostředí Scratch a App Inventor 2 z pohledu úvodní orientace s cílem posoudit, nakolik je programování v obou prostředích kompatibilní a jak dalece je možno s předchozími zkušenostmi z prostředí Scratch přejít do prostředí AI2. Tato prostředí porovnávám v oblasti způsobu práce, komfortu prostředí, snadnosti zápisu a editace algoritmů, práce s daty a s grafikou.

Scratch je bezplatné vizuální programovací prostředí určené pro výuku základů programování a algoritmizace, dnes rozšířené po celém světě. Je určeno pro děti od 6 let bez předchozích zkušeností s programováním. Vytvářejí se zde interaktivní příběhy, hry a animace. Scratch je stejně jako AI2 vyvinut na MIT. Obě programovací prostředí jsou určena k úvodu do programování bez předchozích zkušeností.

Způsob práce

V prostředích AI2 i Scratch se programuje s využitím předpřipravených bloků s příkazy, jejichž přetahováním se skládá kód. Kategorie bloků Události, Operátory, Data, Ovládání a Procedury mají v obou prostředích totožné barvy.

Komfort prostředí

Obě vývojová prostředí AI2 i Scratch jsou cloudové aplikace, není tedy nutná instalace, pouze internetové připojení.

AI2 je dostupný v 10 cizích jazycích a používán ve více než 195 zemích. Český jazyk dostupný není. Scratch je dostupný ve více než 40 cizích jazycích včetně v českém jazyce a je používán ve více než 150 zemích. [6]

V AI2 probíhá přihlášení pomocí účtu Google, který se používá k různým službám. Scratch vyžaduje vlastní přihlašovací účet a registraci konkrétně pro Scratch.

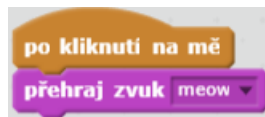
V AI2 jsou dva režimy, Designer pro návrh uživatelského prostředí a Blocks pro skládání kódu. [11] Dále je potřeba zařízení Android, nebo emulátor pro testování

výsledné aplikace. Scratch má návrh uživatelského prostředí, kód i výsledný program v jednom okně webového prohlížeče.

V galerii AI2 se po výběru aplikace z galerie nejprve otevře celý projekt a až poté je možné získat a otevřít mobilní aplikaci k otestování. Scratch umožňuje po výběru programu z galerie okamžitě jej otestovat a poté otevřít projekt a nahlédnout na kód, či jej rovnou převzít a upravovat. Scratch umožňuje tyto projekty přidávat do tzv. studií, které slouží podobně, jako adresáře.

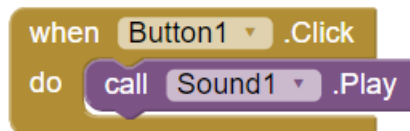
Snadnost zápisu a editace algoritmů

V prostředí Scratch jsou příkazy téměř stejné, jako bychom je někomu řekli v reálném životě.



Obrázek 3.1: Ukázka bloku v Scratch

V AI2 jsou příkazy podobné spíše příkazům v programovacích jazycích pro běžné programování, například Java.



Obrázek 3.2: Ukázka bloku v AI2

Práce s daty

AI2 nabízí na rozdíl od Scratch samostatné kategorie s velkým množstvím bloků k práci s textem, se seznamy a s matematickými funkcemi. Pro prostředí Scratch nabízí jen základní matematické funkce a pouze tři bloky k práci s textem, a to ke zjištění délky slova, získání písmena ze slova dle pozice a spojení dvou slov.

Práce s grafikou.

Scratch nabízí velice dobré zpracování grafiky, dají se zde vytvářet i velmi složité obrazce a animace. Komponenta Canvas v AI2 také umožňuje kreslení a tisk, ale

výsledek zde není dostatečně kvalitní, například nakreslení čáry tahem prstu je velice kostrbaté.

Kompatibilita prostředí

Pro uživatele se zkušenostmi s prostředím Scratch po přechodu do prostředí App Inventor 2 bude nejspíše matoucí, že je v prostředí pouze Designer, režim pro návrh uživatelského prostředí aplikace. K programování aplikace se musí přepnout do režimu Blocks. V prostředí Scratch jsou oba tyto nástroje v jednom okně ještě s výsledným programem. Pro českého uživatele bude asi nepohodlné, že AI2 není v češtině. Pokud uživatel zvládá programování ve Scratch, a umí základní úroveň angličtiny, přestože nemá jiné zkušenosti s programováním, nebude pro něj pravděpodobně složité pochopit, co mohou jednotlivé bloky vykonat, či jak je skládat. Co se týče uživatelského rozhraní, bude spíše nutné využít např. nějakých studijních materiálů k seznámení se s komponentami, jejich možnostmi a využitím. Toto tvrzení je ale velmi subjektivní a záleží na mnoha faktorech, nakolik jsou tato prostředí kompatibilní.

3.1. Porovnání prostředí

Mým cílem není vývojová prostředí hodnotit, ale porovnávat, zvolila jsem tedy měřítko hodnocení znaky + a -, kde + znamená, že v je prostředí v dané oblasti lepší než prostředí druhé, - znamená, že v dané oblasti je horší než prostředí druhé. V případě = jsou v dané oblasti srovnatelná. Porovnávání vychází z mé vlastní zkušenosti s prací v obou vývojových prostředích.

Oblast porovnávání	AI2	Scratch
Způsob práce	=	=
Komfort prostředí	-	+
Snadnost zápisu a editace algoritmu	=	=
Práce s daty	+	-
Práce s grafikou	-	+

Vzhledem k tomu, že v prostředí Scratch i AI2 se programuje na blokovém principu, **způsob práce a snadnost zápisu a editace algoritmu** jsou zde srovnatelná.

Komfort prostředí Scratch je lepší, protože je vše v jednom okně, není nutné přepínat mezi režimy, není třeba připojovat zařízení nebo spouštět emulátor Androidu a velkou výhodou je, že je Scratch dostupný i v češtině.

Práce s daty v AI2 je lepší, protože má mnohem více možností pro práci s textem, i pro práci se seznamy a složitější matematické funkce než Scratch.

Lepší **práci s grafikou** nabízí Scratch. Vzhledem k tomu, že je určen pro tvorbu her, animací a interaktivních příběhů nabízí mnohem rozmanitější práci s grafikou.

Scratch je vhodným prostředím k programování, které předchází programování v App Inventoru. Programování mobilních aplikací je v dnešní době atraktivní způsob výuky programování, žáci si mohou naprogramovat vlastní aplikace do svých telefonů, dochází tedy zde k větší motivaci ke studiu programování mobilních aplikací, avšak pro mladší děti je vhodnější Scratch, protože příkazy bloků jasně říkají, co znamenají v „lidské řeči“ a je snazší k porozumění, zatímco v App Inventoru už jsou příkazy srovnatelnější s příkazy běžného programovacího jazyka například s programovacím jazykem Java.

4. Dostupné studijní materiály k App Inventor 2

Dalším cílem mé bakalářské práce je analyzovat dostupné studijní materiály k programování v prostředí App Inventor 2. Během průzkumu jsem dohledala 9 publikací zabývajících se prostředím MIT App Inventor 2. Všechny jsou v anglickém jazyce. V českém jazyce je dostupná pouze jediná kniha, první edice jedné z dohledaných knih, ale je napsána pro první verzi prostředí App Inventor. Na internetu jsem dohledala dvoje webové stránky v angličtině zabývající se výukou AI2 a webové stránky, které obsahují malé množství metodických a studijních materiálů ve slovenštině. V této kapitole analyzuji materiály, které jsem měla možnost prostudovat celé, nikoliv pouze ukázky, či obsahy knih.

4.1. App Inventor 2: Create your own apps

Autoři: David Wolber, Hal Abelson, Ellen Spertus, Liz Looney

Kniha App Inventor 2 uživatele naučí, jak programovat mobilní aplikace v prostředí MIT App Inventor 2, je určena spíše pro uživatele se základní znalostí programování. Je rozdělena na dvě části, tutoriály a manuál, první část obsahuje 13 tutoriálů k vývoji mobilních aplikací. Každá kapitola obsahuje úvodní text, shrnutí, co uživatel vytvoří a co se naučí, tutoriál, který popisuje podrobně všechny kroky, jak naprogramovat aplikaci, které jsou doplněny obrázky a na závěr jsou zde varianty, jak aplikaci modifikovat a také shrnutí kapitoly.

Druhou částí knihy je manuál s ilustračními obrázky. Dokumentace, kde je vysvětlena architektura aplikací vytvořených v AI2, navrhování a testování aplikací, práce s pamětí, jak vytvářet animované aplikace, větvení, práce se seznamy a databázemi, využití senzorů, komunikace s webem, procedury a využití bloků.

4.2. App Inventor: Vytvořte si vlastní aplikaci pro Android

Autoři: David Wolber, Hal Abelson, Ellen Spertus, Liz Looney

Jediná dohledaná publikace na trhu v českém jazyce, zabývající se prostředím MIT App Inventor je tato, která je první edicí předchozí knihy. Tato edice slouží

ke studiu vývoje Android aplikací v první verzi App Inventoru, přesto je možné ji k programování v nové verzi App Inventor 2 využít. K tomu je potřeba vědět, že App Inventor 2 je na rozdíl od první verze cloudová aplikace, není tedy třeba nic instalovat. Dále jsou tyto dvě verze rozdílné v možnostech připojení mobilního zařízení, kde v AI2 je navíc možné připojit mobilní zařízení pomocí Wi-Fi. Způsoby připojení a základní orientace v AI2 jsou popsány v této práci, viz kapitola 5.

Změny, které obsahuje druhá verze App Inventoru jsou popsány v následujícím odkazu: <http://appinventor.mit.edu/explore/ai2/whats-new.html>. V případě využití první edice doporučuji se s těmito změnami seznámit.

Obsah první edice knihy App Inventor je stejný, jako obsah edice druhé, je pouze upraven pro novou verzi aplikace AI2.

4.3. Hello App Inventor! Android Programming for Kids and the Rest of Us

Autoři: Paula Beer and Carl Simmons

Knihy Hello App Inventor! učí nejen děti bez předchozích zkušeností základům programování a tvorbě aplikací v App Inventoru. Úvodem je zde napsáno, co je to počítačový program, aplikace, Android, App Inventor, historie chytrých telefonů a jejich funkce.

Knihy je napsána velice srozumitelně, obsahuje užitečné informace k problematice, mnoho ilustračních obrázků, které pomohou dětem porozumět textu, tipy, nápovědy, zvýrazněné důležité informace, věty, které srovnávají situace v programování se situacemi v reálném životě, pro lepší představu, vysvětlení různých infromatických či matematických pojmů (např. radius, RAM, copyright, apod.), varování v některých případech.

Knihy je rozdělena do kapitol, kde se každá zabývá nějakým tématem z programování a na závěr obsahuje shrnutí, minitesty a nápady k vyzkoušení. Hello App Inventor! obsahuje skoro 50 návodů na tvorbu aplikací, které jsou rozmístěny do kapitol. Návodů jsou doplněny tipy pro pokročilý vývoj. Kapitola 14. obsahuje dvě větší složitější závěrečné aplikace.

Podle mého názoru je tato kniha velice dobře pedagogicky zpracována, pobízí uživatele k zamyšlení, vede programováním od základu a tvorbou aplikací zábavnou formou a uživatel se může nenásilně naučit novým a užitečným pojmům. Nevýhodou této knihy je, že není dostupná v českém jazyce.

4.4. Educational App Development Toolkit for Teachers and Learners

Autor: Ishan Sudeera Abeywardena

Tato příručka je určena pro studenty i učitele, provází vývojem aplikací v AI2 od vytvoření účtu až publikaci her do obchodu Google play. V této příručce jsou úlohy rozděleny na aktivity a tutoriály, které jsou na sobě nezávislé. Obsahuje 8 aktivit a 6 tutoriálů. Příručka obsahuje též názorný rozvrh třídního workshopu a dotazník pro zpětnou vazbu od účastníků.

Aktivity jsou úlohy, kde se vytváří jednoduché aplikace s jednou, či dvěma funkcemi, které mají uživatele seznámit s využitím základních funkcí a komponent. Aktivity obsahují pracovní list, který obsahuje popis aplikace, seznam využitých komponent a jejich vlastnosti, screenshot aplikace a předpoklad získaných znalostí. Aktivity dále obsahují podrobný návod, který je doplněn ilustračními obrázky a kód aplikace.

Tutoriály jsou složitější aplikace, s více funkcemi na sobě závislých. Předpokládá se zde u uživatele, že využije znalosti z aktivit. Tutoriál je složený pouze z pracovního listu a výsledného kódu.

4.5. Appinventor.mit.edu/explore

Oficiální stránky AI2 nabízí množství studijních materiálů, které jsou určeny pro samouky. Tyto materiály jsou dostupné na <http://appinventor.mit.edu/explore> v nabídce Resources.

Get started – zde se nachází návody, jak připravit a nastavit telefon k živému testování, či emulátoru, jak pracovat v prostředí AI2, jak vytvořit instalační soubory aplikací a sdílet je. Dále je zde odkázáno na úvodní 4 video tutoriály pro úplné začátečníky, které jsou doplněny i textovými tutoriály.

Tutorials - v této nabídce se nachází 31 různých tutoriálů, které jsou seřazeny podle složitosti, od nejjednodušších. Je zde možnost tutoriály filtrovat podle využitých komponent.

4.6. Appinventor.org

Autoři: David Wolber, Felicia Kamriani, Thomas Oropeza, Leonard Fernando

Webové stránky dostupné na <http://www.appinventor.org/> jsou určeny k výuce a studiu vývoje mobilních aplikací v AI2.

Course In A Box – e-learningový kurz určen pro úplné začátečníky v AI2. Kurz obsahuje učební texty, návody, videa, tutoriály, kvízy a kapitoly z knihy App Inventor 2: Create your own apps, autor knihy je zároveň autorem tohoto portálu. Některé tutoriály v tomto kurzu jsou proto na stejné aplikace. V záložce Teach jsou k tomuto kurzu dostupné metodické materiály pro učitele.

Video tutorials – 9 video tutoriálů rozdělených na aplikace pro začátečníky a pokročilé. Tutoriály jsou rozděleny do kroků, mezi nimiž lze přepínat a také je možné stáhnout aplikaci do telefonu, či stáhnout její zdrojový kód do počítače. Opět se zde vyskytuje pár tutoriálů na aplikace z knihy App Inventor 2: Create your own apps.

4.7. Výučba programování mobilních aplikací v App

Inventore 2

Autor: doc. RNDr. Lubomír Šnajder, PhD.

Na webových stránkách pana docenta Šnajdera <http://ics.upjs.sk/~snajder/> jsou dostupné materiály ve slovenštině pro výuku programování mobilních aplikací v AI2. Je zde 11 úloh, z toho některé obsahují pracovní list nebo metodickou příručku a pro úvod do problematiky je k dispozici učební text.

5. Vývoj mobilních aplikací v MIT App Inventor 2

Tato kapitola uvádí do vývoje mobilních aplikací v AI2. Jsou zde popsány základní požadavky k vývoji, jak se přihlásit do prostředí AI2 a seznámení s tímto prostředím a způsobem práce v něm. Dále je zde přehled komponent, funkcí a možností prostředí App Inventor 2.

5.1. Základní požadavky

K programování mobilních aplikací v prostředí MIT App Inventor 2 je potřeba mít připravené následující:

- Připojení k internetu, ideálně Wi-Fi
- Počítač a webový prohlížeč Google Chrome, Safari, Firefox, Microsoft Edge.
- Vlastní uživatelský účet Google
- Chytrý telefon nebo tablet s OS Android nastavený podle následujících instrukcí: Abyste mohli mobilní aplikace na zařízení testovat a instalovat, je nutné na zařízení Android povolit instalaci z neznámých zdrojů. Obvykle se toto nastavení nachází v Nastavení > Zabezpečení > Neznámé zdroje. Po ukončení testování a ladění je vhodné neznámé zdroje opět vypnout.

