

# ANDROID

## programování v MIT App Inventoru

Průvodce do světa pod povrchem Androidu. Kurz programování v blokově řízeném vývojovém prostředí MIT App Inventor.

### Využité programy:

MIT App Inventor  
App Inventor Emulátor  
Emulátor Genymotion  
Konektor aiStarter  
Konektor MIT AI2 Companion  
Android SDK

**Cílová skupina/náročnost:** 1. až 4. ročník SŠ a odpovídající ročníky gymnázií.

Autor:

Ing. Mgr. Filip Vaculík

Všechny uvedené texty, obrázky a videa jsou vlastní, není-li uvedeno jinak. Autory Youtube embed videí lze nalézt při kliknutí na znak Youtube ve videu během přehrávání.

**K plnohodnotnému využití této studijní opory je nutný přístup k on-line zdrojům a materiálům.**

Tento materiál vznikl z finanční podpory Evropského sociálního fondu a státního rozpočtu České republiky v rámci projektu „Popularizace vědy a badatelsky orientované výuky“, reg .č. CZ.1.07/2.3.00/45.0007.

# **1 Základní informace o projektu**

---

**Název: Programování pro Android v prostředí MIT App**

**Inventor Anotace programu/zaměření/hlavní cíl:**

Seznámit žáky s atraktivní možností tvorby aplikací pro Android, přímo ve webovém prohlížeči. Kromě seznámení s prostředím Androidu může být vývojové prostředí App Inventor i vhodným nástrojem k výuce algoritmizace a úvodem do programování. Podpora a přímé využití principů 1:1 a BYOD.

**Cílová skupina**

Technicky orientovaní žáci ZŠ, žáci gymnázií a SŠ, účastníci kurzů programování pro začátečníky.

**Organizační podmínky**

Předpokládá se výuka v počítačové učebně s menším počtem žáků (cca 8 až 15). V případě distančního vzdělávání lze formy a metody výuky přizpůsobit.

**Pomůcky**

PC s parametry:

CPU 1,5GHz, 1GB RAM, doporučené prohlížeče Mozilla Firefox nebo Google Chrome v aktuální verzi. Operační systém není rozhodující, podporovány jsou nejen hlavní 3 OS (Win, Lin, Mac).

Výhodou je vlastní zařízení s OS Android a WiFi připojení k internetu.

V případě použití emulátoru nároky na hardware PC vzrůstají dle typu a požadavků emulátoru.

**Časová náročnost (popř. jak je možné program rozložit – jedná-li se o celoroční program)**

Kurz je dimenzován na 12 dvouhodinových lekcí. Další čas může být věnován volitelným činnostem, které dle časových možností a zájmu žáků mohou i výrazně překročit původní záměr. Podporu pro vedení volitelných činností lze nalézt v materiálech mimo hlavní vzdělávací blok.

**Mezipředmětové vazby**

Vzhledem k prostoru pro volitelné směřování tvorby programů lze do výuky zahrnout mezipředmětové vazby pro jakýkoliv předmět. Charakterem práce se nelze vyhnout mezipředmětovým vazbám na anglický jazyk, matematiku, český jazyk a ICT předměty.

## **2 Motivační rámec projektu**

---

### **Text:**

Elektronická zařízení se velmi rychle stala významnou součástí osobního i profesního života většiny populace. Působení těchto zařízení ovlivňuje myšlení a kulturu v nebývalém měřítku. Motivační potenciál této oblasti je obrovský. Zde stojíme na hranici mezi těmi, kdo vytváří, a těmi, kdo pouze konzumují multimediální obsah. Snad vás tento kurz posune dále do světa tvůrců a přinese vám lepší pochopení technologií, které ovlivňují váš život.

### **Doporučený multimediální materiál:**

video viz. on-line kurz

## **3 Poznámky k využití programů**

---

### **Text:**

Samotný App Inventor je díky svému návrhu poměrně nenáročný. Studenti budou potřebovat vlastní Google účet, případně účet Google for education, pokud jej má škola zřízený. Propojení s vlastním zařízením s Androidem je nenáročné a postup lze nalézt na stránce <http://appinventor.mit.edu/explore/ai2/setup.html>. Problematictější může být zprovoznění emulátorů v případě, kdy žáci nemají vlastní zařízení, případně není dostupné kvalitní bezdrátové připojení pro všechna zařízení žáků.

Zprovoznování emulátorů je v tomto kurzu zpracováno do samostatných volitelných bloků.

### **Doporučený multimediální materiál:**

App Inventor je prostředí, které používají statisíce uživatelů po celém světě. Mnoho materiálů lze nalézt na stránce <http://appinventor.mit.edu/explore/>. Pro práci s tímto prostředím existuje řada ukázkových videí, dostupných například na YouTube.

Některá z videí jsou použita i v tomto kurzu. Stále vznikají nové materiály, tento kurz je toho příkladem. Pokud najdete nový zajímavý materiál, napište nám prosím o něm do wikipedie našeho kurzu.

#### 4 Aktivita 1 - seznámení s prostředím a Hello World

Téma	Programování pro Android v prostředí MIT App Inventor	
Tematický celek	Seznámení s prostředím App Inventoru	
Motivační rámec	Snadné vytvoření aplikace pro Android a její použití.	
Počet žáků	5-15	
Věk studentů	10+	
Pomůcky	PC odpovídajících parametrů, webový prohlížeč Mozilla Firefox nebo Google Chrome, zařízení s Androidem připojené k internetu, případně emulátor zařízení s Androidem, účet pro Google služby.	
Stručný popis aktivity	Přihlášení k prostředí App Inventoru <a href="http://appinventor.mit.edu/">http://appinventor.mit.edu/</a> , základní orientace v prostředí, tvorba první aplikace, propojení se zařízením s Androidem/ s emulátorem, spuštění vlastní aplikace.	
Vhodné místo	Počítačová učebna	
Cíle aktivity	Studenti budou s dopomocí učitele schopni splnit jednotlivé cíle, a pokud se nesetkají s technickými obtížemi, budou schopni tento postup zopakovat v jiném prostředí, například doma.	
Předchozí znalosti	Znalost práce s webovým prohlížečem. Orientace v prostředí komplexní webové aplikace. Základní znalost anglického jazyka.	
Mezipředmětové vztahy	Anglický jazyk, IT předměty.	
Časový plán	Fáze činnosti	Metody a formy, motivace
volitelně 5m	Seznámení a ukázka práce s Moodle.	Frontální výklad, ukázka
volitelně 10m	Přihlášení studentů k LMS Moodle, orientace v prostředí, vyplnění vstupního dotazníku.	Samostatná práce s dopomocí
volitelně 10m	Ukázka založení účtu pro Google služby. Založení účtů studenty.	Ukázka a samostatná práce s dopomocí
5m	Ukázka přihlášení k prostředí App Inventoru a základní orientace v prostředí. Ukázka vytvoření prvního programu, ukázka funkčnosti prvního programu na zařízení nebo v emulátoru.	Frontální výklad, ukázka, diskuze
10m	Přihlášení studentů do prostředí App Inventoru, tvorba prvního programu, připojení zařízení nebo emulátoru, spuštění programu.	Samostatná práce s dopomocí, zhodnocení a diskuze
10-45m	Modifikace programu, řešení individuálních potřeb, průběžná diskuze a spolupráce mezi studenty.	Samostatná a skupinová práce
15m	Prezentace prací žáků, sdílení poznatků, diskuze. Motivace žáků k domácí práci.	Prezentace prací a diskuze
Hodnocení	Zpětná vazba je okamžitá. Student reflektuje srovnání cílů aktivity a vlastního postupu. Student průběžně srovnává vlastní výsledky s výsledky ostatních studentů.	
Návaznosti	Na tuto aktivitu navazují následující aktivity, které prohlubují znalost prostředí App Inventoru.	
Poznámky	Úspěšnost velmi závisí na přípravě prostředí. Doporučuji nepodceňovat přípravu a dopředu připravit a otestovat funkčnost emulátorů, případně schopnost WiFi obsloužit najednou všechna zařízení studentů, připravit záložní Google účty. Pro případ výpadku připojení k internetu doporučuji připravit záložní aktivitu, založenou na podobných principech blokového programování, například offline aplikaci MIT Scratch, BYOB nebo některou z variant programovacích prostředí LOGO.	

#### Doporučený multimediální materiál

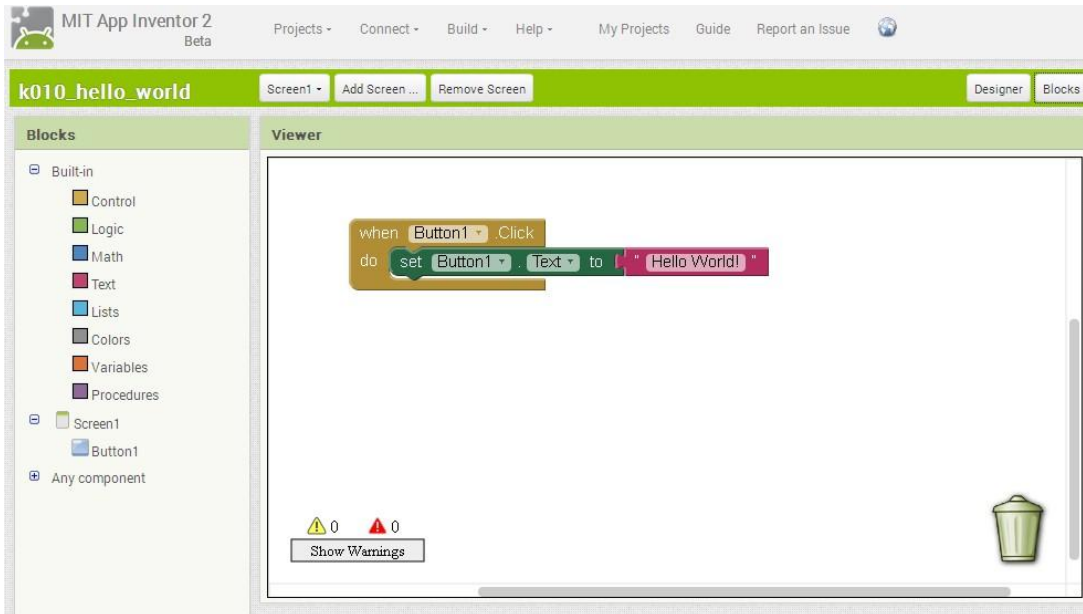
PDF tutoriál k seznámení s prostředím App Inventoru a vytvoření první aplikace.

Video pro seznámení s aplikací a tvorbou první aplikace. Doporučuji však provést vlastní ukázkou.

(materiály viz. on-line kurz)

## Ukázka aplikace Hello World

Účelem je předvést snadnost vytvoření aplikace a ověřit propojení App Inventoru se zařízením studenta nebo s emulátorem.

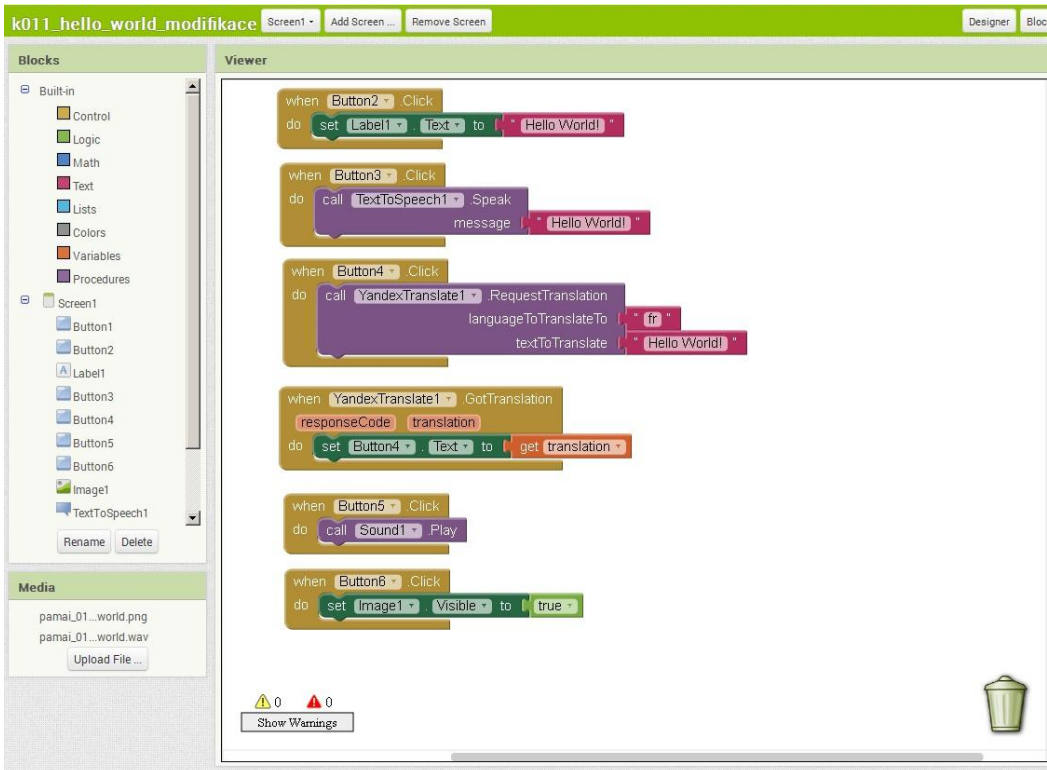
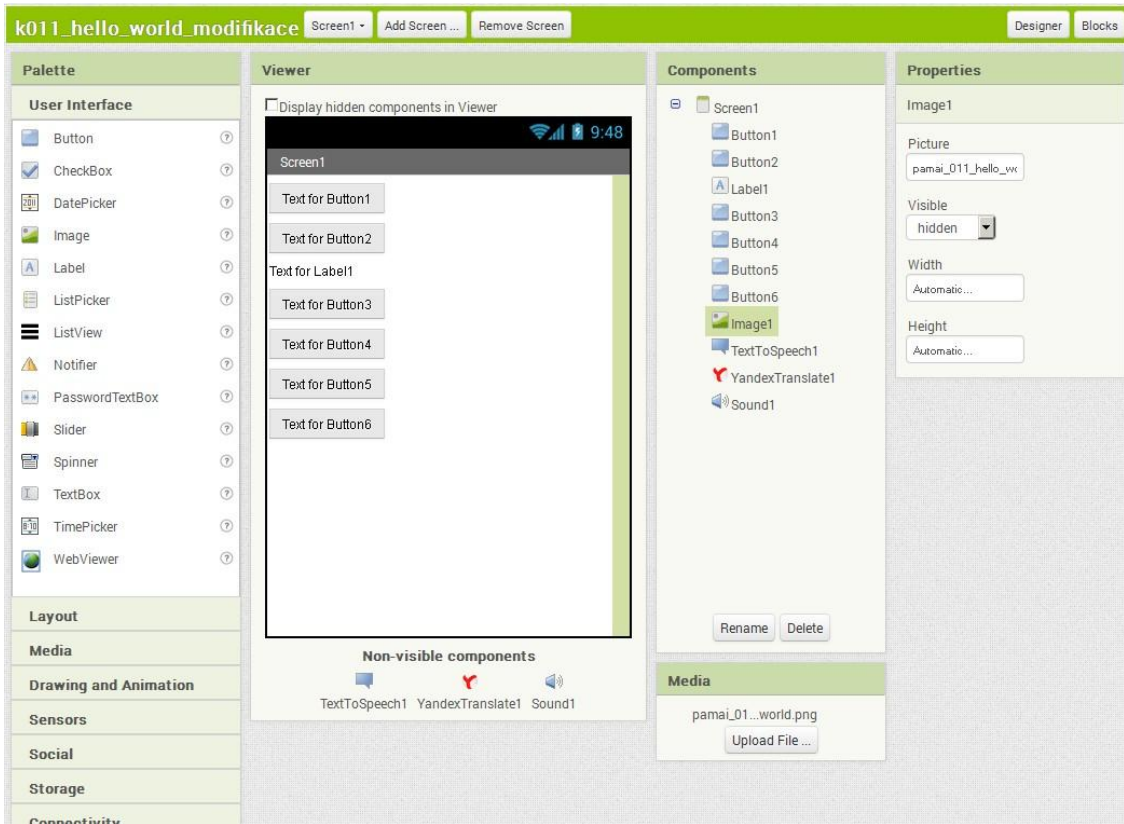


## Modifikace první aplikace

V části modifikací, může být studentům doporučeno několik úprav, ve kterých se opakuje námět "Hello World" v různých modifikacích. Účelem této části lekce je dát prostor studentům, aby se zamysleli nad možnostmi, které jim App Inventor přináší, a vyzkoušeli, nakolik si s tím prostředím rozumí.

V ukázce je obrazovka designeru a bloků programu, který je rozšířen o další funkce. Podobně jako v základním programu, i zde je každá akce spuštěna kliknutím na tlačítko.

- Tlačítko 2 vyvolá akci přejmenování nápisu v komponentě Label1.
- Tlačítko 3 provede strojové čtení textu "Hello World!". Komponenta TextToSpeech je vložena v designeru ze sekce Media. Postup je zobrazen v dříve zmíněném ukázkovém videu video se seznámením. (viz. on-line kurz)
- Tlačítko 4 spustí akci překladu textu. Text "Hello World!" je přeložen pomocí komponenty YandexTranslate opět v designeru ze sekce Media. Po přeložení je zavolána akce when yandexTranslate1.GotTranslation a přeložený text je vložen do názvu tlačítka
- 4. Popis práce s touto komponentou je na videu s ukázkou práce s YandexTranslate. (viz. on-line kurz)
- Tlačítko 5 provede akci přehrátí zvuku "Hello World". Tato akce, stejně jako akce tlačítka 3, vyžaduje, aby zařízení nebo emulátor správně pracovaly se zvukem.
- Tlačítko 6 zobrazí skrytý obrázek s textem "Hello World". Obrázek byl vložen pomocí komponenty Image ze sekce User Interface, následně byl uploadován pomocí vlastnosti Picture (sloupec vpravo) a poslední úprava byla skrytí obrázku pomocí vlastnosti Visiblehidden.



