

Spielidee für den MIT App Inventor

Reaktionsspiel

Tamara Jörns

Spielform: Zu zweit gegeneinander an einem Handy oder Tablet.

Spielidee: Es erscheinen die Schriftzüge „Blau“, „Grün“, „Gelb“ und „Rot“ geschrieben in jeweils einer der vier Farben. Jeder der beiden Spieler besitzt vier Buttons, einen in jeder Farbe. Es muss auf den Button mit der Farbe des ausgeschriebenen Wortes gedrückt werden. Der schnellere Spieler erhält einen Punkt.

Beispiel: Es erscheint das Wort „Blau“, geschrieben in der Farbe rot. Dann müssen die Spieler auf ihren jeweiligen blauen Button drücken.

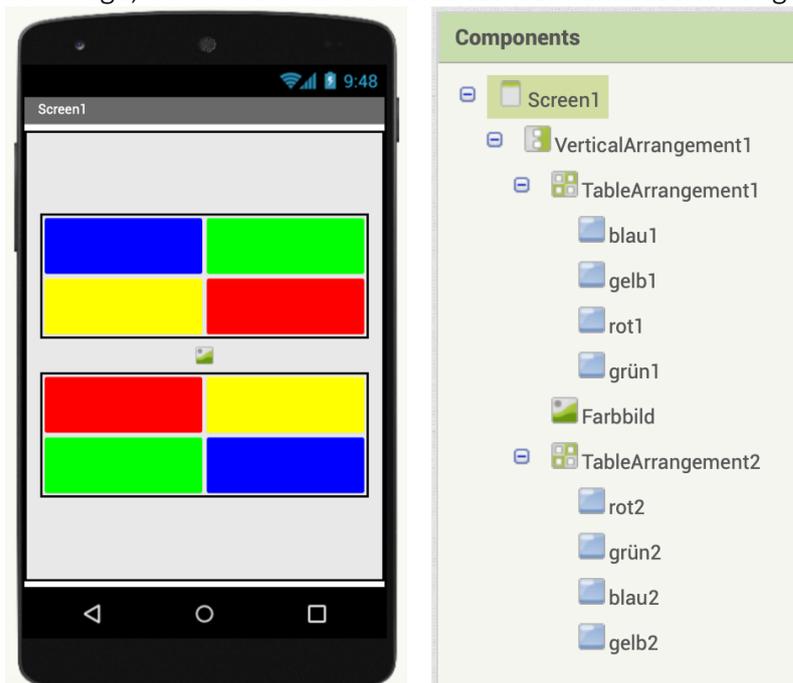
Realisiert wird die Anzeige der farbigen Schriften durch Bilder, da man mit dem MIT App Inventor keine Textfelder um 180 Grad drehen kann und somit einer der Spieler sonst den Text falsch herum sehen würde.



Schrittweise Umsetzung:

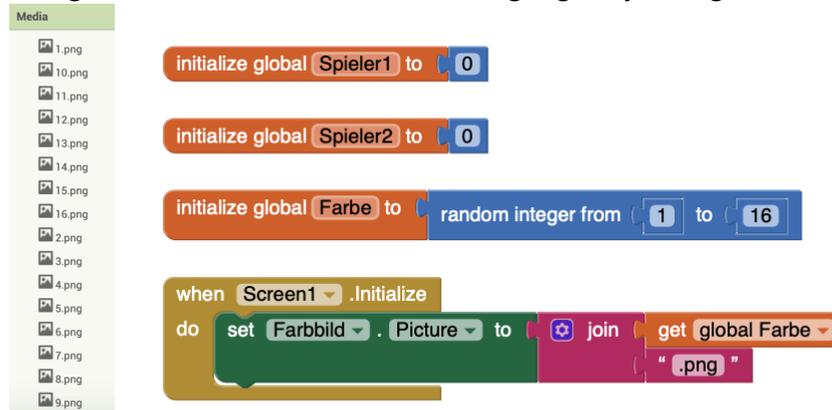
1. Grundgerüst der App basteln.

Benötigt werden jeweils vier Buttons je Spieler in den Farben blau, grün, gelb und rot, sowie ein Image, durch welches die Vorschau der Farbbezeichnungen realisiert wird.

