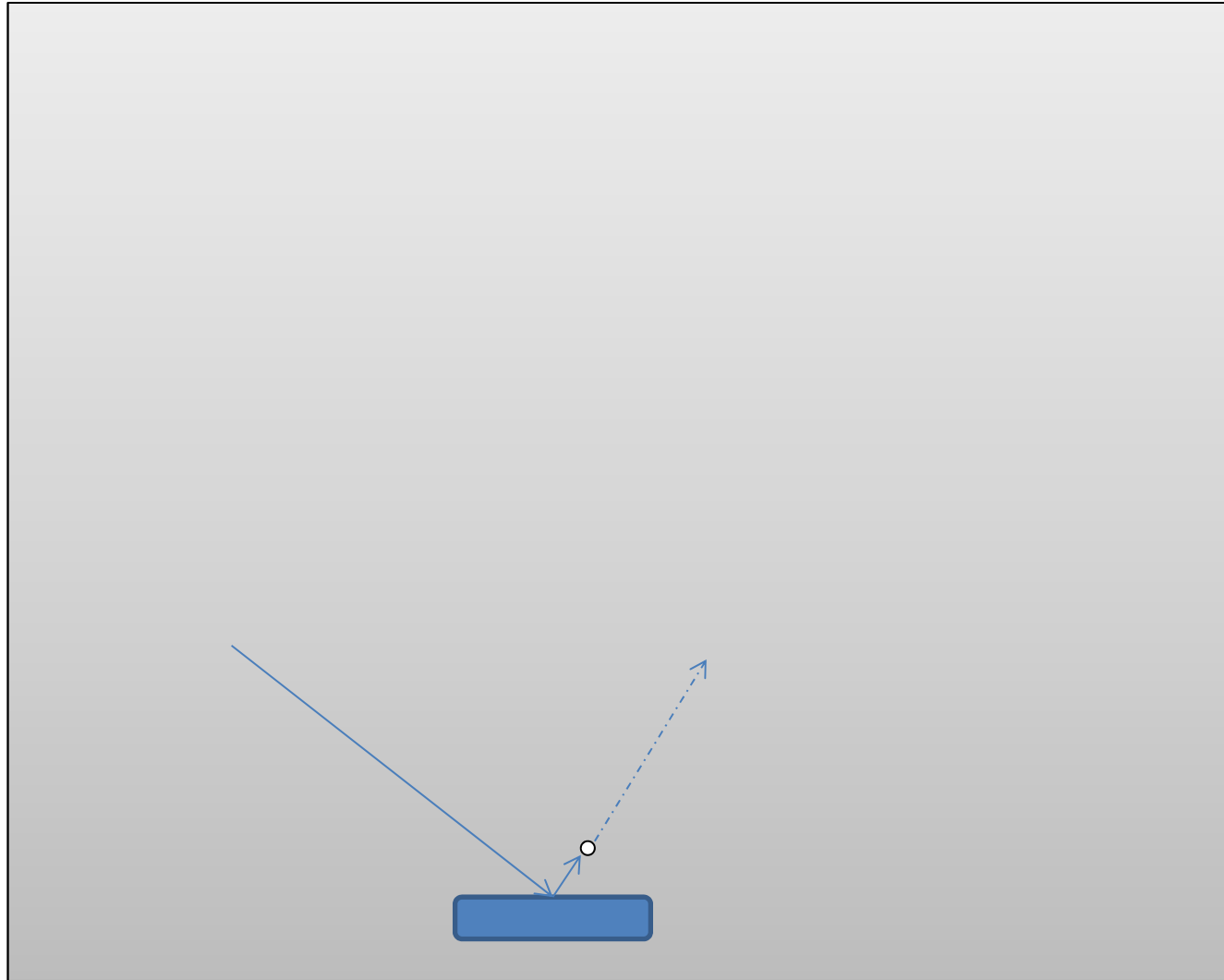
