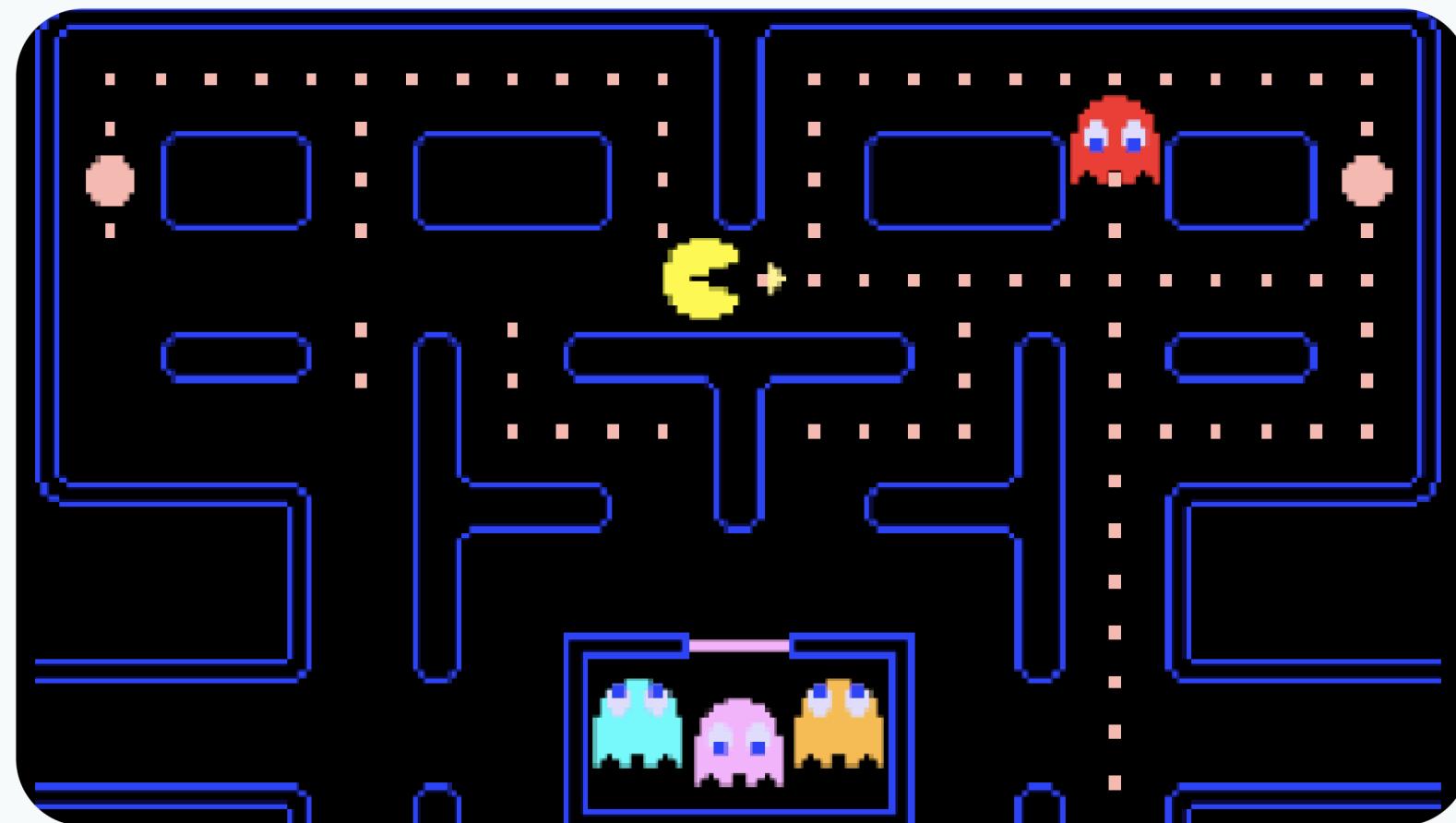


# Reactive JS

Grupo 4



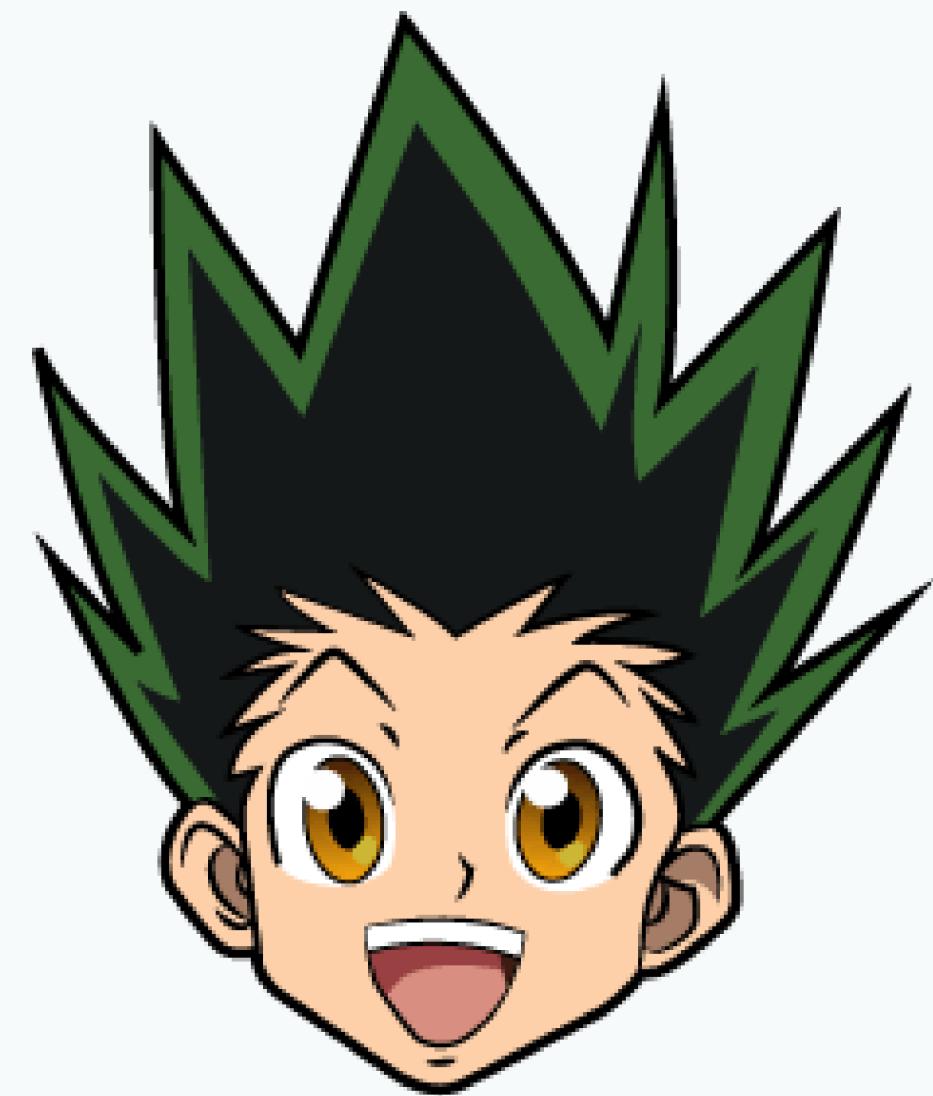
# El Juego



+