## 5 Aktivita 2 - aplikace využívající kreslicí plochu "canvas"

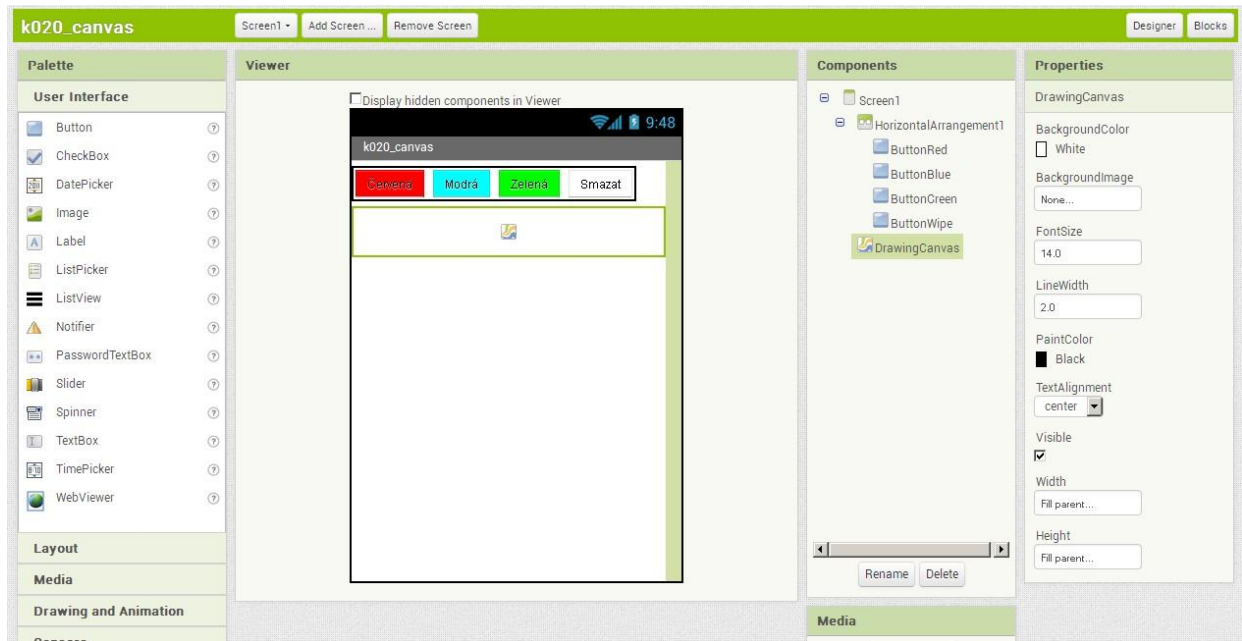
Tematický celek	Kreslení do komponenty kreslicí plocha "canvas"	
Motivační rámec	Tvorba vlastní grafické aplikace s vizuálními efekty	
Počet žáků	5-15	
Věk žáků	10+	
Pomůcky	PC odpovídajících parametrů, webový prohlížeč Mozilla Firefox nebo Google Chrome, zařízení s Androidem připojené k internetu, případně emulátor zařízení s Androidem, účet pro Google služby.	
Stručný popis aktivity	Přihlášení k prostředí App Inventoru <a href="http://appinventor.mit.edu/">http://appinventor.mit.edu/</a> . Tvorba, testy a ladění aplikace v online vývojovém prostředí.	
Vhodné místo	Počítačová učebna	
Cíle aktivity	Studenti zvládnou vytvoření jednoduchého grafického editoru, jehož funkce mohou dále rozvíjet.	
Předchozí znalosti	Aktivita navazuje na předchozí lekci tohoto kurzu.	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
volitelně 5-10m	Zopakování základních informací o prostředí App Inventoru. Ukázka prostředí a popis ovládání.	Frontální výklad, ukázka
5m	Seznámení s tématem lekce. Volitelně ukázka funkční aplikace. Seznámení s výukovými materiály.	Frontální výklad, ukázka
25m	Tvorba aplikace dle tutoriálu a videonávodu.	Samostatná práce s využitím výukových materiálů
15-45m	Modifikace aplikace	Samostatná a skupinová práce
15m	Prezentace prací žáků, sdílení poznatků, diskuze. Motivace žáků k domácí práci.	Prezentace prací a diskuze
Hodnocení	Zpětná vazba je okamžitá. Student reflektuje vzdělávací cíle, svůj postup a dosažené výsledky. Student průběžně srovnává vlastní výsledky s výsledky ostatních studentů.	
Návaznosti	Na tuto aktivitu navazují následující aktivity, které prohlubují znalost prostředí V App Inventoru.	
Poznámky	V případě, že by použití odkazovaných výukových zdrojů v anglickém jazyce činilo studentům problémy, je doporučen postup projít s nimi krok za krokem základní aplikaci a komentovat jednotlivé kroky.  Pro případ výpadku připojení k internetu doporučuji připravit záložní aktivitu, založenou na podobných principech blokového programování, například offline aplikaci MIT Scratch, BYOB nebo některou z variant programovacích prostředí LOGO.	

### Doporučený multimediální materiál

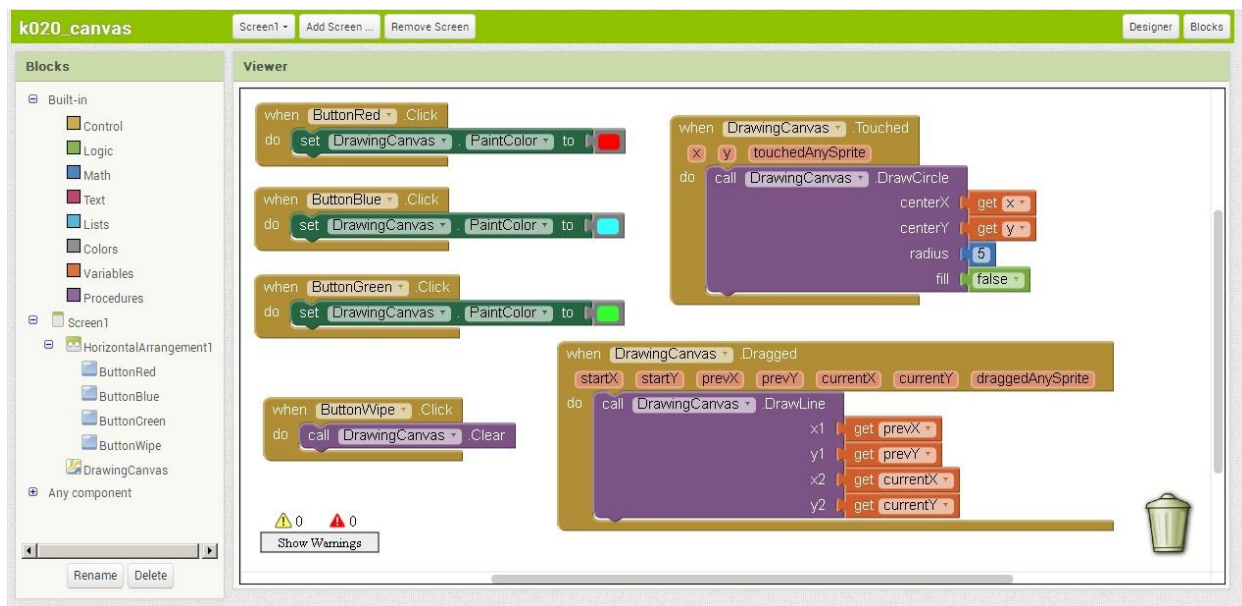
Tutoriál věnující se práci s kreslicí plochou (canvas). Základní aplikace se provedením záměrně blíží aplikaci v tutoriálu. Některé prvky, jako například obrázek na pozadí, jsou z důvodu vyšší přehlednosti vynechány. Uživatelské prostředí aplikace je v českém jazyce, identifikátory objektů v programu jsou pojmenovány v anglickém jazyce pro zachování shody s tutoriálem. (viz. on-line kurz)

## Aplikace pro práci s kreslicí plochou "canvas"

Ukázka základní aplikace pro práci s kreslicí plochou, design.



Ukázka kódu základní aplikace pro práci s kreslicí plochou.



## Modifikace základní aplikace

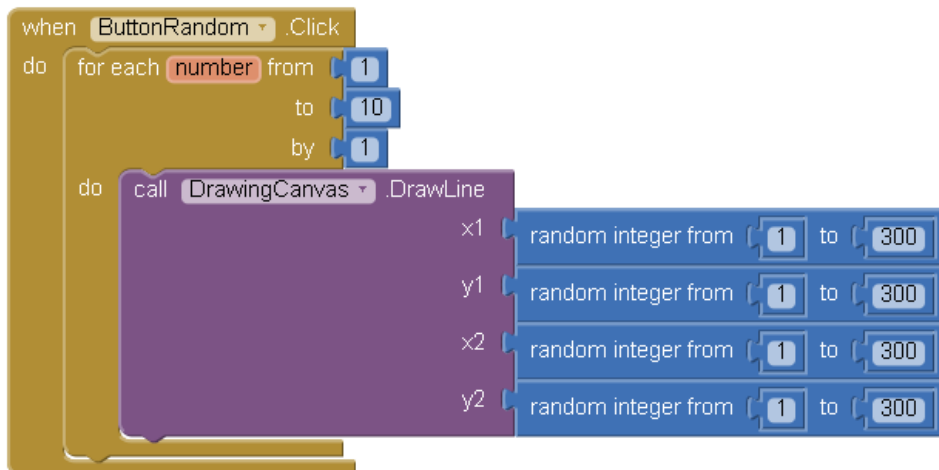
Studenti mohou být ponecháni v experimentování s kreslicí plochou a kombinováním znalostí z předchozí lekce, případně mohou být nasměrováni ke konkrétním modifikacím.

Např.:

- doplnění obrázku na pozadí, přidání více barev,
- generování náhodné kresby **viz obrázek níže**, kresba různých geometrických tvarů,
- zvětšování a zmenšování kreslených geometrických tvarů a čáry, namíchání náhodné barvy,
- uložení obrázku do souboru,
- další náměty lze nalézt ve sborníku kódu <https://puravidaapps.com/snippets1.php#canvas>.



Ukázka kódu generujícího náhodné čáry.



```
when ButtonRandom .Click
do
  for each number from 1 to 10 by 1
  do
    call DrawingCanvas .DrawLine
      x1 random integer from 1 to 300
      y1 random integer from 1 to 300
      x2 random integer from 1 to 300
      y2 random integer from 1 to 300
```

The image shows a Scratch script. It starts with a 'when ButtonRandom .Click' block. Inside, there is a 'do' block containing a 'for each number from 1 to 10 by 1' loop. Inside the loop is a 'do' block containing a 'call DrawingCanvas .DrawLine' block. This block has four arguments: 'x1' (random integer from 1 to 300), 'y1' (random integer from 1 to 300), 'x2' (random integer from 1 to 300), and 'y2' (random integer from 1 to 300).

## 6 Aktivita 3 - aplikace používající grafické objekty "sprites"

Tematický celek	Manipulace s grafickými objekty "sprity"	
Motivační rámec	Tvorba interaktivní grafické aplikace	
Počet žáků	5-15	
Věk žáků	10+	
Pomůcky	PC odpovídajících parametrů, webový prohlížeč Mozilla Firefox nebo Google Chrome, zařízení s Androidem připojené k internetu, případně emulátor zařízení s Androidem, účet pro Google služby.	
Stručný popis aktivity	Přihlášení k prostředí App Inventoru <a href="http://appinventor.mit.edu/">http://appinventor.mit.edu/</a> Tvorba, testy a ladění aplikace v online vývojovém prostředí.	
Vhodné místo	Počítačová učebna	
Cíle aktivity	Studenti zvládnou vytvoření interaktivní grafické aplikace.	
Předchozí znalosti	Aktivita navazuje na předchozí lekci tohoto kurzu.	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
5-10m	Seznámení s tématem lekce. Volitelně ukázka funkční aplikace. Seznámení s výukovými materiály.	Frontální výklad, ukázka
5-10m	Volitelně informace k tvorbě obrázků a průhlednosti. Volitelně ukázka práce s grafickým editorem, export a upload. Volitelně popis kolizí grafických objektů.	Frontální výklad, ukázka
35m	Tvorba aplikace dle tutoriálu a videonávodu.	Samostatná práce s využitím výukových materiálů
15-45m	Modifikace aplikace	Samostatná a skupinová práce
15m	Prezentace prací žáků, sdílení poznatků, diskuze. Motivace žáků k domácí práci.	Prezentace prací a diskuze
Hodnocení	Zpětná vazba je okamžitá. Student reflektuje vzdělávací cíle, svůj postup a dosažené výsledky. Student průběžně srovnává vlastní výsledky s výsledky ostatních studentů.	
Návaznosti	Na tuto aktivitu navazují následující aktivity, které prohlubují a kombinují znalosti prostředí App Inventoru.	
Poznámky	V případě, že by použití odkazovaných výukových zdrojů v anglickém jazyce činilo studentům problémy, je doporučen postup projít s nimi krok za krokem základní aplikaci a komentovat jednotlivé kroky.  Pro případ výpadku připojení k internetu doporučuji připravit záložní aktivitu, založenou na podobných principech blokového programování, například offline aplikaci MIT Scratch, BYOB nebo některou z variant programovacích prostředí LOGO.	

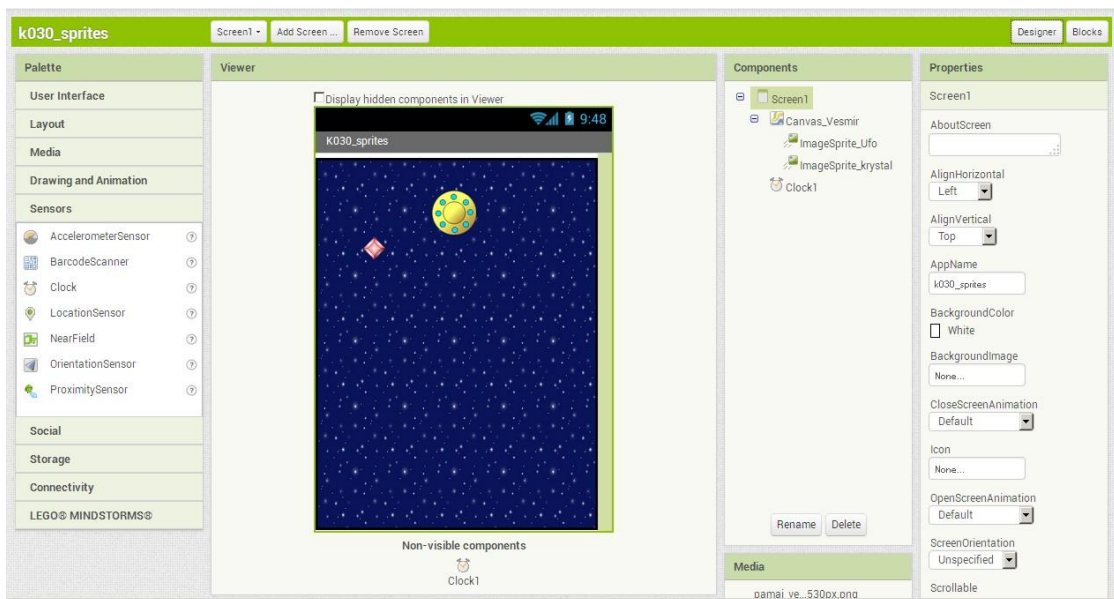
### Doporučený multimediální materiál

Tutoriál provádějící užitím grafických objektů, sprites. (viz. on-line kurz)

## Aplikace pro ukázkou práce s grafickými objekty

Ukázka modelové aplikace.

TIP: Pohyb sprite se řídí z designeru, při vybraném sprite, hodnotou Speed v menu Properties. Grafické soubory naleznete v sekci Grafické soubory použité v kurzu.



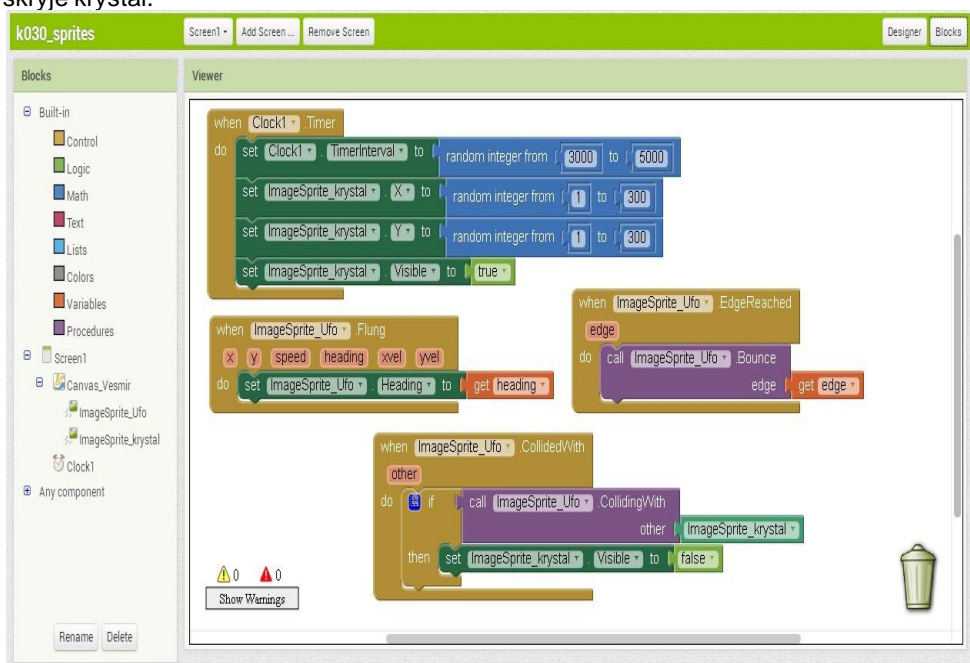
## Ukázka kódu základní aplikace pro práci s grafickými objekty

Bloky when Clock1.timer = dle nastaveného intervalu provádí v náhodném čase 3 až 5 sekund přesun krystalu, jeho součástí je zviditelnění krystalu, pro případ, že by byl skryt.

When ImageSprite\_Ufo.Flung = zajišťuje změnu směru sprite určenou na základě přejetí prstem.

When ImageSprite\_Ufo.EdgeReached = zajišťuje akci v případě, kdy se UFO dostane k okraji plochy.

When ImageSprite\_Ufo.CollideWith = obsluha akce kolize s jiným sprite. V případě, že je kolidujícím objektem krystal, skryje krystal.



## **Modifikace základní aplikace**

Studenti mohou být ponecháni v experimentování s grafickými objekty a kombinováním znalostí z předchozí lekce, případně mohou být nasměrováni ke konkrétním modifikacím.

Např:

doplnění více sprites,  
export 3D sprites ze zdrojového grafického souboru, tvorba vlastních obrázků pro sprites,  
změna chování sprites při detekci kolize meteory jako překážky.

