

Un poco de práctica: OpenAI Gym

Esta entrega consiste en generar un laberinto con un robot en Python, y mover el robot desde el origen hasta la meta.

Con esta práctica aprenderás

- A utilizar un entorno OpenAI Gym que carga un mapa directamente de un fichero .csv (comma-separated values).
- A desarrollar un script Python que mueva tu robot desde el origen hasta la meta utilizando OpenAI Gym.
- Muchos comandos de consola habituales en el uso de sistemas operativos GNU/Linux. 😊

Procedimiento

1. Calcula las dimensiones de tu mapa [m]:

```
python
max_x = len("sustituyeEstoPorTuPrimerApellido") * 2
max_y = len("sustituyeEstoPorTuSegundoApellido") * 2
print(max_x, max_y)
```

(puedes salir de Python vía ctrl+d)

2. Escoge tu resolución [carácter/metro]. A mayor resolución, mayor nivel de detalle. Decide este parámetro para tu mapa. Se recomienda empezar por una resolución 1 (utilizar 2 o 3 para una mayor resolución).

3. Descarga el fichero comprimido: gym-csv.zip. Descomprímelo:

```
unzip gym-csv.zip -d $HOME/
```

4. Instala el entorno gym-csv. Esto sólo es necesario una vez (si haces cambios, se actualiza el comportamiento):

```
cd # Va a $HOME
sudo -H pip install -e gym-csv # Puede ser via pip2 o pip3 dependiendo de tu
config
```

5. Prueba los diferentes ejemplos de gym-csv/examples:

```
cd $HOME/gym-csv/examples
python ./gym-csv-display.py
python ./gym-csv-loop.py
```

6. Edita `$HOME/gym-csv/examples/map1.csv` con cualquier editor (p.ej. `libreoffice $HOME/gym-csv/examples/map1.csv` (recomendado) o `gedit $HOME/gym-csv/examples/map1.csv`) para ajustarlo a tus dimensiones y resolución (p.ej. para una resolución 2 tu mapa deberá tener el doble de filas y columnas que lo computado en metros), atendiendo a los siguientes 2 criterios:
 - i. Las paredes exteriores deben tener de grosor medio de 1 carácter en `map1.csv`.
 - ii. El origen se situará próximo al origen de coordenadas (p.ej. para una resolución 1 y empezando a contar desde 1: en fila 1, columna 1 habría pared, y por tanto el origen se situaría en fila 2, columna 2). La meta se sitúa en la esquina diagonal opuesta (maximizando filas y columnas sin solaparse con pared).
7. El "parser" del `gym-csv` interpreta el mapa pero no es capaz de conocer el origen ni la meta. Edita `$HOME/gym-csv/gym_csv/envs/csv_env.py`, específicamente las líneas:

```
# hard-coded vars (begin)
inFileStr = 'map1.csv'
initX = 2
initY = 2
goalX = 7
goalY = 2
# hard-coded vars (end)
```

8. Tu entorno está listo. Puedes utilizar `gym-csv-loop.py` como plantilla para desarrollar tu script Python con OpenAI Gym para mover tu robot desde su origen hasta la meta! 🍇🍇🍇
- Nota: `gym-csv-r1.py` se subió como ejemplo de integración con un algoritmo de aprendizaje por refuerzo (reinforcement learning), en principio a modo exclusivamente ilustrativo.
 - Nota 2: Puedes cargar, el entorno `csv-pygame-v0` en lugar de `csv-v0`, pero deberás editar los parámetros correspondientes de `$HOME/gym-csv/gym_csv/envs/csv_pygame_env.py`.