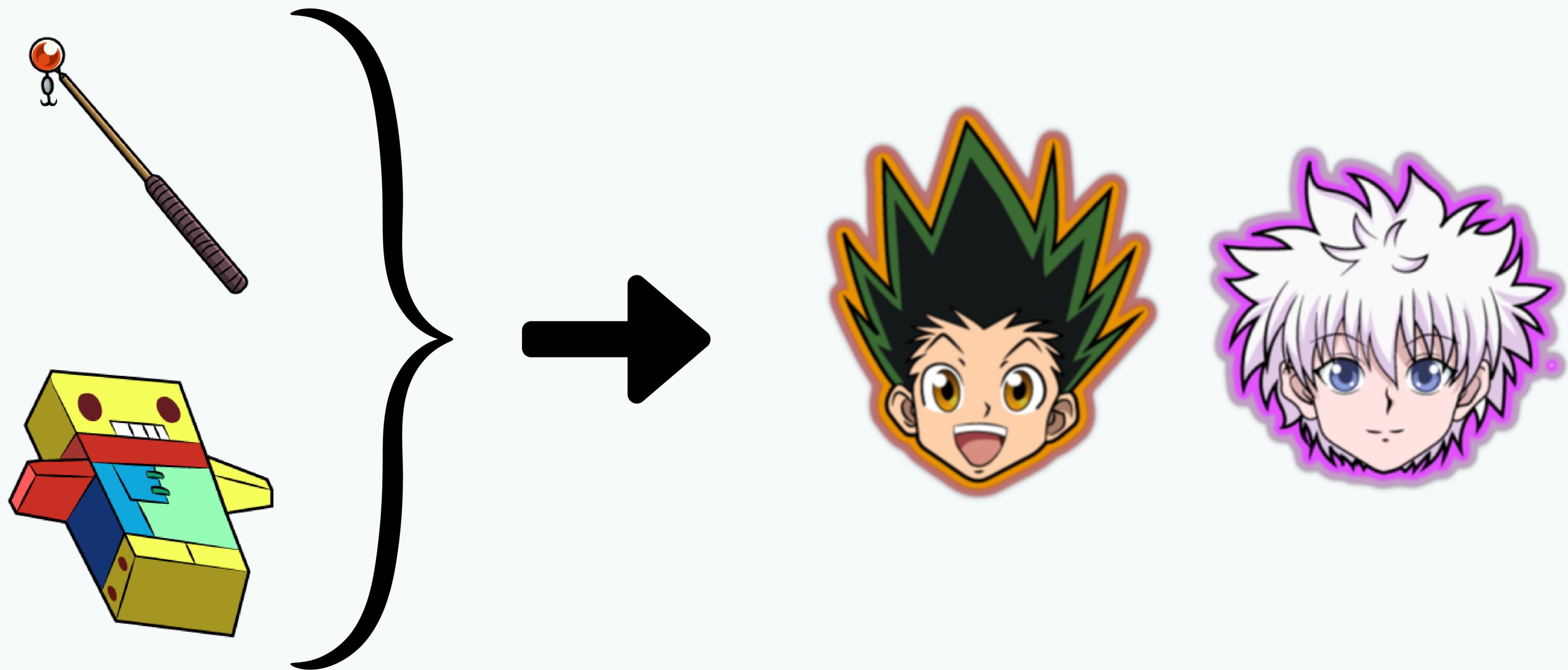
**P1**



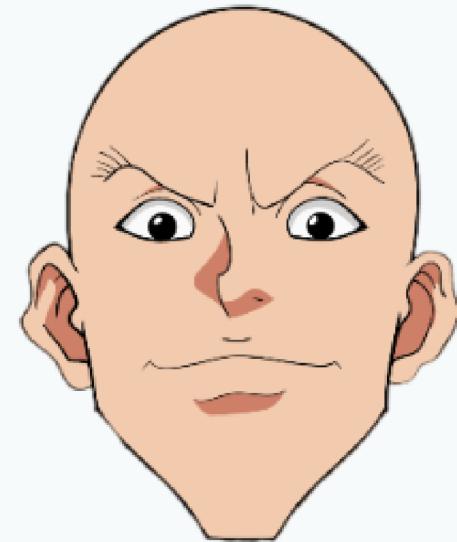
**P2**



# Rewards

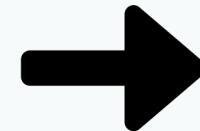


