

Leitprogramm "SHOOT-THE-BALLOON"

In dem Spiel „SHOOT-THE-BALLOON“ soll ein Luftballon abgeschossen werden. Der Spieler zielt mit einem Fadenkreuz auf den Ballon und versucht ihn zu treffen. Dem Spieler werden am Spielfeldrand die verbleibende Munition angezeigt sowie die erzielten Punkte angezeigt. Trifft der Spieler den Ballon, so erhält er 100 Punkte, trifft er daneben, so werden ihm 10 Punkte abgezogen (es kann also auch negative Punktestände geben.)

Der Ballon wird automatisch alle 800 Millisekunden an eine neue Position bewegt.

1. Ereignisse registrieren

Als erstes müssen wir Ereignisse registrieren, das heißt wir müssen festlegen, welche Unterprogramme bei welchem Ereignistyp aufgerufen werden. Die Syntax hierfür lautet:

```
document.getElementById("IdDesElements").
addEventListener("Ereignistyp", UnterprogrammWelchesGeladenWerdenSoll)
```

Registrierte die folgenden Ereignisse:

- Wird auf den Ballon geklickt (Ereignistyp "click"), hat der Spieler getroffen. Es soll das Unterprogramm getroffen aufgerufen werden.
- Wird auf das Spielfeld geklickt, hat der Spieler den Ballon verfehlt. Es soll das Unterprogramm nichtGetroffen aufgerufen werden.
- Wird auf den Text „Neues Spiel“ geklickt, soll ein neues Spiel gestartet werden. Dazu wird das Unterprogramm neuesSpiel aufgerufen.
- Wird die Maus über dem Spielfeld bewegt (Ereignistyp "mousemove"), muss das Fadenkreuz mitgeführt werden. Dies erledigt das Unterprogramm bewegeFadenkreuz.

2. Spiel und Spielfeld initialisieren

Um die verfügbare Munition sowie die erzielten Punkte zu verwalten, benötigen wir zwei Variablen. Diese sollen MUNITION und PUNKTE heißen und müssen mit Anfangswerten belegt werden. Der Spieler startet mit 10 Kugeln und hat 0 Punkte. Danach müssen wir uns noch um die Spieloberfläche kümmern, denn das Spielfeld wird im Moment noch von der Game-Over-Box verdeckt. Um diese auszublenden, müssen wir das Style-Attribut display dieser Box auf den Wert "none" setzen. Zur Erinnerung – Style-Attribute werden folgendermaßen gesetzt:

```
document.getElementById("IdDesElements").style.styleAttribut="Attributwert"
```

Anschließend müssen wir noch die Grundlage dafür schaffen, dass sich der Ballon später bewegt. Das Unterprogramm bewegeBallon wird bei jedem Aufruf den Ballon auf eine zufällige Position rücken. Damit bewegeBallon alle 800ms einmal ausgeführt wird, müssen wir einen Intervall-Timer einrichten:

```
setInterval(nameDesUnterprogramms, ZeitdauerInMillisekunden)
```

	Spieleprojekte mit JavaScript umsetzen
	Programmieren mit JavaScript

Zu guter Letzt sollen Punkte und Munition auch auf der Benutzeroberfläche angezeigt werden. Hierfür muss einmal das Unterprogramm `aktualisiereOberflaeche` aufgerufen werden.

3. Das Unterprogramm `aktualisiereOberfläche` schreiben

Das Unterprogramm `aktualisiereOberflaeche` soll die Benutzeroberfläche verwalten. Bei jedem Aufruf sollen Punkte und Munition innerhalb der entsprechenden Elemente der Webseite angezeigt werden. Zur Erinnerung – den Textinhalt eines HTML-Elementes verändert man wie folgt:

```
document.getElementById("IdDesElements").innerHTML = Wert
```

Zusätzlich muss überprüft werden, ob der Spieler überhaupt noch über Munition verfügt. Denn ist dies nicht der Fall, so hat der Spieler verloren und die Game-Over-Box muss eingeblendet werden. Hierzu setzen wir deren Style-Attribut `display` auf den Wert `""`. Außerdem soll dem Benutzer dann in dem Absatz mit der id `"punkteText"` der erzielte Punktestand mit passendem Text angezeigt werden (z.B.: Du hast 100 Punkte erzielt!).

Wenn du die Webseite jetzt startest, sollten bereits Punkte und Munition korrekt angezeigt werden. Die Game-Over-Box sollte nun nicht mehr zu sehen sein.

