Themen

Sortieralgorithmus Bubble Sort

Die Menge der weltweit erzeugten Daten wächst von Jahr zu Jahr schneller. Deshalb ist es wichtig, Ordnung in die Daten zu bringen. Sortieraufgaben gehören daher zu den Aufgaben, die von Computern am häufigsten bearbeitet werden müssen.

Das einfachste Sortierverfahren ist das Sortieren durch Vertauschen. Dabei werden bei jedem Sortierschritt zwei benachbarte Elemente, also z. B. Zahlen, miteinander verglichen. Ist die linke Zahl größer, tauschen beide Zahlen ihre Plätze. Das wird so lange wiederholt, bis die Eingabeliste vollständig sortiert ist.

Wenn man von klein nach groß sortiert, steht am Ende jedes Sortierdurchlaufs die jeweils größte Zahl ganz rechts. Sie muss im folgenden Durchlauf nicht mehr beachtet werden. Der erste Durchlauf ist folglich der längste. Er umfasst in einer Liste mit n Elementen n – 1 Sortierschritte (Tausche).

Während des Sortierens wandern große Zahlen Schritt für Schritt immer weiter nach rechts und kleine nach links. Im Beispiel kann man das an der 7 und der 1 gut erkennen. Weil die Zahlen wie Blasen im Wasser wandern, wird dieses Sortierverfahren „Bubble Sort“ („Blasen-Sortierung“) genannt.

Die Anzahl der Sortierdurchläufe richtet sich nach der Anzahl der Elemente. Für eine Liste mit n-Elementen sind n – 1 Sortierdurchläufe notwendig. In den Beispielen rechts sind das entsprechend 3 Sortierdurchläufe für die Liste mit 4 Elementen und 5 Sortierdurchläufe für die Liste mit 6 Elementen.

Auch die Anzahl der Sortierschritte (Tausche) lässt sich anhand der Listenlänge berechnen. Für eine Liste mit n Elementen werden maximal

$$\frac{n\,(n-1)}{2} = \frac{1}{2}\,(n^2 - n)$$

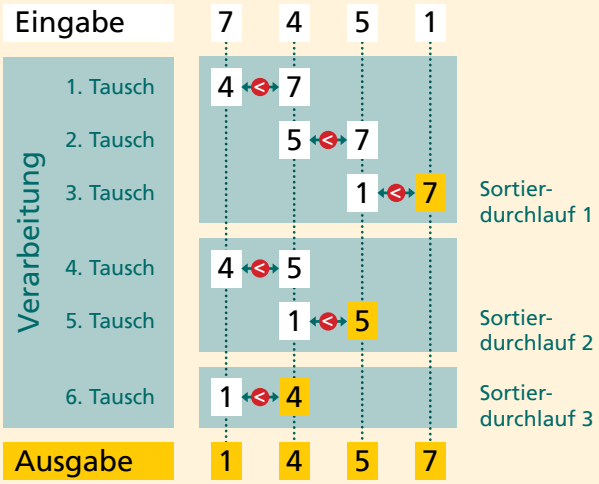
Sortierschritte (Tausche) benötigt.

In unseren Beispielen sind das bei 4 bzw. 6 Elementen entsprechend 6 bzw. 15 Sortierschritte (Tausche).