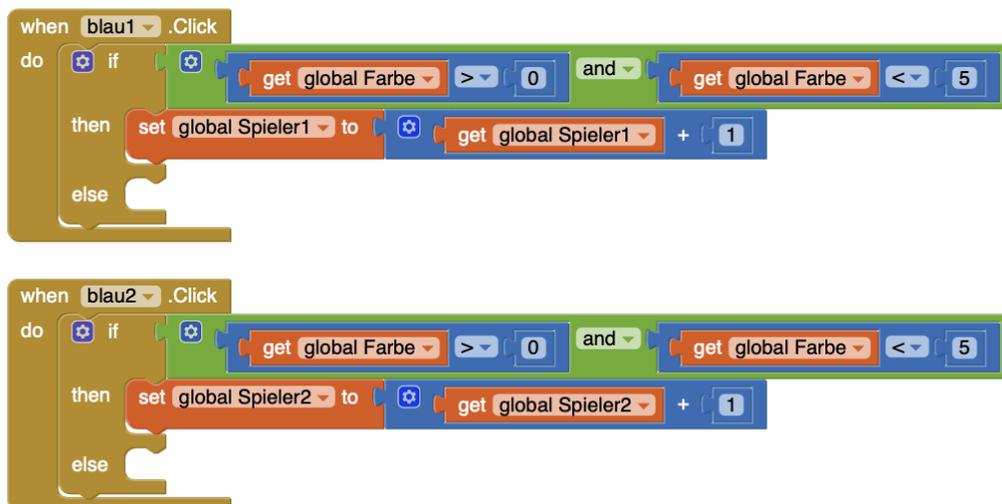


2. Die erste Realisierung.

Die Bilder mit den farbigen Schriften wurden von 1-16 durchnummeriert und in das Programm geladen. Die Bilder 1-4 bilden den Schriftzug „Blau“ einmal in je einer der vier Farben ab. Die Bilder 5-8 stellen den Schriftzug „Grün“, die Bilder 9-12 „Gelb“ und die Bilder 13-16 „Rot“ dar. Diese Benennung der Bilder ermöglicht es, durch Schleifen auf sie zuzugreifen und eine einfache Verzweigung zur jeweiligen Abfrage zu realisieren.



Was soll passieren, wenn eine Farbe angezeigt wurde und ein Spieler einen Button gedrückt hat? → Lag der Spieler richtig, so bekommt dieser einen Punkt. Lag er falsch dann nicht.



Analog für alle sechs weiteren Buttons.

Mögliche Erweiterungen: Hat ein Spieler den falschen Button geklickt, bekommt der Gegner einen Punkt oder der Spieler bekommt einen Punkt abgezogen.

3. Runden hinzufügen.

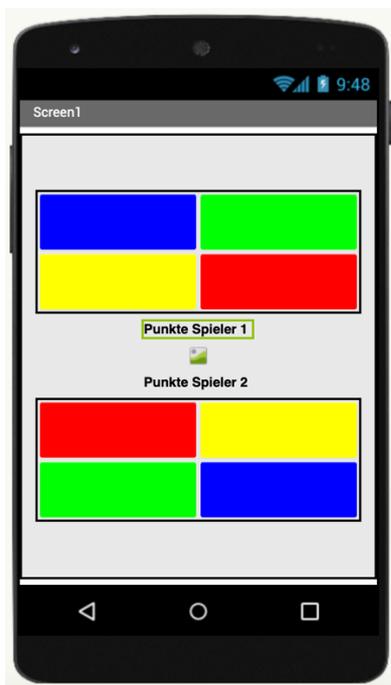
Damit ein Spiel auch ein Ende hat, wird festgesetzt wie viele Runden ein Spiel geht. In diesem Fall werden 10 Runden gespielt.

```
initialize global Runde to 0

when any Button.Click
  component notAlreadyHandled
  do
    if (get global Runde < 10)
      then
        set global Farbe to random integer from 1 to 16
        set Farbbild . Picture to join (get global Farbe) ".png"
        set global Runde to (get global Runde) + 1
      else
        break
```

4. Die Punktestände sollen angezeigt werden.

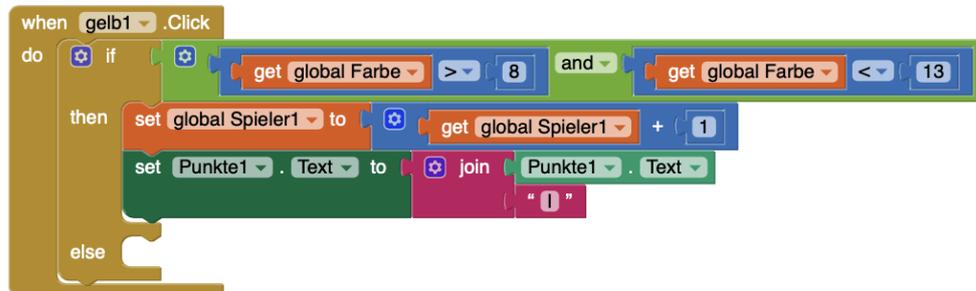
Den Spielern soll bereits während des Spiels angezeigt werden, wie der Punktestand in Echtzeit aussieht.



```
when grün1 .Click
  do
    if (get global Farbe > 4 and get global Farbe < 9)
      then
        set global Spieler1 to (get global Spieler1) + 1
        set Punkte1 . Text to (get global Spieler1)
      else
```

