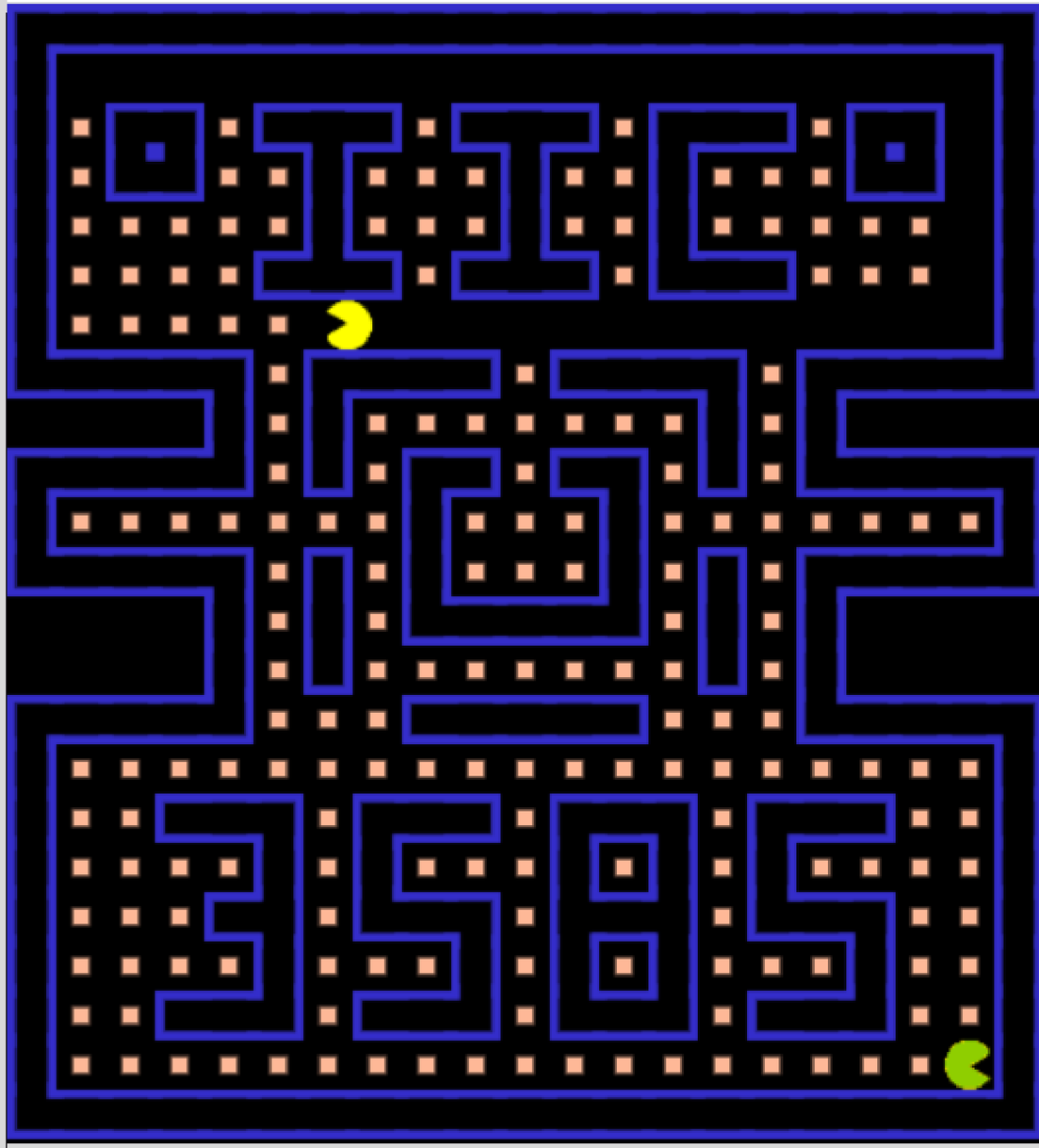
Reactive Pacman



Equipo 09

Nos basamos en

https://github.com/servetgulnaroglu/pacman-js

Código NO programado reactivamente

Contexto

- Uso de clases
- Diseño del mapa
- Configuración de "comidas" y movimiento del pacman

Inicio del juego

```
const clickCanvas$ = rxjs.fromEvent(canvas, 'click');
clickCanvas$
  .pipe(rxjs.take(1))
  .subscribe(() => gameInterval$.subscribe(), { once: true });
```

Ciclo del juego

```
// ciclo principal del juego
const gameInterval$ = rxjs.interval(1000 / fps).pipe(
  rxjs.map(() => update()),
  rxjs.map(() => draw()),
  rxjs.map(() => {
    if (map[pacman.getMapY()][pacman.getMapX()] === 2) {
      map[pacman.getMapY()][pacman.getMapX()] = 3;
      score++;
    }
  }),
  rxjs.map(() => {
    if (map[pacmanSecond.getMapY()][pacmanSecond.getMapX()] === 2) {
      map[pacmanSecond.getMapY()][pacmanSecond.getMapX()] = 3;
      score++;
    }
  })
);
```