## 7 Aktivita 4 - použití proměnných "variable"

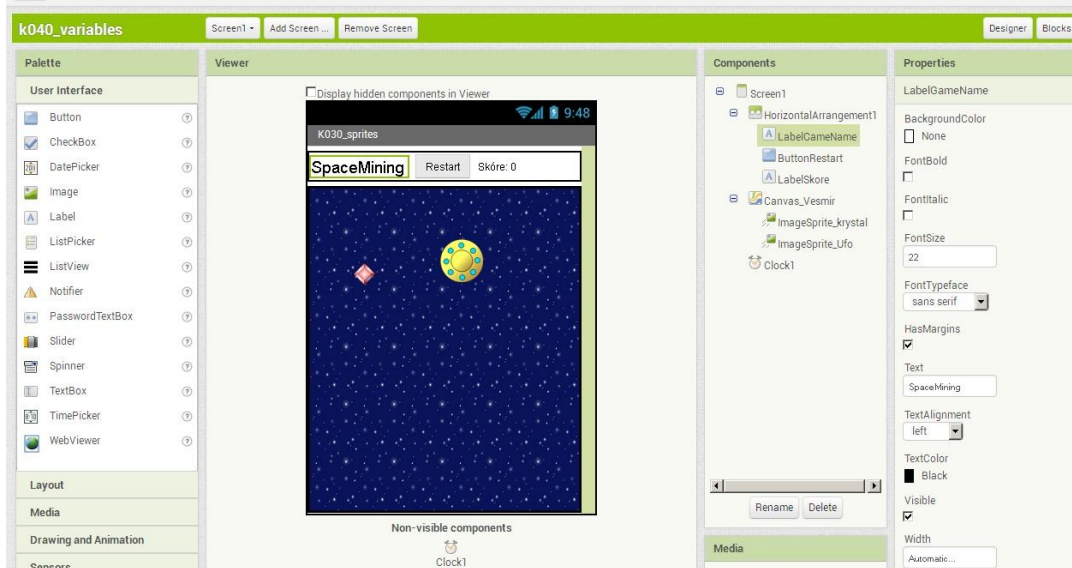
Tematický celek	Použití proměnných "variables"	
Motivační rámec	Rozšíření funkcí interaktivní grafické aplikace.	
Počet žáků	5-15	
Věk žáků	10+	
Pomůcky	PC odpovídajících parametrů, webový prohlížeč Mozilla Firefox nebo Google Chrome, zařízení s Androidem připojené k internetu, případně emulátor zařízení s Androidem, účet pro Google služby.	
Stručný popis aktivity	Přihlášení k prostředí App Inventoru <a href="http://appinventor.mit.edu/">http://appinventor.mit.edu/</a> Tvorba, testy a ladění aplikace v online vývojovém prostředí.	
Vhodné místo	Počítačová učebna	
Cíle aktivity	Studenti si vyzkouší použití proměnných, jejich inicializaci, nastavování a změnu hodnot, používání proměnných.	
Předchozí znalosti	Aktivita navazuje na předchozí lekci tohoto kurzu.	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
5-10m	Seznámení s tématem lekce. Volitelně ukázka funkční aplikace. Seznámení s tématem proměnných. Globální vs lokální proměnné.	Frontální výklad, ukázka
25m	Tvorba aplikace dle ukázek a popisu v této lekci.	Samostatná práce s využitím výukových materiálů tohoto kurzu
15-60m	Modifikace aplikace. Volitelně, inspirace v tutoriálu hry	Samostatná a skupinová práce
15m	Prezentace prací žáků, sdílení poznatků, diskuze. Motivace žáků k domácí práci.	Prezentace prací a diskuze
Hodnocení	Zpětná vazba je okamžitá. Student reflektuje vzdělávací cíle, svůj postup a dosažené výsledky. Student průběžně srovnává vlastní výsledky s výsledky ostatních studentů.	
Návaznosti	Na tuto aktivitu navazují následující aktivity, které prohlubují a kombinují znalosti prostředí App	
Poznámky	Pro případ výpadku připojení k internetu nebo technických problémů s App Inventorem doporučuji připravit záložní aktivitu, založenou na podobných principech blokového programování, například offline aplikaci MIT Scratch, BYOB nebo některou z variant programovacích prostředí LOGO.	

### Doporučený multimediální materiál

Tutoriál ukazující jinou techniku práce s proměnnou, využívající hodnotu v komponentě label. (viz. on-line kurz)

## Aplikace pro ukázkou práce s proměnnými

Zde proti předchozí verzi přibylo několik komponent. Pomocí Layout manageru jsou uspořádány popisek s názvem hry, tlačítko pro restart hry a popisek, pomocí kterého bude zobrazováno skóre hry.

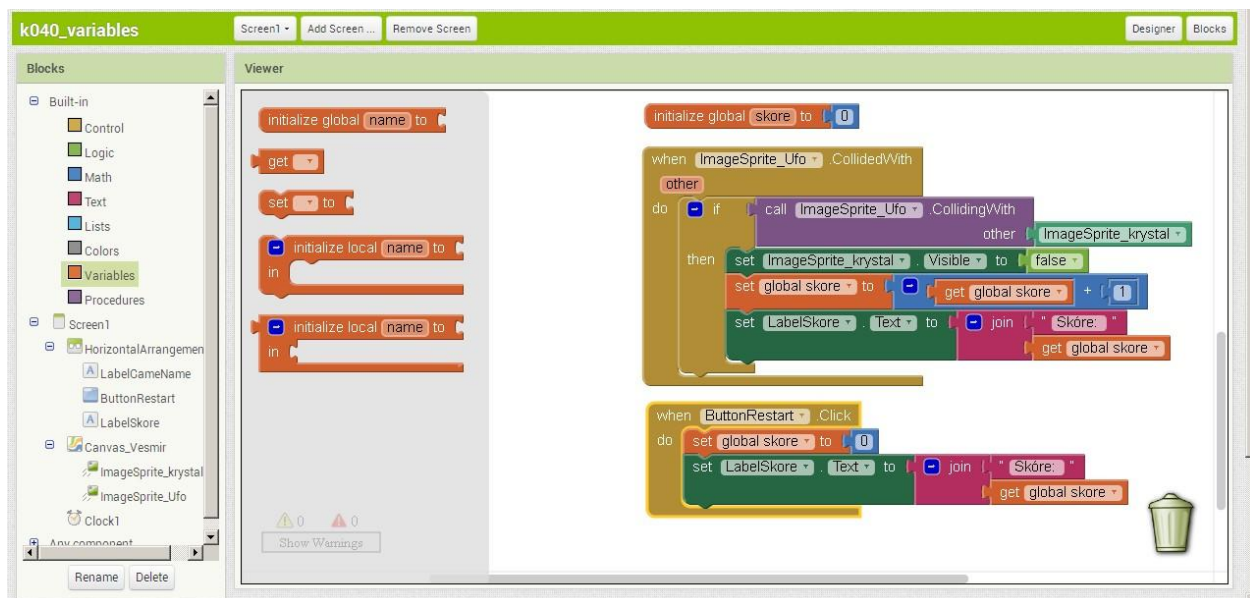


Ukázka kódu základní aplikace pro práci s proměnnými Bloky:

Initialize global score to 0 = výchozí inicializace globální proměnné, ke které lze přistupovat z libovolných bloků programu.

When ImageSprite\_Ufo.ColideWith = blok obsluhy kolize s jiným sprite byl rozšířen a při detekci kolize je navyšována hodnota v proměnné score o 1 a nová hodnota je zobrazována v LabelSkore.

When ButtonRestart Click = provede akci vynulování skóre po kliknutí na tlačítko reset.



## Modifikace základní aplikace

Doporučené modifikace programu:

doplnění více sprites krystalů a přičítání různého skóre pro různé krystaly, vytvoření proměnné pro počet životů a odečítání životů při srážce s meteoridem, vytvoření proměnné pro rychlost UFA a změna rychlosti při sebrání "vylepšení", zrychlení jen na omezený čas, nastavení původní rychlosti po restartu.

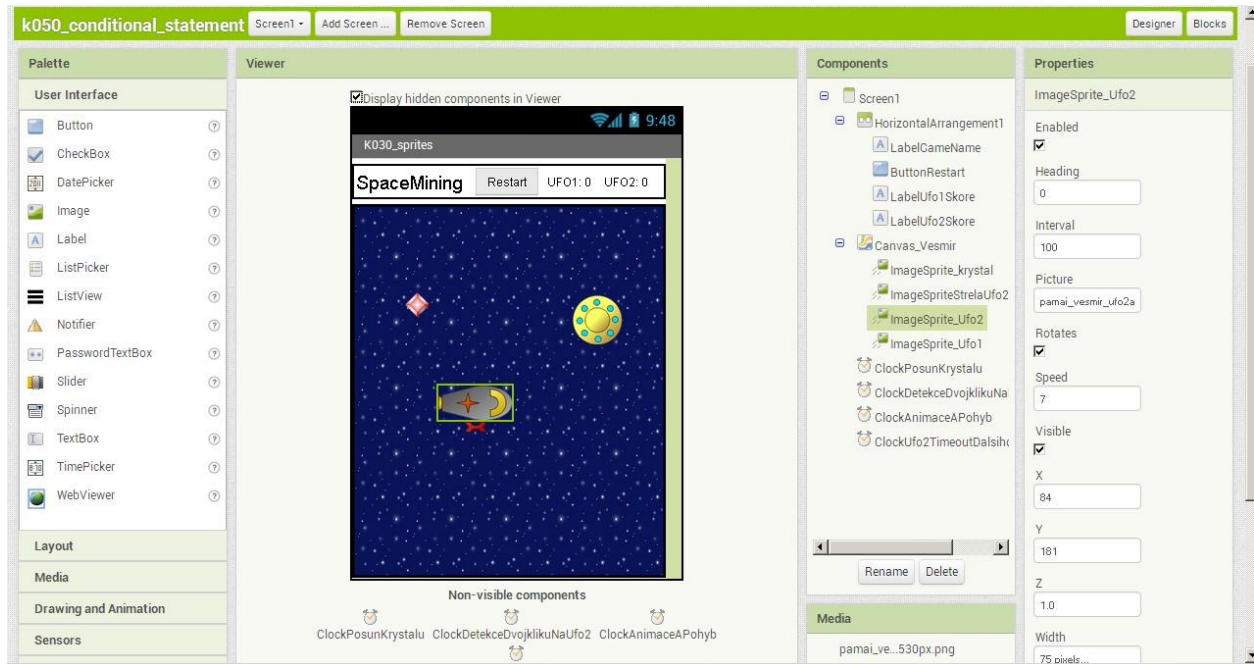
## 8 Aktivita 5 - podmíněné větvení programu

Tematický celek	Použití podmíněného větvení programu	
Motivační rámec	Multiplayer v interaktivní grafické aplikaci	
Počet žáků	5-15	
Věk žáků	10+	
Pomůcky	PC odpovídajících parametrů, webový prohlížeč Mozilla Firefox nebo Google Chrome, zařízení s Androidem připojené k internetu, případně emulátor zařízení s Androidem, účet pro Google služby.	
Stručný popis aktivity	Přihlášení k prostředí App Inventoru <a href="http://appinventor.mit.edu/">http://appinventor.mit.edu/</a> Tvorba, testy a ladění aplikace v online vývojovém prostředí.	
Vhodné místo	Počítačová učebna	
Cíle aktivity	Studenti si vyzkouší použití několika způsobů větvení programu.	
Předchozí znalosti	Aktivita navazuje na předchozí lekci tohoto kurzu.	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
5-25m	Seznámení s tématem lekce. Volitelně ukázka funkční aplikace. Seznámení s tématem větvení programu. Neúplné podmíněné větvení "if", úplné podmíněné větvení "if / else", vícenásobné podmíněné větvení "if / else if / else". Volitelně nákres vývojových diagramů. Volitelně samostatná činnost žáků, vyhledávání informací k tomuto tématu.	Frontální výklad, ukázka, samostatná práce žáků
30-60m	Úprava vlastní aplikace dle ukázek a popisu v této lekci.	Samostatná a skupinová práce s využitím výukových materiálů tohoto kurzu
15-20m	Modifikace aplikace.	Samostatná a skupinová práce
15m	Prezentace prací žáků, sdílení poznatků, diskuze. Motivace žáků k domácí práci.	Prezentace prací a diskuze
Hodnocení	Zpětná vazba je okamžitá. Student reflektuje vzdělávací cíle, svůj postup a dosažené výsledky. Student průběžně srovnává vlastní výsledky s výsledky ostatních studentů.	
Návaznosti	Na tuto aktivitu navazují následující aktivity, které prohlubují a kombinují znalosti prostředí App Inventoru.	
Poznámky	Pro případ výpadku připojení k internetu nebo technických problémů s App Inventorem doporučuji připravit záložní aktivitu, založenou na podobných principech blokového programování, například offline aplikaci MIT Scratch, BYOB nebo některou z variant programovacích prostředí LOGO.	

Video s tipem, jak je možné pracovat s animacemi. (viz. on-line kurz)

## Aplikace pro ukázkou podmíněného větvení programu

Opět přibýlo několik komponent, především několik časovačů, loď pro druhého hráče, střela.



Ukázka části kódu aplikace, ve které je využito několik typů podmíněného větvení programu. Nově přibyla funkčnost:

- loď druhého hráče a obsluha jejího pohybu,
- obsluha výstřelu při dvojkliku na loď druhého hráče, z pozice x:y lodi ve směru letu lodi, animace letící střely,
- animace motorů lodi druhého hráče,
- obsluha kolize střely druhého hráče s lodí prvního hráče, zastavení lodi prvního hráče po zásahu střelou druhého hráče, obsluha kolize lodi druhého hráče s krystalem a počítání skóre,
- obnova nastavení pro střelu a druhého hráče při stisku tlačítka reset, timeout zabraňující opakování výstřelů rychle za sebou.

Významné bloky:

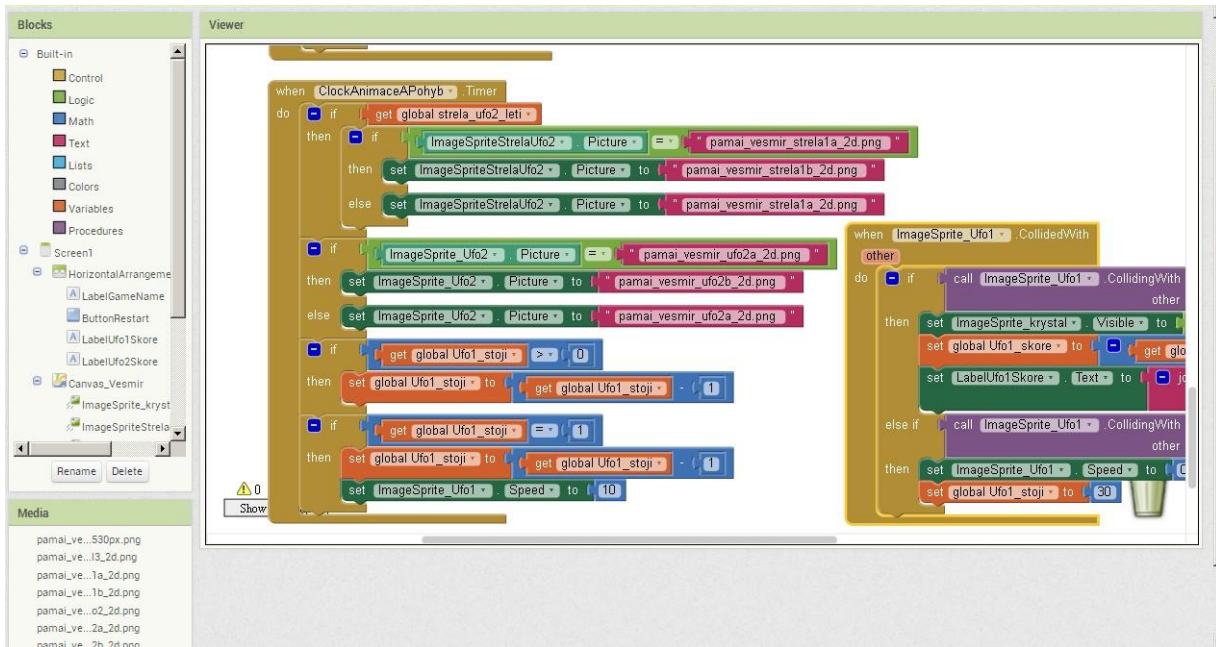
when ClockAnimaceAPohyb.Timer = časovač tikající po 100ms, řešící animaci (změnu obrázků) střely, v případě, že je letící. Animace motoru lodi druhého hráče. Řeší i zastavení lodi prvního hráče po zásahu střelou. Zastavení je řízeno proměnnou Ufo1\_stoji, která je při zásahu nastavena na hodnotu 30. Loď stojí po dobu cca 30x 100Ms.

When ImageSprite\_Ufo.ColideWith = V bloku kolize je navíc řešena obsluha zásahu střelou.

When ImageSprite\_Ufo2.Touched = detekce dvojkliku na loď druhého hráče. V tomto případě je snaha o odfiltrování dotyku, který nastává při ovládní směru pohybu lodi. Za dvojklik je považován dvojitok lodi, který nastane v čase menším než 300Ms.

When ClockDetekceDvojklikuNaUfo2 = pomocný blok, který pomáhá při detekci dvojkliku.





## Modifikace základní aplikace

Doporučené modifikace programu:

možnost střelby i pro prvního hráče, druhý hráč reagující na zásah, animace střelby prvního hráče, animace lodi prvního hráče, detekce kolize lodí hráčů a přenesení lodí na protilehlé strany plochy, životy hráčů a odečítání při zásazích, zastavení a vyhodnocení hry po určitém časovém intervalu / po dosažení počtu bodů.

## 9 Aktivita 6 - cykly a procedury jako řídicí struktury programu

Tematický celek	Použití cyklů a procedur jako dalších řídicích struktur programu	
Motivační rámec	Seznámení s důležitým aspektem programování s širokým využitím při vlastní tvorbě.	
Počet žáků	5-15	
Věk žáků	10+	
Pomůcky	PC odpovídajících parametrů, webový prohlížeč Mozilla Firefox nebo Google Chrome, zařízení s Androidem připojené k internetu, případně emulátor zařízení s Androidem, účet pro Google služby.	
Stručný popis aktivity	Přihlášení k prostředí App Inventoru <a href="http://appinventor.mit.edu/">http://appinventor.mit.edu/</a> Teoretické a praktické seznámení s cykly programu, které jsou k dispozici v programu App Inventor.	
Vhodné místo	Počítačová učebna	
Cíle aktivity	Studenti se seznámí s cykly programu a vyzkouší si jejich použití.	
Předchozí znalosti	Aktivita navazuje na předchozí lekci tohoto kurzu.	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
5-35m	Seznámení s tématem lekce. Dle znalostí studentů seznámení s cykly a procedurami nebo zopakování cyklů a procedur jako dalších struktur pro řízení běhu programu. Seznámení s pojmy/zopakování pojmů jako: nekonečný cyklus/smyčka, cyklus s podmínkou na začátku, cyklus s podmínkou na konci, počítaný cyklus. Volitelně nákres vývojových diagramů. Vysvětlení / připomenutí pojmů procedur / funkcí, jejich definice, volání vstupních a výstupních parametrů.	Frontální výklad, ukázka, samostatná práce žáků
30-60m	Využití znalosti cyklů a procedur při tvorbě vlastní aplikace.	Samostatná a skupinová práce s využitím získaných znalostí, informací v tomto kurzu a externích zdrojů.
15-20m	Prezentace prací žáků, sdílení poznatků, diskuze. Motivace žáků k domácí práci.	Prezentace prací a diskuze
Hodnocení	Zpětná vazba je okamžitá. Student reflektuje vzdělávací cíle, svůj postup a dosažené výsledky. Student průběžně srovnává vlastní výsledky s výsledky ostatních studentů.	
Návaznosti	Na tuto aktivitu navazují následující aktivity, které prohlubují a kombinují znalosti prostředí App Inventoru.	
Poznámky	Pro případ výpadku připojení k internetu nebo technických problémů s App Inventorem doporučuji připravit záložní aktivitu, založenou na podobných principech blokového programování, například offline aplikaci MIT Scratch, BYOB nebo některou z variant programovacích prostředí LOGO.	

Wikipedie k tématu řídicích struktur. (odkaz viz. on-line kurz)

Doplňkový tutoriál na aplikaci Xylofón. Aplikace využívá procedury.

### Aplikace pro ukázkou cyklů a procedur

Zásadní věc je, že App Inventor vytváří jednovláknové aplikace a jakýkoliv blok programu, pokud je spuštěn, je vykonán až do konce. To je třeba mít na paměti, obzvláště pokud jste zvyklí na vícevláknové aplikace. **V programech App Inventoru se každý blok, který své ukončení podmiňuje změnou na jiném místě programu, stává nekonečnou smyčkou.** Z toho důvodu není v App Inventoru implementována funkce pozastavení běhu (sleep / pause / delay) a opakující se akce v určitých intervalech jsou řešeny pomocí časovačů.

