UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA EN INFORMÁTICA

MANUAL DE USUARIO
PUZZLE CANDY CRASH

Programado en Scheme.

Profesor: Roberto González

Ayudante: Roberto González

Fecha de Entrega: 15/05/2017

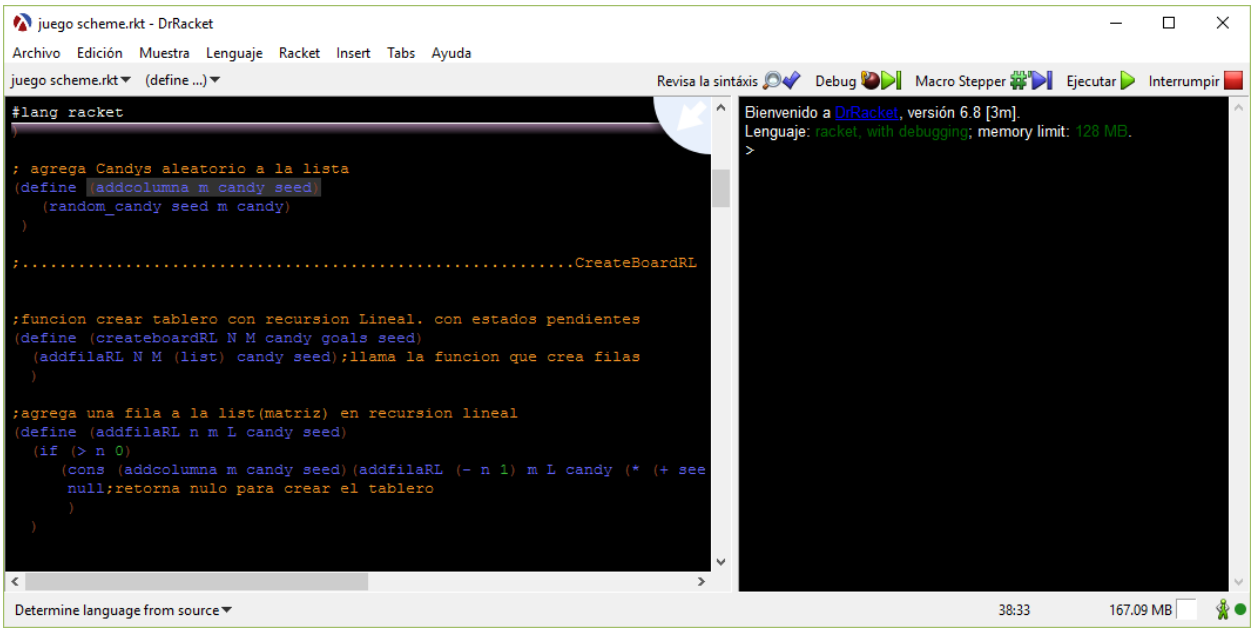
Santiago de Chile
15 - 5 – 2017

MANUAL DE USUARIO “CANDYCRASH”

En el siguiente manual, se pretende explicar la manera de operar de las reglas generadas en el programa, además de ejemplos de cómo implementarlas.

Para su ejecución es necesaria la instalación de DrRacket un intérprete de Scheme, además de conocimientos básico-intermedio sobre programación.

Luego de instalar la aplicación debe abrir el archivo (candyCrash_17411947_OrellanaTamayo.rkt), que lo enviara a la siguiente ventana donde podrás ingresar los comandos en la parte derecha de su pantalla y llamar a las funciones que contiene el juego.



Un resumen de las funciones que contiene el juego y que usted puede realizar.

