
3000
68
HUGO
TWELVE
MAX
I
II
1
0

Master vs. Fighters

- Представяне:
- Master ще се бие срещу група от Fighters.
- Всеки Fighter от групата има умения изразени в числова стойност: Здраве, Атака, Защита.
- Уменията за Fighters са фиксирани в 3 групи (a, b, g)
- Fighters могат да бъдат различен брой при всяко завъртане на играта и с различна последователност на уменията. Например 3 Fighters се изправят срещу Master, като уменията на Fighters са всички от типа „g“.
- Master от своя страна има променливи умения в граници.
- След като въведем списъка с уменията на Fighters които ще се изправят срещу Master, въвеждаме уменията за тази битка на Master и трябва да получим резултата и статистиката от битката.

Условия

- Първи удар нанася първия Fighter от списъка.
- Ако той е с умения „а“ = [‘здраве’ = > 5, ‘атака ‘=> 6, ‘защита‘=> 2];
- Уменията на Master в този двубой са „Master“ = [14, 6, 4];
- ‘а’ нанася удар със сила 6, здравето на Master (14) е подложено на изпитание, но е и защитено със (4).
- Сметката е следната Master (здраве) + Master (защита) – Fighter (атака) = Резултат върху здравето на Master след първия удар на Fighter. 14+4-6 = 12,
- 12 е новата стойност на позиция ‚Здраве‘ в масива Master.
- Следва удар от Master към Fightera. Сметката е аналогична.
- Завъртаме докато един от двамата не загуби всички точки Здраве.
- Ако загуби Master, играта приключва. Ако загуби Fighter, завъртаме отново същата логика със следващия Fighter от списъка. Важно е, че Master не рестартира началните си точки в полето ‚Здраве‘, а продължава битката с каквото му е останало.

Резултати и статистика

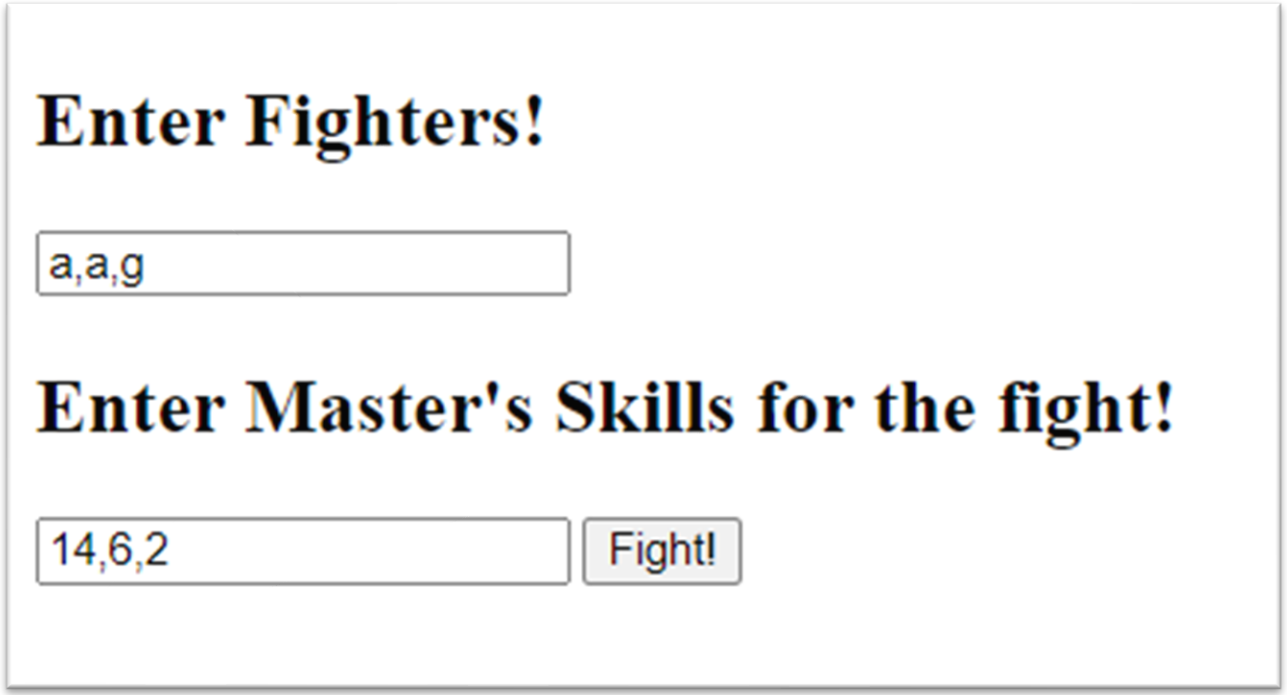
- След края на играта трябва да изведем статистика съдържаща следните данни:
- Победител – Master / Fighters
- Брой победени Fighters
- Брой оцелели Fighters
- Брой на точките ‚здраве‘ на Master останали след битките
- Броя на нанесените от Fighters удари
- Броя на нанесените от Master удари
- Средното поражение което е нанасяло един удар на Fighters
- Средното поражение което е нанасял един удар на Master