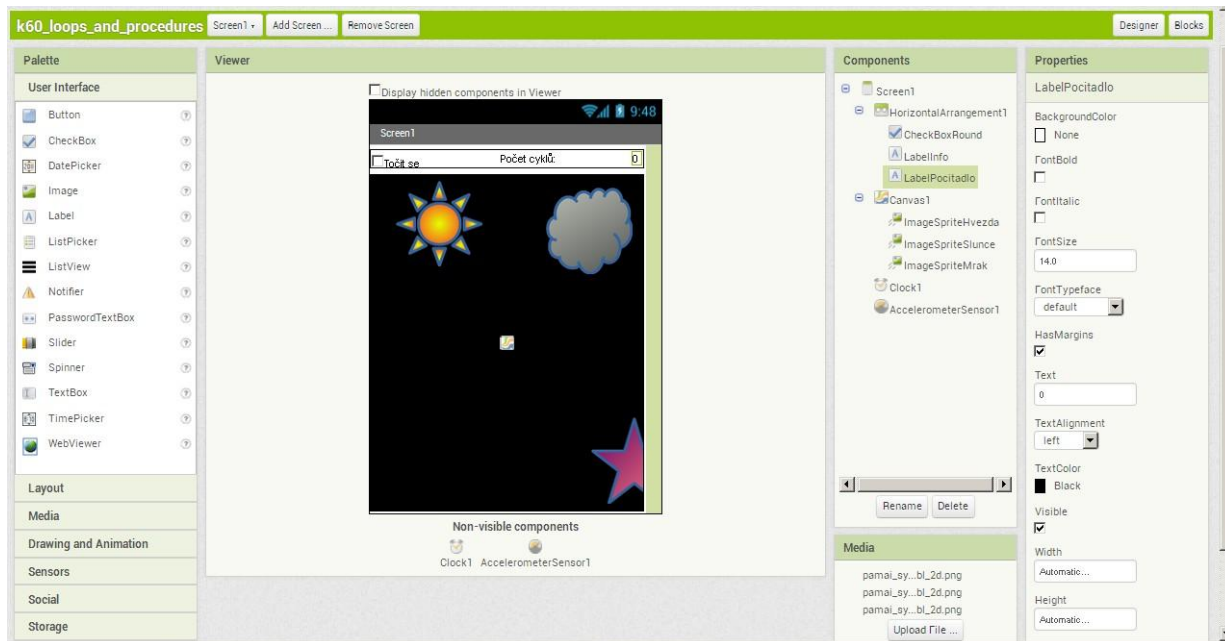
Více o této vlastnosti programu lze nalézt na blogu Jose Florese. (odkaz viz. on-line kurz)

Cyklus "while test" lze používat pro akce, které mají konec uvnitř svého cyklu.

Z pohledu cyklů lze smyčky s podmínkou na začátku nahradit časovači s `TimerEnabled.False` a podmínkou, která časovač zapne nebo nechá vypnutý. Smyčky s podmínkou na konci lze nahradit časovači s `TimerEnabled.True` s podmínkou, která časovače vypne, nebo ponechá zapnuté. Lze použít i jiné techniky pro náhradocyklů.

Na všechny bloky, které reagují na události, lze také pohlížet jako na smyčky programu s podmínkou na začátku.

### Komponenty modelové aplikace.

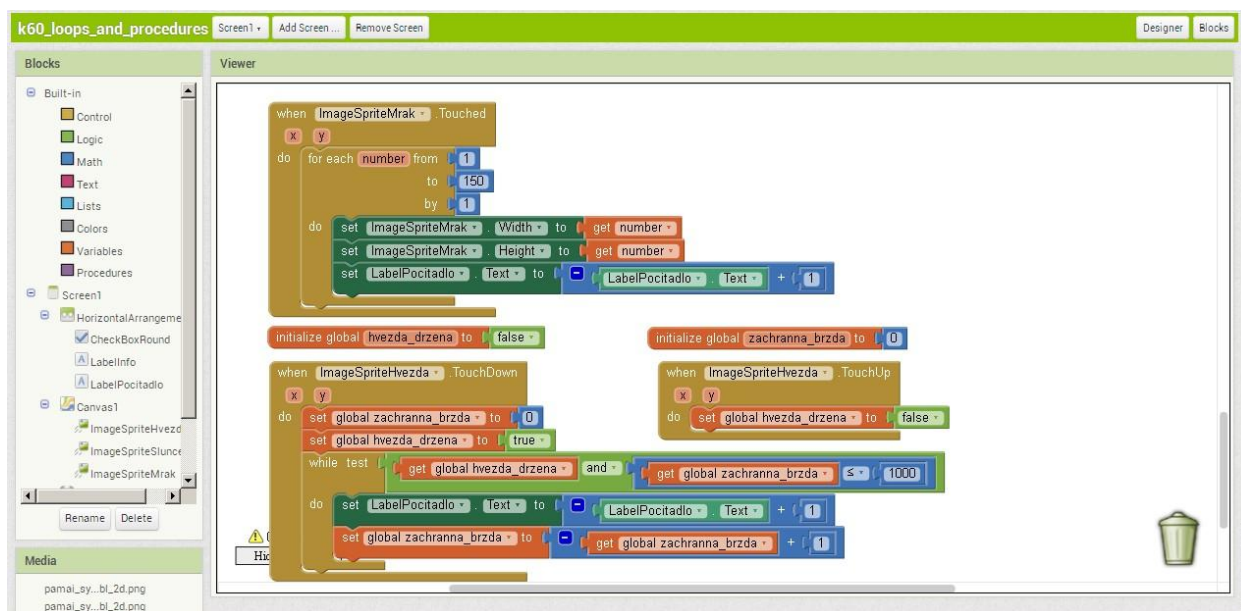


### Ukázka omezení programů z App Inventoru

Blok WhenImageSpriteMrak.Touched je spuštěn po dotyku obrázku mraku. Jde o počítaný cyklus od 1 do 150 s krokem po 1. Vykoná se tedy 150x. Důležité je si všimnout, že není vidět postupné provádění bloku a zvětšování mraku z velikosti 1 na velikost 150.

Program se (dle výkonu zařízení) na chvíli "zasekne" a pak se objeví už zvětšený mrak. Po celou dobu, kdy byl blok prováděn, nedostal program ani možnost překreslovat sprite mraku a zobrazil až výsledek po skončení bloku.

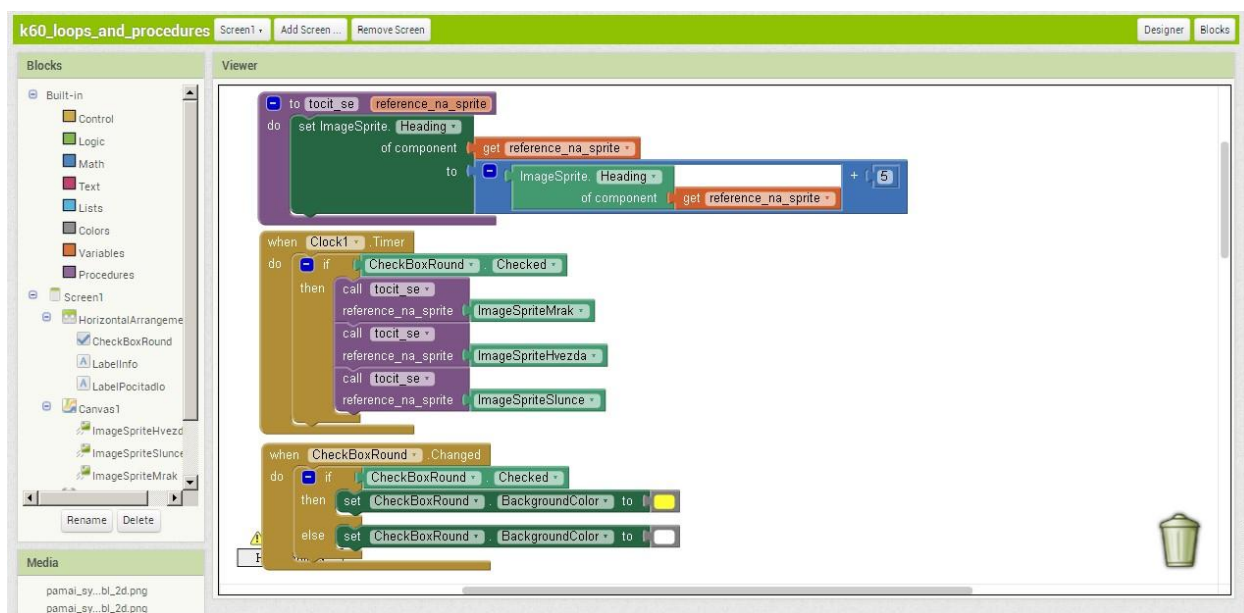
Blok WhenImageSpriteHvezda.TouchDown se samotným SpriteHvezda neprovádí nic, ale po dobu držení hvězdy by měl zvyšovat počítadlo průběhu cyklu a měl by přestat přičítat, když je držení ukončeno. To by měl zajistit blok WhenImageSpriteHvezda.TouchUp. Ve skutečnosti ale program na ukončení dotyku nereaguje a zastaví ho až proměnná zachranna\_brzda.



## Procedury a principDRY

Procedury patří k významným řídicím strukturám programu. Umožňují vykonávat stejný kód pro různé objekty. Někdy jsou procedury použity pro oddělení a logické členění zdrojového kódu, což je opět velmi užitečná funkce. Nezřídka o "životnosti" a úspěšnosti programu rozhoduje právě jeho dobrá čitelnost a přehlednost. Na obrázku je ukázka procedury `tocit_se`, které je jako parametr předána reference na sprite. Funkce zajišťuje otáčení sprite a je jí jedno, jaký sprite je jí předán. Místo tří téměř identických kódů pro 3 sprite v tomto případě, nebo v jiném programu, třeba pro 100 sprite, je použit jeden kód pro všechny.

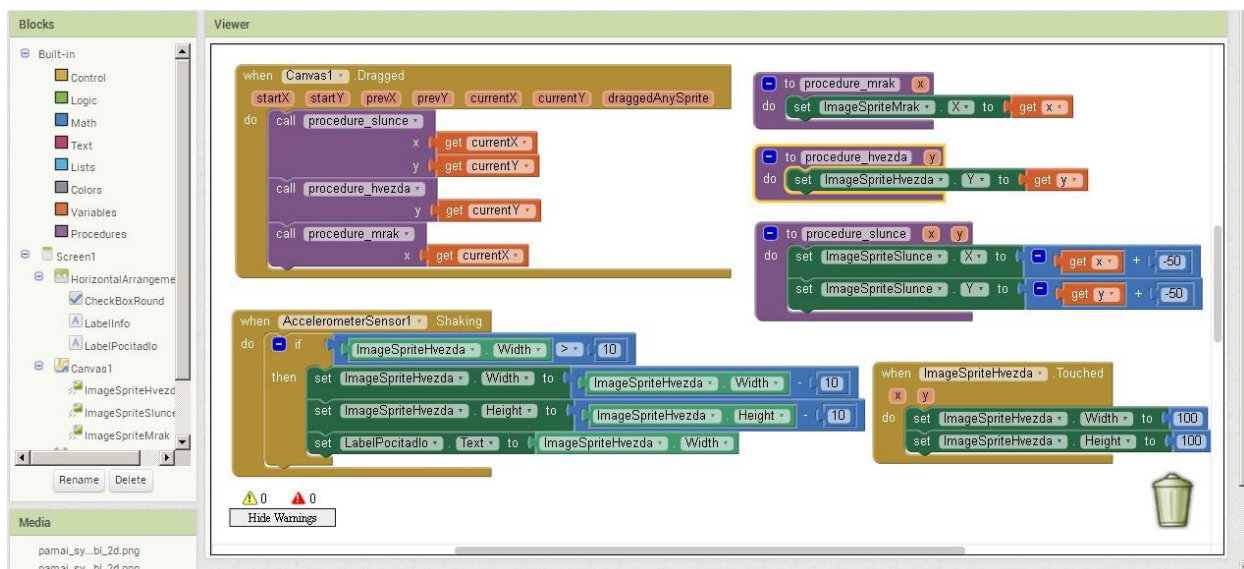
Otáčení je ještě podmíněno zaškrtnutím checkboxu. Jde tedy o variaci na cyklus s podmínkou na začátku.



## Ukázka procedur a použití senzoru akcelerátoru

V poslední části programu je obsluha pohybu objektů po ploše opět řešená pomocí funkcí, které zde plní funkci oddělení kódu. Poslední dva bloky pracují se senzorem akcelerátoru. Při třesení zařízením se hvězda zmenšuje, po dotyku se hvězda zvětší.

Funkce `AccelerationChanged` má v současné verzi (únor 2015 AI nb140b Companion 2.24) chybu, která při použití způsobuje zaseknutí zařízení.



Tipy na modifikace programu:

více sprites s různým chování,  
jiná společná akce všech sprites, například hromadné zmenšení při stisku tlačítka, sprite pohybující se po určené dráze,  
sprite pohybující se po náhodné dráze, malování sprites na canvas.

## 10 Aktivita 7 - datové seznamy a komponenty pro práci se seznamy

Tematický celek	Datový typ seznam a komponenty pro práci se seznamy.	
Motivační rámec	Usnadnění práce s větším množstvím proměnných a zjednodušení programu.	
Počet žáků	5-15	
Věk žáků	10+	
Pomůcky	PC odpovídajících parametrů, webový prohlížeč Mozilla Firefox nebo Google Chrome, zařízení s Androidem připojené k internetu, případně emulátor zařízení s Androidem, účet pro Google služby	
Stručný popis aktivity	Přihlášení k prostředí App Inventoru <a href="http://appinventor.mit.edu/">http://appinventor.mit.edu/</a> Teoretické a praktické seznámení s datovým typem seznam "list".	
Vhodné místo	Počítačová učebna	
Cíle aktivity	Studenti se seznámí s použitím datového typu seznam a komponentami pro práci se seznamy a vyzkouší si je.	
Předchozí znalosti	Aktivita navazuje na předchozí lekci tohoto kurzu.	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
5-15m	Seznámení s tématem lekce. Volitelně předvedení a popis ukázkové aplikace. Dle znalostí studentů zopakování informací o datovém typu seznam a základních operací s tímto datovým typem nebo seznámení s nimi.	Frontální výklad, ukázka.
25-50m	Prohlídka ukázkových aplikací. Modifikace ukázkových aplikací případně využití znalosti seznamů při tvorbě vlastní aplikace.	Samostatná a skupinová práce s využitím získaných znalostí, informací v tomto kurzu a externích zdrojů.
15-30m	Prezentace prací žáků, sdílení poznatků, diskuze. Motivace žáků k domácí práci.	Prezentace prací a diskuze
Hodnocení	Zpětná vazba je okamžitá. Student reflektuje vzdělávací cíle, svůj postup a dosažené výsledky. Student průběžně srovnává vlastní výsledky s výsledky ostatních studentů.	
Návaznosti	Na tuto aktivitu navazují následující aktivity, které prohlubují a kombinují znalosti prostředí App Inventoru.	
Poznámky	Pro případ výpadku připojení k internetu nebo technických problémů s App Inventorem doporučuji připravit záložní aktivitu, založenou na podobných principech blokového programování, například offline aplikaci MIT Scratch, BYOB nebo některou z variant programovacích prostředí LOGO.	

Video o vytváření seznamu v AI2 2 min . Creating lists in App Inventor 2 (viz. on-line kurz)

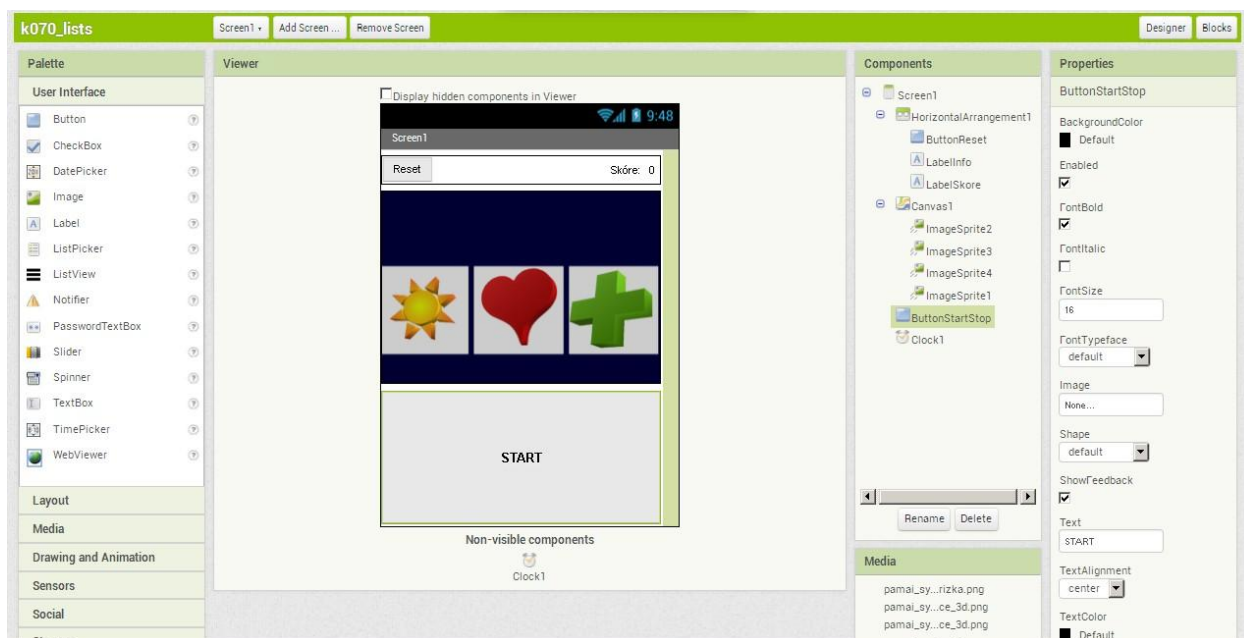
Tutoriál k vytváření seznamu. Making Lists (App Inventor 2) (viz. on-line kurz)

Popis jednotlivých bloků seznamu. AI2 Lists. Aplikace využívající seznamy. Math Blaster (viz. on-line kurz)

### Aplikace pro ukázkou práce s datovými seznamy

Datové seznamy usnadňují práci s programem. Podobně jako procedury zobecňují kód a umožňují ho zjednodušit, seznamy zjednodušují práci s proměnnými a jejich obsahem. Aplikace napodobuje hru videostop. Po stisknutí tlačítka start se mění obrázky tří sprite a tlačítkem stop ze změny zastaví. Hráč zvedá skóre, když se mu podaří zastavit na stejných obrázcích.