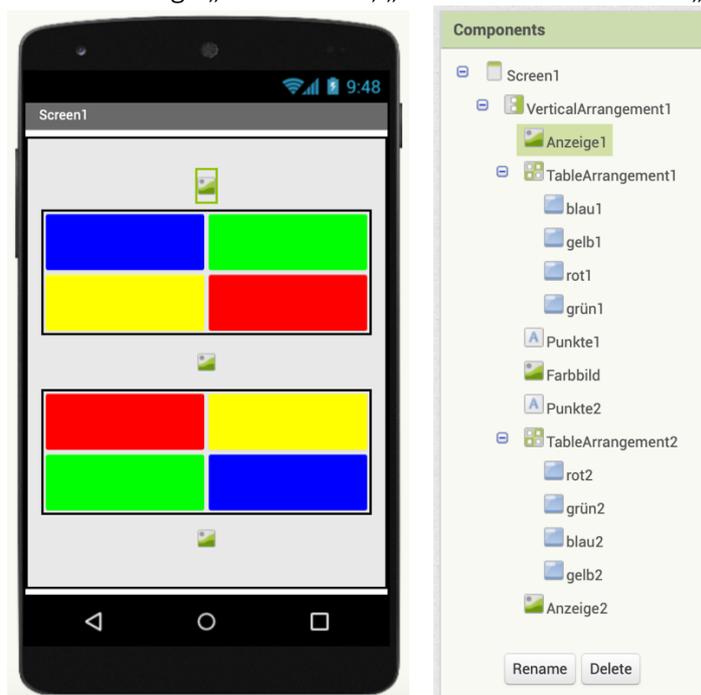
Auch hier ist wieder unschön, dass einer der Spieler seine Punkte auf dem Kopf angezeigt bekommt. Das drehen um 180 Grad eines beliebigen Textfelds ist mit dem MIT App Inventor leider nicht möglich.

Mögliche Lösung: Die Punkte werden durch eine Strichliste (i in Großbuchstaben) angezeigt.



5. Am Ende des Spiels soll der Gewinner angezeigt werden.

Auch hierzu werden wieder Bilder verwendet, um die gespiegelte Anzeige zu ermöglichen. Am Ende eines Spiels erscheinen in den neu hinzugefügten Image Feldern Bilder mit einem der Schriftzüge „Gewinner!“, „Leider verloren.“ oder „Unentschieden.“.



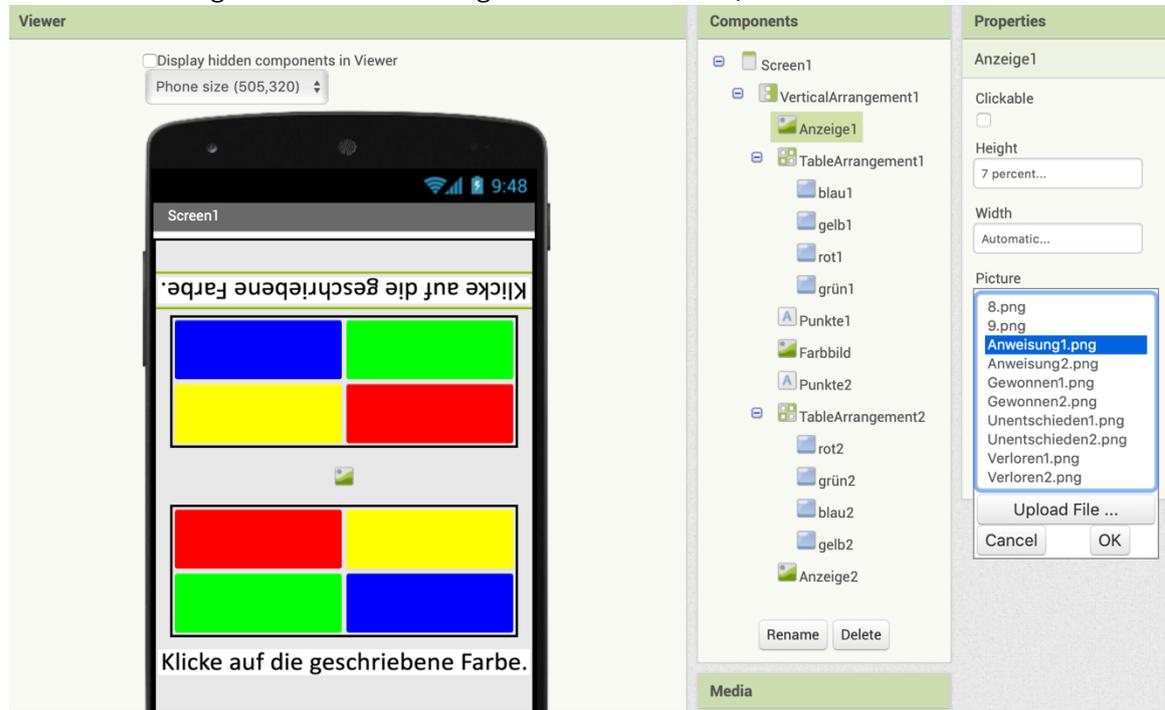
```

when any Button.Click
  component notAlreadyHandled
  do
    if (get global Runde < 10)
      then
        set global Farbe to random integer from 1 to 16
        set Farbbild . Picture to join (get global Farbe) ".png"
        set global Runde to (get global Runde) + 1
      else
        if (get global Spieler1 > get global Spieler2)
          then
            set Anzeige1 . Picture to "Gewonnen1.png"
            set Anzeige2 . Picture to "Verloren2.png"
          else if (get global Spieler2 > get global Spieler1)
            then
              set Anzeige2 . Picture to "Gewonnen2.png"
              set Anzeige1 . Picture to "Verloren1.png"
            else
              set Anzeige1 . Picture to "Unentschieden1.png"
              set Anzeige2 . Picture to "Unentschieden2.png"