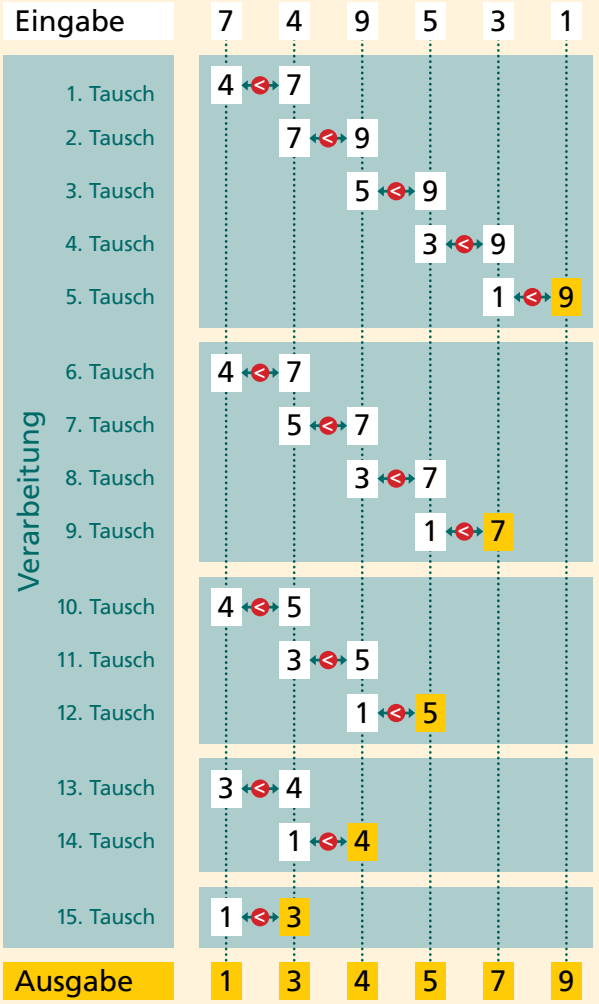
$$\frac{4\,(4-1)}{2} = \frac{1}{2}\,(4^2 - 4) \;=\; 6$$

$$\frac{6\,(6-1)}{2} = \frac{1}{2}\,(6^2 - 6) \;=\; 15$$

Beispiel „Bubble Sort“ mit vier Elementen

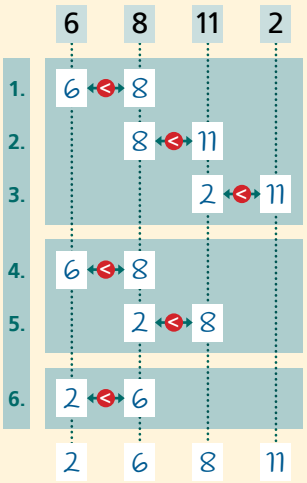


Beispiel „Bubble Sort“ mit sechs Elementen

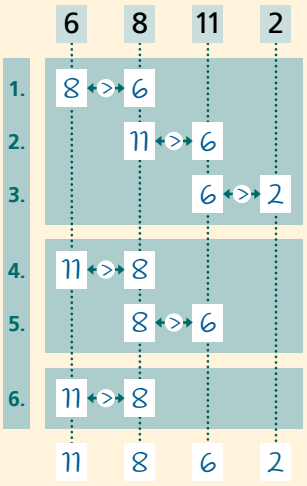


Sortieralgorithmus Bubble Sort

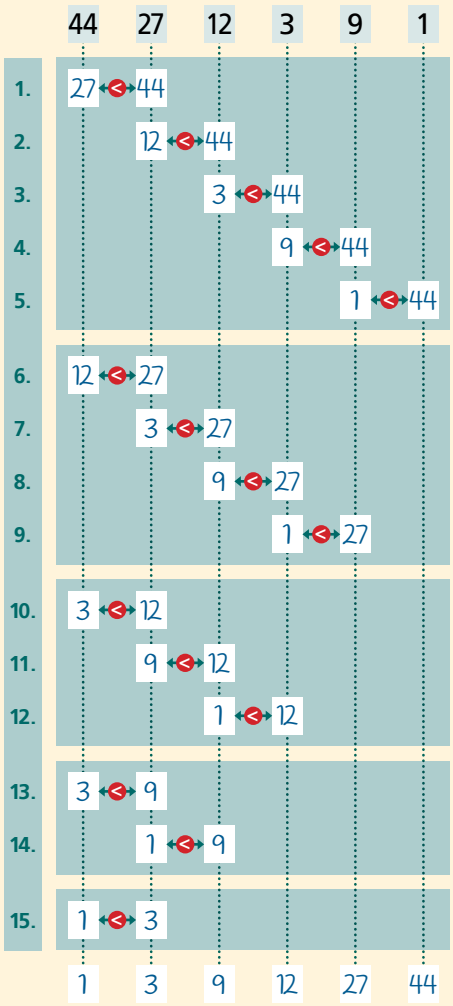
Aufgabe 1
Sortiere diese vier Zahlen von klein nach groß:
6, 8, 11, 2.