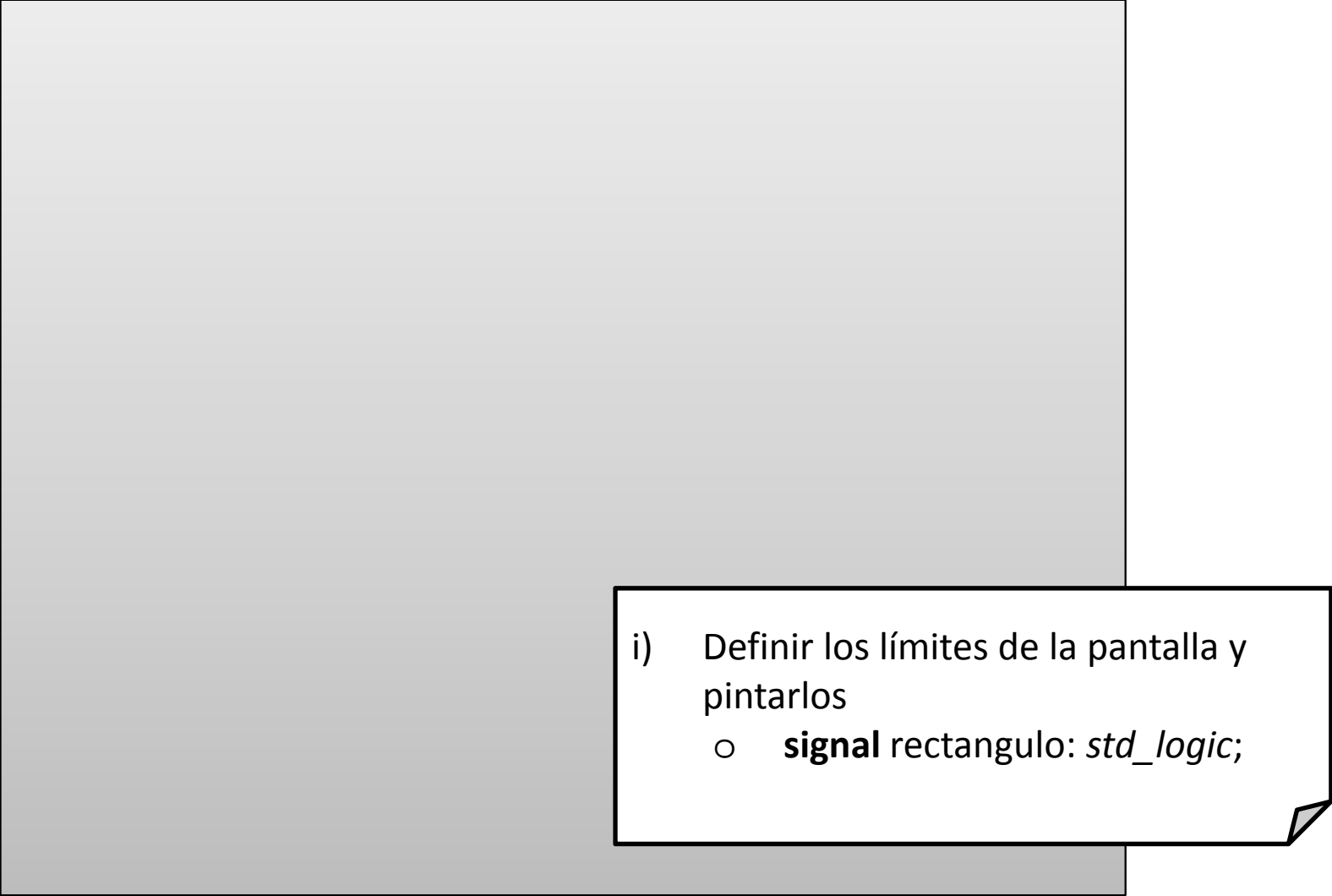