5.2. Přihlášení do aplikace App Inventor 2

MIT App Inventor nevyžaduje zvláštní registraci, postačí účet Google, který se využívá k mnoha dalším službám. Přihlásit se můžete na následujících webových stránkách:

<http://ai2.appinventor.mit.edu>

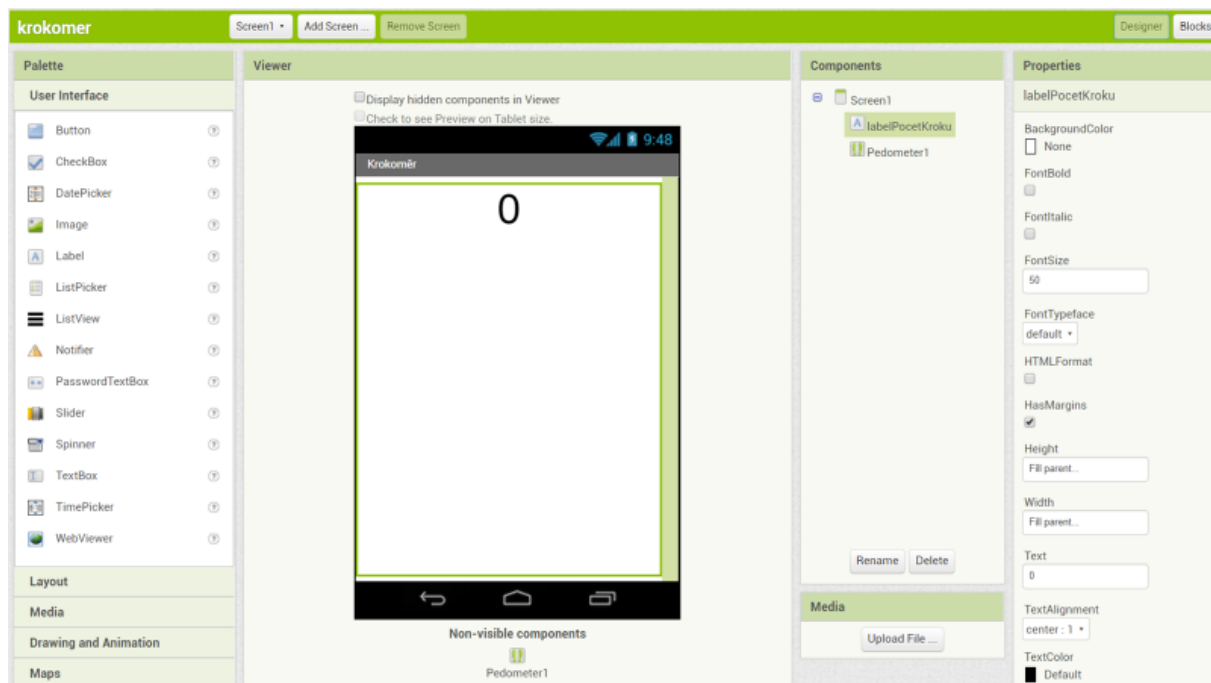
Po zadání přihlašovacích údajů, je požádáno o povolení k přístupu k vašemu Google účtu a také sdílení vaší e-mailové adresy s aplikací MIT App Inventor. Toto slouží pouze k ověření, že povolujete využívání vašeho účtu v této aplikaci.

5.3. Seznámení s prostředím App Inventor 2

Prostředí aplikace MIT App Inventor se dělí na dva režimy: Designer a Blocks. Tyto dvě režimy se mezi sebou přepínají v pravém horním rohu.

Designer

V režimu Designer se navrhuje uživatelské prostředí mobilní aplikace, tedy vše, co uživatel vidí v mobilní aplikaci. Uživatelské prostředí se skládá z komponent. Komponenta je každý objekt aplikace, se kterým se pracuje. Komponenty se rozlišují na viditelné a neviditelné. Viditelné komponenty jsou viditelné prvky obrazovky, například popisky, textová pole, tlačítka, obrázky a podobně. Neviditelné komponenty jsou funkce a senzory mobilního zařízení, které bude aplikace využívat, například akcelerometr, gyroskop, GPS, rozpoznávač řeči, přehrávač médií, fotoaparát apod. Každá komponenta má vlastnosti, které se v pravém panelu Properties mohou editovat. Např. velikost, barva, název, text, zarovnání apod.



Obrázek 5.1: Režim Designer v prostředí MIT App Inventor

Blocks

V této části aplikace MIT App Inventor se programuje funkce a chování aplikace s využitím bloků. Bloky jsou rozděleny do sekcí, podle jejich využití. Jsou zde bloky kontrolní, logické, matematické, textové, bloky s listy, barvami, procedurami a proměnnými.

Po přidání komponenty se zde objeví také bloky, které například nastavují, nebo získávají hodnotu komponenty, či určují nějakou reakci na událost komponenty. Událost může být například kliknutí na tlačítko, reakcí výpis textu na obrazovku.

S bloky se pracuje stylem drag-and-drop, tedy táhni a pusť a skládají se do sebe podobným způsobem, jako puzzle. Správně přiřazené bloky s příkazy do sebe pasují.

Do baňůžku se dají vložit bloky, které chcete uchovat na později, nebo předat do jiných projektů apod. Přetažením bloku do popelnice se blok smaže.



Obrázek 5.2: Režim Blocks v AI2 s kódem jednoduché aplikace krokoměř

5.4. Testování a ladění aplikací

Způsobů, jak testovat aplikace na mobilním telefonu či tabletu je několik, podrobné postupy, jak aplikace testovat naleznete níže.

Propojení zařízení a aplikace MIT App Inventor zrealizujete pomocí nabídky Connect na hlavní liště.

AI Companion

Máte-li zařízení Android a Wi-Fi, doporučeným způsobem je testování v reálném čase pomocí aplikace MIT AI2 Companion.

1. Nainstalujte do svého zařízení aplikaci MIT AI2 Companion z obchodu Google Play.
2. V aplikaci MIT App Inventor z nabídky Connect na hlavní liště vyberte položku AI Companion.
3. V aplikaci MIT AI2 Companion na vašem zařízení vyberte jednu z následujících možností:
Vyplňte kód a potvrďte jej kliknutím na connect with code, nebo naskenujte QR kód zařízením pomocí scan QR code.

Emulator

V případě, že nemáte telefon či tablet s operačním systémem Android, můžete použít emulátor, který simuluje zařízení Android. Nevýhodou je, že na emulátoru nelze testovat funkce senzorů.

1. Podle následujících instrukcí, které jsou dostupné na oficiálních stránkách MIT App Inventoru nainstalujte potřebný software pro váš operační systém:

Instrukce pro Mac OS X - <http://appinventor.mit.edu/explore/ai2/mac.html>

Instrukce pro Windows -

<http://appinventor.mit.edu/explore/ai2/windows.html>

Instrukce pro GNU/Linux - <http://appinventor.mit.edu/explore/ai2/linux.html>

2. Po dokončení instalace spusťte aiStarter.

3. V aplikaci MIT App Inventor z nabídky Connect vyberte položku Emulator, v programu aiStarter se načte simulátor Androidu a v něm aplikace, na které zrovna pracujete. Na OS Mac se aiStarter spustí na pozadí automaticky po přihlášení do App Inventoru.

USB

Pokud máte Android zařízení, ale nemáte Wi-Fi, můžete telefon připojit pomocí USB kabelu. Budete potřebovat software již zmíněný v předchozích způsobech. Pro připojení pomocí USB kabelu postupujte dle následujících kroků:

1. Podle následujících instrukcí, které jsou dostupné na oficiálních stránkách MIT App Inventoru nainstalujeme potřebný software na vybraný operační systém:

Instrukce pro Mac OS X - <http://appinventor.mit.edu/explore/ai2/mac.html>

Instrukce pro Windows -

<http://appinventor.mit.edu/explore/ai2/windows.html>

Instrukce pro GNU/Linux - <http://appinventor.mit.edu/explore/ai2/linux.html>

2. Do telefonu, nebo tabletu nainstalujte aplikaci MIT AI2 Companion. Pokud v případě absence internetového připojení v zařízení nemůžete aplikaci nainstalovat z Google Play, stáhněte instalační soubor MIT AI2 Companion do svého počítače, přeneste do svého zařízení a nainstalujte.

Instalační soubor stáhněte zde:

<http://ai2.appinventor.mit.edu/companions/MITAI2Companion.apk>

Je možné využít i tyto QR kódy:

Kód QR1 – stažení instalačního souboru apk z odkazu uvedeného výše

Kód QR2 – získání aplikace z Google Play



QR kód 1: QR1



QR kód 2: QR2

3. Spusťte aiStarter. Na OS Mac se spustí automaticky na pozadí po přihlášení do účtu.
4. Pro připojení přes USB kabel je nutné povolit ladění USB na telefonu či tabletu. Přejdeme do Nastavení > Informace o telefonu a 7krát poklepejte na číslo sestavení, dokud se vám nezobrazí „Jste v režimu vývojář“. Poté se vraťte do nastavení a zvolte Možnosti vývojáře, kde povolíte ladění USB.
5. Připojte zařízení k počítači a ujistěte se, že je připojeno v režimu „Velkokapacitní přenos“.
6. Otestujte připojení na webové stránce <http://appinventor.mit.edu/test/>. Pokud připojení nefunguje, řešení problému můžete nalézt zde: <http://appinventor.mit.edu/explore/ai2/connection-help.html>.

5.5. Přenos vytvořených aplikací do zařízení

Vytvořené aplikace je potřeba nějakým způsobem umístit a nainstalovat do mobilního zařízení. Aplikaci do telefonu či tabletu můžete stáhnout pomocí QR kódu, který vygenerujete pomocí kliknutím na Build, který se nachází na liště a vyberete možnost App (provide QR code for .apk), nebo stáhnete soubor .apk, kliknutím na App (save .apk to my computer) do počítače a následně ji přesunete a nainstalujete do telefonu.

6. Sada úloh s tutoriály pro MIT App Inventor

Tato sada úloh je určena jako pomůcka k výuce grafického programování aplikací v prostředí MIT App Inventoru ve školách, příručka pro samouky, e-learning. Cílem úloh je orientace v tomto programovacím prostředí, ukázka jeho možností a naučit jednoduše vytvářet mobilní aplikace na zařízení s operačním systémem Android s využitím jeho senzorů a jiných funkcí.

Předpokladem k těmto úlohám je znalost principu základního programování, například zkušenosti s programováním ve Scratch. Cílem sady úloh není výuka programování, ale orientace v prostředí AI2.

Každá úloha obsahuje popis aplikace a její funkce, seznam využitých komponent, základní cíle úlohy, zadání. Dále obsahuje tutoriál k aplikaci, kde zadání v krocích je doplněno detailním návodem ve formě tutoriálu, jak jednotlivé body splnit a s vysvětlením proč. Tutoriál je též doplněn ilustračními obrázky. Dále úloha obsahuje výsledný kód aplikace pro případnou kontrolu, či nápovědu.

Úlohy s tutoriály jsou vhodné jak pro samouky, k e-learningu, tak i pro učitele k výuce. Učitel se sám může rozhodnout, jak úlohy žákům zadá. Může si vybrat části úloh, které žákům předloží. Může jim například předložit popis, zadání v bodech a sám provést výklad a vysvětlit postup, nebo žáky nechat, ať sami hledají řešení, nebo jim dát základní popis a podrobný postup a nechat je aplikaci vytvořit podle návodu. Záleží na každém učiteli, který posoudí, na kolik mají žáci zkušenosti a znalosti s programováním. Pokud mají zkušenosti například se Scratch, nemusí pro ty žáky být po základním seznámením se s prostředím App Inventoru složité tyto aplikace vytvářet například s využitím podrobného zadání v krocích.

Při tvorbě některých úloh jsem se inspirovala aplikacemi z knihy App Inventor 2: Create your own apps od autorů David Wolber, Hal Abelson, Ellen Spertus a Liz Looney.

6.1. Seznam úloh

6.1.1 Hello World

Výpis textu – seznámení s prostředím MIT App Inventor

6.1.2 Virtuální kostka

Simulátor hrací kostky – využití akcelerometru, TextToSpeech a práce s obrázky

6.1.3 Pohlednice

Kreslení a tisk na fotografie – využití hodin, fotoaparátu, ukládání souborů

6.1.4 Průvodce

Průvodce městem – využití Google Maps a webového prohlížeče, více obrazovek

6.1.5 Kde jsi?

Aktuální poloha v SMS zprávě – využití aktuální adresy a příchozí zprávy

6.1.1. Hello World

Hello World je často prvním programem v různých programovacích jazycích, výjimka se tedy nebude dělat ani v MIT App Inventoru. Protože je to program velice jednoduchý, je snadné na něm pochopit princip programování a slouží jako ukázka syntaxe programovacího jazyku. Program vypíše na výstupní zařízení, nejčastěji na obrazovku „Hello World!“. V tomto případě to bude na displej vašeho telefonu, či tabletu. Pozdrav se bude vypisovat po stisknutí tlačítka.

Cíl úlohy:

- Seznámení s prostředím MIT App Inventor
- Vkládání, úprava komponent a jejich využití
- Vkládání bloků a jejich skládání
- První spuštění a testování aplikace

Zadání:

Vytvořte a spusťte aplikaci, která po stisknutí tlačítka vypíše na displej „Hello World!“. Vhodně pojmenujte komponenty a upravte uživatelské prostředí.

Komponenty:

- Label
- Button

Poznámky:

Je dobré vytvořit si návyk vhodně pojmenovávat komponenty, tento návyk bude užitečný později v případě složitějších aplikací.

Hello World - zadání v krocích:

- A.** Založte nový projekt a pojmenujte „helloWorld“.
- B.** Přejmenujte Title u Screen1
- C.** Do prostoru obrazovky telefonu přetáhněte z Palette, z oblasti User Interface komponenty Label a Button.
- D.** Přejmenujte v panelu Components Label1 na „helloLabel“ a Button1 na „clickButton“
- E.** V Properties nastavte tlačítku clickButton text na „Click!“ a width (šířka komponenty) na Fill parent. Smažte text u helloLabel.
- F.** Nyní se přepněte do části Blocks.
- G.** Přetáhněte do prázdného prostoru Viewer blok komponenty clickButton when clickButton.Click do.
- H.** Do bloku when clickButton.click přetáhněte blok patřící pod helloLabel set helloLabel.Text to.
- I.** Nastavte hodnotu labelu helloLabel na text „Hello World!“.
- J.** Otestujte aplikaci.

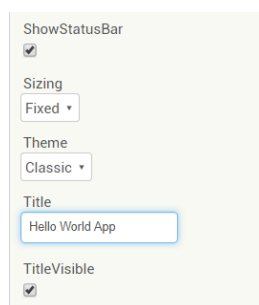
Hello World - podrobný postup:

A. Založte nový projekt a pojmenujte „helloWorld“.

Nový projekt vytvoříte kliknutím na Start new project vlevo na tmavě zelené liště. Vyplňte jeho název – „helloWorld“. Ten může být složen z písmen, čísel a podtržítok. Mezery a ostatní znaky nejsou povoleny. Ideálně tedy využijeme Camel Case, neboli „Velbloudí písmo“, kde každé další slovo začíná velkým písmenem. Projekty později nelze přejmenovat.

B. Přejmenujte Title u Screen1

V pravém panelu Properties se upravují různé vlastnosti komponent. Například jejich text, velikost, zarovnání, barvy apod. Zde změníte i Title u obrazovky Screen1, tedy nadpis v záhlaví obrazovky aplikace.

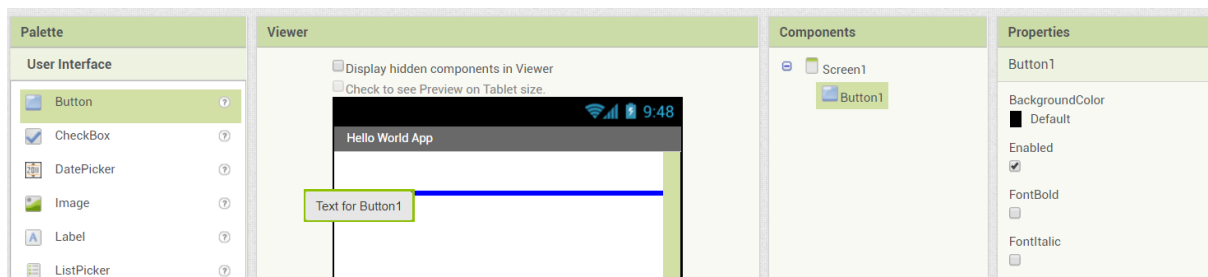


Obrázek 6.1.1: Ukázka z panelu Properties

C. Do prostoru obrazovky telefonu přetáhněte z Palette, z oblasti User Interface komponenty Label a Button.

V levém panelu Palette najdete všechny komponenty, které můžete v aplikaci využít. Jsou rozdělené na viditelné (např. tlačítka, obrázky, popisky) a neviditelné (např. senzory).

Komponenty se přetahují stejným způsobem, jako například ikony na ploše počítače.



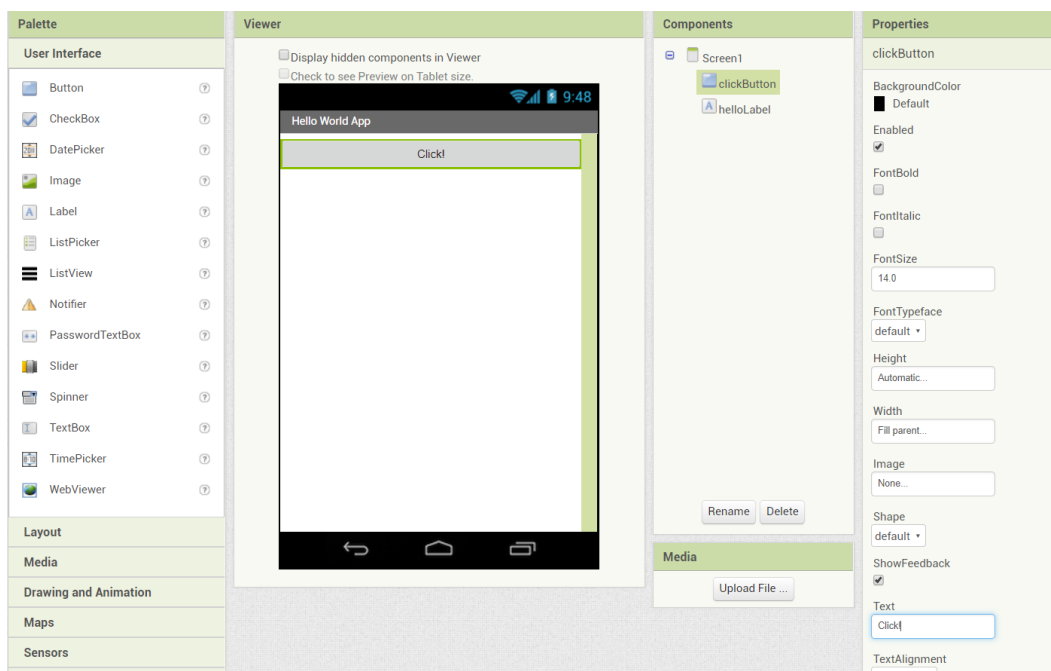
Obrázek 6.1.2: Ukázka přesunu komponenty do Vieweru

D. Přejmenujte v panelu Components Label1 na „helloLabel“ a Button1 na „clickButton“

Komponenty přejmenujete kliknutím na Rename v panelu Components. Názvy komponent jsou pouze doporučené.

