



Was passiert hier?

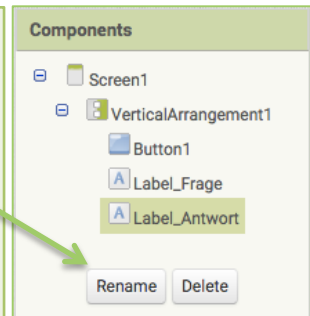
Links siehst du das Design der App und die Komponenten, die du benötigst. Das *VerticalArrangement* findest du in der *Palette* unter *Layout*.

Programmierung: Beim Klick auf die Wahrheitskugel wählst du per Zufall (*pick a random item*) ein Listenelement aus. Die Liste kannst du mit dem Block *make a list* selbst erstellen.



Tipps:

1. Woher kommt *LabelAntwort*? Das ist das *graue Label*. Im Design Bereich kannst du es umbenennen: *Button Rename*.
2. Dem *make a list* Block kannst du weitere Elemente hinzufügen, indem du auf das kleine blaue Feld  klickst.



Du brauchst diese Komponenten:

- 1 Button
- 2 Label
- 1 VerticalArrangement

Bild und Sound Dateien für diese App findest du hier: http://bit.ly/appcamps_wahrheit

Nächste Aufgabe:

- a) Spiel beim Klick auf den Button den *Magie.mp3* Sound ab.
- b) Gib die Antwort der Wahrheitskugel als Sprache aus. Tipp: *Text-to-Speech* Komponente unter *Media*.



Du brauchst diese Komponenten:

- 1 Button
- 2 Label
- 1 VerticalArrangement
- 1 Sound
- 1 TextToSpeech

Was passiert hier?

Wir erweitern die App: Wenn auf die Wahrheitskugel geklickt wird, wird ein Sound abgespielt. Außerdem wird die zufällige Antwort auch per Sprachausgabe ausgegeben.



Tipps:

1. Das *TextToSpeech* Element findest du in der Palette unter Media.
2. Falls dein Gerät mit einem merkwürdigen Akzent spricht, kann das an den Einstellungen liegen. Du musst die Sprache anpassen.

Nächste Aufgabe: Lass den Sound und die Sprachausgabe nacheinander abspielen. **Tipp:** Lösche den Sound und nutze stattdessen die Player Komponente. Den Player findest du ebenfalls unter Media. Mit dem Player hast du mehr Möglichkeiten als mit dem Sound Objekt.



Was passiert hier?

Wir passen den Code an, damit der Sound und die Sprachausgabe nacheinander abgespielt werden.

Um Sounds nacheinander abzuspielen, brauchen wir das *Player* Element. Dieses findest du in der *Palette* unter *Media*. Das *Sound* Element kannst du löschen (Components *Delete*).

```

when Button1.Click
do
  set Label_Antwort.Text to pick a random item list
  make a list
  "Jaaaa!"
  "Auf alle Fälle."
  "Da bin ich mir nicht so sicher."
  "Definitiv nicht."
  "Na klar!"
  call Player1.Start
  
```

```

when Player1.Completed
do
  call TextToSpeech1.Speak
  message Label_Antwort.Text
  
```

Tipp: Das TextToSpeech Element befindet sich jetzt nicht mehr im Block *Button1.Click* sondern in einem anderen Block. Aktionen können also nicht nur durch den Benutzer ausgelöst werden (z.B. durch Klicken), sondern auch durch andere „Events“ (hier: *Player1.Completed*).

Nächste Aufgabe:
 Zeige an wie viele Fragen die Wahrheitskugel bereits beantwortet hat. Um dir die Anzahl zu „merken“, benötigst du eine Variable (Build-in → Variables). Um die Anzahl anzuzeigen, benötigst du ein Label.

Du brauchst diese Komponenten:

- 1 Button
- 2 Label
- 1 VerticalArrangement
- 1 Player
- 1 TextToSpeech



Du brauchst diese Komponenten:

- 1 Button
- 3 Label
- 1 VerticalArrangement
- 1 Player
- 1 TextToSpeech

Was passiert hier?

Wir zählen und zeigen an, wie häufig die Wahrheitskugel antwortet. Dazu fügen wir im Design zunächst ein neues Label hinzu. In der Programmierung legen wir eine Variable an (Build-in → Variables) und nennen diese „zaehler“. Bei jedem Klick wird der Zähler hochgezählt und im Label angezeigt.

Tip: Um sich die Anzahl der Antworten zu „merken“ benötigen wir eine Variable. Diese nennen wir „zaehler“.



Nächste Aufgabe:

Überlege dir, wie du die App noch erweitern kannst. Ideen: Füge weitere Sounds ein, die zufällig abgespielt werden oder ändere die Bilder der Wahrheitskugel.