

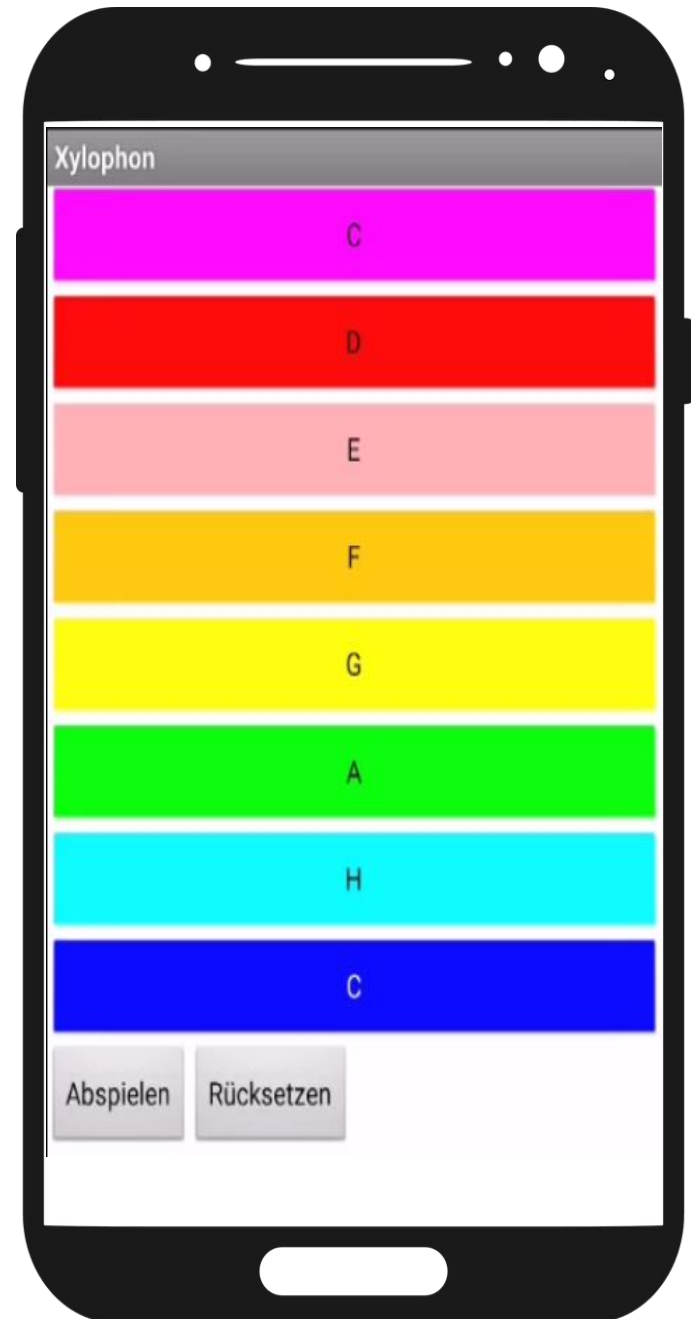
# **Projekt Xylophon**

# Ziel

**Wir möchten eine App, mit der wir Musik spielen und aufzeichnen können.**

Dazu erstellen wir eine App, wie wie ein Xylophon funktioniert. Es gibt Tasten, die Töne abspielen.

Außerdem soll sich die App merken, was wir gespielt haben und die Klangfolge automatisch abspielen können.



# **Teil 1: Töne abspielen**

Im ersten Schritt soll unser Xylophon für jede Taste einen Ton abspielen.