# DISEÑO DE UN JUEGO SIMPLE

# JUEGO SIMPLE



# JUEGO SIMPLE

- 
- i) Definir los límites de la pantalla y pintarlos
    - **signal** rectangulo: *std\_logic*;

# JUEGO SIMPLE

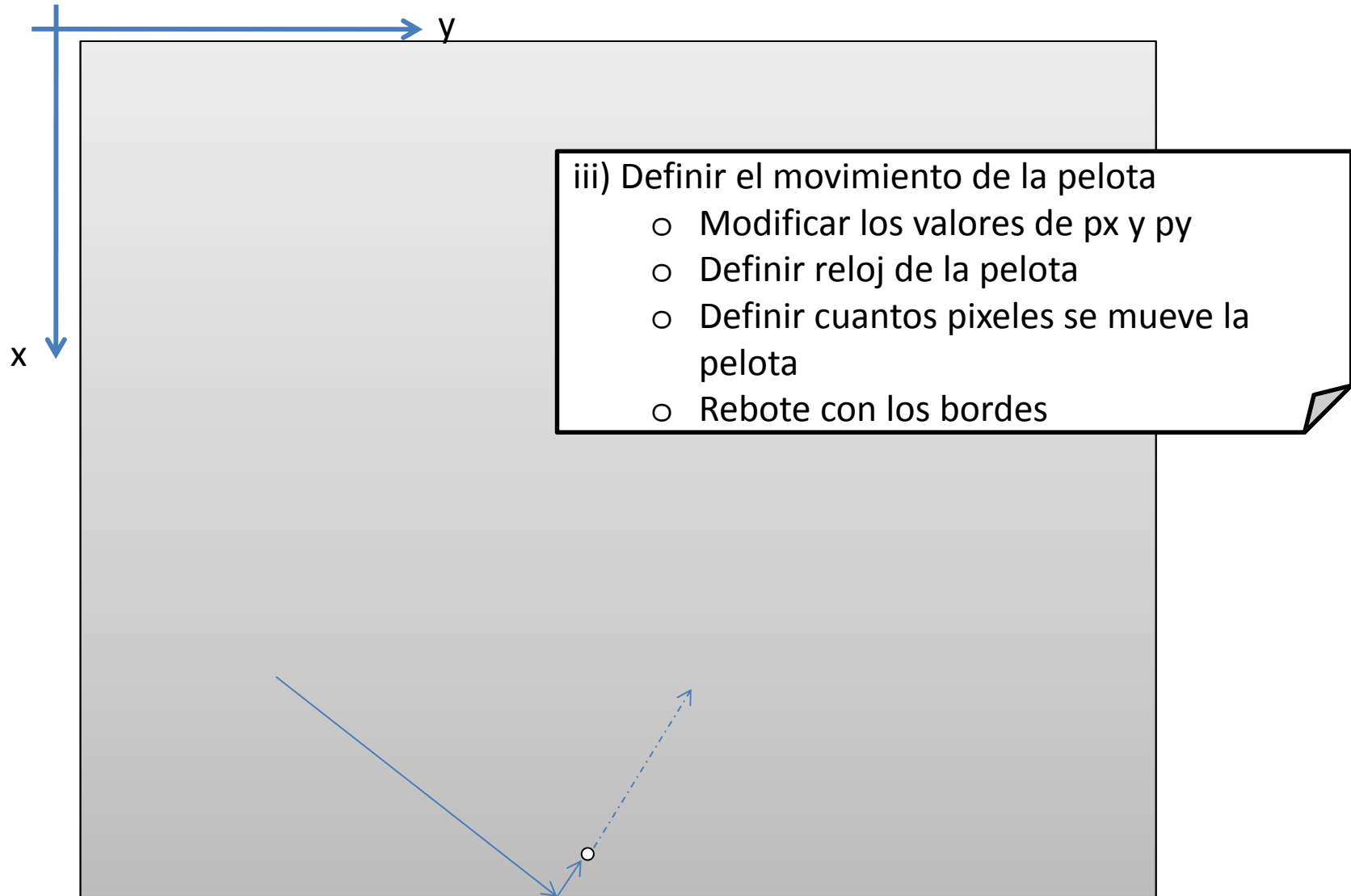


○

ii) Dibujar la pelota

- **signal** pelota: *std\_logic*;
- pelota = '1' when hcnt ...
- Se dibujará en función de r\_px y r\_py mas un offset

# JUEGO SIMPLE



### iii) Movimiento de la pelota

```
if RelojPelota'event and RelojPelota = '1' then
```

```
    r_px <= px; r_py <= py;
```

```
    -----
```

```
case EstadoPelota is
```

```
    when XpositivoYpositivo =>
```

```
        px<=r_px+1;
```

```
        py<=r_py+1;
```

```
    when XpositivoYnegativo =>
```

```
        px<=r_px+1;
```

```
        py<=r_py-1;
```

```
    ...
```