4. Die Unterprogramme `getroffen` und `nichtGetroffen` schreiben

Im Unterprogramm `getroffen` fügen wir als Erstes die Zeile `ereignisObjekt.stopPropagation();` hinzu. Diese Anweisung sorgt dafür, dass das Click-Ereignis, das auf dem Ballon registriert wurde nicht an das Spielfeld weitergegeben wird. Normalerweise würde dies passieren, da das Spielfeld-Element den Ballon beinhaltet (Wer auf den Ballon klickt, hat ja damit gleichzeitig auch in das Spielfeld geklickt). Für uns wäre diese Eigenheit aber unpraktisch, da wir einen Klick auf das Spielfeld als „nicht getroffen“ zählen möchten.

Zudem soll unter der Voraussetzung, dass der Spieler noch über Munition verfügt hat, ein zusätzlicher Schuss dem Munitionsvorrat hinzugefügt, sowie der Punktestand um 100 erhöht werden. Die Voraussetzung, dass der Spieler noch Munition hatte ist wichtig, damit das Spiel nicht weiterläuft, wenn der Spieler keine Munition mehr hat.

Als letztes muss noch das Unterprogramm `aktualisiereOberflaeche` aufgerufen werden, damit die veränderten Werte auch angezeigt werden.

Um das Unterprogramm `nichtGetroffen` zu schreiben, können wir sehr ähnlich vorgehen. Hier wird dann allerdings der verbrauchte Schuss vom Munitionsvorrat abgezogen und die Punktzahl um 10 vermindert. Die Anweisung `StopPropagation` ist hier nicht notwendig (würde aber auch nicht stören).

Wenn du alles richtig gemacht hast, kannst du jetzt das Spiel bereits „spielen“. Allerdings wird sich weder der Ballon, noch das Fadenkreuz bewegen, es wird daher nicht schwierig sein, einen Treffer zu landen.

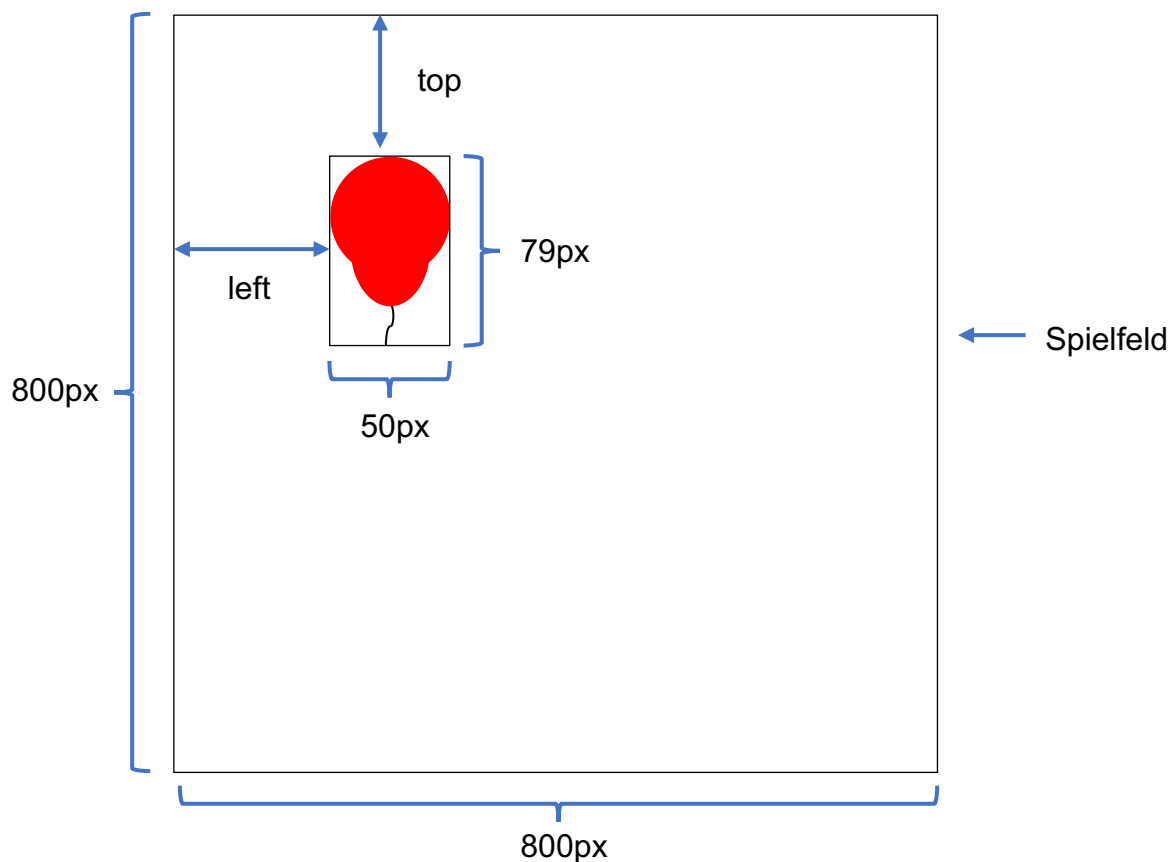
5. Das Unterprogramm `neuesSpiel` schreiben

