
Código reactivo en JavaScript

Grupo 12

Integrantes:

Maximiliano Friedl Schwarzenberg

Tomás Mellado Medina

Noemí Crosby Conget

Tarea 2 IIC3585 - Spaceship Game

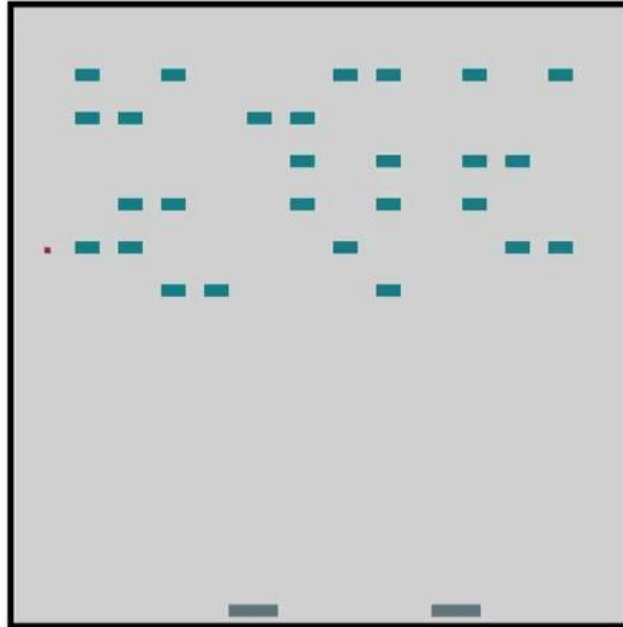
by: Noemi Crosby, Maximiliano Friedl & Tomas Mellado

Demo

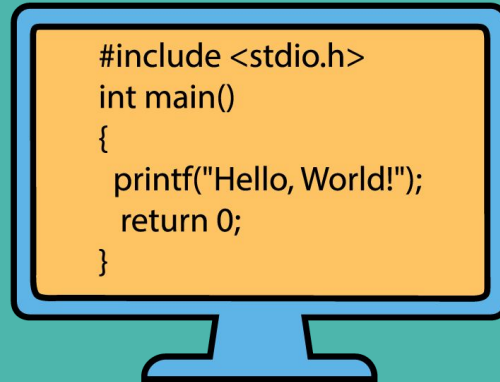
Start game

Reset

Score: 6



Lo más interesante de nuestro código



Recibir Comandos

```
var keyDowns = Rx.Observable.fromEvent(document, 'keydown');
keyDowns.subscribe(function (e) {
  ...
  if (e.key === 'ArrowLeft' && playerInBoard(2, 'l')) {
    state.board.updatePlayer(2, 'l');
    state.player2.x -= 1
  } else if (e.key === 'ArrowRight' && playerInBoard(2, 'r'))
  {
    state.board.updatePlayer(2, 'r');
    state.player2.x += 1;
  }
});
```

Recibir Comandos

```
var keyDowns = Rx.Observable.fromEvent(document, 'keydown');
```

V/S

```
var keyUp = Rx.Observable.fromEvent(document, 'keyup');
```

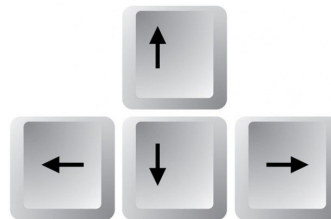
Recibir Comandos

```
var keyDowns = Rx.Observable.fromEvent(document, 'keydown');  
keyDowns.subscribe(function (e) { ... })
```



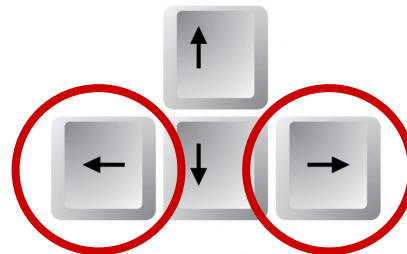
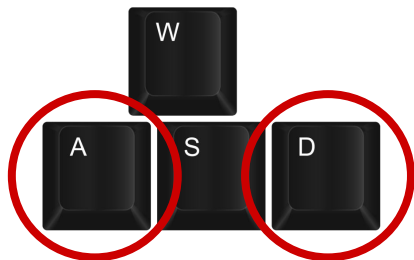
Recibir Comandos

```
var keyDowns = Rx.Observable.fromEvent(document, 'keydown');  
keyDowns.subscribe(function (e) { ... })
```



