



# Image Recognition App Challenge

mit



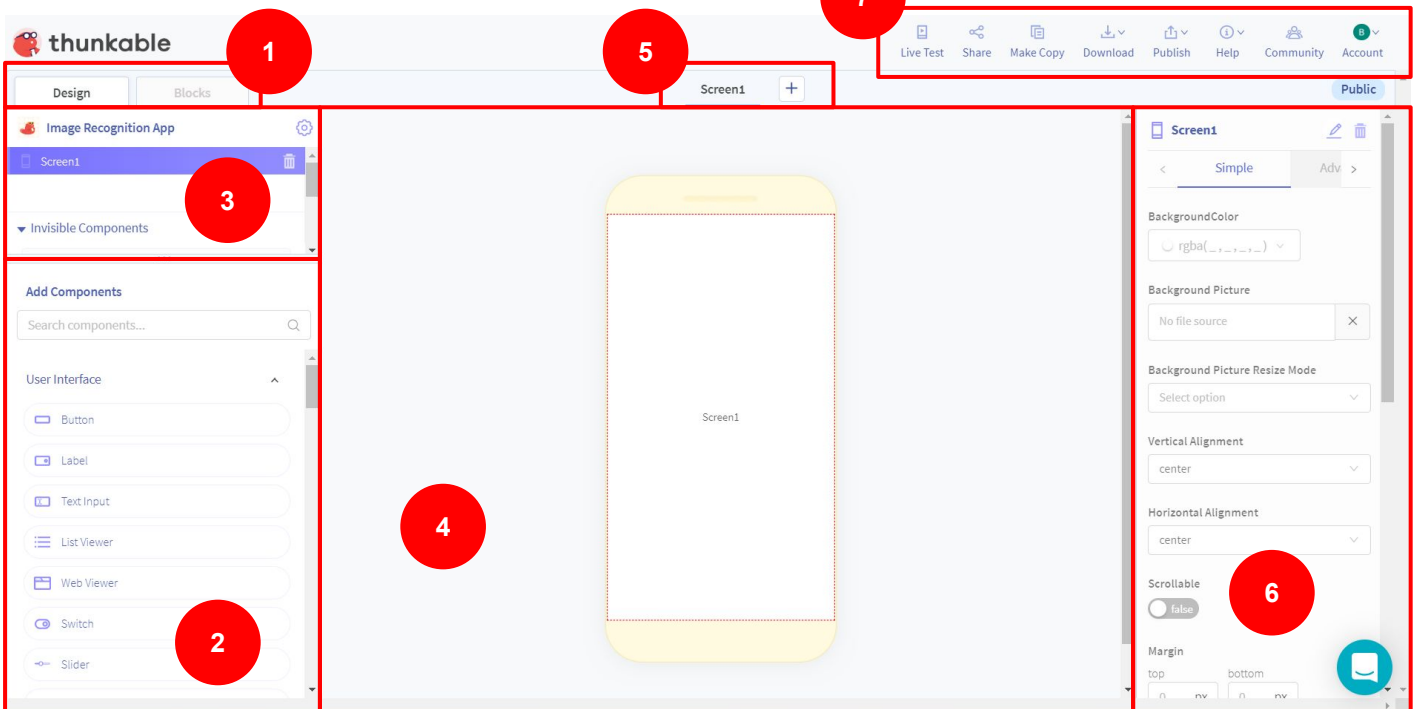
**thinkable**

## 1. Ein neues Projekt starten



Klicke auf “Start Building” oder “Create New App”, um ein neues App-Projekt zu starten. Gib anschließend einen Namen für dein Projekt (deine App) ein, z.B. “Image Recognition App” und klicke auf “Create”.