Aufgabe 2
Wie muss der Suchalgorithmus verändert werden, um von groß nach klein zu sortieren?

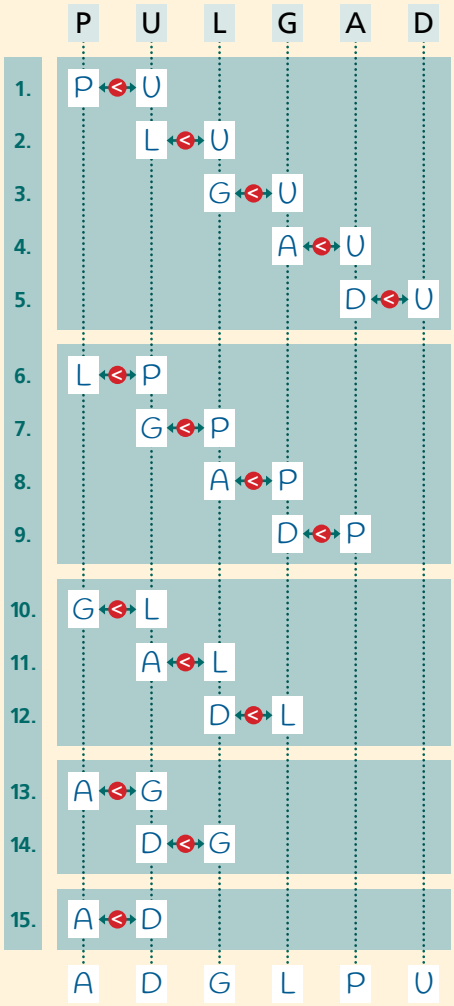


Aufgabe 3
Sortiere diese sechs Zahlen von klein nach groß:
44, 27, 12, 3, 9, 1.

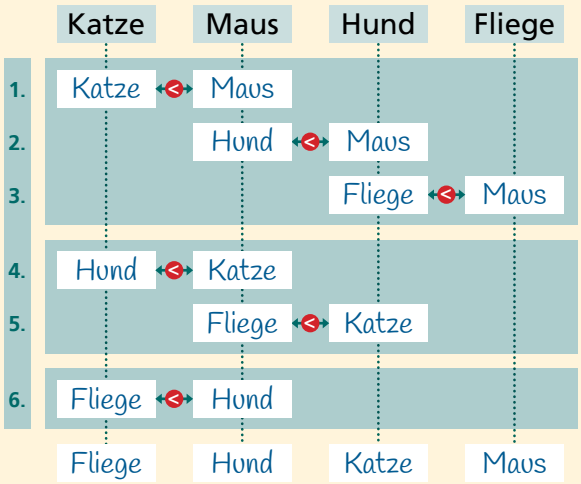


Sortieralgorithmus Bubble Sort

Aufgabe 4
Sortiere diese Buchstaben nach dem Alphabet:
P, U, L, G, A, D



Aufgabe 5
Sortiere diese Tiere nach dem Alphabet:
Katze, Maus, Hund, Fliege



Aufgabe 6
Wie viele Sortierdurchläufe sind bei einer Liste von 10 Elementen maximal notwendig, um sie mit Bubble Sort zu sortieren?

Beim Sortieren einer Liste mit n Elementen mit Bubble Sort sind n – 1 Sortierdurchläufe notwendig. Bei 10 Elementen sind es also 9 Sortierdurchläufe.

Aufgabe 7
a) Wie viele Sortierschritte (Tausche) sind in einer Liste mit 20 Elementen maximal notwendig, bis die erste Zahl von ganz links bis ganz nach rechts gewandert ist?

