

Manual de usuario

Requisitos básicos:

- Antes de ejecutar cualquier de los dos programas, se parte de la base que unicamente se posee instalado Python en cualquier de sus versiones 3, por lo que se necesita instalar todas las librerías necesarias para el correcto funcionamiento. El primer paso es ubicarse en la raíz del proyecto y ejecutar los siguientes comandos:
 - `python -m pip install --upgrade pip`
 - `pip install -r requirements.txt`
 - `pip freeze > requirements.txt`

De esta forma se instalaran todos los paquetes necesarios para que los programas funcionen correctamente.

CONNECT 4

Ejecución:

- Connect 4 Agente vs Humano, lo primero es definir las opciones de configuración para el juego:
 - `-gt` o `--game-type`: identifica el tipo de juego, en este caso para Agente vs Humano debe ser un 0.
 - `-p1c` o `--agentec`: identifica las características del agente (las estrategias y sus correspondientes valores), es un arreglo de la forma `[x,y,z,w]`.
- Connect 4 Agente vs Agente, lo primero es definir las opciones de configuración para el juego:
 - `-gt` o `--game-type`: identifica el tipo de juego, en este caso para Agente vs Agente debe ser un 1.
 - `-a1c` o `--agent1c`: identifica las características del agente1 (las estrategias y sus correspondientes valores), es un arreglo de la forma `[x,y,z,w]`.
 - `-a2c` o `--agent2c`: identifica las características del agente2 (las estrategias y sus correspondientes valores), es un arreglo de la forma `[x,y,z,w]`.

Ejemplos de ejecución

Agente vs Humano

- `python main_game.py -gt 0 -p1c "[0.8,0.6,0.3,0.2]"`
- `python main_game.py --game-type 0 --agentec "[0.3,0.5,0.5,0.7]"`

Agente vs Agente

- `python main_game.py -gt 1 -a1c "[0.8,0.6,0.5,0.3]" -a2c "[0.2,0.4,0.3,0.8]"`
- `python main_game.py --game-type 1 --agent1c "[0.8,0.6,0.5,0.3]" --agent2c "[0.2,0.4,0.3,0.8]"`

ALGORITMO GENÉTICO

Ejecución

- Algoritmo genético: lo primero es definir las opciones de configuración para la ejecución del mismo las cuales son:
 - -p o --population: identifica la población inicial del algoritmo.
 - -g o --generations: identifica el número de generaciones con las cuales se ejecutara el algoritmo.
 - -l o --limiter: identifica la cantidad de individuos que pasan de una generación a otra.
 - -m o --mutation: responde a la probabilidad de mutación de un individuo.

Ejemplos de Ejecución

- `python main_fit_agents.py -p 10 -g 5 -l 5 -m 0.10`
- `python main_fit_agents.py --population 10 --generations 5 --limiter 5 --mutation 0.10`