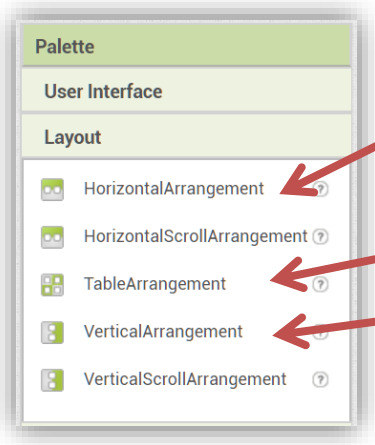
(2) Zeichnen für Fortgeschrittene

Bisher habt ihr euch um die Malfläche und das Hintergrundbild eurer Zeichenapp gekümmert. Jetzt stehen das Aussehen der App sowie das eigentliche Zeichnen auf dem Plan. Konkret werdet ihr...

- ... lernen, wie man **Screen Arrangements** einsetzen kann, um Komponenten einer App auf dem Handy-Display anzuordnen.
- ... dafür sorgen, dass man mit **verschiedenen Farben** auf dem Display zeichnen kann.

Das Aussehen der App

Im App Inventor können die einzelnen Objekte, die später auf dem Handy-Display dargestellt werden sollen, mit Hilfe von ***Screen Arrangements*** angeordnet werden. Ihr findet sie im ***Designer*** in der ***Palette*** unter ***Layout***.

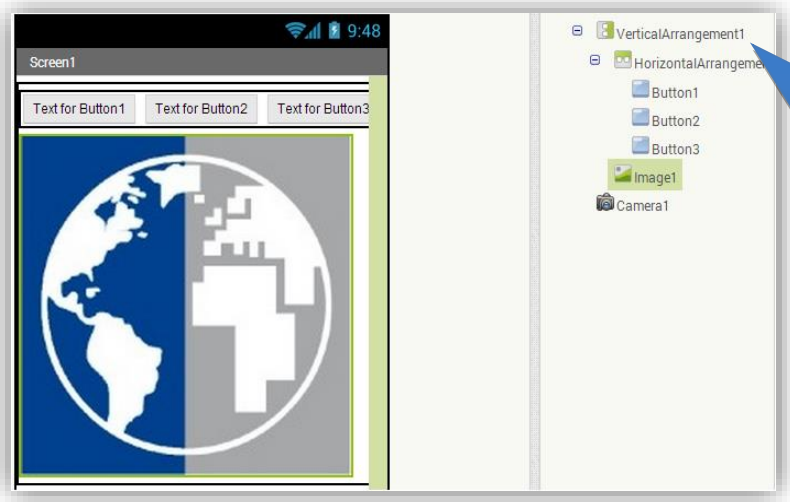


Elemente werden ***nebeneinander*** angeordnet.

Elemente werden in ***Blöcken*** (z.B. 4x4) angeordnet.

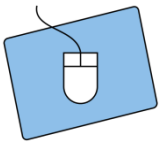
Elemente werden ***untereinander*** angeordnet.

Ihr könnt zwischen nicht scrollbaren und scrollbaren Arrangements wählen. Die einzelnen Varianten können auch kombiniert werden, wie z.B. im folgenden Bild zu sehen:



