Arbeitsblatt 06 _ Guess The Number

Lösung: Gewinnchancen

Gewinnchance durch Binärsplit::

$P = \frac{1}{100} + \frac{1}{50} + \ + \frac{1}{25} + \frac{1}{12.5} = \ldots$

Am besten ist es, das zu codieren. Zum Beispiel unten in Octave-Online (unten nicht ein kompakter Code, aber einfacher zu folgen):

```
a = 100./2.^[0:1:9];

b = zeros(1,10);
for i=1:10
    if i==1
     b(i) = 1/a(i);
    else
      b(i) = b(i-1) + 1/a(i);
    end
end

Matrix = [[1:1:10];a;100*b];
disp(" Draw Options  Chances(%)")
disp(int16(Matrix'))
```

Wie viele Optionen gibt es in jeder Ziehung:

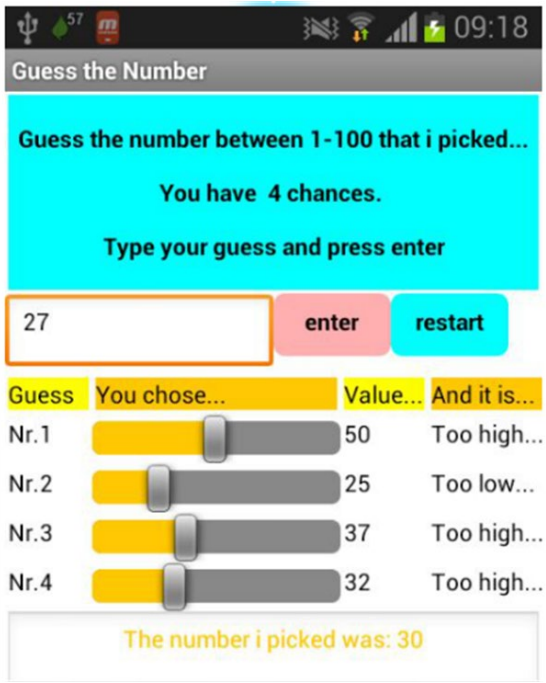
a = 100 50 25 12.5 6.25 3.125 1.5625 0.7813 0.3906 0.1953

Die Gewinnchancen bei 1 bis 10 Ziehungen sind:

b = 0.01 0.03 0.07 0.15 0.31 0.63 1.27 2.55 5.11 10.23

Gruppierung der Ergebnisse in einer Matrix zur einfacheren Visualisierung

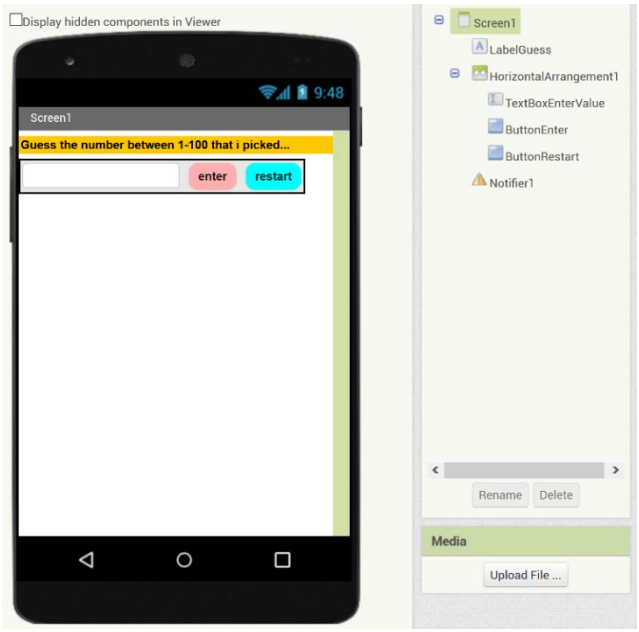
Draw	Options	Chances(%)
1	100	1
2	50	3
3	25	7
4	13	15
5	6	31
6	3	63
7	2	127
8	1	255
9	0	511
10	0	1023



Für ein faires Spiel muss die BenutzerIn also 6 Chancen haben.

Unten zeigen wir einige mögliche Implementierungen. Nicht alle haben 6 Chancen, aber das lässt sich leicht ändern!

