Abb. 1: Bei Kartenspielen müssen zu Beginn die Karten zufällig ausgeteilt werden.

Wir arbeiten mit Zufallszahlen

Jedesmal wenn ein neues Patience-Spiel gestartet wird, muss das Computerprogramm die Karten zunächst austeilen (Abb. 1). Wenn es dabei die Karten nach einem fest vorprogrammierten Schema anordnen würde, ergäbe sich immer wieder dieselbe Anfangssituation. Das würde natürlich für den Spieler schnell langweilig werden. Deswegen werden die Karten von dem Programm zunächst gemischt; dabei greift es auf so genannte Zufallszahlen zurück. Das sind Zahlen, welche wie von einem Würfel zufällig erzeugt werden.

Auch bei anderen Spielprogrammen werden Zufallszahlen benutzt. Selbst bei strategischen Spielen wie z. B. Schach kommen sie zum Einsatz. Häufig kommt es nämlich vor, dass das Programm nicht nur einen, sondern gleich mehrere gleich gute Züge findet. In diesem Fall sucht es sich unter diesen wieder einen Zug mithilfe einer Zufallszahl aus.

Wir erzeugen Zufallszahlen

Zur Erzeugung von Zufallszahlen stellt JavaScript die Funktion

```
Math.random()
```

zur Verfügung. Diese liefert als Rückgabewert eine Zufallszahl zwischen 0 und 1. In vielen Fällen – etwa beim Würfeln – benötigt man nicht Dezimalzahlen zwischen 0 und 1, sondern ganze Zahlen – im Fall des Würfels zwischen 1 und 6. Derartige Zahlen kann man nun aber leicht mit der `random`-Funktion des Math-Objekts bilden.

Wie dies geschieht, wollen wir hier am Beispiel der Würfel-Funktion `wuerfel` zeigen. Diese Funktion soll als Rückgabewert eine ganze Zufallszahl zwischen 1 und 6 liefern. Die Idee ist: Zunächst multiplizieren wir den Rückgabewert der `random`-Funktion mit der Zahl 6; auf diese Weise erhalten wir (nicht ganzzahlige) Zufallszahlen zwischen 0 und 6. Diese Zahlen müssen jetzt noch gerundet werden. Allerdings müssen wir vorher noch die Zahl 0,5 addieren; ansonsten würden wir als Ergebnis manchmal auch die Zahl 0 erhalten (Abb. 2).

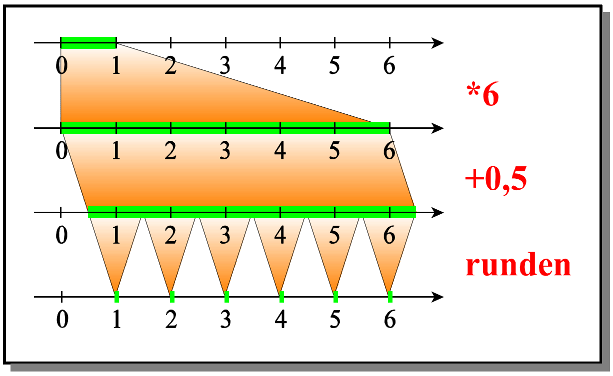

Abb. 2: So erhält man Würfelzahlen

```
function wuerfel()
{
     var z1 = Math.random();
     var z2 = z1*6 + 0.5;
     var z3 = Math.round(z2);
     return z3;
}
```

Aufgaben

1. Programmiere eine Funktion `wuerfel(m)`, die als Rückgabewert eine ganze Zufallszahl zwischen 1 und m besitzen soll. Teste sie aus.

2. Programmiere ein Zahlenrätselspiel.

Zufallszahlen auf dem Prüfstand

Sind die Zahlen, welche unsere Würfel-Funktion liefert, tatsächlich genauso zufällig wie von einem Würfel? Schaut man sich die gelieferten Zahlen der Reihe nach an, fällt eine Entscheidung schwer. Manchmal taucht eine Zahl gehäuft oder sogar auch direkt mehrfach hintereinander auf. Kann man das noch Zufall nennen? Immerhin müssen wir bedenken, dass derartige Ereignisse auch bei einem richtigen Würfel auftreten. Jeder hat schon einmal eine Glückssträhne beim Mensch-Ärgere-Dich-Nicht-Spiel gehabt und 3 mal hintereinander eine Sechs gewürfelt. Dies geschieht zwar nicht sehr häufig, aber es kommt vor.

