

# Interfaz Sokoban

Andres Contreras, Alejandro Torres, Michael Rincon, Santiago Salamanca

9 de Noviembre del 2018

## Documentación

A continuación se presenta el diseño de la entrega final del tablero de Sokoban, con las clases, archivos, métodos, constantes y todo lo necesario para su correcto funcionamiento:

### Archivos

- **manual.py** : Este archivo contiene el menu de inicio y los llamados iniciales para el uso de la interfaz, mediante el cual se puede seleccionar a través de este, el tipo de jugador que resolverá el tablero, uno manual o uno automático. Dentro de este se instancia la interfaz y se llaman los diferentes métodos que la componen.
- **Sokoban.py**: Este archivo representa la interfaz donde se encuentran las funcionalidades correspondientes para el inicio del tablero, los movimientos del jugador en Sokoban y la validación para las restricciones del juego. Más adelante se explicará cómo funciona cada una de ellas.
- **const.py**: En este archivo se encuentran las constantes que serán utilizadas en los archivos explicados anteriormente. Estas constantes representan los caracteres del tablero de Sokoban. Posteriormente, se explicará cómo se representa cada una de ellas.

### Clases

Se usará una clase llamada Board para crear nuestros propios tipos de datos, ya que las clases proveen una forma de empaquetar datos y funcionalidad juntos. Al crear una nueva clase, se crea un nuevo tipo de objeto, permitiendo crear nuevas instancias de ese tipo. Cada instancia de clase puede tener atributos adjuntos para mantener su estado.

### Métodos del archivo manual.py

- **jugadorManual()**:
  - Parámetros: No tiene ningún parámetro de entrada
  - Retorno: No tiene ningún retorno.
  - Descripción: Este método utiliza la interfaz de Sokoban para inicializar un tablero por defecto, y espera los comandos del usuario para mandárselos a la interfaz y que esta muestre el resultado generado, y seguirá haciendo esto mientras que el jugador no gane o pierda.
- **jugadorAutomatico( )**:
  - Parámetros: No tiene ningún parámetro de entrada.
  - Retorno: No tiene ningún retorno.

- Descripción: Este método utiliza la interfaz de Sokoban para inicializar un tablero por defecto, para después ejecutar un método que tiene la funcionalidad de un jugador automatico.

## Principales Métodos del archivo Sokoban.py

### ■ `__init__ ( )`:

- Parámetros: Variable arch, que representa el nombre del archivo.
- Retorno: No tiene ningún retorno.
- Descripción: Este método inicializa el tablero de Sokoban , dado un archivo que llega como parámetro, el cual contiene el ancho, el alto y el tablero representado en caracteres ASCII.

### ■ `Print ( )`:

- Parámetros: No tiene ningún parámetro de entrada .
- Retorno: No tiene ningún retorno.
- Descripción: Imprime el tablero de Sokoban.

### ■ `movimientos ( )`:

- Parámetros: Cadena de caracteres que representan los movimientos que realizara el jugador. Está cadena puede representar tanto un movimiento como varios.
- Retorno: Retorna las coordenadas del jugador dentro del tablero.
- Descripción: Este método examina la cadena de caracteres, y para cada caracter llama a la función mover.

### ■ `estadoJugador ( )`:

- Parámetros: No tiene ningún parámetro de entrada.
- Retorno: Retorna si el jugador puede seguir jugando o no.
- Descripción: Este método verifica si el usuario puede seguir jugando, ya sea porque gano o porque alguna de las cajas se encuentra bloqueada.

## Constantes

Las constantes utilizadas para representar las diferentes características del tablero son las siguientes:

Constante	Valor
JUGADOR	@
CAJA	\$
META	M
CAJAM	*
MURO	X
LIBRE	—
JUGADORM	&

Figura 1: Tabla de constantes

## Instrucciones para el uso de la interfaz

Para las llamadas a la interfaz, hay que hacer uso del archivo `manual.py`, ya que este se ocupa de mandarle los datos necesario para el correcto inicio del tablero con los valores en sus respectivas casillas dependiendo del nivel que vaya a ser cargado.

Para iniciar, hay que abrir la terminal y dirigirse a la ruta donde se encuentre el proyecto, una vez ahí, se debe ejecutar el siguiente comando:

- `python ./manual.py`

Una vez ejecutado el comando, nos aparecerá el menú para escoger entre un jugador manual o un jugador aleatorio, opción libre para el usuario y deberá digitar el “1” o el “2” para llamar a alguno de los métodos para comenzar el juego.

En caso de que el usuario haya seleccionado “1”:

- Se mostrará información detallada del tablero tal como su ancho y alto, además de la posición inicial del usuario y el tablero inicial.

- El sistema esperará hasta que el usuario digite un comando y le de enter. Para que se refleje el cambio de posición del jugador en el tablero, los comandos deben ser los que ya fueron explicados en el diseño.
- El sistema va a seguir pidiendo entradas al programa hasta que el jugador gane o pierda.

Si fue digitado el “2”:

- Se mostrará cada cierto tiempo los movimientos generados aleatoriamente en el tablero.
- Cuando gane o pierda, se acaba el juego.

## Implementar la interfaz al proyecto

Para implementar la interfaz de Sokoban y que funcione correctamente a cada uno de los proyectos que la van a utilizar, por favor seguir los siguientes pasos:

- Importar el archivo de la interfaz (`import Sokoban`).
- Colocar en una variable el path del archivo en donde se encuentra el tablero con el que deseen probar.
- Inicializar el tablero de Sokoban llamando el constructor `Board()` y entregarle el path del tablero como parámetro de entrada. Implementar la heurística propuesta por cada grupo.
- Utilizar la variable con la que se inicializó el tablero para utilizar sus métodos respectivos, que son los siguientes:
  - `movimientos()`
  - `jugadorAutomatico()`
  - `estadoJugador()`
  - `Print()`