# Enemies



# El Juego

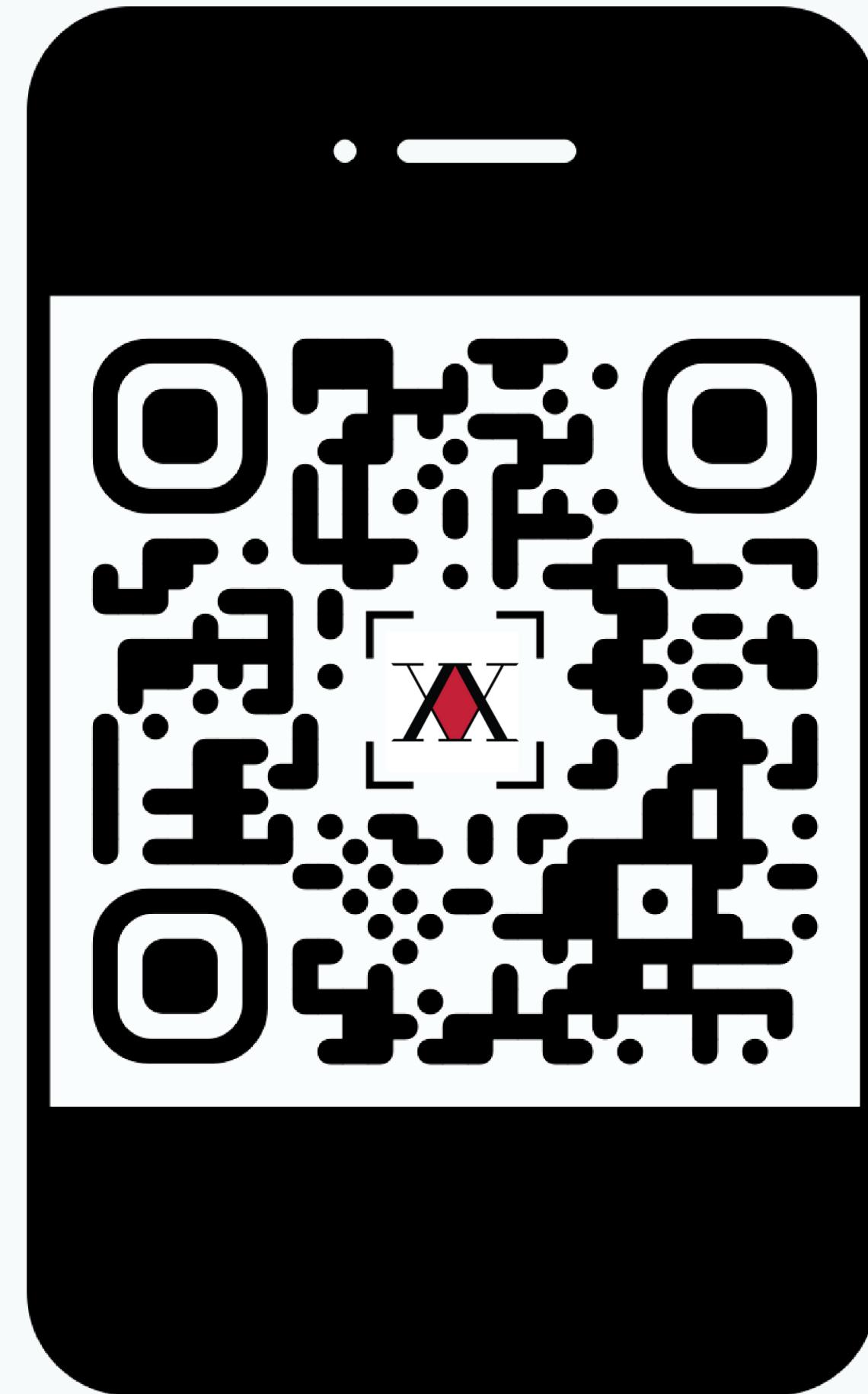
**Se gana** si todos  
los enemigos  
mueren