Um im ersten Sortierdurchlauf eine Zahl von ganz links bis ganz nach rechts zu bewegen, werden maximal n – 1, also 19 Sortierschritte (Tausche) benötigt.

b) Wie viele Sortierschritte (Tausche) werden maximal benötigt, um die komplette Liste zu sortieren?

Insgesamt werden für das Sortieren einer Liste mit n Elementen so viele Sortierschritte (Tausche) benötigt:

$$\frac{n\,(n-1)}{2} = \frac{1}{2}\,(n^2 - n)$$

Für eine Liste mit 20 Elementen ergibt sich daraus

$$\frac{20\,(20-1)}{2} = \frac{1}{2}\,(20^2 - 20) = 190$$

Um die komplette Liste zu sortieren, werden 190 Sortierschritte (Tausche) benötigt.

Zufallszahlen

Wenn ein Ereignis nicht vorhersagbar ist, spricht man von Zufall. Es gibt in diesem Fall keine erkennbaren Gesetzmäßigkeiten oder Erklärungen.

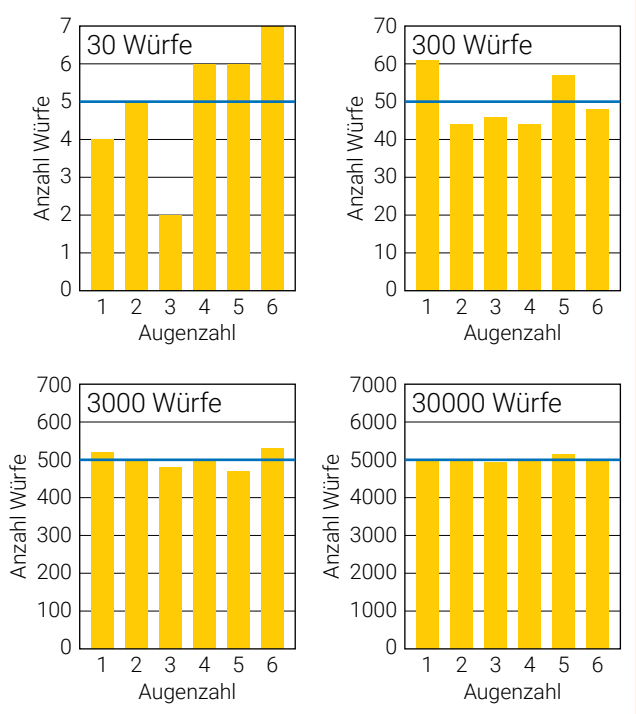
Zufällige Ereignisse entstehen beispielsweise als Ergebnis von Zufallsexperimenten wie dem Werfen einer Münze oder eines Würfels.

Für Zufallsexperimente ist charakteristisch, dass
- sie unter den gleichen Bedingungen beliebig oft durchführbar sind,
- alle möglichen Ergebnisse eindeutig zu benennen sind und
- nicht vorhersagbar ist, welches der möglichen Ergebnisse des Experiments jeweils eintritt.

Das Würfeln mit einem sechsseitigen Würfel hat sechs mögliche Ergebnisse. Die theoretische Wahrscheinlichkeit ist für alle sechs Ergebnisse gleich und beträgt 1/6.

Diese theoretische Häufigkeit der einzelnen Augenzahlen wird beim Würfeln in der Praxis kaum erreicht. Bei wenigen Würfen liegen die Ergebnisse entsprechend weit auseinander.

Diese vier Beispiele zeigen jedoch, dass die Häufigkeit der Augenzahlen sich mit wachsender Wurfanzahl immer mehr dem theoretischen Wert von 1/6 annähert.



Würfeln mit zwei Würfeln

Würfelt man mit zwei Würfeln und addiert die Augenzahl, erhält man elf Summen von 2 bis 12. Anders als beim Würfeln mit einem Würfel treten die elf Summen jedoch nicht gleich häufig auf.

Aus den möglichen Kombinationen der Augenzahlen ergibt sich die folgende Verteilung mit einem Maximum bei der Augenzahlsumme 7.