1) **CreateBoardRL**
2) **CreateBoardRC**
3) **CheckBoard**
4) **Play**
5) **CheckCandie**
6) **Board→String**

CreateBoardRL

```
(define (createboardRL N M candy goals seed)
```

Esta función crea un tablero con Recursión Lineal, la función es llamada createboardRL, y esta función contiene 5 parámetros, N que contiene el tamaño de las Filas, M contiene el tamaño de las Columnas, Candy contiene la cantidad de tipos de dulces que desea dentro del tablero, el rango no puede ser menor a 3 tampoco mayor que 5, goals es la cantidad de candys que debes destruir para superar el nivel, seed es la semilla que te permite crear el mismo tablero mientras la semilla sea la misma, si cambias el número de la semilla el tablero se generara con nuevos candys. Por ejemplo:

N: corresponde al tamaño de la fila. (8)
M: corresponde al tamaño de la columna. (8)
Candy: corresponde a la cantidad de tipos de Candy en el tablero, desde el 5 hasta 7. (5)
Goals: corresponde a la cantidad de candys a reventar para superar el nivel, (10)
Seed: corresponde a la semilla para crear el tablero, si cambia la semilla cambia el tablero. (2)

```
(createboardRL 8 8 5 10 2)
```

Esta función crea un tablero de 8x8 con candys del 1 al 5, con 10 dulces que destruir para superar en nivel y con semilla (2) para generar un tablero.

```
> (createboardRL 8 8 5 10 2)
'((3 3 4 4 3 4 1 2)
  (3 3 4 5 1 3 2 1)
  (2 1 1 4 2 5 4 3)
  (5 4 2 3 2 4 5 5)
  (2 4 4 4 1 1 5 2)
  (1 4 5 5 5 2 4 3)
  (2 1 3 4 4 3 5 5)
  (1 2 4 2 4 3 4 4))
>
```

CreateBoardRC

```
(define (createboardRC N M candy goals  seed)
```

Esta función crea un tablero con Recursión de Cola, la función es llamada createboardRC, y esta función contiene 5 parámetros, N que contiene el tamaño de las Filas, M contiene el tamaño de las Columnas, Candy contiene la cantidad de tipos de dulces que desea dentro del tablero, el rango no puede ser menor a 3 tampoco mayor que 5, goals es la cantidad de candys que debes destruir para superar el nivel, seed es la semilla que te permite crear el mismo tablero mientras la semilla sea la misma, si cambias el número de la semilla el tablero se generara con nuevos candys. Por ejemplo:

N: corresponde al tamaño de la fila. (8)
M: corresponde al tamaño de la columna. (8)
Candy: corresponde a la cantidad de tipos de Candy en el tablero, desde el 5 hasta 7. (5)
Goals: corresponde a la cantidad de candys a reventar para superar el nivel, (10)
Seed: corresponde a la semilla para crear el tablero, si cambia la semilla cambia el tablero. (2)

```
(createboardRC 8 8 5 10 2)
```

Esta función crea un tablero de 8x8 con candys del 1 al 5, con 10 dulces que destruir para superar en nivel y con semilla (2) para generar un tablero.

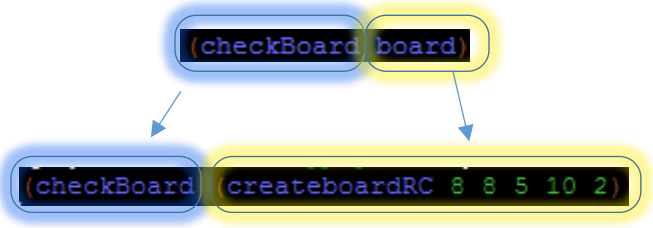
```
> (createboardRC 8 8 5 10 2)
'((1 2 4 2 4 3 4 4)
  (2 1 3 4 4 3 5 5)
  (1 4 5 5 5 2 4 3)
  (2 4 4 4 1 1 5 2)
  (5 4 2 3 2 4 5 5)
  (2 1 1 4 2 5 4 3)
  (3 3 4 5 1 3 2 1)
  (3 3 4 4 3 4 1 2))
>
```