Das Unterprogramm `neuesSpiel` wird aufgerufen, wenn auf die Schaltfläche „Neues Spiel“ angeklickt wird. Da auch diese Schaltfläche über dem Spielfeld liegt, müssen wir wie im vorherigen Abschnitt die Weitergabe des Click-Ereignisses unterbinden.

Darüber hinaus müssen wir die Variablen PUNKTE und MUNITION auf ihre Startwerte zurücksetzen, die Oberfläche aktualisieren (damit beides auch korrekt angezeigt wird), sowie die Game-Over-Box wieder ausblenden (siehe 2.).

6. Das Unterprogramm bewegeBallon schreiben

Das Ballon-Objekt hat in CSS das Style-Attribut `position: absolute` erhalten. Damit kann dieses innerhalb seines Eltern-Elements (also der Spielfläche) frei positioniert werden. Diese Positionierung wird mit den Style-Attributen `top` und `left` vorgenommen, die den Abstand des Elements zum oberen bzw. linken Rand des beinhaltenden Elements in Pixeln beziffern:



Damit der Ballon eine zufällige Position auf dem Spielfeld einnimmt, lassen wir die Werte für `top` und `left` von einem Zufallsgenerator erzeugen. Überlege genau, in welchen Zahlenbereichen die Werte jeweils liegen müssen.

Zur Erinnerung – eine ganzzahlige Zufallszahl zwischen 0 und 99 erzeugt man zum Beispiel mit der folgenden Funktion:

```
Math.floor(Math.random()*100)
```

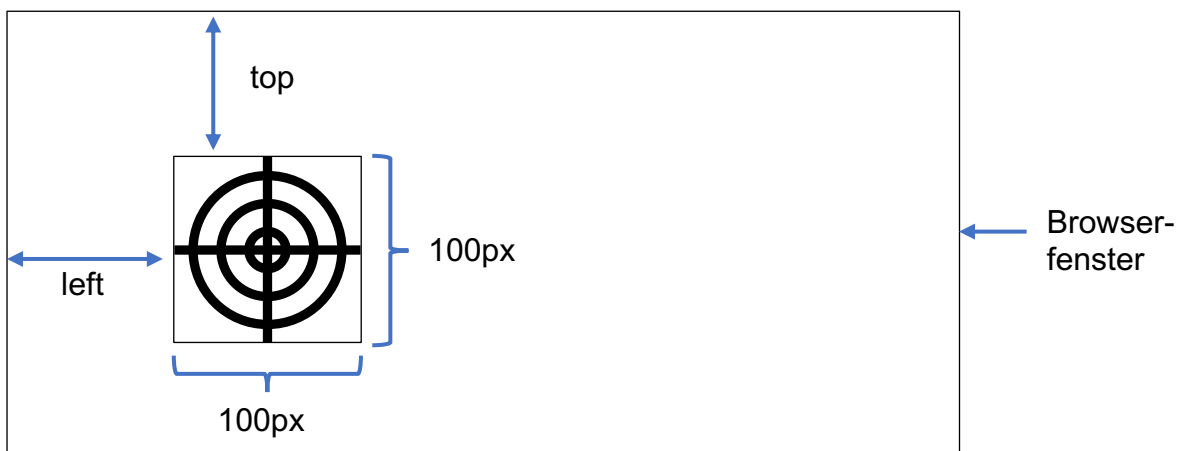
Ein Style-Attribut verändert man auf die folgende Weise:

```
document.getElementById("IdDesElements").style.styleAttribut = Wert
```