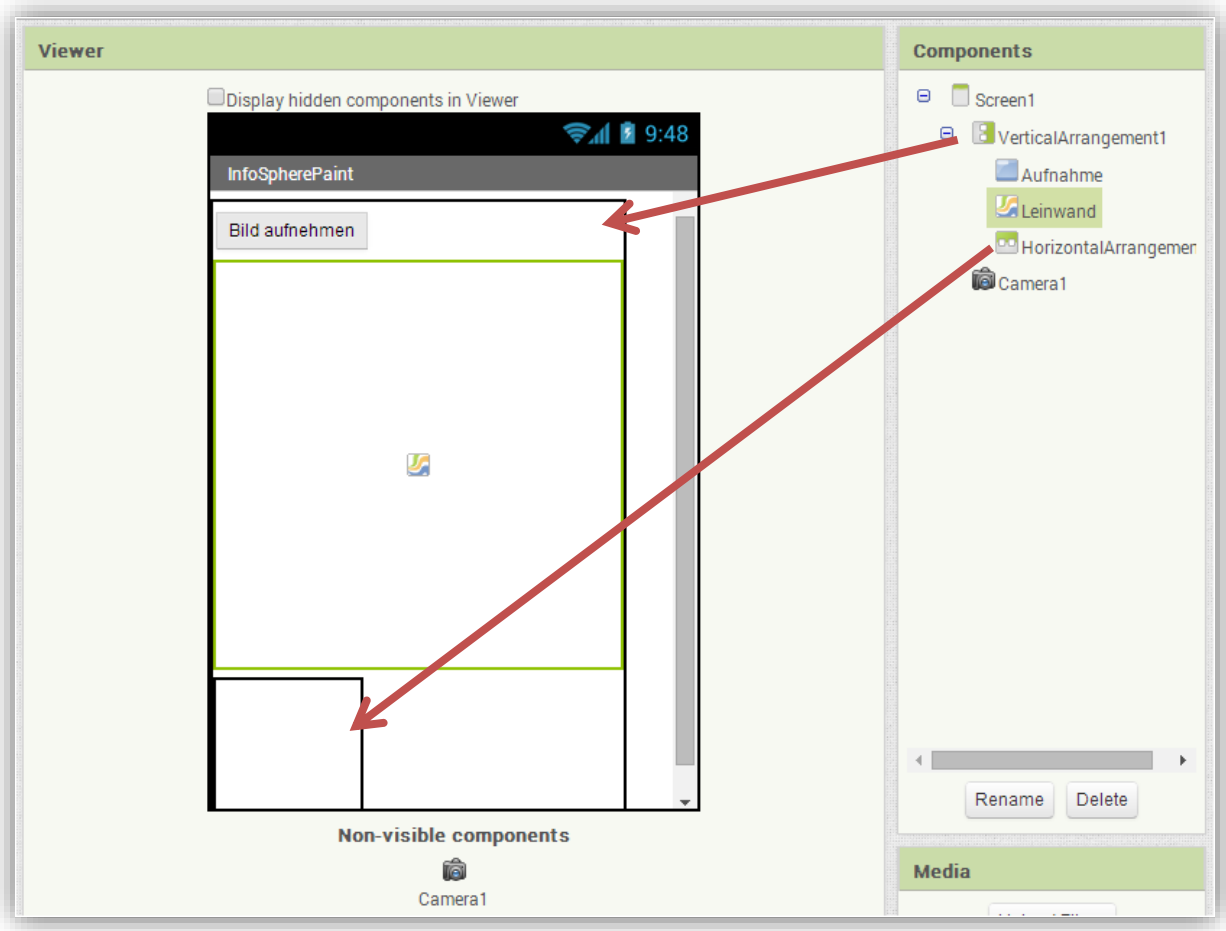
Hier wurden ein *vertikales* und ein *horizontales* Arrangement kombiniert, um drei Buttons nebeneinander (horizontal) über einem Bild (vertikal) darzustellen.

(2) Zeichnen für Fortgeschrittene



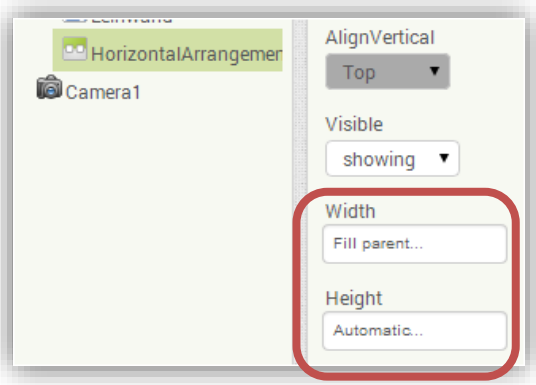
Ihr seid dran:

a) Zieht ein ***vertikales Screen Arrangement*** in die App im Viewer.
b) Zieht den Aufnahme-Button und die Leinwand in das Arrangement.
c) Zieht ein ***horizontales Arrangement*** in das vertikale Arrangement hinein.



