

# TP2

## Sokoban 2: *La venganza de Sokoban*

El objetivo de este TP es implementar el juego Sokoban, pero ahora *en serio*. Aprovecharemos la implementación de las reglas básicas hecha en el TP1, y le agregaremos una **Interfaz gráfica** (utilizando Gamelib) y **progresión de niveles**.

## Estructura del programa

Recomendamos tener al menos los siguientes archivos:

- `soko.py`: el mismo que hiciste para el TP1. Si hiciste todo bien lo más probable es que no necesites modificarlo.
- `gamelib.py`: el mismo que usamos para el EJ2.
- `main.py`: acá implementaremos las nuevas funcionalidades.
- `niveles.txt`: definición de los niveles disponibles.
- `teclas.txt`: configuración de teclas y acciones asociadas

Para el archivo `main.py` recomendamos comenzar con el siguiente esqueleto:

```
import soko
import gamelib

def main():
    # Inicializar el estado del juego

    gamelib.resize(300, 300)

    while gamelib.is_alive():
        gamelib.draw_begin()
        # Dibujar la pantalla
        gamelib.draw_end()

        ev = gamelib.wait(gamelib.EventType.KeyPress)
        if not ev:
            break

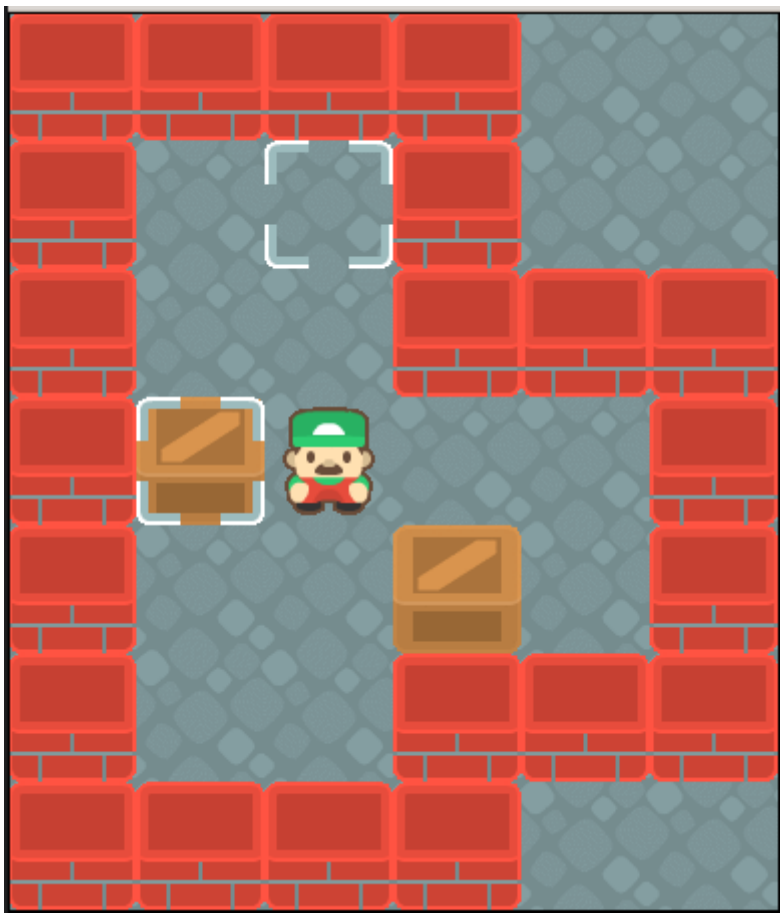
        tecla = ev.key
        # Actualizar el estado del juego, según la `tecla` presionada

gamelib.init(main)
```

## Interfaz gráfica

Al igual que en el EJ2, no hay un requerimiento fijo en cuanto a cómo tiene que lucir el juego: lo importante es que el jugador pueda entender fácilmente dónde está el personaje, las cajas, los objetivos, las paredes, etc. Mientras cumpla eso, la representación gráfica puede ser tan simple o compleja como quieras.

Dejamos a tu disposición un conjunto de imágenes que pueden servir para que el juego se vea de esta manera:



### Pack de imágenes

Si decidís usar estas imágenes, descomprimí el archivo y ubicá la carpeta `img` en el lugar donde está tu programa. Luego, para dibujar una de las imágenes en la ventana (por ejemplo `player.gif`):

```
gamelib.draw_image('img/player.gif', x, y)
```

El pack de imágenes fue obtenido de [esta fuente](#). Si querés podés utilizar otras imágenes o crear las tuyas propias.

## Controles

El jugador debe poder controlar el juego mediante el teclado, y la configuración de asignación de teclas a sus respectivas acciones se tomará de un archivo `teclas.txt`.

Cada línea del archivo tiene el formato `<tecla> = <acción>`, donde:

- `<tecla>` representa a una tecla del teclado, en el mismo formato que utiliza Gamelib (la variable `tecla` del esqueleto)
- `<acción>` es una de las acciones posibles en el juego:
  - `NORTE` : mover hacia arriba
  - `SUR` : mover hacia abajo
  - `ESTE` : mover hacia la derecha
  - `OESTE` : mover hacia la izquierda
  - `REINICIAR` : reiniciar el nivel actual
  - `SALIR` : salir del juego

Una acción puede estar asignada a más de una tecla.

El archivo además puede tener líneas vacías, que deben ser ignoradas.

Ejemplo del archivo `teclas.txt`:

```
Up = NORTE
Left = OESTE
Down = SUR
Right = ESTE

w = NORTE
a = OESTE
s = SUR
d = ESTE

r = REINICIAR

Escape = SALIR
```

## Niveles

El juego debe cargar la definición de los niveles del juego leyendo el archivo `niveles.txt`. Descargá el archivo mediante el siguiente link:

[niveles.txt](#)

El archivo contiene 155 niveles y fue obtenido de este [sitio web](#). El sitio contiene más sets de niveles que podés utilizar si querés.

El juego debe leer el archivo al iniciar y cargar todos los niveles en la memoria del programa.

Dentro del archivo, los niveles están separados por una línea vacía. Cada nivel tiene un formato como el siguiente:

```
Level 148
'from (Original 18)'
#####
# . # #
# #. @ #
# #.# #####
## ## $ $ $ $ #
##           #
#####
```

- La primera línea indica el número de nivel.
- La segunda línea es **opcional** (puede no estar), e indica el título del nivel entre comillas simples.
- Las líneas siguientes contienen la descripción del nivel como se especificó en el TP1.

## Entrega

La entrega del trabajo (a través del [formulario de entregas](#)) debe incluir todos los archivos necesarios para ejecutar el juego comprimidos en un archivo `zip`.