Recibir Comandos

```
var keyDowns = Rx.Observable.fromEvent(document, 'keydown');  
keyDowns.subscribe(function (e) { ... })
```



Interval

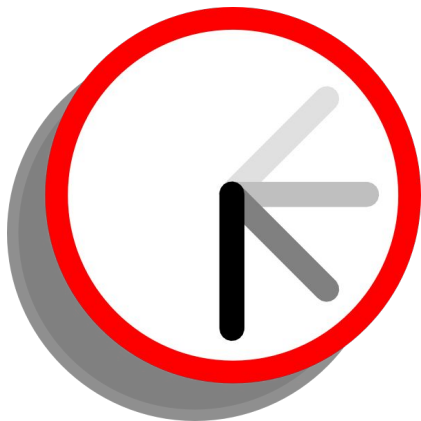
```
document.getElementById("start-button").onclick = function () {  
  const loop = Rx.Observable.interval(50);  
  loop.subscribe(() => {  
    if (state.blocks.length === 0 || state.ball.y + 1 === boardSize -  
1) {  
      document.getElementById('end').innerHTML = 'GAME OVER';  
      loop.unsubscribe();  
    } else {  
      moveBall();  
      state.board.updateBall(state.ball.x, state.ball.y);  
    }  
  });  
};
```

Interval

```
const loop = Rx.Observable.interval(50);  
loop.subscribe(() => { ... });
```

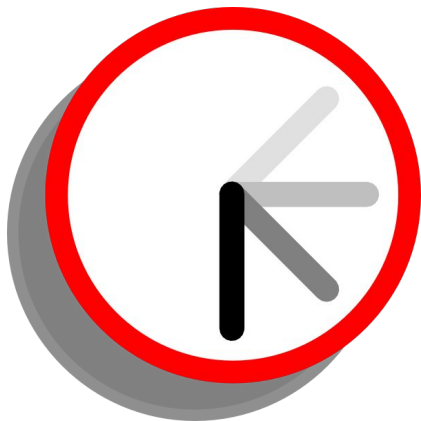
Interval

```
const loop = Rx.Observable.interval(50);  
loop.subscribe(() => { ... });
```



Interval

```
const loop = Rx.Observable.interval(50);  
loop.subscribe(() => { ... });
```



Interval

```
const loop = Rx.Observable.interval(50);
loop.subscribe(() => {
  if (state.blocks.length === 0 || state.ball.y + 1 === boardSize - 1) {
    document.getElementById('end').innerHTML = 'GAME OVER';
    loop.unsubscribe();
  }
});
```

Interval

```
const loop = Rx.Observable.interval(50);
loop.subscribe(() => {
  if (state.blocks.length === 0 || state.ball.y + 1 === boardSize - 1) {
    document.getElementById('end').innerHTML = 'GAME OVER';
    loop.unsubscribe();
  }
});
```

Interfaz

```
export default class Board {  
  
  constructor() {...}  
  
  getElement(x, y) {...}  
  
  paintDot(x, y, color) {...}  
  
  initBoard() {...}  
  
  updatePlayer (player, direction) {...}  
  
  updateBall (x, y) {...}  
  
  drawBlock (x, y) {...}  
  
  deleteBlock (x, y) {...}  
  
}
```

Interfaz

```
export default class Board {  
  constructor() {...}  
  getElement(x, y) {...}  
  paintDot(x, y, color) {...}  
  initBoard() {...}  
  updatePlayer (player, direction) {...}  
  updateBall (x, y) {...}  
  drawBlock (x, y) {...}  
  deleteBlock (x, y) {...}  
}
```



Interfaz

Game
logic

```
export default class Board {  
  
  constructor() {...}  
  
  getElement(x, y) {...}  
  
  paintDot(x, y, color) {...}  
  
  initBoard() {...}  
  
  updatePlayer (player, direction) {...}  
  
  updateBall (x, y) {...}  
  
  drawBlock (x, y) {...}  
  
  deleteBlock (x, y) {...}  
  
}
```

The HTML logo is a rounded square with a vertical gradient from yellow at the top to red at the bottom. The letters "HTML" are written in white, bold, sans-serif font in the center.

