

Trabajo N° 1

Lunes 28 de Marzo Grupos 1 al 6

Miércoles 30 de Marzo Grupos 7 al 12

Presentación de lo que consideren más interesante en **máximo 10 minutos**

Objetivo

El objetivo es desarrollar una solución funcional en JavaScript a un problema de cierta complejidad. Se espera que utilicen el máximo de elementos funcionales posibles. Por ejemplo:

- Combinators
- Iterators y Generators
- La librería Lodash
- Currying y partial evaluation
- Composición y Pipes
- Chaining

Trabajo a Desarrollar

El trabajo es grupal y se espera que cada grupo

- Presente su trabajo en clases 10 minutos por grupo.
- Deje todo su material: Láminas, y el Código compartirlo en GitHub.
- Detalles de la entrega verlos con Felipe (ayudante del curso) frvalenzuela@uc.cl

Evaluación

La evaluación considera

- calidad del ejemplo desarrollado (belleza del código, uso de técnicas) 60%
- efectividad de la presentación (explicaciones claras, didáctica) 40%

El Problema

Se trata de implementar una simulación de un juego de dardos

En un juego de dardos, los jugadores toman turnos para lanzar 3 dardos a un blanco. Los puntos obtenidos en cada lanzamiento se cuentan de acuerdo al siguiente esquema (ver figura):

- el puntaje base obtenido por cada dardo depende del lugar en que quede (así puede caer en el 20, en el 9, etc) pero ...
- ... hay un multiplicador de ese puntaje base que depende del punto exacto del sector correspondiente al número (x1, x2 y x3). Así un dardo que cayó en la zona del 20 podría tener un x3 y dar 60 puntos
- hay 2 casos especiales que corresponden a aciertos en el centro mas pequeño (Double Bull o DB) que entrega 50 puntos o en un círculo mas grande (Single Bull o SB) que da 25 puntos
- si el dardo no cae en una zona punteada corresponde a un Null y gana 0 puntos



Los jugadores (a lo menos 2) parten con 501 puntos al inicio de la competencia. A este número se le va restando lo obtenido en los lanzamientos hasta llegar a cero. Es importante llegar a cero en forma exacta puesto que si se llega a un valor negativo, es número en valor absoluto passa a ser el puntaje del jugador (por ejemplo el jugador tenía 20 puntos y obtiene 30 en sus lanzamientos, entonces $20 - 30 = -10$ y quedaría con 10 puntos)

Como se sugiere implementar

1. Una función que reciba el nombre del jugador, su puntaje y sus lanzamientos y devuelva el nuevo score del jugador

Ejemplo:

```
ingresar_jugada('Jaime', 501, ['DB', [3,20], [3,19]])
```

334

```
ingresar_jugada('Ema', 501, ['SB', [2,20], [3,20]])
```

376

Jaime obtuvo un Double Bull (50 pts), un 20 con x3 (60 pts) y un 19 con x3 (57 pts) con un total de 167 puntos que restados a 501 deja su score en 334.

Ema obtuvo un Single Bull (25 pts), un 20 x2 (40 pts) y un 20 x3 (60 pts) con un total de 125 puntos que restados a 501 deja el score en 376

2. Una función que inicializa el juego dejando a cada jugador con 501 puntos

Ejemplo:

```
init_game('Jaime', 'Ema')
```

3. Una función que permita manejar el diálogo con el simulador

Ejemplo:

```
play_game('Jaime', 'Ema')
```

Juego inicializado con jugadores Jaime y Ema. Ingrese lanzamientos de Jaime

```
['DB', [3,20], [3,19]]
```

Jaime queda con 334 puntos. Ingrese lanzamientos de Ema

```
['SB', [2,20], [3,20]]
```

Ema queda con 376 puntos. Ingrese lanzamientos de Jaime

...

```
[[1, 18], [1, 10], [1, 12]]
```

Ema queda con 0 puntos y gana el juego. Felicitaciones Ema !!

Pueden descomponer el problema de otra forma, pueden usar otro formato para la entrada de puntajes, etc. Lo importante es que el programa funcione como se indica, pero en este caso es mucho más importante el **mantenerse dentro del paradigma funcional**.