

Paradigmas de Programación

Práctica I - Curso 2019/20

El tercer clon (I)

1. Objetivo de la Práctica

En 2014 se creó un juego indie de tipo puzle para móviles, *Threes!*, que tuvo mucho éxito por lo que al poco tiempo surgió un clon suyo, *1024*, y poco después se desarrolló un clon de este último, *2048*, que se hizo famoso a nivel mundial y del que se crearon múltiples variantes.

El objetivo de la práctica es la creación de un nuevo clon de *2048*, en el que vamos a introducir los siguientes cambios:

- Poder tener tableros de cualquier tamaño (siempre cuadrados), no solo 4x4
- La posibilidad de que existan celdas que actúen como obstáculos permanentes
- El tener más opciones al representar las celdas (letras, nivel, 1024 y 2048)



La dinámica y apariencia del juego van a ser muy parecidas a las del juego *2048* y sus antecesores: El juego presenta una cuadrícula cuadrada de $n \times n$ celdas que o bien son **obstáculos** (no se mueven ni cambian en toda la partida) o bien están **vacías** o bien contienen un **bloque**, al que se le asocia un **nivel** (un entero positivo en el rango 1..12).

En el transcurso de la partida los bloques se pueden desplazar en línea recta pasando por celdas vacías y se van a poder **fusionar** incrementando su nivel, de acuerdo a unas reglas que detallaremos posteriormente.

En las celdas se va a mostrar información que indique si son un obstáculo, si están vacías o en el caso de que contengan un bloque, la información según su nivel de acuerdo a alguno de los siguientes modos:

- Modo alfabético: Se muestra una letra (nivel 1 \rightarrow A, nivel 2 \rightarrow B, etc.)
- Modo nivel: Se muestra directamente el número del nivel (máximo 2 dígitos)
- Modo 1024: Se muestra el valor $2^{\text{nivel}-1}$ (máximo 4 dígitos)
- Modo 2048: Se muestra el valor 2^{nivel} (máximo 4 dígitos)

A lo largo de la partida el usuario realiza jugadas indicando **movimientos** (izquierda, derecha, arriba o abajo) que afectan a la totalidad de los bloques del tablero: Todos los bloques se desplazan lo más posible en la dirección del movimiento indicado. Los desplazamientos son siempre en línea recta y atravesando casillas vacías (un bloque no puede *saltar* obstáculos ni otros bloques).

Si al desplazarse un bloque se topa con otro bloque **de su mismo nivel** entonces ambos bloques se **fusionan** en un único bloque de nivel uno más que los originales. En una misma jugada cada bloque puede sufrir una fusión como máximo.

Tras la realización de cada jugada (movimiento de los bloques), se crea un **nuevo bloque** en una de las celdas vacías (elegida al azar). El nivel de este nuevo bloque se elige al azar entre los valores 1 (con probabilidad 75%) y 2 (probabilidad 25%). Si no existieran celdas vacías en las que insertar el nuevo bloque **la partida finaliza**.

La puntuación de cada partida consiste simplemente en acumular, para cada fusión que se haya producido, el nivel del bloque resultante.

El objetivo de la práctica será crear una aplicación Python que implemente este juego de acuerdo a la descripción detallada que se proporciona a continuación. Entre las características que se incluirán estarán la posibilidad de elegir el tamaño y número de obstáculos del tablero, el poder cambiar el modo de presentación de los bloques y el poder leer y salvar tableros a fichero.

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

2. Descripción técnica

2.1 Estructura de la aplicación

La aplicación comienza mostrando un menú similar a éste:

```
----- CLON-3 -----
- Práctica de Paradigmas de Programación 2019-20 -
-----

1. CREAR NUEVO TABLERO
2. LEER TABLERO DE FICHERO
3. SALIR

Indique opción: _
```

Este menú permite escoger la forma en que se va a inicializar el tablero de juego. Si se escoge la primera opción se pide al usuario que indique el tamaño del tablero y el número de obstáculos, los cuales se asignarán al azar: (se muestran en azul los datos que introduce el usuario)

```
1. CREAR NUEVO TABLERO
2. LEER TABLERO DE FICHERO
3. SALIR

Indique opción: 1
Tamaño del tablero: 3
Numero de obstáculos: 2

+---+---+
|*| | |
+---+---+
| | | |
+---+---+
|*| | |
+---+---+

MOVIMIENTOS = 0 | PUNTUACIÓN = 0
Pulse [Enter] para mostrar inserción del nuevo bloque
```