In das leere Arrangement werden gleich die Elemente für das Zeichnen eingebaut.

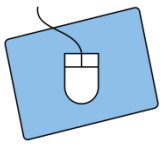
Um den Darstellungsbereich eures Handys voll auszunutzen, könnt ihr die ***Breite (Width)*** der beiden Screen Arrangements auf ***Fill Parent*** setzen.



Auf der nächsten Seite wird endlich gezeichnet...

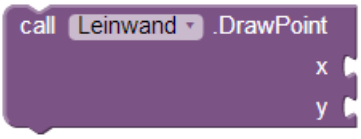
(2) Zeichnen für Fortgeschrittene

Zuerst ein Punkt...

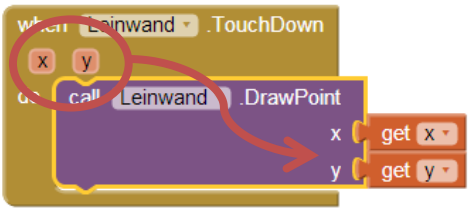


a) Sucht euch im **Blocks Editor** aus dem Menü für die Leinwand diesen Block:

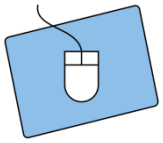
b) In die Lücke gehört der lila Block rechts, den ihr ebenfalls unter Leinwand findet. Er zeichnet einen Punkt *(Draw Point = Zeichne Punkt)* an die Koordinaten, die bei x und y stehen.



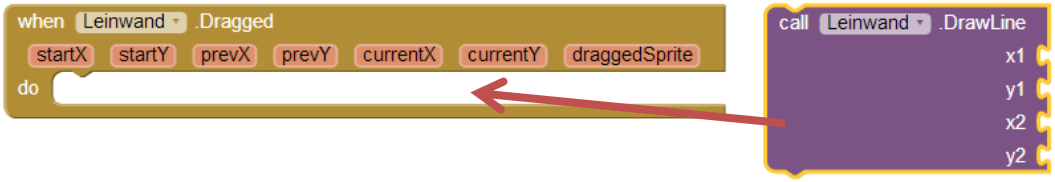
c) Fehlt noch der Platz, an dem der Punkt erscheinen soll. Fahrt mit der Maus über x bzw. y und zieht die **get-Blöcke** in die entsprechende Lücke. Diese liefern euch die Koordinaten des Platzes.

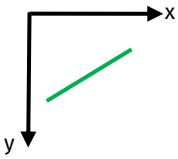


... dann ein Strich!



Ganz ähnlich funktioniert das mit einer Linie: Sucht euch die folgenden Blöcke aus dem **Leinwand-Menü** und setzt sie zusammen:

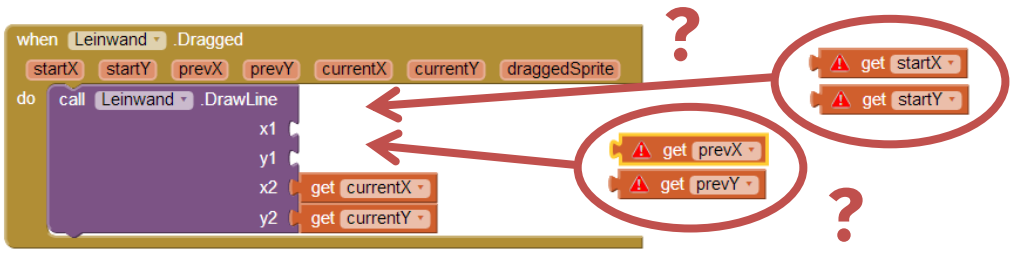




Der braune Block legt fest, was passiert, wenn man den Finger über den Bildschirm zieht *(drag = ziehen)*: Der lila Block sorgt dafür, dass eine Linie zwischen den **Punkten (x1, y1) und (x2, y2)** gezeichnet wird *(draw line = Linie zeichnen)*. In die Lücken für x2 und y2 gehören die get-Blöcke aus currentX bzw. currentY *(current = aktuell)*.

