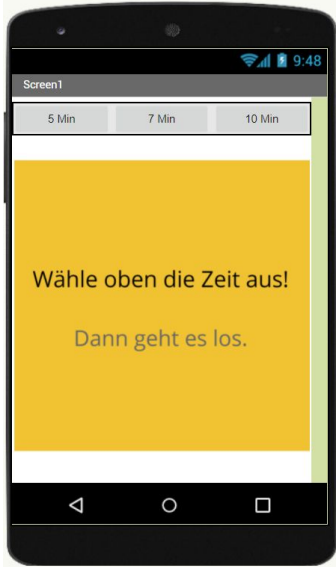




Du brauchst diese Komponenten:

- 1 HorizontalArrangement
- 3 Buttons
- 1 Image
- 2 Clocks
- 1 Sound

Tipps

Unsere Bilder und Klänge¹ findet ihr hier: appcamps.link/workout

Ihr könnt eine Komponente umbenennen, indem ihr rechts unter der Liste auf "Rename" klickt.

Aufgabe

1. Gebt appinventor.mit.edu in den Browser ein und klickt oben links auf [Create App!](#).
2. Loggt euch ein und erstellt ein neues Projekt über "My Projects" → "Start new project".

Ihr befindet euch nun im Design.

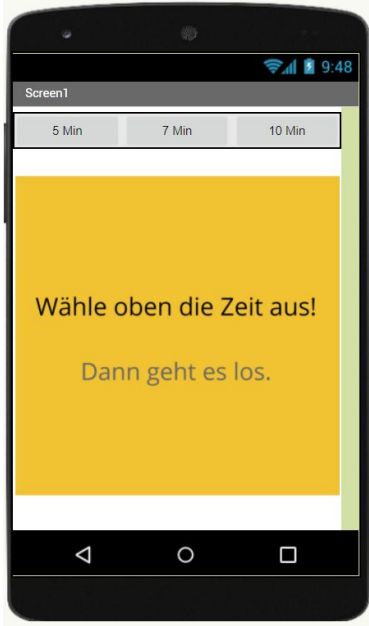
Designer Blocks

3. Ladet die Bilder und Sounds (siehe Tipp) unten rechts unter "Media" hoch.
4. Fügt dann alle Komponenten gemäß der Tabelle ein. Das Aussehen könnt ihr bestimmen.

Anzahl	Typ	Titel	Eigenschaften
1	Screen	Screen1	AlignHorizontal: <i>Center</i> : 3
1	Horizontal Arrangement	HorizontalArrangement1	Width: <i>Fill parent</i>
3	Button	Button5Min, Button7Min, Button10Min	Befinden sich im Arrangement, Width: <i>Fill parent</i> (alle Buttons), Text: <i>5 Min / 7 Min / 10 Min</i>
1	Image	Image1	Picture: <i>start.png</i> , Height: <i>Fill parent</i>
2	Clock (Sensors)	ClockGesamt, ClockUebung	Unsichtbare Komponenten, TimerEnabled: ☐
1	Sound (Media)	SoundUebungen	Unsichtbare Komponente

Nächste Aufgabe

Erstellt zwei Variablen und macht daraus Listen. Fügt die Dateinamen der Bilder in die eine Liste ein und die passenden Namen der Audios in die andere Liste an die gleichen Positionen.

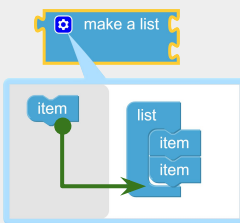


Tipps

Man initialisiert eine Variable mit diesem Block. Den Namen der Variablen könnt ihr dort eintragen.

initialize global **name** to

Erweitert die Liste, indem ihr auf das blaue Zahnrad klickt und das item von links in die Liste rechts zieht.



Aufgabe

Wechselt nun in die Programmierung.