## 2. Das Interface von Thinkable



- (1) Diese zwei Tabs sind wichtig. Mit Ihnen kannst du zwischen der **visuellen Ansicht (Design)** und der **Ansicht zum Programmieren (Blocks)** wechseln.
- (2) Das hier ist die **Übersicht aller Komponenten**, die du zur App hinzufügen kannst, es gibt z.B. **sichtbare** User Interface-Elemente wie Buttons oder Text (Labels), aber auch **unsichtbare** Elemente wie “Text-to-Speech”, “Kamera” oder “Image Recognition”. Es gibt auch Layout-Elemente mit denen du die Anordnung der Elemente steuern kannst.
- (3) **Komponentenstruktur:** Hier findest du alle **Komponenten, die du bereits zur App hinzugefügt hast**. Am Anfang gibt es nur eine Komponente: den leeren Bildschirm “Screen1”. Hier kannst du auch die Reihenfolge der Komponenten und ihre Struktur (ähnlich einer Ordnerstruktur) festlegen.
- (4) Hier siehst du eine **visuelle Vorschau** des aktuellen Screens. Achtung: das hier ist noch **keine Simulation** der App, sondern nur eine Vorschau der Elemente in der App.
- (5) Apps bestehen meistens aus mehreren **Screens**. Hier kannst du **weitere Screens hinzufügen** und siehst, welcher Screen aktuell gerade angewählt ist.
- (6) Hier siehst du die **Properties (Eigenschaften)** der in der Komponentenübersicht (3) aktuell ausgewählten Komponenten. Hier lässt sich alles mögliche einstellen - z.B. Schriftfarben, Hintergrundfarben, Abstände, Größe und vieles mehr...
- (7) **Menüpunkte**, u.a. zum **Live-Test** (Simulation) in der App “Thinkable Live” sowie Veröffentlichung und Download der App.

### 3. Ein Hintergrundbild einfügen

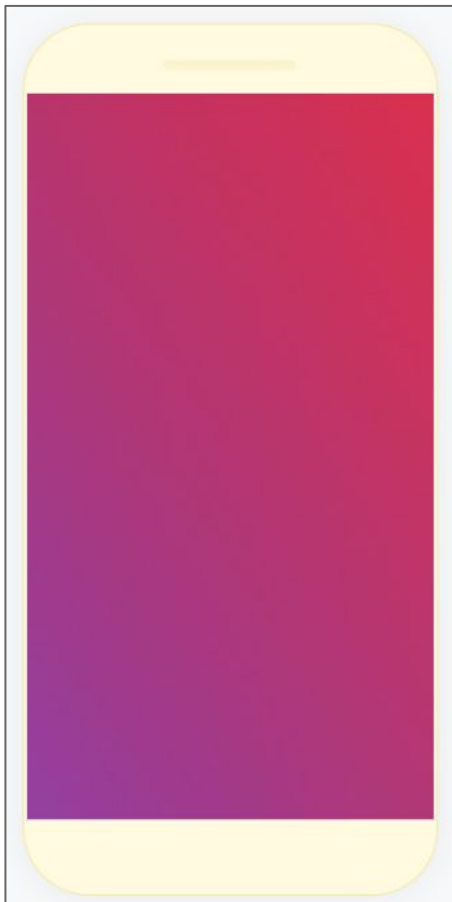
- Als erstes wollen wir nun ein Hintergrundbild einfügen. Öffne dazu als erstes die Seite [www.unsplash.com](http://www.unsplash.com) und klicke dort auf ein Bild, das dir gefällt. Lade das Bild anschließend mit der rechten Maustaste (**Bild speichern unter...**) herunter. Speichere es in den Download-Ordner deines PCs.
- Klicke nun auf **Screen1** in der **Komponentenstruktur**. In den **Properties** klicke auf das Textfeld unter **Background Picture**:



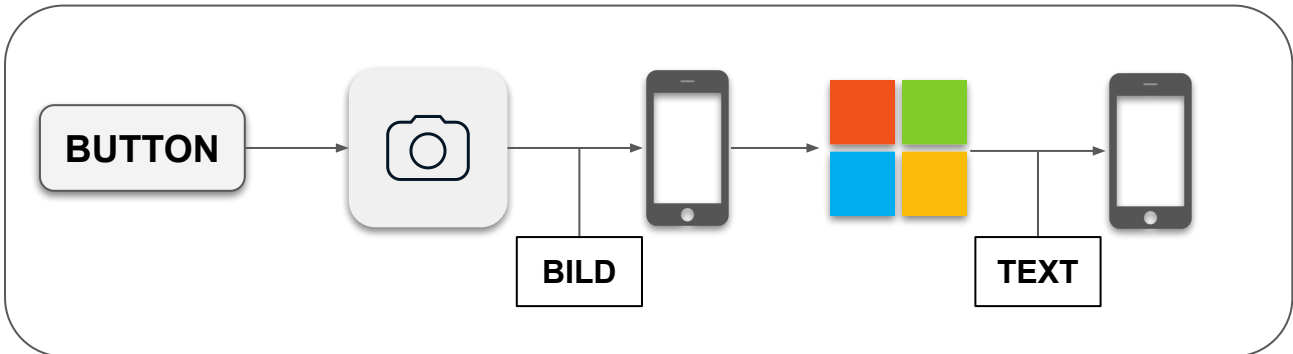
- Klicke auf **“Upload files”** und doppelklicke auf das Bild, das du vorhin im **Download-Ordner** gespeichert hast. Für die Property **“Background Picture Resize Mode”** wähle **“Cover”**, damit der ganze Screen mit dem Hintergrundbild bedeckt ist:



- Das **Hintergrundbild** sollte nun in der **App-Vorschau** zu sehen sein und den ganzen Screen bedecken:



### 3. Die Komponenten und wie sie zusammenhängen

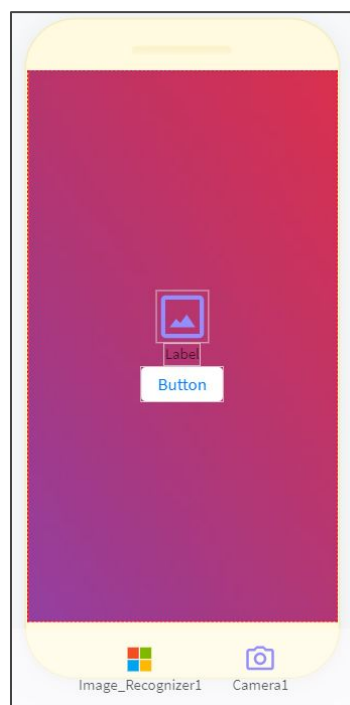
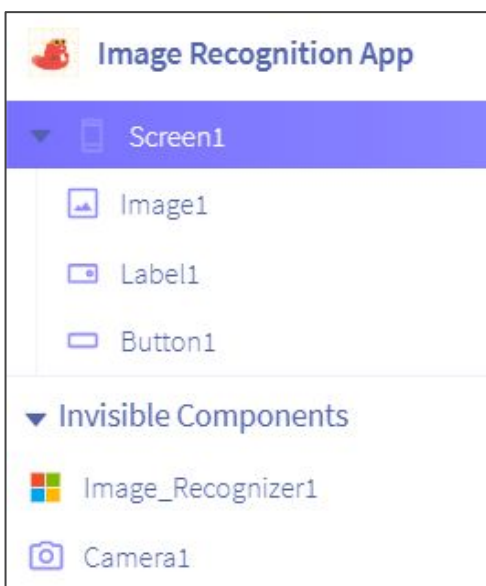


Wir wollen eine App programmieren, die Folgendes kann:

- Wenn man auf einen bestimmten **Button** drückt, soll sich die **Kamera** öffnen.
- Wenn ein **Foto** gemacht wird, soll das Foto als **Bild in der App** angezeigt werden.
- Das Foto soll an einen **Bilderkennungsdienst** (von *Microsoft Cognitive Services*) weitergeleitet werden.
- Der **Inhalt**, den der Bilderkennungsdienst im Foto erkennt, soll in der App **als Text (Label)** angezeigt werden.

Daher müssen wir folgende **Komponenten** in unsere App (in die App-Vorschau) ziehen:


Sichtbare Komponenten	Unsichtbare Komponenten
<ul style="list-style-type: none"> <li>• <b>Image</b> (Bild)</li> <li>• <b>Label</b> (Text, der angezeigt wird)</li> <li>• <b>Button</b> (Knopf)</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Camera</b> (Kamera)</li> <li>• <b>Image Recognizer</b> (Bilderkennung)</li> </ul>



Wie wir sehen, kleben die Komponenten derzeit noch ziemlich aneinander - die Abstände passen nicht. Außerdem lässt sich der Text im Label derzeit noch nicht besonders gut lesen.

