

```

1 <?php
2     #print_r($_POST); # Testausgabe der übergebenen Werte
3
4     $datei = "spiel.json";
5
6     /* for($i = 0; $i<100; $i++)
7     {
8         echo rand(0,1); # Testversuch für Zufallszahl
9     } */
10
11     if (isset($_POST["aufstellung"])) # Spieler will sich registrieren
12     {
13         registrierung($datei);
14     }
15     else if (isset($_POST["spielzug"])) # Spielzug setzen
16     {
17         #echo "Spielzug: ".$_POST["spielzug"]."\n"; # Testausgabe
18         $spieldaten = file_get_contents($datei); # Dateiauslesen
19         $json = json_decode($spieldaten); # JSON String decodieren -> Objekt
20         $spiel = new Spiel($json->spieler1, $json->spieler2, $json->token, $json->aufstellung1
21             , $json->aufstellung2); # Spielobjekt neu erzeugt mit Parametern
22         $check = explode(";", $spiel->token);
23
24         if ($check[0] === "Spieler1") # Spieler1 dran -> Tokenwechsel
25         {
26             #echo $_POST["spielzug"].$_POST["spielerID"]."???\n"; # Testausgabe
27             #print_r($array); # Testausgabe
28             #echo $_POST["spielzug"]; # Testausgabe
29             $spiel->token = "Spieler2"; # Wechsel des Tokens an ID 2. Spieler
30         }
31         else if ($check[0] === "Spieler2") # Spieler2 dran -> Tokenwechsel
32         {
33             #echo $_POST["spielzug"].$_POST["spielerID"]."!!!\n"; # Testausgabe
34             #print_r($array); # Testausgabe
35             #echo $_POST["spielzug"]; # Testausgabe
36             $spiel->token = "Spieler1"; # Wechsel des Tokens an ID 1. Spieler
37         }
38
39         $auswertung = auswerten($_POST["spielerID"], $_POST["spielzug"], $spiel, $datei); #
40         Arraybearbeitung
41         $spiel->token .= ";" . $_POST["spielzug"];
42         echo $auswertung;
43         #echo "Spielzug erfolgt."; # Testausgabe
44         # Auswertung ob (W)asser / (T)reffer / (V)ersenkt
45
46         $json = json_encode($spiel); # JSON String generieren
47         file_put_contents($datei, $json); # in Datei schreiben
48     }
49     else if (isset($_POST["spielerID"])) # Tokenabfrage
50     {
51         $spieldaten = file_get_contents($datei); # Dateiauslesen
52         $json = json_decode($spieldaten); # JSON String decodieren -> Objekt
53         echo $json->token; # Token zurückgeben (SpielerID;Spielzug)
54     }
55     else # Spieler1 will Spieler2 Namen
56     {
57         $spieldaten = file_get_contents($datei); # Dateiauslesen
58         $json = json_decode($spieldaten); # JSON String decodieren -> Objekt
59         echo $json->spieler2; # Spieler2 zurückgeben (Namen)
60     }
61
62     function registrierung($datei) # Registrierung in spiel.json - Datei
63     {
64         if (!file_exists($datei)) # Datei existiert nicht -> Spieler1 mit Aufstellung1
65             reinschreiben
66         {
67             $json = json_decode($_POST["aufstellung"]); # JSON String decodieren -> Array
68             $spiel = new Spiel($_POST["spieler"], null, null, $json, null); # Objekt Spiel
69             erzeugen mit Parametern