---

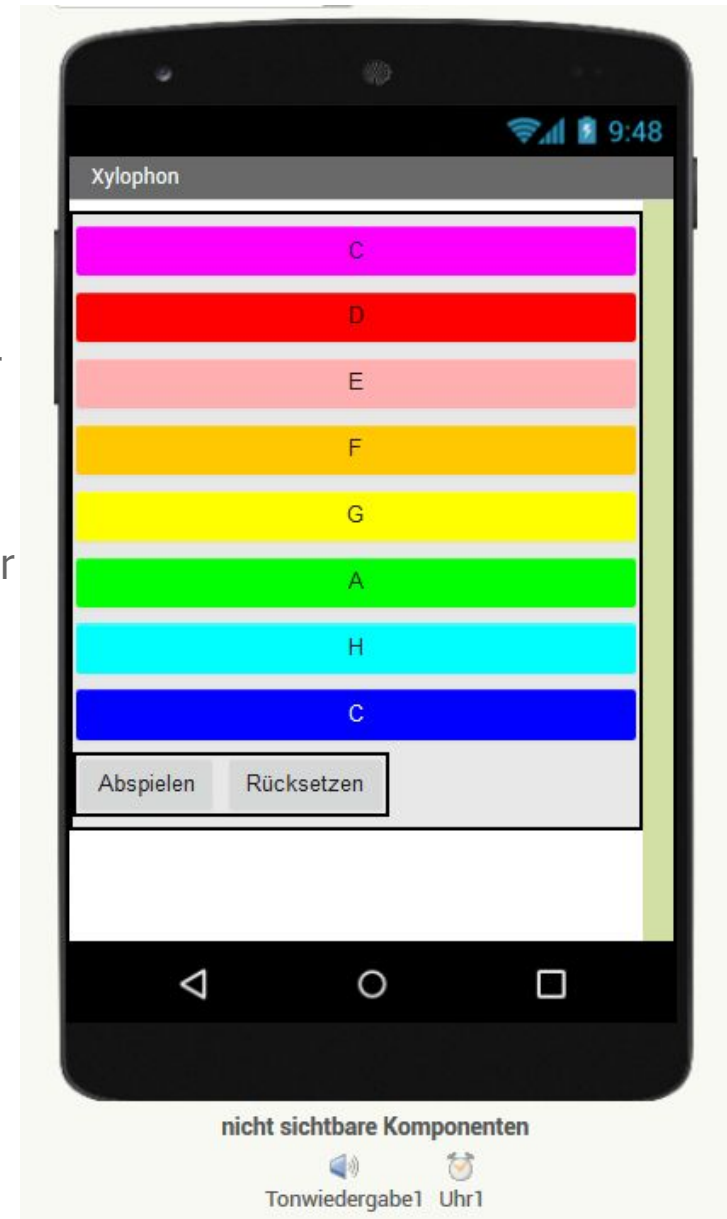
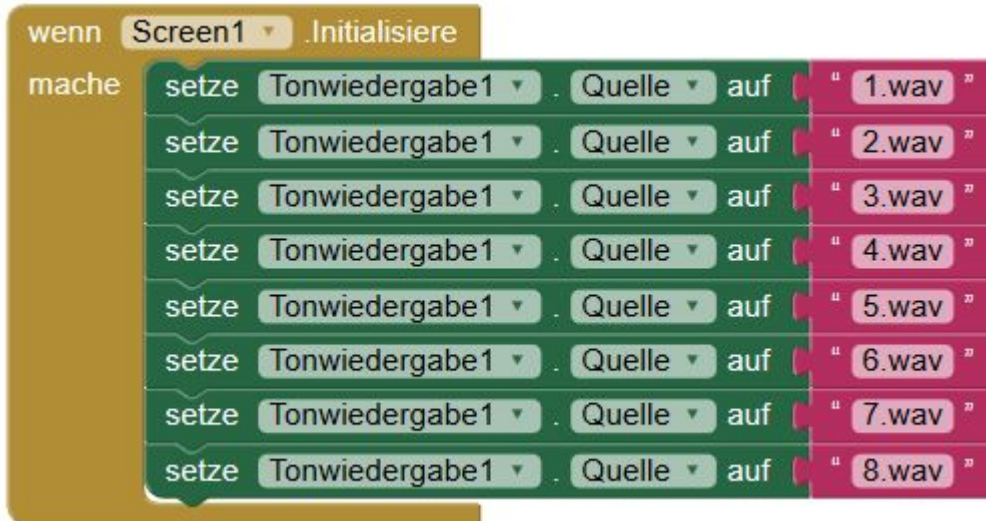
# Anleitung (1)

1. Erstelle ein neues Projekt.
2. Lade die Töne des Xylophons herunter. Es sind acht Töne:
  - <http://appinventor.org/bookFiles/Xylophone/1.wav>
  - <http://appinventor.org/bookFiles/Xylophone/2.wav>
  - <http://appinventor.org/bookFiles/Xylophone/3.wav>
  - <http://appinventor.org/bookFiles/Xylophone/4.wav>
  - <http://appinventor.org/bookFiles/Xylophone/5.wav>
  - <http://appinventor.org/bookFiles/Xylophone/6.wav>
  - <http://appinventor.org/bookFiles/Xylophone/7.wav>
  - <http://appinventor.org/bookFiles/Xylophone/8.wav>
3. Lade die Töne als Medien in das Projekt



# Anleitung (2)

- Erstelle eine Oberfläche, die so aussieht:  
Als nicht sichtbare Komponenten brauchen wir eine Tonwiedergabe und eine Uhr.
- Wechsle zu den Blöcken. Beim Initialisieren der App müssen wir zuerst alle Töne in die Tonwiedergabe laden.