Im nächsten Schritt werden wir uns daher die **Properties** (Eigenschaften) der einzelnen Komponenten ansehen und die **Abstände, Größen, Farben usw.** einstellen.

## 4. Die Komponenten stylen

Um die Übersicht zu behalten, ist es immer gut, die Komponenten zu benennen. Dazu gibt es in den Properties, das **Bleistift-Icon**: 

Benenne die Komponenten folgendermaßen um:

- Screen1 => **Homescreen**
- Image1 => **Foto**
- Label1 => **Beschreibungstext**
- Button1 => **Startbutton**

Nun können wir beginnen die Komponenten zu stylen - hier einige **Beispiele**, was du alles verändern kannst, gerne kannst du auch andere Sachen ausprobieren:

### Foto

Height

Relative Size (e.g. "50%")

60%

Width

Relative Size (e.g. "50%")

80%

Visible

true

### Beschreibungstext

Text

Noch kein Inhalt erkannt...

Font Size

18

Color

rgba(0, 0, 0, 1)

BackgroundColor

rgba(255, 255, 255, 1)

### Startbutton

Text

Mach ein Foto

Text Color

rgba(208, 2, 27, 1)

BackgroundColor

rgba(255, 255, 255, 1)

Font Size

20

Width

Relative Size (e.g. "50%")

80%

So ähnlich könnte das **Ergebnis** nach dem Styling aussehen:



Width

Relative Size (e.g. "50%")

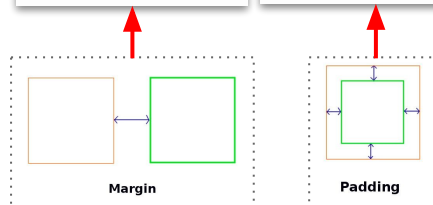
80%

Margin

top: 10 px, bottom: 30 px, left: 0 px, right: 0 px

Padding

top: 5 px, bottom: 5 px, left: 5 px, right: 5 px



Border

width: 3

color: rgba(208, 2, 27, 1)

style: solid

border radius: 15

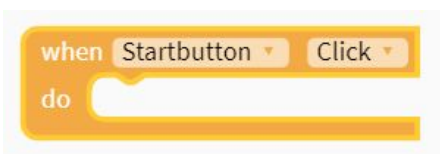
## 5. Die Komponenten programmieren

Nun können wir endlich in die Thinkable Programmieroberfläche wechseln. Hierzu klicken wir auf "Blocks":

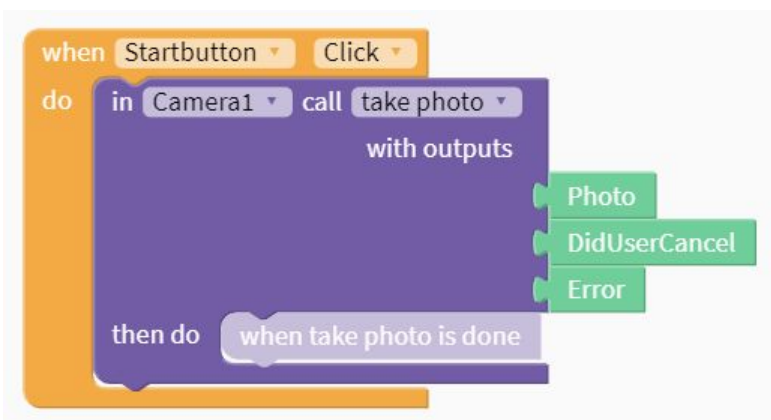


### 5a. Foto machen, wenn Button gedrückt wird

Als erstes soll nun der **Startbutton** so programmiert werden, dass die Kamera-App startet, wenn auf den Button geklickt wird. Hierzu klicken wir auf die Komponente **Startbutton** und ziehen den Block "when Startbutton.Click do" in die Programmieroberfläche hinein:

