

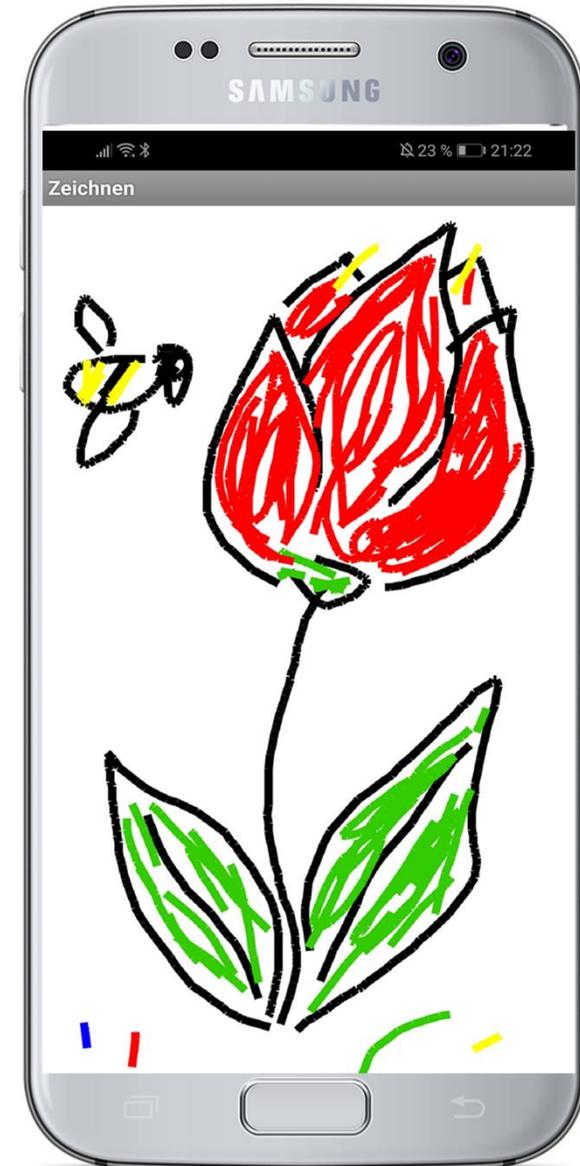
Projekt 2: Zeichen-App

Ziel

Wir wollen mit dem Finger auf dem Handy zeichnen.

Dazu erstellen wir eine App, mit der wir auf dem Touchscreen zeichnen können. Wenn wir das Handy schütteln, soll die Zeichenfläche wieder gelöscht werden.

Später werden wir diese App noch erweitern und verschönern.