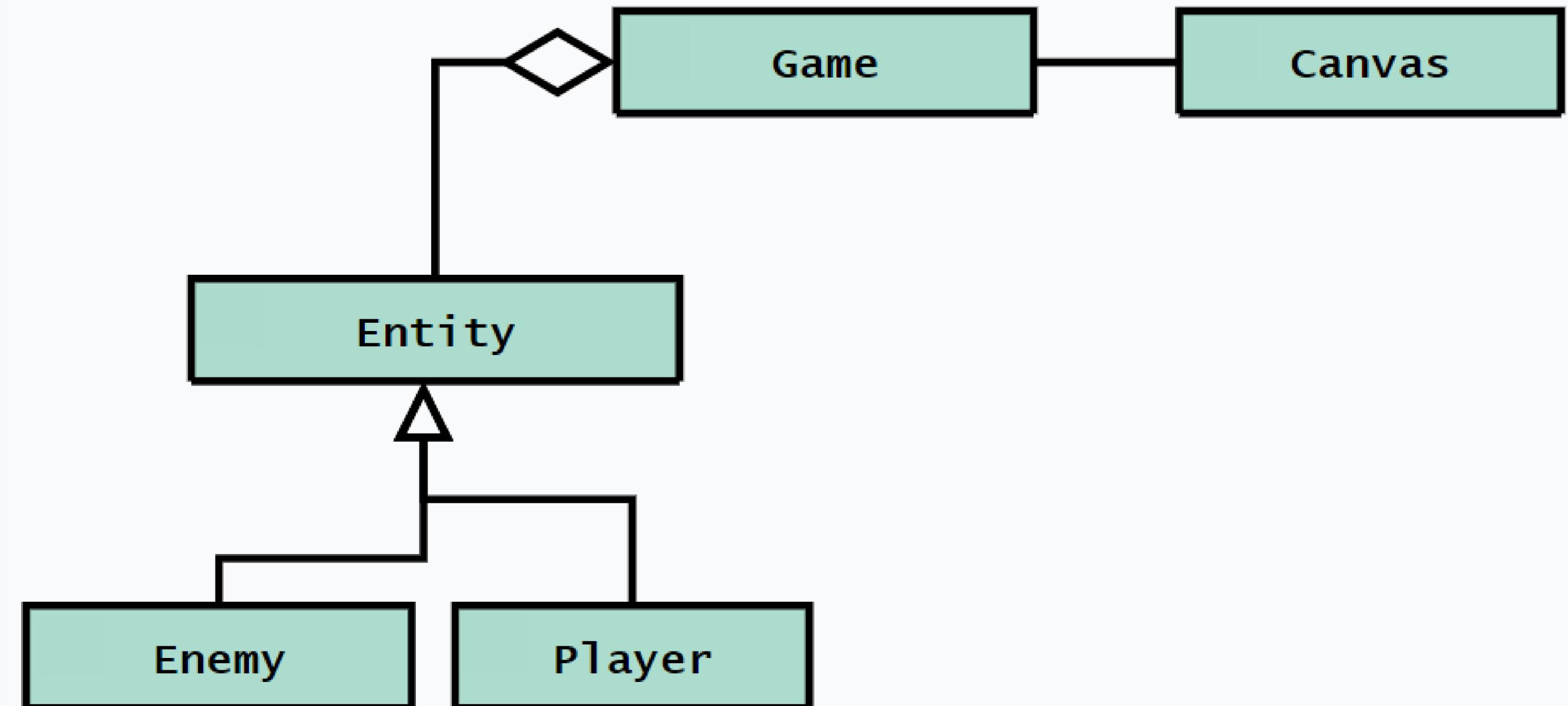
**Se pierde** si  
todas las vidas  
se pierden