Описание на участниците

Role	Health	Attack	Resistance
Master	From 1 to 20	From 6 to 10	From 1 to 5
a	5	6	2
b	6	8	2
g	8	6	5

Резюме на кода

- 1. Форма с input за fighters a, a, g , втора за уменията на master-а 14, 6, 4 , input type="submit".



Резюме на кода

- 2. Проверяваме дали полетата са празни.

Резюме на кода

- 3. Форматираме постъпилите данни, като премахваме празни пространства, кавички, квадратни скоби.

```
10      $arr_f = str_replace('"', "", preg_replace('/[\[(\s+)\]]/', '', explode(',', $_POST['fighters_on_ring']))); //Cut what we don't need from the data.
11      $arr_m = str_replace('"', "", preg_replace('/[\[(\s+)\]]/', '', explode(',', $_POST['master_skill'])));
```

- Експолоудвам получения масив.

```
Master array!Array
(
    [0] => 14
    [1] => 6
    [2] => 2
)

Fighters array!Array
(
    [0] => a
    [1] => a
    [2] => g
)
```

Получаваме 2 масива. Единият с уменията на Master, а другия с комбинацията от умения
на Fighters.
Сега за Master имаме стойности с които можем да смятаме, но за Fighters имаме масив от букви.
Трябва да намерим как към всяка буква да присвоим набора от умения който и съответства.
Аз обаче искам да заместя числовия индекс с името на умението за което отговаря 0 = health, 1 = attack, 2 = resistance.

Резюме на кода

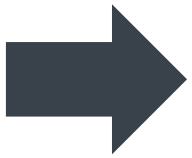
4. Променяме индекса на съответното умение за което отговаря със следния код:

```
14  ////////////////////////////////////////////////////////////////
15      $new_key = [0 => 'helth', 1 => 'attack', 2 => 'resistance'];//
16      $master = [];                                               //
17          foreach ($arr_m as $key => $value) {                    //
18              $master[$new_key[$key]] = $value;                   //
19          }                                                       //
20  ////////////////////////////////////////////////////////////////
```

Получаваме следната промяна:

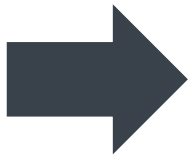
```
Master array!Array
(
    [0] => 14
    [1] => 6
    [2] => 2
)

Fighters array!Array
(
    [0] => a
    [1] => a
    [2] => g
)
```



```
Master array!Array
(
    [helth] => 14
    [attack] => 6
    [resistance] => 2
)

Fighters array!Array
(
    [0] => a
    [1] => a
    [2] => g
)
```



Сега, с цикъл и условие ще проверяваме стойността
за всеки индекс от масива на Fighters и ако той е равен на „a“ ще му присвояваме стойности, така ще тестваме и за „b“ и „g”

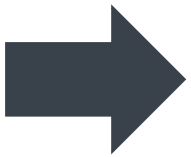
Резюме на кода

5. Създаваме 3 масива за уникалните групи от умения на Fighters:

```
29  ////////////////////////////////////////////////////////////////
30      $a = ['helth' => 5,'attack' => 6,'resistance' => 2,];      //
31      $b = ['helth' => 6,'attack' => 8,'resistance' => 2,];      //
32      $g = ['helth' => 8,'attack' => 6,'resistance' => 5,];      //
33  ////////////////////////////////////////////////////////////////
```

Присвояваме уменията към съответстващата буква в получения масив от метода POST.
За целта ще създадем първо един празен масив с името $fighters_on_ring, и една променлива брояща елементите на масива получени с метода POST.