E. V Properties nastavte tlačítku clickButton text na „Click!“ a width (šířka komponenty) na Fill parent. Smažte text u helloLabel.

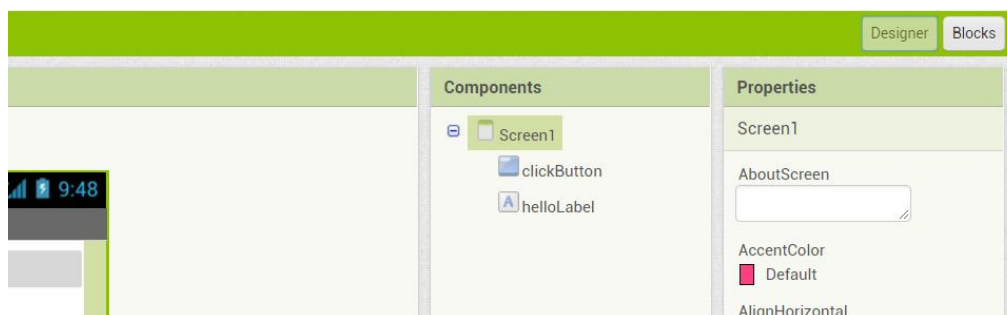
Šířku nebo výšku komponent je možné nastavit automaticky, v pixelech, v procentech displeje, nebo na Fill parent, což znamená, že šířka bude totožná s šířkou displeje, či layoutu. Můžete také změnit například barvy, velikost, řez písma, tvar tlačítka apod. Veškeré změny vlastností provedete v panelu Properties, nebo s využitím bloků.



Obrázek 6.1.3: Ukázka splněného kroku E

F. Nyní se přepněte do části Blocks.

Přepnutí najdete v pravém horním rohu, nad panelem Properties. Zde naprogramujete aplikaci její chování skládáním předpřipravených bloků, které budete přetahovat tak, aby do sebe zapadaly.



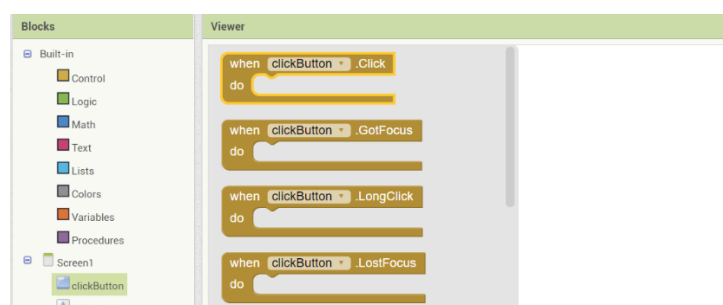
Obrázek 6.1.4: Přepínání mezi Designer a Blocks

G. Přetáhněte do prázdného prostoru Viewer blok komponenty `clickButton` when `clickButton.Click` do.

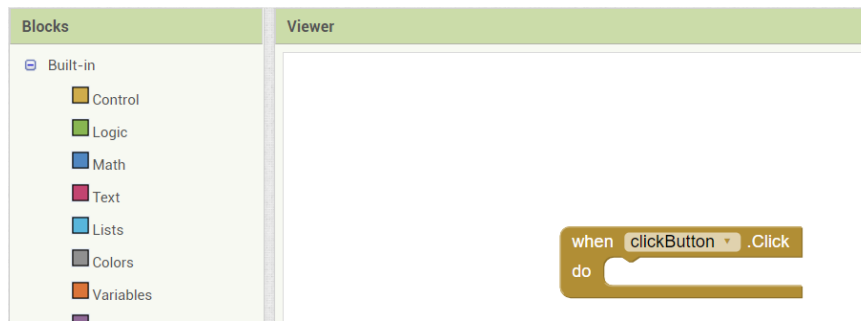
Každá komponenta má předpřipravené bloky, s jejichž využitím je možné s komponentou pracovat.

Blok when `clickButton.Click` do naleznete v levém panelu Blocks, po rozkliknutí `clickButton`. Objeví se všechny bloky, procedury patřící k této komponentě.

Tento blok aplikaci říká, co má udělat po kliknutí na tlačítko `clickButton`. Bloky patřící k tlačítkům umožňují nastavit, jaký úkol se má provést například při dlouhém stisknutí tlačítka, při uvolnění tlačítka apod. Chování po stisknutí tlačítka nastavíte v následujícím kroku.



Obrázek 6.1.5: Ukázka bloků komponenty Button



Obrázek 6.1.6: Ukázka splněného kroku G

H. Do bloku `when clickButton.click` přetáhněte blok patřící pod `helloLabel set helloLabel.Text to`.

Výhoda MIT App Inventoru je, že hned poznáme, když k sobě části kódu nepatří, v takovém případě do sebe bloky nepasují.

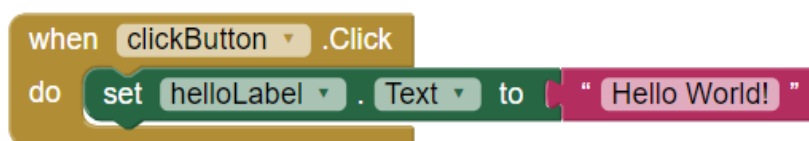
Bloky `set` nastavují hodnotu. Nastavit můžete například vlastnosti komponent, text, číslo, či logický datový typ `True` nebo `False`. V tomto případě se nastavuje text labelu `helloLabel`, tedy to, co se v aplikaci vypíše na místě, kde se na obrazovce nachází komponenta `label`. Text můžete nastavit i v Designeru tam, kde jste nastavili text prázdný. To proto, že v této aplikaci je třeba nastavit text až po stisknutí tlačítka, nastaví se tedy až pomocí kódu.

I. Nastavte hodnotu labelu `helloLabel` na text „Hello World!“.

Z levého panelu `Blocks` ze sekce `Text` přetáhněte blok s uvozovkami “ ” a připojte k bloku `set helloLabel.text to`. Do uvozovek doplňte „Hello World!“.

Právě jste naprogramovali první aplikaci v MIT App Inventoru. Výsledný kód znamená:

Když se stiskne tlačítko s názvem `clickButton`, nastaví se text labelu `helloLabel` a vypíše se „Hello World!“.



Obrázek 6.1.7: Výsledný kód Hello World

J. Otestujte aplikaci.

Aplikaci je třeba vyzkoušet na Android telefonu nebo tabletu, pomocí aplikace MIT AI2 Companion, s využitím USB kabelu, či pokud nemáte žádné Android zařízení, můžete využít Android emulátor. viz. kapitola Možnosti ladění, testování

Aplikaci si také můžete stáhnout a nainstalovat do svého zařízení viz. kapitola Přenos vytvořených aplikací do Android zařízení.

6.1.2. Virtuální kostka

Aplikace Virtuální kostka, jak už název napovídá, simuluje hody hrací kostky. Po zatřesení zařízením se na displeji objeví obrázek vykreslující stranu hrací kostky s náhodně vygenerovaným číslem pomocí funkce `Random integer` a zařízením vám s využitím funkce `TextToSpeech` nahlas řekne, jaké číslo bylo „hozeno“.

Cíl úlohy:

- Využití senzoru akcelerometr zařízení
- Využití komponenty `TextToSpeech` k převodu textu na řeč
- Práce s obrázky, jejich nahrání a využití
- Inicializace, deklarace a využití proměnné
- Náhodný výběr čísla
- Využití bloku `join`

Zadání:

Naprogramujte simulaci hrací kostky. Zatřesení telefonem bude mít stejný význam, jako reálný hod kostkou. Na displej se nastaví obrázek se stranou kostky, kde se počet teček bude rovnat vygenerovanému číslu a zároveň zařízením toto číslo „vysloví“. Využijte blok `Random integer`, `Accelerometr` a `TextToSpeech`.

Komponenty:

- `Label`
- `Image`
- `TextToSpeech`

Poznámky:

Aplikaci je vhodné testovat po přidání nové funkce. Bude snazší odhalit případnou chybu.

Virtuální kostka - postup v krocích

- A.** Vytvořte nový projekt a pojmenujte jej „hraciKostka“. Změňte text Title u Screen1 na „Virtuální kostka“.
- B.** Připravte k použití komponenty Label, Image, AccelerometerSensor a TextToSpeech.
- C.** Label zarovnejte na střed a změňte text na „Zatřes se mnou a hod' kostkou!“.
- D.** Nastavte šířku komponenty Image na Fill parent.
- E.** Stáhněte do počítače a nahrajte do aplikace obrázky stěn kostky
- F.** V Blocks inicializujte (vytvořte) proměnnou randomCislo typu Integer.
- G.** Připravte k použití blok určující chování aplikace při zatřesení zařízením.
- H.** Po zatřesení zařízením nastavte proměnnou randomCislo na náhodně vygenerované číslo 1 až 6.
- I.** Nastavte vhodně Picture komponenty Image na obrázek, odpovídající randomCislo.
- J.** Otestujte aplikaci
- K.** V tomto kroku zajistěte pomocí TextToSpeech, aby zařízení vyslovilo číslo hodu kostkou.
- L.** Otestujte a nainstalujte aplikaci do svého zařízení.

Virtuální kostka - podrobný postup

A. Vytvořte nový projekt a pojmenujte jej „hraciKostka“. Změňte text Title u Screen1 na „Virtuální kostka“.

B. Připravte k použití komponenty Label, Image, AccelerometerSensor a TextToSpeech.

Přetáhněte do prostoru obrazovky telefonu. Label a Image získáte v levém panelu Palette v sekci User Interface. AccelerometerSensor, je umístěný v Sensors a TextToSpeech komponenta, která mění text na řeč naleznete v sekci Media. AccelerometerSenzor umožňuje zařízení zjistit, když je se zařízením zatřesen, změni orientaci displeje apod.

C. Label zarovnejte na střed a změňte text na „Zatřes se mnou a hod' kostkou!“.

V pravém panelu Properties změňte Text, nastavte Width u Label1 na Fill parent a zarovnejte text na střed změnou textAligment na center:1. Zarovnání lze také nastavit nalevo, nebo napravo.

Jelikož jsou komponenty v této aplikaci jednoznačné, vyskytují se pouze jednou, není nutné je přejmenovávat.

D. Nastavte šířku komponenty Image na Fill parent.

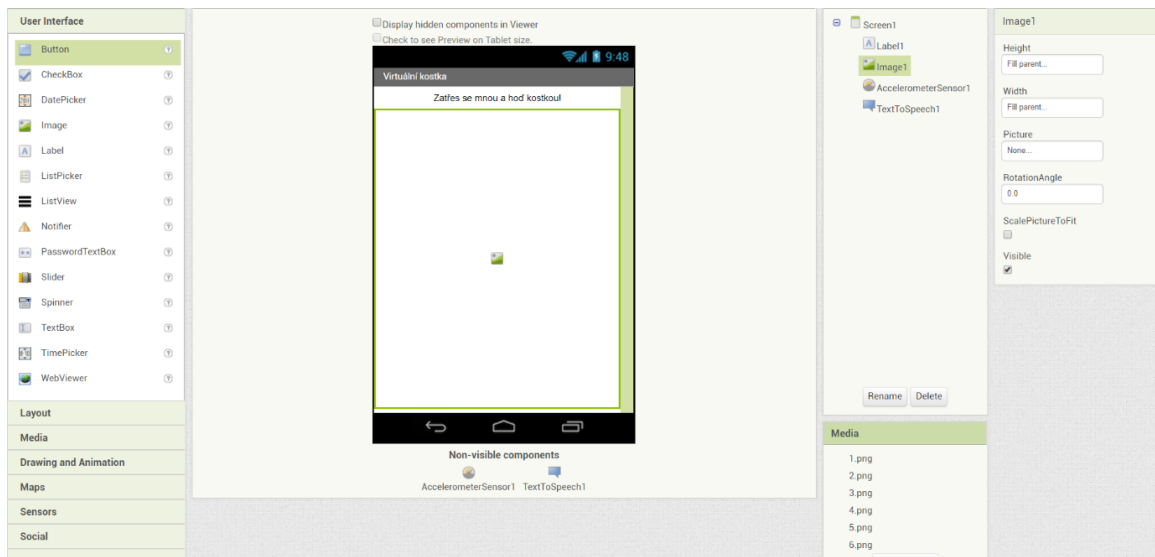
V Properties změňte hodnotu Width na Fill parent.

E. Stáhněte do počítače a nahrajte do aplikace obrázky stěn kostky

Budete potřebovat obrázky, které zobrazí stěny kostky. Takové obrázky si sami můžete vytvořit v nějakém grafickém editoru, nebo stáhnout například z <https://openclipart.org/>.

Obrázky pojmenujte podle počtu teček, který zobrazují. Tzn. obrázek stěny s jednou tečkou, se jmenuje 1 apod.

Aby se mohli obrázky použít, musíte je nejprve nahrát do panelu Media, který se nachází pod panelem Components.



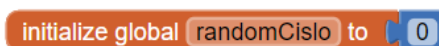
Obrázek 6.1.8: Výsledek práce v Designeru

F. V Blocks inicializujte (vytvořte) proměnnou `randomCislo` typu Integer.

Proměnná slouží k uchování hodnoty, informace, kterou můžete opakovaně používat. Může uchovávat například číslo, text, či logický datový typ `True` nebo `False`. Proměnná `randomCislo` bude typu `Integer`, protože bude uchovávat náhodně vygenerované číslo. Toto číslo bude simulovat hod kostky.

Proměnné najdete v panelu `Blocks` jako `Variables`, odtud přetáhněte blok `initialize global name to`. Název proměnné `name` přepište na „`randomCislo`“. Tímto je proměnná inicializována. Nyní je třeba nastavit typ proměnné na číslo. Ze sekce `Math` přetáhněte a přidejte k předchozímu bloku první položku ze seznamu, blok, ve kterém je zapsána nula.

V sekci `Math` najdete různé bloky, které můžete využít k různým početním operacím



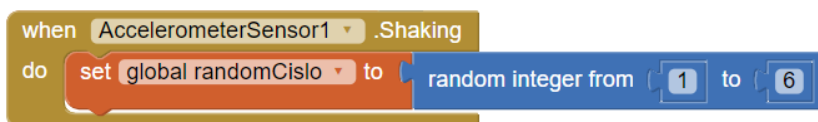
Obrázek 6.1.9: Inicializace proměnné `randomCislo`

G. Připravte k použití blok určující chování aplikace při zatřesení zařízení.

K tomuto je třeba přetáhnout do pracovního prostoru Viewer blok `when AcceleratorSensor1.Shaking do`, který naleznete po rozkliknutí `AcceleratorSensor1` v panelu `Blocks`.

H. Po zatřesení zařízením nastavte proměnnou `randomCislo` na náhodně vygenerované číslo 1 až 6.

Ze sekce `Variables` přetáhněte do připravené události z předchozího kroku blok `set to` a po rozkliknutí malého trojúhelníku vyberte proměnnou `randomCislo`. K tomuto bloku připojte ze sekce `Math` blok `random integer from` a změňte hodnoty bloku na 1 a 6. To znamená, že program bude náhodně vybírat číslo z rozmezí od 1 do 6 včetně. Vygenerované číslo se uloží do proměnné `randomCislo` k dalšímu využití.



Obrázek 6.1.10: Náhodný výběr čísla po zatřesení zařízením

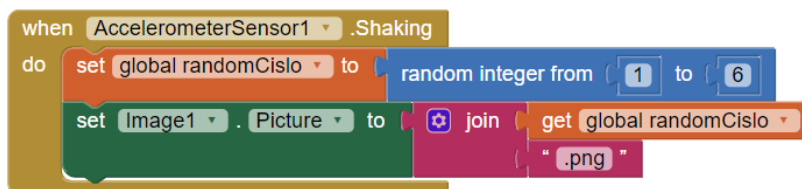
I. Nastavte vhodně `Picture` komponenty `Image` na obrázek, odpovídající `randomCislo`.

Je třeba naprogramovat blok tak, aby se na displeji zobrazila stěna kostky s počtem teček, který se rovná hodnotě `randomCislo`, tedy virtuálnímu hodu kostky. Využijte toho, že jsou obrázky pojmenované čísly, která odpovídají počtu teček stěny, které vyobrazují.

