

Projekt 5:

Elfmeterschießen

Ziel

Wir möchten ein kleines Fußballspiel für unser Handy haben.

Dazu erstellen wir eine App, bei der wir Elfmeter schießen können. Es gibt einen Torwart, der sich in eine zufällige Richtung bewegt und versucht, den Ball zu halten.

Später werden wir diese App noch erweitern und verschönern.



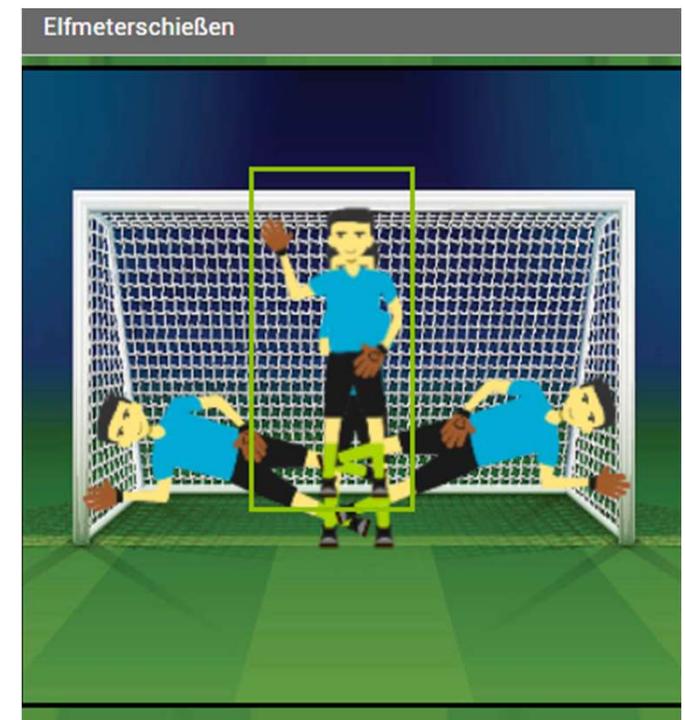
Teil 1: Ball und Torwart bewegen

In diesem Teil erstellen wir das Spielfeld. Wir programmieren die Tasten so, dass wir den Ball in eine Richtung schießen können.

Der Torwart bewegt sich nach dem Schuss in eine zufällig gewählte Richtung.

Anleitung (1)

1. Lade die Datei *Elfmeter.zip* herunter und entpacke sie.
2. Richte die Benutzeroberfläche und den Hintergrund ein:
 - a. Lege ein neues Projekt an. Nenne das Projekt *Elfmeter* oder so ähnlich.
 - b. Setze den Titel von *Screen1* auf *Elfmeterschießen* und das *HintergrundBild* auf *Rasen_unten.png*
 - c. Platziere eine Zeichenfläche auf dem *Screen1*. Setze die Eigenschaften der Zeichenfläche so: *Höhe=280 Pixel*, *Breite = Fuelle alles...*, *HintergrundBild=Tor_Rasen.png*.
3. Platziere die Zeichenanimationen für die Torhüter:
 - a. Schiebe eine *Zeichenanimation* auf die Zeichenfläche. Die Animation soll *TorwartStehend* heißen und das Bild *torwart_stehend.png* als Bild haben.
 - b. Richte die Zeichenanimation so aus, dass der Torwart mitten im Tor steht.
 - c. Platziere drei weitere Zeichenanimationen *TorwartLinks*, *TorwartMitte*, *TorwartRechts* mit den entsprechenden Bildern auf der Zeichenfläche. (Wir werden die Blöcke später so programmieren, dass immer nur eines der vier Torwartbilder zu sehen ist).



Anleitung (2)

4. Erstelle die Tasten für die Schussrichtung:

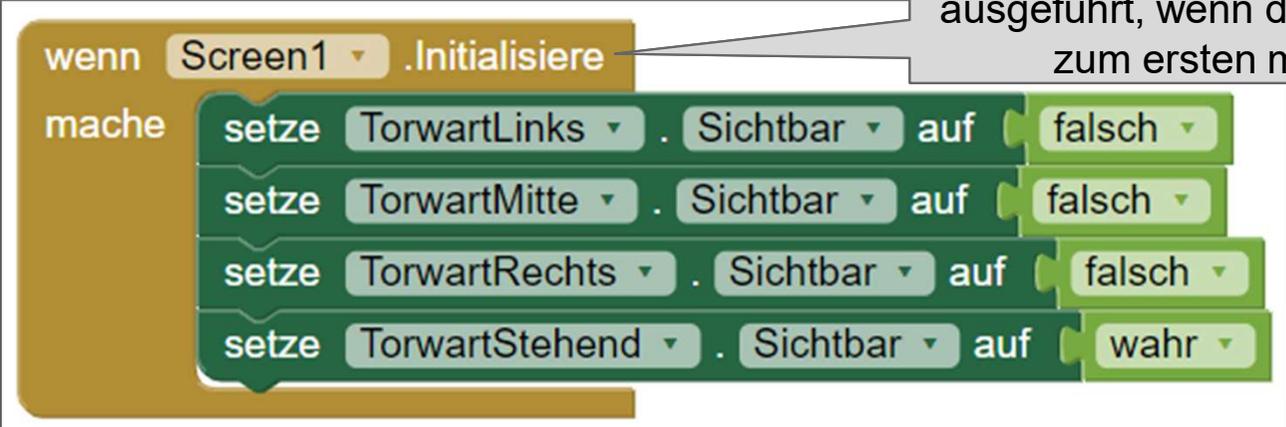
- Platziere unter *Zeichenfläche1* eine *HorizontaleAusrichtung* (*AusrichtungHorizontal= Mitte: 3, HintergrundFarbe=weiß, Breite=Fuelle alles...*)
- Platziere auf der *HorizontaleAusrichtung* drei Tasten mit den Namen *TasteLinks*, *TasteOben* und *TasteRechts*. Setze *Höhe* und *Breite* aller Tasten jeweils auf *40 pixel*. Setze danach auf alle Tasten als Bild den jeweils passenden Pfeil.
- Tipp: Wenn du zwischen den Tasten jeweils ein *Bezeichnung*-Feld ohne Text mit *Breite 20 Pixel* platzierst, führt das zu einem kleinen Abstand zwischen den Tasten..



