

# Paradigmas de Programación

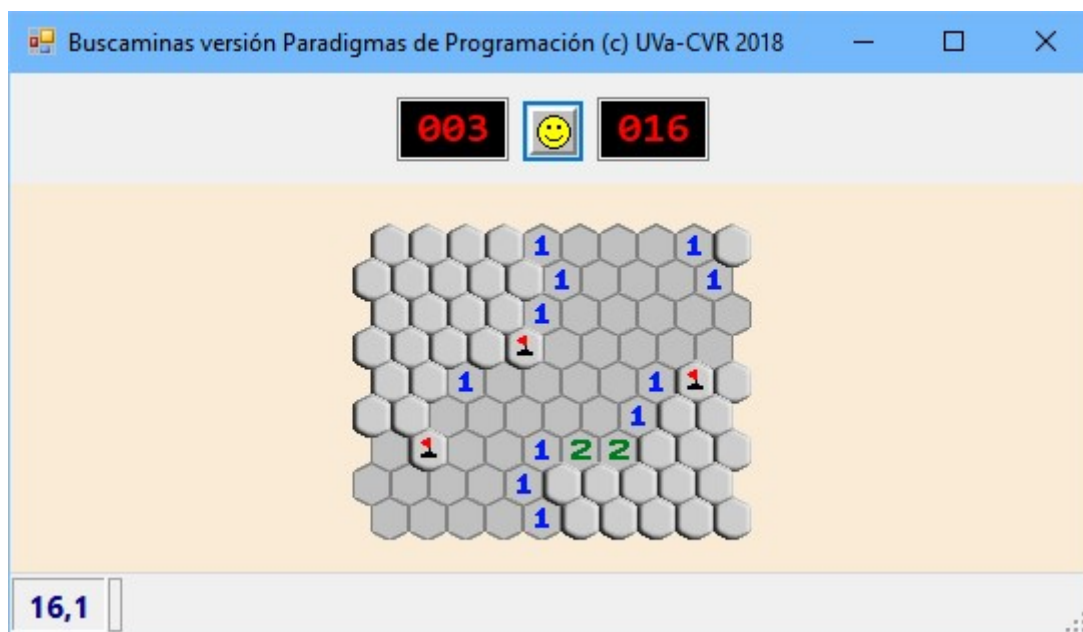
## Práctica I - Curso 2017/18

### Buscaminas (I)

#### 1. Objetivo de la Práctica

El objetivo de la práctica es el desarrollo de una aplicación en lenguaje Python que implemente el legendario juego Buscaminas con algunas modificaciones, las más importantes son:

- Se jugará en una rejilla hexagonal, por lo que cada celda tendrá 6 vecinas en vez de las 8 del juego original.
- Los números de las celdas abiertas indicarán las minas **por descubrir** en las celdas vecinas, en vez del número de minas existentes.
- Se contará el tiempo transcurrido desde el inicio de cada partida.
- Además de los tableros de tamaños estándar, generados al azar, será posible leer un tablero definido en un fichero de texto.



En ésta primera práctica el juego tendrá una entrada/salida **por consola**. En la segunda práctica se le dotará de una interfaz gráfica.

El objetivo de este juego es el de encontrar todas las minas escondidas en un área dividida en celdas mediante el marcado y apertura de celdas (el marcado cuando se supone que existe mina y la apertura cuando se supone que no existe). Como ayuda al jugador se le muestra información, en las celdas abiertas, sobre el número de minas existentes en las celdas cerradas que la rodean (en nuestro caso el número de minas por descubrir – descontando las ya marcadas – en las celdas vecinas).

## 2. Descripción general

La aplicación consistirá en un bucle principal en el que cada iteración corresponde al juego de una partida, dividida en dos etapas:

- 1) Generación del tablero: Se presentará un menú (ver 3.3) que permitirá elegir entre las opciones:
  - Generar un tablero al azar de entre 3 opciones predefinidas con dimensiones y número de minas prefijadas
  - Leer un fichero de definición de tablero (ver 3.4)
  - Terminar la aplicación
- 2) Desarrollo de la partida: Consistirá en un bucle en el que se presentará el estado del tablero (ver 3.1) y se pedirá al usuario que introduzca su(s) jugada(s) (ver 3.5), las cuales se validarán (ver 3.6) y evaluarán (ver 3.7) tras lo cual se actualizará el estado (ver 3.2) y se detectará (ver 3.8) si el juego termina (por explosión de una mina o porque se han marcado o abierto todas las celdas).

