

PROGRAMACIÓN REACTIVA CON RXJS

13/04/2020

Grupo 2

Paul Heinsohn

Sebastián Carreño

Profesor Jaime Navón

BEER CHALLENGE

- 2 a 4 jugadores



BEER CHALLENGE

- 2 a 4 jugadores
- Acabar la cerveza (100 puntos)



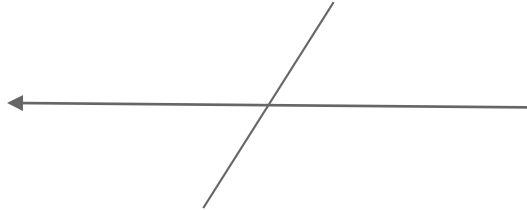
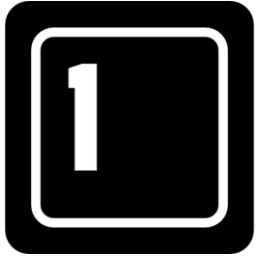
BEER CHALLENGE

- 2 a 4 jugadores
- Acabar la cerveza (100 puntos)
- 2 teclas por jugador
 - Tomar (puntaje actual + 1)



BEER CHALLENGE

- 2 a 4 jugadores
- Acabar la cerveza (100 puntos)
- 2 teclas por jugador
 - Tomar (puntaje actual + 1)
 - Contar un chiste (penalizaciones: puntaje actual - 1)



BEER CHALLENGE DEMO

REACTIVIDAD CON RXJS

- Definir N° de jugadores
- Enviar N° de jugadores
- Enviar teclas de usuarios

```
// Query Selectors
const numBtn = document.querySelector('#decrease, #increase');
const sendPlayersBtn = document.querySelector('#sendPlayers');
const sendKeysBtn = document.querySelector('#sendKeys');

// Observables
const numBtnObs = Rx.Observable.fromEvent(numBtn, 'click');
const sendPlayersObs = Rx.Observable.fromEvent(sendPlayersBtn, 'click');
const sendKeysObs = Rx.Observable.fromEvent(sendKeysBtn, 'click');

// Observadores
const numObserver = numBtnObs.subscribe(event => {
  // Busca el span y lo actualiza acorde al número de jugadores
});
const sendPlayersObserver = sendPlayersObs.subscribe(() => {
  // Crea los elementos del tipo input para que los usuarios ingresen sus teclas
});
const sendKeysObserver = sendKeysObs.subscribe(() => {
  // Valida los inputs evitando repetidos y nulos
});
```

N° of Players (2 - 4): 2

Choose Keyboard Inputs

Player 1:

Player 2:

REACTIVIDAD CON RxJS

- Temporizador de cuenta regresiva

```
const startTimer = Rx.Observable.create(observer => {  
  let value = 2;  
  setInterval(() => {  
    if (value === 0) observer.complete();  
    else observer.next(value);  
    value--;  
  }, 1000);  
});  
const startTimerObserver = {  
  next: value => statusH.innerHTML = `${value}`,  
  
  error: err => console.error(err),  
  
  complete: () => {  
    statusH.innerHTML = "Chug (key N°1)!";  
    const kObs = keysObs.subscribe(keysObserver);  
    ...  
  }  
};
```



3 2 1



Chug (key N°1)!

REACTIVIDAD CON RXJS

- Manejo de input de teclas

```
const keysObs = Rx.Observable.fromEvent(document, "keyup").map(keyHandler);
```

```
const keysObserver = {  
  next: scores => {  
    playerScores.innerHTML = "";  
    Object.values(scores).forEach((score, player) => {  
      playerScores.innerHTML += `P${player + 1}: ${score} `;  
    });  
    scoresBehaviouralSubject.next(scores);  
  },  
  error: err => console.log(err),  
  complete: () => console.log(`Got a winner`)  
};
```



P1: 7 P2: -2

```
const keyHandler = (event) => {  
  const key = event.key;  
  Object.values(playersKeys).forEach((playerKeys, i) => {  
    // Si el jugador i presiona la tecla 1 y no se está riendo --> scores[i]++  
    // Si el jugador i presiona la tecla 1 y se está riendo --> scores[i]--  
    // Si el jugador i presiona la tecla 2 y no se puede contar un chiste --> scores[i]--  
    // Si el jugador i presiona la tecla 2 y se puede contar un chiste --> bloquearParticipantes(i)  
  });  
  return scores;  
};
```

BEHAVIOUR SUBJECT

- Para supervisar requisito de finalización del juego

```
const scoresBehaviouralSubject = new Rx.BehaviorSubject(scores);
```

```
scoresBehaviouralSubject
  .takeWhile(scores => Object.values(scores).every(x => x < WINNERKEYUPS))
  .finally(() => {kObs.unsubscribe(),
    | | | | | | | printWinner(scores),
    | | | | | | | timers[timers.length - 1].unsubscribe()})
  .subscribe();
```

```
scoresBehaviouralSubject.next(scores);
```

TIMERS CONSECUTIVOS PARA CHISTES

- Primer timer activa secciones de ‘contar chistes’

Chug (key N°1)!

TIMERS CONSECUTIVOS PARA CHISTES

- Primer timer activa secciones de ‘contar chistes’
- **Jugadores pueden presionar su segunda tecla**

Tell a joke (key N°2)!

TIMERS CONSECUTIVOS PARA CHISTES

- Primer timer activa secciones de ‘contar chistes’
- Jugadores pueden presionar su segunda tecla
- **Solo el jugador que presiona primero la tecla, puede beber**

Player 1 tells a joke!

Everyone laughs

TIMERS CONSECUTIVOS PARA CHISTES

- Primer timer activa secciones de ‘contar chistes’
- Jugadores pueden presionar su segunda tecla
- Solo el jugador que presiona primero la tecla, puede beber
- **Segundo timer desactiva las penalizaciones**

Chug (key N°1)!