

CLIPS Laboratorio 4

Rubén Agudo
Jonatan Galean
Iñigo Ochoa de Erive

17 de diciembre de 2013

ÍNDICE GENERAL

1	CREACIÓN DEL CARRETE DE 24	4
2	IMPLEMENTACIÓN DE LA CÁMARA	5
2.1	Reglas necesarias	5
A	NOTAS	7

Implementación del laboratorio 4, con las reglas necesarias para hacer funcionar la cámara

1 | CREACIÓN DEL CARRETE DE 24

Para realizar este cambio, basta con cambiar en la `defclass CARRETE_DE_DOS` en el slot `N_FOTOS` el range de 0 2 a 0 24.

Es decir:

```
1 (defclass CARRETE_DE_24
2   (is-a CARRETE)
3   (role concrete)
4   (single-slot N_FOTOS
5     (type INTEGER)
6     (range 0 24)
7     (default 0)
8     (create-accessor read-write)
9   )
10 )
```

2 | IMPLEMENTACIÓN DE LA CÁMARA

Para realizar la implementación primero hemos pensado a ver cuales eran las restricciones que se aplican a la hora de realizar una foto de manera analógica.

Y pensamos que son las siguientes:

- Que tenga un carrete puesto
 - Que ese carrete no este terminado
- Que la tapa este abierta
- Que el seguro esté quitado
- Que este cargada (en nuestra opinión esto es redundante, ya que si tiene un carrete puesto, y no está terminado obviamente está cargada.)

2.1 REGLAS NECESARIAS

Para realizar la siguiente funcionalidad implementamos las siguientes reglas.

```
1 (defrule sacarfoto
2   (eq ABIERTA (send [miCamara] get-TAPA_CIERRA))
3   (eq SI (send [miCamara] get-CARGADA))
4   (eq QUITADO (send [miCamara] get-SEGURO))
5   ?carrete <- (send [miCamara] get-PELICULA)
6   (test < (send [miCamara] get-N_FOTOS) (send ?
7     carrete get-N_FOTOS))
7 =>
8   (send [miCamara] putt-N_FOTOS (+ 1 send [
9     miCamara] get-N_FOTOS))
9 )
```

Pero entonces pensamos, ¿que sucede si la tapa esta cerrada, o el seguro quitado?

Para solventar el problema creamos las dos siguientes reglas:

```
1 (defrule quitarSeguro
2   (eq PUESTO (send [miCamara] get-SEGURO))
3 =>
4   (send [miCamara] putt-SEGURO QUITADO)
5 )
6
7 (defrule quitarTapa
8   (eq CERRADA (send [miCamara] get-TAPA_CIERRE))
9 =>
10  (send [miCamara] putt-TAPA_CIERRE ABIERTA)
11 )
```

En este instante ya somos capaces de quitar el seguro y de abrir la tapa en caso de que estuviera cerrada. Ahora solo queda cambiar el carrete si lo hemos terminado, pero para ello antes es necesario rebobinar la cámara, ya que si abrimos la cámara sin rebobinarla, se nos van a velar las fotos, y no queremos eso, ¿verdad?

```

1 (defrule rebobinar
2   (eq NO (send [miCamara] get-REBOBINADA))
3   (eq (send [miCamara] get-N_FOTOS) (send ?
        carrete get-N_FOTOS))
4 =>
5   (send [miCamara] putt-REBOBINADA SI)
6   (send [miCamara] putt-N_FOTOS o)
7   (send [miCamara] putt-PELICULA (make-instance [
        nil] of CARRETE))
8   (send [miCamara] putt-CARGADA NO)
9 )

```

Esta regla, básicamente lo que hace es, si no esta rebobinada, y el número de fotos realizada con la cámara, es igual al número de fotos que se pueden hacer con ese carrete, la rebobinamos, reiniciamos el contador de fotos de la cámara y ponemos una película nueva, en este caso utilizamos [nil] para decir que es null. Y ponemos que no esta cargada.

Por último, queda cargar un nuevo carrete.

```

1 (defrule cargarPelicula
2   (eq SI (send [miCamara] get-REBOBINADA))
3   ?nombreInstancia <- (INSTANCE-NAME-TO-SYMBOL (
        send [miCamara] get-PELICULA))
4   (eq ?nombreInstancia nil)
5   (eq NO (send [miCamara] get-CARGADA))
6 =>
7   (send [miCamara] putt-PELICULA (make-instance [
        miCarrete] of CARRETE_DE_DOS))
8   (send [miCamara] putt-CARGADA SI)
9 )

```

Lo que hace esta regla es: si tenemos la cámara rebobinada, y la película que tiene la cámara es nil, es decir, no tiene película, y no está cargada, pues creamos una nueva instancia de carrete, y se la asignamos a la cámara, y ponemos que está cargada.

Ya tenemos la cámara lista para disparar.

A | NOTAS

Hemos programado todo sin haberlo podido probar, ya que andábamos justos de tiempo. De todas formas, como se puede observar, todo lo que hemos hecho programado esta justificado, es decir, no ha sido programado al tuntún.

La idea principal es esa, este código es muy mejorable, por ejemplo haciendo que pueda trabajar con mas de una cámara, eliminando el slot CARGADA y SACAR ya que creemos que son completamente irrelevantes.