

Reactive programming

Grupo 2

<https://reactive.daleal.dev>

Manejo de eventos



```
const keyDowns = fromEvent(document, 'keydown');
```



```
const keyDowns = fromEvent(document, 'keydown');

const keyPresses = keyDowns.pipe(
  filter(
    (event) => allowedKeys.includes(event.key),
  ),
  map((event) => event.key),
  distinctUntilChanged((x, y) => x === y),
);
```

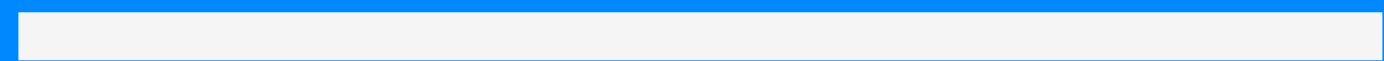


```
const keyDowns = fromEvent(document, 'keydown');

const keyPresses = keyDowns.pipe(
  filter(
    (event) => allowedKeys.includes(event.key),
  ),
  map((event) => event.key),
  distinctUntilChanged((x, y) => x === y),
);

keyPresses.subscribe(
  (key) => {
    pacman.changeDirectionIntent(keyToDirection(key));
  },
);
```

Movimiento





```
interval(REFRESH_RATE).subscribe(() => this.move());
```


Colisiones con los fantasmas



// Ghost

```
movementObservable = new Observable((subscriber) => {  
    interval(REFRESH_RATE).subscribe(() => {  
        subscriber.next({ positionX: this.positionX, positionY: this.positionY });  
    });  
});
```



// PacMan

```
listenForCollisions(...observables) {  
  this.#collisionObservables = observables;  
}  
  
start() {  
  merge(...this.#collisionObservables).subscribe(this.handleGhostCollision.bind(this));  
}
```



```
// PacMan
```

```
notifier = new Subject();
```

```
handleGhostCollision({ positionX, positionY }) {  
  const bordersX = [this.positionX, this.positionX + TILE_SIZE];  
  const bordersY = [this.positionY, this.positionY + TILE_SIZE];  
  
  const isBetweenX = (  
    isBetween(positionX, ...bordersX)  
    || isBetween(positionX + TILE_SIZE, ...bordersX)  
  );  
  const isBetweenY = (  
    isBetween(positionY, ...bordersY)  
    || isBetween(positionY + TILE_SIZE, ...bordersY)  
  );  
  
  if (isBetweenX && isBetweenY) {  
    this.notifier.next({ status: 'game_over' });  
  }  
}
```

Muchas gracias

Grupo 2