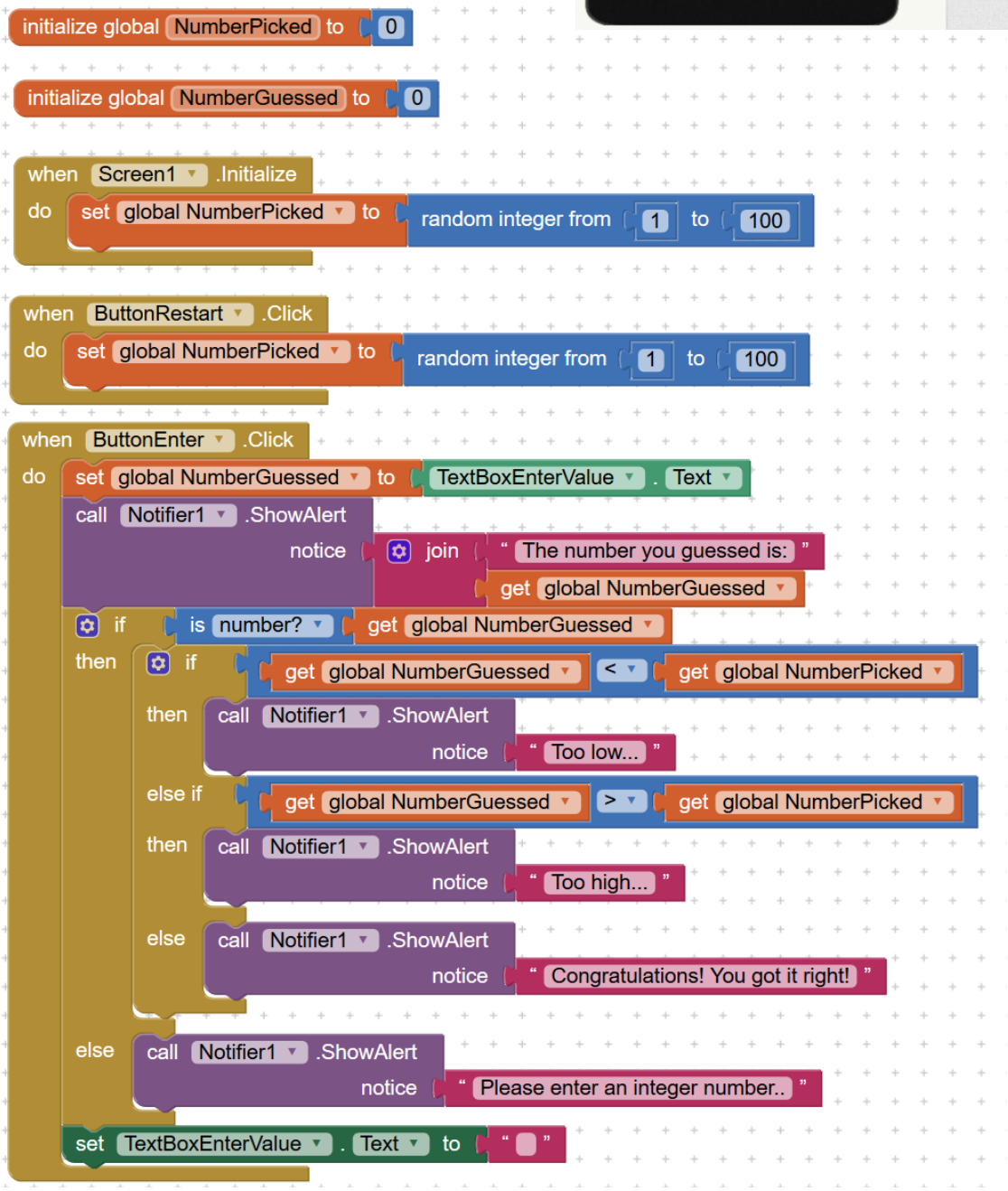
Variante-1 : Basic



Eigenschaften:

- Unbegrenzte Anzahl von Rateversuche.
- Benutzer-Feedback (zu hoch, zu niedrig, richtig) mit Benachrichtigung (Notifier).
- Neustart-Button, Erneuerung der zu erratenden Zufallszahl.