5. Als letztes platzieren wir noch den Ball als *ZeichenAnimation* mit dem Namen *Fußball* und dem entsprechenden Bild mittig vor dem Tor. *Breite* und *Höhe* des Balls sind jeweils *30 pixel*.

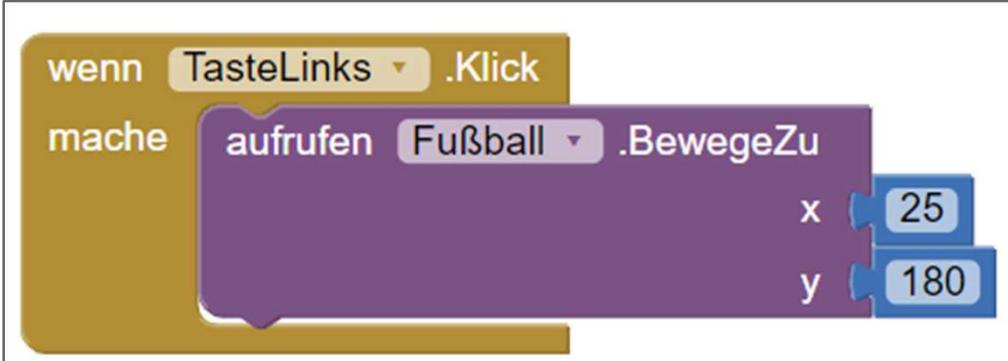
Anleitung (3)

6. Wechsle in die Ansicht Blöcke und programmiere zuerst, was die App beim Starten machen soll. Hier kannst du die Torhüter verstecken, die gerade nicht aktuell sind. Das geht so:



The screenshot shows a Scratch script for the 'wenn Screen1 .Initialisiere' event. The script is enclosed in a 'mache' loop and contains four 'setze' blocks. The first three blocks set the 'Sichtbar' property of 'TorwartLinks', 'TorwartMitte', and 'TorwartRechts' to 'falsch'. The fourth block sets the 'Sichtbar' property of 'TorwartStehend' to 'wahr'. A callout box points to the event block with the text: 'Das Ereignis Initialisiere wird ausgeführt, wenn der Screen1 sich zum ersten mal öffnet.'

7. Erstelle für alle Tasten Blöcke, die das Klick-Ereignis behandeln. Wenn eine Taste geklickt wird, soll sich der Ball in die passende Richtung bewegen: Wenn *TasteLinks* geklickt wird, soll der Ball zu Position $x=25$, $y=180$. Bei *TasteRechts*: $x=250$, $y=180$. Bei *TasteOben*: $x=125$, $y=60$.



The screenshot shows a Scratch script for the 'wenn TasteLinks .Klick' event. The script is enclosed in a 'mache' loop and contains one 'aufrufen Fußball .BewegeZu' block. The 'x' coordinate is set to 25 and the 'y' coordinate is set to 180.

Anleitung (4)

8. Als nächstes müssen wir uns merken, welche Richtung angeklickt wurde. Lege dazu eine globale Variable *RichtungBall* an.

global initialisieren `RichtungBall` auf `0`

Wir speichern die Richtung als Zahlen, und zwar so:

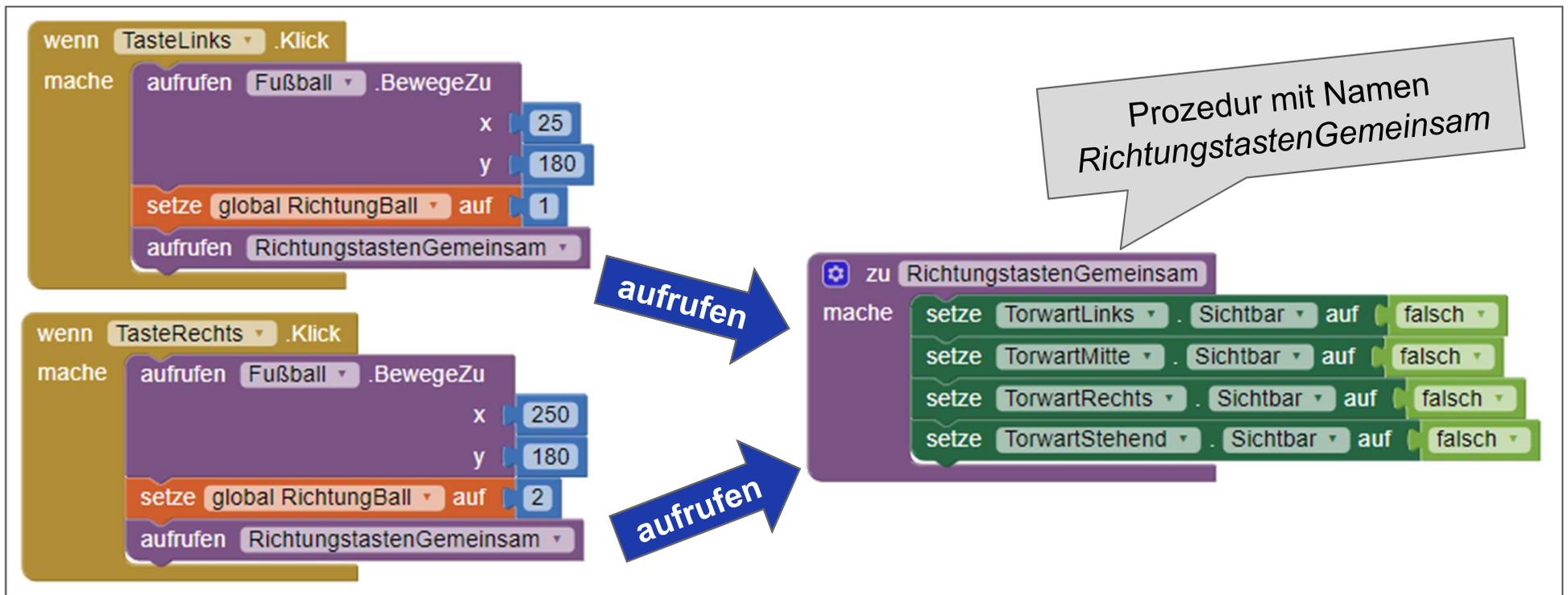
Richtung	Zahl
links	1
mitte	2
rechts	3

Dementsprechend müssen wir in die Klick-Behandler der *TasteLinks* einen Block einfügen, der *global RichtungBall* auf 1 setzt. Für *TasteOben* setzen wir *global RichtungBall* auf 2, für *TasteRechts* auf 3.

Gut zu wissen: Prozeduren

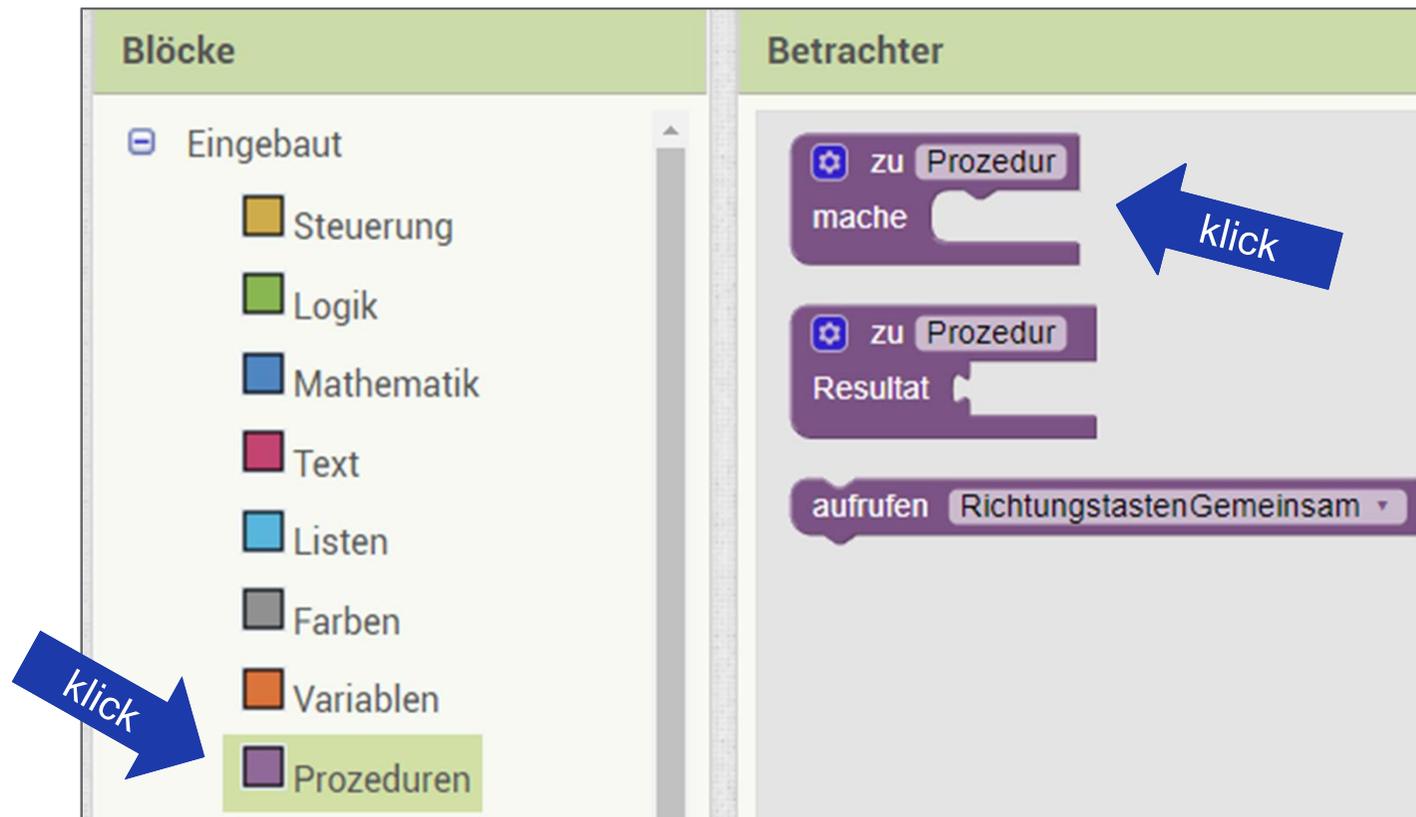
Nach jedem Tastendruck soll sich der Torwart in eine Richtung bewegen. Das ist für alle drei Tasten genau gleich. Damit wir die Blöcke nicht für jede Taste kopieren müssen, legen wir eine **Prozedur** an.