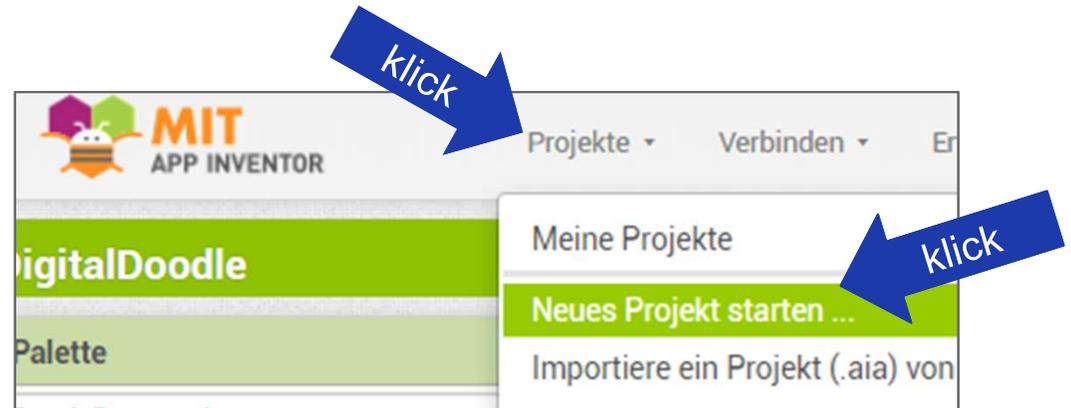


Teil 1: Zeichnen und Löschen

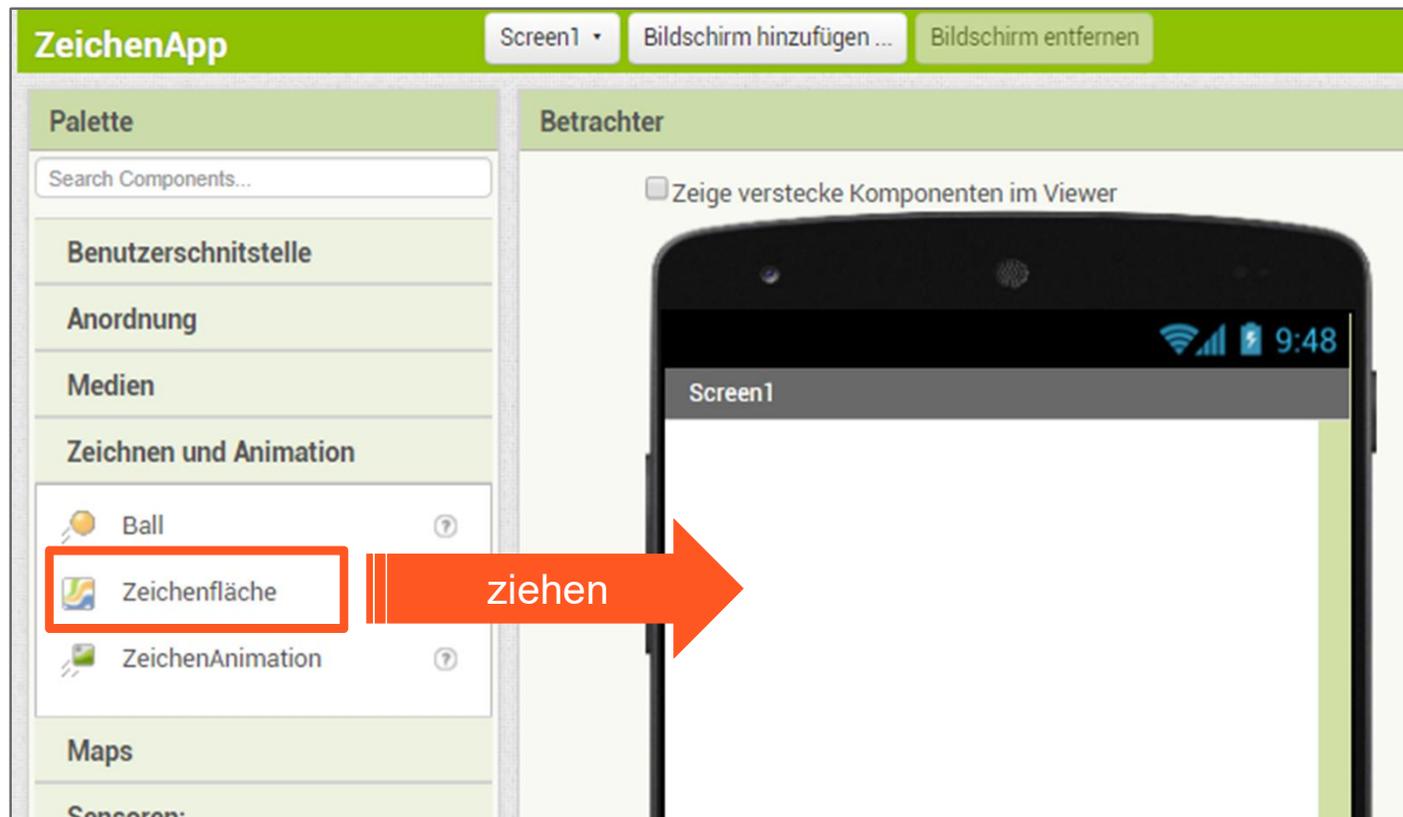
Im ersten Teil programmieren wir unsere App so, dass wir mit dem Finger auf dem Bildschirm zeichnen können. Außerdem soll der Bildschirm gelöscht werden, wenn man das Handy schüttelt.

Anleitung (1)

1. Lege im MIT App Inventor ein neues Projekt an.



2. Ziehe aus dem linken Bereich *Zeichnen und Animation* eine *Zeichenfläche* auf das abgebildete Handy.



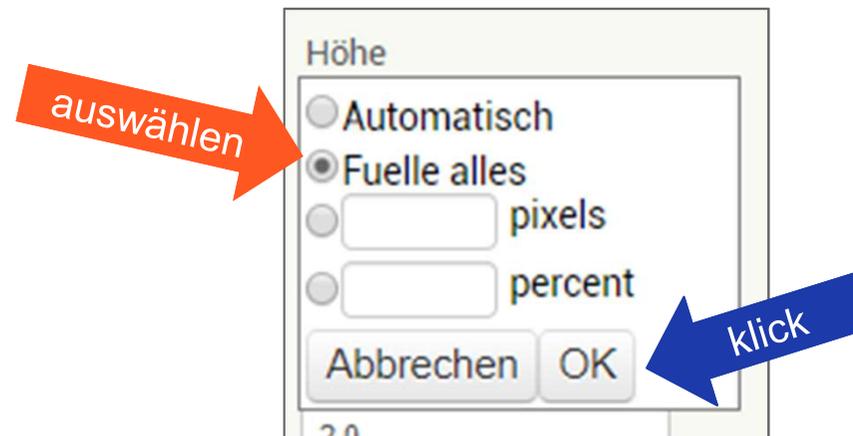
Anleitung (2)

3. Damit die Zeichenfläche den ganzen Bildschirm ausfüllt, müssen wir ihre *Eigenschaften* ändern. Das geht so:

a. Klicke die Zeichenfläche an



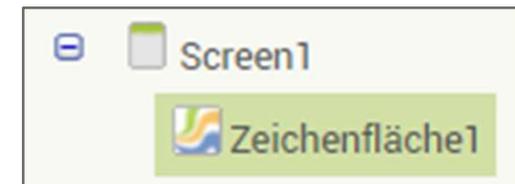
b. Im Bereich *Eigenschaften* (ganz rechts) können wir jetzt unsere Zeichenfläche einrichten. Hier setzen wir die *Höhe* und die *Breite* jeweils auf *Fülle Alles*.



Anleitung (3)

4. Unsere Benutzeroberfläche ist jetzt fertig. Wechsle jetzt in die Ansicht *Blöcke*.

5. Klicke im linken Bereich *Blöcke* auf die Zeichenfläche.



6. Ziehe den Block *wenn Zeichenfläche1.Gezogen* auf die weiße Fläche.

The screenshot shows the 'Blöcke' (Blocks) view of the 'ZeichenApp' software. The interface is divided into several sections:

- Header:** 'ZeichenApp' title, 'Screen1' dropdown, 'Bildschirm hinzufügen ...' and 'Bildschirm entfernen' buttons, and 'Designer' / 'Blöcke' tabs.
- Blöcke (Left Panel):** A category list including 'Eingebaut' (Steuerung, Logik, Mathematik, Text, Listen, Farben, Variablen, Prozeduren), 'Screen1', and 'Zeichenfläche1'.
- Betrachter (Right Panel):** A list of event blocks for 'Zeichenfläche1':
 - .Gezogen:** Contains parameters 'startX', 'startY', 'vorherX', 'vorherY', 'aktuellesX', 'aktuellesY', and 'draggedAnySprite'.
 - .Geschleudert:** Contains parameters 'x', 'y', 'Geschwindigkeit', 'Ueberschrift', 'xvel', 'yvel', and 'SprungAnimation'.
 - .DrueckeRunter:** Contains parameters 'x' and 'y'.
- Drawing Area (Far Right):** A white canvas with a blue backpack icon. A blue arrow labeled 'ziehen' points from the '.Gezogen' block to this area.