Frage:
Prüfe auf eine mögliche Verbesserung durch ein Prozedur!

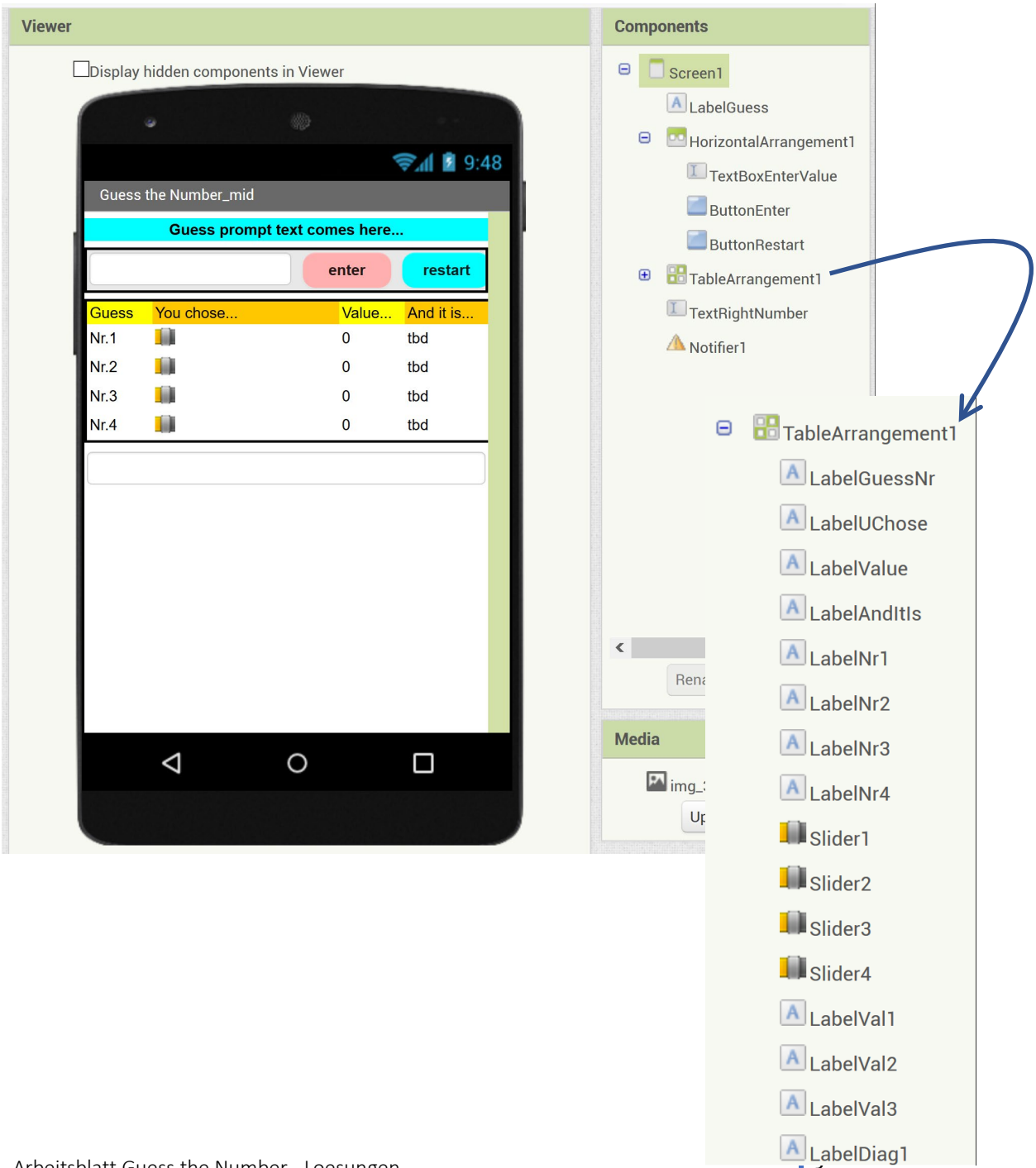


Variante-2: Mittelkomplex

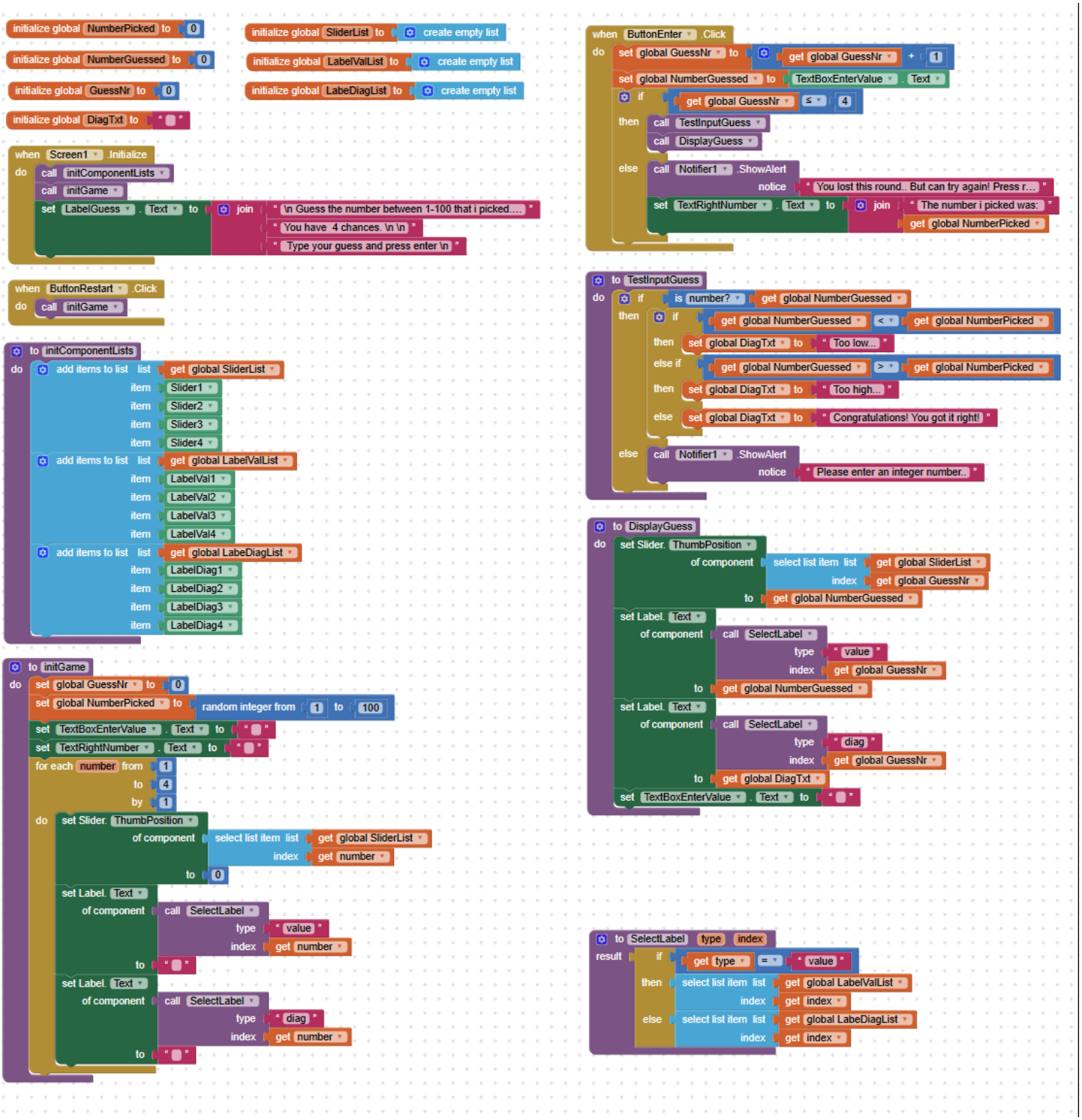
Half-Brute-Force mit Irgendeine-Komponente (Any-Component)

Eigenschaften:

- Begrenzte und feste Anzahl von Rateversuche. Visualisierung mit Tabelle.
- Initialisierung-Prozedur.
- Benutzerfeedback mit Schieberegler.
- Bezeichnung und Schieberegler (labels and sliders) Blöcke sind irgendeine-Komponente (Any-Component) Blöcke, die innerhalb der Schleife angesprochen werden.
- Interessant, um Listen und Indizierung einzuführen.
- Neustart-Button, Zufallszahl erneuern, die zu erraten ist.