## 3. Descripción detallada

### 3.1 Representación del tablero

El tablero se puede representar internamente como una matriz rectangular con un determinado número de filas y columnas. El que la rejilla sea hexagonal sólo afecta a la manera de localizar las celdas vecinas, ya que vamos a establecer un mismo número de columnas para todas las filas. Respecto a la determinación de celdas vecinas, hay que darse cuenta de que los desplazamientos de fila/columna son distintos en las filas impares respecto a las filas pares.

En el recuadro siguiente se muestra un ejemplo del tablero indicado en el apéndice A en un momento dado del juego (tras las jugadas Hd\*De!Gb!Ei! – se explicará esta notación en 3.5):

	a	b	c	d	e	f	g	h	i	j
A	█	█	█	█	1				1	█
B	█	█	█	█	█	1				1
C	█	█	█	█	1					
D	█	█	█	█	X					
E	█	█	1					1	X	█
F	█	█						1	█	█
G		X			1	2	2	█	█	█
H					1	█	█	█	█	█
I					1	█	█	█	█	█

En el tablero de ejemplo se ha marcado en amarillo las celdas pertenecientes a la columna 6 (g), se puede apreciar que, dado el enrejado hexagonal, la posición "de dibujo" de cada celda está desplazada según la fila sea par o impar.

También se ha coloreado en azul la celda en posición (4,2) para mostrar sus 6 celdas vecinas en naranja. Las celdas situadas en los bordes pueden tener menos de 6 vecinas.

Información sobre la representación:

- Un tablero no puede tener más de 30 filas ni 30 columnas.
- Las filas se nombran con una letra de la secuencia ABCDEFGHIJKLMNOPQRSTUVWXYZ@#%&
- Para las columnas se usa la secuencia abcdefghijklmnopqrstuvwxyz+ - : /
- Se recomienda mostrar el tablero en consola usando una representación igual o similar a la que se utiliza en éste documento, para ello es necesario usar los caracteres Unicode que se indican en el apéndice B

En el recuadro de la izquierda se muestra la forma de **nombrar** cada celda y en el de la derecha la fila y columna de cada celda:

	a	b	c	d	e	f	g	h	i	j
A	Aa	Ab	Ac	Ad	Ae	Af	Ag	Ah	Ai	Aj
B	Ba	Bb	Bc	Bd	Be	Bf	Bg	Bh	Bi	Bj
C	Ca	Cb	Cc	Cd	Ce	Cf	Cg	Ch	Ci	Cj
D	Da	Db	Dc	Dd	De	Df	Dg	Dh	Di	Dj
E	Ea	Eb	Ec	Ed	Ee	Ef	Eg	Eh	Ei	Ej
F	Fa	Fb	Fc	Fd	Fe	Ff	Fg	Fh	Fi	Fj
G	Ga	Gb	Gc	Gd	Ge	Gf	Gg	Gh	Gi	Gj
H	Ha	Hb	Hc	Hd	He	Hf	Hg	Hh	Hi	Hj
I	Ia	Ib	Ic	Id	Ie	If	Ig	Ih	Ii	Ij

	a	b	c	d	e	f	g	h	i	j
A	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
B	1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9
C	2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9
D	3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9
E	4,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9
F	5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9
G	6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9
H	7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9
I	8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9

### 3.2 Estado de las celdas

El estado de cada celda en un momento dado de la partida depende de 3 valores de tipo lógico: Si la celda está **abierta** o cerrada, si ha sido **marcada** o no y si contiene o no una **mina**.

Al comienzo de la partida todas las celdas están cerradas y sin marcar, y contienen o no minas dependiendo de la forma en que se ha inicializado el tablero (al azar o mediante lectura de fichero).

El carácter que se muestra en pantalla para cada celda viene dado por las siguientes reglas, donde se define  $n = \text{número de celdas vecinas con mina} - \text{número de celdas vecinas marcadas}$  (es decir, el número estimado de minas **por descubrir** en celdas vecinas):

En situación normal (ninguna celda abierta contiene mina, todas las celdas marcadas están cerradas):

- '■' si está cerrada y no marcada (celda *sombreada*)
- 'X' si está cerrada y marcada
- ' ' si está abierta y  $n = 0$
- '?' si está abierta y  $n < 0$  (se han marcado un número excesivo de celdas vecinas)
- El dígito que indica  $n$  si está abierta y  $n > 0$

Al revelar el tablero (ya sea por explosión de mina o por finalización correcta) todas las celdas pasan a estado abierto, y se pueden dar los siguientes estados adicionales:

- '#' si está abierta, marcada, y **no** contiene mina (marcado erróneo)
- '\*' si está abierta, **no** marcada, y contiene mina (ausencia de marcado)

### 3.3 Menú de inicio de partida

```
BUSCAMINAS
-----
1. Principiante (9x9, 10 minas)
2. Intermedio (16x16, 40 minas)
3. Experto (16x30, 99 minas)
4. Leer de fichero
5. Salir

Escoja opcion: _
```

La apariencia de éste menú será similar a la que se muestra en el recuadro. Las opciones 1-3 crearán un tablero con las dimensiones (número de filas y columnas) indicadas en el que se colocarán el número de minas indicado en posiciones al azar.

