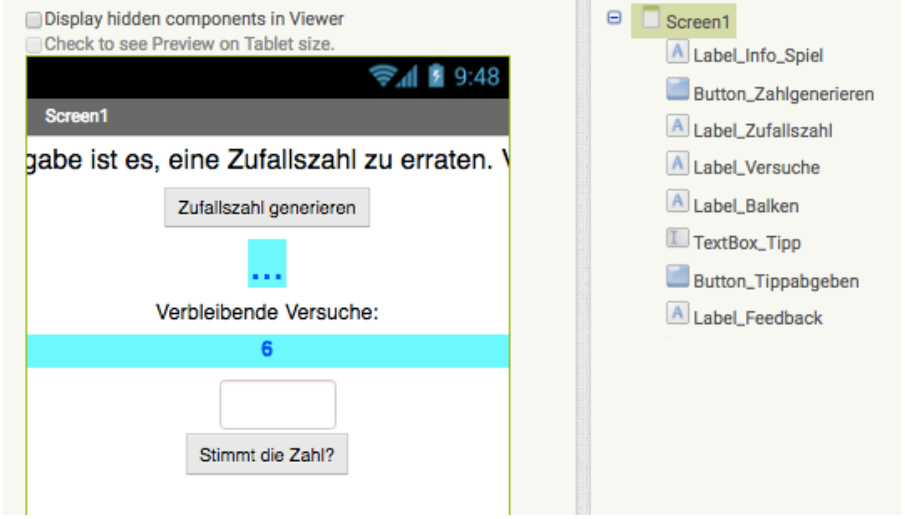
Aufgabe

Programmiere eine App, die per Buttonklick eine Zufallszahl zwischen 1 und 100 generiert, die von der Benutzerin erraten werden muss.

- Der Benutzer soll die Zahl erraten, indem er einen Tipp in eine Textbox eingibt und einen Button klickt.
- Der Benutzer hat 6 Versuche, um die richtige Zahl zu erraten. Die Anzahl der Versuche wird per Balken angezeigt, der nach jedem Versuch kleiner wird.
- Gib der Benutzerin Feedback wenn sie die Zahl erraten hat bzw. alle Versuche aufgebraucht sind.
- Per Klick auf den Button **Zufallszahl generieren** wird eine neue Zufallszahl generiert und der Benutzer hat wieder 6 Versuche.

Design der App

Hier siehst du welche **Komponenten** du benötigst. Benenne sie am besten so wie hier.



Tipps:

Der türkise Balken ist ein Label (*Label_Balken*) mit *Width = 100%*, *Text = 6* und *BackgroundColor = Cyan*

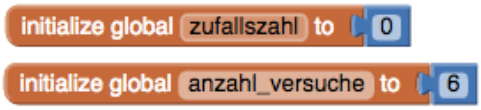
Das *Label_Feedback* ist anfangs ohne Text (d.h. hier nicht sichtbar).

Um alles zu zentrieren bitte bei Screen1 unter Properties *AlignHorizontal = Center* auswählen.

Programmierung der App

1: Variablen initialisieren

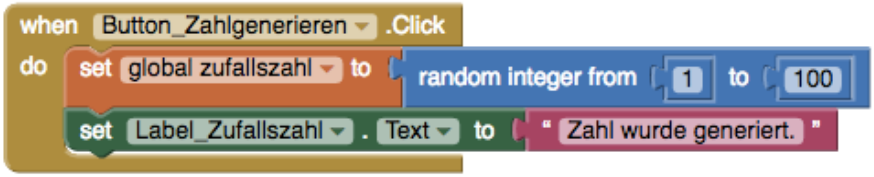
Du brauchst zwei Variablen, um die Anzahl der Versuche und die Zufallszahl zu speichern. Benenne und initialisiere die beiden Variablen bitte wie folgt:



Initialisierung bedeutet, dass man einer Variablen einen Wert zuweist.

2: Zufallszahl generieren

Um das Ratespiel zu starten, muss der Benutzer zunächst per Klick auf **Button_Zahlgenerieren** eine Zufallszahl generieren. Das **Label_Zufallszahl** wird angepasst, damit der Benutzer weiß, dass die Zahl generiert wurde.