Přidejte pod blok, který nastavuje `randomCislo` blok s metodou `setImage1.Picture to`, která patří ke komponentě `Image1` z panelu `Blocks`. `Picture`, tedy obrázek, který zobrazuje komponenta `Image1` nastavte na název složený z `randomCislo` a koncovky, která vyjádří formát obrázku. Například, když se `randomCislo` rovná 1, název obrázku se nastaví na `1.png` a obrázek se stejným názvem se vybere z knihovny médií, kam předtím byly nahrány. Ke spojení těchto hodnot potřebujete spojující blok `join`, který je v sekci `Text`. V případě potřeby můžete rozkliknutím na ozubené kolečko přidat prostor pro další hodnoty.

Do prvního „zubu“ komponenty join přidejte ze sekce Variables blok `get` a vyberte proměnnou `randomCislo`. Do druhého přidejte blok s formátem vašeho obrázku, pravděpodobně „.png“.

Procedury `get` se používají, když je nutné získat a použít nějakou hodnotu, proměnnou nebo informaci. V tomto případě získáváte číslo uložené v proměnné `randomCislo`.



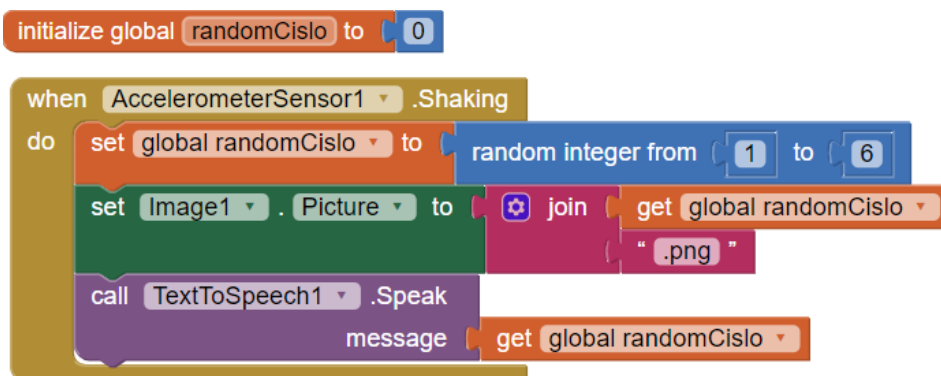
Obrázek 6.1.11: Kroky G - I

J. Otestujte aplikaci

Nyní je vhodná doba na to, otestovat aplikaci a zjistit, zda vše funguje, jak má, případně opravit chyby.

K. V tomto kroku zajistěte pomocí TextToSpeech, aby zařízení vyslovilo číslo hodu kostkou.

Přidejte proceduru `call TextToSpeech.Speak message` komponenty `TextToSpeech` pod ostatní bloky. Tento blok zavolá předpřipravenou proceduru, která zařídí, aby zařízení přehráló určenou zprávu či text. Zavolat proceduru znamená, říct programu, aby se procedura vykonala. V této funkci má zařízení vyslovit číslo hodu kostky, přidejte tedy do bloku proceduru `get randomCislo`.



Obrázek 6.1.12: Výsledný kód Virtuální kostka

L. Otestujte a nainstalujte aplikaci do svého zařízení.

6.1.3. Pohlednice

Vytvořte si aplikaci, která bude sloužit ke tvorbě pohlednic z fotografií, které vyfotíte. Aplikace umožní vámi nastavený text otisknout na místo vašeho dotyku na displeji a také na fotografii kreslit několika barvami. Tato aplikace bude mít funkce fotit, nahrát nové plátno, uložit obrázek do telefonu a ukončit aplikaci.

Cíl úlohy:

- Seznámení s layouty
- Využití fotoaparátu zařízení
- Práce s komponentou Canvas
- Ukládání souborů přes aplikaci
- Získání aktuálního času a datumu

Zadání:

Vytvořte uživatelské prostředí dle ukázky. Naprogramujte aplikaci, která bude využívat komponentu Canvas na níž bude možné kreslit a tisknout text, který se bude měnit pomocí textového pole. Tlačítko kamera spustí fotoaparát a vyfocený snímek se nastaví jako pozadí plátna. Tlačítko nový vytvoří nové plátno. Zajistěte, aby aplikace měla funkce ukládat obrázky s názvem složeným i aktuálního datumu a času. Stisknutím ukončit ukončete aplikaci.

Komponenty:

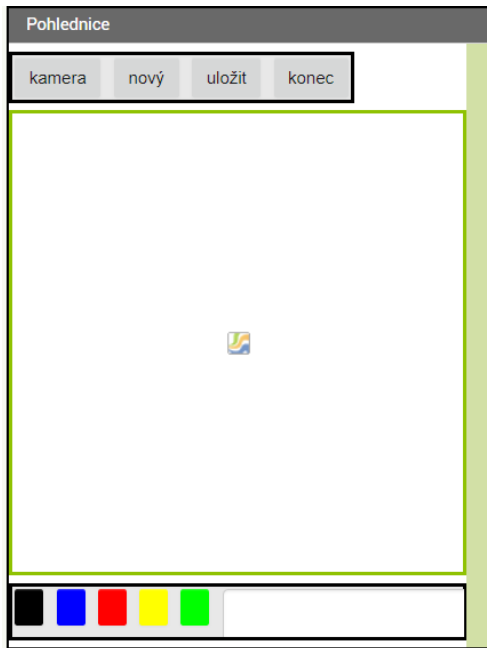
- HorizontalArrangement
- Button
- Canvas
- TextBox
- Camera
- Clock

Poznámky:

Bloky se mohou duplikovat kliknutím na blok pravým tlačítkem myši, nebo lze využít i klávesových funkcí ctrl + c ke kopírování a ctrl + v k vložení.

Pohlednice - zadání v krocích

- A.** Vytvořte nový projekt a pojmenujte jako „pohlednice“, stejný název bude mít i Title u Screen1.
- B.** Vytvořte uživatelské prostředí podle ukázky viz. obrázek 7.3.1

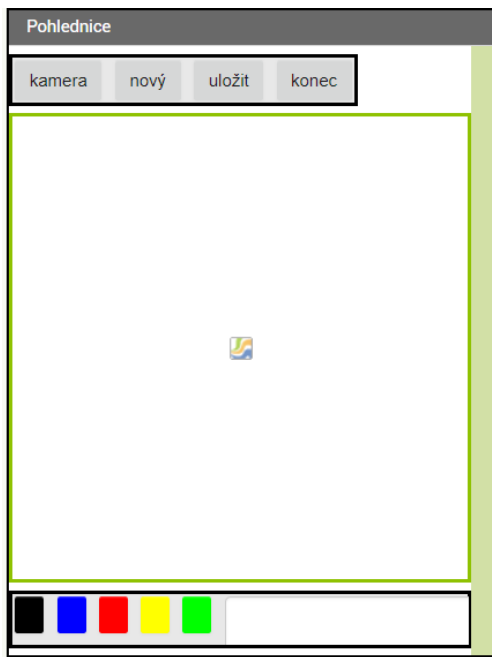


Obrázek 6.1.13: Uživatelské rozhraní Pohlednice

- C.** Po spuštění aplikace zajistěte v Blocks spuštění fotoaparátu a možnost focení.
- D.** Nastavte vyfocenou fotografii na pozadí plátna Canvas1.
- E.** Zařídte, aby se v aplikaci mohlo kreslit táhnutím prstem po displeji.
- F.** Zajistěte, aby se po stisknutí na barevná tlačítka kreslilo na plátno zvolenými barvami.
- G.** Naprogramujte tlačítkům kamera, nový a ukončit jejich funkce.
- H.** Zajistěte, aby se při kliknutí do plátna otisknul text na totožné místo.
- I.** Naprogramujte funkci tlačítka uložit.

Pohlednice - podrobný postup

- A.** Vytvořte nový projekt a pojmenujte jako „pohlednice“, stejný název bude mít i Title u Screen1.
- B.** Vytvořte uživatelské prostředí podle ukázky viz. obrázek 7.3.1



Obrázek 6.1.14: Uživatelské rozhraní Pohlednice

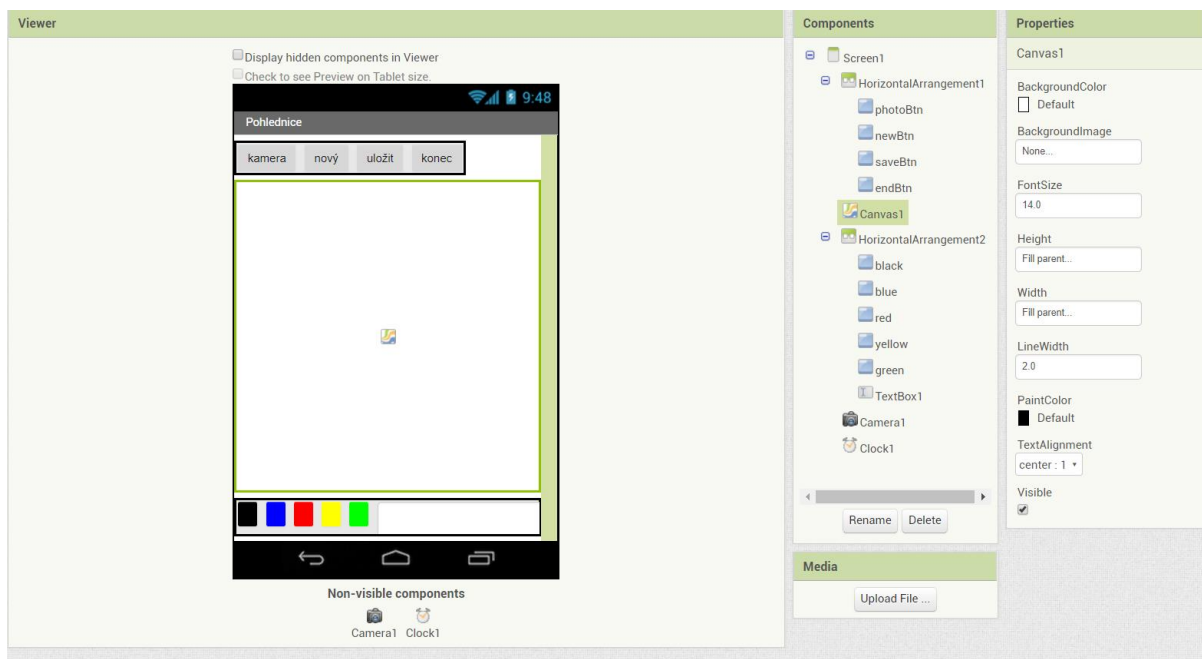
Nejprve na obrazovku přesuňte layout `HorizontalArrangement`. Do tohoto layoutu vložte 4x `Button` a nastavte text na „kamera“, „nový“, „uložit“, „ukončit“, tato tlačítka podle jejich textu přejmenujte na „photoBtn“, „newBtn“, „saveBtn“, „endBtn“.

Díky layoutům můžete seřazovat komponenty vedle sebe, nebo pod sebe.

Pod layout přidejte `Canvas`, komponentu, která představuje plátno, na které se dá např. kreslit. Naleznete jej sekci `Drawing and Animation`. Šířku a délku plátna nastavte na `Fill parent`.

Pod `Canvas` přidejte další `HorizontalArrangement` layout a do něj vložte 4 x `Button` A `TextBox`. Prvním 4 tlačítkům vymažte jejich text a nastavte jim barevné pozadí dle vašeho výběru. Například černé, modré, červené a žluté. Přejmenujte tlačítka tak, aby jejich názvy odpovídaly barvám jejich pozadí.

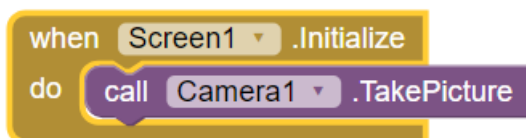
Přidejte do aplikace neviditelné komponenty Camera a Clock. Camera je v sekci Media a Clock najdete mezi senzory.



Obrázek 6.1.15: Výsledek práce v Designeru

C. Po spuštění aplikace zajistěte v Blocks spuštění fotoaparátu a možnost focení.

Potřebujete blok, který definuje chování aplikace ihned po jejím spuštění, respektive po načtení její obrazovky. Blok naleznete právě u procedur `Screen1` s popisem `when Screen1.Initialize` do. Při načtení obrazovky potřebujete zavolat proceduru spuštění kamery a focení, příkaz `call Camera1.TakePicture` je mezi bloky komponenty `Camera1`.

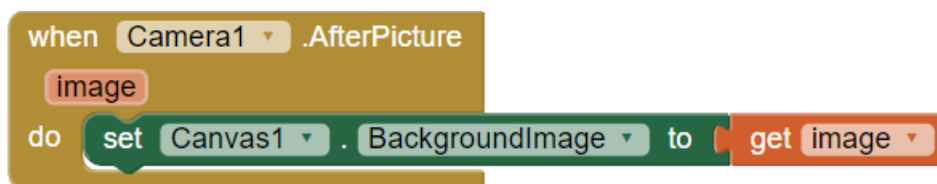


Obrázek 6.1.16: Krok C

D. Nastavte vyfocenou fotografii na pozadí plátna `Canvas1`.

Další procedura komponenty `Camera1` je `when Camera1.AfterPicture` do s parametrem `Image`. Tato procedura určuje, které bloky se vykonají po pořízení snímku, který je uložený v parametru této procedury v `image`.

Do této procedury přidejte blok patřící ke komponentě Canvas1 set Canvas1.BackgroundImage. to a doplňte blokem get image, který získáte kliknutím na parametr image. Tímto se pořízená fotografie po potvrzení automaticky nastaví na pozadí plátna Canvas1.

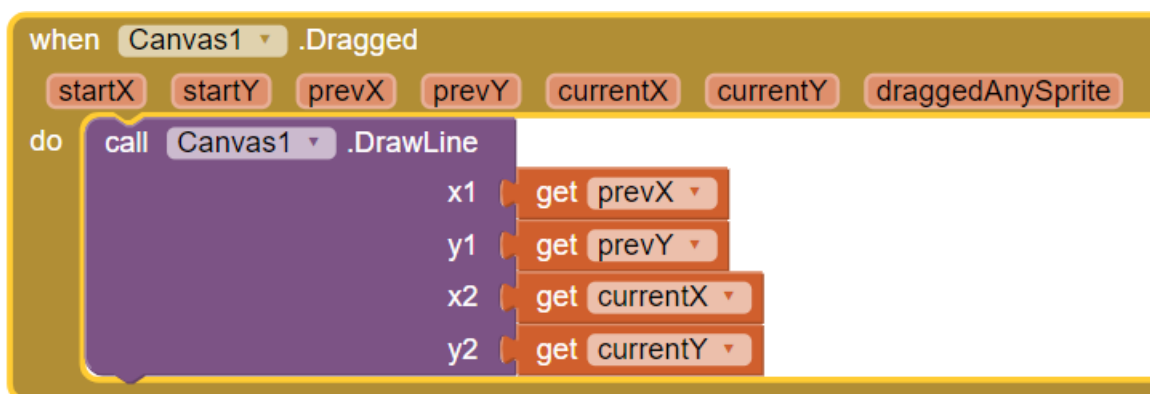


Obrázek 6.1.17: Krok D

E. Zařídte, aby se v aplikaci mohlo kreslit táhnutím prstem po displeji.

Tuto funkci provedete pomocí bloku when Canvas1.dragged do. Funkce má několik parametrů, v této aplikaci budete potřebovat prev X, prev Y, current X a current Y, kde parametry prev x,y určují předchozí souřadnice doteku na displeji a current x,y souřadnice následující.

Aby tah začal po táhnutí prstem či stylusem kreslit čáru, je třeba v tomto bloku zavolat proceduru call Canvas1.DrawLine. Přidejte do procedury získané parametry, kde k souřadnicím x_1 a y_1 patří bloky get prev X a get prev Y a x_2 a y_2 přiřaďte bloky get current X a current Y.



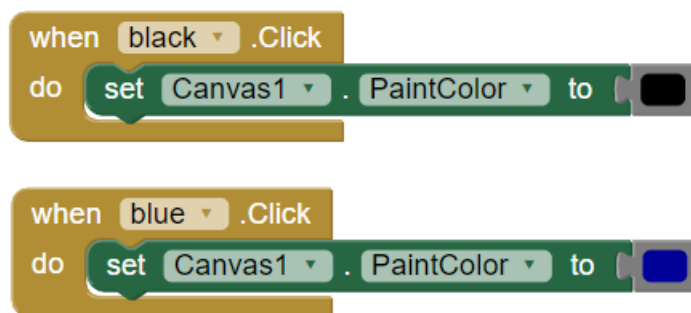
Obrázek 6.1.18: Krok E

F. Zajistěte, aby se po stisknutí na barevná tlačítka kreslilo na plátno zvolenými barvami.

Komponentě plátna Canvas1 se může nastavit barva kreslení pomocí procedury set Canvas1.PaintColor a bloku ze sekce Color. Přiřaďte ke každému barevnému

tlačítku proceduru s nastavením odpovídající barvy. Například k tlačítku červené, do bloku `when red.Click` do přidáme proceduru `set`

`Canvas1.PaintColor` to a nastavíme na červenou barvu ze sekce Colors. Tyto kroky zopakujte, dokud nebudete mít nastavena všechna barevná tlačítka.