Prozeduren fassen Blöcke zusammen. Jede Prozedur hat einen Namen. Über den Namen kann man die Prozedur dann an anderen Stellen aufrufen.



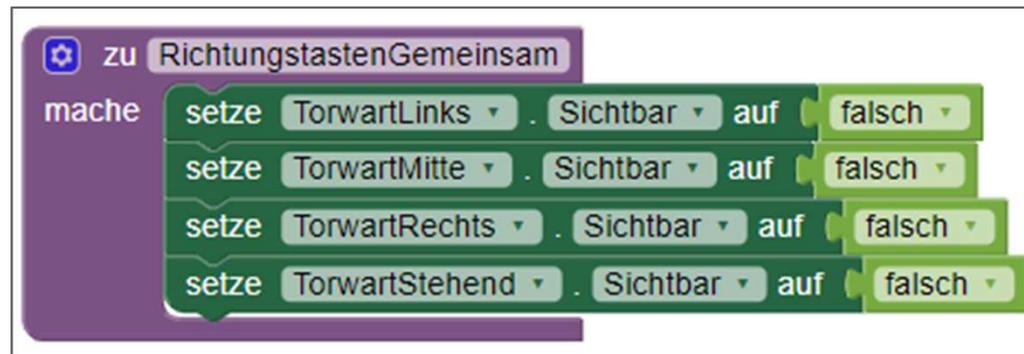
Anleitung (5)

9. Lege jetzt eine Prozedur an und nenne sie *RichtungstastenGemeinsam*.



Anleitung (6)

10. Füge in die Prozedur Blöcke ein, die erstmal alle Torhüter verschwinden lassen:



10. Erzeuge jetzt in der Prozedur eine zufällige Zahl zwischen 1 und 3. Das ist dann die Richtung, in die sich der Torwart bewegt. Die Zahl merken wir uns in einer globalen Variablen.

- Lege die globale Variable *RichtungTorwart* an und initialisiere Sie mit 0.
- Füge in die Prozedur *RichtungstastenGemeinsam* einen Block ein, der eine Zufallszahl erzeugt und in die globale Variable *RichtungTorwart* schreibt.



Anleitung (7)

