

Entrega práctica: Gymnasium

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

Con esta entrega práctica aprenderás

- A utilizar un entorno Gymnasium 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 Gymnasium.
- Comandos de consola habituales en distintos sistemas operativos. 😊

Procedimiento

1. Instalar [Python](#) y [pip](#) (en Windows, Python incluye pip).
2. Calcula las dimensiones de tu mapa [m] en una consola (Windows: Menú inicio > cmd)(Linux: ctrl+alt+t)(puedes salir de Python vía quit()):

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

3. Descarga el fichero comprimido de Aula Global: [gymnasium-csv.zip](#) y descomprímelo (botón derecho, Extraer todo...). Alternativa (solo Linux): a través de una consola (ctrl+alt+t): `unzip gymnasium-csv-026.zip -d $HOME/`
4. Instala el entorno gymnasium-csv. Esto sólo es necesario una vez (si haces cambios, se actualiza el comportamiento):

```
cd gymnasium-csv-026 # Entrar en la contiene README.md y setup.py
pip install -e . # Puede ser via pip2 o pip3, o `sudo -H` antes
```

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

```
cd gymnasium-csv-026/examples
python render.py
python render-text.py
python loop.py
python loop-wrapped.py
```

6. Edita (de lo descargado) `gymnasium-csv-026/examples/map1.csv` con cualquier editor de tablas (p.ej. Excel o Libreoffice) o texto plano (p.ej. Linux: `gedit assets/map1.csv`), manteniendo siempre el formato `.csv`. Ajústalo a tus dimensiones previamente calculadas y preferencias (cantidad y forma de obstáculos que quieras introducir) atendiendo a los siguientes 2 criterios **obligatorios**:
 - 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).
8. Tu entorno está listo. Puedes utilizar cualquiera de los ejemplos como plantilla para desarrollar tu script Python con Gymnasium para mover tu robot desde su origen hasta la meta! 🍌🍌🍌
- Nota: `q-learning.py` se subió como ejemplo de integración con un algoritmo de aprendizaje por refuerzo (reinforcement learning), en principio a modo exclusivamente ilustrativo. Hace falta ``pip install termios`` (solo Linux?) o comentar las líneas correspondientes aproximadamente L50-L60 que simplemente introducen una espera de tecla teclado.