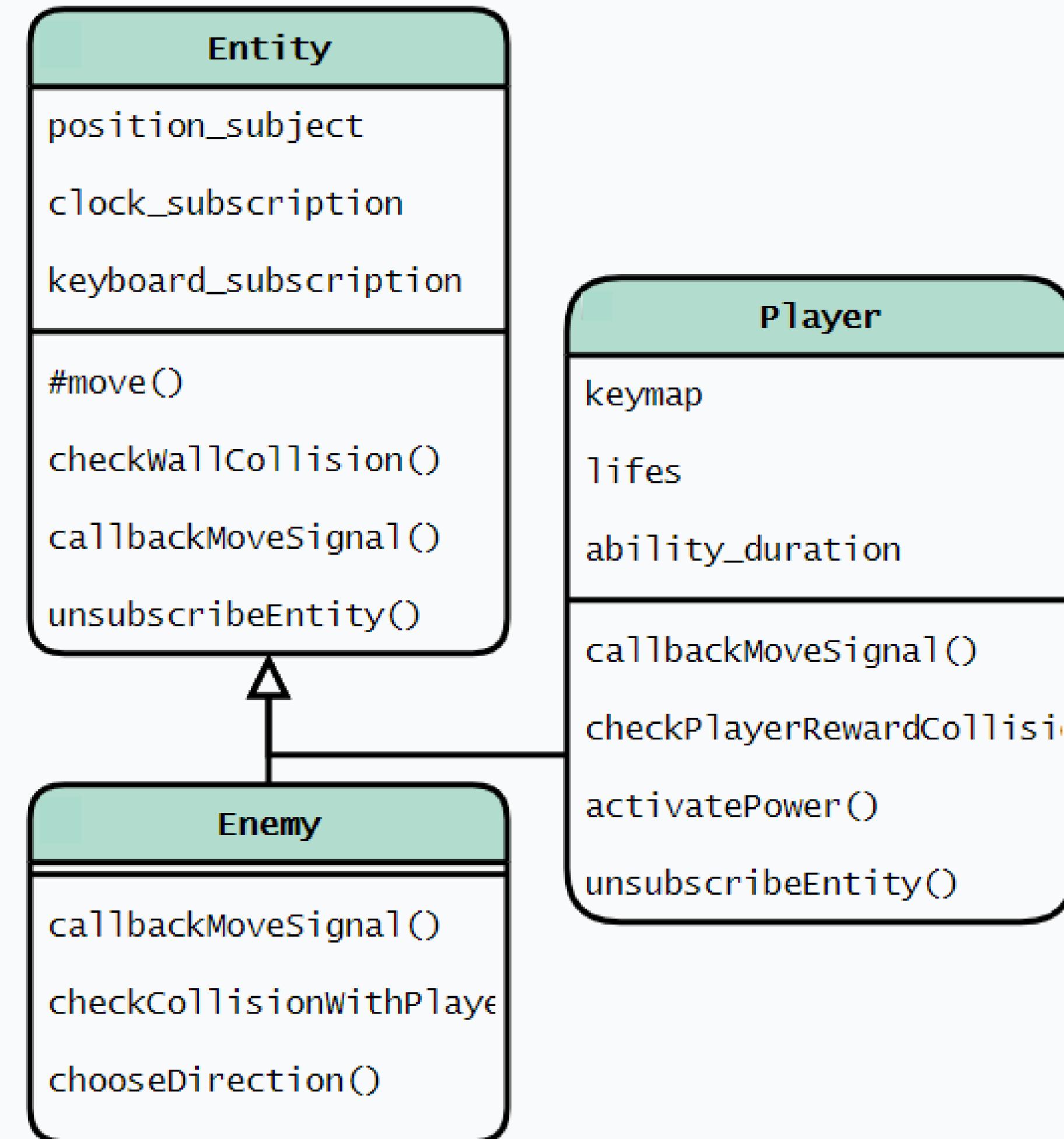
# DEMO



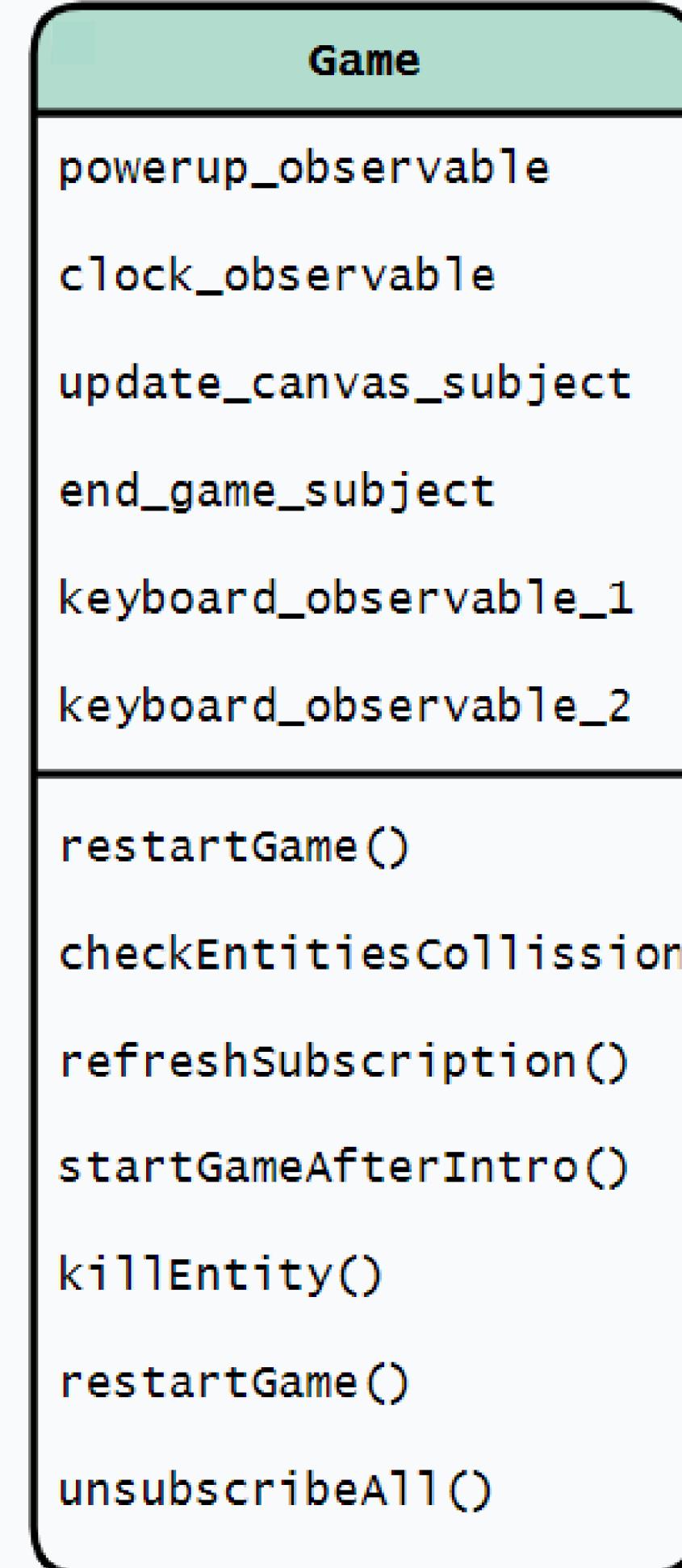