Variante 2- Code

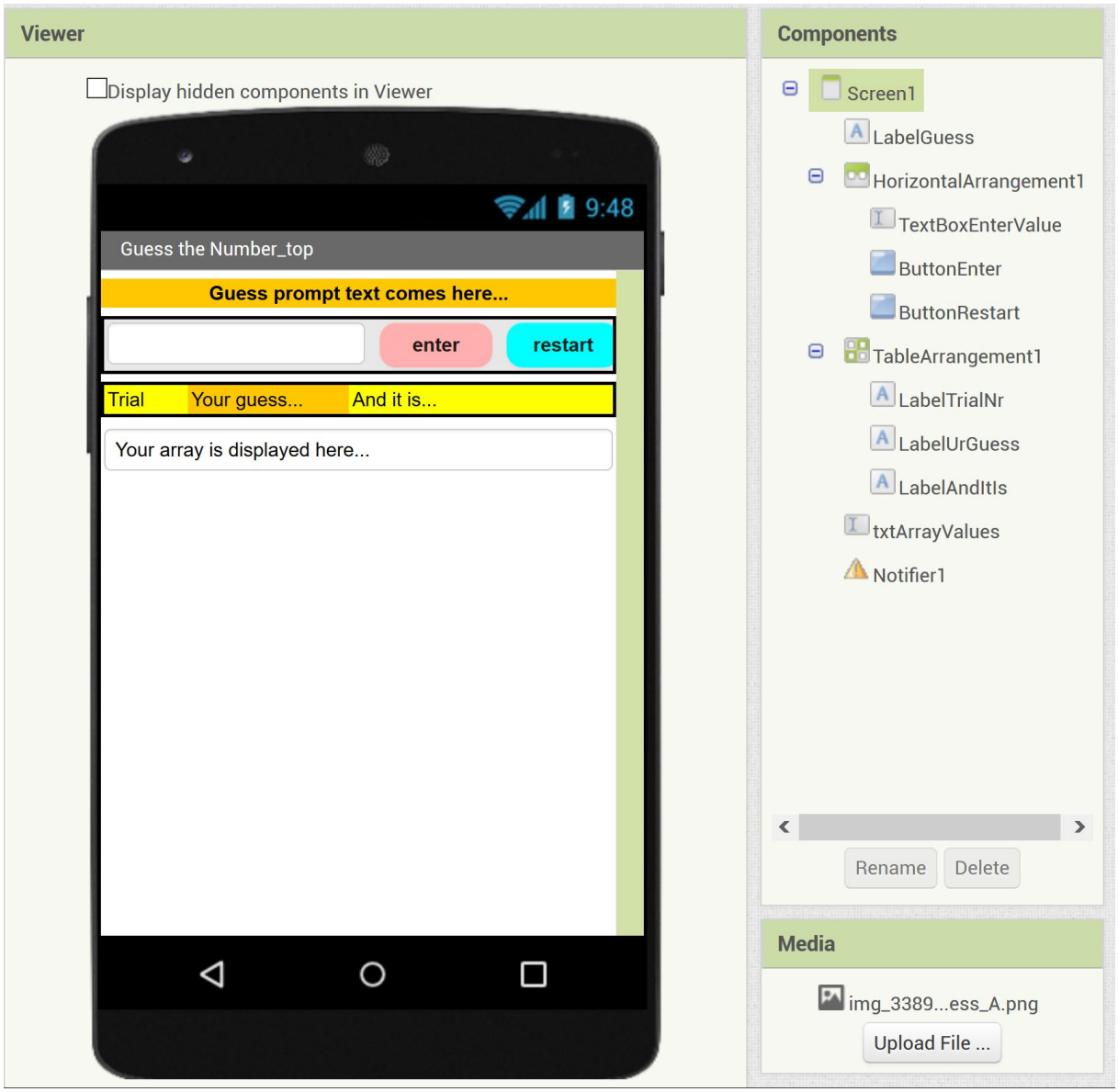


Variante 3: Mittel-hoch-komplex