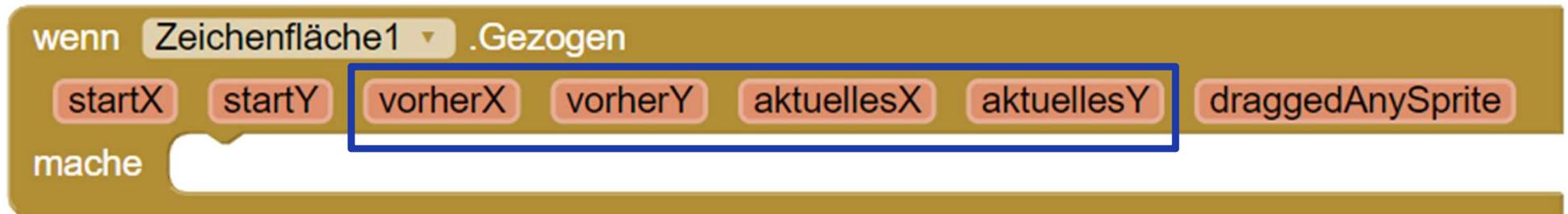
Einschub: Ereignisbehandlung

Die braunen Blöcke, die mit “*wenn*” beginnen. Reagieren auf Ereignisse im Programm. Wir kennen schon den Ereignisbehandler für den Klick auf einen Button.



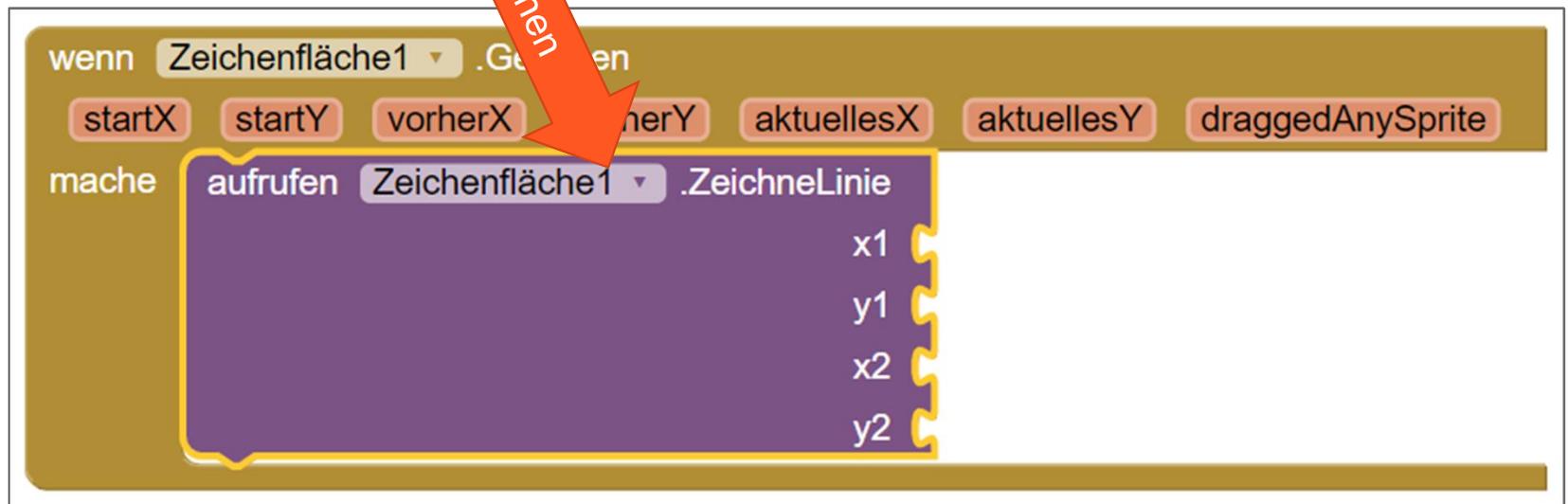
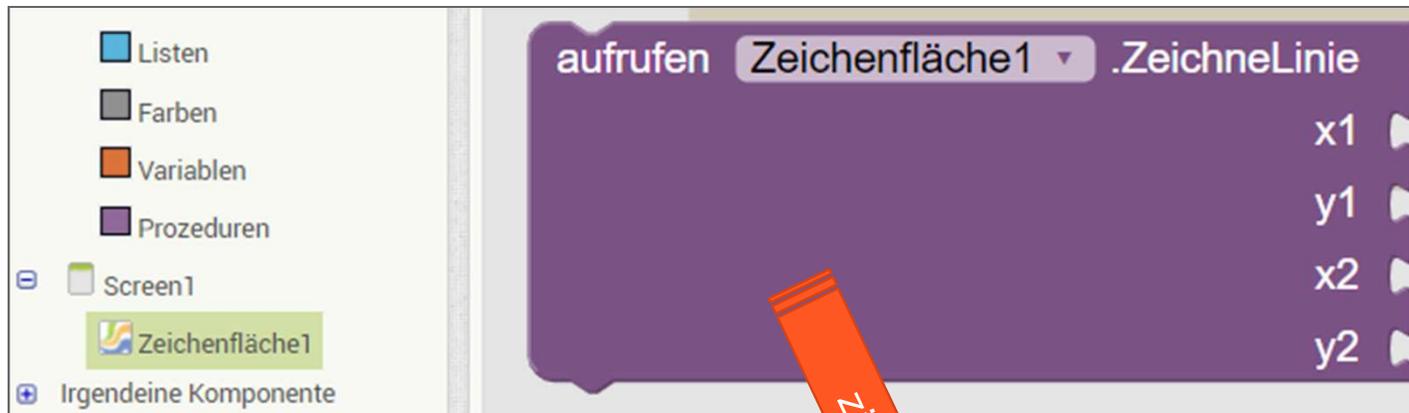
Die Blöcke, die wir hier reinschieben, werden immer ausgeführt, wenn der *Button1* angeklickt wird.

