

(2) Farbtopf-Spiel (Paint Pot)

Wir programmieren ein Farbtopf-Spiel, so dass man mit farbigen Punkten und Strichen auf ein Hintergrundbild zeichnen kann. Diese Aufgabe hat drei Teile. Aber bereits am Ende von jedem Teil hast du eine komplette App zum Spielen.

Wie werden einige neue Komponenten und Blöcke brauchen:



TIPPS:



COMPONENTS

Canvas:

Leinwand, um darauf zu zeichnen

Arrangements:

Anordnung, um mehrere Taste nebeneinander zu platzieren

Camera:

Aktiviert und verbindet die Kamera des Mobiltelefons. So kannst du z.B. ein Foto machen und als Hintergrundbild benutzen.

BLOCKS

Variables

Diese werden als Zwischenspeicher benutzt. Man kann einen Wert hineinlegen, solange die App läuft.

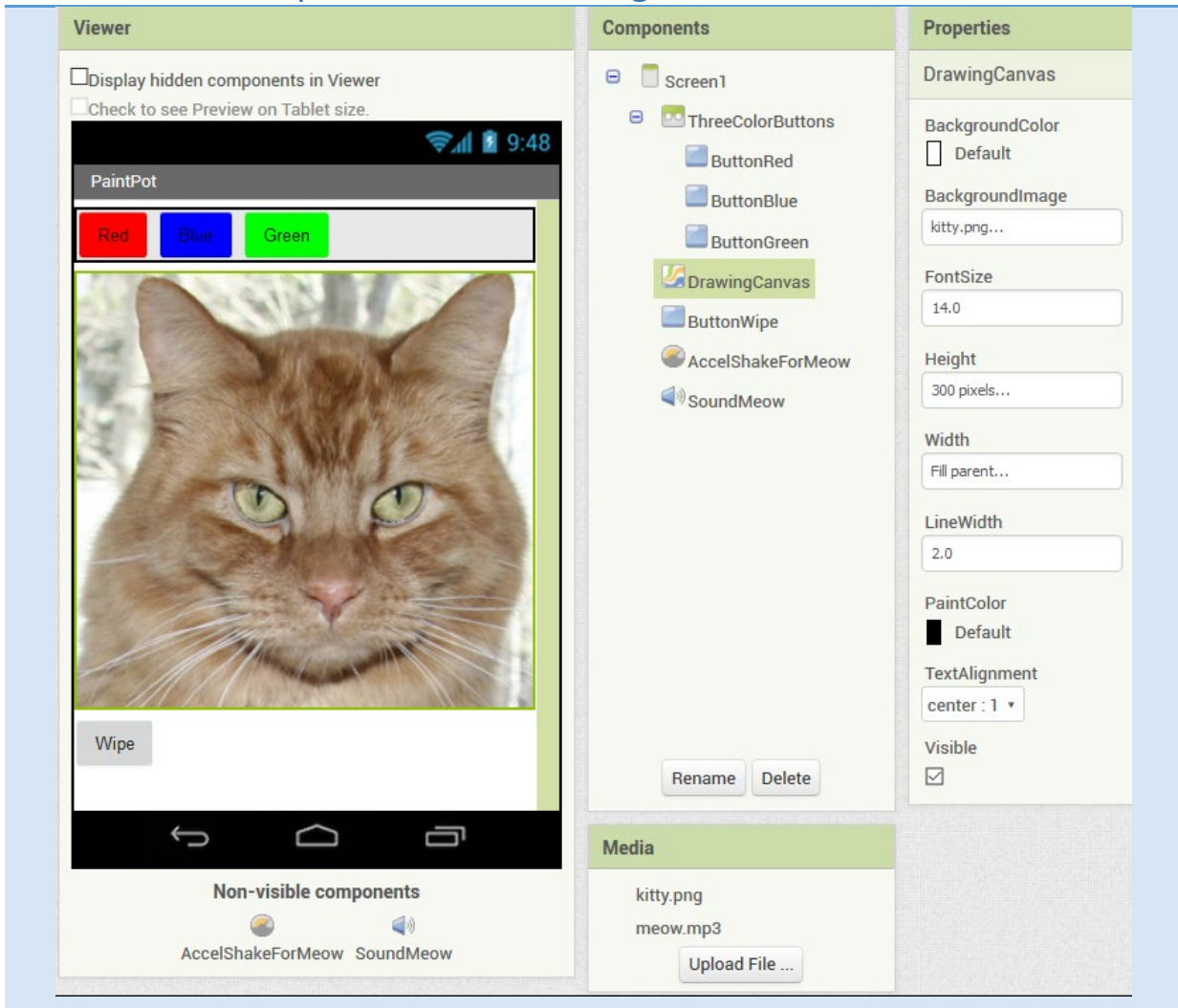
Farbtopf: Teil 1

Starte ein neues Projekt und nenne dieses FarbtopfTeil1. Füge die folgenden Komponenten im Viewer (Designer-Fenster) ein. Du musst jede Komponente umbenennen und gewisse Eigenschaften anpassen. Bemerkung: Du kannst auch neue Namen auf Deutsch wählen!

Komponenten	Palette (Kategorie)	Umbenennen in	Wofür plus Eigenschaften