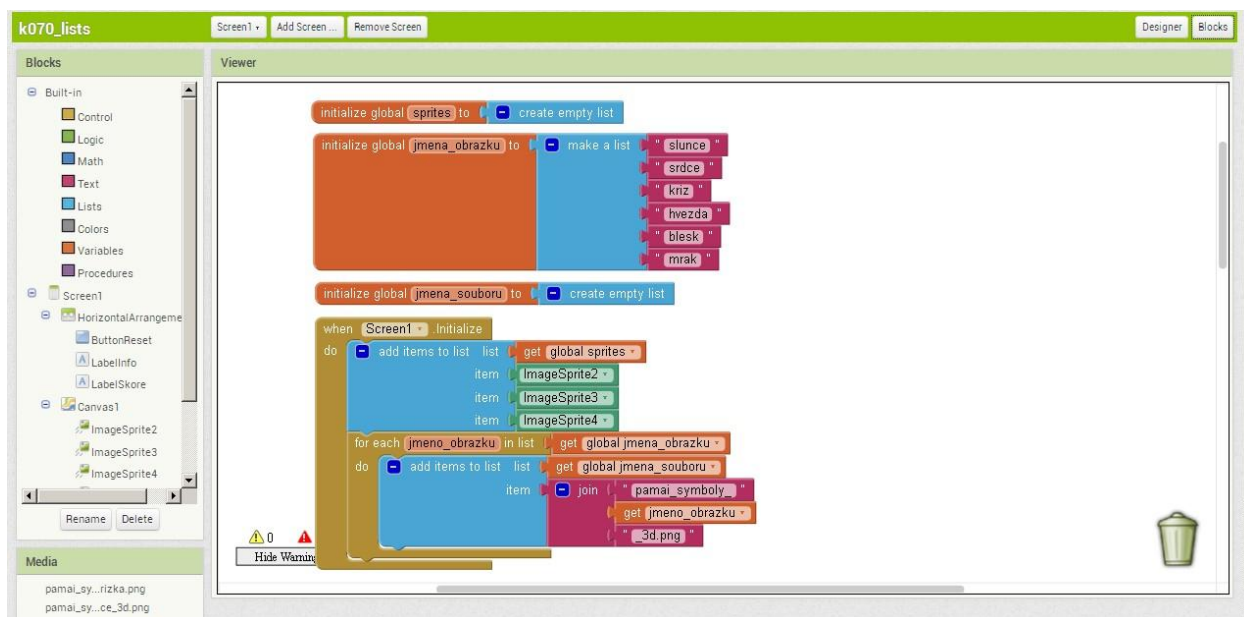
## Komponenty modelové aplikace



### Inicializace seznamů

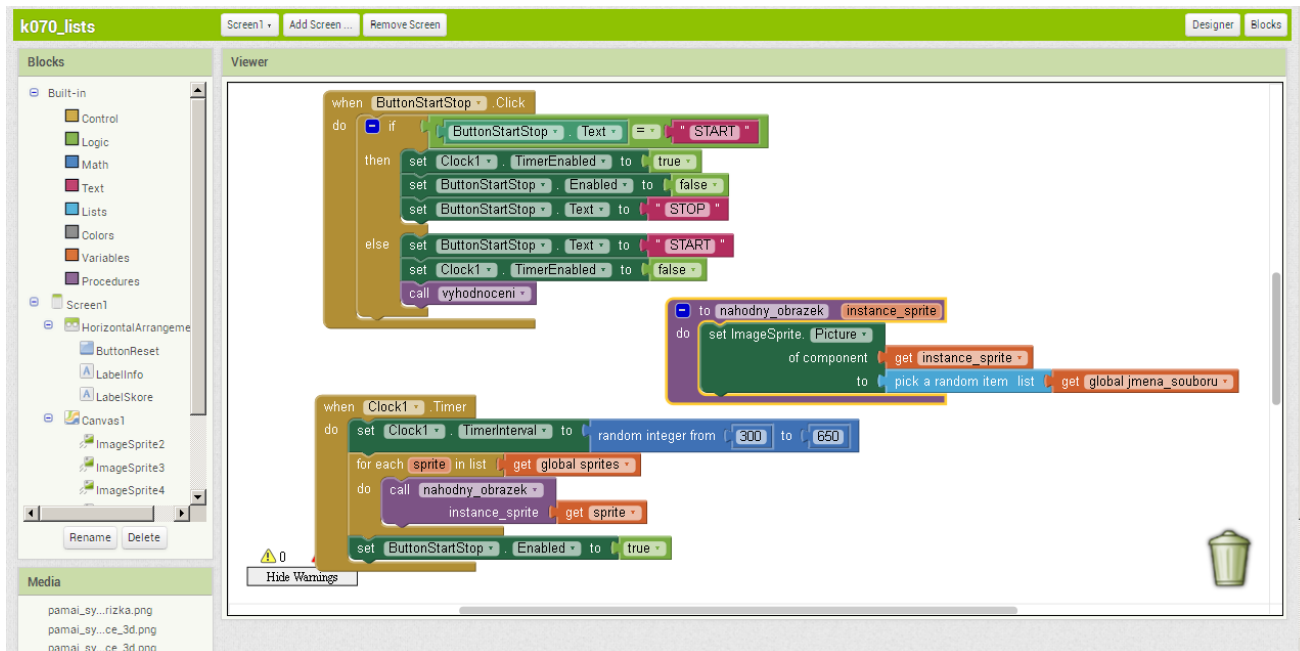
V ukázce jsou vytvořeny tři seznamy. sprites a jmena\_souboru jsou inicializovány jako prázdné seznamy a jsou naplněny v bloku when Screen.Initialize.

Seznam jmena\_obrazku je naplněn při inicializaci. Obsahuje řetězce, kterými se jednotlivé obrázky liší. V bloku when Screen.Initialize je nejdříve vytvořen seznam instancí sprites, následně je vytvořen seznam jmena\_souboru, který obsahuje celé názvy souborů s obrázky. Seznamy tedy mohou obsahovat nejen čísla a textové řetězce, mohou obsahovat také instance objektů.



## Cyklus nad objekty seznamu

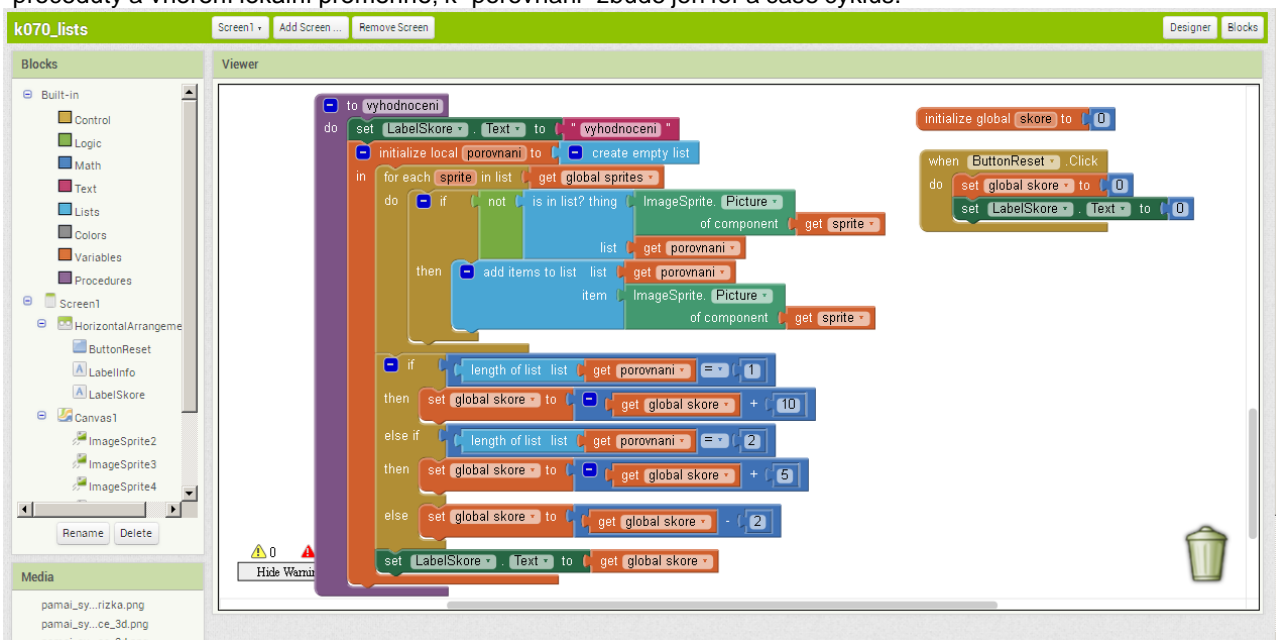
Tlačítkem ButtonStartStop je řízeno pouštění a zastavení časovače. Časovač Clock1 spouští v cyklu "for each sprite in list get global sprites" cyklus nad dříve vytvořeným seznamem sprites. Pro každý sprite zavolá proceduru nahodny\_obrazek, která vybere ze seznamu obrázků jeden náhodný. Po druhém stisknutí tlačítka ButtonStartStop je zavolána funkce pro vyhodnocení skóre.



## Vyhodnocení výsledku

Uvnitř procedury vyhodnoceni je vytvořen seznam jako lokální proměnná. Pak potřebujeme zjistit, kolik obrázků je stejných. Místo porovnávání obrázků každého s každým je tento úkol řešen pomocí seznamu. V cyklu for each se projdou všechny zobrazené obrázky a přidávají se do seznamu, pokud tam ještě nejsou. Pokud jsou po cyklu for each v seznamu 3 obrázky, znamená to, že hráč nestřelil ani dva stejné. Pokud jsou v seznamu dva obrázky, hráč střelil dva stejné a jeden jiný obrázek. Pokud je v seznamu jen jeden obrázek, znamená to, že hráč střelil 3 stejné obrázky.

Procedura vyhodnoceni může vypadat na první pohled komplikovaně, ale když si odmyslíte dvě vrstvy, vnoření samotné procedury a vnoření lokální proměnné, k "porovnání" zbude jen for a case cyklus.





Tipy na modifikace programu:

více než 6 obrázků, které se střídají,  
více než 3 sprites pro zobrazování obrázků,  
náhodné zastavení obrázku a ponechání beze změny,  
počítání počtu pokusů, případně omezení na x pokusů, pak reset.

## 11 Aktivita 8 - aplikace s více obrazovkami a správci rozložení

Tematický celek	Tvorbena aplikace s více obrazovkami a kombinování správců rozložení.	
Motivační rámec	Získání větší kontroly nad vzhledem a organizací aplikace.	
Počet žáků	5-15	
Věk žáků	10+	
Pomůcky	PC odpovídajících parametrů, webový prohlížeč Mozilla Firefox nebo Google Chrome, zařízení s Androidem připojené k internetu, případně emulátor zařízení s Androidem, účet pro Google služby.	
Stručný popis aktivity	Přihlášení k prostředí App Inventoru <a href="http://appinventor.mit.edu/">http://appinventor.mit.edu/</a> Prohloubení znalostí o chování a využití správců rozložení "layout managerů". Teoretické a praktické seznámení s konceptem více obrazovek "screens" v rámci jedné aplikace.	
Vhodné místo	Počítačová učebna	
Cíle aktivity	Studenti si vyzkouší kombinaci layout managerů. Studenti se seznámí s konceptem obrazovek a vyzkouší si vyrobit vlastní aplikaci s více obrazovkami.	
Předchozí znalosti	Aktivita navazuje na předchozí lekci tohoto kurzu.	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
5-25m	Seznámení s tématem lekce. Volitelně předvedení a popis ukázkové aplikace. Popis jednotlivých správců rozložení, jejich kombinování a ovlivnění jejich chování pomocí parametrů, zarovnání, výška a šířka.  Seznámení s konceptem obrazovek. Seznámení s omezeními při sdílení kódu a proměnných mezi obrazovkami.	Frontální výklad, ukázka.
25-50m	Prohlídka ukázkových aplikací. Modifikace ukázkových aplikací, případně využití znalosti seznamů při tvorbě vlastní aplikace.	Samostatná a skupinová práce s využitím získaných znalostí, informací v tomto kurzu a externích zdrojů.
15-30m	Prezentace prací žáků, sdílení poznatků, diskuze. Motivace žáků k domácí práci.	Prezentace prací a diskuze
Hodnocení	Zpětná vazba je okamžitá. Student reflektuje vzdělávací cíle, svůj postup a dosažené výsledky. Student průběžně srovnává vlastní výsledky s výsledky ostatních studentů.	
Návaznosti	Na tuto aktivitu navazují následující aktivity, které prohlubují a kombinují znalosti prostředí App Inventoru.	
Poznámky	Pro případ výpadku připojení k internetu nebo technických problémů s App Inventorem doporučuji připravit záložní aktivitu, založenou na podobných principech blokového programování, například offline aplikaci MIT Scratch, BYOB nebo některou z variant programovacích prostředí LOGO.	

Informace ke správcům rozložení. Informace k blokům kódu řídících screen. (viz. on-line kurz)

Video-tutoriál k tvorbě více obrazovek v jedné aplikaci. (viz. on-line kurz)

Video-tutoriál k tvorbě více "virtuálních" obrazovek v jedné aplikaci pomocí skrývání bloků správce rozložení. Creating Multiple Screens with App Inventor (viz. on-line kurz)

### Informace ke správcům rozložení a obrazovkám

Správci rozložení "layout managers" pomáhají uspořádat objekty grafického prostředí "widgety". AI2 nabízí 3 správce rozložení, vertikálního, horizontálního a tabulkového. Tito správci mohou být zanořeni do sebe. Důležité pro určení chování widgetů jsou i další parametry, jako je určení šířky / výšky, případně vyplnění celého prostoru nadřazené komponenty, vertikální a horizontální zarovnání.

Obrazovky "Screens" jsou významný koncept práce s aplikací, v prostředí Androidu velmi rozšířený. Na screen lze v AI2 pohlížet jako na samostatnou aplikaci. V editoru má svůj vlastní design a má svůj vlastní kód. S jinými obrazovkami ve stejné aplikaci nesdílí ani proměnné. Na práci s obrazovkami je třeba si v AI2 trochu zvyknout. Při práci s návrhářem se vám v zařízení nebo emulátoru přepínají obrazovky podle toho, který máte v AI2 vybraný. Ne vždy je reakce na přepnutí rychlá, nežádka k ní vůbec nedojde. Bohudík výrazně lepší odezva a funkčnost je při použití vyexportované .apk.

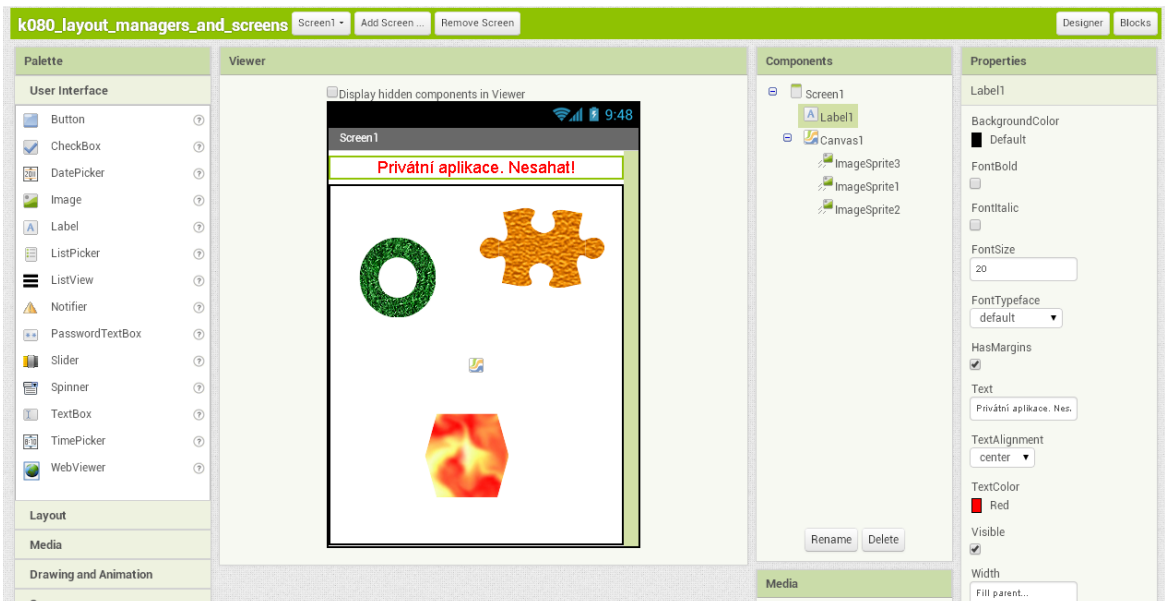
Názvy obrazovek nemohou být přejmenovány, proto se vyplatí vhodnému názvu věnovat patřičnou pozornost. Screen1 nemůže být smazán.

## Aplikace pro ukázkou práce s vnořenými správci rozložení a s více obrazovkami

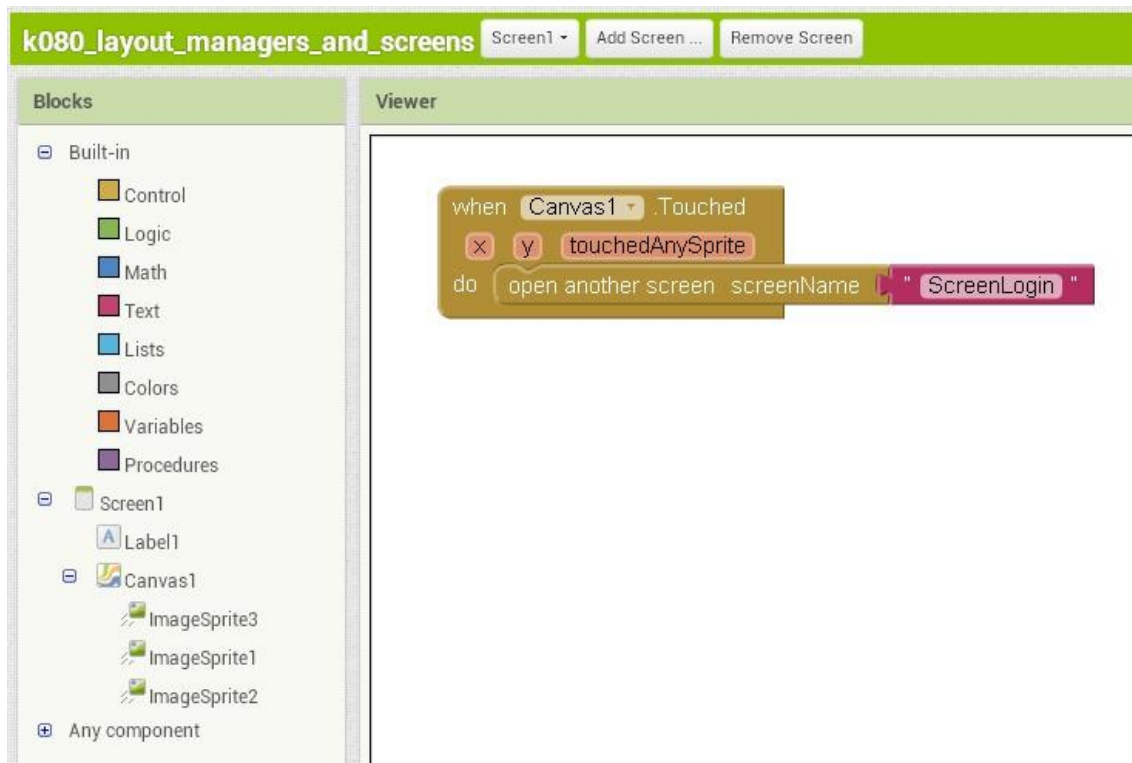
Práce se správci rozložení a s obrazovkami je ukázána na aplikaci, která zabezpečuje přístup k "tajnému" obsahu. Aplikace generuje klávesnici s náhodným zamícháním kláves, aby nebylo snadné vysledovat heslo. Aplikace umožňuje zadat i heslo pro případ nebezpečí. Heslo je přijato, ale místo tajného obsahu je zobrazen falešný obsah.

