7 Praktijkopdracht - Simulatie

Praktijkopdrachten worden individueel uitgevoerd. Je moet dus je eigen code maken.
Praktijkopdrachten worden apart beoordeeld (los van het huiswerk).

Inleiding

Stel je wordt gevraagd of je mee wilt doen met een spel. Het voorstel is dat je voor 1 euro met drie dobbelstenen mag gooien. Als je 18 gooit, dat zijn dus drie zessen, dan win je 100 euro. Is het slim om dit spel te spelen? Denk je dat je winst maakt?

In deze praktijkopdracht leer je om uit te vinden wat de kans is om 18 te gooien. Als je weet hoelang het gemiddeld duurt voordat je 18 gooit met drie dobbelstenen dan weet je of dit een lucratief (=winstgevend) spel is.

In deze les komt alles aan de orde wat we tot nu hebben geleerd. Er komen ook functies aan de orde. Deze heb je bij de andere vakken (Java Script, Java) al gehad, maar er zal ook nog kort worden uitgelegd hoe een functie in PHP werken.

Praktijkopdracht - Simulatie

In opdracht 2b van een vorige les (PHP-1, 6.2) gooi je net zo lang met 3 dobbelstenen totdat je drie maal een 6 heb gegooid. Hoe groot is nu de kans om 3x een 6 te gooien? Vanuit de wiskunde kun je dit berekenen, maar als programmeur kun je dit ook bepalen met een simulatie.

We gaan nu via een simulatie bepalen wat de kans is om 3 x 6 te gooien. Omdat te doen gaan we 1000 keer net zolang gooien totdat we 3 x 6 hebben gegooid. We hebben dan 1000 uitkomsten. Elke uitkomst geeft aan hoe lang het duurde om 3 x 6 te gooien.

Om dit probleem aan te pakken delen we het op in kleine stapjes.

Functies

We gaan gebruik maken van functies.

Functies werken hetzelfde als in andere talen en in de JavaScript lessen hebben we het hier over gehad. Functies hebben we nog niet gehad in PHP, maar in het volgende filmpje wordt uitgelegd hoe wat een functie ook alweer is en hoe je die in PHP moet gebruiken: Uitleg over functions.

Stap 1 - maak functie throwDice($numberOfDice)

Maak een functie die je aanroept met `throwDice($numberOfDice)` De functie krijgt het aantal dobbelstenen mee dat moet worden gegooid en de functie returned het totaal aantal ogen dat je hebt gegooid.

Stap 2 - voer de functie net zolang uit totdat de uitkomst 18 is

Drie maal zes geeft een uitkomst van 18. Voer de functie dus net zolang uit totdat de uitkomst 18 is. Tel hoeveel keer je de functie hebt aangeroepen.

Maak ook hiervan een functie. Noem de functie `waitForResult($result)`. De functie krijgt nu 18 mee als parameter en returned hoe vaak het heeft moeten 'gooien' om tot het resultaat te komen. Vanuit deze functie `waitForResult` roepen we dus de functie `throwDice` aan.

Dus bijvoorbeeld:

Je roept waitForResult(18) aan en je krijgt 101 terug. Dat betekent dat de computer 101 x heeft gegooid en dat er toen pas 18 was gegooid.

Of je roeptwaitForResult(7) en je krijgt 12 terug. Dat betekent dat de computer 12 keer heeft gegooid en dat er toen 7 was gegooid.

Stap 3 - voer de simulatie tig-keer uit.

In stap 2 voeren we de simulatie één keer uit. We wachten één keer op een bepaalde uitkomst bij het werpen van de dobbelstenen.

De uitkomst hangt af van geluk. Het kan zijn dat we gelijk in één keer 18 gooien, maar het kan ook meer dan 100 worden duren. Als we nu heel vaak gooien en het gemiddelde berekenen dan filteren we de geluksfactor er een beetje uit. We gaan in deze stap dan ook 1000x de functie waitForResult(18) uitvoeren en bepalen het gemiddelde van de return waarde van deze functie.

De functie geeft terug hoeveel worpen het kostte om 18 te gooien. Als we dat 1000 keer doen en we bepalen het gemiddelde van al deze aantal worpen dan weten we hoelang je gemiddeld moet wachten op een uitkomst van 18 als je met drie dobbelstenen gooit.

Stap 4 - nog een stapje verder

En wat is het antwoord? Met je huidige simulatieprogramma kun je ook heel eenvoudig bepalen hoe lang het duurt om bijvoorbeeld 11 te gooien of 10, of 9,....

Kun je van alle mogelijke uitkomsten 3 tot en met 18 nu in een loop bepalen hoe lang het gemiddeld duurt om dit aantal ogen met drie dobbelstenen te gooien?

De output ziet er als volgt uit:

```
To throw a total of 3 with 3 dice, it took on average ... throws.
To throw a total of 4 with 3 dice, it took on average ... throws.
To throw a total of 5 with 3 dice, it took on average ... throws.
To throw a total of 6 with 3 dice, it took on average ... throws.
To throw a total of 7 with 3 dice, it took on average ... throws.
To throw a total of 8 with 3 dice, it took on average ... throws.
To throw a total of 9 with 3 dice, it took on average ... throws.
To throw a total of 10 with 3 dice, it took on average ... throws.
To throw a total of 11 with 3 dice, it took on average ... throws.
To throw a total of 12 with 3 dice, it took on average ... throws.
To throw a total of 13 with 3 dice, it took on average ... throws.
To throw a total of 14 with 3 dice, it took on average ... throws.
To throw a total of 15 with 3 dice, it took on average ... throws.
To throw a total of 16 with 3 dice, it took on average ... throws.
To throw a total of 17 with 3 dice, it took on average ... throws.
To throw a total of 18 with 3 dice, it took on average ... throws.
```

Voor de duidelijkheid de vertaling in het Nederlands luidt:

> Om in totaal 18 te gooien met 3 dobbelstenen, duurde het gemiddeld ... worpen.

--

Revision #13
Created 2020-03-01 18:37:08 UTC by Max
Updated 2020-05-04 20:47:25 UTC by Max