Si se elige la opción 4 se preguntará por el nombre del fichero y el tablero se creará de acuerdo a los datos leídos en el fichero (ver siguiente sección).

**Nota:** No es necesario que la aplicación compruebe la existencia ni la corrección del fichero.

Si se elige la opción 5 se terminará la aplicación. Cualquier otro valor distinto de 1-5 provocará que se vuelva a preguntar por la opción a elegir.

### 3.4 Formato de los ficheros de definición de tableros

Son ficheros de texto que constan de una primera línea donde se indica el número de filas y columnas del tablero (2 enteros en el rango 1..30 separados por un espacio) a la que siguen una serie de líneas (una por cada fila) de igual longitud (un carácter por columna) donde la presencia de una mina se indica con el carácter asterisco (\*).

Ejemplo: El recuadro indica el contenido de un fichero que define un tablero de 2 filas y 4 columnas con 3 minas situadas en las celdas (0,0), (0,2) y (1,3).

```
2 4
*.*
...*
```

### 3.5 Petición de la(s) jugada(s)

Existen dos tipos de acciones que se pueden realizar: **marcar** una celda (cuando se cree que contiene una mina) y **abrir** una celda (cuando se cree que no la contiene). La manera de indicar cada acción es nombrar la celda seguida de un carácter de exclamación (!) si se desea marcar y de un carácter asterisco (\*) si se desea abrir. El usuario **puede indicar más de una acción** en la misma jugada, se entiende que en ese caso se ejecutan en secuencia **mientras** sean acciones válidas (y no se produzca una explosión, claro). Si se encuentra una acción no válida se muestra el mensaje de error adecuado (ver siguiente sección) y el resto de acciones de la cadena se desecha.

No existe una acción específica para **desmarcar** una celda, simplemente si se pide marcar una celda ya marcada se quita la marca existente (la celda sigue cerrada, por supuesto).

Si se pide **abrir una celda ya abierta**, se comprueba si su número de minas por descubrir<sup>1</sup> es menor o igual a cero, y en ese caso se entiende que se desean abrir todas sus celdas vecinas cerradas y no marcadas (si no es menor o igual a cero se muestra un error, ver siguiente sección)

<sup>1</sup> El valor  $n$  definido en la sección 3.2

Se mostrará el estado del tablero, el número de celdas marcadas, el número de minas por descubrir y el tiempo transcurrido y se pedirá al usuario que introduzca su jugada(s) con un formato similar al mostrado:

MINAS RESTANTES: 16   MARCADAS: 3   TIEMPO: 12960.6 seg.										
	a	b	c	d	e	f	g	h	i	j
A	■	■	■	■	1				1	■
B	■	■	■	■	■	1				1
C		■	■	■	■	1				
D	■	■	■	■	X					
E		■	■	1					1	X
F	■	■						1	■	■
G			X			1	2	2	■	■
H					1	■	■	■	■	■
I					1	■	■	■	■	■

Indique celda y accion (! marcar, \* abrir): Hf!Ed\*

En el ejemplo (cadena en rojo) el usuario pediría marcar la celda Hf y abrir las celdas vecinas de la celda Ed (porque esa celda ya está abierta y su número de minas a descubrir es cero).

### 3.6 Validación de cada jugada

Tal como se ha indicado en la sección anterior una jugada puede constar de varias acciones, cada una de las cuales consta siempre de tres caracteres. Una acción es inválida en los siguientes casos:

- Cuando tiene menos de tres caracteres o la fila o columna están mal especificadas o el tercer carácter no es ! o \*: Se debe escribir "ENTRADA ERRONEA".
- Cuando es una acción de marcado y si se llevara a cabo el número total de celdas marcadas en el tablero pasaría a ser superior al número de minas: Se debe escribir "NO SE PUEDEN MARCAR MAS CELDAS QUE MINAS"
- Cuando se pide abrir una celda marcada: Se debe escribir "NO SE PUEDE ABRIR UNA CELDA MARCADA"
- Cuando se pide marcar una celda abierta: Se debe escribir "NO SE PUEDE MARCAR UNA CELDA ABIERTA"
- Cuando se pide abrir una celda ya abierta en la que su número de minas a descubrir<sup>2</sup> es mayor que cero: Se debe escribir "CELDA YA ABIERTA. NO SE PUEDEN ABRIR LAS CELDAS VECINAS POR NUMERO INSUFICIENTE DE MARCAS"

Cuando una acción es inválida no se realiza, se omiten las que le pudieran seguir en la jugada, y se vuelve a pedir al usuario que introduzca una nueva jugada.