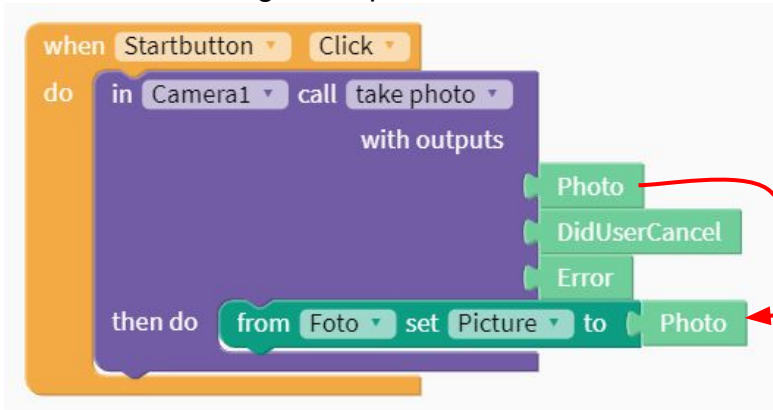


Nun können wir angeben, was passieren soll, wenn der Button geklickt wird. Die Kamera soll ein Foto machen. Dazu gibt es bei der Komponente "Camera1" den Block "in Camera1 call: take photo". Diesen ziehen wir nun in den ersten Block hinein:



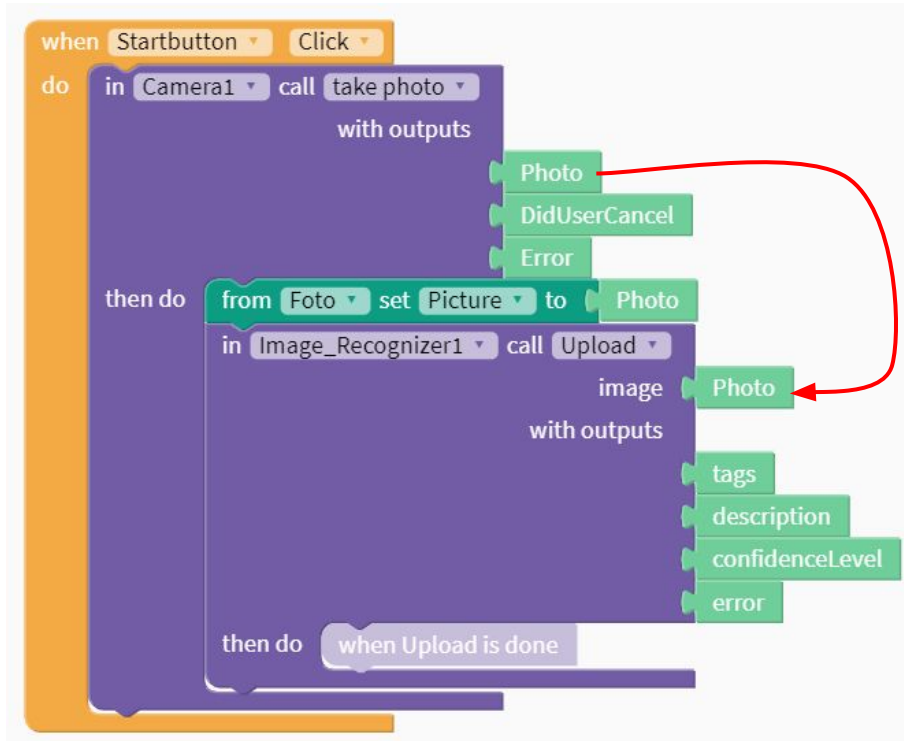
## 5b. Foto in der App anzeigen

Das Foto, das wir gemacht haben, soll jetzt in der App angezeigt werden. Dazu müssen wir es in unsere **Image-Komponente** (die wir **Foto** genannt haben) hineinladen. Das können wir machen, indem wir das **Foto aus den Outputs der Kamera** mit dem Block **“from Foto set Picture to: Photo”** in die Image-Komponente hineinladen:



## 5c. Foto an den Bilderkennungsdienst senden

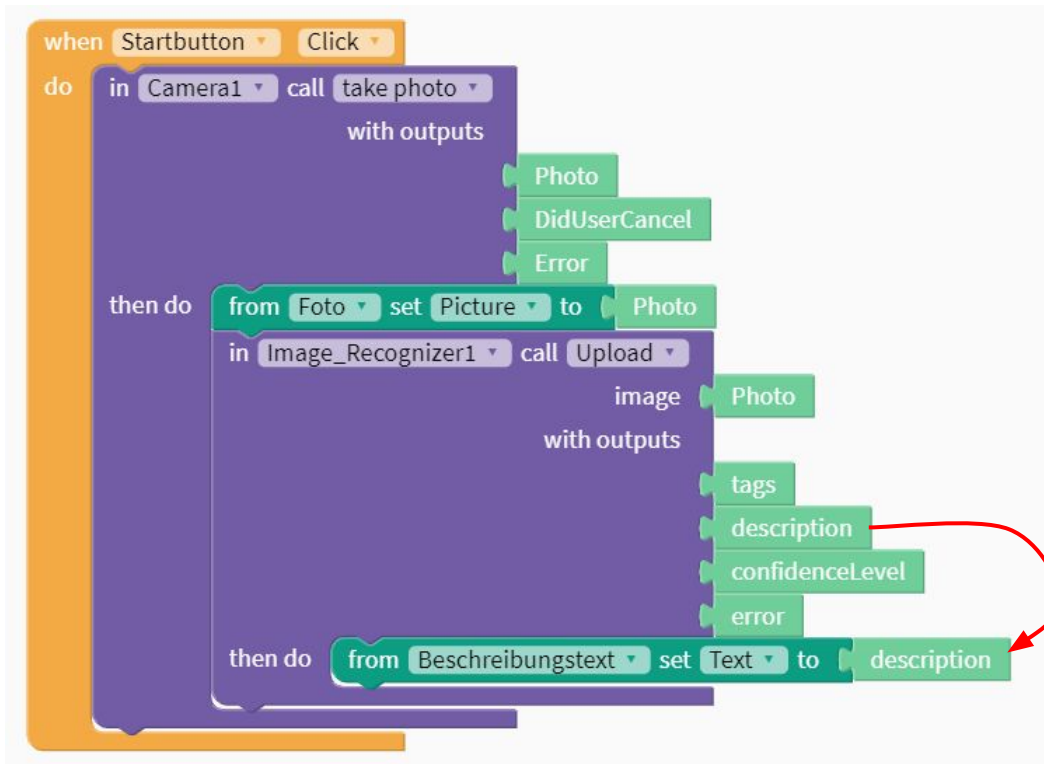
Jetzt, wo wir ein Foto gemacht haben, können wir dieses an den **Bilderkennungsdienst** senden. Das geht mit der **Image-Recognizer-Komponente**. Bei dieser gibt es den Block **“in Image\_Recognizer1 call upload”**. Mit diesem Block können wir das Foto jetzt an diesen Bilderkennungsdienst in der Cloud senden:





## 5d. Den Text setzen

Vom Bilderkennungsdienst erhalten wir als Output eine Beschreibung. Das ist ein Text, der von einer Künstlichen Intelligenz generiert wird, in dem drin steht, was diese KI im Bild erkannt hat. Dieser Text befindet sich im **Output "description"**. Diesen Text laden wir jetzt in unser **Label** hinein, das wir **"Beschreibungstext"** genannt haben:



**Gratuliere, du hast den Programmiereteil abgeschlossen!**

Jetzt können wir unsere App mit der App **"Thinkable Live"** (verfügbar im **Google Play Store** und **Apple AppStore**) testen.



## 6. Die App testen

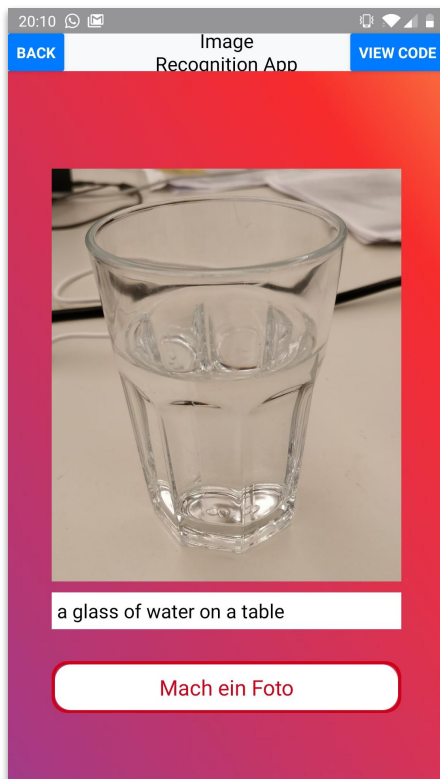
Nun wollen wir die App endlich testen. Hierzu müssen wir auf einem Smartphone oder Tablet die App "Thinkable Live" (verfügbar im **Google Play Store** und **Apple AppStore**) installieren.

