Documentazione

La seguente è una breve documentazione di classi e metodi del nostro progetto. L'obiettivo del progetto è quello di sviluppare un motore di inferenza che permetta di effettuare ragionamento deduttivo e abduttivo a partire da una base di conoscenza di clausole di Horn.

engine.py Le classi contenute in engine:

- Clause: rappresenta una clausola
- Predicate: rappresenta un predicato
- Variable: rappresenta una variabile
- Constant: rappresenta una costante
- Engine: presenta i seguenti metodi:
    - load_kb: carica una base di conoscenza da un file
    - how: restituisce l'insieme delle regole usate per provare l'obiettivo.
    - prove: dimostra la query in input, prende in input i seguenti flag:
      - prove_one: specifica se fermarsi dopo aver trovato una singola clausola di risposta che dimostra la query
      - abduce: specifica se effettuare diagnosi abduttiva.
      - occurs_check: specifica se effettuare il controllo di occorrenza durante l'unificazione di due espressioni.

  È presente, inoltre, la funzione parse che effettua il parsing di una espressione scritta nel linguaggio, e lancia ParseException se sono presenti errori sintattici

built-in.py contiene:
- predicati: restituiscono vero o falso
- funzioni: restituiscono un simbolo di costante

path_finding.py presenta la classe Graph che serve a modellare il grafo delle stanze dell'ambiente e le seguenti funzioni:
- euclidean_distance: funzione euristica che restituisce la distanza tra due posizioni
- A_star: restituisce il percorso migliore tra due posizioni

agent.py presenta la classe Agent con i seguenti metodi:

- act: prende in input la stringa di testo passata nella barra di input dell'applicazione e modella il comportamento dell'agente.
- check_conflicts: controlla la presenza di conflitti all'interno della kb e specifica in quale ordine controllare gli assumibili di un conflitto.

create_scene.py presenta una serie di funzioni per la creazione dell'ambiente grafico

gui.py presenta delle classi per la creazione dell'interfaccia

main.py main del progetto contenente la classe MyGame che estende la classe View della libreria arcade, che permette di creare giochi in 2D e simulazioni.

Cartella resource contiene sprite e texture di muri e oggetti.

Cartella test contiene vari test per il motore.