Wenn wir den Finger über die Zeichenfläche ziehen löst diese Zeichenfläche das Ereignis *Gezogen* aus. Wie bekommen gleich mitgeliefert, von wo (*vorherX*, *vorherY*) nach wo (*aktuellesX*, *aktuelleY*) der Finger gezogen wurde.



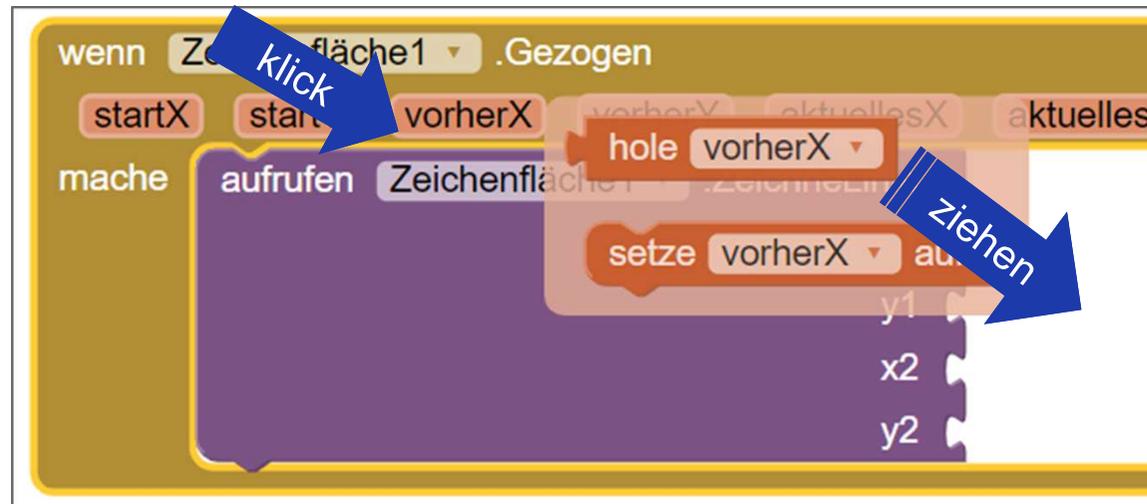
Anleitung (4)

7. Wähle links im Bereich Blöcke die *Zeichenfläche1* aus und ziehe den Block *aufrufen Zeichenfläche1.ZeichneLinie* in das Ereignis.



Anleitung (5)

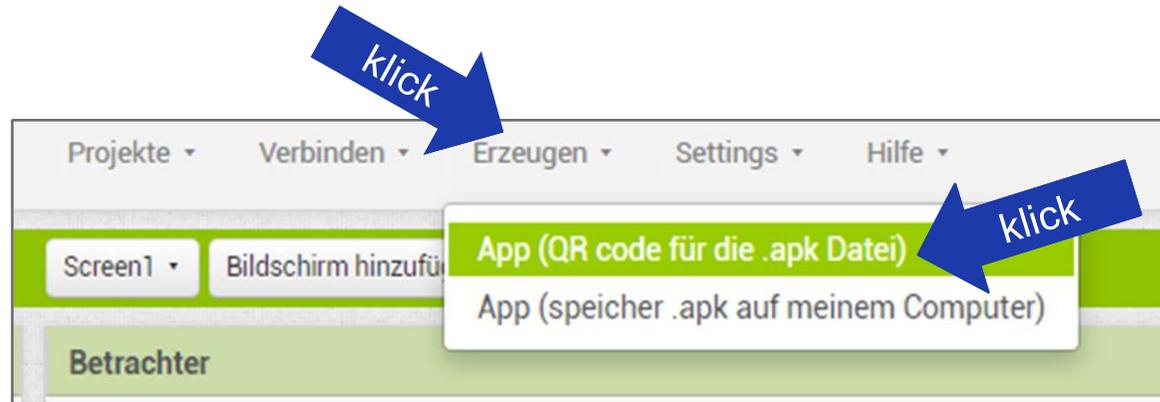
8. Der Block *ZeichneLinie* zeichnet eine Linie auf die Zeichenfläche. Wir müssen nur noch sagen, von wo nach wo die Linie gezeichnet wird. Da nehmen wir die Daten, die uns das Ereignis *Gezogen* liefert und ziehen die in den *ZeichneLinie*-Block.



Am Ende soll es so aussehen.

Anleitung (6)

9. Fertig! Jetzt können wir die App herunterladen und testen.

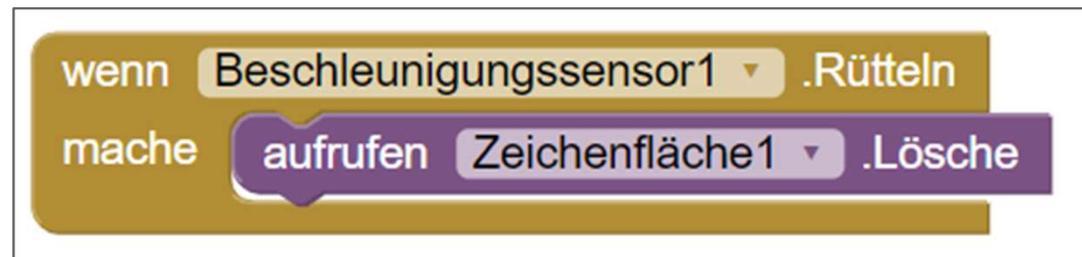


Den QR-Code, der erscheint, können wir einscannen, um die App zu laden.