Welches Koordinatenpaar (start oder prev) gehört in die (x1, y1)-Lücken?
Was passiert, wenn man die falschen wählt?

startX und *startY* liefern den Punkt, an dem der Bildschirm zuerst berührt wurde; *prevX* und *prevY* liefern den Punkt, an dem der Finger vorher (previous = vorher) auf dem Bildschirm war. Einfach mal testen!



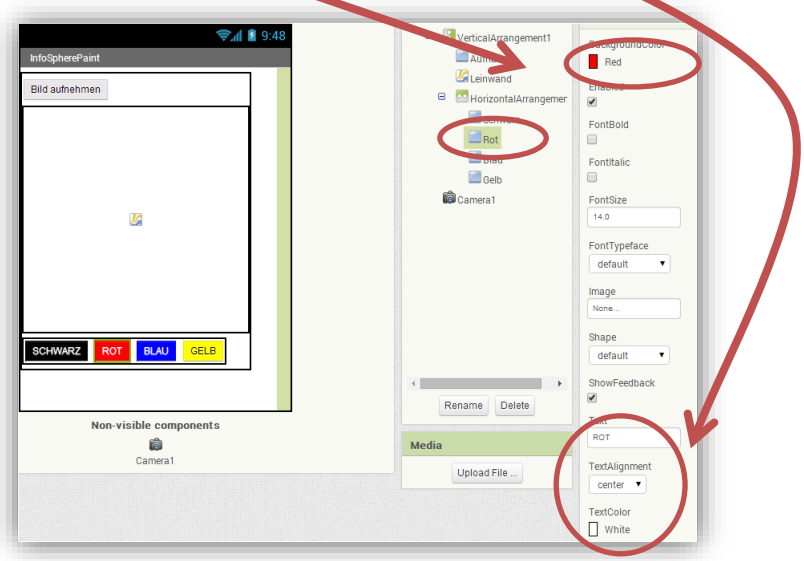
(2) Zeichnen für Fortgeschrittene

Alles wird bunt!

Damit ihr mit eurer App auch farbig zeichnen könnt, sorgt ihr jetzt dafür, dass man zwischen verschiedenen Farben und Strichstärken wählen kann:

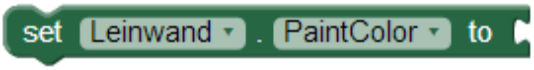
a) Wechselt in den Designer und zieht **vier Buttons** zur Farbauswahl in das leere horizontale Arrangement.
b) Benutzt **„Rename“**, um die Buttons in **Schwarz**, **Rot**, **Blau** und **Gelb** umzubenennen und ändert anschließend auch den Text der Buttons. Ihr könnt natürlich auch andere Farben verwenden.
c) Nutzt die Properties, um die Buttons zu gestalten. Ihr könnt die Funktionen *Backgroundcolor* (Hintergrundfarbe) oder *Textcolor* (Textfarbe) nutzen.

Und so könnte das Ganze dann aussehen:



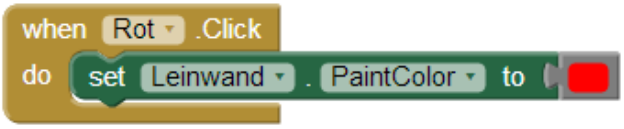
a) Sucht euch im Blocks Editor einen Farb-Button und zieht den Click-Block auf die Arbeitsfläche:

b) Im Auswahlmenü eurer Leinwand findet ihr diesen Block:



Er gehört in die Lücke des Click-Blocks.