Horizontal Arrangement	Layout	ThreeColorButtons	Anordnung, um die Tasten für die Farben nebeneinander zu platzieren
Button (x3)	User Interface	ButtonRed ButtonGreen ButtonBlue	Für die Auswahl der Farbe zum Zeichnen. Platziere die Tasten innerhalb der Anordnung <i>ThreeColorButtons</i> , und ändere die Text- und Hintergrundfarbe der Tasten entsprechend (ROT / GRÜN / BLAU)
Canvas	Drawing & Animation	DrawingCanvas	Leinwand um darauf zu zeichnen Eigenschaften: - Height : 300 Pixels - Width: Fill parent - BackgroundImage: kitty.png
Button (x1)	User Interface	ButtonWipe	Löscht alle Zeichnungen - Text: Wipe (löschen)
Accelerometer Sensor	Sensor	AccelShake4Meow	Misst ob das Telefon geschüttelt wird
Sound	Media	SoundMeow	Spielt ein Audio Datei - Source: Meow.mp3

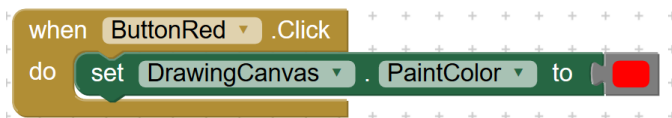
Überprüfe die Zusammenfassung des Designer-Teils. Hast du alles programmiert?

Übersicht Farbtopf Teil 1 : Designer-Fenster



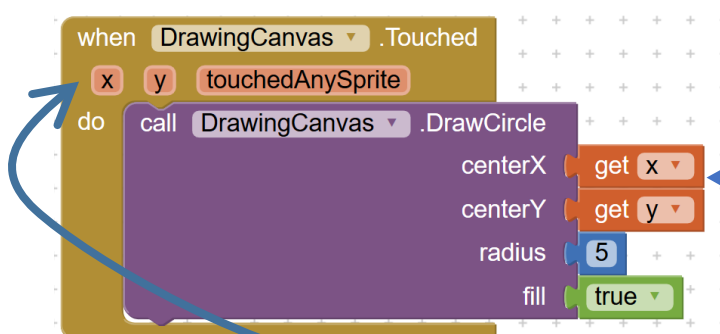
Jetzt müssen wir die entsprechenden Aktionen im Blocks-Fenster programmieren. Denk daran, deine App immer wieder zu testen.