Erweiterung

Jetzt soll die Zeichenfläche gelöscht werden, wenn wir das Handy schütteln. Mache dazu folgendes:

1. Wechsle zurück in die Ansicht *Designer*.
2. Ziehe aus dem linken Bereich *Palette* aus dem Bereich *Sensoren* den *Beschleunigungssensor* auf das Handy.
3. Wechsle in Ansicht *Blöcke*.
4. Ziehe das Ereignis *Rütteln* von *Beschleunigungssensor1* auf die Blockfläche.
5. Platziere in dem Ereignis den Block *aufrufen Zeichenfläche1.Lösche*.



6. Fertig! Jetzt können wir die App wieder herunterladen und testen.

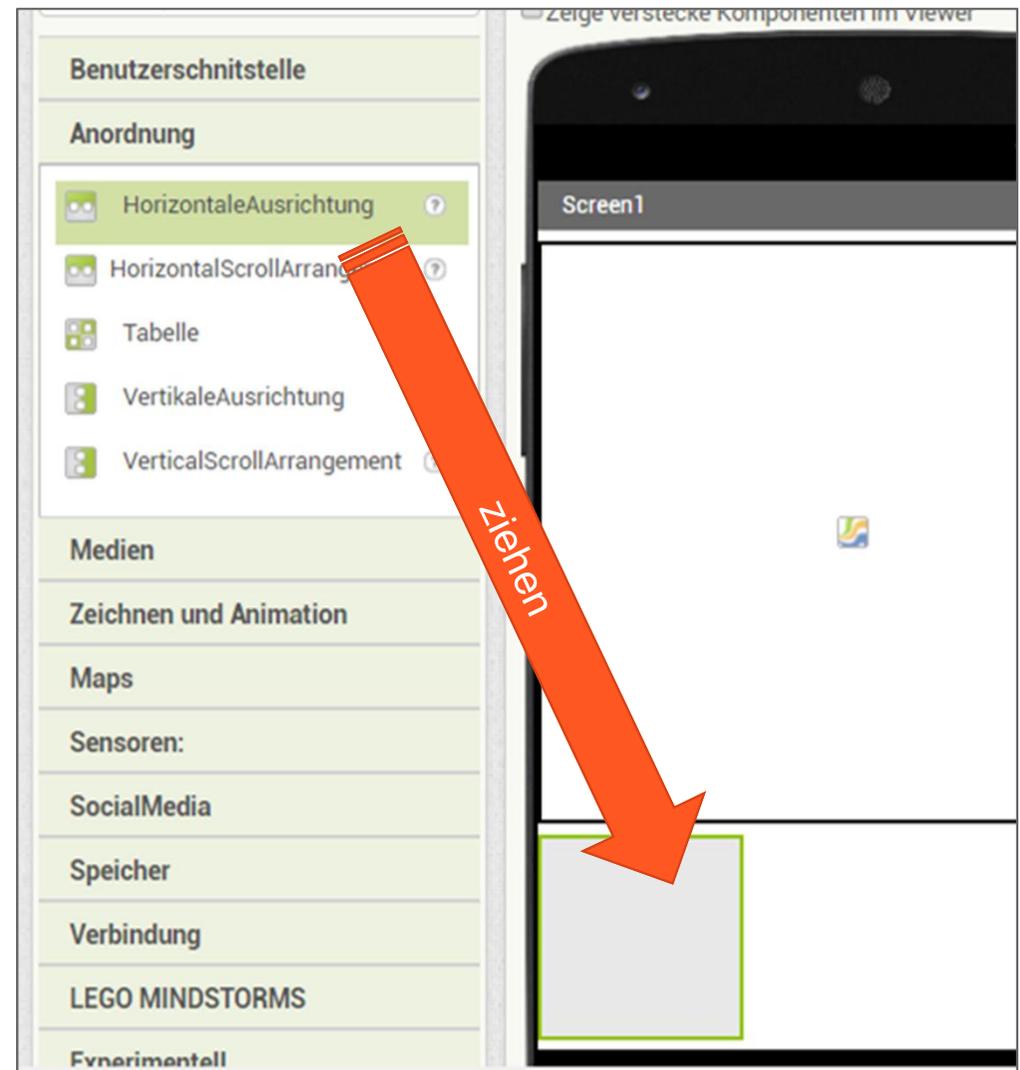
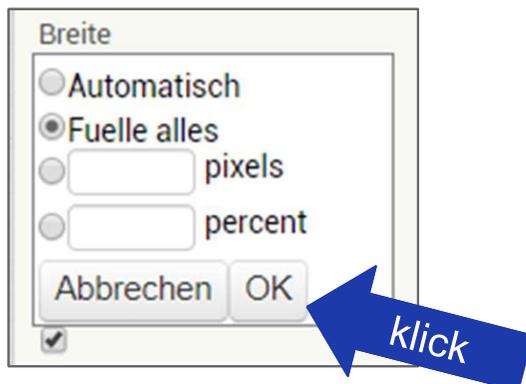
Teil 2: Zeichenfarbe ändern

Im zweiten Teil fügen wir der App Buttons hinzu, mit denen wir die Zeichenfarbe ändern können.

Wenn das funktioniert können wir farbige Bilder malen.