Auf lange Sicht werden beim Würfeln aber alle Zahlen von 1 bis 6 (annähernd) gleich häufig auftauchen; ansonsten muss man davon ausgehen, dass der Würfel gezinkt ist. Haben die Zahlen, welche unsere `wuerfel`-Funktion liefert, auch diese Eigenschaft? Um dies zu untersuchen, rufen wir die Funktion `wuerfel` 6000 mal auf und kontrollieren nach, wie oft die einzelnen Ergebnisse von 1 bis 6 auftauchen. Natürlich führen wir diese Kontrolle nicht von Hand durch, sondern überlassen sie einem Programm:

```
function kontrolle(anzahl)
{
      var ergebnis = new Array();
      for (var i=1; i<=anzahl; i++)
      {
            zufallszahl = wuerfel();
            ergebnis[zufallszahl] = ergebnis[zufallszahl]+1;
      }
      var text = "";
      for (var j = 1; j<=6; j++)
      {
            text = text + j + ": " + ergebnis[j];
      }
      anzeigeform.anzeige.value = anzeigetext;
}
```

Das Programm merkt sich, wie oft jede Zufallszahl gebildet wird, und speichert die einzelnen Anzahlen in dem Array `ergebnis` ab. Dazu wird in der ersten Schleife bei jedem Schleifendurchlauf zunächst eine Zufallszahl gebildet. Wenn diese Zufallszahl z. B. 4 ist, wird nun der Wert des Elements `ergebnis[4]` um 1 erhöht.

In der zweiten Schleife werden die einzelnen Elemente des `ergebnis`-Arrays in einer Zeichenkette `text` gesammelt, damit sie in der Textarea `anzeige` angezeigt werden können. Abb. 3 zeigt ein solches Ergebnis. Wir sehen, dass jede Zufallszahl wie erwartet ungefähr 1000 mal auftaucht.

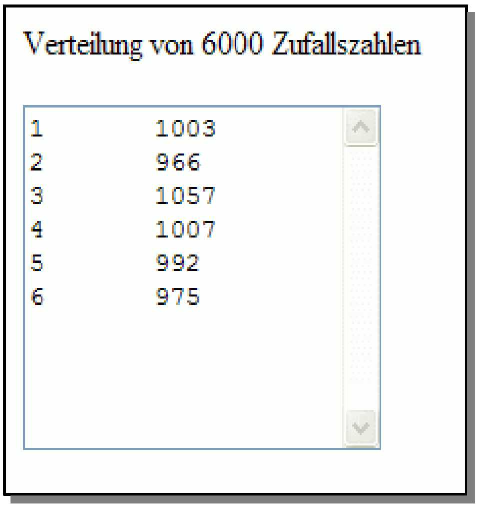

Abb. 3: Das Ergebnis von 6000 „Würfen"

Aufgaben

1. Die relative Häufigkeit der Zahl 6 in Abb. 3 ist $\frac{975}{6000} \approx 0{,}1625$. Ermittle die relative Häufigkeit dieser Zahl in verschiedenen, auch längeren Serien von Zufallszahlen. Vergleiche die Ergebnisse mit dem erwarteten Wert $\frac{1}{6}$.

2. Für das Auftreten eines Sechser-Paschs erwartet man eine relative Häufigkeit von $\frac{1}{36}$. Ermittle für längere „Wurfserien" die relative Häufigkeit für das Auftreten von zwei aufeinander folgenden Sechsen.

3. Facharbeit: Informiere dich über den so genannten Pokertest von Zufallszahlen. Stelle ihn dar und prüfe damit unsere Funktion `wuerfel`.