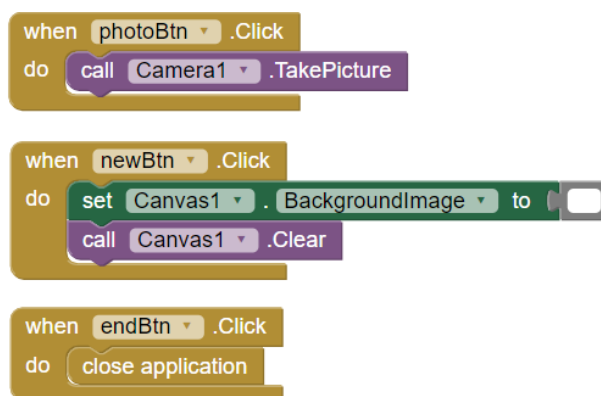
Obrázek 6.1.19: Krok F

G. Naprogramujte tlačítkům kamera, nový a ukončit jejich funkce.

Tlačítko pro spuštění kamery nastavíme stejným způsobem, jako ve třetím kroku, zavoláním procedury `call Camera1.TakePicture`. Procedura se zavolá z bloku `when Camera1.Click` do. Nastavení fotografie na pozadí plátna je již nastavené z předchozích kroků.

Tlačítko nový nastavuje nové, bílé pozadí komponentě `Canvas1` pomocí procedury `set Canvas1.BackgroundImage` to a k ní přidáme bílou barvu, případně jinou. Tímto se nahradí předchozí fotografie novým plátnem, zbývá ještě vymazat již nakreslené čáry a text, zavoláním procedury `call Canvas1.Clear`.

Tlačítko ukončit získá funkci tím, že do bloku `when endBtn.Click` do přidáte příkaz `close application`, který se nachází v sekci Control.

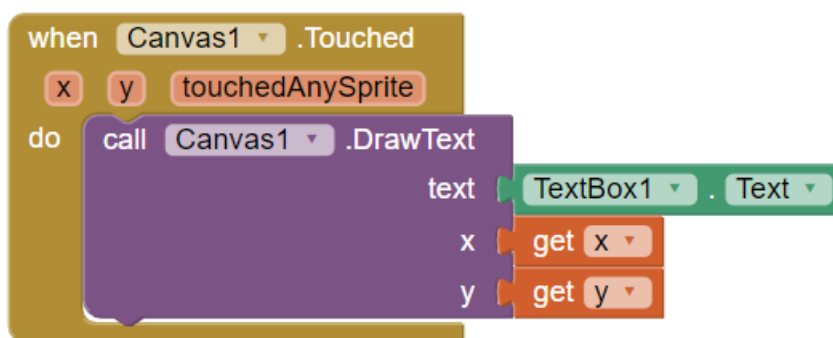


Obrázek 6.1.20: Krok G

H. Zajistěte, aby se při kliknutí do plátna otisknul text na totožné místo.

Tato funkce je podobná kreslení táhnutím, s tím rozdílem, že místo `Dragged` bude `Touched` a `DrawLine` nahradí `DrawText`.

Vyberte příkaz `when Canvas1.Touched` do a doplňte procedurou `call Canvas1.DrawText`. K souřadnicím `x` a `y` přidejte bloky `get y` a `get x`, které uloží souřadnice místa doteku. K textu přiřadte blok, který patří ke komponentě `TextBox1`, `textBox1.Text`. Tento blok předá text z textového pole `TextBox1`. Funguje podobně, jako procedura `get`.



Obrázek 6.1.21: Krok H

I. Naprogramujte funkci tlačítka uložit.

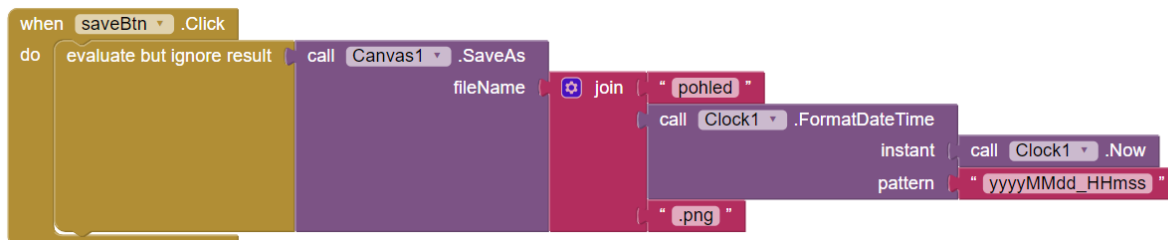
Vytvořené obrázky je třeba po dokončení uložit, aby to bylo možné, do bloku `when saveBtn.Click` přidejte blok, který naleznete v sekci `Control evaluate but ignore result`. Tento blok se používá, když je třeba zavolat proceduru, aby se vykonala, ale nepotřebujete znát její návratovou hodnotu, její výsledek. K tomuto bloku připojte proceduru `call Canvas1.SaveAs file`.

Pokud obrázky budeme ukládat pod stejným názvem, může nastat problém, protože se obrázky budou přepisovat. To znamená, že obrázek nahradí dříve uložený obrázek. Je tedy třeba názvy ukládaných obrázků jednoznačně definovat. K tomu využijte aktuální čas a datum, získaný pomocí senzoru `Clock`.

Pomocí bloku `join` spojte následující 3 hodnoty, text, s výchozím názvem obrázku, například „pohled“, časový údaj, který získáte zavoláním procedury `call Clock.FormatDateTime` s parametry `instant` a `pattern`. K `instant` přiřadte volání procedury, která přiřadí parametru čas a datum v okamžiku zavolání procedury, tedy v okamžiku kliknutí na tlačítko uložit. K parametru `pattern` je předpřipravený

text s formátem času a datumu, ten upravte podle požadovaného tvaru, například na yyyyMMdd_HHmss. Poslední hodnotou bude text s formátem obrázku, např. „.png“.

Třetí „zub“ bloku přidáme kliknutím na ozubené kolečko, přetažením string do zobrazeného bloku napravo



Obrázek 6.1.22: Krok I

Pohlednice - výsledný kód

```
when Screen1.Initialize
do call Camera1.TakePicture

when Camera1.AfterPicture
image
do set Canvas1.BackgroundImage to get image

when Canvas1.Dragged
startX startY prevX prevY currentX currentY draggedAnySprite
do call Canvas1.DrawLine
    x1 get prevX
    y1 get prevY
    x2 get currentX
    y2 get currentY

when Canvas1.Touched
x y touchedAnySprite
do call Canvas1.DrawText
    text TextBox1.Text
    x get x
    y get y

when black.Click
do set Canvas1.PaintColor to black

when blue.Click
do set Canvas1.PaintColor to blue
```

```
when red .Click
do set Canvas1 . PaintColor to [red]
```

```
when yellow .Click
do set Canvas1 . PaintColor to [yellow]
```

```
when green .Click
do set Canvas1 . PaintColor to [green]
```

```
when photoBtn .Click
do call Camera1 .TakePicture
```

```
when newBtn .Click
do set Canvas1 . BackgroundImage to [image]
call Canvas1 .Clear
```

```
when endBtn .Click
do close application
```

```
when saveBtn .Click
do evaluate but ignore result call Canvas1 .SaveAs
  fileName join [ "pohled "
  call Clock1 .FormatDateTime
  instant call Clock1 .Now
  pattern "yyyyMMdd_HHmss "
  ".png "
```

6.1.4. Průvodce

Představte si, že za vámi přijede váš známý na návštěvu, ale vy nemáte čas se mu věnovat celý den, abyste ho provedli po městě. V takovém případě by se mohl hodit průvodce, a tím bude tato aplikace. Po výběru z několika možností zobrazí informace o památce pomocí zabudovaného webového prohlížeče a jedním kliknutím zobrazí polohu památky na mapě, s využitím mobilní aplikace Google maps, odkud je možné dalším kliknutím zobrazit trasu a spustit navigaci, nebo vrátí zpět na úvodní stránku.

Cíl úlohy:

- Práce s více obrazovkami a jejich sdílení informací
- Využití webového prohlížeče v aplikaci
- Spouštění aktivity – jiné služby v rámci této aplikace
- Ukládání informací do malé databáze a její využití
- Manipulace se seznamy

Zadání:

Naprogramujte aplikaci, která bude sloužit jako průvodce vybraného města. Aplikace bude obsahovat dvě obrazovky. Na úvodní obrazovce bude libovolný obrázek města, a pod ním výběr ze seznamu míst ve městě. Po výběru položky se objeví druhá obrazovka, na které bude s využitím WebViewera zobrazena webová stránka s informacemi o vybraném místě a pod ním budou dvě tlačítka. Jedno, které spustí další aktivitu – aplikaci Google maps v rámci vaší aplikace, kde bude již vyhledáno vybrané místo a druhé tlačítko zajistí návrat na úvodní stránku. Ke spuštění další aktivity použijte ActivityStarter.

Komponenty:

- Label
- Image
- ListPicker
- TinyDB
- Button
- WebViewer
- ActivityStarter

Poznámky:

Aplikaci testujte pravidelně, nejlépe vždy po přidání nové funkce. Bude snazší odhalit případnou chybu.

Průvodce - zadání v krocích

- A.** Založte nový projekt nazvaný „pruvodce“. Vyberte si město, o kterém bude tato aplikace.
- B.** Na Screen1 vytvořte uživatelské prostředí podle ukázky viz níže.
- C.** Nahrajte tematickou fotografii do Image, nastavte výšku i šířku na Fill parent.
- D.** Nastavte popisku Label2 text na „Kam se chcete podívat?“ a ListPickeru na „Vybrat“.
- E.** Vytvořte seznam položek „mista“ s názvy památek nebo míst ve městě.
- F.** Nastavte položky ListPicker1 na položky seznamu v proměnné mista.
- G.** Přidejte další obrazovku Screen2.
- H.** Vložte do uživatelského prostředí obrazovky Screen2 TinyDB, ActivityStarter, WebViewer a pod něj vložte tlačítko „Vyhledat v Google maps“. Výchozí titulek obrazovky vymažte.
- I.** Přepněte se zpět na Screen1 a zajistěte načtení druhé obrazovky po výběru ze seznamu.
- J.** Přepněte se do Screen1. Uložte index vybrané položky ze seznamu do databáze s tagem „selection“.
- K.** Ve Screen2 inicializujte proměnnou vyber a nastavte ji na hodnotu selection z databáze.
- L.** Vytvořte seznam totožný se seznamem s místy ze Screen1.
- M.** Nastavte titulek druhé obrazovky na název vybraného místa

N. Vytvořte seznam s odkazy.

O. Nastavte ve WebVieweru odkaz webu s informacemi o památce dle výběru.

P. Zajistěte funkčnost tlačítka mapsBtn, vyhledání na mapě.

Průvodce - podrobný postup

A. Založte nový projekt nazvaný „pruvodce“. Vyberte si město, o kterém bude tato aplikace.

Jako ilustrační případ bude v této úloze sloužit město České Budějovice.

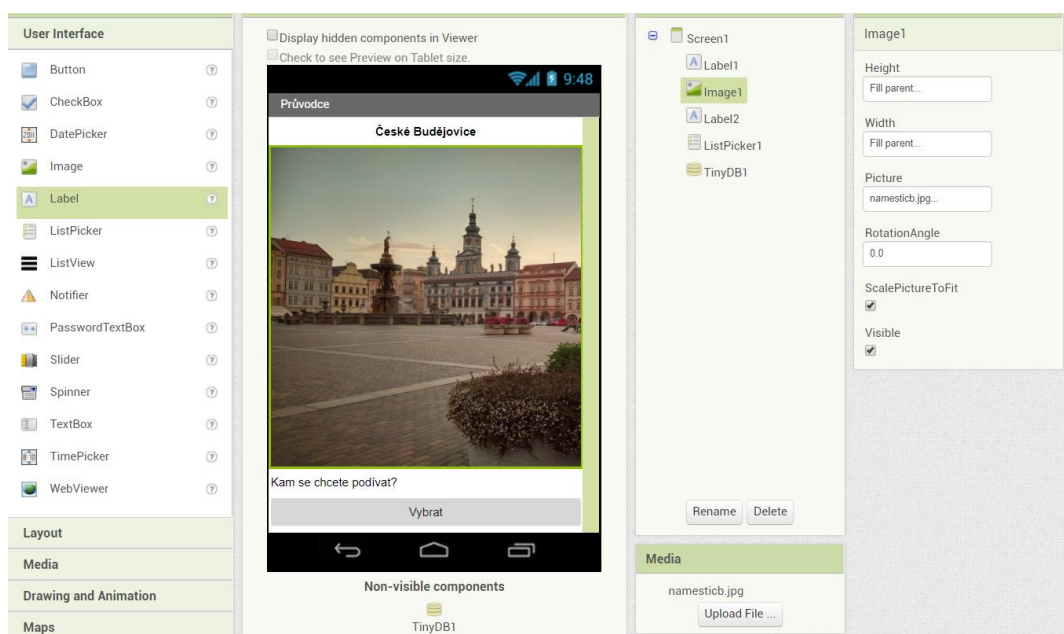
B. Na Screen1 vytvořte uživatelské prostředí podle ukázky viz níže.

U obrazovky Screen2 nastavte ScreenOrientation na portrait, aby se orientace displeje neměnila a Title přejmenujte na „Průvodce“. Do této obrazovky přidejte komponenty ze sekce User Interface v tomto pořadí: Label1, Image, Label2, Button a ListPicker, dále TinyDB ze sekce Storage.

Text u Label1 bude název vámi vybraného města. V properties ho zarovnejte do středu, TextAligment přepněte na center:1 a písmo nastavte tučné zaškrtnutím FontBold.

C. Nahrajte tematickou fotografii do Image, nastavte výšku i šířku na Fill parent.

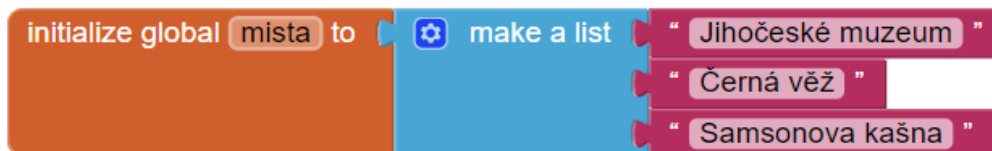
D. Nastavte popisku Label2 text na „Kam se chcete podívat?“ a ListPickeru na „Vybrat“.



Obrázek 6.1.23: Výsledek práce v Designeru

E. Vytvořte seznam položek „mista“ s názvy památek nebo míst ve městě.

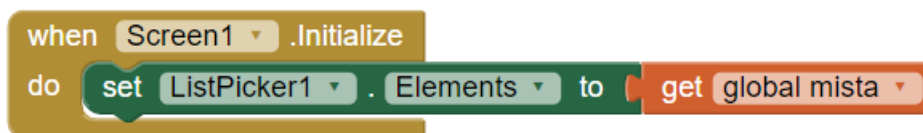
Do ListPickeru potřebujete vložit položky, které se objeví při rozkliknutí ListPickeru, výběru ze seznamu položek. Seznam položek je třeba uložit do proměnné, inicializujte tedy proměnnou `mista`, do té vložte blok `make a list`, který je v sekci Lists. Připojte textové bloky s názvy míst, které se v průvodci budou vyskytovat.



Obrázek 6.1.24: Seznam s položkami

F. Nastavte položky ListPicker1 na položky seznamu v proměnné `mista`.

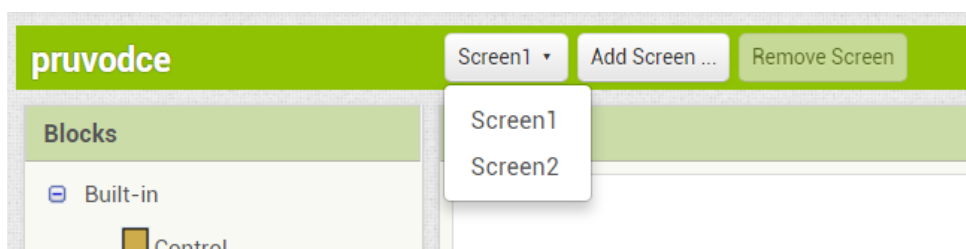
Seznam položek se musí nastavit po spuštění aplikace, přesuňte tedy do Vieweru blok `when Screen1.Initialize do` a do něj přidejte proceduru `set ListPicker1.Elements to` a nastavte položky na seznam přidáním bloku `get global mista`.



Obrázek 6.1.25: Nastavení položek do Listpickeru

G. Přidejte další obrazovku Screen2.

Novou obrazovku přidáte kliknutím na Add Screen na tmavě zelené liště pod hlavní lištou. Zde také můžete obrazovky přepínat mezi sebou, či je smazat. Obrazovky nelze přejmenovat, pouze při přidávání lze obrazovku pojmenovat.