Designer Blocks

1. Initialisiert zwei Variablen *Bilder* und *Audio* und macht daraus jeweils eine Liste.
2. Fügt in die Liste *Bilder* die Dateinamen der Bilder ein und in die Liste *Audio* die Namen aller Sounddateien, die ihr hochgeladen habt. Achtet darauf, dass die Übungen in den Listen in der gleichen Reihenfolge stehen.
3. Erstellt eine weitere Variable *Zaehler* und initialisiert sie mit 1. Diese Variable merkt sich, welche Übung gerade dran ist.

initialize global **Zaehler** to 1

initialize global **Bilder** to make a list

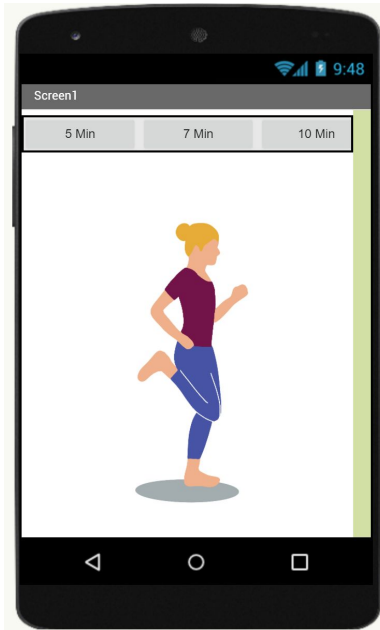
- "Anfersen.png"
- "Hampelmann.png"
- "Kniebeuge.png"
- "Kniehebelauf.png"
- "Plank.png"
- "Radfahren.png"
- "Sit-ups.png"
- "Strecksprung.png"

initialize global **Audio** to make a list

- "Anfersen.m4a"
- "Hampelmann.m4a"
- "Kniebeuge.m4a"
- "Kniehebelauf.m4a"
- "Plank.m4a"
- "Radfahren.m4a"
- "Sit-ups.m4a"
- "Strecksprung.m4a"

Nächste Aufgabe

Schreibe eine Prozedur "ZeigeUebung". Diese soll das Timer Intervall der *ClockUebung* setzen und sie einschalten. Zeige außerdem das aktuelle Bild an und spiele den passenden Sound. Nutze die Variable *Zaehler*, um dir die aktuelle Übung zu merken.



Tipps

Wenn ihr einen Block nicht in der Kategorie der Farbe findet, schaut auf der linken Seite unter der Komponente nach, für die der Block gelten soll.

Die Clocks erwarten immer Millisekunden. Eine Sekunde entspricht 1 000 Millisekunden. So könnt ihr auch andere Zeiten einstellen.

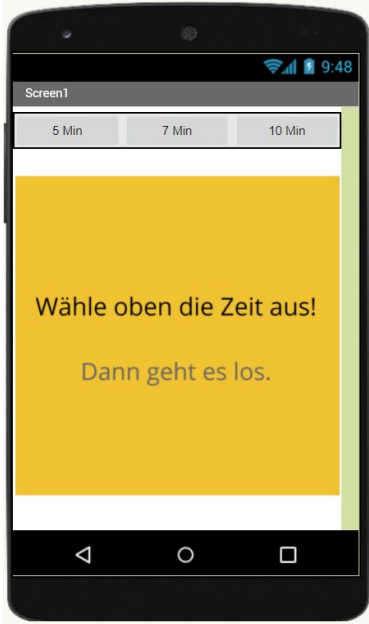
Aufgabe

1. Erstellt dann eine Prozedur *ZeigeUebung*.
2. Setzt darin zunächst das *TimerInterval* von der *ClockUebung* auf 5 000 ms. So wird jede Übung für 5 Sekunden angezeigt. Diese Zahl könnt ihr nach Belieben anpassen.
3. Anschließend müsst ihr noch den Timer der *ClockUebung* aktivieren. Setzt dafür *TimerEnabled* auf "true".
4. Zeigt dann im *Image1* das Bild aus der Liste *Bilder* an der Position des *Zaehlers* an.
5. Spielt außerdem das passende Audio ab. Setzt dazu zunächst die Quelle des *SoundUebungen* auf das Audio von der Position des *Zaehlers* und ruft dann "Play" auf.
6. Zählt zum Schluss den *Zaehler* hoch. Falls er das Ende der Liste übersteigt, setzt ihn auf 1.

```
to ZeigeUebung
do
  set ClockUebung . TimerInterval to 5000
  set ClockUebung . TimerEnabled to true
  set Image1 . Picture to select list item list get global Bilder
                           index get global Zaehler
  set SoundUebungen . Source to select list item list get global Audio
                                index get global Zaehler
  call SoundUebungen .Play
  set global Zaehler to get global Zaehler + 1
  if get global Zaehler > length of list list get global Bilder
  then set global Zaehler to 1
```

Nächste Aufgabe

Programmiert die Buttons, sodass sie das *Timer Interval* der *ClockUebung* auf die entsprechende Zeit setzen. Ruft anschließend jeweils die Prozedur *ZeigeUebung* auf.