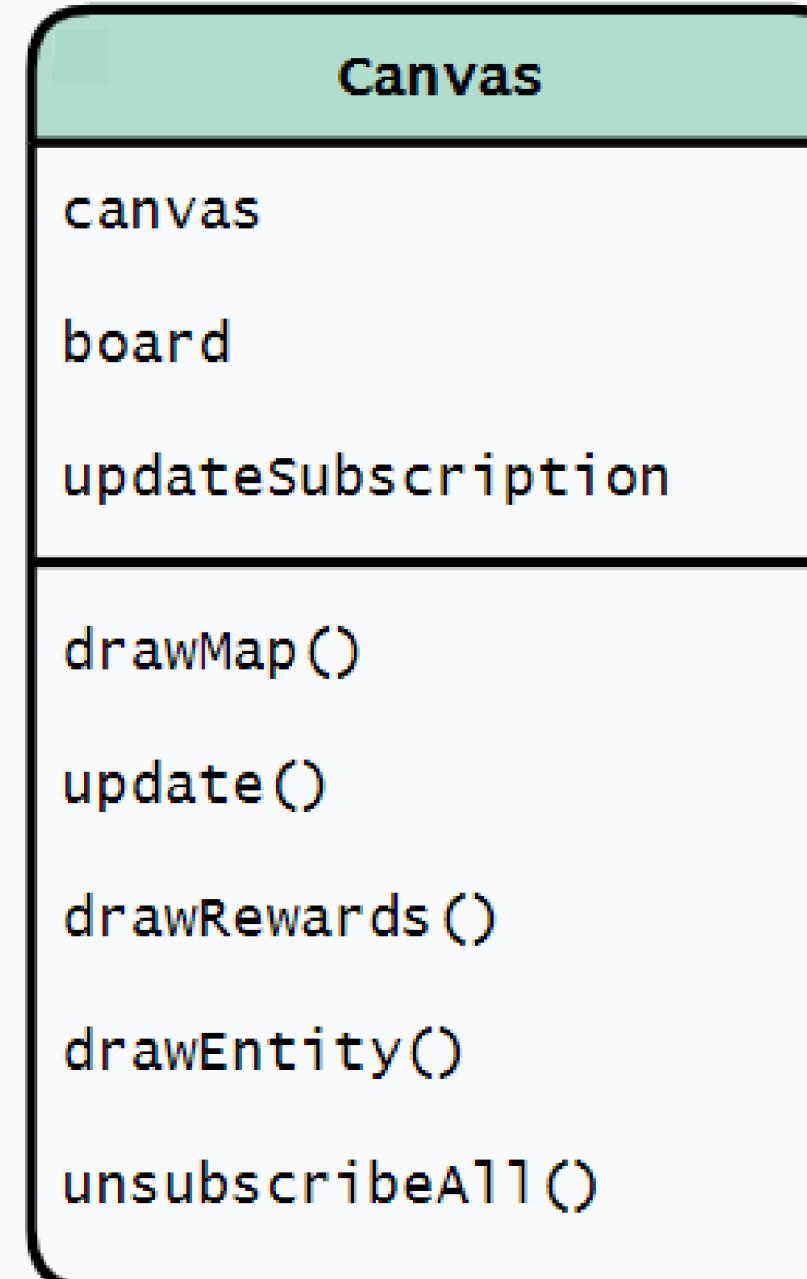
# Clases y Funciones