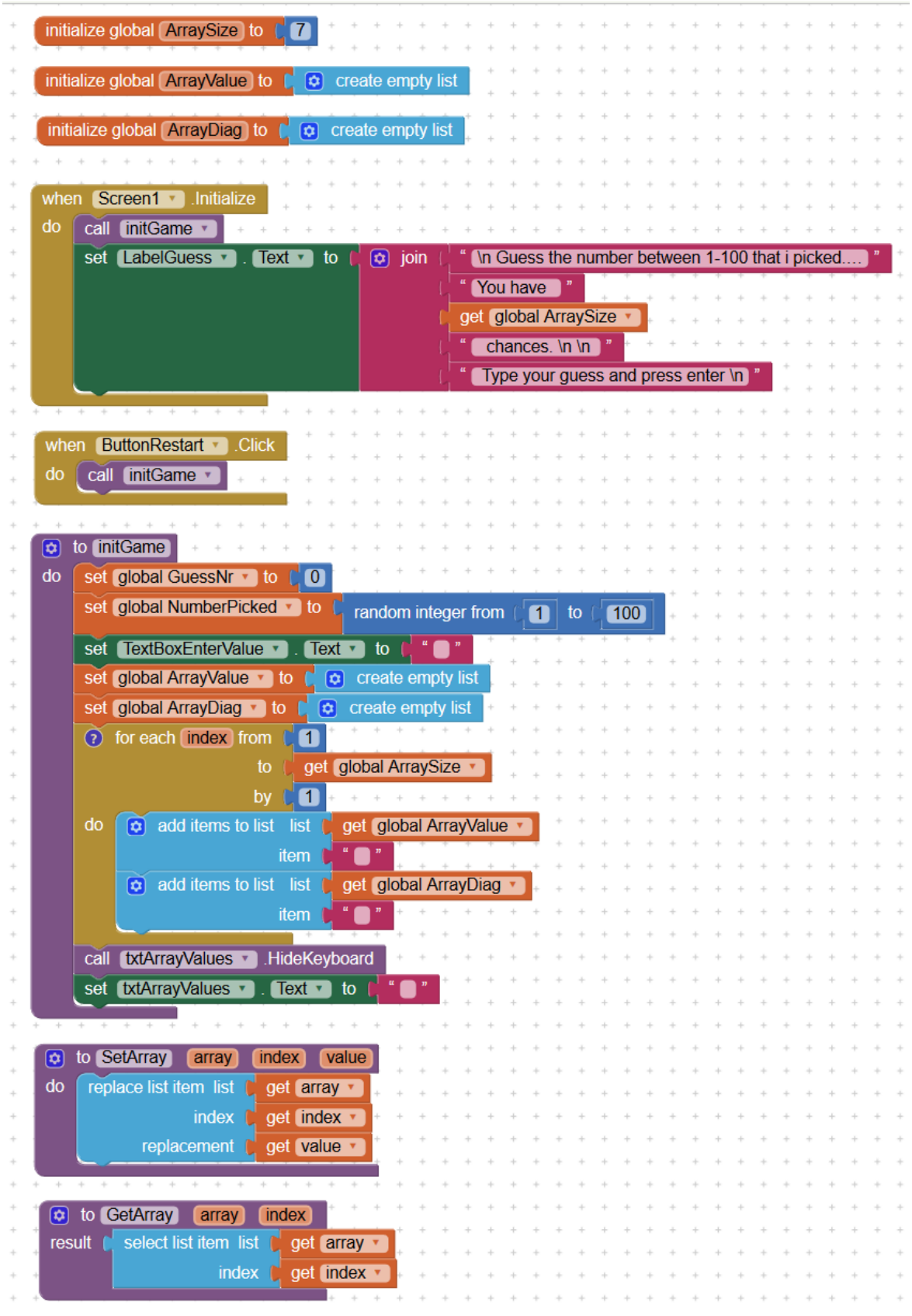
Einstellbare Listen für flexible Anzahl von Rateversuchen

Eigenschaften:

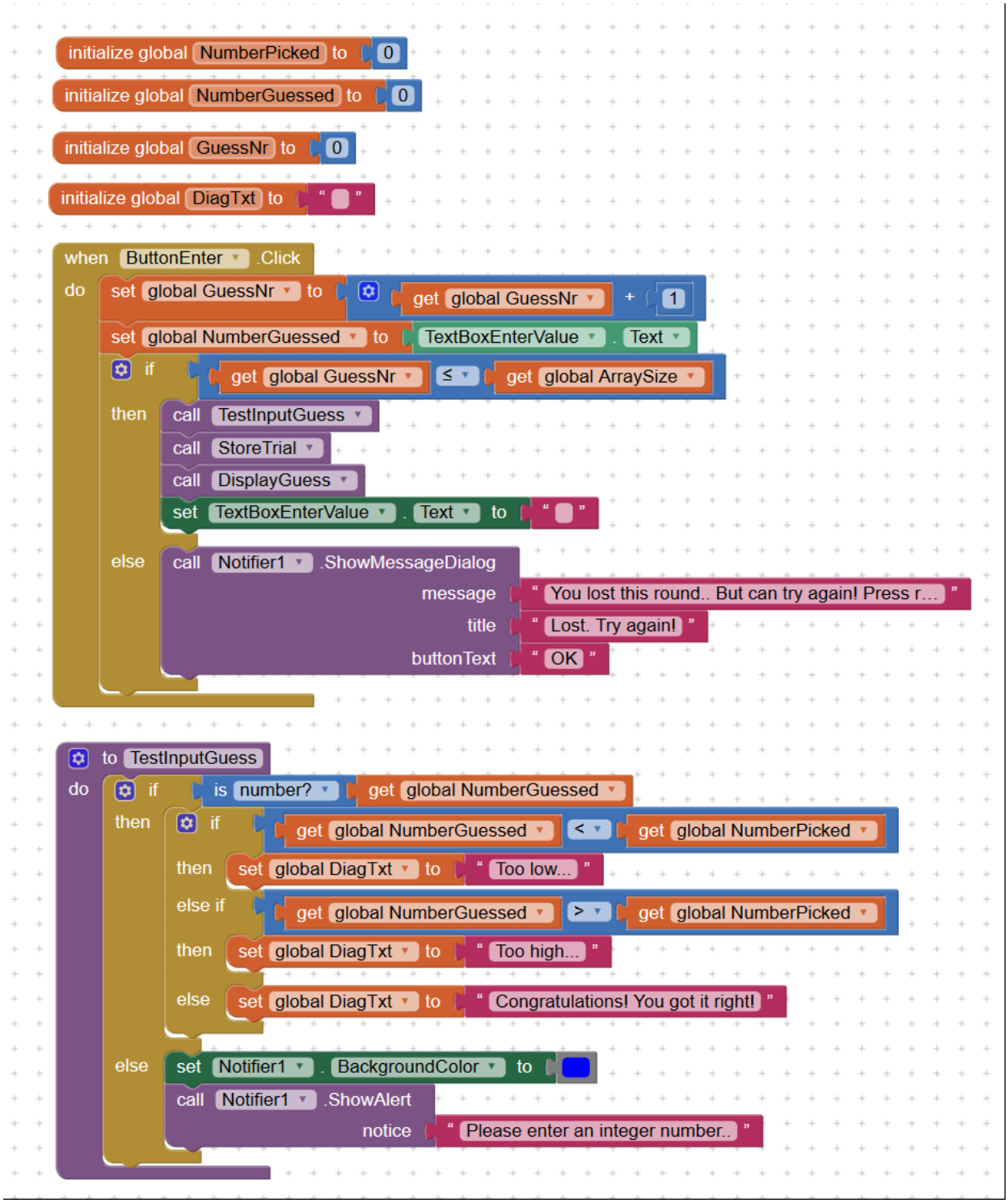
- Anzahl der Rateversuche in globaler Konstante definiert, und kann einfach geändert werden, ohne dass weitere Änderungen im Code notwendig sind.
- Umfangreiche Verwendung von Prozeduren. Nettes Beispiel für "Teile und herrsche". Aber da der Code in AI2 flach ist, wird er für einen einzelnen Bildschirm sehr groß.



Variante 3: Code Teil 1/3



Variante 3: Code Teil 2/3



Variante 3: Code Teil 3/3