Aufgabe

Designer Blocks

1. Wird der *Button5Min* angeklickt, setzt das *TimerInterval* der *ClockGesamt* auf 300 000 ms. Das sind fünf Minuten.
2. Setzt außerdem *TimerEnabled* auf true, damit die Uhr auch los zählt.
3. Ruft zum Schluss die selbstgeschriebene Prozedur *ZeigeUebung* auf, um die erste Übung anzuzeigen und somit das Workout zu starten.
4. Programmiert die beiden anderen Buttons genauso.
7 Minuten entsprechen 420 000 ms und 10 Minuten sind 600 000 ms.

```
when Button5Min .Click
do
  set ClockGesamt . TimerInterval to 300000
  set ClockGesamt . TimerEnabled to true
  call ZeigeUebung
```

```
when Button7Min .Click
do
  set ClockGesamt . TimerInterval to 420000
  set ClockGesamt . TimerEnabled to true
  call ZeigeUebung
```

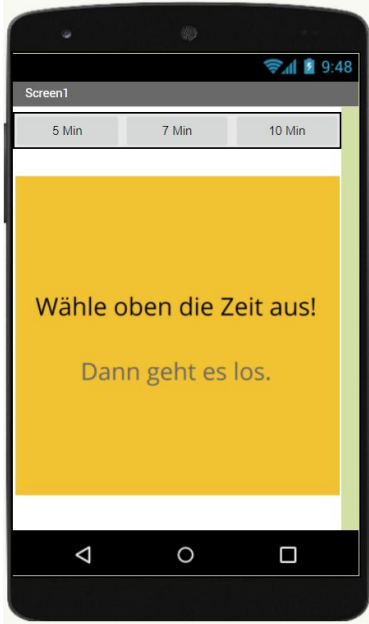
```
when Button10Min .Click
do
  set ClockGesamt . TimerInterval to 600000
  set ClockGesamt . TimerEnabled to true
  call ZeigeUebung
```

Tipp

Zum Testen könnt ihr kürzere Zeiten in die Blöcke eintragen. So müsst ihr nicht immer ganze 5 / 7 / 10 Minuten warten, bis ihr seht, ob eure Programmierung funktioniert.

Nächste Aufgabe

Wenn die *ClockUebung* abläuft, zeigt die nächste Übung an. Wenn die *ClockGesamt* abläuft, soll das Workout beendet werden und wieder das Startbild zu sehen sein.



Aufgabe

Designer Blocks

1. Läuft der Timer von der *ClockUebung* nach fünf Sekunden ab, ruft erneut eure Prozedur *ZeigeUebung* auf. So wird die nächste Übung angezeigt.
2. Wenn hingegen der Timer von der *ClockGesamt* abläuft, sind die fünf / sieben / zehn Minuten eures Workouts vorbei und ihr stoppt das gesamte Workout.
3. Setzt dafür *TimerEnabled* bei beiden Clocks auf false.
4. Zeigt außerdem im *Image1* als "Picture" wieder das "start.png" wie zu Anfang an.

```
when ClockUebung .Timer
do call ZeigeUebung
```

```
when ClockGesamt .Timer
do
  set ClockGesamt . TimerEnabled to false
  set ClockUebung . TimerEnabled to false
  set Image1 . Picture to "start.png"
```

Schon fertig?

Super! Dann testet die App doch gleich mal auf deinem Endgerät und startet das Workout!

Kennt ihr eigentlich die Vorteile von einer Workout-Routine?

1. Bewegung macht glücklich (mentale und körperliche Bewegung) ²
2. Disziplin hilft, Ziele zu erreichen, auch wenn es mal nicht so läuft oder sein muss ²
3. Sport sorgt für Balance zwischen Körper und Geist → steigert Konzentration ³

Und nun viel Spaß mit dem Workout in eurer App!