**Wichtig:** In der App müssen wir uns mit dem **selben Google Account einloggen**, wie bisher (z.B. [student77@davincilab.at](mailto:student77@davincilab.at)).

Nun in Thinkable (im Browser) rechts oben auf den Button "Live test" klicken:



Jetzt können wir die App auf unserem eigenen Smartphone testen! (**Tipp:** Sollte das nicht gleich funktionieren, hilft es manchmal, Thinkable auf dem Handy über den App Switcher zu killen und neu zu starten.)



## 7. Die App teilen



Erstelle zum Schluss einen **Link zu deiner App**, indem du auf "**Share**" und dann auf "**Generate Link**" klickst. Diesen Link kannst du kopieren und an deine Freunde versenden. Jede/r der/die einen Thinkable-Account hat und eingeloggt ist, kann diesen Link - und damit dein Projekt - öffnen.

## 8. Wie geht's weiter?

Beim Erstellen von Apps sind deiner Fantasie keine Grenzen gesetzt! Allerdings braucht es manchmal Zeit und Geduld bis man herausgefunden hat, wie sich die eigenen Ideen umsetzen lassen.

Deshalb haben wir eine kleine Auswahl an Links für dich zusammengestellt, die dir bei deinen nächsten Apps weiterhelfen können:

- **Thinkable Docs**

<https://docs.thinkable.com/>

Hier findest du detaillierte Erklärungen zu allen Thinkable **Komponenten** (z.B. Button, Label, Text Input...) und **Programmierblöcken**. Es lohnt sich, sich diese Seite genauer anzusehen, insbesondere, wenn du verstehen möchtest, wofür eine bestimmte Komponente oder ein bestimmter Programmierblock gut ist.

- **Thinkable Spiele**

<https://docs.thinkable.com/gaming>

In den Docs findest du auch Spiele, die mit Thinkable programmiert wurden. Du kannst dir anschauen, wie sie programmiert wurden und sie sogar remixen, um ein eigenes Spiel zu erstellen.

- **Thinkable Community**

<https://community.thinkable.com/>

Bei Problemen zu allen möglichen Sachen kann dir die Thinkable Community weiterhelfen.

- **Tutorials auf der Thinkable Community Website**

<https://community.thinkable.com/t/thinkable-x-beginner-tutorials/40418>

Auf der Website der Thinkable Community gibt es Tutorials für Einsteiger/innen, erstelle z.B. einen Generator für Zufallsantworten, nutze Text-to-Speech, erstelle einen Übersetzer oder erstelle eine Schnitzeljagd-App.

- **YouTube-Kanal: Thinkable X**

<https://www.youtube.com/channel/UCTVZRyybOCDBL2zLXSeQVsw/videos>

Der offizielle YouTube-Kanal von Thinkable. Hier gibt es Anleitungen, zu allem, was das Herz begehrt. Beispiele:

- Currency Converter with API

[https://www.youtube.com/watch?v=bNoz9hl\\_w54](https://www.youtube.com/watch?v=bNoz9hl_w54)

- How to use the Thinkable Web API: Weather App

<https://www.youtube.com/watch?v=V-OO0wEYcrs>

- How to use Thinkable Web API Component: QR Code Scanner App

<https://www.youtube.com/watch?v=MfMGA6S5YvE>

- und viele mehr...

- **YouTube-Kanal: Thinkable X Tutorials**

<https://www.youtube.com/channel/UCHDDJy-6nbgwdrJpSZlfCOA>

Hier findest du zahlreiche Video-Tutorials. Beispiele:

- 50 Thinkable X Tips and Tricks

<https://www.youtube.com/watch?v=Gsf8XtnAHUs>

- How to Take a Photo in Thinkable X with the Camera Component

<https://www.youtube.com/watch?v=rik-A44itK8>

- und viele mehr...