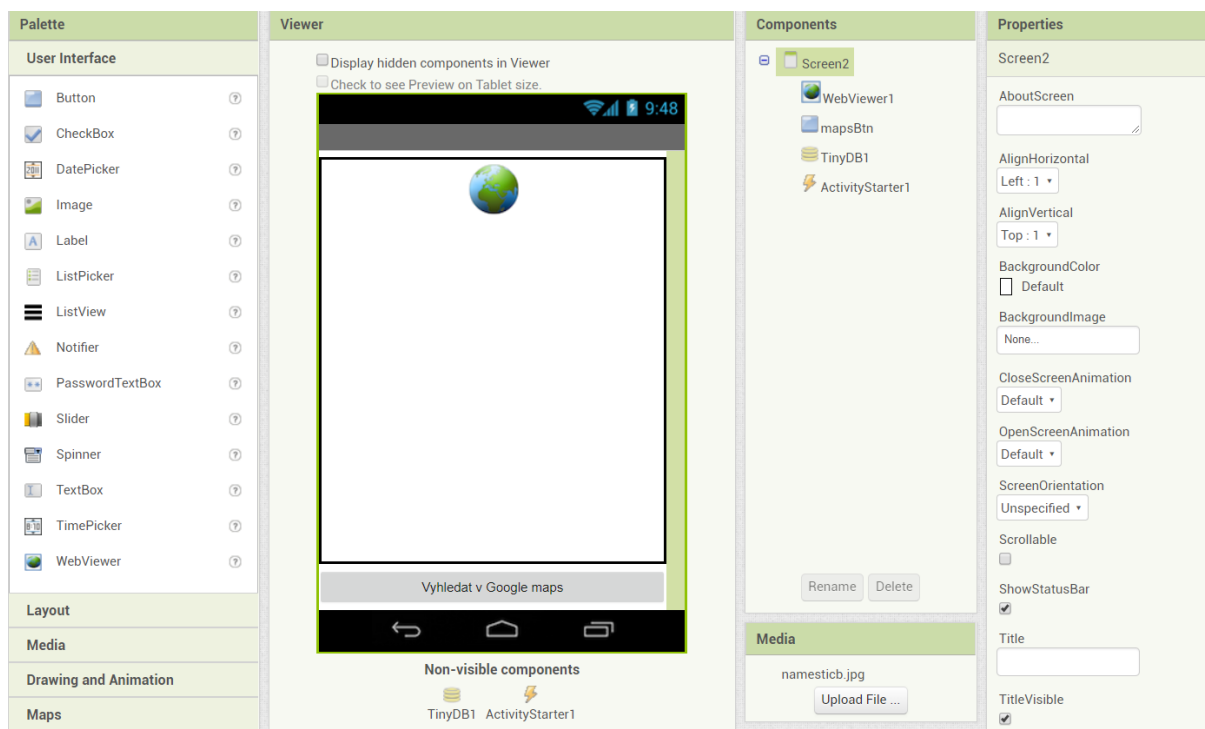
Obrázek 6.1.26: Přidání nové obrazovky

H. Vložte do uživatelského prostředí obrazovky Screen2 TinyDB, ActivityStarter, WebViewer a pod něj vložte tlačítko „Vyhledat v Google maps“. Výchozí titulek obrazovky vymažte.

Do obrazovky Screen2 přetáhněte neviditelné komponenty TinyDB ze sekce Storage a ActivityStarter z Connectivity.

WebViewer naleznete v sekci User Interface, nastavte u něj výšku na Fill parent. Pod něj vložte tlačítko a pojmenujte jej jako „mapBtn“ s textem „Vyhledat v Google maps“. Tlačítku nastavte šířku na Fill Parent.

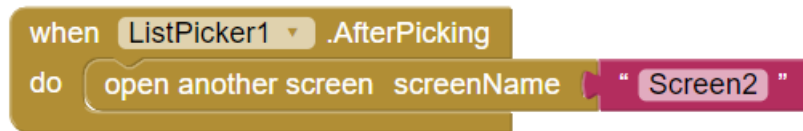
Text titulku u Screen2 vymažte a nechte prázdný.



Obrázek 6.1.27: Uživatelské rozhraní Screen2

I. Přepněte se zpět na Screen1 a zajistěte načtení druhé obrazovky po výběru ze seznamu.

V komponentách ListPickeru je dostupný block when ListPicker1.AfterPicking do, který určuje chování po výběru ze seznamu. V sekci Control naleznete blok pro načtení nové obrazovky open another screen parametrem screenName, který doplníte textovým polem s názvem nové obrazovky „Screen2“.



Obrázek 6.1.28: Otevření Screen2

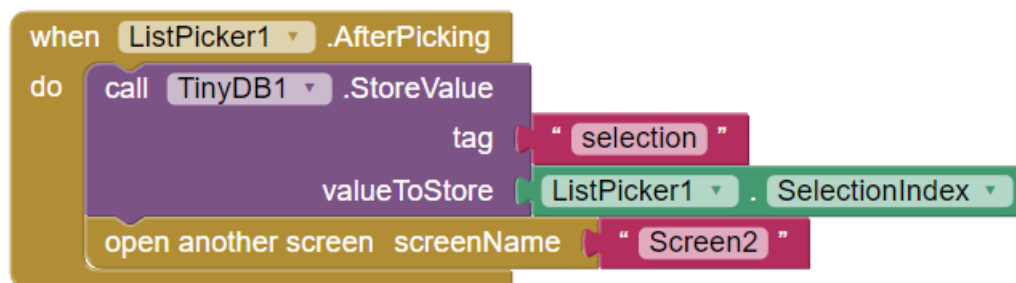
J. Přepněte se do Screen1. Uložte index vybrané položky ze seznamu do databáze s tagem „selection“.

Druhá obrazovka aplikace potřebuje k jejím funkcím znát index vybrané položky, to znamená pořadí položky v seznamu. Díky indexu bude obrazovka Screen2 vědět, jaká položka byla ze seznamu vybrána a doplní titulek obrazovky, webový prohlížeč a vyhledání na mapě podle výběru.

Jelikož mezi obrazovkami nelze sdílet proměnné, je třeba využít jiný způsob, jak předat informace z jedné obrazovky do druhé. Toto provedete využitím malé databáze TinyDB, do které se požadovaná informace uloží hned po výběru z položek.

Po výběru ze seznamu se nyní načte druhá obrazovka. Je třeba, aby se nejdříve index uložil do databáze a až poté se otevřela nová obrazovka, protože poté se už žádné příkazy ze Screen1 již neprovedou.

Nad blok `open another screen` tedy vložte proceduru `call TinyDB1.StoreValue`, kterou naleznete v blocích komponenty TinyDB. Procedura má parametry `tag` a `valueToStore`. Parametr `tag` je název, pod kterým je hodnota v databázi uložena a s jehož využitím pak tuto hodnotu použijete, získáte. Do `valueToStore` se přidá samotná hodnota, informace, text, to, co chcete do databáze uložit.



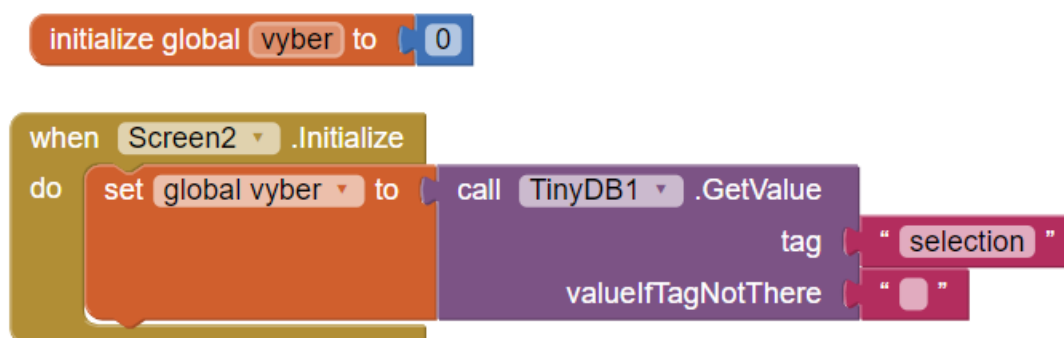
Obrázek 6.1.29: Krok K

K. Ve Screen2 inicializujte proměnnou `vyber` a nastavte ji na hodnotu `selection` z databáze.

Po otevření druhé obrazovky, je třeba načíst index výběru z databáze. Získanou hodnotu je třeba uložit do proměnné. Blok inicializace proměnné najdete v sekci Variables. Proměnnou přejmenujte na „vyber“ a její počáteční hodnotu nastavte na nulu.

Nejdříve přetáhněte do Vieweru blok `when Screen2.Initialize do`, protože budete určovat, co se stane po otevření této obrazovky.

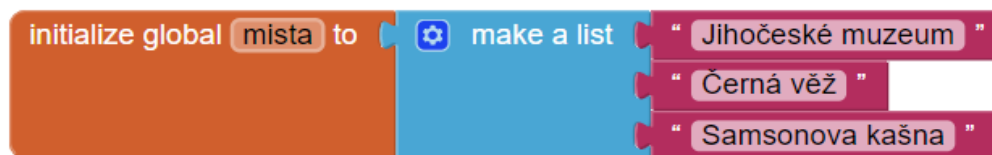
Do tohoto bloku vložte další blok `set global vyber to` a nastavte proměnnou na index výběru. Informaci z databáze, tedy index výběru získáte pomocí procedury `call TinyDB1.GetValue` s tagem, pod kterým je index uložen, tedy „selection“.



Obrázek 6.1.30: Krok L

L. Vytvořte seznam totožný se seznamem s místy ze Screen1.

Seznam uložte do proměnné místa a přidejte textové položky s místy ve stejném pořadí, jako ve Screen1. Dodržet stejné pořadí je velmi důležité, protože pomocí získaného indexu – pořadí vybrané položky a tohoto seznamu obrazovka Screen2 určí, která položka byla vybrána.

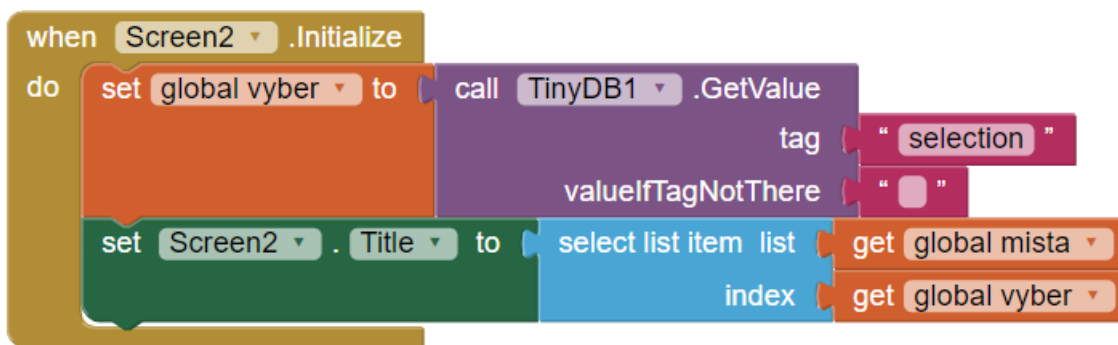


Obrázek 6.1.31: Seznam s položkami

M. Nastavte titulek druhé obrazovky na název vybraného místa

Položky seznamu místa jsou přesné názvy míst, které bude průvodce nabízet. Lze je tedy využít k nastavení titulku obrazovky na název vybraného místa, po načtení této obrazovky.

Do bloku `when Screen2.Initialize` do přidejte proceduru `set Screen2.Title` to, k němu připojte blok ze sekce Lists `select list item` s parametry `list` a `index`. Parametr `list` určuje, z jakého seznamu se bude vybírat, zde připojte blok `get global místa` a parametr `index` určí, jaké pořadí má požadovaná položka, připojte blok `get global vyber`. Tímto získáte název vybrané položky, který se doplní do titulku obrazovky.



Obrázek 6.1.32: Krok N

N. Vytvořte seznam s odkazy.

Podobným způsobem, jako nastavení titulku obrazovky podle indexu výběru, se bude nastavovat url adresa komponenty `WebViewer` na odpovídající webovou stránku. Vytvořte tedy seznam, uložený do proměnné `odkazy` a přidejte k němu stejný počet textových polí, jako je počet položek v seznamu `místa`. Ve správném pořadí přidejte odkazy, na požadované webové stránky, týkající se míst, které jsou dostupné v průvodci.

Například v seznamu `místa` je na první položkou „Jihočeské muzeum“, první položkou v seznamu `odkazy` tedy bude odkaz na stránku s informacemi o Jihočeském muzeu. Takto doplňte všechny odkazy.

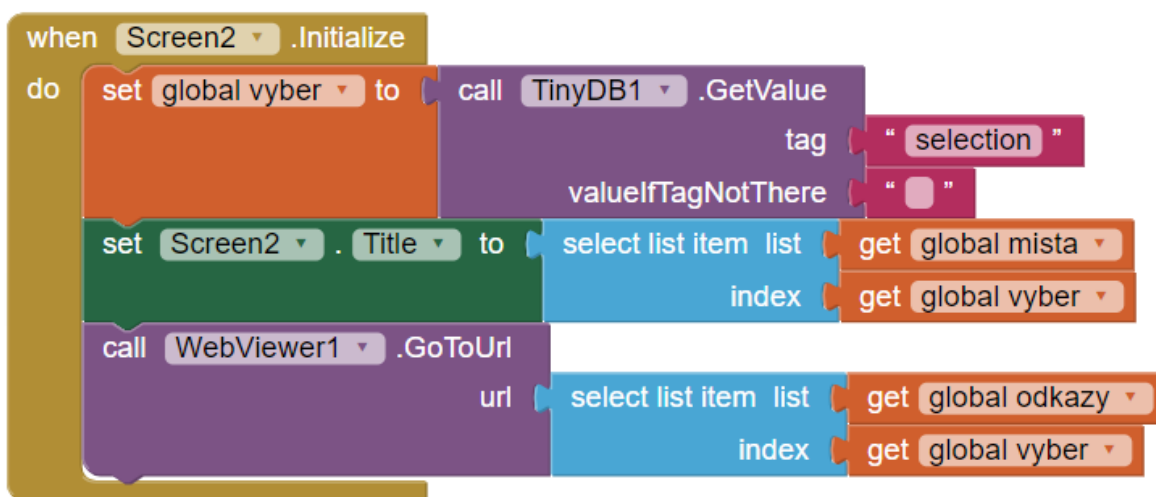


Obrázek 6.1.33: Seznam s odkazy

O. Nastavte ve `WebVieweru` odkaz webu s informacemi o památce dle výběru.

Nyní nastavte url adresu webového prohlížeče `WebView1` po načtení této obrazovky. Nastavující blok tedy přidáte do existujícího bloku `when Screen2.Initialize do`. V blocích komponenty `WebView1` je dostupná procedura `call WebView1.GoToUrl`, s parametrem `url`. Tato procedura ve vestavěném webovém prohlížeči `WebView` nastaví požadovanou webovou stránku, která je nastavena v parametru `url`.

Do parametru `url` je třeba nastavit odpovídající odkaz ze seznamu odkazy a má stejný index, jako výběr položky na úvodní stránce. Nastavení provedete přidáním bloku `select list item`, kde parametr `list` bude `get global odkazy`, protože budete vybírat ze seznamu odkazy a pořadí index určíte dosazením bloku `get global vyber`.



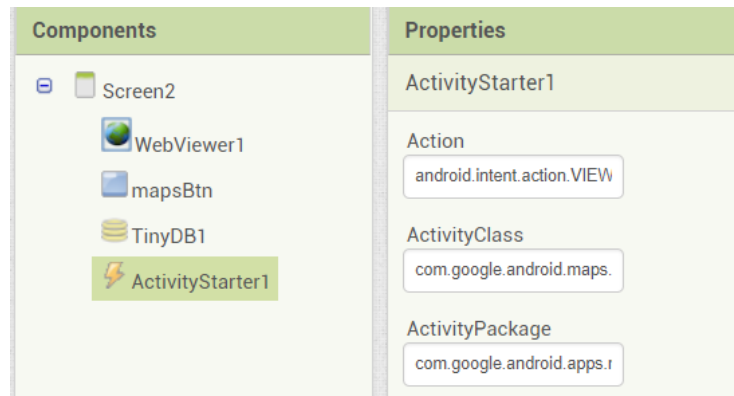
Obrázek 6.1.34: Kroky L, N, P

P. Zajistěte funkčnost tlačítka `mapsBtn`, vyhledání na mapě.

Pomocí neviditelné komponenty `ActivityStarter` je možné spustit jinou službu – aktivitu vašeho zařízení v rámci této aplikace. To znamená, že spuštění aplikace nezpůsobí, že opustíte aktuální aplikaci, v tomto případě průvodce, ale ze spuštěné aplikace se tlačítkem zpět vrátíte do aplikace původní. Můžete například spustit webové vyhledávání, konkrétní webovou stránku, či jinou aplikaci.

Nejprve je třeba nastavit této komponentě její vlastnosti, podle spouštěné aktivity. Přepněte se tedy do `Designeru` a v `Properties` vyplňte vlastnosti komponenty `ActivityStarter` dle následující tabulky:

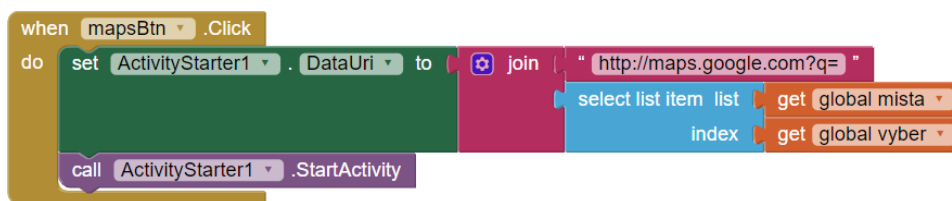
Action	android.intent.action.VIEW
ActivityClass	com.google.android.maps.MapActivity
ActivityPackage	com.google.android.apps.maps



Obrázek 6.1.35: Nastavení ActivityStarter

Do bloku `when mapsBtn.Click` přidejte proceduru komponenty `set ActivityStarter1.DataUri` to a blok `join`, kterým spojíte text „<http://maps.google.com?q=>“ a název vybrané položky. Ten určité stejně, jako tomu bylo při nastavení titulku obrazovky. Přidejte blok `select list item`, kde list bude `get global místa` a `index` `get` vyber.