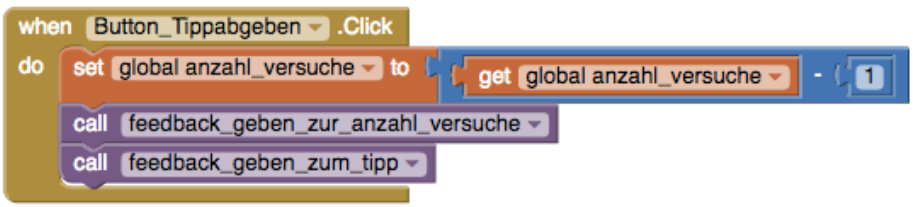
3. Button Tippabgabe programmieren & Methoden

Jedes mal wenn ein Tipp abgegeben wird, muss ein Versuch abgezogen werden. Gleichzeitig bekommt der Benutzer Feedback zur Anzahl der verbleibenden Versuche und ob der Tipp richtig, zu groß oder zu klein ist.

Damit der Code übersichtlicher ist, verwenden wir zwei Methoden. Wir legen die Methoden Blöcke zunächst an und programmieren sie später.

Tipp: Einen Methoden „Block“ findest du links unter Procedures. Benenne **procedure** bitte um, damit deine Methoden Blöcke wie folgt aussehen.

Programmiere den **Button_Tippabgeben** wie folgt. Hierfür brauchst du die beiden Methoden, die du eben angelegt hast.



Programmierung der App

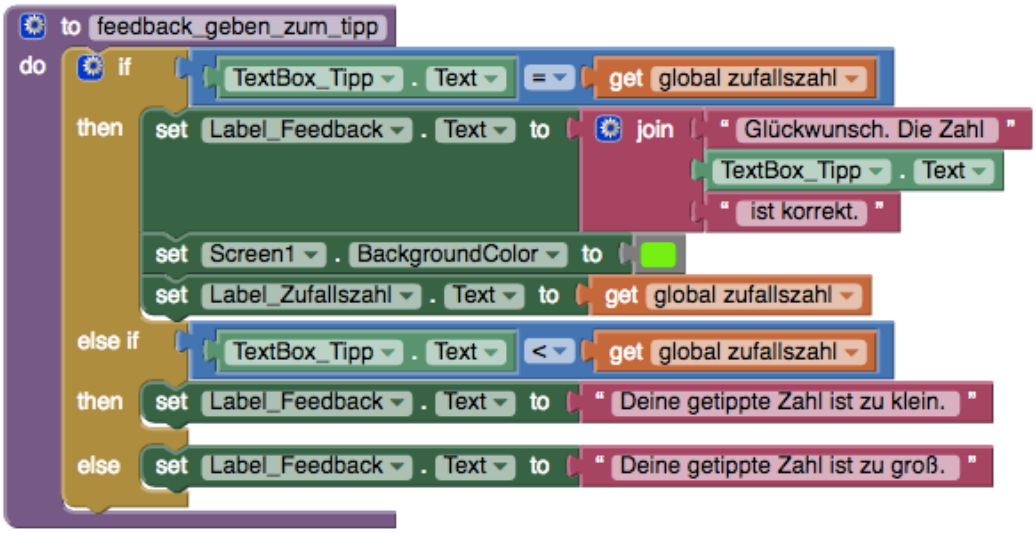
4: Methode feedback_geben_zur_anzahl_versuche programmieren

Im vorhergehenden Schritt hast du die Methode angelegt. Jetzt muss sie noch programmiert werden, damit sie auch tut, was sie tun soll.

Wenn (if) der Benutzer noch Versuche übrig hat (anzahl_versuche > 0), wird der Balken kleiner und die neue Anzahl an Versuchen wird ausgegeben.
Sonst (else) wird ausgegeben, dass keine Versuche übrig sind. Die Zufallszahl wird angezeigt (damit der Benutzer weiß, was richtig gewesen wäre), der Screen Hintergrund wird auf rot gesetzt und der Balken verschwindet.