Richtung	Zahl
links	1
mitte	2
rechts	3

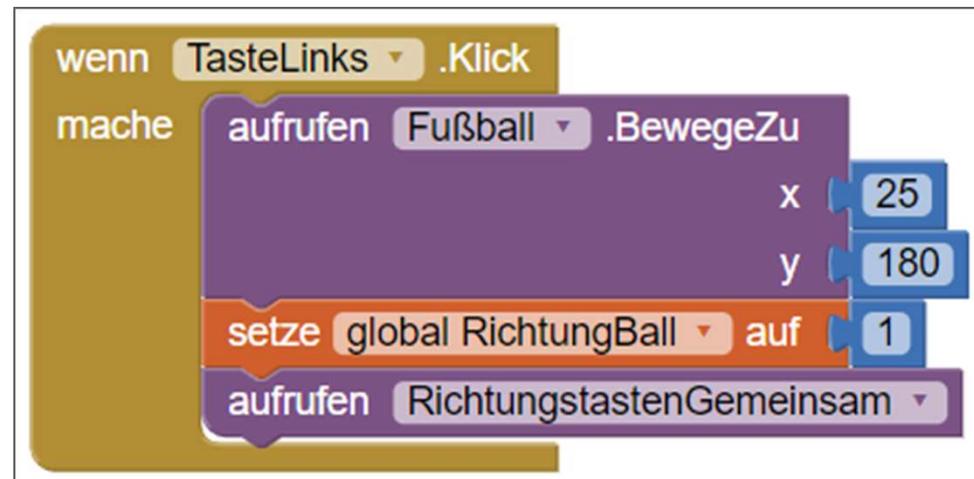
11. Bisher sind ja alle Torhüter-Bilder noch unsichtbar. Mache jetzt das Bild sichtbar, das zum zufällig erzeugten Wert von *RichtungTorwart* passt. Wenn *RichtungTorwart* gleich 1 ist, setzt du also *TorwartLinks* auf sichtbar, usw..

Füge dazu in der Prozedur *RichtungstastenGemeinsam* folgendes ein:

```
wenn (hole global RichtungTorwart) = 1  
dann setze TorwartLinks . Sichtbar auf wahr  
wenn (hole global RichtungTorwart) = 2  
dann setze TorwartMitte . Sichtbar auf wahr  
wenn (hole global RichtungTorwart) = 3  
dann setze TorwartRechts . Sichtbar auf wahr
```

Anleitung (8)

12. Rufe jetzt die Prozedur *RichtungstastenGemeinsam* in den Klick-Behandlern der drei Tasten auf:



13. Fertig! Lade die App herunter. Jetzt müssten sich Ball und Torwart bewegen.

Teil 2: Rücksetzen und Punkte zählen

Im zweiten Teil sollen der Torwart und der Ball vor jedem Schuss wieder auf die Ausgangsposition zurückgesetzt werden.

Außerdem sollen Punkte gezählt werden.

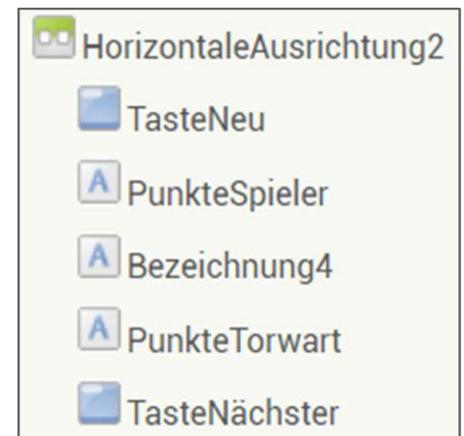
Anleitung (1)

Neu

0 : 0

Nächster

1. Wir brauchen jetzt Tasten für Ausgangsposition und Neustart. Außerdem brauchen wir eine Anzeige für den Spielstand. Erstelle diese Elemente wie folgt:
 - a. Füge unter der *HorizontaleAusrichtung1* eine *Bezeichnung* ohne Text ein. Das sorgt für einen kleinen Abstand zu den Richtungstasten.
 - b. Füge unter der Bezeichnung eine weitere *HorizontaleAusrichtung* ein.
(*AusrichtungHorizontal = mitte, AusrichtungVertikal=mitte, HintergrundFarbe=weiss, Breite = Fuelle alles...*)
 - c. Platziere in der neuen *HorizontalenAusrichtung* folgende Elemente:
 - i. eine Taste *TasteNeu* (*Schriftgröße=18, Breite=Fuelle alles..., Höhe=Fuelle alles..., Form=rechteckig*)
 - ii. eine Bezeichnung *PunkteSpieler* (*Schriftgröße=30, Breite=30 pixel, HasMargins=abgehakt, TextAusrichtung=rechts*)
 - iii. eine Bezeichnung (*Text=:, Schriftgröße=30, Breite=30 pixel, HasMargins=abgehakt, TextAusrichtung=mitte*)
 - iv. eine Bezeichnung *PunkteTorwart* (*Schriftgröße=30, Breite=30 pixel, HasMargins=abgehakt, TextAusrichtung=links*)
 - v. eine Taste *TasteNächster* (*Schriftgröße=18, Breite=Fuelle alles..., Höhe=Fuelle alles..., Form=rechteckig*)



Anleitung (2)

- Die Taste *Nächster* soll den Ball auf den Ausgangspunkt zurücklegen und den *TorwartStehend* wieder einblenden. Mache alle anderen Torwartbilder wieder unsichtbar.