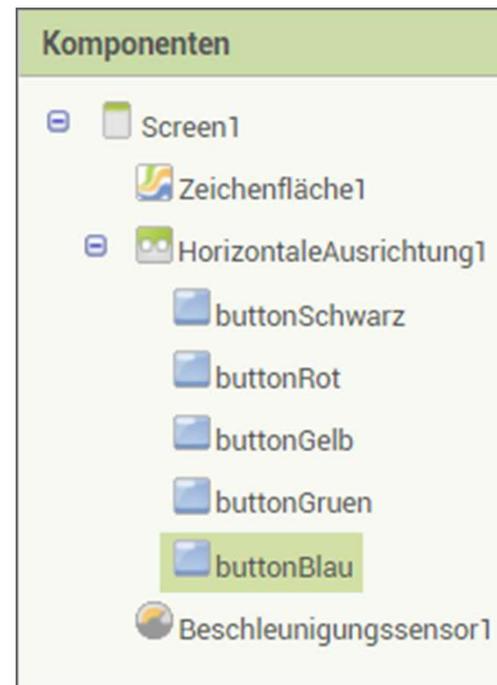
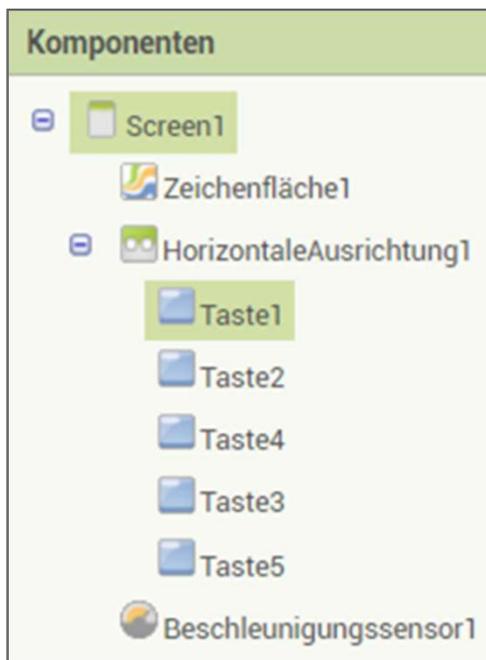
Anleitung (1)

1. Für die Auswahl der Farbe platzieren wir mehrere Buttons unterhalb der Zeichenfläche. Das machen wir mit den folgenden Schritten:
 - a. Wechsle in die Ansicht *Designer*.
 - b. Ziehe aus dem Bereich *Palette* aus dem Abschnitt *Anordnung* eine *HorizontaleAusrichtung* unter die Zeichenfläche.
 - c. Setze die Breite von *HorizontaleAusrichtung1* auf *Fülle alles*.



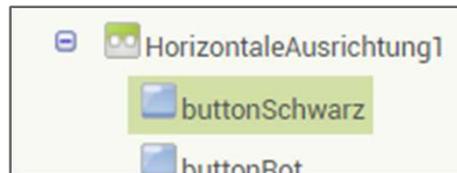
Anleitung (2)

2. Ziehe aus dem linken Bereich *Palette* auf dem Abschnitt *Benutzerschnittstelle* fünf Buttons (die heißen auf deutsch “*Taste*”) in die *HorizontaleAusrichtung1*. Die Buttons werden automatisch nebeneinander platziert.
3. Benenne Buttons *Taste1*, ... *Taste5* im Bereich *Komponenten* um in *buttonSchwarz*, *buttonRot*, *buttonGelb*, *buttonGruen* und *buttonBlau*. Klicke dazu auf die Namen der Buttons in der Liste und klicke auf *Umbenennen*. Dann kannst du den neuen Namen eingeben.

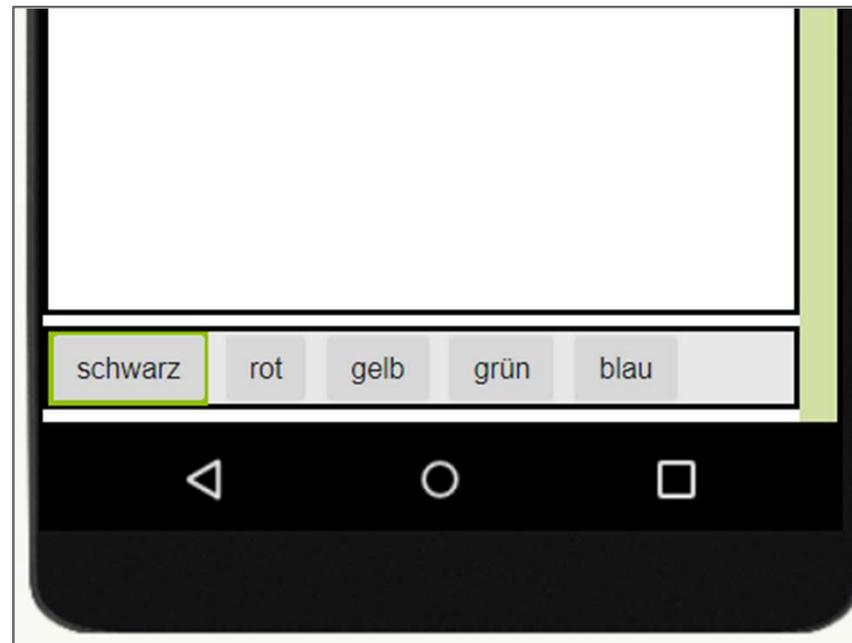


Anleitung (3)

4. Wähle im Bereich *Komponenten* nacheinander alle Buttons aus. Ändere für jeden Button den angezeigten Text, indem du im Bereich *Eigenschaften* in das Feld *Text* den Namen der entsprechenden Farbe schreibst.

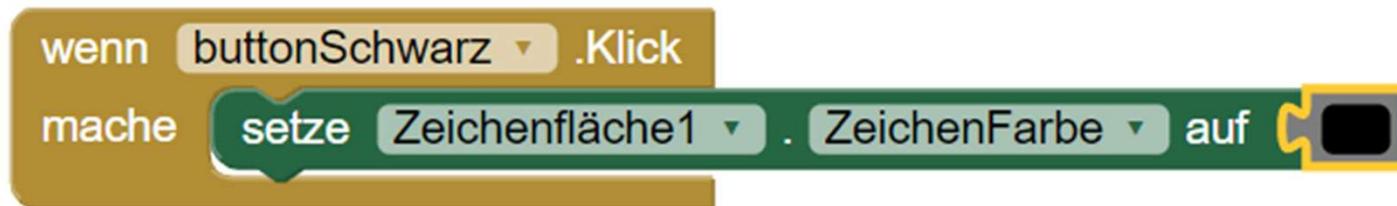


Insgesamt soll das dann so aussehen



Anleitung (4)