1. Hole die nötigen Funktionen, wie unten gezeichnet, so dass man bei anklicken der Farben-Tasten eine neue Zeichnungsfarbe auswählen kann:



2. Wiederhole das für die blaue und grüne Taste.
Tipp: mit der rechte Maustaste und "duplicate" wird man viel schneller!

3. Füge jetzt die Funktion, welche Kreise zeichnet, wenn man die Leinwand berührt:



Bemerkung:

die Kreise werden durch Mittelpunkt (X, Y) und Grösse (Radius) beschrieben.

Das Teil "get x" holt man oben in «when...touched» Block selbst, in orangen Rechteck « x ».

4. Jetzt füge die Funktion ein, welche Striche zeichnet, wenn man einen Finger über die Leinwand zieht.

```

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

```

Bmk.:

die Striche wird zwischen die vorherige (x1,y1) und die aktuelle (x2,y2) Position gezeichnet.

5. Beim Anklicken von ButtonWipe sollen alle Zeichnungen gelöscht werden. Auf Englisch heisst das: clear. Suche und füge die passende Blöcke ein.
6. Beim Schütteln des Smartphones soll der Ton «Meow.mp3» abgespielt werden. Das kannst du gleich programmieren wie in der ersten App «Hallo Katze».

Hast du nun alle Aktionen programmiert und getestet? Überprüfe die Zusammenfassung des Block-Teils auf der nächsten Seite und korrigiere eventuelle Fehler.

Übersicht Paint-Pot Teil 1 : Block-Fenster

```

when ButtonRed .Click
do set DrawingCanvas .PaintColor to red

when ButtonBlue .Click
do set DrawingCanvas .PaintColor to blue

when ButtonGreen .Click
do set DrawingCanvas .PaintColor to green

when ButtonWipe .Click
do call DrawingCanvas .Clear

when AccelShakeForMeow .Shaking
do call SoundMeow .Play

```

```

when DrawingCanvas .Touched
  x y touchedAnySprite
do call DrawingCanvas .DrawCircle
  centerX get x
  centerY get y
  radius 5
  fill true

```

```

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

```

TIPPS:



- (1) Falls Du mit einem Gmail-Gastkonto gearbeitet hast (nicht dein eigenes Konto), speichere deinen Projektcode unter:> Projects > Export selected project (.aia) to my computer
- (2) So kannst du dir selbst diesen Code (.aia) per E-Mail schicken, und später zu Hause oder in der Schule den Code weiter bearbeiten.

Die originelle detaillierte Anweisung für das «Farbtopf Teil-1» Tutorial findet man unten:

<http://appinventor.mit.edu/explore/ai2/paintpot-part1.html>

Farbtopf: Teil 2

Bist du bereit, ein paar Extras zu deiner App hinzuzufügen? Wie wär's, z.B. die Grösse der Punkte zu verändern oder das Hintergrundbild gegen ein eigenes auszuwechseln, das du selbst gemacht hast? Gehen wir Schritt nach Schritt:

Speichere deine App «FarbtopfTeil1» als neues Projekt «FarbtopfTeil2». Unter dem Menu:
> Projects > Save project as

Füge die folgenden Komponenten im Designer-Fenster ein, und passe gemäss der unteren Tabelle die Namen und Eigenschaften an.

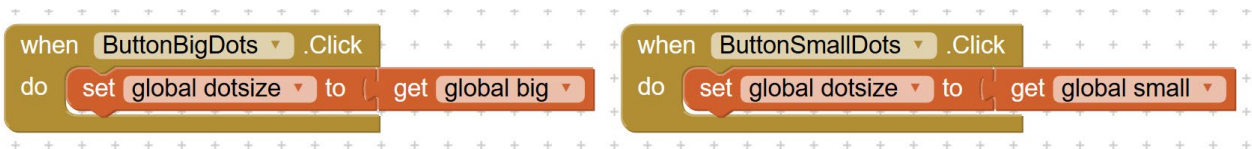
Komponenten	Palette (Kategorie)	Umbenennen in	Wofür plus Eigenschaften
Horizontal Arrangement	Layout	BottomButtons	Anordnung um neue Tasten unterhalb der Leinwand zu platzieren
Button (3x)	User Interface	ButtonBigDots ButtonSmallDots ButtonTakePicture	Tasten um die Grösse der Punkte zu ändern (grosser oder kleiner) Um die Kamera zu starten
Camera	Media	Camera1	Als Verbindung zur Kamera (Ein- und Ausgänge tauschen)

Jetzt müssen wir die entsprechenden Aktionen im Blocks-Fenster programmieren.