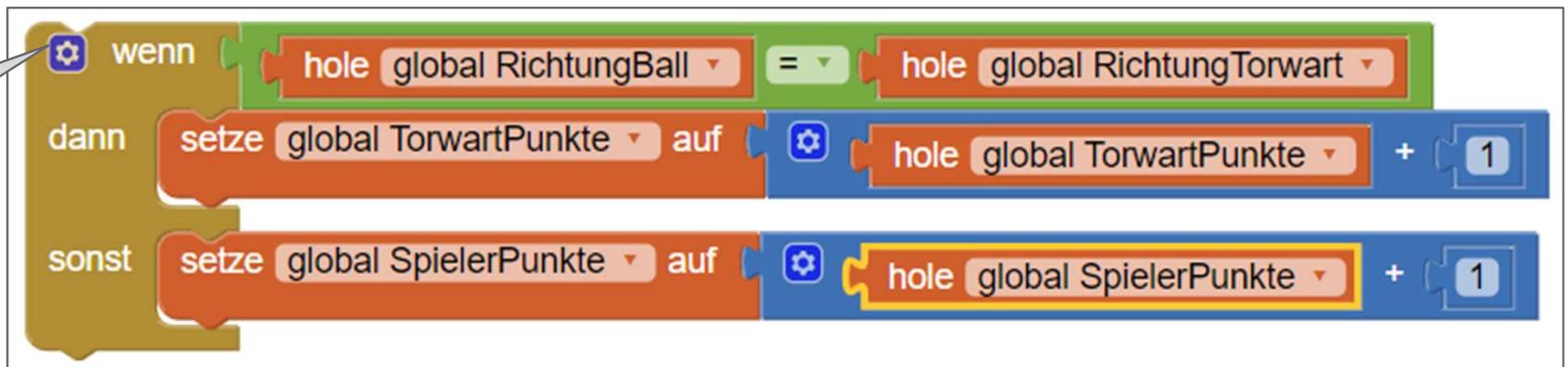
```
wenn TasteNächster .Klick
  mache
    setze TorwartLinks .Sichtbar auf falsch
    setze TorwartMitte .Sichtbar auf falsch
    setze TorwartRechts .Sichtbar auf falsch
    setze TorwartStehend .Sichtbar auf wahr
    aufrufen Fußball .BewegeZu
      x 140
      y 260
```

The image shows a Scratch code block for a button click event. The event is triggered when the 'TasteNächster' button is clicked. The code block contains a 'make' loop with four 'setze' blocks and one 'aufrufen' block. The 'setze' blocks set the 'Sichtbar' property of 'TorwartLinks', 'TorwartMitte', and 'TorwartRechts' to 'falsch', and 'TorwartStehend' to 'wahr'. The 'aufrufen' block calls the 'BewegeZu' method of the 'Fußball' object, with 'x' set to 140 and 'y' set to 260.

Anleitung (3)

3. Als nächstes zählen wir die Punkte für den Schützen und den Torhüter. Wenn nach dem Schuss die *RichtungTorwart* mit der *RichtungBall* übereinstimmt, hat der Torwart den Ball gehalten und bekommt einen Punkt. Wenn der Torwart in die falsche Richtung gesprungen ist, bekommt der Schütze einen Punkt.
 - a. Lege zwei globale Variablen *TorwartPunkte* und *SpielerPunkte* an. Initialisiere beide Variablen mit dem Wert 0.
 - b. In der Prozedur *RichtungstastenGemeinsam* prüfen wir am Ende, ob *RichtungTorwart* und *RichtungBall* gleich sind. Falls ja, bekommt der Torwart einen Punkt dazu. Falls nicht, bekommt der Spieler einen Punkt dazu.

Nimm einen **wenn**-Block und klicke auf das Zahnrad, um das **sonst** hinzuzufügen.



Anleitung (4)

- c. Wenn die Punkte des Spielers bzw. des Torwarts erhöht wurden, müssen wir das noch anzeigen. Setze dazu den Text der *Bezeichnungen PunkteSpieler* und *PunkteTorwart* auf den jeweiligen Wert.



3. **Fertig!** Lade die App jetzt runter und probiere sie aus. Jetzt müssten die Punkte gezählt werden.

Teil 3: Spielende und Neustart

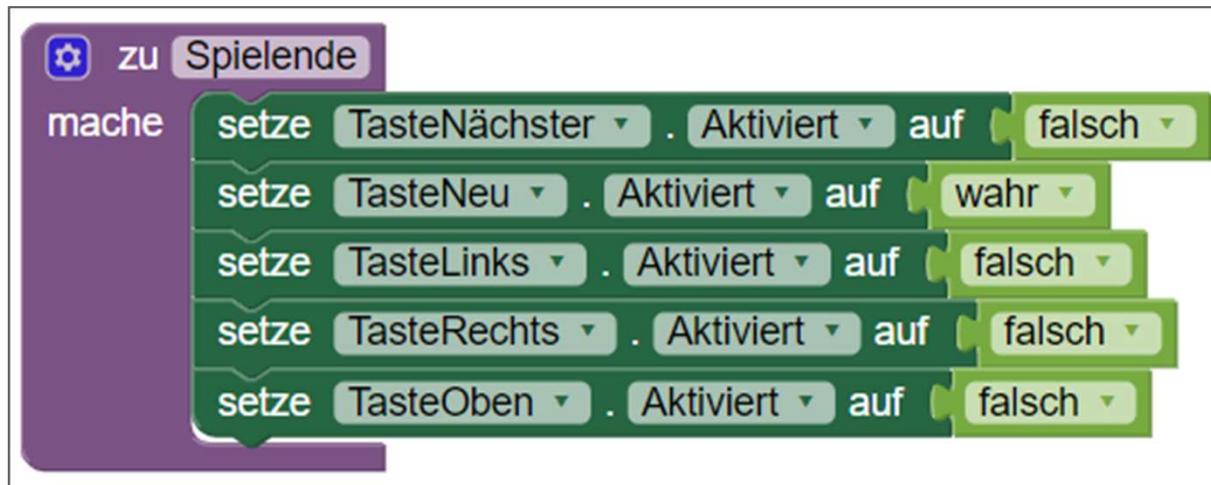
Jetzt soll das Spiel beendet werden, wenn der Spieler oder der Torwart fünf Punkte haben.