2	3	4	5	6	7	8	9	10	11	12
1+1	1+2	1+3	1+4	1+5	1+6	2+6	3+6	4+6	5+6	6+6
	2+1	2+2	2+3	2+4	2+5	3+5	4+5	5+5	6+5	
		3+1	3+2	3+3	3+4	4+4	5+4	6+4		
			4+1	4+2	4+3	5+3	6+3			
				5+1	5+2	6+2				
					6+1					

Computergenerierte Zufallszahlen

Eine grundlegende Eigenschaft von Computern ist, dass sie mit fest definierten Algorithmen arbeiten. Gleiche Anfangsbedingungen führen damit immer zum gleichen Ergebnis. Das bedeutet, dass ein Computer kein Ergebnis erzeugen kann, das tatsächlich zufällig ist.

In Scratch gibt es einen eigenen Block für das Erzeugen von Zufallszahlen:

Dahinter stecken mathematische Algorithmen, mit deren Hilfe Scratch Folgen von Zahlen erzeugt, die zufällig aussehen.

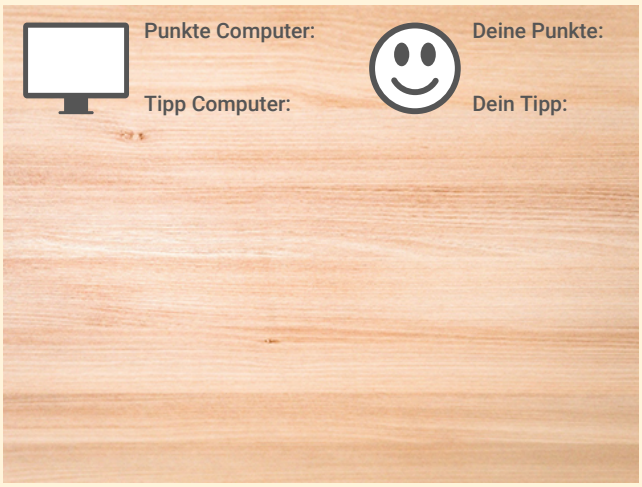
Da es sich bei diesen Computer-erzeugten Zahlen nicht um echte Zufallszahlen handelt, nennt man sie **Pseudo**zufallszahlen.

Kennt man den Algorithmus, nach dem der Pseudozufallszahlen-Generator arbeitet, sind alle dadurch erzeugten Zufallszahlen vollständig vorhersehbar. Für die Steuerung von Computerspielen stellt das kein Problem dar. Setzt man einen Pseudozufallszahlen-Generator jedoch ein, um Passwörter oder die Gewinnzahlen in einem Casino zu generieren, benötigt man sehr komplizierte Algorithmen, die durch Hacker nicht geknackt werden können.

Zufallszahlen

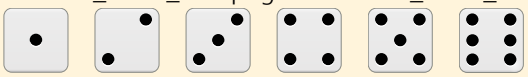
Bühnenbild

Tisch_Wuerfelspiel.png

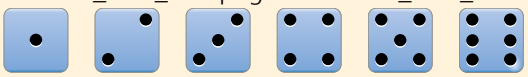


Figuren

Wuerfel_weiss_eins.png bis Wuerfel_weiss_sechs.png



Wuerfel_blau_eins.png bis Wuerfel_blau_sechs.png



(Würfel 1 jeweils als Figur, die restlichen Augenzahlen als Kostüme anlegen)

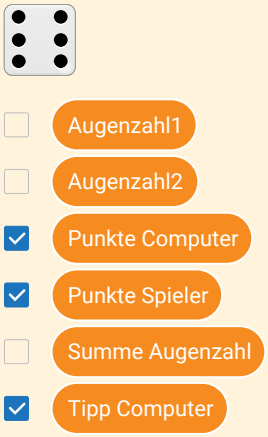
Story

In 10 Spielrunden geben Spieler und Computer jeweils einen Tipp ab, welche Augenzahlsumme beim nächsten Wurf gewürfelt wird. Für jeden richtigen Tipp gibt es einen Punkt. Wer nach 10 Runden mehr Punkte hat, gewinnt das Spiel.