<sup>2</sup> El valor  $n$  definido en la sección 3.2

### 3.7 Evaluación de una acción válida

El pedir marcar una celda cambia su estado de no marcada o marcada o viceversa. Atención porque esto cambiará el número mostrado en las celdas vecinas abiertas (ver 3.2)

El pedir abrir una celda puede tener las siguientes consecuencias, aparte de cambiar su estado:

- Si la celda contiene una mina, se termina la partida (ver siguiente sección)
- Si el número de minas a descubrir (ver 3.2) de la celda es menor o igual que cero entonces se abren **recursivamente** todas sus celdas vecinas no abiertas ni marcadas

### 3.8 Detección de fin de partida

Una partida termina cuando:

- Se abre una celda que contiene una mina, ya sea por indicación directa de la acción del usuario o por la apertura recursiva mencionada en el apartado anterior. En este caso se muestra un mensaje indicando al usuario que ha perdido la partida.
- Cuando se cumple que hay tantas celdas marcadas como minas y que todas las celdas del tablero se han abierto o están marcadas. En ese caso se muestra un mensaje al usuario indicando que ha ganado la partida y mostrándole el tiempo en segundos que ha tardado en completarla.

En ambos casos se muestra el tablero con **todas las celdas abiertas**, el mensaje descrito, y se pasa al menú de inicio de partida.

## 4. Presentación y Evaluación de la práctica

La práctica se realizará **por parejas** (para otras alternativas consulte con su profesor de prácticas) y su evaluación se divide en dos etapas:

1. Presentación electrónica del/los ficheros que componen la práctica (ficheros \*.py). Para ello se habilitará en el Aula Virtual de la E.T.S. Informática ([www.inf.uva.es](http://www.inf.uva.es) -> menú Aula Virtual) una tarea de subida de ficheros cuya fecha límite será el **domingo 18 de marzo a las 23:59**. Al principio de todos los ficheros debe aparecer un comentario con el nombre de quienes la han desarrollado.
2. Evaluación **presencial**, en laboratorio, ante el profesor. Se realizará en el lugar, día y hora correspondiente al horario de prácticas del subgrupo al que pertenezca durante la semana del 19 al 23 de marzo.

En el caso de realización por parejas (la situación habitual), tan sólo es necesario que uno cualquiera de ellos realice la presentación electrónica. En la evaluación, sin embargo, si es necesaria la presencia de **ambos** y la evaluación puede ser distinta para cada uno de ellos.

En la evaluación de la práctica se tendrá en cuenta, entre otros, los siguientes factores:

- Autoría y participación en la misma.
- La correcta resolución del problema así como la modularidad, documentación y robustez de la solución presentada. El uso de orientación al objeto y otras técnicas contempladas en la asignatura tiene una influencia positiva en la evaluación.

## Apéndice A: Tablero de ejemplo y algunos casos de prueba

9 10

```

**.....*
.**.*.....
*.*.....
.**.*.....
.....*
.....*
*.....
.....****
.....*
```

### Partida correcta:

```

Gd*Gb!Ac*De!Hf!Ei!Aj!Be!Hg!Fc*Hh!Hi!Aa!Ab!
Ba*Cb*Bb!Bc!Ca!Cc!Db!Dc!Fi!Ea*Ec*Ie*Ij!Dj*
```

### Partida fallida:

```

Ea*Eb*Eg*Fb!Db! (Comprobar casilla Ea con carácter ?)
Ae*Aj! (Comprobar Ae* acción no válida, casilla Aj sigue sin marcar)
Eb* (Se produce la explosión)
```

## Apéndice B

# Caracteres para dibujar cuadros

```

COE = u'\u2500' # —
CNS = u'\u2502' # |
CES = u'\u250C' # ┌
CSO = u'\u2510' # ┐
CNE = u'\u2514' # └
CON = u'\u2518' # ┘
COES = u'\u252C' # ┐
CNES = u'\u251C' # ┌
CONS = u'\u2524' # ┌
CONE = u'\u2534' # └
CSOM = u'\u2593' # ▯
```