Si se escoge la segunda opción el programa solicitará la ruta del fichero y leerá el tablero. En este caso (lectura del tablero) no se inserta un nuevo bloque sino que directamente se pasa a pedir la acción a realizar:

1. CREAR NUEVO TABLERO
2. LEER TABLERO DE FICHERO
3. SALIR

Indique opción: **2**

Nombre del Fichero: **ej1.tab**

```

+---+---+
| | | |A|
+---+---+
|B|B| | |
+---+---+
| |*| |B|
+---+---+
|B| | |C|
+---+---+

```

MOVIMIENTOS = 3 | PUNTUACIÓN = 17

[S]ubir, [B]ajar, [I]zda, [D]cha | [M]odo, [G]uardar, [F]in: _

En ambos casos, tras generar el tablero inicial, la aplicación entra en el **bucle de juego**, en el cual se solicita al usuario que introduzca una letra que indique la acción a realizar:

- [S],[B],[I],[D]: El usuario pide que se ejecute la jugada consistente en ese movimiento. Para facilitar la depuración del programa el resultado del movimiento se muestra en dos pasos, primero se muestra el movimiento y a continuación (tras pulsar enter) el nuevo bloque que se ha creado (se sigue el ejemplo anterior):

```

[S]ubir, [B]ajar, [I]zda, [D]cha | [M]odo, [G]uardar, [F]in: B
+---+---+
| | | | |
+---+---+
| |B| |A|
+---+---+
| |*| |B|
+---+---+
|C| | |C|
+---+---+

MOVIMIENTOS = 4 | PUNTUACIÓN = 20
Pulse [Enter] para mostrar inserción del nuevo bloque
+---+---+
| | |A| |
+---+---+
| |B| |A|
+---+---+
| |*| |B|
+---+---+
|C| | |C|
+---+---+

[S]ubir, [B]ajar, [I]zda, [D]cha | [M]odo, [G]uardar, [F]in:_

```

Nuevo bloque
colocado en una
posición vacía
elegida al azar

- [G]: Esta acción hace que el programa solicite al usuario el nombre del fichero en el que se va a escribir el estado actual de la partida, tras lo cual la partida continua en el punto donde se encontraba.
- [M]: Esta acción permite cambiar el modo de representación de las celdas. Se presenta al usuario un menú con las 4 opciones:

```
[S]ubir, [B]ajar, [I]zda, [D]cha | [M]odo, [G]uardar, [F]in: M
```

ESCOJA MODO VISUALIZACION:

1. Alfabeto
2. Nivel
3. 1024
4. 2048

Escoja opción: _

A partir de ese momento el tablero se presentará según la opción elegida, en la siguiente tabla se puede ver el aspecto del mismo tablero según los distintos modos:

<pre> +---+---+ +---+---+ B A +---+---+ * A B +---+---+ C C +---+---+ </pre>	<pre> +---+---+---+ +---+---+---+ 2 1 +---+---+---+ ** 1 2 +---+---+---+ 3 3 +---+---+---+ </pre>	<pre> +---+---+---+---+ +---+---+---+---+ 2 1 +---+---+---+---+ **** 1 2 +---+---+---+---+ 4 4 +---+---+---+---+ </pre>	<pre> +---+---+---+---+ +---+---+---+---+ 4 2 +---+---+---+---+ **** 2 4 +---+---+---+---+ 8 8 +---+---+---+---+ </pre>
Alfabeto	Nivel	1024	2048

- [F]: Esta opción finaliza la partida actual y hace que el programa vuelva a mostrar el menú principal.

2.2 Movimientos

En lo que sigue vamos a suponer que el movimiento elegido es **derecha**. Todo lo que se va a indicar es aplicable a cualquier otro tipo de movimiento, lo único que cambia entre ellos es si afectan a filas (subir, bajar) o columnas (izquierda, derecha) y el orden en que se actualizan los elementos. Se recomienda al alumno que en vez de crear código diferente para cada tipo de movimiento explore la posibilidad de escribir un código único que trate cada fila/columna, extraída de la forma adecuada del tablero según el movimiento.

Si el movimiento es **derecha**, significa que debemos tratar las filas. Supongamos una fila con el siguiente contenido (la celda oscura es un obstáculo):

0	1	2	3	4	5	6	7	8	9	10	11	12
D		C	C		B	B		B	A		A	

Para procesar la fila se debe recorrer *en orden contrario* al movimiento, en este caso (derecha) vamos hacia atrás desde el índice 12 hasta el 0. Si el movimiento fuera izquierda lo recorreríamos desde el 0 hasta el 12.