### Úvodní obrazovka modelové aplikace

Tato obrazovka po dotyku plochy zavolá obrazovku s přihlašovacím dialogem.

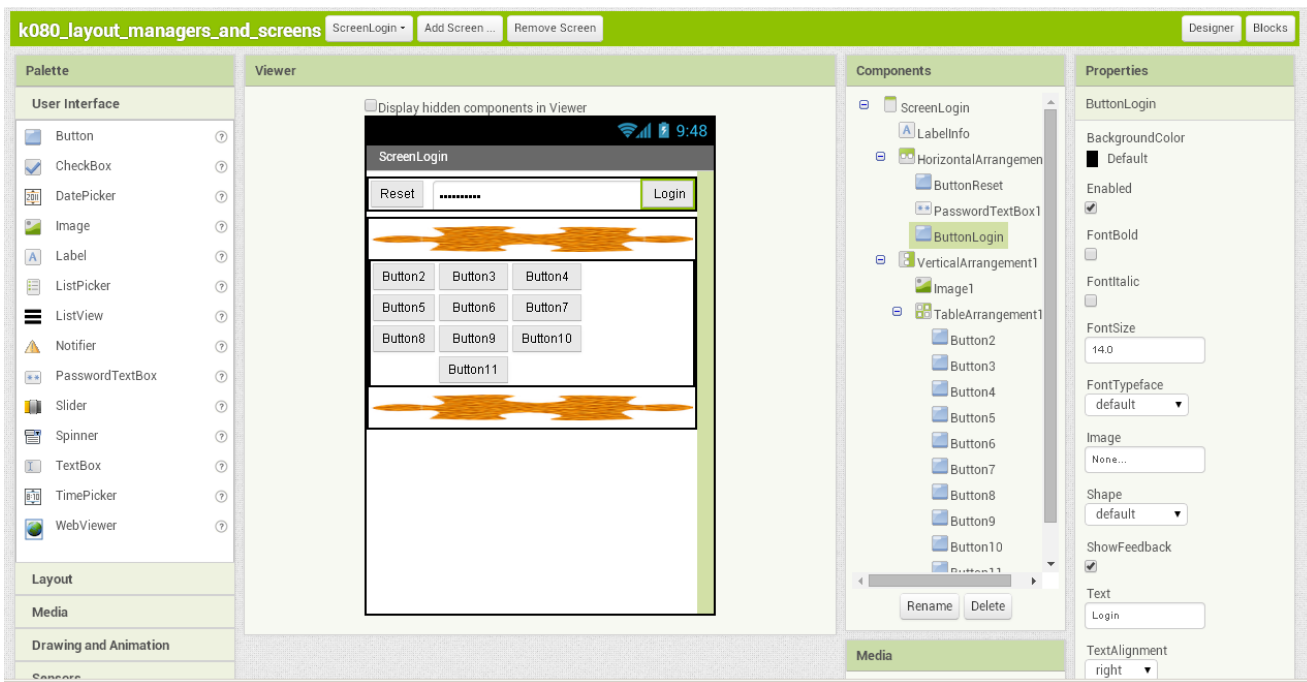


Kód obsahuje jediný blok, který zavolá obrazovku "ScreenLogin".



## Obrazovka pro přihlášení uživatele

Je zde ukázáno použití vnoření vertikálního a tabulkového správce rozložení. Poprvé je použito také políčko pro zadání hesla. Chová se stejně jako prosté textové pole, ale místo znaků zobrazuje tečky.

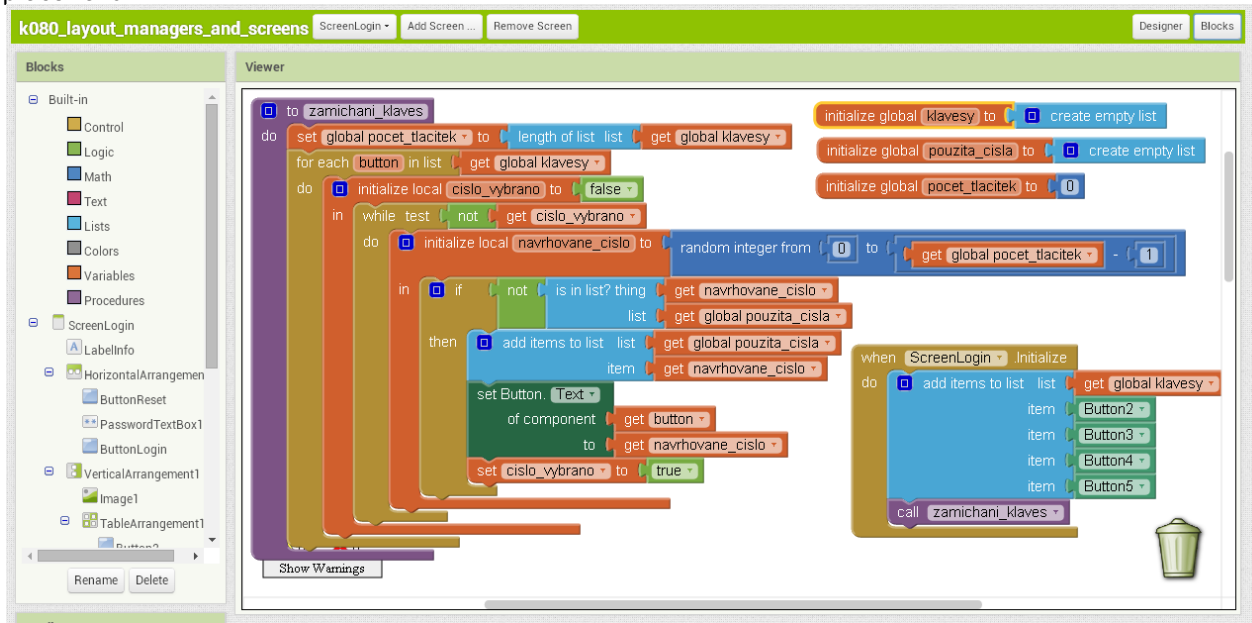


## Kód obrazovky pro přihlášení

Tomuto kódu dominuje procedura pro zamíchání číslic na klávesách. Postup je následující: Nejdříve jsou vytvořeny 2 seznamy. Seznam klavesy nelze naplnit ihned a je třeba ho plnit až v bloku inicializace obrazovky. Seznam je naplněn instancemi tlačítek a pak je zavoláno zamíchání.

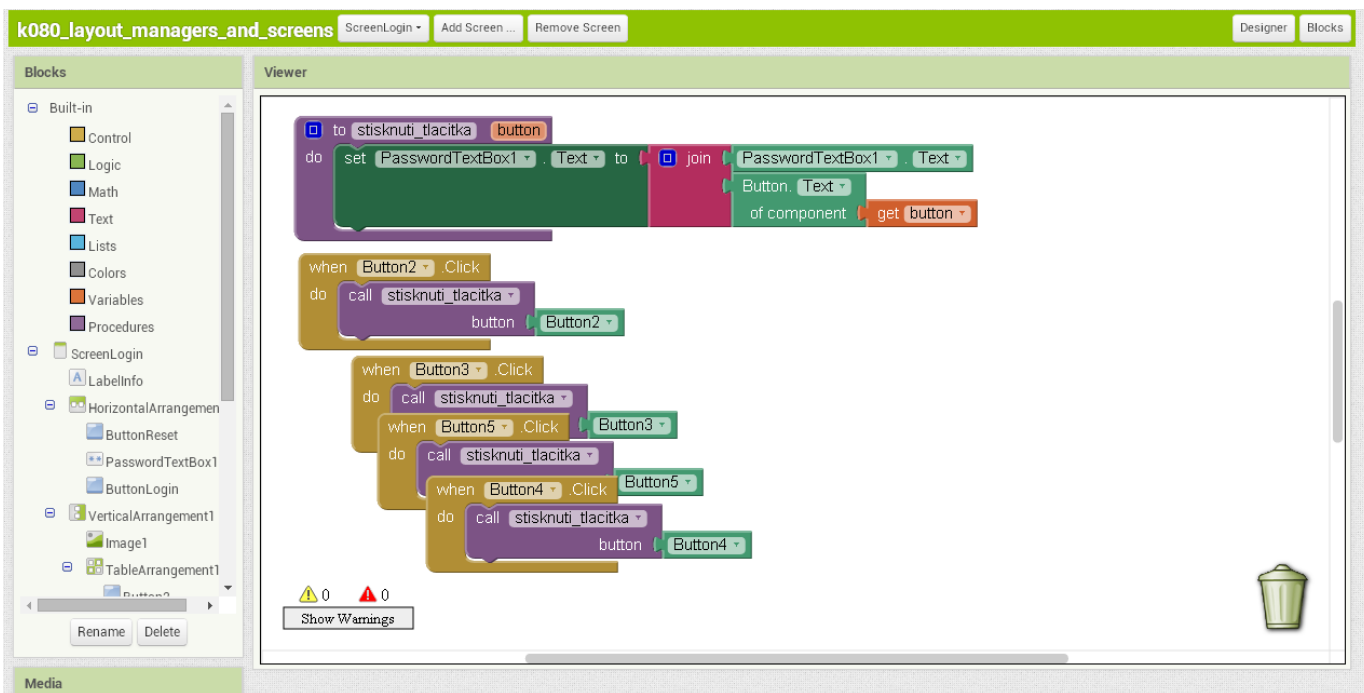
Pro každé tlačítko / button je volán cyklus, který vybere náhodně jednu číslici, kterou tlačítku přiřadí. Přiřazenou číslici dále nechceme přiřazovat dalšímu tlačítku, a tak je vytvořen seznam pouzita\_cisla, do kterého se použitá čísla ukládají. Cyklus while test se opakuje tak dlouho, dokud není pomocí random vybráno takové číslo, které ještě žádné tlačítko nemá.

V programu se náhodná čísla generují jen pro 4 tlačítka. Obsluha dalších tlačítek může být předmětem samostatné práce žáků



## Zápis vybraných znaků do pole password

Uživatel může zapisovat heslo rovnou do pole password, ale pokud píše na zabezpečené klávesnici, pak je třeba obloužit předávání stisknutých kláves. Pro všechna tlačítka jde o stejnou akci, kterou obsluhuje procedura stisknuti\_tlacitka. Pokud je v AI2 mnoho stejných bloků, doporučuji si je poskládat přes sebe a ušetřit místo.

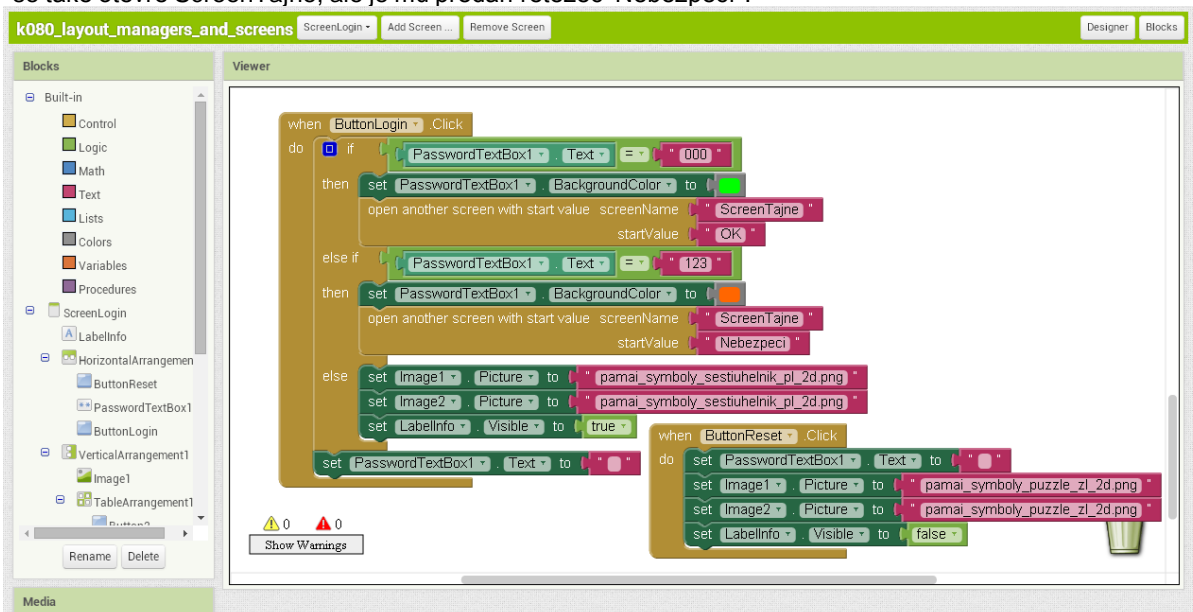


## Ověření hesla

V poslední ukázce kódu se řeší ověření hesla a reset hodnot.

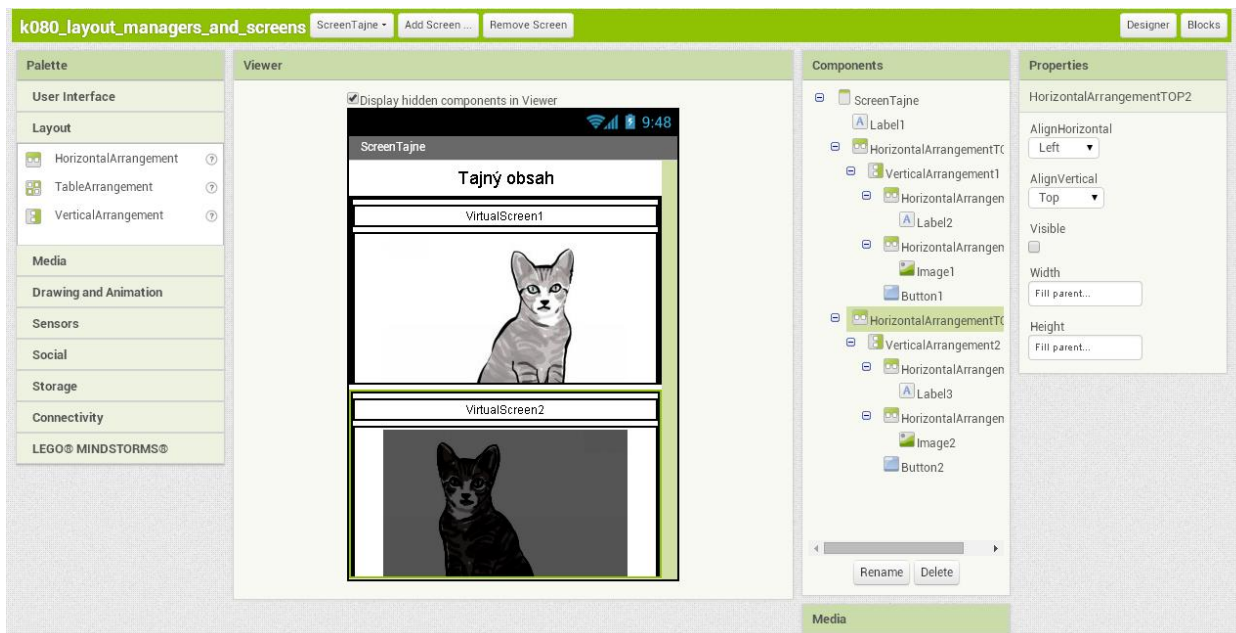
Předem je třeba přiznat, že zabezpečení hesla je u této aplikace velmi slabé. Takto zadané heslo by případný útočník, který by se dostal k .apk naší aplikace, velmi snadno našel ve zdrojovém kódu. V praxi se heslo neporovnává přímo s konkrétním řetězcem, ale bývá uložena jeho zašifrovaná podoba.

V této ukázce nám však jde o managery rozložení a o práci s obrazovkami. Při zadání správného hesla "000" (můžete samozřejmě libovolně změnit) se otevře obrazovka s tajným obsahem. Při zadání bezpečnostního hesla se také otevře ScreenTajne, ale je mu předán řetězec "Nebezpečí".



## Obrazovka s tajným obsahem

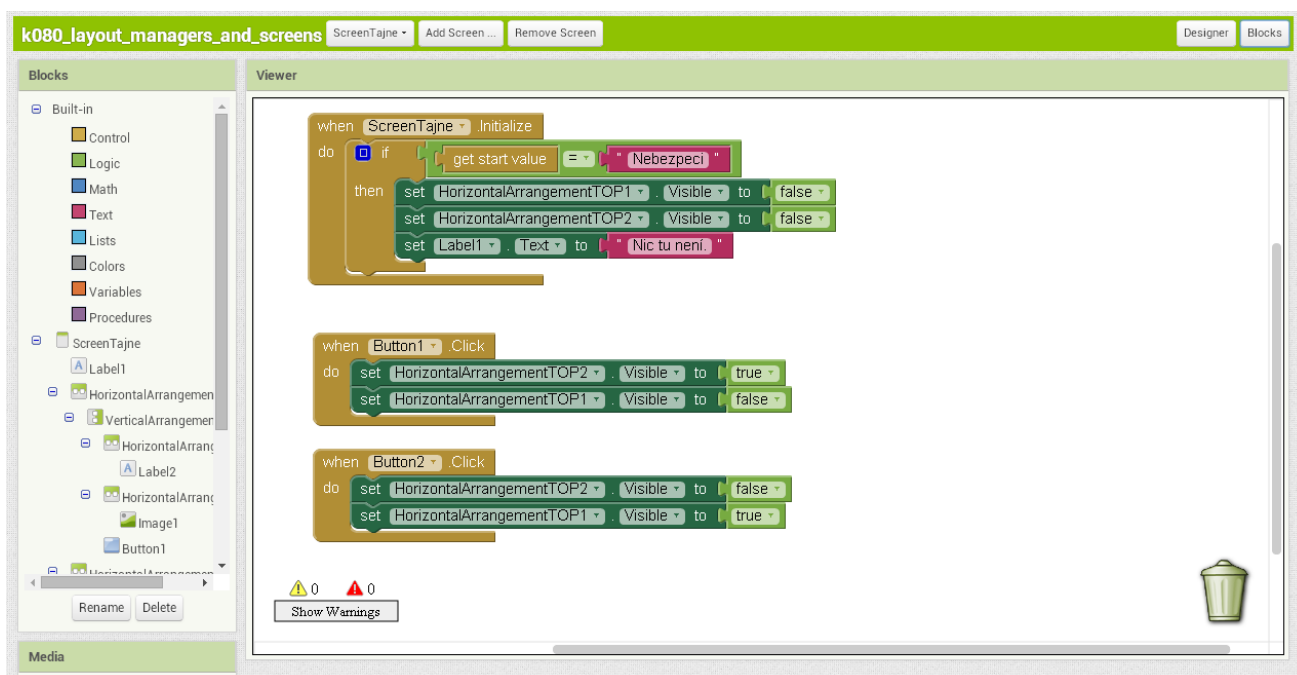
Obrazovka s tajným obsahem využívá ještě více správce rozložení. Používá je i k akci tzv. "virtuálních obrazovek", kdy pouze jeden vrcholový layout manager je zobrazen, zatímco ostatní jsou skryti. Tím lze vytvořit iluzi více obrazovek. Na obrázku si všimněte, že "Display hidden component in Viewer" nad ukázkou obrazovky je zapnutý a v properties je pro HorizontalArrangementTOP2.visible vypnuto.



## Kód obrazovky s tajným obsahem

V bloku when ScreenTajne.initialize se vyhodnocuje, zda byla obrazovka otevřena bezpečnostním heslem. Pokud ano a obdržela žetězec "Nebezpečí", pak zajistí skrytí tajného obsahu a přidá falešný obsah "Nic tu není."