Wenn auf die Taste *Neu* geklickt wird, soll ein neues Spiel starten.

Anleitung (1)

Jetzt beenden wir das Spiel, wenn der Spieler oder der Torwart fünf Punkte haben. Um das Spiel zu beenden, deaktivieren wir die Richtungstasten und die *TasteNeu*. Deaktivierte Tasten kann man nicht mehr anklicken. Dann kann man keine neuen Schüsse mehr machen.

1. Erzeuge eine neue Prozedur *Spielende*.



Anleitung (2)

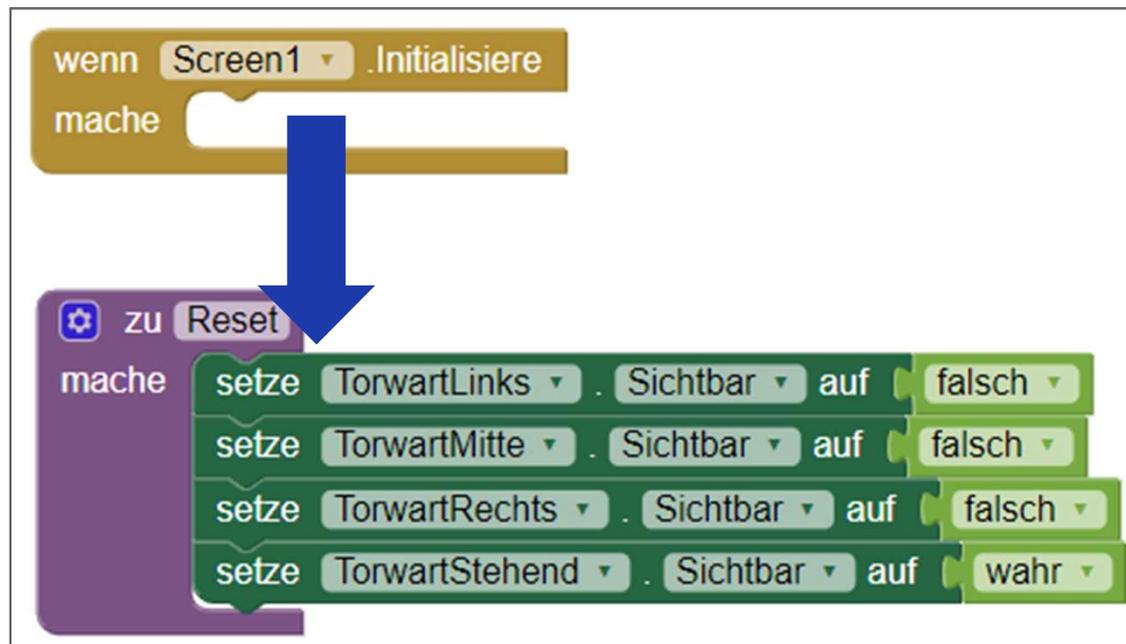
2. Füge nach dem Hochzählen der Punkte in der Prozedur *RichtungstastenGemeinsam* eine Überprüfung ein. Wenn der Spieler oder der Torwart fünf Treffer haben, soll die Prozedur *Spielende* aufgerufen werde.



3. Jetzt verlangen wir noch, dass die *TasteNeu* nach jedem Schuss gedrückt werden muss, bevor der nächste Schuss kommt. Mache dazu folgendes:
 - a. In der Prozedur *RichtungstastenGemeinsam*: *TasteLinks*, *TasteRechts*, *TasteOben* deaktivieren, *TasteNächster* aktivieren.
 - b. In *wenn TasteNächster.Klick*: *TasteLinks*, *TasteRechts*, *TasteOben* aktivieren, *TasteNächster* deaktivieren.

Anleitung (3)

3. Programmiere als letztes noch das Verhalten der *TasteNeu*. Die Taste soll ein neues Spiel starten. Weil beim Start und beim Neustart genau das gleiche gemacht werden muss, erstellen wir auch hier wieder eine Prozedur.
 - a. Erstelle eine neue Prozedur *Reset*.
 - b. Verschiebe den Inhalt von *wenn Screen1.Initialisiere* in die Prozedur *Reset*.