Solo hay que hacer algo en las celdas que contienen bloques. La idea general es que cada bloque se desplaza en la dirección de movimiento hasta que encuentre otro bloque o un obstáculo. Si es otro bloque hay que examinar si procede fusionar los bloques o no.

Vamos a mostrar como procesaría el algoritmo la fila del ejemplo, marcaremos en azul el índice de la celda por la que pasamos:

0	1	2	3	4	5	6	7	8	9	10	11	12
D		C	C		B	B		B	A		A	

0	1	2	3	4	5	6	7	8	9	10	11	12
D		C	C		B	B		B	A			A

En este caso como el bloque acaba situado al lado de un bloque del mismo nivel ambos se fusionan:

0	1	2	3	4	5	6	7	8	9	10	11	12
D		C	C		B	B		B			A	A

0	1	2	3	4	5	6	7	8	9	10	11	12
D		C	C		B	B		B				B

0	1	2	3	4	5	6	7	8	9	10	11	12
D		C	C		B	B		B				B

Como antes, el bloque acaba situado al lado de un bloque del mismo nivel, pero ahora no se fusionan porque el bloque de la celda 12 **es producto de una fusión**. La norma dice que cada bloque solo puede sufrir una fusión en cada movimiento.

0	1	2	3	4	5	6	7	8	9	10	11	12
D		C	C		B	B					B	B

Nuevamente el bloque acaba situado al lado de un bloque del mismo nivel. Como ese bloque no ha sufrido ninguna fusión, ahora ambos si que se pueden fusionar.

0	1	2	3	4	5	6	7	8	9	10	11	12
D		C	C		B						C	B

0	1	2	3	4	5	6	7	8	9	10	11	12
D		C	C							B	C	B

0	1	2	3	4	5	6	7	8	9	10	11	12
D		C	C							B	C	B

Aunque no se puede mover, como la celda de la derecha es de su mismo nivel se fusiona con ella

0	1	2	3	4	5	6	7	8	9	10	11	12
D			D							B	C	B

Por último, aunque el bloque acaba al lado de uno del mismo nivel, como es producto de una fusión en este mismo movimiento no se pueden fusionar.

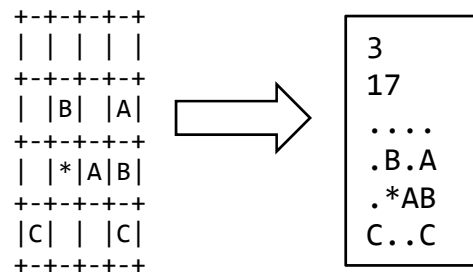
0	1	2	3	4	5	6	7	8	9	10	11	12
		D	D							B	C	B

La puntuación de la partida se incrementa cuando se producen fusiones. Como en este movimiento se han producido 3 fusiones dando como resultado un bloque de nivel 2 (B), uno de nivel 3 (C) y otro de nivel 4 (D) la puntuación se incrementaría en 9 puntos.

2.3 Formato de los ficheros

El estado de un tablero se almacena en un fichero de texto, las dos primeras líneas indican el número de movimientos realizados y la puntuación, y en las siguientes se muestra el contenido del tablero de la siguiente forma: Cada fila del tablero se convierte en una línea de texto con un carácter por cada celda: Un punto '.' si la celda está vacía, un asterisco '*' si es un obstáculo y la letra correspondiente del modo alfabético si es un bloque.

En el siguiente ejemplo se supone que se han realizado 3 movimientos y la puntuación es 17:



2.4 Validación de las entradas

Para aligerar el código en esta práctica se puede suponer que el formato y contenido de los ficheros que almacenan tableros son correctos y que las entradas procedentes del usuario son siempre válidas, no es necesario escribir código para su validación.

Tampoco es necesario verificar si en el transcurso de una partida se obtiene un bloque con un nivel superior al máximo, se puede suponer que eso nunca va a suceder.

3. 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 fichero(s) que componen la práctica (ficheros *.py). Para ello se habilitará en el Aula Virtual de la E.T.S. Informática (www.inf.uva.es -> menú Aula Virtual) una tarea de subida de ficheros cuya fecha límite será el **domingo 22 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 23 al 27 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 las técnicas impartidas hasta ese momento en la asignatura tiene una influencia positiva en la evaluación.