Bilder: djedj, Clker-Free-Vector-Images (Pixabay)

Aufgabe 1 – Variablen

a) Lege diese Variablen an:
- Augenzahl1
- Augenzahl2
- Summe Augenzahl
- Tipp Computer
- Punkte Spieler
- Punkte Computer



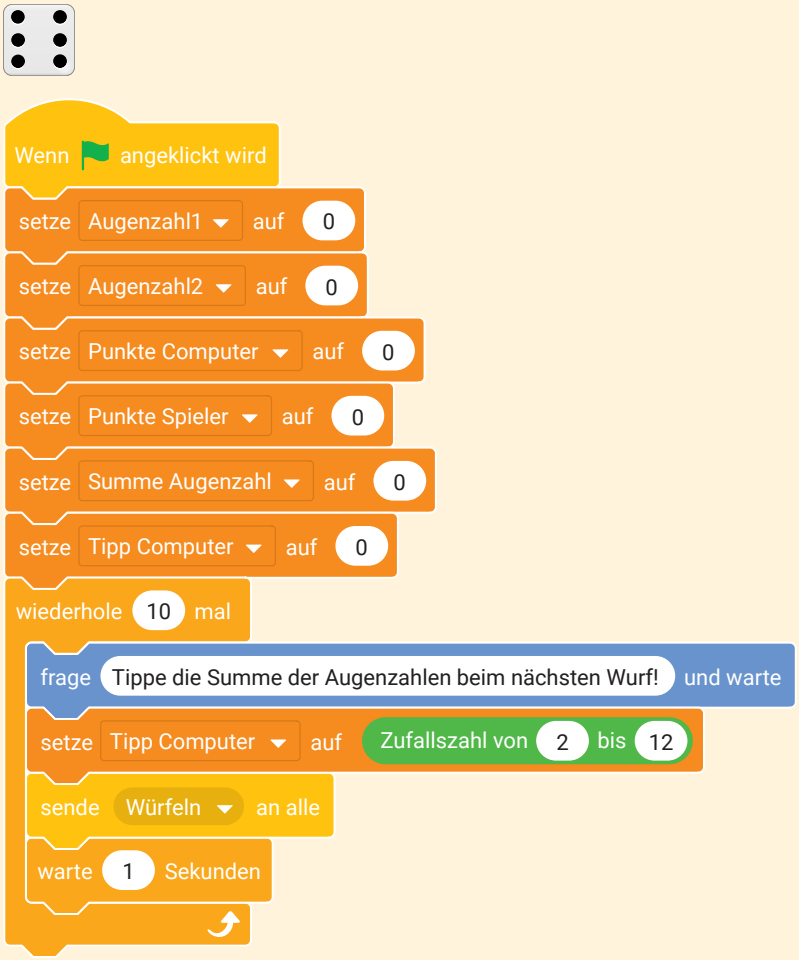
b) Sobald auf die grüne Fahne geklickt wird, sollen alle Variablen den Wert 0 annehmen.

Zufallszahlen

Aufgabe 2 – Tipps

Zehnmal soll

- eine Abfrage erscheinen, die den Spieler nach einem Tipp fragt,
- der Tipp des Computers mit Hilfe einer Zufallszahl gesetzt wird,
- die Nachricht „Würfeln“ an alle gesendet wird,
- eine Sekunde gewartet wird