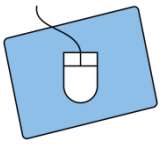
c) Unter **Colors** findet ihr Blöcke für die Farben. Zieht die passende Farbe in die hintere Lücke:



d) Wiederholt das Ganze für die übrigen Farben.

(2) Zeichnen für Fortgeschrittene

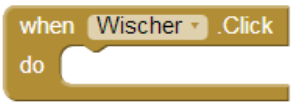
Stiftbreite und Wischer

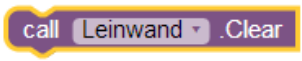


Um die Stiftbreite einzustellen und einen Wischer einzubauen wechselt ihr wieder in den Designer.

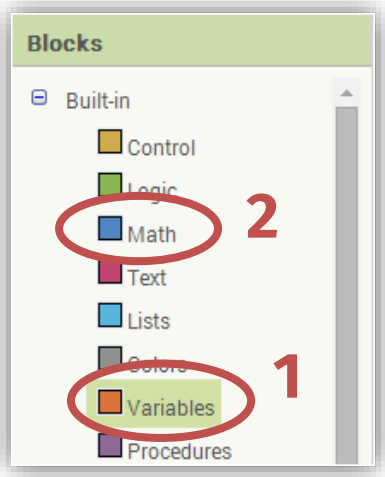
a) Zieht ein **neues horizontales Screen Arrangement** in die App, in das die neuen Elemente eingebaut werden.
b) Zieht zwei **Label** (Textfelder) aus *User Interface* in das neue Arrangement.
c) Gebt dem ersten Label sowohl den Namen als auch den Text *„Stiftbreite“*. Das zweite Label erhält den Namen *„Wert_Stiftbreite“* und als Text eine *2* (Voreinstellung für die Stiftbreite).
d) Als nächstes benötigt ihr **zwei Buttons**, die für die Einstellung der Stiftbreite zuständig sind. Gebt ihnen passende Namen, z.B. „Breite_rauf“ und „Breite_runter“ und ändert ihre Texte sinnvoll.

e) Fehlt noch der Wischer, der die Leinwand abwischt. Hier benötigt ihr noch einen **Button**, der eine passende *Bezeichnung* und eine *Beschriftung* bekommt.
f) **Wechselt in den Blocks Editor.**
g) Für den Wischer braucht ihr die folgenden Blöcke:





Könnt ihr euch denken, wo ihr die herbekommt?



Damit der Stift (euer Finger) malen kann, müsst ihr ihm eine Stiftbreite zuweisen. Dazu legt ihr eine **Variable (siehe 1)** an und nennt sie „Stiftbreite“:

Hier klicken, um den Namen zu ändern.

Hier fehlt noch der Startwert.

Den Startwert (die voreingestellte Stiftbreite) legt ihr fest, indem ihr den Zahlen-Baustein aus **Math (siehe 2)** in die Lücke einbaut und die 2 eintragt. Das sieht dann so aus:

Wie ihr die Stiftbreite verändern könnt, erfahrt ihr auf der nächsten Seite.

(2) Zeichnen für Fortgeschrittene

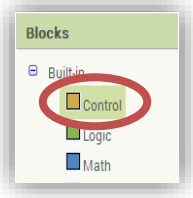
Hier lernt ihr, wie man die Stiftbreite einstellt…