Následující dva bloky se starají o přepínání mezi virtuálními obrazovkami.



Tipy na modifikace programu:

doplnění funkce bezpečnostní klávesnice pro další tlačítka,  
vyladění vzhledu bezpečnostní klávesnice, např. zarovnání na střed obrazovky, velikost a  
rozmístění tlačítek, přidání vlastní obrazovky, na kterou by byl uživatel odkázán při zadání jiného  
hesla,  
drobné zabezpeční hesla, aby nebylo v programu jako viditelný text "plain text".

## 12 Aktivita 9 - ukládání dat na paměťové kartě "storage"

Tematický celek	Použití komponent pro ukládání dat na kartě / disku, a jejich načtení.	
Motivační rámec	Seznámení s důležitým aspektem programování, ukládání a načítání dat do non-volatilní paměti.	
Počet žáků	5-15	
Věk žáků	10+	
Pomůcky	PC odpovídajících parametrů, webový prohlížeč Mozilla Firefox nebo Google Chrome, zařízení s Androidem připojené k internetu, případně emulátor zařízení s Androidem, účet pro Google služby.	
Stručný popis aktivity	Přihlášení k prostředí App Inventoru <a href="http://appinventor.mit.edu/">http://appinventor.mit.edu/</a> Praktické vyzkoušení ukládání dat a jejich načtení v prostředí App Inventor.	
Vhodné místo	Počítačová učebna	
Cíle aktivity	Studenti se seznámí a vyzkouší si použití cyklů programu.	
Předchozí znalosti	Aktivita navazuje na předchozí lekci tohoto kurzu.	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
5-25m	Seznámení s tématem lekce. Volitelně teorie o ukládání dat v PC (složky, soubory, databáze). Volitelně ukázka modelové aplikace a popis jednotlivých bloků, týkajících se práce s daty.  Volitelně samostatná činnost žáků, vyhledávání informací k tomuto	Frontální výklad, ukázka, samostatná práce žáků
30-60m	Analýza, úpravy a modifikace modelové aplikace, volitelně zakomponování funkcí pro ukládání a načítání dat do vlastní aplikace.	Samostatná a skupinová práce s využitím získaných znalostí, informací v tomto kurzu a externích zdrojů.
15-20m	Prezentace prací žáků, sdílení poznatků, diskuze. Motivace žáků k domácí práci.	Prezentace prací a diskuze
Hodnocení	Zpětná vazba je okamžitá. Student reflektuje vzdělávací cíle, svůj postup a dosažené výsledky. Student průběžně srovnává vlastní výsledky s výsledky ostatních studentů.	
Návaznosti	Na tuto aktivitu navazují následující aktivity, které prohlubují a kombinují znalosti prostředí App Inventoru.	
Poznámky	Pro případ výpadku připojení k internetu nebo technických problémů s App Inventorem doporučuji připravit záložní aktivitu, založenou na podobných principech blokového programování, například offline aplikaci MIT Scratch, BYOB nebo některou z variant programovacích prostředí LOGO.	

Wikipedie k tématu ukládání dat. (viz. on-line kurz)

Doplňkový tutoriál s aplikací malování. Práce s canvas, sprites, více obrazovek, vlastní procedury, ukládání dat.

(viz. on-line kurz)

Popis komponentstorage. (viz. on-line kurz)

### Aplikace využívající ukládání a načítání dat z paměťové karty

AI2 používá k ukládání a načítání dat několik komponent. V ukázkové aplikaci budeme používat dvě komponenty. TinyDB a File. TinyDB je databáze vázaná ke konkrétní aplikaci. Protože všechny aplikace spouštěné z AI2 pomocí AI Companion jsou z pohledu systému jedna aplikace, je třeba pro ukládání do TinyDB dostatečně unikátní slova, aby si aplikace navzájem nepřepisovaly data. Storage file umožňuje ukládat data do souborů přímo do souborového systému.

Rozšíříme funkce aplikace z aktivity 6 o ukládání skóre hráčů a uložení snímku aplikace. Většina kódu aplikace zůstala stejná, bloky, které byly změněny nebo doplněny, jsou v této ukázce.

V minulé aktivitě bylo zmíněno, že jednotlivé screeny nesdílí proměnné. Sdílení proměnných mezi screeny se dá vyřešit právě pomocí TinyDB.



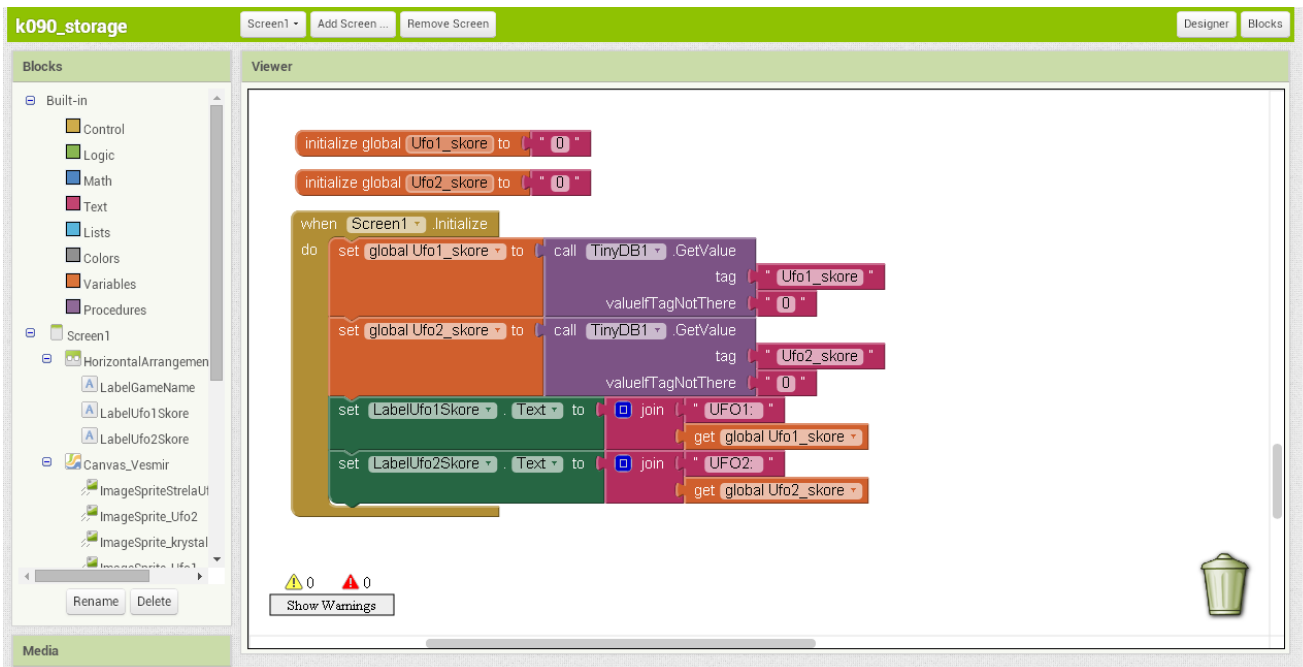
## Komponenty modelové aplikace

U komponent se mnoho nezměnilo. Tlačítko Restart bylo přesunuto vlevo dolů a přibyla tlačítka Pause a Save. Významnou změnou jsou však komponenty TinyDB1 a File1.



## Inicializace a načtení dat z databáze

Protože nelze proměnné inicializovat přímo hodnotami z databáze, je konstrukce následující. Nejdříve jsou vytvořeny globální proměnné. Následně, v bloku when Screen1.Initialize jsou načteny a předány hodnoty z databáze. Pokud by v databázi požadovaná data nebyla, použije se výchozí hodnota z valueIfTagNotThere. Na závěr inicializace jsou hodnoty zapsány do LabelUfo1Skore a LabelUfo2Skore.

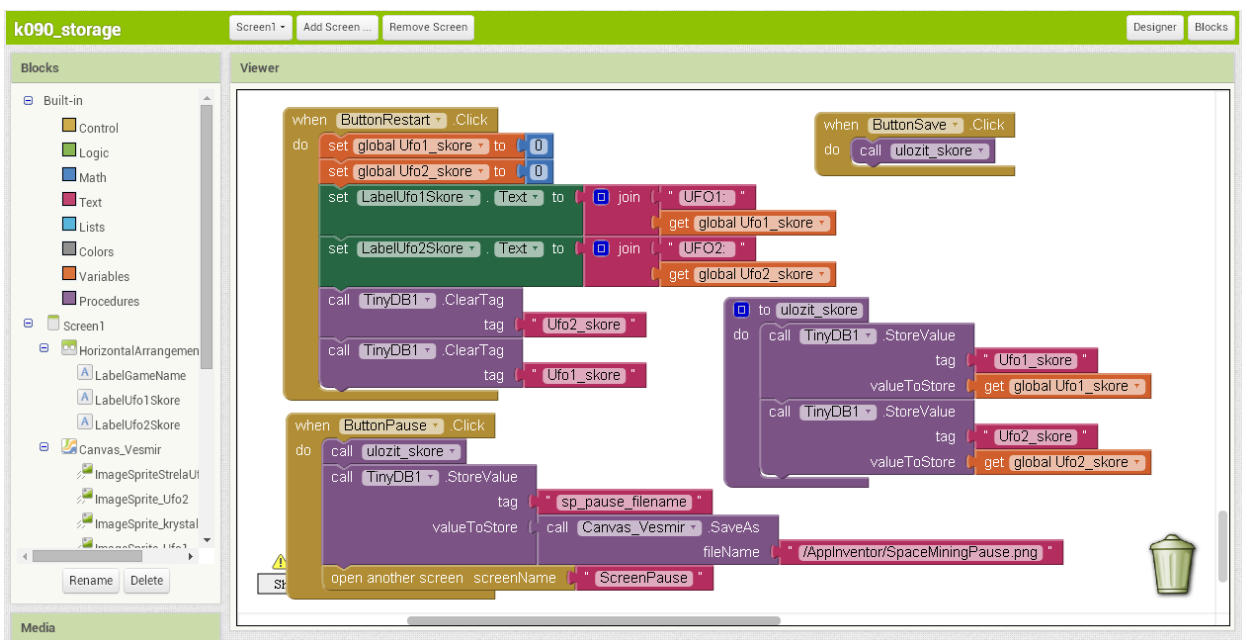


## Bloky využívající ukládání dat a mazání dat

Blok when ButtonRestart.click řeší vynulování obsahu globálních proměnných, zobrazení dat v LabelUfo1Skore a LabelUfo2Skore a stará se i o smazání dat v databázi.

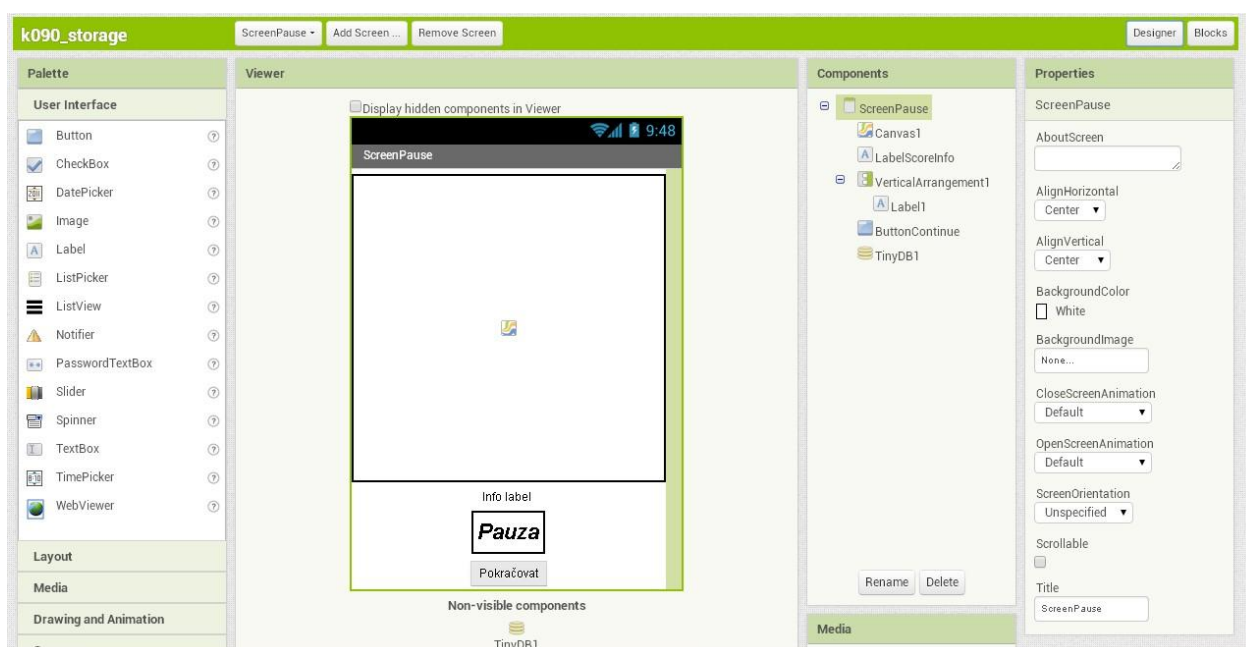
When ButtonSave.Click volá proceduru ulozit\_skore, která zapíše hodnoty globálních proměnných do databáze. V databázi jsou hodnoty uloženy pod názvem (tagem) Ufo1\_skore a Ufo2\_skore.

V obsluze tlačítka Pause se kromě uložení skóre řeší vytvoření snímku obrazovky, jeho uložení na disk do souboru /AppInventor/SpaceMining.png uložení cesty a názvu do databáze a otevření obrazovky ScreenPause. Cesta



## Design obrazovky ScreenPause

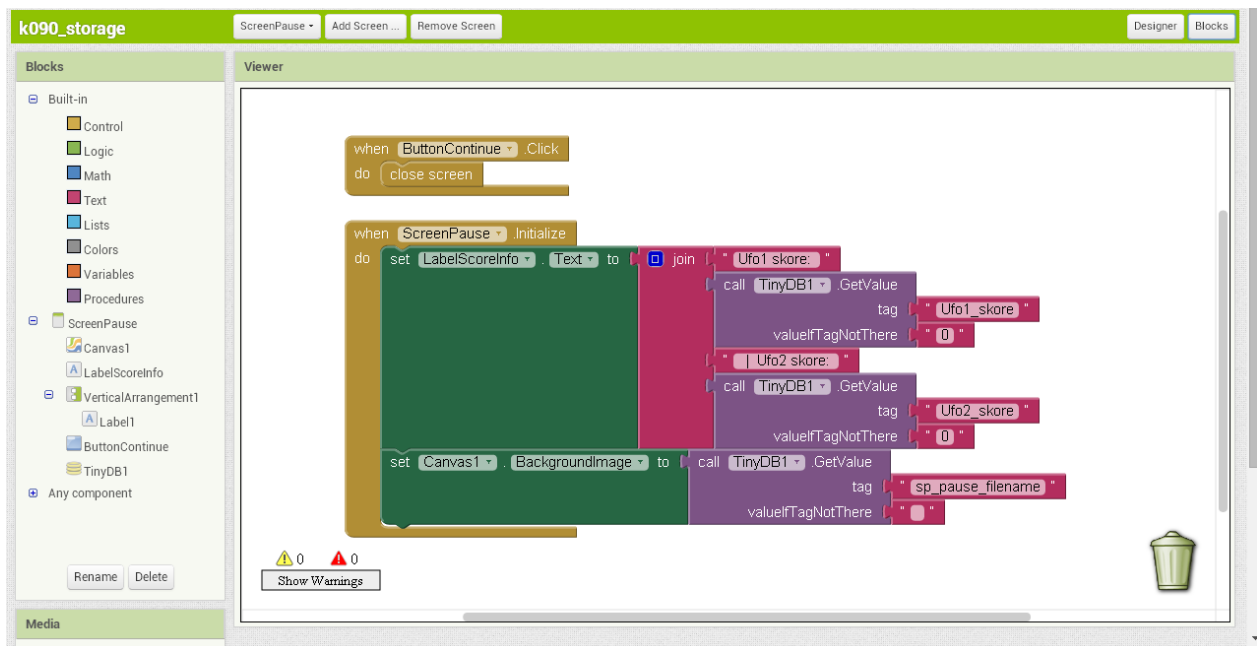
Většinu plochy zabírá komponenta Canvas1, ve které se budou zobrazovat screenshoty hry, před jejím pozastavením. Za povšimnutí stojí opětovné zařazení komponenty TinyDB. Pro každý screen, ve kterém se bude pracovat s uloženými daty v databázi, je třeba znovu přidat komponentu TinyDB.



## Bloky obrazovky ScreenPause

Obsluha tlačítka Pokračovat nepoužívá otevření obrazovky Screen1, ale zavírá sama sebe. Kdyby tomu tak nebylo, vznikl by cyklus, který by znesnadňoval použití systémového tlačítka Zpět.

Blok when ScreenPause.Initialize řeší zobrazení skóre hráčů, načtení a zobrazení screenshotu z předchozí obrazovky.



Tipy na modifikace programu:

záznam více údajů, (např. životů, počet srážek, souřadnice lodi a její směr), uložení, načtení a zobrazení těchto údajů, ukládání snímků pomocí dalšího tlačítka, na obrazovce ScreenPause prohlížení uložených screenshotů, přihlašování ke hře, zapamatování jmen hráčů a jejich skóre, zobrazování jmen hráčů, místo názvů UFO1 UFO2, výběr lodí, jejichž parametry (obrázek, rychlost pohybu, ...) by byly načítány z databáze.

### **13 Aktivita 10 - další prvky pro interakci s uživatelem a vlastní webový prohlížeč**

Tematický celek	Seznámení s dalšími prvky pro interakci s uživatelem a tvorba vlastního webového prohlížeče.	
Motivační rámec	Rozšíření znalostí o funkci a použití užitečných prvků grafického prostředí a tvorba vlastního webového prohlížeče.	
Počet žáků	5-15	
Věk žáků	10+	
Pomůcky	PC odpovídajících parametrů, webový prohlížeč Mozilla Firefox nebo Google Chrome, zařízení s Androidem připojené k Internetu, případně emulátor zařízení s Androidem, účet pro Google služby.	
Stručný popis aktivity	Přihlášení k prostředí App Inventoru <a href="http://appinventor.mit.edu/">http://appinventor.mit.edu/</a> Seznámení s dalšími prvky uživatelského prostředí. Obohacení vlastní aplikace o některé z prvků GUI. Tvorba vlastního webového prohlížeče.	
Vhodné místo	Počítačová učebna	
Cíle aktivity	Studenti se seznámí a vyzkouší si použití nových prvků grafického uživatelského prostředí. Studenti si vyzkouší vytvořit vlastní webový prohlížeč.	
Předchozí znalosti	Aktivita navazuje na předchozí lekci tohoto kurzu.	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
5-25m	Seznámení s tématem lekce. Volitelně teorie o ukládání dat v PC (složky, soubory, databáze). Volitelně ukázka modelové aplikace a popis jednotlivých bloků, týkajících se práce s daty.  Volitelně samostatná činnost žáků, vyhledávání informací k tomuto tématu.	Frontální výklad, ukázka, samostatná práce žáků
30-60m	Analýza, úpravy a modifikace modelové aplikace, volitelně zakomponování funkcí pro ukládání a načítání dat do vlastní aplikace.	Samostatná a skupinová práce s využitím získaných znalostí, informací v tomto kurzu a externích zdrojů.
15-20m	Prezentace prací žáků, sdílení poznatků, diskuze. Motivace žáků k domácí práci.	Prezentace prací a diskuze
Hodnocení	Zpětná vazba je okamžitá. Student reflektuje vzdělávací cíle, svůj postupu a dosažené výsledky. Student průběžně srovnává vlastní výsledky s výsledky ostatních studentů.	
Návaznosti	Na tuto aktivitu navazují následující aktivity, které prohlubují a kombinují znalosti prostředí App Inventoru.	
Poznámky	Pro případ výpadku připojení k internetu nebo technických problémů s App Inventorem doporučuji připravit záložní aktivitu, založenou na podobných principech blokového programování, například offline aplikaci MIT Scratch, BYOB nebo některou z variant programovacích prostředí LOGO.	