Anleitung (4)

c. Erweitere die Prozedur *Reset*, sodass sie so aussieht:

The image shows a Scratch code block for a procedure named 'Reset'. The code is as follows:

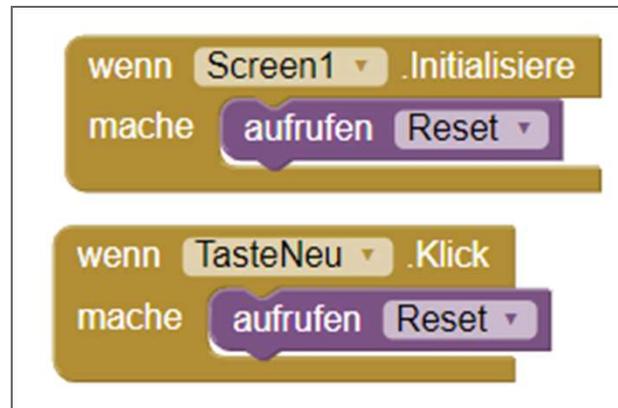
```
zu Reset
mache
  setze TorwartLinks . Sichtbar auf falsch
  setze TorwartMitte . Sichtbar auf falsch
  setze TorwartRechts . Sichtbar auf falsch
  setze TorwartStehend . Sichtbar auf wahr
  aufrufen Fußball . BewegeZu
    x 140
    y 260
  setze TasteLinks . Aktiviert auf wahr
  setze TasteRechts . Aktiviert auf wahr
  setze TasteOben . Aktiviert auf wahr
  setze TasteNächster . Aktiviert auf falsch
  setze global SpielerPunkte auf 0
  setze global TorwartPunkte auf 0
  setze PunkteSpieler . Text auf hole global SpielerPunkte
  setze PunkteTorwart . Text auf hole global TorwartPunkte
```

Explanatory callouts on the right side of the code:

- Nur Bild für Torwart in Ausgangsposition anzeigen (points to the visibility settings for the goalkeeper).
- Ball auf Ausgangsposition (points to the 'BewegeZu' block).
- Schusstasten aktivieren, Taste Nächster deaktivieren. (points to the 'Aktiviert' settings for the buttons).
- Spielstand auf 0:0 setzen. (points to the global variable settings for 'SpielerPunkte' and 'TorwartPunkte').
- Anzeige des Spielstands aktualisieren (points to the 'Text' settings for 'PunkteSpieler' and 'PunkteTorwart').

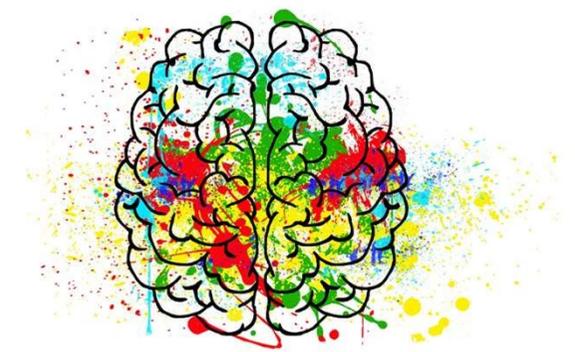
Anleitung (5)

- d. Rufe jetzt die Prozedur *Reset* jeweils auf, wenn die App gestartet wird (*wenn Screen1.Initialisiere*) und wenn die *TasteNeu* angeklickt wird (*wenn TasteNeu.Klick*).



- d. **Fertig!** Lade die App herunter.

Erweiterungen



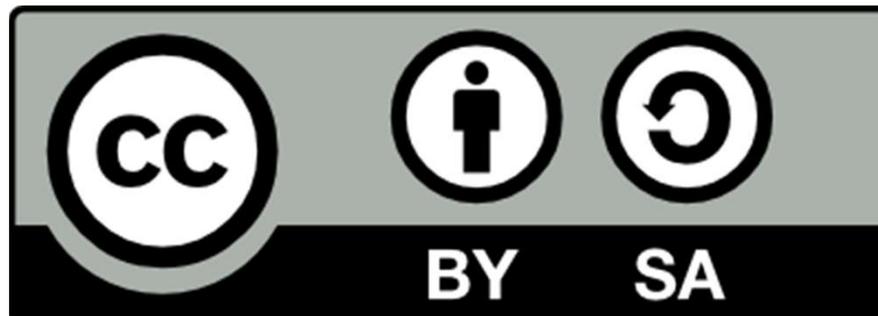
Funktion

- Spiele passende Töne ab, wenn der Schütze trifft und wenn der Torwart den Ball hält.
- Füge zusätzliche Schussrichtungen ein (z.B. links oben, rechts oben). Der Torwart soll sich dann auch an diese Positionen bewegen können.

Design

- Ändere die Farben und die Form der Tasten nach deinem Geschmack ab.
- Wähle für die Punkteanzeige eine andere Schriftart und ändere die TextFarbe und die HintergrundFarbe so, dass es aussieht wie eine klassische Anzeigetafel.
- Kopiere das Projekt und passe es durch die Wahl anderer Bilder auf eine andere Sportart an (z. B. Eishockey, Handball).

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