HTML

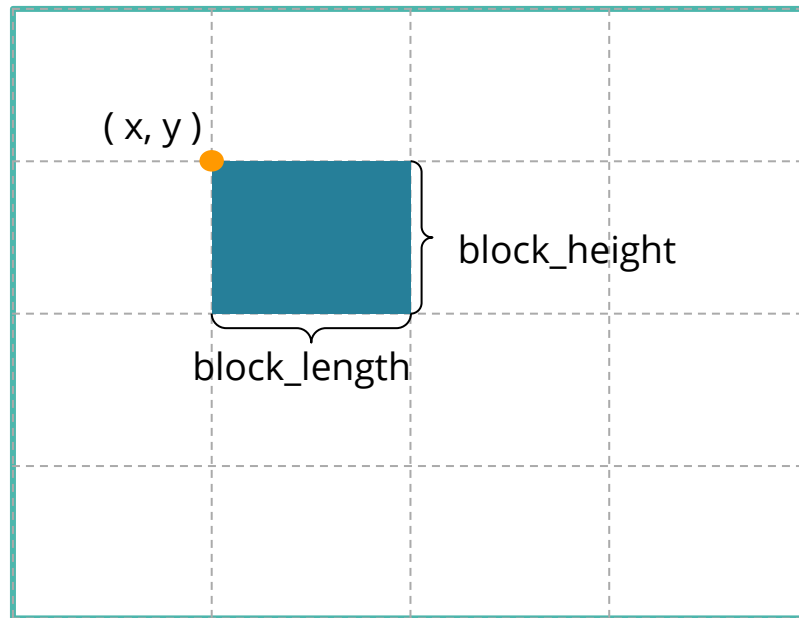
Posicionando bloques

```
const makePoints = ( n, minX, maxX, minY, maxY ) => {  
  while ( state.blocks.length < n ) {  
    var p = {  
      x: random(minX, maxX, block_length),  
      y: random(minY, maxY, block_height)  
    };  
    if (!inList(p)) state.blocks.push(p);  
  }  
}
```

Posicionando bloques



Posicionando bloques



Posicionando bloques

```
const random = (first, last, step) => {  
  var r = Math.floor(Math.random()*(last - first)/step);  
  return r * step + first;  
}
```

```
const inList = ( p ) => {  
  for ( var i = 0; i < state.blocks.length; i++ ) {  
    if ( state.blocks[i].x == p.x && state.blocks[i].y == p.y ) return true;  
  }  
  return false;  
}
```

Posicionando bloques

```
const random = (first, last, step) => {  
  var r = Math.floor(Math.random()*(last - first)/step);  
  return r * step + first;  
}
```

Posicionando bloques

```
const random = (first, last, step) => {  
  var r = Math.floor(Math.random()*(last - first)/step);  
  return r * step + first;  
}
```

$(last - first)$ = el espacio a ocupar

Posicionando bloques

```
const random = (first, last, step) => {  
  var r = Math.floor(Math.random()*(last - first)/step);  
  return r * step + first;  
}
```

$(last - first)$ = el espacio a ocupar

$(last - first)/step$ = cantidad de puntos que caben en el espacio

Posicionando bloques

```
const random = (first, last, step) => {  
  var r = Math.floor(Math.random()*(last - first)/step);  
  return r * step + first;  
}
```

$(last - first)$ = el espacio a ocupar

$(last - first)/step$ = cantidad de puntos que caben en el espacio

$Math.random()*(last - first)/step$ = elegir uno de estos puntos al azar

Posicionando bloques

```
const random = (first, last, step) => {  
  var r = Math.floor(Math.random()*(last - first)/step);  
  return r * step + first;  
}
```

$(last - first)$ = el espacio a ocupar

$(last - first)/step$ = cantidad de puntos que caben en el espacio

$Math.random()*(last - first)/step$ = elegir uno de estos puntos al azar

$Math.floor(Math.random()*(last - first)/step)$ = asegurarse de que es un entero

Posicionando bloques

```
const random = (first, last, step) => {  
  var r = Math.floor(Math.random()*(last - first)/step);  
  return r * step + first;  
}
```

$(last - first)$ = el espacio a ocupar

$(last - first)/step$ = cantidad de puntos que caben en el espacio

$Math.random()*(last - first)/step$ = elegir uno de estos puntos al azar

$Math.floor(Math.random()*(last - first)/step)$ = asegurarse de que es un entero

$r * step + first$ = obtener el valor del punto elegido

Muchas gracias por su atención