# Clases y Funciones



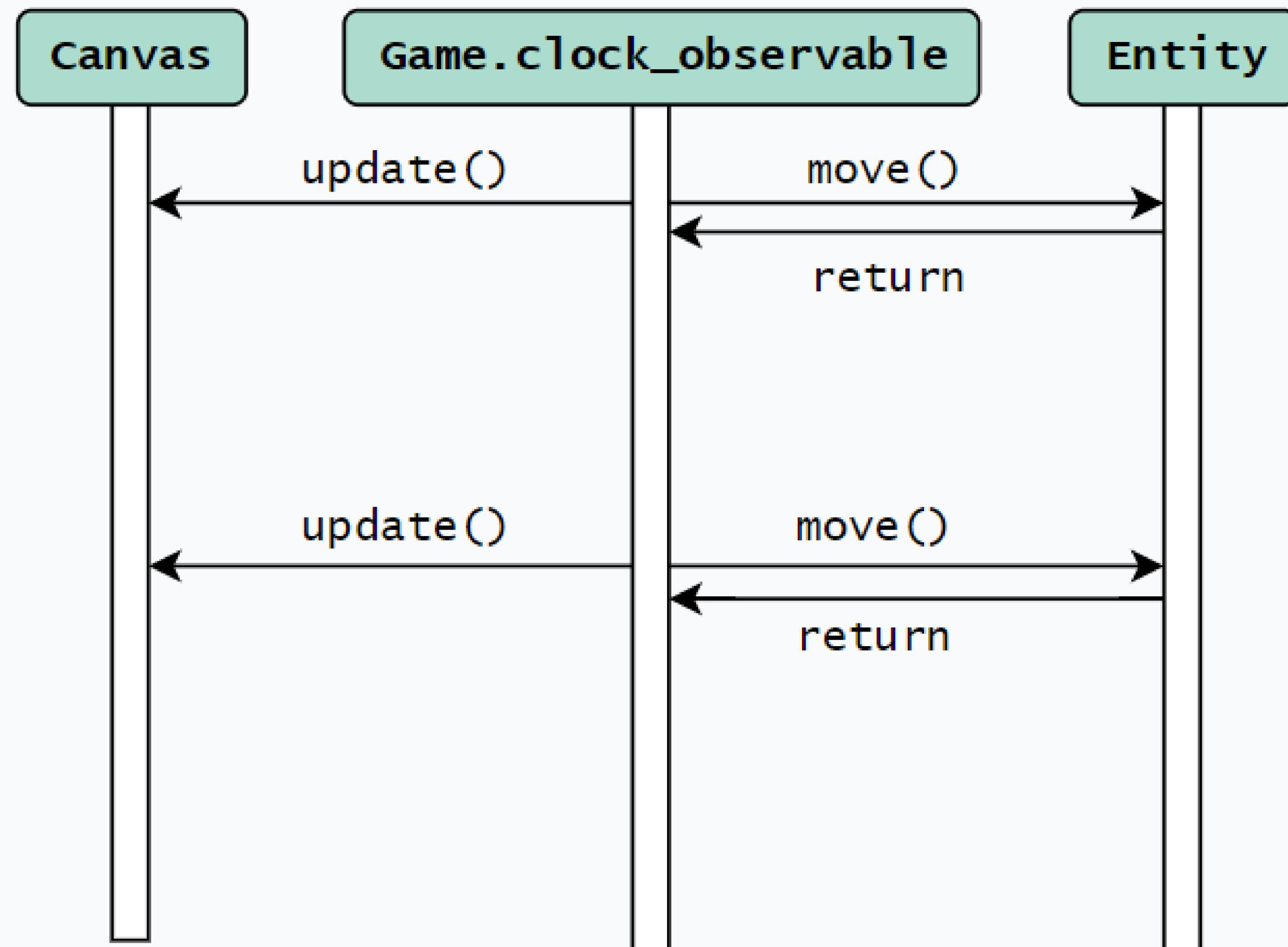
# Clases y Funciones



# Tareas Clave



# Movimiento de entidades



# Tareas Clave



# Movimiento de entidades



```
startGameAfterIntro() {
    this.players.forEach((player) => {
        player.clock_subscription = this._clock_observable.subscribe(() => {
            player.callbackMoveSignal(this.board);
            player.keyboard_subscription = this._keyboard_observables[
                player.id
            ].subscribe((direction) =>
                player.callbackKeyboardEventSignal(direction)
            );
        });
    });
    this.enemies.forEach((enemy) => {
        enemy.clock_subscription = this._clock_observable.subscribe(() => {
            enemy.callbackMoveSignal(this.board, this.players);
        });
    });
    this.refreshSubscription();
}
```