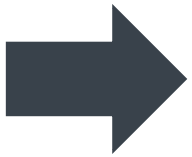
```
35  ////////////////////////////////////////////////////////////////
36  $count = count($arr_f);                                        //
37  $fighters_on_ring = [];                                        //
38                                                                 //
39   for($i = 0; $i < $count;$i++){                                //
40      $fighters_on_ring[$i] = [];                                //
41          for ($j = 0; $j <= 3;$j++){                            //
42              if($arr_f[$i] == 'a'){                             //
43                  $fighters_on_ring[$i] = $a;                    //
44              } elseif ($arr_f[$i] == 'b') {                     //
45                  $fighters_on_ring[$i] = $b;                    //
46              } elseif ($arr_f[$i] == 'g'){                      //
47                  $fighters_on_ring[$i] = $g;                    //
48              }                                                  //
49          }                                                      //
50      }                                                          //
51  ////////////////////////////////////////////////////////////////
```



Резюме на кода

```
Master array!Array
(
    [helth] => 14
    [attack] => 6
    [resistance] => 2
)

Fighters array!Array
(
    [0] => a
    [1] => a
    [2] => g
)
```



```
Master array!Array
(
    [helth] => 14
    [attack] => 6
    [resistance] => 2
)

Fighters array!Array
(
    [0] => Array
        (
            [helth] => 5
            [attack] => 6
            [resistance] => 2
        )

    [1] => Array
        (
            [helth] => 5
            [attack] => 6
            [resistance] => 2
        )

    [2] => Array
        (
            [helth] => 8
            [attack] => 6
            [resistance] => 5
        )

)
```

Резюме на кода

6. Преди основния цикъл ще създадем няколко променливи които ще броят елементи нужни ни за крайната статистика.

```
53  ////////////////////////////////////////////////////////////////////
54  $master_hits = 0;                                                 //
55  $fighter_hits = 0;                                                //
56  $dead_fighters = 0;                                               //
57  $mhd = 0;                                                         //
58  $total_md = 0;                                                    //
59  $total_fd = 0;                                                    //
60  ////////////////////////////////////////////////////////////////////
```

$master_hits = Брояч на ударите нанесени от Master
$Fighter_hits = Брояч на ударите нанесени от Fighters
$dead_fighters = Брояч на победените Fighters
$mhd = Средна стойност на поражението нанесено от Master над Fighters /временна/
$fhd = Средна стойност на поражението нанесено от всеки един от Fighters над Master (дефинира се в цикъла, защото при всеки нов Fighter средната стойност може да е различна и се нулира след края на битката м/у Master и всеки от Fighters). /временна/
$total_md = Общият резултат на пораженията нанесени от Master над Fighters за играта
$total_fd = Общият резултат на пораженията нанесени от всички Fighters над Master за играта.