CheckBoard

```
(define (checkBoard board)
```

Esta función recibe un tablero (board) y permite verificar si este cumple con los criterios para ser considerado un tablero valido, si el tablero es válido entrega `#t`, y si no es un tablero valido entrega `#f` la función es llamada checkBoard, y esta función contiene 1 tablero (board), como en el ejemplo anterior board tiene 5 parámetros, N que contiene el tamaño de las Filas, M contiene el tamaño de las Columnas, Candy contiene la cantidad de tipos de dulces que desea dentro del tablero, el rango no puede ser menor a 3 tampoco mayor que 5, goals es la cantidad de candys que debes destruir para superar el nivel, seed es la semilla que te permite crear el mismo tablero mientras la semilla sea la misma, si cambias el número de la semilla el tablero se generara con nuevos candys. Por ejemplo:

N: corresponde al tamaño de la fila. (8)
M: corresponde al tamaño de la columna. (8)
Candy: corresponde a la cantidad de tipos de Candy en el tablero, desde el 5 hasta 7. (5)
Goals: corresponde a la cantidad de candys a reventar para superar el nivel, (10)
Seed: corresponde a la semilla para crear el tablero, si cambia la semilla cambia el tablero. (2)



Esta función verifica el tablero de 8x8 con candys del 1 al 5, con 10 dulces que destruir para superar en nivel y con semilla (2) para generar un tablero, si todos los parámetros son verdaderos retorna un #T (verdadero).

```
> (checkBoard (createboardRC 8 8 5 10 2))
#t
>
```

Al ingresar un valor fuera de los parámetros de un tablero valido retorna #f (falso)

```
> (checkBoard (createboardRC 8 8 4 10 2)
#f
>
```

Play

```
(define (play board pOri pDest seed)
```

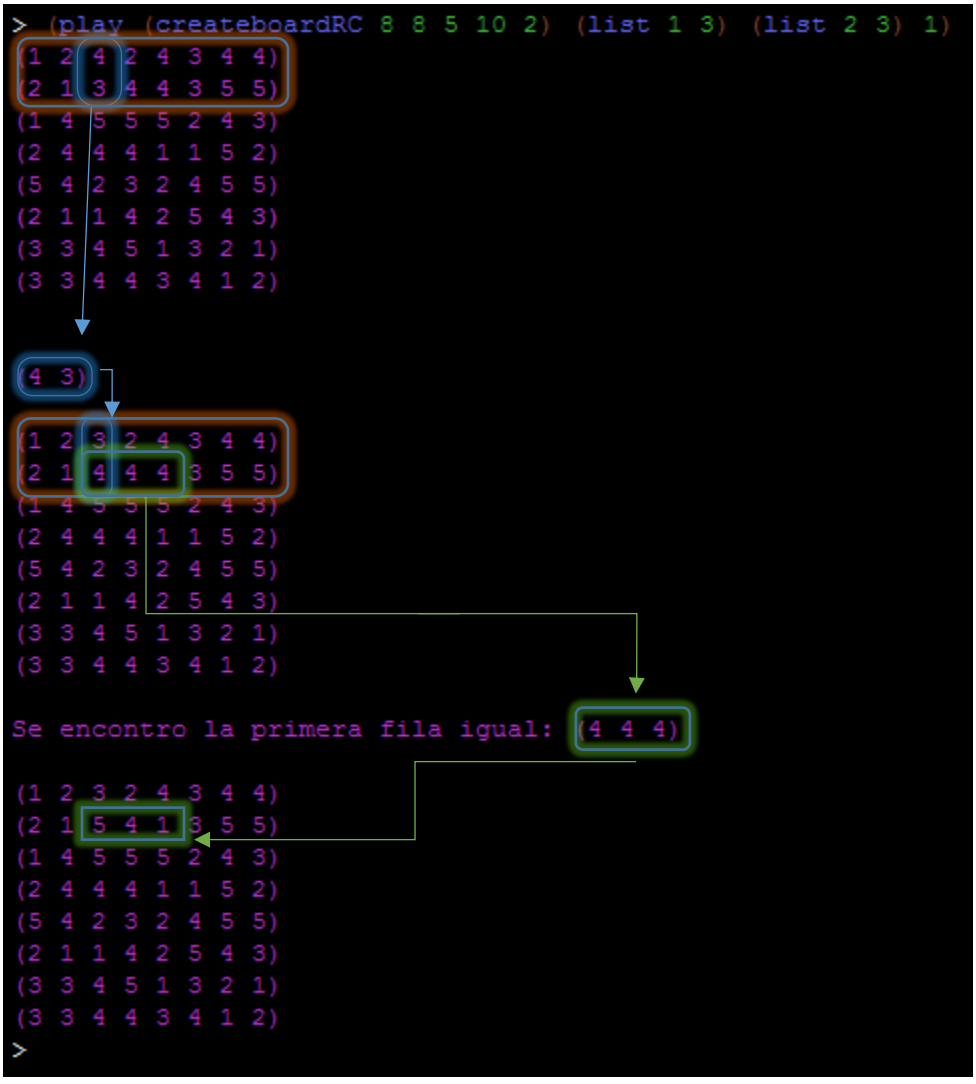
Esta función Realiza las jugadas en el tablero, la función es llamada play y recibe como parámetros un tablero (board), posición de origen de Candy (pOri), posición de destino del Candy (pDest) y una semilla (seed) para generar nuevos dulces en el caso de que se encuentre con 3 o mas candis iguales en la misma línea.

