

Paradigmas de Programación

Práctica I - Curso 2016/17

¡Desbloquéame! (I)

Objetivo de la Práctica

El objetivo de la práctica es el desarrollo de una aplicación en lenguaje Python que implemente el famoso juego "¡Desbloquéame!" en un entorno de texto (no gráfico).



El objetivo de éste juego es mover los coches estacionados en un parking de forma que nuestro objetivo (el coche rojo en la imagen) pueda salir de él. Se considera que el parking forma una matriz de 6x6 celdas, los coches tienen una longitud de 2, 3 o 4 celdas y sólo se pueden mover hacia delante o hacia atrás un número discreto de celdas.

Los coches estacionados horizontalmente sólo se pueden mover a izquierda o derecha (siempre en la misma fila) y los estacionados verticalmente sólo se pueden mover arriba o abajo (siempre en la misma columna).

La salida se encuentra siempre en la tercera fila y la columna más a la derecha.

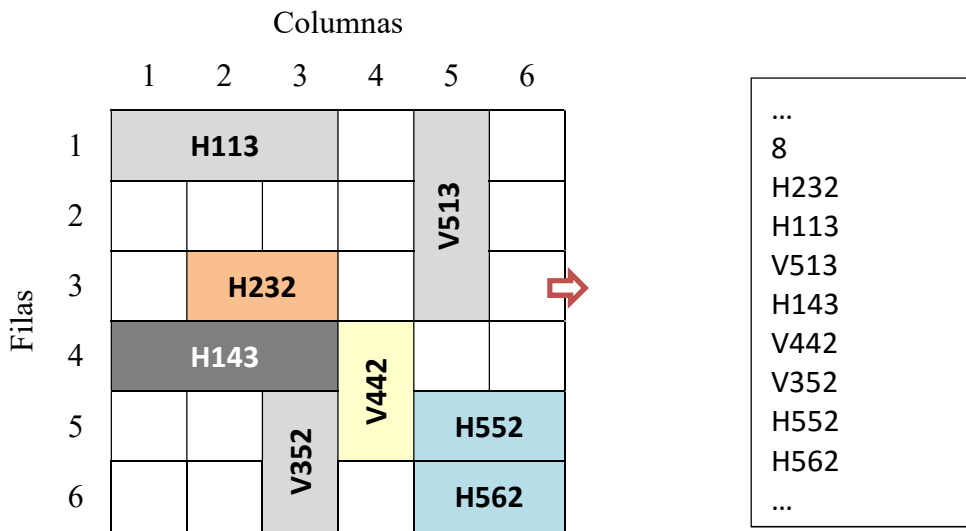
Descripción detallada

La aplicación trabajará con dos ficheros de texto: "niveles.txt" contiene la descripción de todos los niveles que se pueden jugar (un tablero por cada nivel), y "records.txt" contiene, para cada nivel jugado por algún usuario, el número mínimo de movimientos que se han necesitado para completarlo. El fichero "niveles.txt" no se modifica tras ejecutar la aplicación y su formato está prefijado (se explica a continuación), pero el fichero "records.txt" sí que se actualiza tras cada ejecución y podéis definir su formato como más os convenga.

Al comenzar la ejecución la aplicación leerá ambos ficheros y preguntará al usuario el nivel en el que quiere comenzar, pero sólo permite elegir hasta uno más que el máximo nivel en que existan records (si hay records para los niveles 1..4, permite elegir un nivel entre 1 y 5).

El formato del fichero "niveles.txt" es el siguiente: En la primera línea existe un entero con el número total de niveles existentes. Las siguientes líneas describen los niveles, donde cada nivel consta de una línea indicando el número de coches a la que siguen una línea de 4 caracteres por cada coche, que indica su orientación (H → Horizontal, V → Vertical), la columna y fila de su extremo superior izquierda (un dígito entre 1 y 6 cada valor) y su longitud (un dígito con el valor 2, 3 ó 4).

En el siguiente gráfico se puede observar cómo sería la descripción del nivel mostrado en el apartado anterior (en cada coche se muestra su código descriptivo):