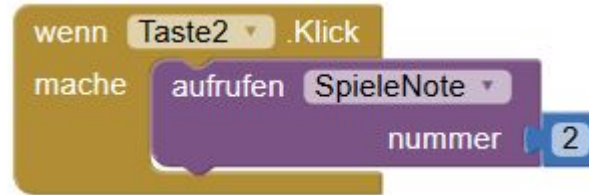


# Anleitung (3)

- Das Abspielen der Töne läuft immer gleich ab. Deswegen können wir dafür eine eigene Prozedur erstellen. Die Nummer des Tons geben wir der Prozedur als Parameter:



- Wenn die Ton-Tasten geklickt werden, rufen wir die Prozedur auf und sagen, welchen Ton wir abspielen wollen.



und so weiter...

- Fertig.** Lade die App herunter. Jetzt solltest du Musik spielen können.

# **Teil 2: Tonfolge merken**

Jetzt soll unsere App die Tonfolge aufzeichnen, damit wir sie wieder abspielen können.

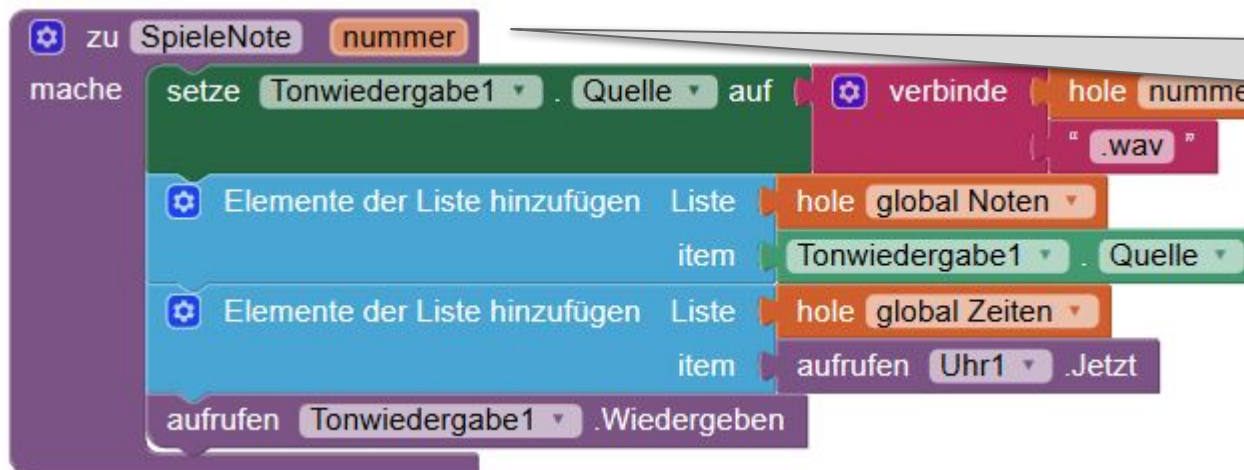
---

# Anleitung (1)

1. Um uns die gespielten Töne und die Pausen zwischen den Tönen zu merken, brauchen wir zwei Listen:



2. Erweitere jetzt die Prozedur *SpielerNote* so, dass die zu spielende Note und die aktuelle Zeit in die Listen eingefügt werden:



Die Prozedur gibt es schon. Wir müssen sie nur erweitern.



# Anleitung (2)

3. Wenn die Taste *Rücksetzen* geklickt wird, löschen wir beide Listen und vibrieren kurz:

```
wenn Rücksetzen .Klick
mache
  setze global Noten auf leere Liste erstellen
  setze global Zeiten auf leere Liste erstellen
  aufrufen Tonwiedergabe1 .Vibriere
  Millisecs 50
```

4. Um die gespeicherten Noten abzuspielen, brauchen wir eine Hilfsprozedur. Die Prozedur gibt zuerst einen Ton aus. Dann ruft sie sich selbst wieder auf, um den zweiten Ton auszugeben, usw.

```
global initialisieren zähler auf 0

zu NotenAbspielen
mache
  setze Tonwiedergabe1 .Quelle auf Listenelement wählen Liste
  hole global Noten
  Index
  hole global zähler
  aufrufen Tonwiedergabe1 .Wiedergeben
  wenn
    hole global zähler < Länge der Liste Liste
    hole global Noten
  dann
    setze global zähler auf hole global zähler + 1
    aufrufen NotenAbspielen
```

## Anleitung (3)

5. Wenn der Abspielen-Button geklickt wird, starten wir das Ganze.



6. **Fertig.** Jetzt müsste es schon tun. Du musst aber genau hinhören, wenn du etwas erkennen willst.

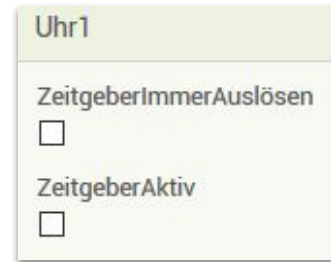
# Teil 3: Pausen einfügen

Unsere App spielt die Töne beim Playback bisher einfach nacheinander ab - unabhängig davon, ob wir Pausen gemacht haben oder nicht.