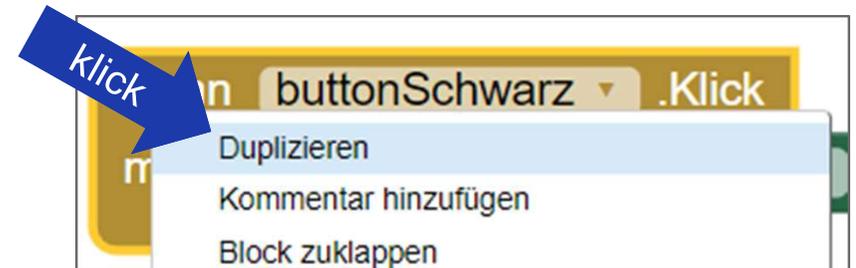
5. Jetzt müssen wir noch die Zeichenfarbe der Zeichenfläche ändern, wenn die Button geklickt werden. Wechsle dazu in die Ansicht *Blöcke*.
6. Klicke links im Bereich *Blöcke* den Button *buttonSchwarz* an und ziehe das Ereignis *wenn buttonSchwarz.Klick* auf die Code-Fläche.
7. Klicke links im Bereich *Blöcke* die *Zeichenfläche1* an. Ziehe den grünen Block *setze Zeichenfläche1.Zeichenfarbe auf* in den Klick-Block.
8. Klicke links im Bereich *Blöcke* auf *Farben* und füge das schwarze Puzzleteil an den Block mit der Zeichenfarbe an.



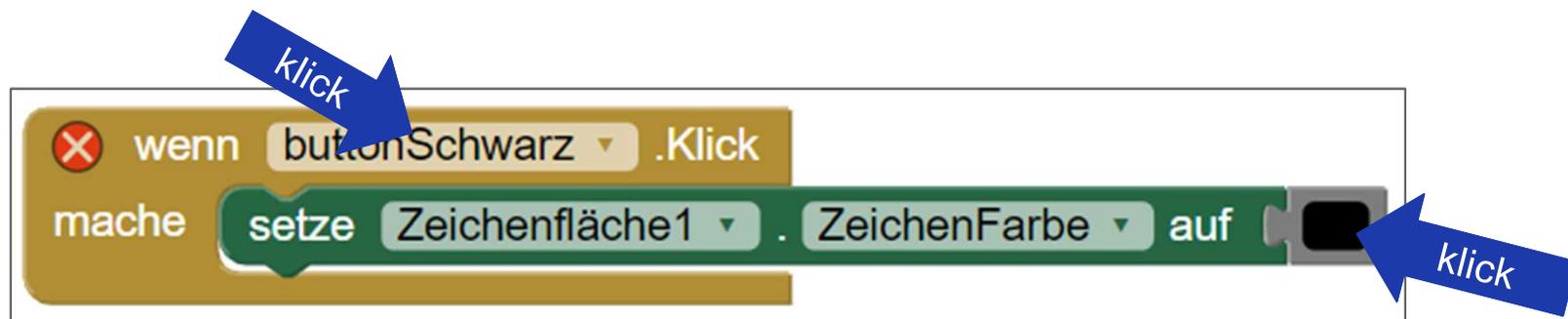
9. Setze nun auch für die anderen Buttons die Zeichenfarbe auf die entsprechende Farbe. (Beachte dazu den Tipp auf der nächsten Seite)
- 10. Fertig!** Jetzt können wir die App wieder herunterladen und testen.

Tipp

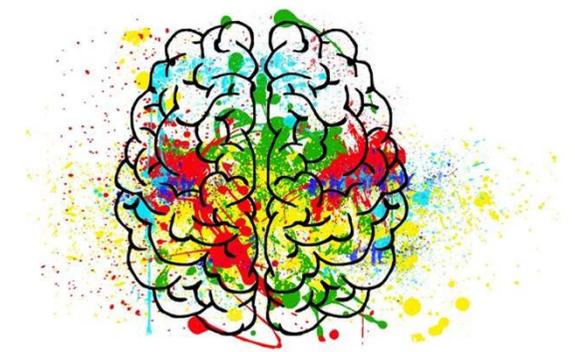
Wenn man viele ähnliche Code-Blöcke braucht, kann man die auch kopieren und dann die Kopien ändern. Mit unseren Farb-Buttons können wir das so machen:



Den kopierten Block kann man jetzt ändern, indem man einen anderen Button und eine andere Farbe wählt.



Erweiterungen



Design 1

- Ersetze den App-Titel “Screen1” über die *Properties* durch einen anderen Text,
- Erhöhen die *Linienbreite* der *Zeichenfläche1* im Bereich *Eigenschaften* in der *Designer-Ansicht*. Dann wird der Strich dicker.
- Ändere die *Hintergrundfarbe* der *Zeichenfläche1* im Bereich *Eigenschaften* in der *Designer-Ansicht*. Klicke dazu mit der Maus auf das weiße Kästchen.

Design 2

- Anstatt der Farbnamen sollen nun die Farben auf den Buttons sein. Das erreichst du so:
 - a. Lade die Datei *Farben.zip* von www.markuskaupp.de/mobile-Apps-AG Entpacke die Datei und merke dir, wo die Daten liegen.
 - b. Wähle einen Button im Bereich *Komponenten* aus und klicke im Bereich *Eigenschaften* auf das Feld unter *Zeichen*.
 - c. Es öffnet sich ein Fenster klicke dort auf *Datei hochladen...* und wähle die passende Farbdatei aus.
 - d. Der Button ist jetzt farbig. Wenn du magst, kannst du jetzt den Text löschen.

Teil 3: Fotos einfügen

Die App soll so erweitert werden,
dass man Fotos aufnehmen und
einfügen kann.

Man kann dann in die Fotos
zeichnen.