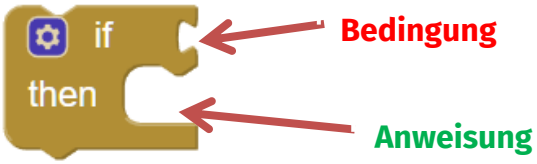
Die Stiftbreite sollte sich in einem Bereich von 1 bis 5 Pixeln *(Pixel = Bildpunkt)* bewegen. Also müsst ihr testen, ob sich die Stiftbreite in diesem Bereich befindet, bevor ihr sie ändert. Dafür nutzt ihr die **if-then-Anweisung** (Wenn-dann-Anweisung), die dafür sorgt, dass die Ausführung eines Programmteils nur erfolgt, wenn eine bestimmte Bedingung erfüllt ist, in etwa so:

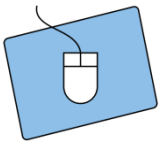
Wenn morgen die Sonne scheint, **dann gehe ich ins Freibad**.

Bedingung — Anweisung

Im App Inventor ist hierfür der **if-then-Block** zuständig, den ihr im Blocks-Editor unter **Control** findet.



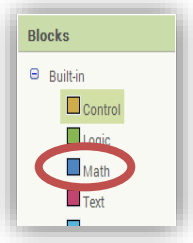


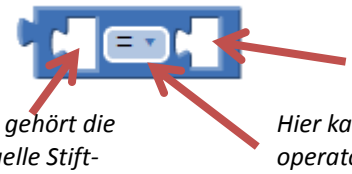


Die Stiftbreite erhöhen:

a) Sucht euch den Click-Block für den Button, der die Stiftbreite erhöhen soll. In die Lücke baut ihr den **if-then-Block** aus **Control** ein.

b) Für die Bedingung müsst ihr hier testen, ob die Stiftbreite noch **kleiner als** der Maximalwert ist. Ihr braucht also einen mathematischen Vergleich aus **Math**:





Hier steht der Maximalwert, also die Zahl 5.

Hier gehört die aktuelle Stift-breite hin.

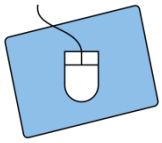
Hier kann der Vergleichs-operator ausgewählt werden (ihr benötigt <).

Die aktuelle Stiftbreite für den Vergleich erhaltet ihr über den **get-Block** aus der globalen Variablen, die ihr vorhin angelegt habt.

Die komplette Be-dingung sollte am Ende so aussehen:

*Um den **then-Teil** kümmert ihr euch auf der nächsten Seite.*

(2) Zeichnen für Fortgeschrittene



Im **then**-Teil sind die folgenden Schritte zu erledigen:

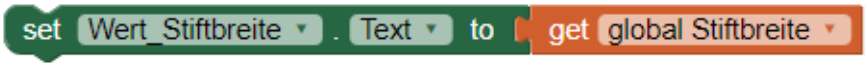
a) Die Stiftbreite um 1 erhöhen:

Den *set*- und *get-Block* erhaltet ihr wieder aus der globalen Variablen, den Block für die Addition aus **Math**, genau wie den Block für die Zahl.

b) Die Stiftbreite an die Leinwand weitersagen:

Den grünen Block findet ihr unter **Leinwand**. Er setzt die Linienbreite (Linewidth) der Leinwand auf den Wert, der hinten angehangen wird.

c) Das Label mit der Stiftbreite aktualisieren:



Den set-Block gibt euch das Menü für euer Label Wert_Stiftbreite.

Wenn ihr mit allen Schritten fertig seid, sollte euer Block zum Erhöhen der Stiftbreite so aussehen:

Genauso könnt ihr jetzt den Block zum Verringern der Stiftbreite zusammensetzen. Allerdings müsst ihr hier testen, ob der Wert der Stiftbreite noch **größer als das Minimum** ist und, wenn ja, die **Stiftbreite um 1 verringern**.

Damit ist eure Zeichenapp fertig. Ihr könnt sie jetzt testen und noch am Layout arbeiten oder euch das Blatt „(3) Zeichnen für Profis“ abholen und Zusatzfunktionen einbauen. Wenn ihr wollt, könnt ihr die App auch auf euer eigenes Android-Handy ziehen. Bittet einen Betreuer um Hilfe dabei.