board contiene los 5 parámetros ya mencionados anterior mente
N: corresponde al tamaño de la fila. (8)
M: corresponde al tamaño de la columna. (8)
Candy: corresponde a la cantidad de tipos de Candy en el tablero, desde el 5 hasta 7. (5)
Goals: corresponde a la cantidad de candys a reventar para superar el nivel, (10)
Seed: corresponde a la semilla para crear el tablero, si cambia la semilla cambia el tablero. (2)

pOri contiene una lista con dos parámetros para el Candy origen, fila y columna. (list 3 3)
pDest contiene una lista con dos parámetros para el Candy destino, fila y columna. (list 4 3)
seed contiene la semilla en caso de que se genere una fila de 3 o más Candys iguales (1)

Esta función recibe un Candy origen en la posición fila = 1, columna = 3, (1 3) y posición de destino en fila = 2, columna = 3, (2 3). los parámetros del tablero son de 8x8 con Candys del 1 al 5, con 10 dulces a destruir para superar en nivel, con una semilla (2) para generar un tablero, y una semilla (1) para generar los Candy destruidos.

El juego nos imprime el tablero como fue creado según la semilla (2), luego no muestra los dulces que serán cambiados según las posiciones indicadas anteriormente (1 3) ↔ (2 3), posteriormente nos muestra el tablero con los dulces intercambiados e indica que encontró una línea con 3 Candys iguales (4 4 4) y finalmente nos muestra el tablero con los dulces ya eliminados y nueva mente generados aleatoriamente

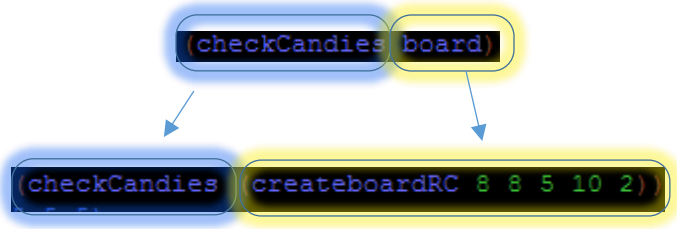


CheckCandie

```
(define (checkCandies board)
```

Esta función recibe un tablero (board) y permite verificar si 3 o más Candy están unidos en una misma fila, la función es llamada checkCandies, y esta función contiene 1 tablero como en los ejemplos anteriores.

board contiene los 5 parámetros ya mencionados anterior mente
N: corresponde al tamaño de la fila. (8)
M: corresponde al tamaño de la columna. (8)
Candy: corresponde a la cantidad de tipos de Candy en el tablero, desde el 5 hasta 7. (5)
Goals: corresponde a la cantidad de candys a reventar para superar el nivel, (10)
Seed: corresponde a la semilla para crear el tablero, si cambia la semilla cambia el tablero. (2)



Esta función comprueba si exististe alguna línea de Candys con 3 dulces iguales o más juntos en un tablero de 8x8 con candys del 1 al 5, con 10 dulces que destruir para superar en nivel y con semilla (2) para generar un tablero, retorna la primera línea de dulces iguales encontrados.