```

6. Spielanleitung hinzufügen.

Damit die Spieler wissen was zu tun ist, wird diesen eine Spielanleitung angezeigt. Dazu werden die Image Felder für die Ausgabe des Gewinners/Verlieres verwendet.



7. Anzeigen einer korrekten Eingabe.

Nachdem ein Spieler einen Button geklickt hat, soll ihm angezeigt werden, ob seine Eingabe korrekt war.

In diesem Fall wird dies durch farbliche Markierung des Label Feldes realisiert, um das Problem der 180 Grad Drehung zu umgehen. Die Hintergrundfarbe des Labels wird auf grün für „richtig“ oder rot für „falsch“ gestellt.

```
when rot1 .Click
do
  if
    get global Farbe > 12 and get global Farbe < 17
  then
    set global Spieler1 to get global Spieler1 + 1
    set Punkte1 . Text to join Punkte1 . Text
    "1"
    set Punkte1 . BackgroundColor to green
  else
    set Punkte1 . BackgroundColor to red
```

Extra: Rücksetzen der Hintergrundfarbe des Labels eines Spielers, wenn der Gegner einen Button gedrückt hat.

```
when rot2 .Click
do
  if
    get global Farbe > 12 and get global Farbe < 17
  then
    set global Spieler2 to get global Spieler2 + 1
    set Punkte2 . Text to join Punkte2 . Text
    "1"
    set Punkte2 . BackgroundColor to green
    set Punkte1 . BackgroundColor to grey
  else
    set Punkte2 . BackgroundColor to red
    set Punkte1 . BackgroundColor to grey
```

8. Neues Spiel starten.

Ein Spiel ist beendet, Gewinner und Verlierer sind ermittelt, was nun? Ein neues Spiel soll gestartet werden können.

Dies wird hier realisiert, indem alles zurückgesetzt wird, sobald nach Ende eines Spiels ein beliebiger Button gedrückt wird. Dadurch startet das neue Spiel und es kann von vorne gespielt werden.

Eine Anweisung hierzu erscheint im Image Feld der Farbschriften nach beenden eines Spiels.

```
when any Button.Click
  component notAlreadyHandled
  do
    set global Runde to get global Runde + 1
    if get global Runde < 10
      then
        set global Farbe to random integer from 1 to 16
        set Farbbild . Picture to join get global Farbe ".png"
      else if get global Runde > 10
        then
          set global Runde to 0
          set global Spieler1 to 0
          set global Spieler2 to 0
          set Anzeige1 . Picture to "Anweisung1.png"
          set Anzeige2 . Picture to "Anweisung2.png"
          set Punkte1 . Text to ""
          set Punkte2 . Text to ""
          set Punkte1 . BackgroundColor to #cccccc
          set Punkte2 . BackgroundColor to #cccccc
          set global Farbe to random integer from 1 to 16
          set Farbbild . Picture to join get global Farbe ".png"
        else
          if get global Spieler1 > get global Spieler2
            then
              set Anzeige1 . Picture to "Gewonnen1.png"
              set Anzeige2 . Picture to "Verloren2.png"
            else if get global Spieler2 > get global Spieler1
              then
                set Anzeige2 . Picture to "Gewonnen2.png"
                set Anzeige1 . Picture to "Verloren1.png"
            else
              set Anzeige1 . Picture to "Unentschieden1.png"
              set Anzeige2 . Picture to "Unentschieden2.png"
          set Farbbild . Picture to "newgame.png"
```

Mögliche Erweiterungen: Es wird gezählt wer wie viele Spiele gewinnt.