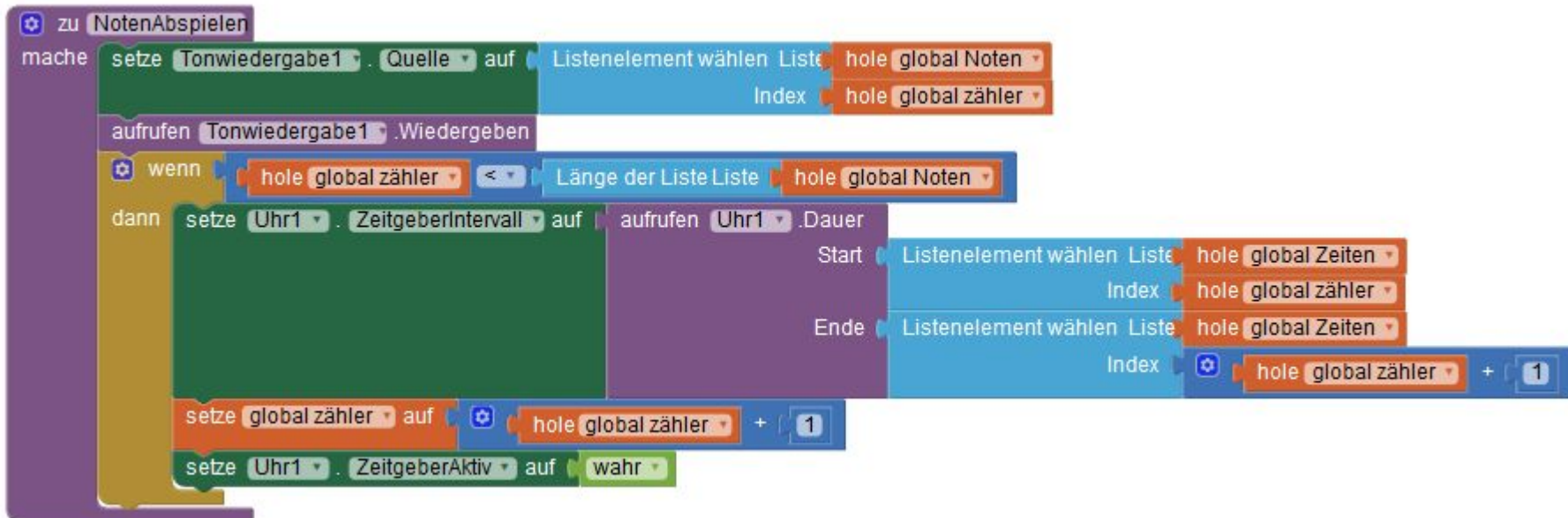
Jetzt bauen wir das so aus, dass auch Pausen berücksichtigt werden.

---

# Anleitung (1)



1. Jetzt möchten wir auch die Pausen drin haben, damit wir auch tatsächlich etwas hören. Dazu brauchen wir den Zeitgeber. Setze den Zeitgeber im Designer auf nicht aktiv (siehe oben).
2. Jetzt bauen wir die Prozedur *NotenAbspielen* so aus, dass sich die Prozedur nicht mehr selbst aufruft, sondern einen Zeitgeber mit der Zeit zwischen den Noten stellt. Erst, wenn der Zeitgeber auslöst, wird die nächste Note gespielt.



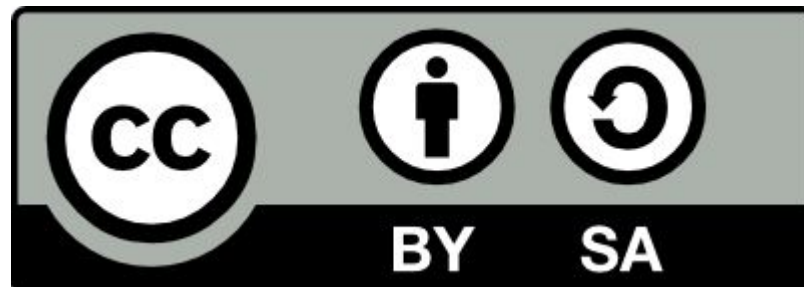
# Anleitung (2)

3. Der Zeitgeber-Aufruf sieht dann so aus:



4. Fertig. Jetzt müsste alles tun!

# 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: Prof. Dr. Markus Kaupp, Hochschule Esslingen

**Quellangaben:** [1] Eigener Screenshot, enthält einen Kartenausschnitt von [maps.google.de](https://maps.google.de), abgerufen am 17.02.2020