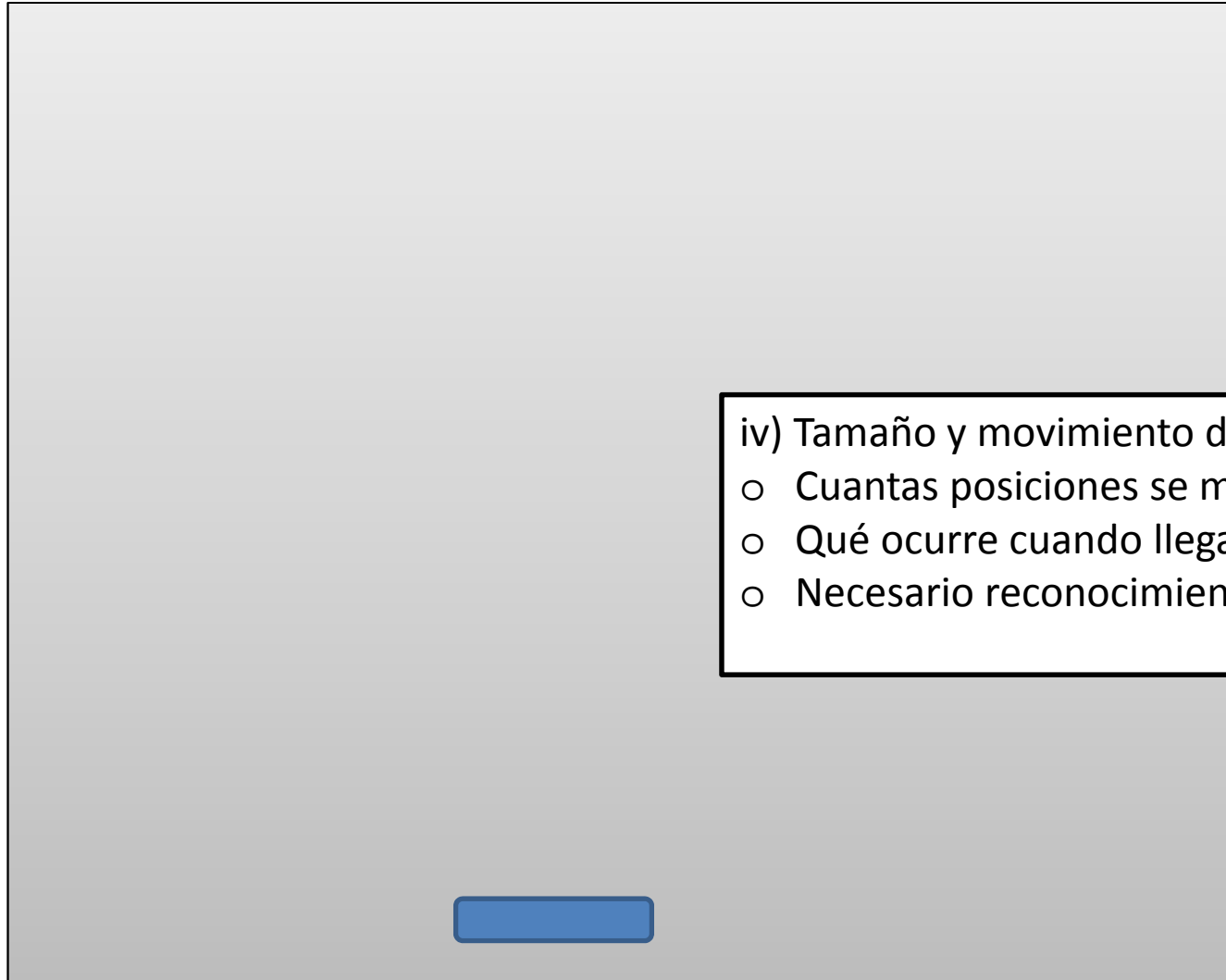
Siempre incide y  
rebota con 45º

px y py son las posiciones de la pelota en todo momento, cuando hcnt y vcnt concidan con la posición de la pelota (+ un offset) ésta se pintará

RelojPelota es la velocidad a la cual se refresca la pantalla, sin embargo el reloj del controlador (distintas máquinas de estado) deberá ser el de la FPGA

El cambio de estado dependerá del estado en el que se encuentre la pelota y el extremo del cuadrado con el que choque