Резюме на кода

Основен цикъл:

```
62  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
63  for ($i = 0; $i < count($fighters_on_ring); $i++){                                                                            //
64                                                                                                                                //
65   $fhd = 0; // Fighters Hits Damage, counter start from 0 for each new fighter                                                //
66                                                                                                                                //
67      while ($master['helth'] > 0 && $fighters_on_ring[$i]['helth'] > 0){                                                       //
68                                                                                                                                //
69          $fighter_strvalike = ($master['helth'] + $master['resistance']) - $fighters_on_ring[$i]['attack'];                    //
70                  $master['helth'] = $fighter_strvalike;                                                                        //
71                  $fighter_hits += 1;                                                                                           //
72                  $fhd = $fighters_on_ring[$i]['attack'] - $master['resistance'];                                               //
73                  $total_fd += $fhd;                                                                                            //
74                                                                                                                                //
75          if ($master['helth'] <= 0){                                                                                           //
76          break;                                                                                                                //
77          } else {                                                                                                              //
78                                                                                                                                //
79          $master_strvalike =($fighters_on_ring[$i]['helth'] + $fighters_on_ring[$i]['resistance']) - $master['attack'];        //
80                  $fighters_on_ring[$i]['helth'] = $master_strvalike;                                                           //
81                  $master_hits +=1;                                                                                             //
82                  $mhd = $master['attack'] - $fighters_on_ring[$i]['resistance'];                                               //
83                  $total_md += $mhd;                                                                                            //
84                  if ($fighters_on_ring[$i]['helth'] <= 0 ){                                                                    //
85                      $dead_fighters++;                                                                                         //
86                  }                                                                                                             //
87          }                                                                                                                     //
88      }                                                                                                                         //
89  }                                                                                                                             //
90  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

В основни линии логиката е следната: Върти цикъла докато стойността в масивите $master **И** $fighters_on_ring със индекс „helth” стана по-малка от 0.
Защо „и“, а не „или“ !? Защото искаме когато Master е с 0 точки, цикъла да спре… не да го ритаме и докато е мъртъв. Аналогично и за Fighters. (

Добре обаче Fighters са повече от Master, т.е. за тях условието ще трябва да се извърта и да рестартира при следващия подред. За целта ще използваме for около while-а, така ще стартираме горния цикъл while толкова пъти колкото участници има от страна на Fighters.
Преди да влезем отново във while-а обаче се пита дали и двете страни (Master и Fighter) все още имат положително число на жизнените си показатели. Ако условието е вярно продължаваме вътре в while-а.

Какво се случва там? По условие знаем, че Fighter удря пръв, следователно първото уравнение гласи:

$Fighter_strike = ($master['helth'] + $master['resistance']) - $fighters_on_ring[$i]['attack'];

Fighter нанася удар с който нанася поражения над здравето на Master. Новия резултат за здравето на Master се презаписва, така при следващия удар ще изваждаме от вече новата стойност.

$master['helth'] = $Fighter_strike;

Вторият важен момент в този цикъл е ударът на Master над Fightera:

$master_strike =($fighters_on_ring[$i]['helth'] + $fighters_on_ring[$i]['resistance']) - $master['attack‘];

$Fighters_on_ring[$i]['helth'] = $master_strike;

До тук добре! Но какво ще стане, ако например, Fighter победи Master още при първия удар? While-а ще продължи към второто уравнение. При вече нулево „здраве“ Master не би трябвало да може да нанесе удар.
За това второто уравнение изразяващо удара на Master ще го поставим в if. Този if ще проверява дали „здравето“ на Master е по-голямо от 0 и ако е ще изпълни уравнението, ако не е ще прекъсне програмата и ще принтира стойностите до момента.

```
Master array!Array
(
    [helth] => -2
    [attack] => 6
    [resistance] => 2
)

Fighters array!Array
(
    [0] => Array
        (
            [helth] => -3
            [attack] => 6
            [resistance] => 2
        )

    [1] => Array
        (
            [helth] => 1
            [attack] => 6
            [resistance] => 2
        )

    [2] => Array
        (
            [helth] => 8
            [attack] => 6
            [resistance] => 5
        )