Wir erzeugen Zufallszahlen

Computer haben keine eingebauten Würfel; bei ihnen läuft – solange sie keinen Defekt haben – auch nichts zufällig ab. Vielmehr gehorcht alles einem festen Plan, so wie er in den Programmen festgelegt wurde. Deswegen kann ein Computer auch keine wirklich zufälligen Zahlen erzeugen. Tatsächlich greift er auch hier auf ein Programm zurück. Allerdings liefert dieses Programm Zahlen, bei denen man eine Regelmäßigkeit in der Abfolge nicht so einfach erkennen kann; außerdem sind die erzeugten Zahlen auf lange Sicht annähernd gleichmäßig verteilt. Aus diesem Grund spricht man in diesem Zusammenhang häufig auch von **Pseudozufallszahlen**.

```
function zufall(r, a)
{
     r = Math.pow(r+a,8);
     r = r- Math.floor(r);
     return r
}
```

Abb. 4: Ein Zufallszahlengenerator

Man könnte meinen, ein Programm, welches Pseudozufallszahlen erzeugen soll, muss kompliziert sein. Überraschenderweise gibt es einige Verfahren, die gar nicht so schwer sind. Eines davon wollen wir dir hier vorstellen. Es benutzt die Funktion `zufall` in Abb. 4. Für die erste Zufallszahl setzt man zwei Werte für r und a ein, z. B. a = 3,1315 und r = 0,5. In der ersten Zeile des Funktionsrumpfes wird der Ausdruck

$$(0{,}5 + 3{,}1315)^8 = 30247{,}4235676$$

berechnet. In der nächsten Zeile wird von dieser Zahl der ganzzahlige Anteil 30247 subtrahiert. Der Rückgabewert liefert dann schon die erste Pseudozufallszahl, in unserem Fall 0,4235676. Zur Bestimmung einer weiteren Pseudozufallszahl setzt man diese gerade bestimmte Zahl 0,4235676 wieder als Parameterwert für r in die Funktion `zufall` ein; der Wert des Parameters a bleibt. Auf diese Weise erhält man als neue Zufallszahl 0,2558625. Genauso erhält man mit dieser neuen Zufallszahl wieder eine neue und so verfährt man immer weiter.

Wählt man einen anderen Startwert r oder einen anderen Parameter a, so erhält man eine andere Serie von Zufallszahlen.

Aufgaben

1. Wieso liefert das geschilderte Verfahren zur Bestimmung von Pseudozufallszahlen immer Zahlen zwischen 0 und 1?

2. Berechne mit dem Taschenrechner eine Serie von 5 Zufallszahlen. Beginne mit den Werten r = 0,3 und a = 2,7881. Verfahre dabei jeweils auf zweierlei Weise:

2.1 Benutze keine gerundeten Werte; benutze lieber den Speicher des Taschenrechners.
2.2 Benutze auf 4 Stellen hinter dem Komma gerundete Werte. Gib die gerundeten Zahlen jeweils neu ein.
2.3 Versuche die Unterschiede bei den Ergebnissen von 2.1 und 2.2 zu erklären.
2.4 Dieselben Verfahren zur Berechnung von Pseudozufallszahlen führen bei den verschiedenen Browsern manchmal zu unterschiedlichen Ergebnissen. Woran könnte dies liegen?

3. Häufig benutzt man Datums- und Zeitangaben, um einen Startwert für die Berechnung von Pseudozufallszahlen zu gewinnen. Welchen Vorteil bietet diese Vorgehensweise?

4. Mit der Funktion `zufall` kann man auch eine Würfel-Funktion bilden. Prüfe für verschiedene Startwerte nach, ob es zu einer nahezu gleichmäßigen Verteilung der Rückgabewerte kommt.

5. Facharbeit: Suche in der Literatur oder im Internet nach weiteren Verfahren zur Erzeugung von Pseudozufallszahlen. Stelle eines dieser Verfahren dar und programmiere es. Teste auf Gleichverteilung. Suche auch nach einem weiteren Zufallszahlen-Test; stelle ihn dar und wende ihn sowohl auf die in JavaScript eingebaute Zufallsfunktion als auch auf das neue Verfahren an.

6. Facharbeit: Programmiere ein Glücksspiel. Schreibe auch eine Spielanleitung.

7. Facharbeit: Informiere dich über das „Gesetz der großen Zahlen“. Stelle es dar, führe dazu Versuchsreihen durch und gib die Ergebnisse auch in graphischer Form an.