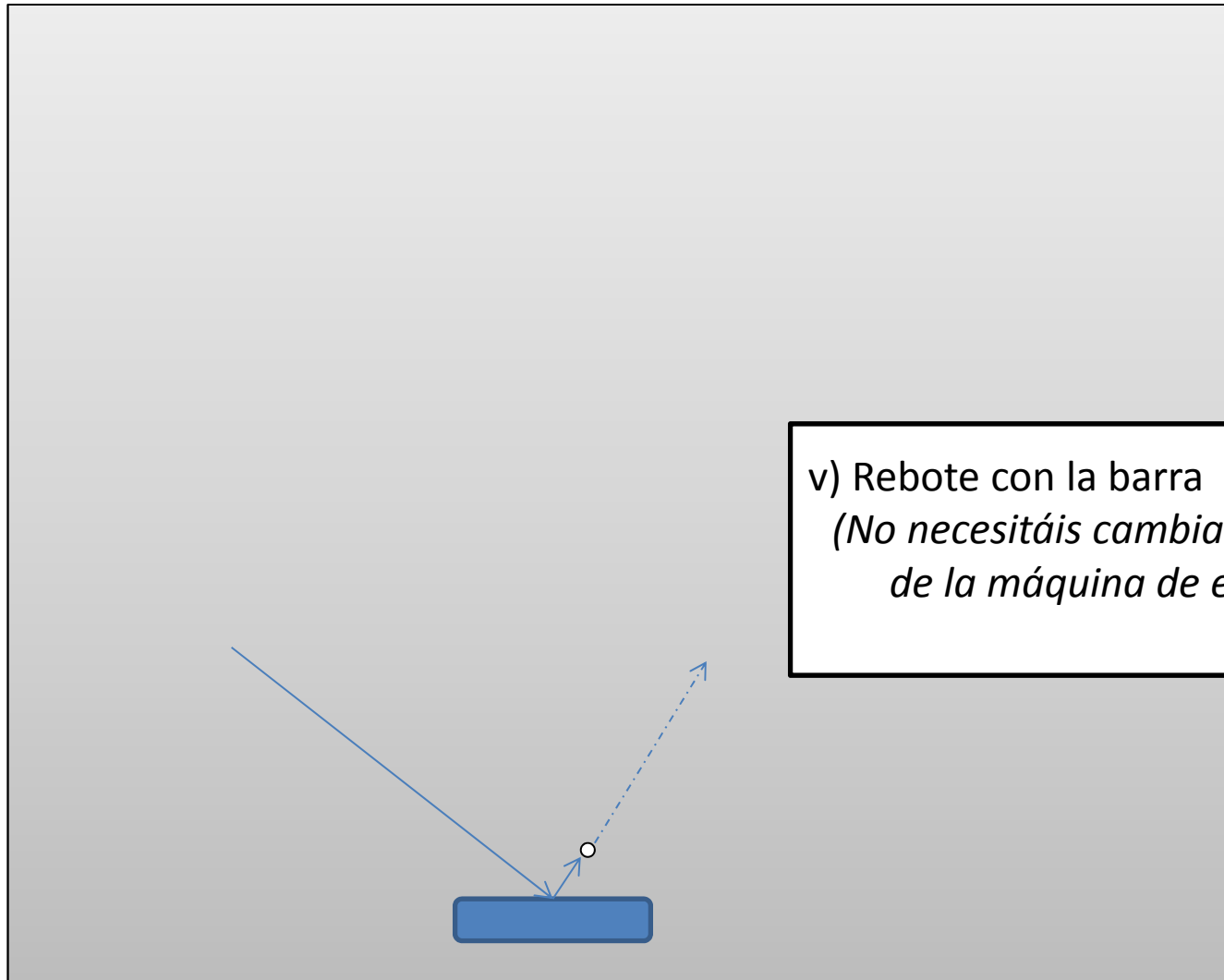
# JUEGO SIMPLE



iv) Tamaño y movimiento de la barra

- Cuantas posiciones se mueve
- Qué ocurre cuando llega al borde
- Necesario reconocimiento de tecla

# JUEGO SIMPLE



v) Rebote con la barra  
*(No necesitáis cambiar casi nada  
de la máquina de estados)*



# JUEGO SIMPLE