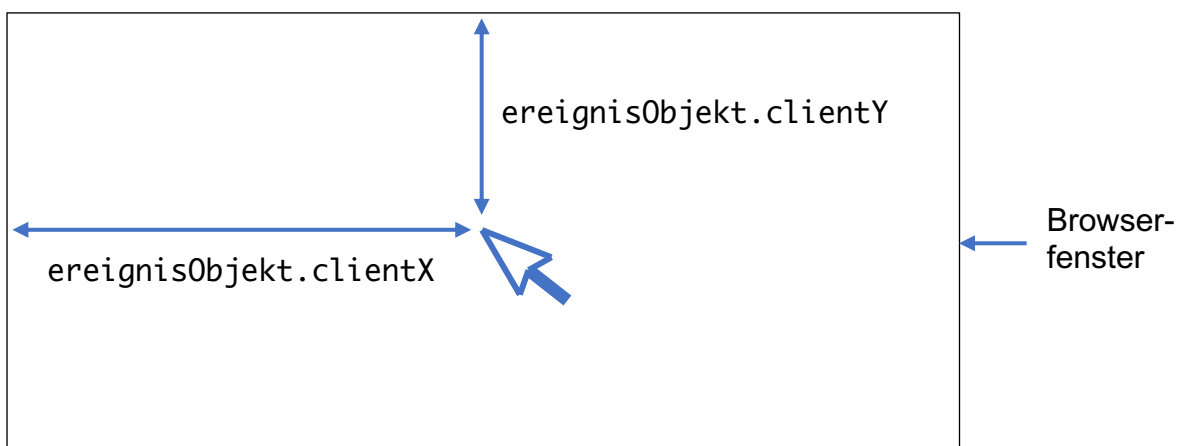
Beachte auch noch, dass der Ballon nur dann bewegt wird, wenn der Spieler noch nicht verloren hat (Er muss also noch über Munition verfügen).
Wenn du alles richtig gemacht hast, bewegt sich der Ballon über die Spielfläche, sobald du das Spiel ausprobierst.

7. Das Unterprogramm `bewegeFadenkreuz` schreiben

Im Gegensatz zum Ballon-Objekt liegt das Fadenkreuz in der HTML-Struktur nicht im Spielfeld sondern innerhalb des `body`-Elements. Wir können dieses also mit den Style-Attributen `left` und `top` im gesamten Browser-Fenster frei positionieren. Diesen Vorteil nutzen wir aus, um das Fadenkreuz immer so zu verschieben, dass dessen Mittelpunkt exakt über der Spitze des Mauszeigers liegt.



Bei der Programmierung der vorherigen Unterprogramme ist dir mit Sicherheit aufgefallen, dass wir bei einigen Unterprogrammen einen Parameter unter dem Bezeichner `ereignisObjekt` entgegennehmen. Dies liegt daran, dass JavaScript immer dann, wenn Unterprogramme aufgrund von Ereignissen aufgerufen werden, ein sogenanntes Ereignis-Objekt erzeugt und an das entsprechende Unterprogramm übergibt. Dieses enthält Informationen zu dem ausgelösten Ereignis – unter anderem auch die Position des Mauszeigers zum Zeitpunkt der Auslösung des Ereignisses. Indem wir in dem Unterprogramm einen Parameter anlegen, können dieses Ereignis-Objekt im Unterprogramm verwenden. Die Position des Mauszeigers kann dann folgendermaßen ausgelesen werden:



	Spieleprojekte mit JavaScript umsetzen
	Programmieren mit JavaScript

Mit diesen Informationen sind wir in der Lage, das Fadenkreuz bei jeder Mausbewegung so zu positionieren, dass die Fadenkreuzmitte exakt über der Spitze des Mauszeigers liegt. Ob du dies korrekt umgesetzt hast, siehst du, wenn du das Spiel ausprobierst. Zum Schluss sorgen wir nun noch dafür, dass der Mauszeiger nicht mehr angezeigt wird sobald sich die Maus über dem Spielfeld befindet. Dies bewerkstelligen wir, indem wir in der CSS-Datei die Kommentarklammern von der Anweisung `cursor: none;` in `div.spiel Feld` entfernen.

Arbeitsauftrag für Schnelle (also nur bearbeiten, wenn bereits alles funktioniert)

Überlege dir mögliche Erweiterungen für dein Spiel und setze diese um.

Mögliche Ideen:

- ein zweiter Ballon (oder ein anderes Objekt)
- einstellbares Bewegungstempo
- der Smiley bewegt sich flüssig (springt also nicht hin und her)
- von Zeit zu Zeit Einblendung von Sonderzielen, die besonders viele Punkte oder zusätzliche Munition geben