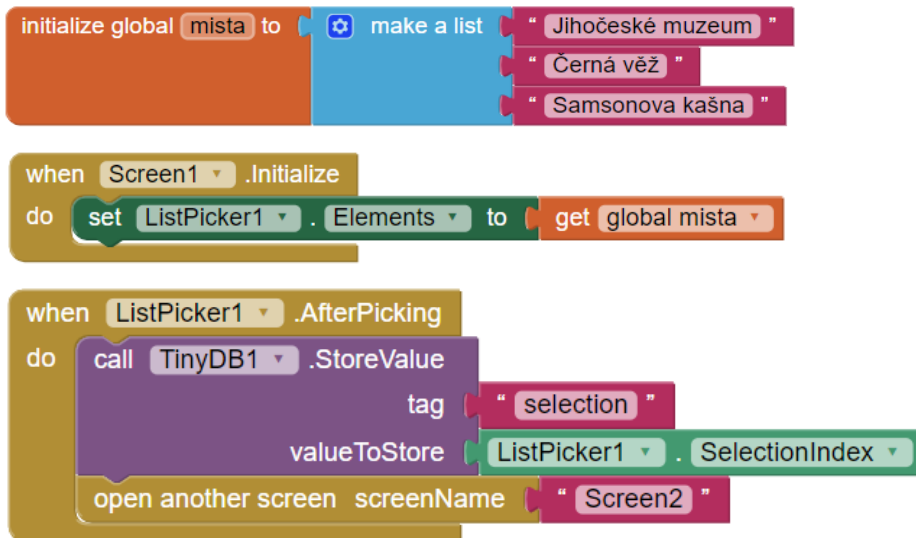
Přidejte poslední blok, který vykoná spuštění nové aktivity, najdete jej mezi ostatními bloky komponenty `ActivityStarter1`, `call ActivityStarter1.StartActivity`.



Obrázek 6.1.36: Funkce tlačítka mapsBtn

Průvodce – výsledný kód

Screen1:

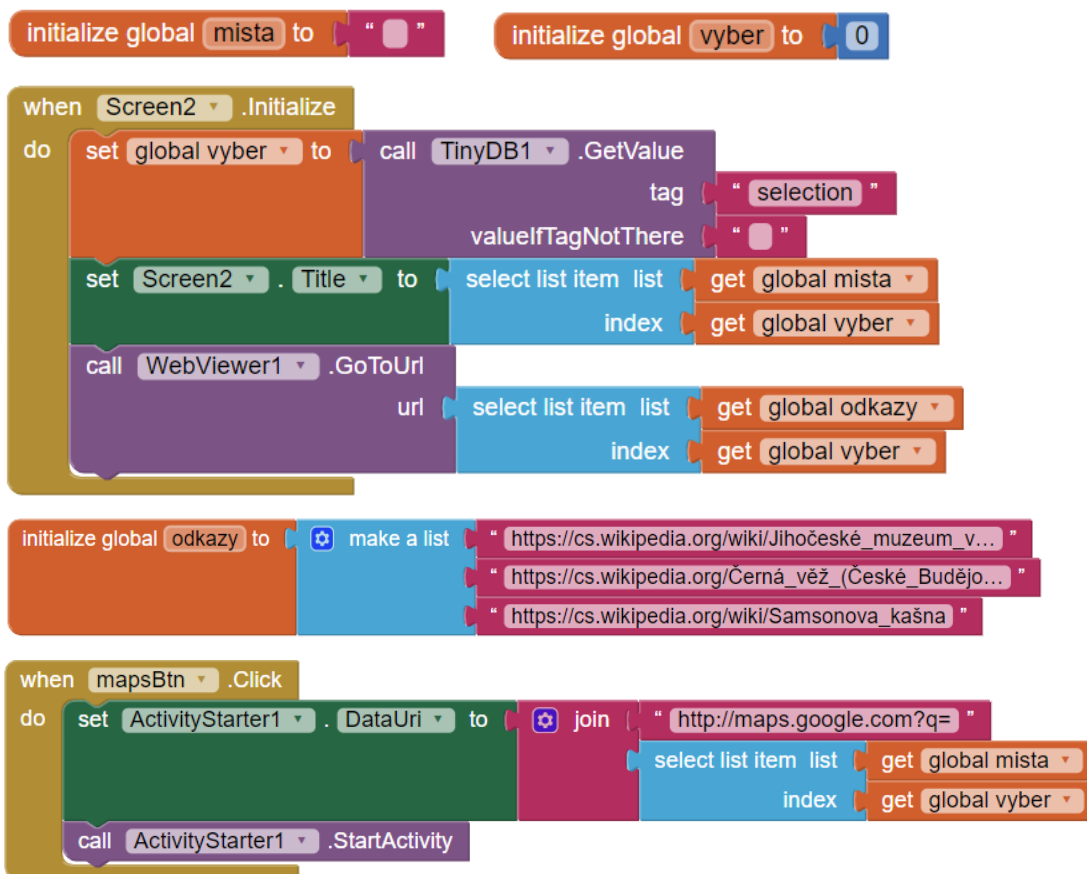


```
initialize global mista to make a list " Jihočeské muzeum "
" Černá věž "
" Samsonova kašna "

when Screen1 .Initialize
do set ListPicker1 . Elements to get global mista

when ListPicker1 .AfterPicking
do call TinyDB1 .StoreValue
tag " selection "
valueToStore ListPicker1 . SelectionIndex
open another screen screenName " Screen2 "
```

Screen2:



```
initialize global mista to " " initialize global vyber to 0

when Screen2 .Initialize
do set global vyber to call TinyDB1 .GetValue
tag " selection "
valueIfTagNotThere " "
set Screen2 . Title to select list item list get global mista
index get global vyber
call WebViewer1 .GoToUrl
url select list item list get global odkazy
index get global vyber

initialize global odkazy to make a list " https://cs.wikipedia.org/wiki/Jihočeské_muzeum_v... "
" https://cs.wikipedia.org/Černá_věž_(České_Budějo... "
" https://cs.wikipedia.org/wiki/Samsonova_kašna "

when mapsBtn .Click
do set ActivityStarter1 . DataUri to join " http://maps.google.com?q= "
select list item list get global mista
index get global vyber
call ActivityStarter1 .StartActivity
```

6.1.5. Kde jsi?

V předchozí úloze jste vytvořili průvodce města pro svého známého. Co když se ve městě ztratí? Nebo ho jen chcete někde vyzvednout, ale neví, kde se právě nachází? Tato aplikace umožní odeslat SMS zprávu s adresou aktuální polohy zařízení. Stačí obdržet SMS s otázkou „Kde jsi?“ a text zprávy se okamžitě vyplní k odeslání.

Cíl úlohy:

- Práce s komponentou Texting
- Získání aktuální adresy využitím LocationSensor
- Použití podmínky if

Zadání:

Naprogramujte aplikaci, která vypíše aktuální adresu zařízení do textového pole, do kterého se zapisuje text SMS zprávy k odeslání. Toto se provede buď automaticky po obdržení zprávy, která obsahuje text „kde jsi“, nebo kliknutím na tlačítko „Vypiš aktuální adresu“. Kliknutím na tlačítko „Odeslat zprávu“ se odešle SMS zpráva s textem textového pole na číslo získané z textového pole s číslem odesílatele.

Po obdržení SMS zprávy se na obrazovku vypíše text zprávy a telefonní číslo odesílatele. Číslo odesílatele se vypíše do textového pole.

Komponenty:

- Button
- Label
- TextBox
- HorizontalArrangement
- LocationSensor
- Texting

Poznámky:

Odeslaná SMS zpráva se účtuje podle tarifu vašeho telefonního operátora.

Kde jsi? - zadání v krocích

- A.** Založte nový projekt s názvem „kdeJsi“ a titulek obrazovky aplikace přejmenujte na „Kde jsi?“
- B.** Upravte uživatelské prostředí dle předlohy viz níže.
- C.** Vypište telefonní číslo odesílatele přijaté zprávy do textového pole a text zprávy na obrazovku.
- D.** Nastavte tlačítko „Vypiš aktuální polohu“ tak, aby vypsalo adresu do textu zprávy k odeslání.
- E.** Zařídte, aby se při obdržení zprávy obsahující text „kde jsi“ vypsala adresa do textu zprávy k odeslání.
- F.** Nastavte telefonní číslo komponenty Texting na číslo vypsané v textovém poli.
- G.** Naprogramujte tlačítko „Odeslat zprávu“.

Kde jsi? - podrobný postup

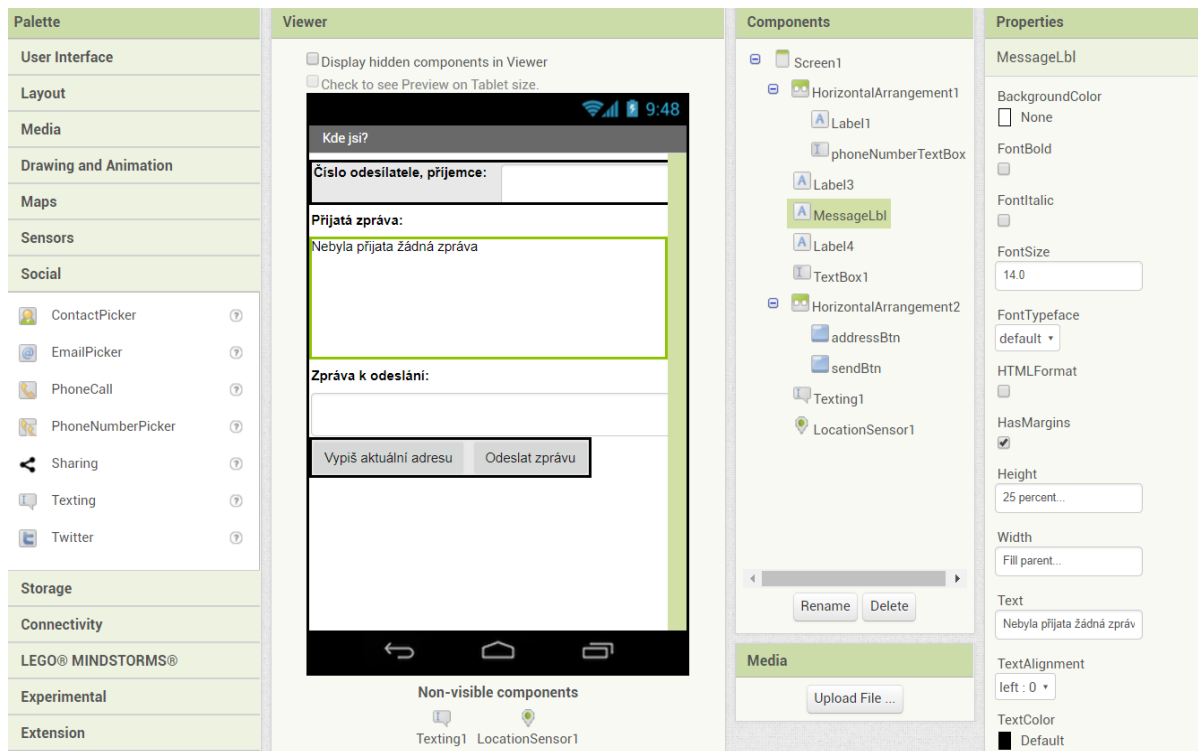
A. Založte nový projekt s názvem „kdeJsi“ a titulek obrazovky aplikace přejmenujte na „Kde jsi?“

B. Upravte uživatelské prostředí dle předlohy viz níže.

Přidejte do obrazovky horizontální layout. Do něj vložte label s tučným textem „Číslo odesílatele, příjemce:“ a textové pole s názvem „phoneNumberTextBox“.

Pod layout přidejte label s tučným textem „přijátá zpráva: “ a další label s textem „nebyla přijata žádná zpráva“. Tomuto labelu změňte název na „messageLbl“ a nastavte výšku na 25 %. Vložte poslední label s textem „Zpráva k odeslání“.

Dále přidejte textové pole s šířkou Fill parent, poté horizontální layout, do nějž vložte dvě tlačítka. Prvnímu nastavte text na „Vypiš aktuální adresu“ a název na „addressBtn“, druhý bude mít text „Odeslat zprávu“ a název „sendBtn“.

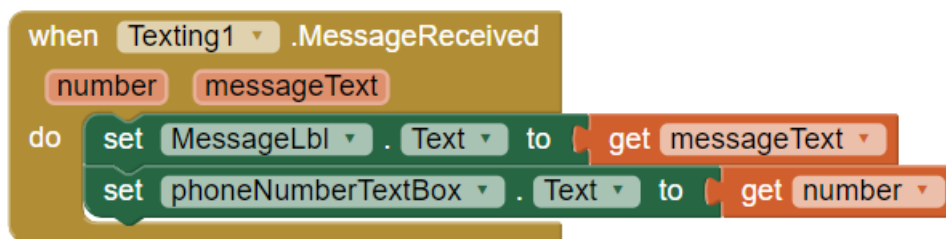


Obrázek 6.1.37: Uživatelské prostředí

C. Vypište telefonní číslo odesílatele přijaté zprávy do textového pole a text zprávy na obrazovku.

V blocích komponenty `Texting1` naleznete blok `when Texting1.MessageReceived` do, který určuje chování aplikace po obdržení zprávy, s parametry `number` – telefonní číslo odesílatele přijaté zprávy a `messageText` – text přijaté SMS zprávy.

Pomocí bloků `set MessageLbl.Text` to a `get messageText` vypište text zprávy do labelu `MessageLbl` a spojením bloků `set phoneNumberTextBox.Text` to a `get number` vypište číslo odesílatele do textového pole `PhoneNumberTextBox`. Tyto příkazy vložte do bloku `when Texting1.MessageReceived` do.

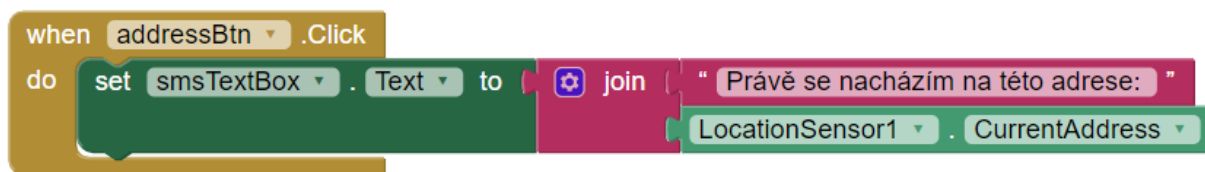


Obrázek 6.1.38: Výpis informací na obrazovku po obdržení zprávy

D. Nastavte tlačítko „Vypiš aktuální polohu“ tak, aby vypsalu adresu do textu zprávy k odeslání.

Snímač polohy `LocationSensor` je komponenta, která získává například aktuální adresu, zeměpisné souřadnice, nadmořskou výšku a rychlost.

Do bloku `when addressBtn.Click` vložte blok `set TextBox1.Text`, protože po stisknutí tlačítka potřebujete nastavit text textového pole. K tomuto bloku připojte spojovací blok `join` kterým spojíte následující: text „Právě se nacházím na této adrese: “ a blok `LocationSensor1.CurrentAddress`, který získá aktuální polohu zařízení. Naleznete jej v blocích snímače polohy.



Obrázek 6.1.39: Výpis aktuální polohy

E. Zařídte, aby se při obdržení zprávy obsahující text „kde jsi“ vypsal adresy do textu zprávy k odeslání.

Do bloku, který zajišťuje chování po obdržení zprávy vložte blok, který zjistí, zda SMS zpráva obsahuje textový řetězec „kde jsi“. Pokud ano, vypíše se do textového pole `messageText` aktuální poloha zařízení.

Toto zajišťuje blok `if, then`, tedy když je podmínka splněna, pak proved' následující příkazy. V této aplikaci to bude: když zpráva obsahuje otázku „kde jsi?“, adresa se vypíše. Pokud ne, nestane se nic.

Tento blok přidejte ze sekce Control pod příkazy nastavující text zprávy a číslo odesílatele, příjemce.