```

```

66     $json = json_encode($spiel); # JSON String generieren
67     #print_r($spiel); # Testausgabe
68     file_put_contents($datei, $json); # in Datei schreiben
69     echo "Spieler1;registriert."; # Ausgabe für JS
70 }
71 else # Spieler2 will sich registrieren
72 {
73     $spieldaten = file_get_contents($datei); # Dateiauslesen
74     $json = json_decode($spieldaten); # JSON String decodieren -> Objekt
75
76     # echo "Spieler1: ".$json->spieler1."\n"; # Wertzugriff mit Schlüssel ->
    Testausgabe
77     $spiel = new Spiel($json->spieler1, $json->spieler2, $json->token, $json->
    aufstellung1, $json->aufstellung2); # Spielobjekt neu erzeugt mit Parametern
78
79     if ($json->spieler2 !== null)
80     {
81         echo "Spiel bereits im Gange."; # Ausgabe für JS
82     }
83     else
84     {
85         $spiel->spieler2 = $_POST["spieler"];
86         $json = json_decode($_POST["aufstellung"]); # String in Array umwandeln
87         $spiel->aufstellung2 = $json; # Array eintragen
88         #print_r($spiel); # Testausgabe
89         echo "Spieler2;registriert.;Gegner."; # Ausgabe für JS, getrennt durch ";"
90         echo $spiel->spieler1 . " "; # Spieler 1 Namen zurückgeben
91         #$spiel->token = "test123"; # Testeintrag in JSON
92
93         if (rand(0, 1) === 0) # Zufallsentscheid, wer anfängt von 0 bis 1 (Ganzzahl)
94         {
95             $spiel->token = "Spieler1"; # ID 1. Spieler
96         }
97         else
98         {
99             $spiel->token = "Spieler2"; # ID 2. Spieler
100         }
101         $spiel->token .= " "; # danach kommt letzter gemachter Zug, bei Erstzug leer
102
103         echo $spiel->token; # Rückgabe des Tokens
104         $json = json_encode($spiel); # JSON String generieren
105         file_put_contents($datei, $json); # in Datei schreiben
106         #echo rand(0,1); # Spieler bestimmen der anfängt
107     }
108 }
109 }
110
111 class Spiel # Vorlage für die Speicherung in JSON-Format
112 {
113     public $spieler1;
114     public $spieler2;
115     public $token; # 1. Stelle: Spieler der dran ist; 2. Stelle: letzter Spielzug
    ("Spieler1" oder "Spieler2" -> IDs)
116     public $aufstellung1;
117     public $aufstellung2;
118
119     function __construct($spieler1, $spieler2, $token, $aufstellung1, $aufstellung2)
120     {
121         $this->spieler1 = $spieler1; # Spielernamen
122         $this->spieler2 = $spieler2; # Spielernamen
123         $this->token = $token;
124         $this->aufstellung1 = $aufstellung1; # Schiffe
125         $this->aufstellung2 = $aufstellung2; # Schiffe
126     }
127 }
128
129 function auswerten($SID, $spielzug, $spiel, $datei) # SID = spielerID
130 {
131     # Logik mit return der Meldung

```

```

132 #echo "Spielzug: ".$spielzug; # Testausgabe
133 #print_r($schiffsammlung); # Testausgabe
134 $status = "W"; # (W)asser
135 $schiffsammlung;
136
137 if ($SID === "Spieler1") # Spieler1 schießt auf Aufstellung von Spieler2
138 {
139     $schiffsammlung = $spiel->aufstellung2;
140 }
141 else # Spieler2 schießt auf Aufstellung von Spieler1
142 {
143     $schiffsammlung = $spiel->aufstellung1;
144 }
145
146 $leer = true;
147
148 foreach ($schiffsammlung as &$schiff) # & => Referenz
149 {
150     foreach ($schiff as &$feld) # & => Referenz
151     {
152         # prüfen ob getroffen; übergebe Array-Index -> $treffer
153         if (($treffer = array_search($spielzug, $schiff)) !== false)
154         {
155             #print_r($schiff); # Testausgabe
156             #echo "Treffer: ".$treffer; # Testazsgabe
157             unset($schiff[$treffer]); # Feld aus Schiff-Array nehmen
158             $schiff = array_values($schiff); # Entfernen von Index (alle)
159             $status = "T"; # (T)reffer
160
161             if (count($schiff) === 0) # prüfen ob keine mehr beschießbaren Felder
162             {
163                 $status = "V"; # (V)ersenkt
164             }
165             #print_r($schiff); # Testausgabe
166         }
167     }
168     if ($leer) # solange leer ist überprüfe, ob Schiff leer
169     {
170         $leer = empty($schiff); # nicht leeres Schiff gefunden
171     }
172     #print_r($schiffsammlung); # Testausgabe
173 }
174
175 if ($leer)
176 {
177     $status = "G"; # (G)ewonnen
178 }
179
180 if ($SID === "Spieler1") # Aktualisierung der Schiffsammlung Spieler2
181 {
182     $spiel->aufstellung2 = $schiffsammlung;
183 }
184 else # Aktualisierung der Schiffsammlung Spieler1
185 {
186     $spiel->aufstellung1 = $schiffsammlung;
187 }
188
189 $json = json_encode($spiel); # JSON String generieren
190 file_put_contents($datei, $json); # in Datei schreiben
191
192 return $status;
193 }
194 ?>
195

```