- Wir brauchen eine Variable als Zwischenspeicher. Eine Variable ist wie eine Schublade mit einer Etikette drauf (**Variablen-Name**), und einem Zettel drin, auf dem ein Wert aufgeschrieben ist (**Variablen-Wert**). Den nötigen Block findest du unten > Built-in > Variables Für jede Variable vergibst du einen Namen und einen Anfangswert.



- Wenn der Benutzer auf die entsprechende Taste drückt, soll diese Auswahl für die Grösse der Punkte gespeichert werden. Das heisst: der Wert der Variable **dotsize** wird verändert.



- Als Nächstes musst du herausfinden, in welchem Block du die Variable **dotsize** brauchst; sodass die Punkte tatsächlich kleiner oder grösser gezeichnet werden. Teste, ob das klappt!

Sehr schön! Jetzt wollen wir aber nicht mehr auf der Katze zeichnen, sondern selber ein Foto knipsen und darauf zeichnen. Dafür brauchst du zwei Aktionen.

- Erste Aktion: Wenn man auf «ButtonTakePicture» drückt, soll die Kamera aktiviert werden, um ein Foto zu knipsen. Kannst du so eine Aktion finden? Wo würdest du das suchen?
- Zweite Action: Nachdem man das Foto geknipst hat («AfterPicture»), soll es als Hintergrundbild (Englisch: **BackgroundImage**) der Leinwand (DrawingCanvas) eingesetzt werden.

Alles parat? Zeit zum Testen und eventuelle Fehler zu korrigieren. Überprüfe deinen Code mit der Übersicht unten.

Übersicht Farbtopf Teil 2: Designer-Fenster

The screenshot shows the Android Studio Designer interface for an application named "PaintPot".

- Viewer Panel (Left):** Contains a preview of the app's UI. At the top, there are checkboxes for "Display hidden components in Viewer" and "Check to see Preview on Tablet size.". Below these is a status bar showing the time 9:48 and various icons. The app's title bar says "PaintPot". The main content area features three colored buttons: "Red" (red), "Blue" (blue), and "Green" (green). Below the buttons is a large image of an orange cat. At the bottom of the preview, there are four buttons: "Wipe", "BigDots", "SmallDots", and "Take Picture". A "Non-visible components" section at the very bottom lists "AccelShakeForMeow", "SoundMeow", and "Camera1" with their respective icons.
- Components Panel (Middle):** A tree view of the app's components. The root is "Screen1", which contains a "ThreeColorButtons" component (with sub-components "ButtonRed", "ButtonBlue", and "ButtonGreen"), a "DrawingCanvas", and a selected "BottomButton" component (with sub-components "ButtonWipe", "ButtonBigDots", "ButtonSmallDots", and "ButtonTakePicture"). Other components listed include "AccelShakeForMeow", "SoundMeow", and "Camera1". "Rename" and "Delete" buttons are located at the bottom of this panel.
- Properties Panel (Right):** Shows the properties for the selected "BottomButton" component. The "Visible" property is checked. Other properties include "AlignHorizontal" (Left: 1), "AlignVertical" (Top: 1), "BackgroundColor" (Default), "Height" (Automatic...), and "Width" (Fill parent...).
- Media Panel (Bottom Right):** Lists media files "kitty.png" and "meow.mp3" with an "Upload File ..." button.