Anleitung (1)

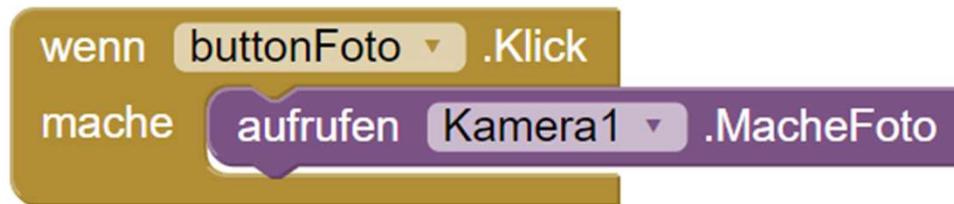
1. Füge in die *HorizontaleAusrichtung1* einen weiteren Button ein. Auf dem Button soll entweder der Text *Foto* stehen oder es soll das Kamera-Foto auf der Datei *Farben.zip* zu sehen sein.
2. Benenne den Button um in *buttonFoto*.
3. Falls nicht alle Buttons zu sehen sind, kannst du im Bereich *Eigenschaften* die Höhe und die Breite aller Buttons auf 40 Pixel setzen.



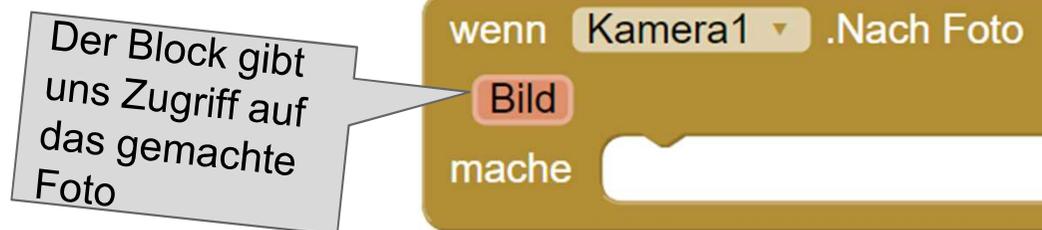
4. Ziehe aus dem linken Bereich *Palette* aus dem Abschnitt *Medien* eine *Kamera* auf das Handy im Bereich *Betrachter*. Bei den *nicht sichtbaren Komponenten* erscheint dann das Objekt *Kamera1*.
5. Die Oberfläche ist jetzt fertig. Wechsel in die Ansicht *Blöcke*.

Anleitung (2)

6. Wenn der Button *buttonFoto* angeklickt wird, soll die *Kamera1* ein Foto machen. Erstelle hierzu einen Code-Block, der so aussieht:

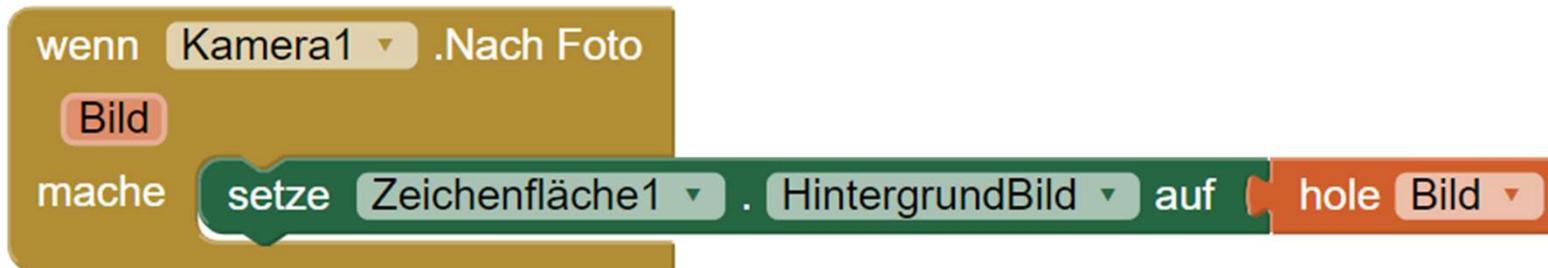


7. Wenn wir jetzt auf den Button klicken würden, dann würde sich die Kamera öffnen. Der Benutzer kann dann selber ein Foto machen. Jetzt müssen wir noch sagen, was passieren soll, wenn ein Foto gemacht wurde. Dafür gibt es das Ereignis *wenn Kamera1.Nach Foto*. Platziere diesen Ereignisblock im Bereich *Betrachter*.



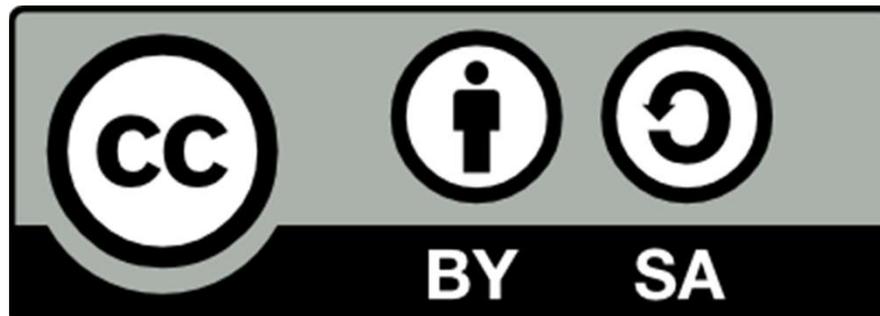
Anleitung (3)

8. Wenn das Ereignis *Kamera1.NachFoto* auftritt, soll das gemachte Foto als Hintergrundbild der *Zeichenfläche1* gesetzt werden. Füge den entsprechenden Code ein. Der Block soll jetzt so aussehen:



9. Das war's! **Jetzt ist unsere App fertig**. Bevor du die App ausführst, musst du ihr auf deinem Handy zwei Berechtigungen geben: *Foto* und *Speicherzugriff*. Viel Spaß beim Ausprobieren.

Lizenz



Dieses Material steht unter der Creative-Commons-Lizenz Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie <http://creativecommons.org/licenses/by-sa/4.0/>.

Urheber: Markus Kaupp