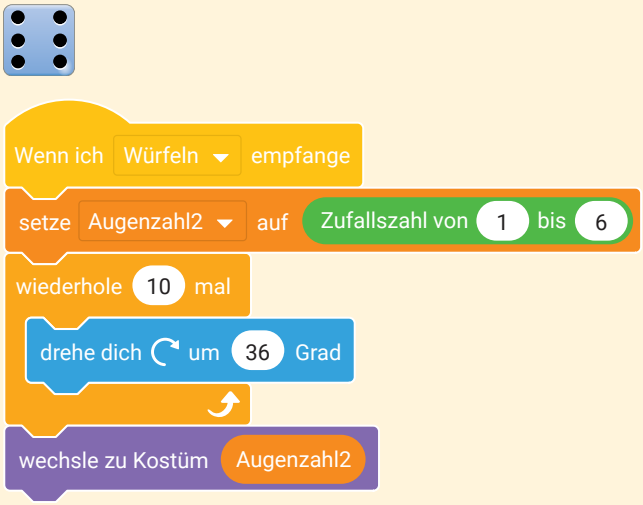


Aufgabe 3 – Würfeln

Wenn die Nachricht „Würfeln“ empfangen wird, soll die Variable Augenzahl2 auf eine Zufallszahl von 1 bis 6 gesetzt werden.

Das Würfeln wird symbolisiert, indem sich der Würfel in 10 Schritten um 360 Grad dreht.

Anschließend wird zum Kostüm gewechselt, das der gewürfelten Zahl entspricht.

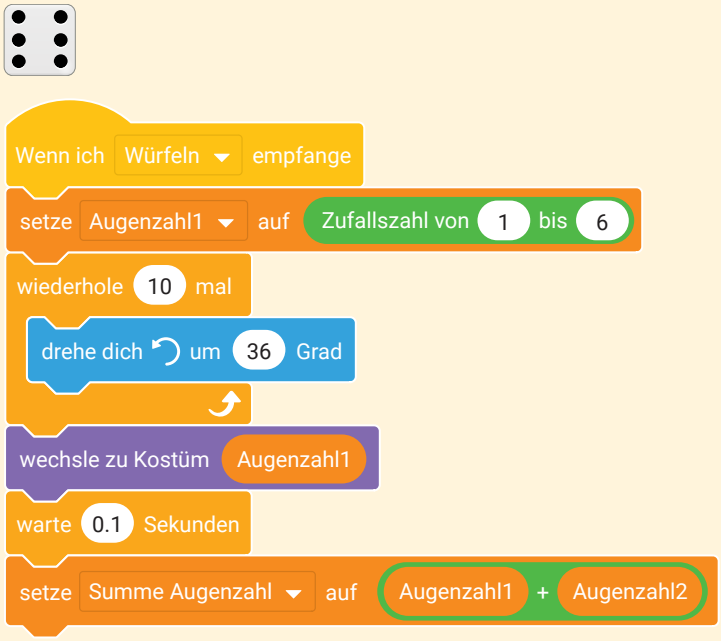


Zufallszahlen

Aufgabe 4 – Würfeln

Programmiere das Würfeln ebenso wie beim blauen Würfel.

Nach dem Kostümwechsel soll nach einer kurzen Wartezeit von 0.1 Sekunden die Variable „Summe Augenzahl“ auf die Summe der beiden gewürfelten Augenzahlen gesetzt werden.



Aufgabe 5 – Punktvergabe

Füge in das Skript aus Aufgabe 2 nach der Wartezeit zwei Abfragen ein:

Falls der Tipp des Spielers der gewürfelten Augenzahlsumme entspricht, soll die Punktzahl des Spielers um 1 erhöht und eine Meldung ausgegeben werden.