# Problema Encontrado



## Skipear Intro



```
skipIntro() {  
    //...  
    document.removeEventListener("keydown", this.skipIntro);  
    //...  
}  
  
startGameIntro(){  
    //Start game  
}  
  
playAudioIntro() {  
    //...  
    this.audio.addEventListener("ended", this.startGame);  
    document.addEventListener("keydown", this.skipIntro);  
    //...  
}
```

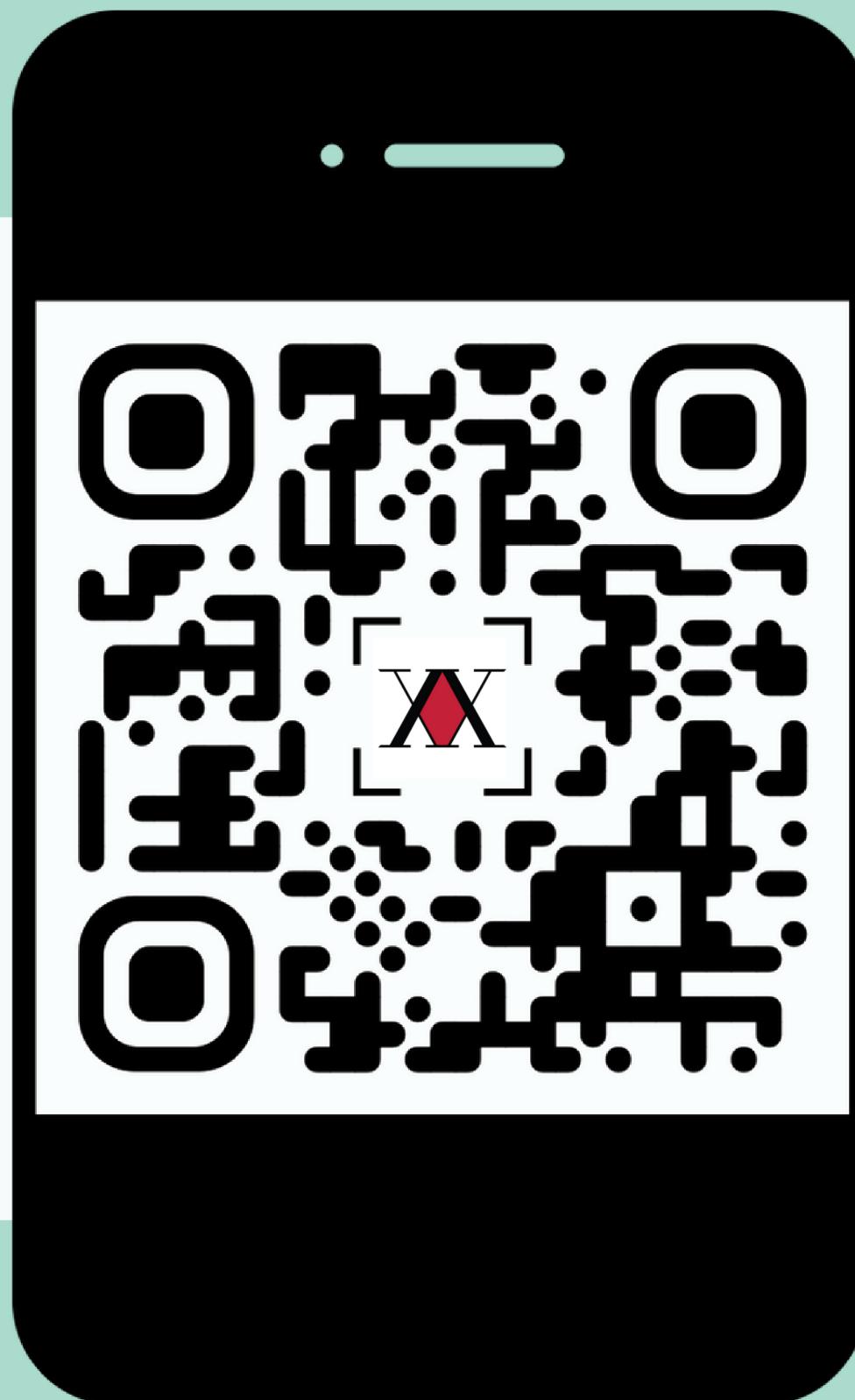
# Solución



```
playAudioIntro() {
    this.audio_intro.play();
    const control_key = rxjs
        .fromEvent(document, "keydown")
        .pipe(rxjs.filter((event) => event.key === "Control"));

    const stop_listening_subject = new rxjs.Subject();
    const audio_end_observable = rxjs
        .fromEvent(this.audio_intro, "ended")
        .pipe(rxjs.takeUntil(stop_listening_subject));
    const control_pressed_observable = control_key.pipe(
        rxjs.takeUntil(stop_listening_subject)
    );

    rxjs
        .merge(audio_end_observable, control_pressed_observable)
        .pipe(rxjs.take(1))
        .subscribe((event) => {
            if (event.type === "ended") {
                this.startGameAfterIntro();
            } else if (event.type === "keydown") {
                this.skipIntro();
            }
            stop_listening_subject.next();
        });
}
```



# Reactive JS



Grupo 4