5: Methode feedback_geben_zum_tipp programmieren

Wenn (if) der Benutzer die Zufallszahl richtig getippt hat, wird das im Label_Feedback entsprechend rückgemeldet, die Hintergrundfarbe wird grün und die Zufallszahl wird ausgegeben.
Sonst wenn (else if) der abgegebene Tipp kleiner ist als die Zufallszahl, wird im Label_Feedback ausgegeben, dass die Zahl zu klein ist.
Sonst (else) wird ausgegeben, dass die Zahl zu groß ist.



App testen und weitere Features entwickeln

Deine App sollte nun funktionieren. Teste sie bitte. Dir wird auffallen, dass es an einigen Stellen noch Verbesserungsbedarf gibt.

Hier findest du ein paar Ideen, wie du die App weiterentwickeln kannst. Versuche die Aufgaben zunächst selbst zu lösen. Die Lösung findest du auf den nächsten Seiten.

a) Spiele eine kurze Musik ab, wenn der Button „Zufallszahl generieren“ geklickt wird. Wenn die Musik fertig ist, gib aus, dass die Zufallszahl generiert wurde. So hat die Benutzerin das Gefühl, dass es einen Moment dauert, die Zufallszahl zu generieren.
Tipp: Hierfür brauchst du den Player. Einen Sound findest du hier **http://bit.ly/knobelapps**

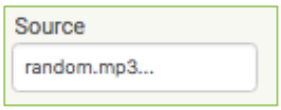
b) Wenn keine Zahl in das Textfeld eingegeben wird, darf eine Tippabgabe nicht möglich sein. Und der Benutzer muss informiert werden, dass eine Zahl eingegeben werden muss.
Tipp: Du brauchst keine neue Komponente hierfür. Eine extra If-Anweisung reicht hier aus.

c) Programmiere den Button „Zufallszahl generieren“ so, dass man das Ratespiel neu starten kann. D.h. man hat wieder 6 neue Versuche, der Balken ist wieder da usw. Du stellst sozusagen den Ausgangszustand der App wieder her.
Tipp: Du musst den Block **when Button_Zahlgenerieren.click** erweitern.

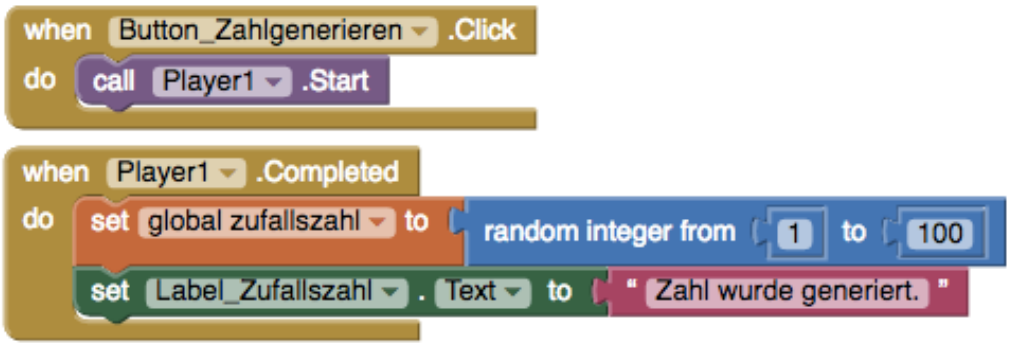
d) Denke dir eigene Features aus, die du gerne entwickeln willst.

Aufgabe a

Ergänze im Design-Bereich einen Player und verknüpfe eine Sound-Datei (z.B. random.mp3) mit dem Player (im Bereich Properties).