Wikipedie k tématu grafického uživatelského prostředí GUI. Popis komponent uživatelských ovládacích prvků

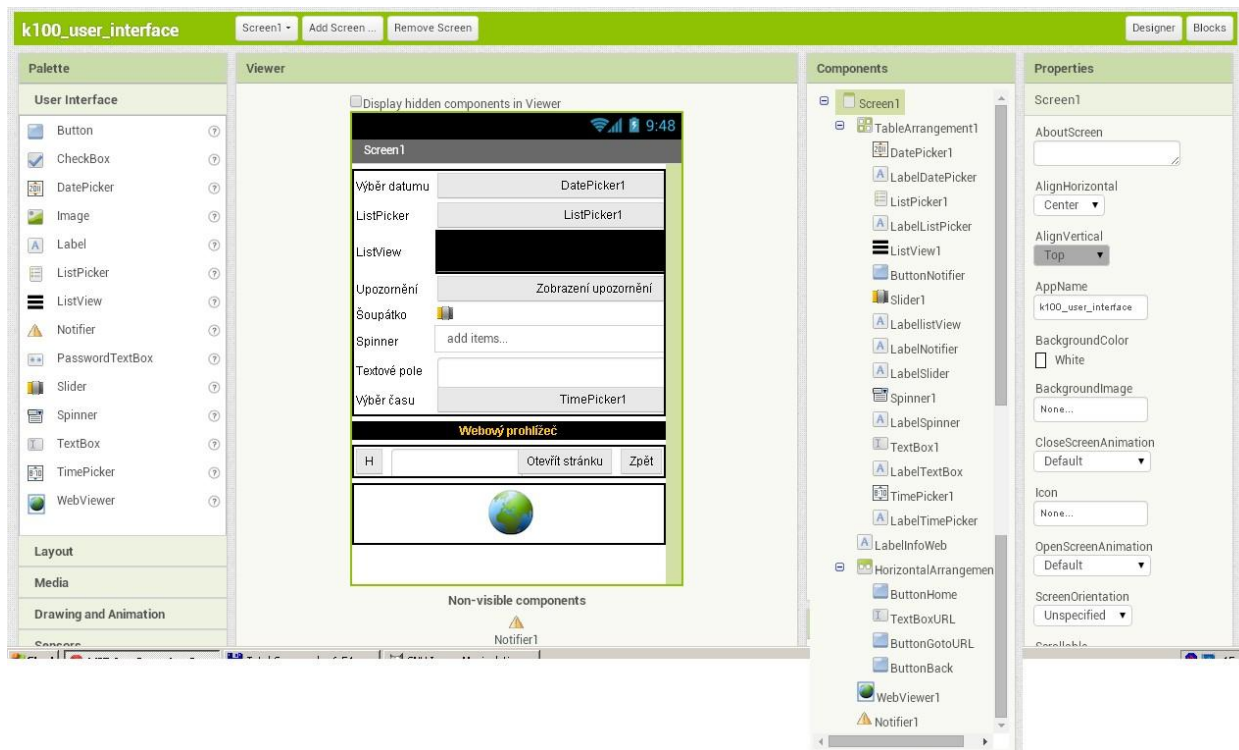
User Interface. (viz. on-line kurz)

Video-návod, jak pracovat s komponentou How to work with listpicker. (viz. on-line kurz)

Video-návod, jak vytvořit vlastní webový prohlížeč. Ai2 app Inventor tutorial WebView. (viz. on-line kurz)

## Ukázková aplikace prvků z nabídky User Interface a WebView

V této aplikce je ukázka všech prvků z User Interface, které jsme zatím v nepoužívali. Pro každý prvek je doplněn Label, který reaguje na uživatelskou akci. V dolní části je komponenta WebView, do které budeme zobrazovat obsah webových stránek. Screen1 má povolenu vlastnost Scrollable.

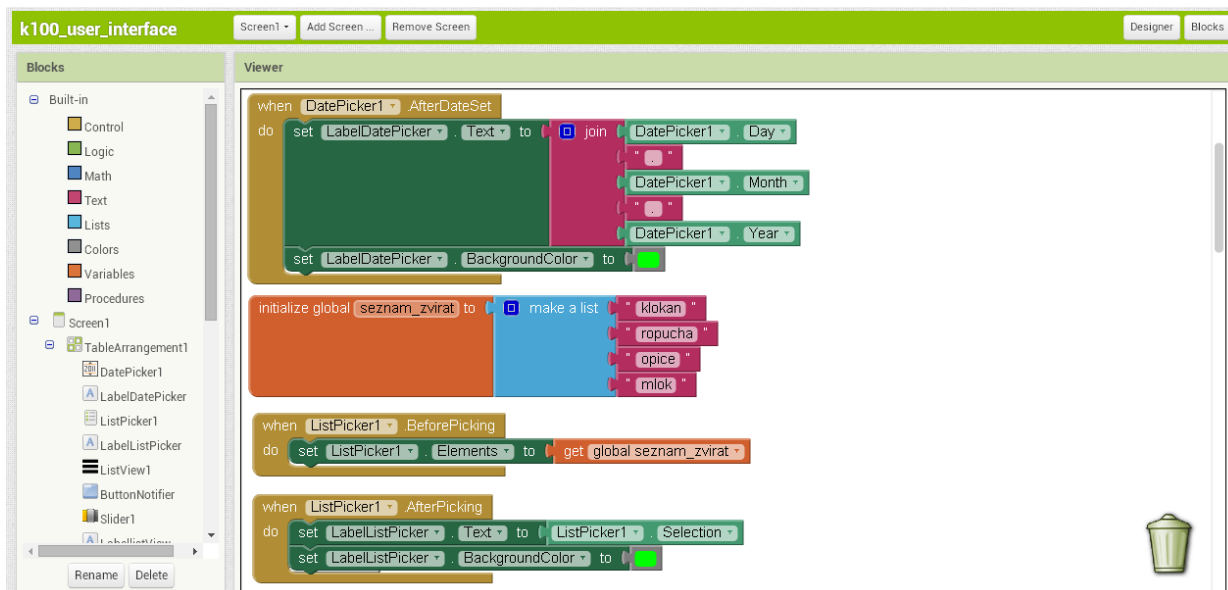


## Bloky ukázkové aplikace

Výběr data (den, měsíc, rok) reaguje na událost dokončení výběru. V tu chvíli se složí text, který je vložen do popisku LabeldataPicker. Popiskům, které byly použity pro výstup, je nastaveno zelené pozadí.

Následuje vytvoření testovacího seznamu.

U výběrového seznamu ListPicker1 obsluhujeme dvě události. BeforePicking, ve kterém komponentu propojíme se seznamem. A AfterPicking, ve které předáme vybranou položku do textu popisku.



## Bloky ukázkové aplikace

Některé komponenty potřebují inicializaci. Tu můžeme provést například v bloku when Screen1.Initialize. V našem případě inicializujeme komponenty ListView1 a Spinner1 naším testovacím seznamem. Z komponenty ListView1 pak při události AfterPicking předáváme hodnotu vybrané položky do popisku. Spinner1 je ukázán později.

S obsluhou upozornění "notifikace" se pracuje jiným způsobem. Po stisknutí tlačítka ButtonNotifier je zavolána procedura upozornění, které je předán řetězec jako obsah zprávy. Zobrazení upozornění je závislé na systému. Většinou jde o vrstvu, která se zobrazí nad plochou aplikace.

Šoupátko "slider" umožňuje výběr z předem definovaného rozsahu hodnot. Reaguje na změnu pozice a aktuální hodnotu předává do popisku.

The screenshot shows the 'k100\_user\_interface' application designer. The 'Blocks' panel on the left lists various UI components like LabelListView, LabelNotifier, LabelSlider, Spinner1, LabelSpinner, TextBox1, LabelTextBox, TimePicker1, LabelTimePicker, LabelInfoWeb, HorizontalArrangement, ButtonHome, TextBoxURL, ButtonGotoURL, ButtonBack, WebViewer1, and Notifier1. The 'Viewer' panel on the right displays the following code blocks:

```
when Screen1.Initialize
do
  set ListView1.Elements to get global seznam_zvirat
  set Spinner1.Elements to get global seznam_zvirat

when ListView1.AfterPicking
do
  set LabelListView.Text to ListView1.Selection
  set LabelListView.BackgroundColor to #00FF00

when ButtonNotifier.Click
do
  call Notifier1.ShowAlert
  notice "\n\n Upozornění na událost\n\n Co se stalo? \n Prase kozu potrkalo! \n\n "
  set LabelNotifier.BackgroundColor to #00FF00

when Slider1.PositionChanged
do
  thumbPosition
  set LabelSlider.Text to Slider1.ThumbPosition
  set LabelSlider.BackgroundColor to #00FF00
```

## Bloky ukázkové aplikace

Spinner1 je rotující výběrový seznam. Po výběru předá hodnotu vybrané položky popisku.

TextBox1 patří k základním nástrojům komunikace s uživatelem. V našem případě probíhá předání hodnoty při události LostFocus, tedy při ztrátě výběru. K této události dochází, pokud jiná komponenta získá Focus. V naší modelové aplikaci většina komponent Focus nepoužívá, a tak nemusí být snadné tuto akci vyvolat.

Výběr času je komponenta velmi podobná výběru data (den, měsíc, rok), i obsluha je podobná.

The screenshot shows a visual programming environment with a 'Blocks' panel on the left and a 'Viewer' panel on the right. The 'Blocks' panel lists various UI components like LabelListView, LabelNotifier, LabelSlider, Spinner1, LabelSpinner, TextBox1, LabelTextBox, TimePicker1, LabelTimePicker, LabelInfoWeb, HorizontalArrangement, ButtonHome, TextBoxURL, ButtonGotoURL, ButtonBack, WebViewer1, and Notifier1. The 'Viewer' panel displays three event-driven code blocks:

- when Spinner1 AfterSelecting**: A 'do' block containing:
  - set LabelSpinner . Text to get selection
  - set LabelSpinner . BackgroundColor to green
- when TextBox1 LostFocus**: A 'do' block containing:
  - set LabelTextBox . Text to TextBox1 . Text
  - set LabelTextBox . BackgroundColor to green
- when TimePicker1 AfterTimeSet**: A 'do' block containing:
  - set LabelTimePicker . Text to join TimePicker1 . Hour and TimePicker1 . Minute
  - set LabelTimePicker . BackgroundColor to green

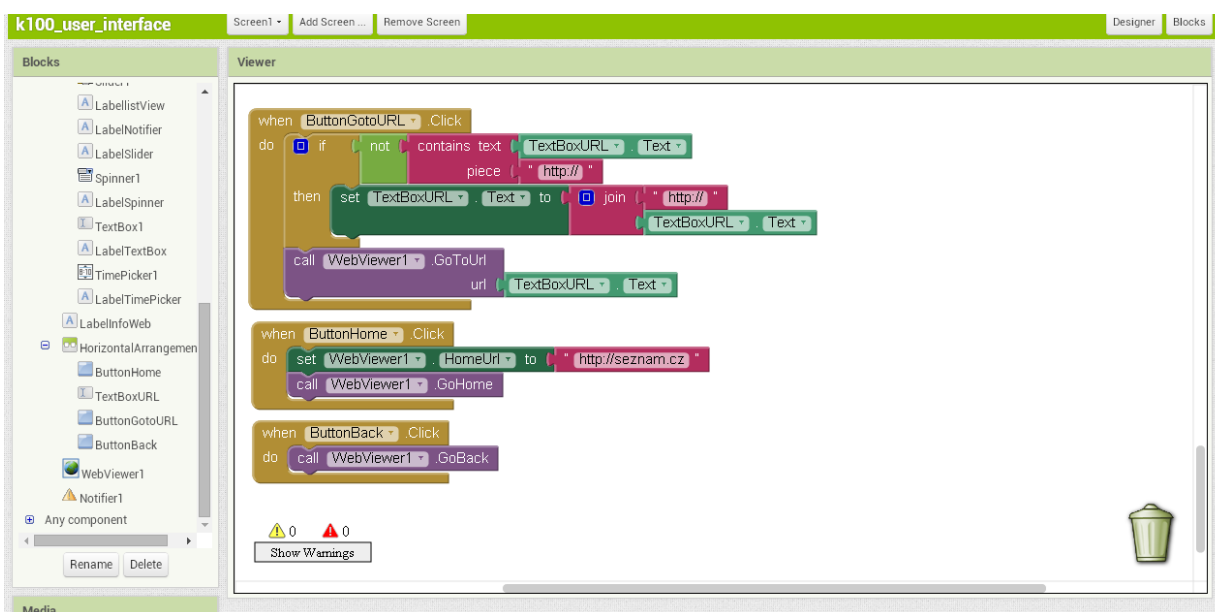
At the bottom of the 'Viewer' panel, there are warning indicators (0 yellow and 0 red triangles) and a 'Show Warnings' button. A trash icon is also visible in the bottom right corner of the viewer area.

## Bloky obsluhy komponenty WebView

K webovým stránkám neodmyslitelně patří prohlížeče webových stránek. Jak by bylo možné vyrobit vlastní prohlížeč, když na tak náročném úkolu se podílí tisíce programátorů? Velmi snadno, pokud využijeme připravené komponenty, která většinu funkcí obslouží ve své základní výbavě.

Pro výrobu vlastního prohlížeče potřebujeme pouze zaznamenat a předat URL adresu webové stránky, kterou má prohlížeč načíst a zobrazit. To řešíme pomocí textového pole TextBoxURL. Je zde doplněna kontrola, zda adresa stránky obsahuje řetězec http://, a pokud ne, doplní ho. Pak je adresa stránky předána proceduře komponenty WebView1, která se postará o načtení a zobrazení stránky.

Ukázkový prohlížeč je doplněn o tlačítko s přechodem na domovskou stránku. Podobným způsobem by bylo možné vyrobit tlačítka pro přímý přechod na jiné stránky. Protože by v případě použití systémového tlačítka Zpět byla vypnuta naše aplikace, nelze ho využít pro přechod zpět v prohlížeči. Proto je prohlížeč doplněn o tlačítko zpět, které lze použít k cestě zpět navštívenými stránkami.



### Tipy na další práci:

vytvořit si samostatnou aplikaci s prohlížečem stránek a vyřešit snadný přechod na X oblíbených stránek, rozšířit vlastní aplikaci o některé komponenty z User Interface, vyzkoušet pomocí slideru ovládat parametry některých komponent, např. zvětšování / zmenšování obrázku.



## 14 Závěrem

---

### Co dál?

V čase, který můžete věnovat další tvorbě, doporučuji věnovat čas vašim vlastním programům. Na programech můžete pracovat samostatně, lépe však ve skupinkách. Tvorba programů není jen o programování, je to také o návrhu aplikace, plánování a dalších činnostech, které patří spíše k softwarovému inženýrství.

Doufám, že vám programovací prostředí App Inventor a tento kurz pomohly nahlédnout pod pokličku programování pro prostředí Android. Pro některé z Vás to možná bude jediný a dostačující způsob, jak pro Android vytvářet programy, pro některé z vás to bude jen jeden z prvních kroků, díky kterým si ochočíte komplikovanější vývojová prostředí.

Děkuji za společnou procházku tímto kurzem.

Ať už se vaše kroky budou ubírat různými cestami, ať je vaše putování prospěšné a radostné.

Filip Vaculík

## Příloha

---

### Odkazy

Odkazy na MIT App Inventor

Hlavní stránka vývojového prostředí MIT App Inventor: <http://appinventor.mit.edu/>

Odkaz do vlastního vývojového prostředí: <http://ai2.appinventor.mit.edu/>

Popis charakteristik a použitých konceptů AI2: <http://appinventor.mit.edu/explore/ai2/concepts.html>

Odkaz na databázi ukázkových aplikací: <http://appinventor.mit.edu/explore/ai2/tutorials.html>

Sborník velkého množství ukázkových kódů snippets. <http://puravidaapps.com/snippets.php>

Informace o aktualizacích vývojového prostředí. <http://appinventor.mit.edu/ai2/ReleaseNotes.html>

Odkazy na informace v angličtině

Android na Anglické Wikipedii: [http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))

Hlavní stránka Androidu: <http://www.android.com/>

Hlavní databáze aplikací pro Android: <https://play.google.com/store>

Odkazy na informace v češtině

Android na České Wikipedii: [http://cs.wikipedia.org/wiki/Android\\_\(operační\\_systém\)](http://cs.wikipedia.org/wiki/Android_(operační_systém))

Hlavní české stránky Androidu: <http://www.svetandroida.cz/>

Český slovník výrazů a zkratk týkajících se Androidu: <http://slovník.androidforum.cz/>

České diskuzní fórum: <http://androidforum.cz/>

Česká odnož Play Store: <http://www.androidmarket.cz/>

Odkazy na pomocné nástroje

Emulátor mnoha typů zařízení pro Android: <http://www.genymotion.com/>

Hlavní vývojové prostředí pro Android: [http://en.wikipedia.org/wiki/Android\\_Studio](http://en.wikipedia.org/wiki/Android_Studio)

### Práva k použitým materiálům

Koláž, logo kurzu:

Logo MIT app inventor (CC)

Logo Android a nápis Android

Reklamní fotky mobilů

Vlastní screenshot kódu

Test, obrázek loga projektu Replicant

Test, výšeč grafu zveřejněného pod licencí (CC BY-SA 3.0).

Vlastní tvorba (vydáno s licencí CC [Attribution 3.0 Unported](https://creativecommons.org/licenses/by-sa/3.0/)):

Zdrojové kódy ke všem ukázkovým programům.

Zvuk Hello World v první lekci, záznam strojového čtení.

Obrázek Hello World v první lekci.

Screenshots obrazovek z prostředí App Inventoru.

Screenshots z prostředí emulátorů.

Screenshots z prostředí OS.

Kliparty k tématu vesmír a z nich vyexportované obrázky.

Kliparty skupiny symboly a z nich vyexportované obrázky.

Loga, značky firem, značky produktů jsou užívány pouze v přímé souvislosti s původními vlastníky a užití v tomto kontextu není omezeno zákonem § 8/1 zákona o ochranných známkách.

[Cat\\_illustration.jpg](#) Jan Gillbank, Creative Commons Attribution 3.0 Unported

Odkaz na licenční podmínky:

[CC BY-SA 3.0](https://creativecommons.org/licenses/by-sa/3.0/)

CC [Attribution 3.0 Unported](https://creativecommons.org/licenses/by-sa/3.0/)

(odkazy viz. on-line kurz)