Übersicht Farbtopf Teil 2: Blocks-Fenster

```

initialize global dotsize to 2
initialize global small to 2
initialize global big to 8

when ButtonRed .Click
do set DrawingCanvas . PaintColor to red

when ButtonBlue .Click
do set DrawingCanvas . PaintColor to blue

when ButtonGreen .Click
do set DrawingCanvas . PaintColor to green

when ButtonWipe .Click
do call DrawingCanvas . Clear

when ButtonBigDots .Click
do set global dotsize to get global big

when ButtonSmallDots .Click
do set global dotsize to get global small

when DrawingCanvas .Touched
x y touchedAnySprite
do call DrawingCanvas . DrawCircle
   centerX get x
   centerY get y
   radius get global dotsize
   fill true

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

when ButtonTakePicture .Click
do call Camera1 . TakePicture

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

```

Die originelle detaillierte Anweisung für das «Farbtopf Teil 2» Tutorial findet man unten:

<http://appinventor.mit.edu/explore/ai2/paintpot-part2.html>

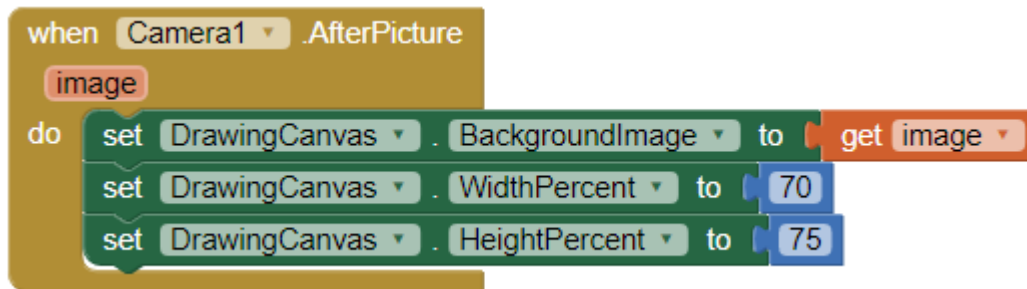
Farbtopf: Teil 3

Hast du einige Ideen für weitere Funktionen in diesem Farbtopf App? Hier unten ein paar Vorschläge als Inspiration:

- Erstelle eine Variable für die Dicke der Linien (ähnlich wie für die Grösse der Punkte)
- Mache die Grösse der Punkte und Linien frei wählbar. Füge dafür neue + und – Tasten hinzu
- Wie sieht das Foto auf deiner Leinwand aus? Versuche es nicht mehr verzerrt, sondern im richtigen Seitenverhältnis darzustellen.
- Das Foto 90° zu drehen ist etwas schwieriger. Eine mögliche Lösung kann man mit eine AppInventor Erweiterung implementieren. Siehe: <http://puravidaapps.com/image.php>
- Erweitere die Auswahl der Farben. Statt mit Tasten, soll man jetzt eine Liste für die Auswahl zeigen. Infos über Listen und entsprechenden Funktionen findest du unten: <http://appinventor.mit.edu/explore/ai2/support/blocks/lists.html>
- Oder du könntest als Hintergrund ein Foto aus deine Foto-Galerie auswählen. Dieses Video zeigt wie: <https://www.youtube.com/AppInventor2ImagePicker>

For the workshop instructors:

Small improvement for photo aspect ratio



Another step/question is to check how to control the orientation (landscape / portrait) of the photo taken. But apparently the solutions are not trivial. Saw 2 options:

- Display image inside a Sprite, since these can be turned. But some tricks on the way are needed. Check : <https://groups.google.com/forum/appinventor/imageturned>
- Or via an image extension from Pura Vidas: <http://puravidaapps.com/image.php>

Future improvements (further points for extending this App or building similar one):

- Add details on the implementation of the extension using ListPicker for color selection.
- Add a «Lernpunkte» rubric on Listpicker and Imagepicker, to support the extension ideas for PaintPot
- Add an alternative App implementation, for example: "Graffiti Artist" from book Hello App Inventor.