)
```

До тук успяхме да получим двата нови масива съдържащи данните за уменията на Master и Fighters след битките.
Става ясно следното:

- Победител: fighters;
- Победени Fighters: 1
- Не победени Fighters: 2
- Брой на точките „здраве“ останали на Master: -2 (трябва да принтираме 0)

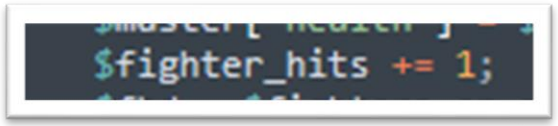
Остава да разберем: броя на ударите нанесени от Master и от Fighters, средното поражение нанесено от Fighters и това нанесено от Master.

Тук ще се обърнем към променливите които дефинирахме по-горе.

Брояч на нанесените удари в играта от страна на Master и Fighters

$master_hits = Брояч на ударите нанесени от Master. Дефиниран е извън циклите защото искаме стойността в него да се натрупва през цялата игра. Намира се в while под уравнението отговарящо за ударите на Master, като представлява прост брояч добавящ 1 всеки път когато преминем през уравнението. Не забравяйте, че преди уравнението имаме if който няма да позволи на Master да удари ако няма положителен здравен резултат. Т.е. и брояча ще спре да бори ударите.

Аналогична е ситуацията с **$Fighter_hits** = Брояч на ударите нанесени от Fighters.



Средна стойност на поражението нанесено от Master над Fighters

- **$mhd** = Средна стойност на поражението нанесено от Master над Fighters /временна/

За да отчетем колко разрушителен е бил един удар на Master за Fighter, то на тази променлива присвояваме уравнението Master [‘атака’] - Fighter [‘защита’] = Чистата числова стойност на пораженията които нанася Master със своя удар.

Този резултат обаче трябва да бъде събран за всяка една битка.

Тук на помощ идва новата променлива, дефинирана извън цикъла, за да се нулира след края на всеки цикъл от битки, а не след само една битка с един Fighter от масива.

- **$total_md** = Общият резултат на пораженията нанесени от Master над Fighters за играта

Този резултат ще ни послужи по-късно когато смятаме средната разрушителност на удара на Master за битката.

$avg_mhd = $total_md / $master_hits;

Дефинираме $avg_mhd след цикъла и в нея изчисляваме средната разрушителност на удъра на Master за тази битка, като просто делим натрупаното в променливата $total_md на натрупания брой удари в променливата $master_hits.

Средна стойност на поражението нанесено от всеки един от Fighters над Master

- **$fhd** = Средна стойност на поражението нанесено от всеки един от Fighters над Master (дефинира се в цикъла FOR, защото при всеки нов Fighter средната стойност може да е различна и се нулира след края на битката м/у Master и всеки от Fighters). /временна/
- **$total_fd** = Общият резултат на пораженията нанесени от всички Fighters над Master за играта.
- Отново ще събира в себе си резултатите от средните стойности на разрушителност на удар по аналогична формула на тази от горния пример.
- **$avg_fhd** = $total_fd / $fighter_hits; - отново изчисляваме средната стойност за разрушителност на нанесените удари в играта, като делим на броя нанесени удари.

Брояч на победените Fighters.

- **$dead_fighters** = Брояч на победените Fighters. Дефиниран е извън циклите защото искаме стойността в него да се натрупва през цялата игра. Намира се в while след уравнението отговарящо за ударите на Master, като представлява прост брояч добавящ 1 всеки път когато преминем през уравнението.
- Не забравяйте, че преди уравнението имаме if който няма да позволи на Master да удари ако няма положителен здравен резултат.
- Т.е. и брояча ще спре да бори ударите.
  В допълнение този брояч е поставен в if като се пита дали стойността на елемент с индекс „здраве“ в масив Fighter за съответното $i е по-малък или равен на 0, ако това е вярно, брояча добавя 1 към общия си резултат, ако не е вярно продължава по кода.

```
if ($fighters_on_ring[$i]['health'] <= 0 ){
    $dead_fighters++;
}
```

Събиране на крайната статистика.

```
//////////////////////////////////////////////////////////////////
$winer = '';                                                    //
if ($master['health'] > 0 ) {                                   //
    $winer = 'Master';                                          //
    } else {                                                    //
        $winer = 'Fighters';                                    //
    }                                                           //
//////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////
$avg_fhd = $total_fd / $fighter_hits;                           //
$avg_mhd = $total_md / $master_hits;                            //
$alive_fighters = count($fighters_on_ring) - $dead_fighters;     //
//////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////
        $master_health = 0;                                     //
        if ($master['health'] < 0) {                            //
            $master_health = 0;                                 //
        } else {                                                //
            $master_health = $master['health'];                 //
        }                                                       //
//////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////
echo '["'.($winer).'",'."<br>",                                 //
'"'.($dead_fighters).'",' . "<br>",                             //
'"'.($alive_fighters).'",' . "<br>",                            //
'"'.($master_health).'",'. "<br>",                              //
'"'.($fighter_hits).'",' . "<br>",                              //
'"'.($master_hits).'",' . "<br>",                               //
'"'.(round($avg_fhd, 2)).'",'. "<br>",                          //
'"'.(round($avg_mhd,2)).'"]';                                   //
//////////////////////////////////////////////////////////////////
```

Намираме победителя от двубоя. С if питаме дали здравето на Master е по-голямо от 0 и ако е, на променливата Winer присвояваме „Master”, ако не е – “Fighters”.

Бяха подробно описани в предходните слайдове.

Понеже резултатът при „Здраве“ на Master след двубоя може да е число по-малко от нула с този if казваме, че ако е под 0, то принтирай 0, ако не е под 0, принтирай каквото е.

Принтираме резултатите по посочения в заданието формат.