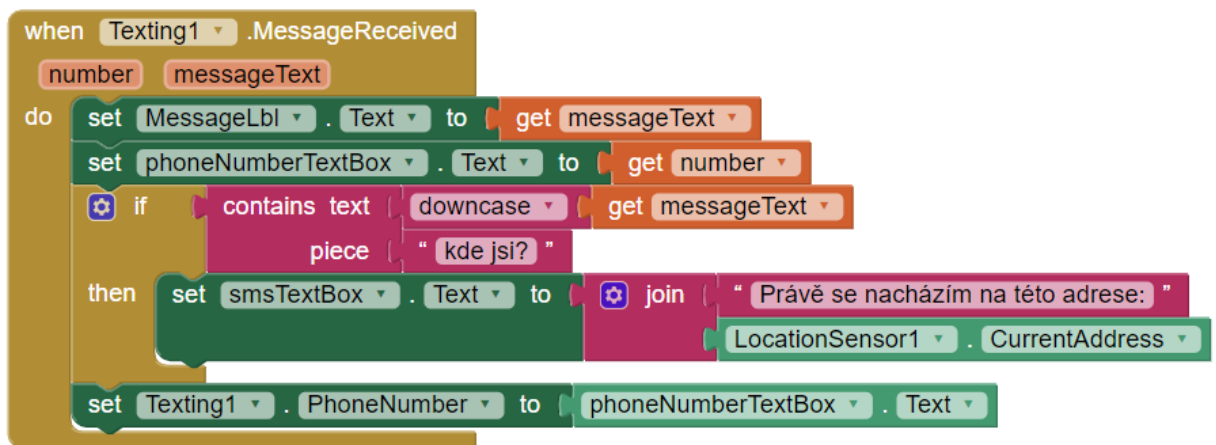
Blok `contains` ze sekce Text zjišťuje, zda text obsahuje určitý text. Má parametry `text`, který označuje text, se kterým se bude pracovat a `piece` je kousek textu, o němž zjišťujete, zda je v textu obsažen.

Parametr `piece` bude v této aplikaci text „kde jsi?“ malými písmeny, je ale možné, že text zprávy bude písmeny velkými. Pro jistotu je tedy dobré přijatou zprávu převést na malá písmena spojením bloků `downcase` ze sekce Text a `get messageText`. Tyto bloky dosaďte do parametru `text`. Parametr `piece` bude text „kde jsi?“. Tyto bloky budou podmínka k vypsání adresy.

Příkaz, který se má provést bude složen z bloku `set smsTextBox.Text to` a bloku `join`, který bude spojovat opět text „Právě se nacházím na této adrese: “ a aktuální polohu získanou pomocí bloku `LocationSensor1.CurrentAddress`.

F. Nastavte telefonní číslo komponenty Texting na číslo vypsané v textovém poli.

Pod předchozí bloky přidejte blok `set Texting1.PhoneNumber to phoneNumberTextBox.Text`. Takto se nastaví telefonní číslo příjemce na číslo, které je vypsané v textovém poli `phoneNumberTextBox`.

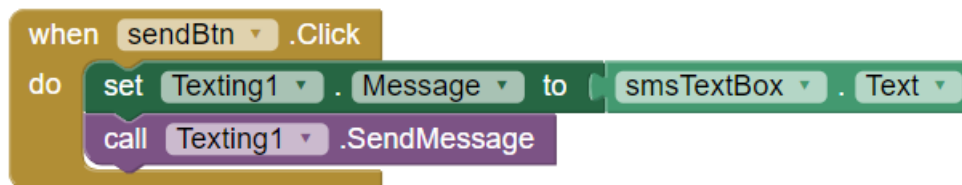


Obrázek 6.1.40: Kroky C, E, F

G. Naprogramujte tlačítko „Odeslat zprávu“.

Do bloku `when sendBtn.Click` přidejte bloky `set Texting1.Message to smsTextBox.Text`. Text zprávy, která se odešle po stisknutí na tlačítko `sendBtn` se nastaví na text z připravené zprávy v textovém poli `smsTextBox.Text`.

Zavolejte proceduru `call Texting1.SendMessage` patřící ke komponentě `Texting1` k odeslání SMS zprávy.



Obrázek 6.1.41: Nastavení tlačítka `sendBtn`

Kde jsi? – výsledný kód

```
when Texting1 .MessageReceived
  number messageText
do
  set MessageLbl . Text to get messageText
  set phoneNumberTextBox . Text to get number
  if contains text lowercase get messageText
    piece " kde jsi? "
  then set smsTextBox . Text to join " Právě se nacházím na této adrese: "
    LocationSensor1 . CurrentAddress
  set Texting1 . PhoneNumber to phoneNumberTextBox . Text
```

```
when addressBtn .Click
do
  set smsTextBox . Text to join " Právě se nacházím na této adrese: "
    LocationSensor1 . CurrentAddress
```

```
when sendBtn .Click
do
  set Texting1 . Message to smsTextBox . Text
  call Texting1 .SendMessage
```

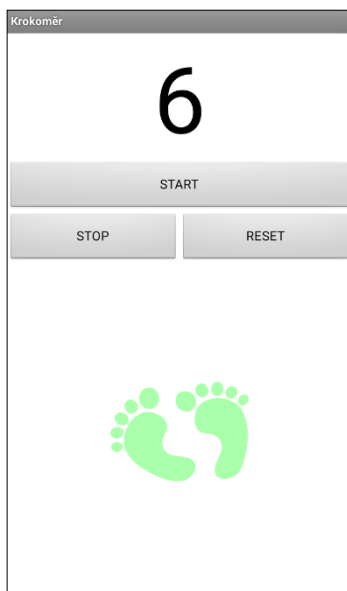
6.1.6. Závěrečný projekt – Fit App



Obrázek 6.1.42: Menu Fit App

Závěrečným projektem je mobilní aplikace, která slouží jako ukázka, jaká aplikace se dá v AI2 vytvořit. V rámci závěrečného projektu byla vytvořena mobilní aplikace Fit App se čtyřmi obrazovkami a se třemi funkcemi. První funkcí je krokoměr, druhá umožňuje výpočet BMI a jeho vyhodnocení. Třetí funkce zaznamenává množství vypité vody.

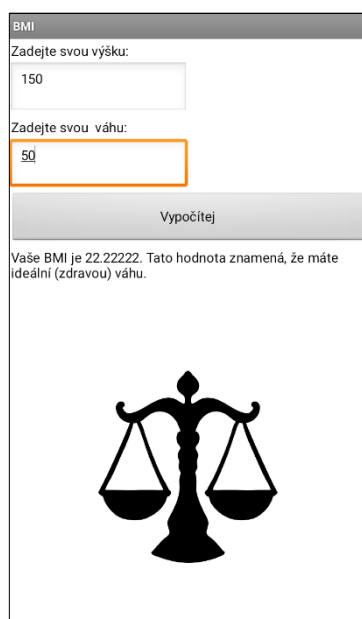
Krokoměr



Obrázek 6.1.43: Krokoměr

Krokoměr využívá sensor `Pedometer`, který snímá každý krok uživatele. Kroky se vypisují na obrazovku. Pro zahájení zaznamenávání počtu kroků je třeba stisknout tlačítko `START`. Naopak stisk tlačítka `STOP` krokoměr zastaví. Stisknutím tlačítka `RESET` se celkový počet kroků vynuluje. Krokoměr uchovává počet kroků v databázi `TinyDB`. Krokoměr se tak po opuštění aplikace nevynuluje a dokud se aplikace neukončí nebo krokoměr nezastaví, kroky se stále počítají a po opětovném načtení aplikace se načte i počet ušlých kroků.

BMI



Obrázek 6.1.44: BMI

BMI – Body Mass Index, neboli index tělesné hmotnosti vyhodnocuje tělesný stav podle váhy vzhledem k výšce a to podvýživu, podváhu, zdravou váhu, nadváhu a tři stupně obezity. Do aplikace se zadá výška v centimetrech, váha v kilogramech a stisknutím na tlačítko Vypočítej se na obrazovce vypíše hodnota BMI a jeho význam. Tato funkce má zabudovaný akcelerometr, tedy `AccelerometerSensor`. Zatřesením telefonu se vymažou zadané hodnoty i výsledek a výpočet je možné opakovat.

Pitný režim



Obrázek 6.1.45: Pitný režim

Funkce pitný režim zaznamenává množství vypité vody. V nabídce jsou tři sklenice, s hodnotami 150 ml, 300 ml a 500 ml. Stiskem na libovolnou sklenici se započítá množství vody dle vybrané velikosti sklenice. Množství zaznamenané vody se vypíše na obrazovce v mililitrech. Tato hodnota lze vynulovat. Pitný režim se ukládá do databáze `TinyDB` pouze na den, kdy je množství vypité vody zaznamenáno. Další den je množství vypité vody vynulováno. Toto je zajištěno využitím senzoru `Clock`, který zjišťuje aktuální den.

Veškeré projekty ze sady úloh a tento závěrečný projekt jsou sdíleny v galerii AI2 (dostupné na: <http://ai2.appinventor.mit.edu/>) pod účtem **emkinventor**.

7. Ověření sady úloh ve výuce

Sada úloh byla ověřována ve výuce na Střední škole strojní a elektrotechnické ve Velešíně. Výuka byla vedena Mgr. Příbylem, který se s prostředím App Inventor 2 setkal již dříve. Sada úloh se ověřovala se studenty 3. ročníku oboru Elektrotechnika – Počítačové systémy v předmětu Praxe.

Výuka začala krátkým úvodem do problematiky a představením prostředí MIT App Inventor 2. Následovala příprava Android zařízení k testování aplikací. Studenti mobilní aplikace testovali v Android telefonech, či tabletech. Zde měli někteří žáci krátce technické potíže, žádné jiné výrazné problémy se později neobjevily.

Žáci řešili první dvě úlohy – Hello World a Virtuální kostka – společně s vyučujícím. Ostatní úlohy dostali žáci v souboru PDF a pracovali samostatně. Při ověřování, které trvalo přibližně dvě a čtvrt vyučovací hodiny byly vyřešeny všechny úlohy ze sady. Vyučovací hodina Praxe trvá 60 minut. Zbýlý čas třetí vyučovací hodiny využili žáci k vytváření aplikací dle svého námětu, například přidání funkce do úlohy Pohlednice, propojit tlačítko v aplikaci, aby ovládalo LED diodu připojenou na Rapsberry Pi, vytvořit jednoduchou hru, nebo jen k průzkumu dalších komponent AI2.

Podle pana učitele je MIT App Inventor 2 výborná pomůcka pro úvod do tvorby mobilních aplikací a uvažuje o výuce v tomto prostředí. App Inventor již doporučit svým kolegům. Zpětná vazba žáků byla zcela pozitivní, kromě těch, kteří vlastní telefony s iOS. Vadila jim nedokončená podpora pro iPhone. Ti si přinesli tablety s OS Android.

K mé práci nebylo nic vytknuto, podle pana magistra je sada úloh přehledná, dobře strukturovaná a úlohy jsou vhodně zvoleny. Popis úloh je prý velmi podrobný, při příštím použití ve výuce, které plánuje následující školní rok žákům nebudou kromě první úlohy poskytnuty podrobné postupy.

8. Závěr

Po analýze materiálů jsem zjistila, že publikací k nové verzi App Inventor není mnoho, avšak ty dostupné jsou kvalitně zpracované. Malé množství kompenzují také poměrně velké množství materiálu na internetu. V českém jazyce je bohužel dostupná jediná kniha a to k první verzi aplikace.

Porovnáním prostředí Scratch a App Inventor jsem dospěla k názoru, že Scratch je vhodným prostředím k výuce programování, které předchází programování v App Inventoru. Uživatel prostředí Scratch poté může využít své znalosti v AI2 a vyzkoušet si naprogramovat něco jiného. V dnešní době AI2 může být velkou motivací k programování, protože si uživatel může naprogramovat vlastní mobilní aplikaci.

Mým původním cílem bylo, že vypracuji metodickou příručku k výuce v prostředí MIT App Inventor 2. Během vypracovávání práce jsem zjistila, že AI2 není vhodný k výuce tolik, jako například prostředí Scratch. V AI2 je hlavní náplní práce práce s grafickým prostředím a využívání senzorů a funkcí telefonu a možnosti jsou více jednoznačné. Je vhodný spíše k úvodu do vývoje mobilních aplikací. Metodický materiál by tak nebyl tolik přínosný, vytvořila jsem proto popis a návod k orientaci v AI2 a sadu úloh s tutoriály, která lze využít jako pomůcka do výuky, ale i pro samouky. Aktivita tedy neučí schopnosti programovat, ale spíše jeho uplatnění v prostředí pro tvorbu mobilních aplikací. Uživatel se naučí orientovat a vyvíjet mobilní aplikace v prostředí AI2, nikoliv programovat. Sada úloh nabízí hodně možností a způsobů využití a přestože předpokladem jsou znalosti základů programování, uživatel bez předchozích zkušeností podle podrobného postupu z úlohy také zvládne vytvořit aplikace, které jsou zahrnuty v sadě.

Vytvořila jsem závěrečný vzorový projekt, která ukazuje, jakou aplikaci je možné v AI2 vytvořit. Všechny využití komponenty a senzory jsou probrány v sadě úloh. Pro uživatele, který splnil všechny úlohy by tedy nemuselo být složité aplikaci Fit App, či jinou rozsáhlejší aplikaci vytvořit.

Veškeré projekty včetně závěrečného projektu jsou sdíleny na v galerii AI2 pod účtem **emkinventor** na <http://ai2.appinventor.mit.edu/>.

Podle mého názoru má prostředí AI2 veliký potenciál, nabízí spoustu možností pro tvorbu mobilních aplikací na OS Android s různými funkcemi a senzory. MIT App Inventor 2 je do výuky vhodný jako zpestření a k úvodu do vývoje mobilních aplikací. S porovnáním vývoje mobilních aplikací v Android studiu je vývoj v AI2 nesrovnatelně jednodušší a rychlejší. Ovšem možnosti jsou stále omezené.

9. Literatura a ostatní použité zdroje

- [1] ABEYWARDENA, Ishan Sudeera.: *Educational App Development Toolkit for Teachers and Learners*. Burbany: Commonwealth of Learning, 2015. ISBN 978-1-894975-73-5.
- [2] BEER, Paula, SIMMONS, Carl. *Hello app inventor!.: Android programming for kids and the rest of us*. Shelter Island, NY: Manning Publications, 2015. ISBN 9781617291432.
- [3] FUKA, Přemysl.: *Mobilní aplikace se zaměřením na platformu Android*. Praha, 2016. Diplomová práce. Bankovní institut vysoká škola Praha. Katedra informatiky a kvantitativních metod. Vedoucí práce doc. Ing. Alena Buchalcevová, Ph.D.
- [4] KREJSA, Jan.: *Výuka základů programování v prostředí Scratch*. Diplomová práce. České Budějovice: Jihočeská univerzita v Českých Budějovicích. Fakulta pedagogická, 2014. Vedoucí práce doc. PaedDr. Jiří Vaníček, Ph.D.
- [5] LAU, Michelle.: *The Showdown Between App Inventor and Scratch*. [online]. Dostupné na:
<http://3ohands.com/blog/the-showdown-between-app-inventor-and-scratch/>
- [6] MIT App Inventor.: *About Us*. [online]. 2016. Dostupné na:
<http://appinventor.mit.edu/explore/about-us.html>
- [7] MIT App Inventor.: *App Inventor 2*. Vývojové prostředí. [online]. Dostupné na:
<http://ai2.appinventor.mit.edu/>
- [8] MIT App Inventor.: *Explore MIT App Inventor*. [online]. Dostupné na: <http://explore.appinventor.mit.edu/>
- [9] MIT App Inventor.: *Setting Up App Inventor*. [online]. 2016. Dostupné na:
<http://appinventor.mit.edu/explore/ai2/setup.html>
- [10] MIT App Inventor.: *Tutorials for App Inventor 2*. [online]. Dostupné na:
<http://explore.appinventor.mit.edu/ai2/tutorials>

- [11] NĚMEC, Petr. *Naprogramujte si aplikaci pro mobilní telefon*. Metodický portál RVP.CZ. [online]. 2016. Dostupné na:
<http://spomocnik.rvp.cz/clanek/18203/NAPROGRAMUJTE-SI-APLIKACI-PRO-MOBILNI-TELEFON.html>
- [12] Patrik VOŠTINÁR.: *Programovanie mobilných vzdéľavacích aplikácií v App Inventor a Android studio*. Banská Bystrica: Univerzita Mateja Bela v Banskej Bystrici, 2017.
- [13] ŠNAJDER, Ľubomír. *Výučba programovania mobilných aplikácií v App Inventor 2*. [online]. Dostupné na: <http://ics.upjs.sk/~snajder/ai2/>
- [14] ŠNAJDER, Ľubomír.: *Metodika výučby programovania mobilných zariadení*. Zborník konferencie DidInfo 2015. Banská Bystrica: Univerzita Mateja Bela, 2015. ISBN 978-80-557-0852-2. - s. 18-27
- [15] TOMCSÁNYIOVÁ, Monika, TOMCSÁNYI, Peter.: *Vyučovanie programovania vývojom androidových aplikácií v App Inventore*. Zborník konferencie DidInfo 2015. Banská Bystrica: Univerzita Mateja Bela, 2015. ISBN 978-80-557-0852-2. - s. 160-163
- [16] WIKIPEDIA.: *Index tělesné hmotnosti*. [online]. 2016 Dostupné na:
https://cs.wikipedia.org/wiki/Index_t%C4%9Blesn%C3%A9_hmotnosti
- [17] WOLBER, David, ABELSON, Hal, SPERTUS, Ellen, LOONEY, Liz.: *App Inventor 2: Create your own Android Apps*. O'Reilly Media, 2014. ISBN 978-1491906842
- [18] WOLBER, David. *Learn to build Android apps*. [online]. Dostupné na:
<http://www.appinventor.org>
- [19] WOLBER, David.: *The App Inventor Course-in-a-Box*. [online]. Dostupné na:
<http://www.appinventor.org/course-in-a-box-intro>

Přílohy

Příloha č. 1: CD – ROM • Bakalářská práce ve formátu PDF