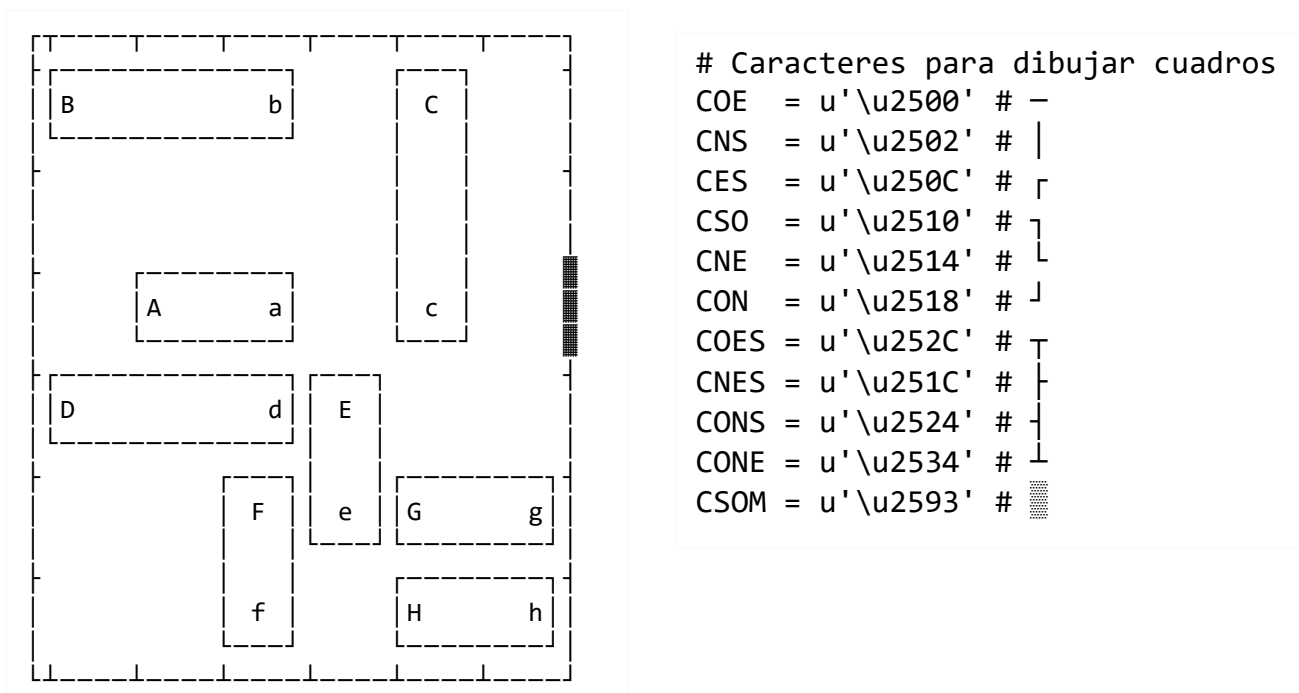
El coche objetivo siempre es el **primero** que aparece en la lista, y se considera que el nivel se ha completado cuando su extremo derecho está en la columna 7, fila 3 (flecha roja).

La aplicación comenzará a jugar el nivel indicado, ejecutando un bucle en el que mostrará el estado del tablero, pedirá al usuario que indique uno o varios movimientos y los efectuará mostrando el resultado hasta que el nivel se haya completado. Tras completar el nivel, preguntará al usuario si quiere pasar al siguiente nivel. Si el usuario responde que no, terminará la ejecución actualizando el fichero de records.

Para facilitar que el usuario pueda indicar que coche quiere mover y en qué sentido, se adoptará la siguiente estrategia:

- Se identificará a cada coche por una letra 'A', 'B', 'C', ...
- Al mostrar los coches en pantalla, se escribirá esa letra en mayúsculas en el extremo superior/izquierda y en minúsculas en el extremo inferior/derecho
- Si el usuario introduce una letra mayúscula significa que quiere que ese coche **retroceda** una casilla y si introduce una letra minúscula significa que quiere que **avance** una casilla

A continuación se muestra un ejemplo de cómo debe ser la representación de un tablero:



En el recuadro de la derecha se muestra la manera de especificar en Python los caracteres Unicode que permiten "dibujar" un tablero con el formato que se desea en ésta práctica.

El usuario podrá indicar varios movimientos en cada jugada y la aplicación les aplicará en secuencia (sólo es necesario que muestre el resultado final, no los tableros intermedios). Si alguno de los movimientos es incorrecto, ya no ejecutará el resto de movimientos de la cadena. A la hora de contar sólo se contarán los movimientos correctos.

Ejemplo de ejecución:

```

Elija nivel (1-3): 1
NIVEL 1 - RECORD 40 movimientos

```

```

Movimientos = EEddAFFDDGGGGHHHHBcc
Movimiento B imposible por bloqueo

```

```

Movimientos = ccceebbbFaaaa

```

```

ENHORABUENA, HA COMPLETADO EL NIVEL!
Movimientos: 35 (NUEVO RECORD!)
Desea pasar al siguiente nivel? [S/N]_

```

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 -> menú Aula Virtual) una tarea de subida de ficheros cuya fecha límite será el **domingo 26 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 27 al 31 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.