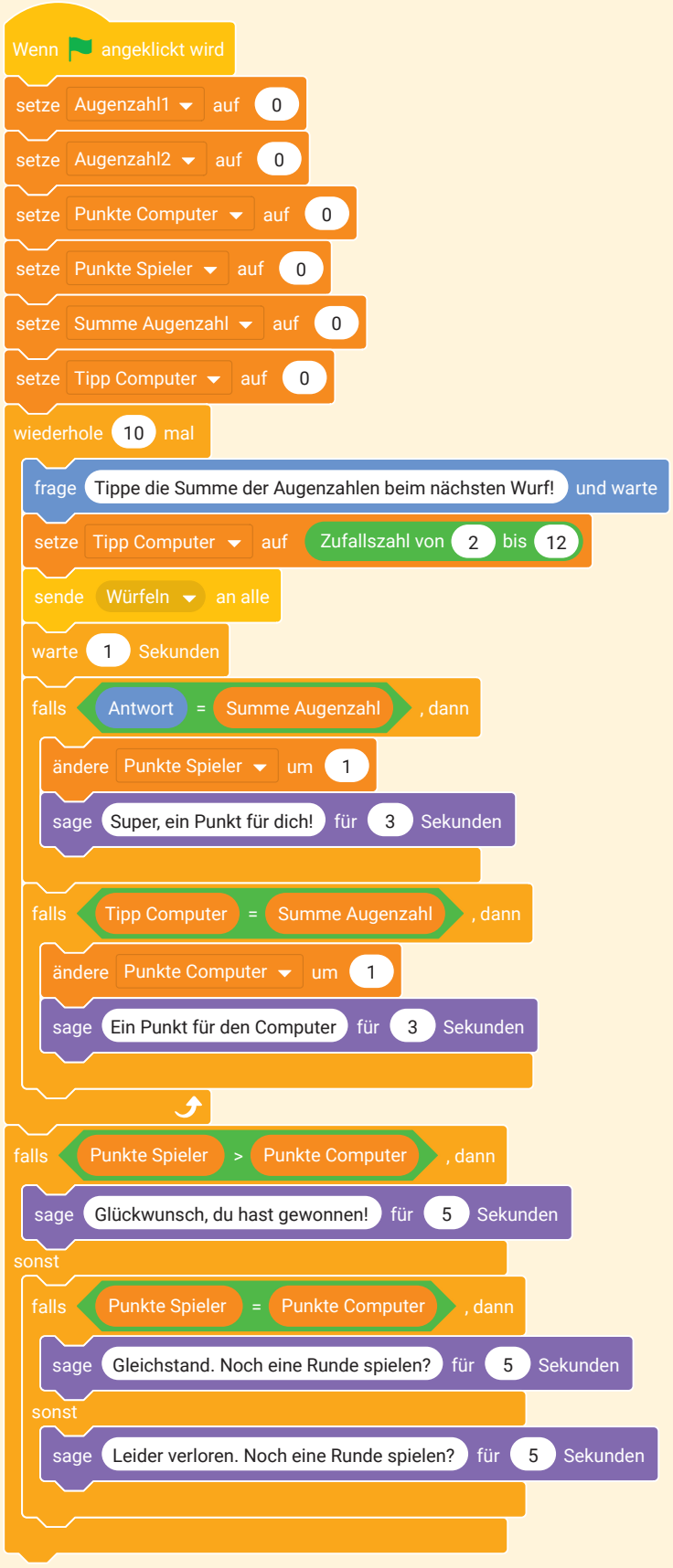
Falls der Tipp des Computers der gewürfelten Augenzahlsumme entspricht, soll die Punktzahl des Computers um 1 erhöht und eine Meldung ausgegeben werden.

Zufallszahlen

Aufgabe 6 – Ende des Spiels

Nach der zehnten Runde sollen die Punkte von Spieler und Computer verglichen und eine Meldung ausgegeben werden, falls

- der Spieler mehr Punkte hat,
- der Computer mehr Punkte hat,
- Spieler und Computer gleich viele Punkte haben



Zufallszahlen

Aufgabe 7

Warum ist der Spieler gegenüber dem Computer im Vorteil?

Der Tipp des Computers wird als Zufallszahl von 2 bis 12 erzeugt. Der Computer tippt dadurch alle Augenzahlsummen von 2 bis 12 theoretisch gleich häufig.

Der Spieler kann hingegen beim Tippen berücksichtigen, dass die Augenzahlsummen nicht gleich häufig gewürfelt werden. Dadurch ist der Spieler im Vorteil gegenüber dem Computer.

Und wie müsste das Skript verändert werden, damit dieser Vorteil beseitigt wird?

Um „Gerechtigkeit" herzustellen, müsste der Computer zwei einzelne Augenzahlen von 1 bis 6 ermitteln, die dann zur getippten Augenzahlsumme addiert werden.

Dadurch würden auch die Tipps des Computers der unterschiedlichen Häufigkeit der Augenzahlsummen entsprechen, die sich aus den addierten Augenzahlen ergibt.