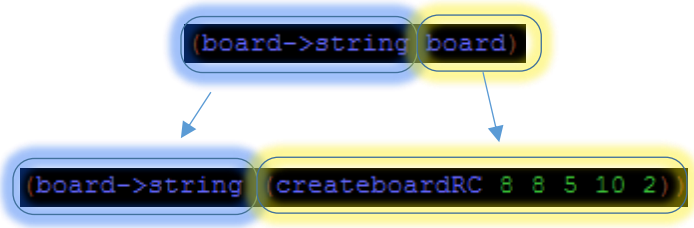
```
> (checkCandies (createboardRC 8 8 5 10 2))
'(5 5 5)
```

Board→String

```
(define (board->string board)
```

Esta función recibe un matriz (board) e imprime el tablero por pantalla para una mejor visualización para el usuario, la función es llamada board→string, y esta función contiene 1 tablero como en los ejemplos anteriores.

board contiene los 5 parámetros ya mencionados anterior mente
N: corresponde al tamaño de la fila. (8)
M: corresponde al tamaño de la columna. (8)
Candy: corresponde a la cantidad de tipos de Candy en el tablero, desde el 5 hasta 7. (5)
Goals: corresponde a la cantidad de candys a reventar para superar el nivel, (10)
Seed: corresponde a la semilla para crear el tablero, si cambia la semilla cambia el tablero. (2)



Esta función imprime el tablero de 8x8 con candys del 1 al 5, con 10 dulces que destruir para superar en nivel y con semilla (2) para generar un tablero, retorna el tablero

```
> (board->string (createboardRC 8 8 5 10 2))
(1 2 4 2 4 3 4 4)
(2 1 3 4 4 3 5 5)
(1 4 5 5 5 2 4 3)
(2 4 4 4 1 1 5 2)
(5 4 2 3 2 4 5 5)
(2 1 1 4 2 5 4 3)
(3 3 4 5 1 3 2 1)
(3 3 4 4 3 4 1 2)
```

Se Imprime un tablero de 12x20

```
> (board->string (createboardRC 12 20 5 10 2))
(2 4 5 5 3 4 4 4 5 2 5 4 1 4 1 3 5 1 3 1)
(2 1 3 3 2 1 1 4 2 5 2 4 5 2 5 3 3 5 2 4)
(1 1 2 4 2 3 2 4 2 1 3 5 3 4 4 5 3 5 2 5)
(5 5 5 3 4 4 3 4 2 1 2 5 1 3 3 5 4 4 2 4)
(1 2 4 2 4 3 4 4 4 4 3 2 3 3 4 3 3 5 1 4)
(2 1 3 4 4 3 5 5 1 5 2 5 5 3 5 2 4 3 1 3)
(1 4 5 5 5 2 4 3 4 2 3 1 1 2 4 5 4 4 4 2)
(2 4 4 4 1 1 5 2 5 2 5 1 2 2 1 1 3 3 4 2)
(5 4 2 3 2 4 5 5 1 4 5 1 4 3 1 3 5 2 1 5)
(2 1 1 4 2 5 4 3 1 2 4 1 4 2 5 3 3 2 3 2)
(3 3 4 5 1 3 2 1 2 1 2 4 3 3 4 4 4 4 1 5)
(3 3 4 4 3 4 1 2 5 3 3 2 1 1 3 5 3 3 5 1)
```