

# JAVASCRIPT SPIELE

## AUFGABE - SPIEL HANDYWEITWURF ANGABE 1



- In diesem Spiel sollst du mit deinem Handy ein Monster treffen
- Du musst dazu den Winkel und die Kraft eingeben mit der du das Handy wirfst um das Monster zu treffen
- Die Entfernung zum Monster muss immer gleich sein
- Drei Versuche darf es geben
  - für Fortgeschrittene: die Anzahl an Versuchen kann vom User eingegeben werden
- Definiere die Fallbeschleunigung
  - Für Fortgeschrittene: Lass den User eingeben ob er auf der Erde, dem Mond, Mars oder Jupiter ist. Dementsprechend wird die Fallbeschleunigung zur Berechnung genommen
- Gib dem User aus ob er getroffen hat oder wie weit weg das Monster noch ist und wie weit er geworfen hat
- Achte auf ein angenehmes Styling der Seite

## AUFGABE - SPIEL HANDYWEITWURF ANGABE 2



- let fallingSpeed steht für die Fallbeschleunigung der Wert fallingSpeed beträgt auf der
  - Erde 9.81 m/s²
  - Mond 1.62 m/s<sup>2</sup>
  - Mars 3.69 m/s²
  - Jupiter 24.79 m/s²
- let speed ist die Anfangsgeschwindigkeit und ist mit einem Formular und einem Inputfeld vom User einzugeben
- let angle ist der Winkel in dem der User wirft. Es darf nur eine Zahl zwischen 1 und 90 sein. Dieser Wert ist auch mit einem Inputfeld in einem Formular vom User einzugeben
  - Die Berechnung die du hier brauchst:
     angle = angle \* (Math.PI / 180)
- let throwingDistance berechnet sich aus dem Wurfwinkel und der Anfangsgeschwindigkeit
  - Die Berechnung sieht folgendermaßen aus: throwingDistance = ((speed \* speed) \* Math.sin(2 \* angle)) / fallingSpeed Achtung: Runde das Ergebnis
- let distanceToMonster speichert den Wert der zufällig gewählt wird. Er stellt die Entfernung zum Monster dar
  - distanceToMonster = Math.random() \* 90 + 10;
     Mit dieser Zeile generierst du eine zufällige Zahl
  - distanceToMonster = Math.round(distanceToMonster)
  - Hier wird der Wert noch gerundet um eine ganze Zahl zu erhalten

#### Weitere hilfreiche Angaben

- Math.round() rundet eine Zahl
- if(throwingDistance == distanceToMonster) { ... }
   fragt ab ob der User getroffen hat
- Math.PI gibt die Zahl PI an
- Math.sin berechnet den Sinus

## 02 - SCHERE, STEIN PAPIER



Jeder kennt das Spiel. Nun versuch daraus ein Browserspiel zu machen ;)

Variablen die du dazu benötigst:

- let symbolPC;let symbolUser;let pcCounter;let userCounter;let winCondition = 3;
  - Für Fortgeschrittene: Der User kann mittels einer Eingabe in ein Inputfeld selbst definieren über wie viele Runden er spielen möchte

### Was passiert:

- 1. Der User wählt ein Symbol
- Der Computer denkt sich jeder Runde aus, was er macht (Math.random())
- 3. Es wird dann verglichen, wer die Runde gewonnen hat. Haben beide das gleiche Symbol, ist die Runde unentschieden
- 4. Regel:
  - Schere gewinnt gegen Papier
  - Papier gewinnt gegen Stein
  - Stein gewinnt gegen die Schere
- 5. Hat einer drei Mal gewonnen, wird das Spiel beendet

**Tipp:** Du wirst hier oft die if-Verzweigung brauchen Achtung: Vergiss nicht die Seite in eine ansehnliche Form zu bringen, so dass man das Spiel auch gerne spielt!

#### 03 - ZAHLEN RATEN



## Errate eine ausgedachte Zahl

- Der User legt eine obere Grenze der Zahl fest und der Computer wählt eine beliebige Zahl zwischen 0 und der vom User definierten oberen Grenze.
- Für dieses Spiel brauchst du nur drei Variablen:
  - let upperLimit;
  - let guessNumber;
  - let counter;
- Die Anzahl der Versuche soll der User ebenfalls selbst festlegen können
- Du benötigst für dieses Spiel insgesamt:
  - Vier if-Verzweigungen
  - Drei Variablen

#### Weitere hilfreiche Angaben

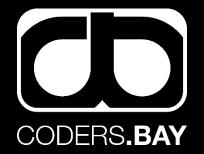
- Math.round() rundet eine Zahl
- Math.random() legt eine zufällige Zahl fest

### 04 - CODE BREAKER



In diesem Spiel sollst du einen 3 stelligen Zahlencode erraten

- Dafür hast du ein Input Feld mit einem mindestwert von 111 und einem Maximalwert von 999
- Der Zahlencode generiert sich automatisch
  - Tipp: du generierst drei zahlen zwischen 1 und 9
- Im JavaScript
  - du holst dir den Wert aus dem Inputfeld und speicherst alle drei Werte in eine eigene Variable
    - dazu benötigst du die Methode .charAt()
- du benötigst außerdem Variablen für:
  - einen Zähler für die Versuche (maximal dürfen es 12 Versuche sein)
  - die richtig geratene Stelle
  - die richtig geratene Zahl
- du vergleichst die einzelnen geratenen Zahlen mit den einzelnen Zahlen des Geheimcodes
  - folgende Ausgaben gibt es:
    - die Runde
    - · wie viele Zahlen richtig geraten sind
    - · wie viele Stellen richtig geraten wurden



# **E**NDE