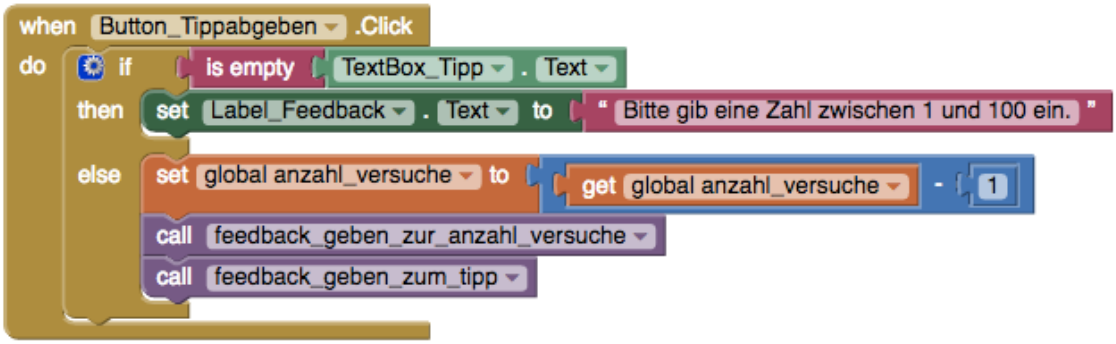
- Wenn der Button_Zahlgenerieren geklickt wird, starte den Player.
- Wenn der Player fertig ist, erstelle die Zufallszahl und gib aus, dass sie generiert wurde.



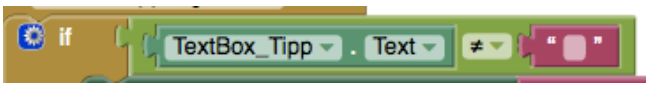
Aufgabe b

Wenn der Button_Tippabgeben geklickt wird, muss zunächst abgefragt werden, ob die TextBox leer ist (is empty). Falls ja, gib im Label_Feedback aus, dass eine Zahl eingegeben werden muss.

Ein Versuch wird nicht abgezogen und die beiden Methoden werden auch nicht aufgerufen.



Eine andere Möglichkeit (statt i**s empty**) für die if-Abfrage:

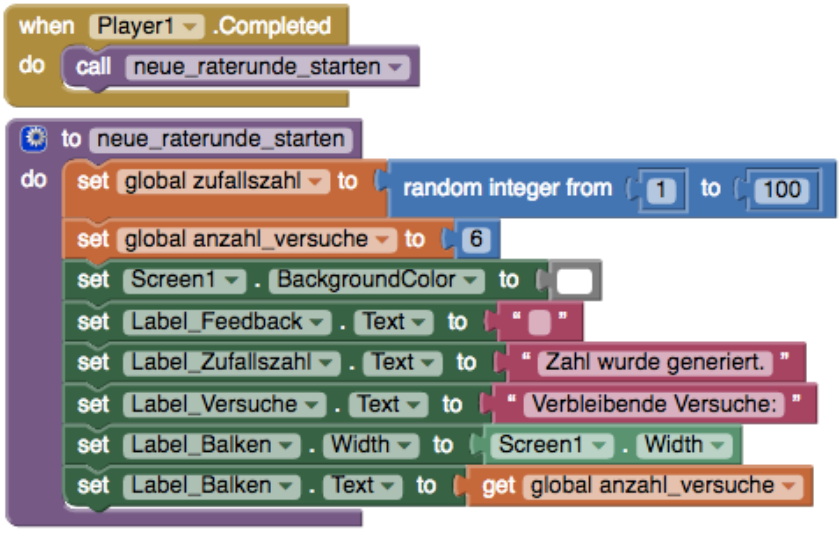


Aufgabe c

Erstelle eine neue Methode und gib ihr den Namen **neue_raterunde_starten**. Wenn der Player die Musik fertig abgespielt hat, rufe die neue Methode **neue_raterunde_starten** auf.

Das was vorher direkt ausgeführt wurde, als der Player fertig war, wird nun auch in der neuen Methode ausgeführt. Zusätzlich musst du folgende Dinge tun:

- Anzahl Versuche auf 6 setzen
- Screen Hintergrund weiß färben
- Den Feedback Text löschen
- Verbleibende Versuche als Labeltext anzeigen
- Den Balken wieder auf die Bereite des Screens setzen und hier die Anzahl der Versuche (6) als Text anzeigen.



Damit der Code übersichtlicher ist, werden erst die Variablenwerte zugewiesen und dann alle anderen Dinge angepasst. Die Reihenfolge der Anweisungen ist egal – bis auf eine Ausnahme. Findest du sie?

Aufgabe d

Hast du noch weitere Ideen, wie du die App anpassen kannst? Viel Spaß!