Movimiento de los jugadores

```
const keyboardObservable = (pacman, keys) => {
  const keyboardEvent$ = rxjs.fromEvent(window, 'keydown').pipe(
    rxjs.filter((event) => {
      return Object.keys(keys).includes(event.key);
    })
  );

  keyboardEvent$.subscribe((event) => {
    pacman.nextDirection = keys[event.key];
  });
};
```

Movimiento de los jugadores

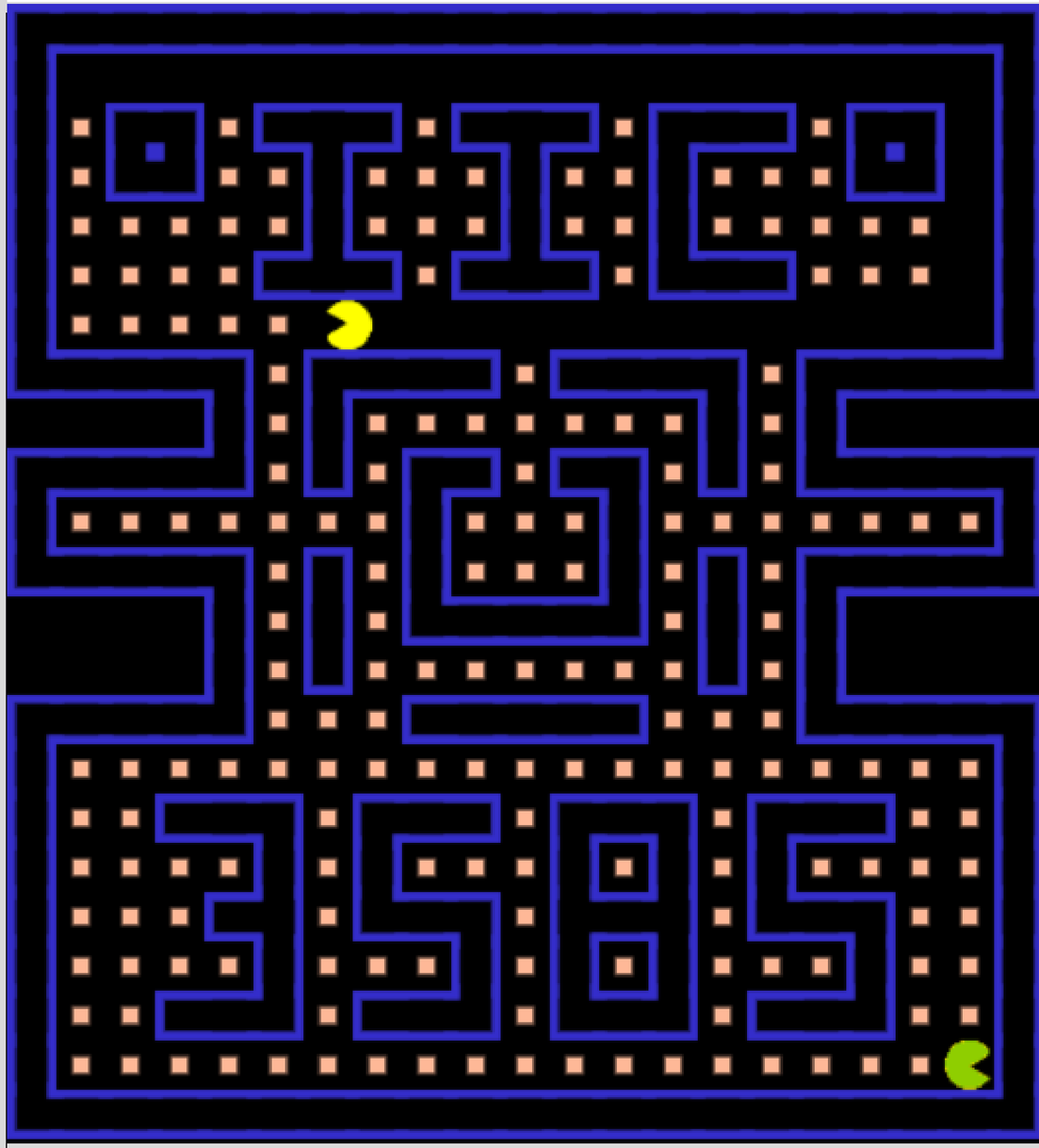
```
const keysPacman = {
  ArrowLeft: DIRECTION_LEFT,
  ArrowUp: DIRECTION_UP,
  ArrowRight: DIRECTION_RIGHT,
  ArrowDown: DIRECTION_BOTTOM,
};

const keysPacmanSecond = {
  a: DIRECTION_LEFT,
  w: DIRECTION_UP,
  d: DIRECTION_RIGHT,
  s: DIRECTION_BOTTOM,
};

// keyboard observer
keyboardObservable(pacman, keysPacman);
keyboardObservable(pacmanSecond, keysPacmanSecond);
```

Conclusiones

Reactive Pacman



Equipo 09