

PIANIFICAZIONE AUTOMATICA PER L'IRRIGAZIONE IN UNA SERRA

Obiettivo implementazione: Sviluppare un programma di automated planning che ha lo scopo di gestire l'irrigazione all'interno di una serra nell'arco di una settimana. Le diverse specie di piante presenti nella serra devono essere irrigate in modo tale da mantenere un livello di umidità indispensabile per la loro salute. La pianificazione automatica si propone di ridurre i consumi dell'acqua allo stretto necessario.

Descrizione del modello: Ogni singolo irrigatore è associato univocamente ad una pianta e la irriga rifornendosi da un serbatoio comune, il quale ha un livello massimo di capacità. Nel terreno sono posizionati sensori che misurano il livello di umidità della pianta classificandolo in terreno di alta, media o bassa umidità. Una pianta si trova in condizione di salute se il terreno possiede un livello di umidità medio-alta, viceversa se il terreno possiede un'umidità bassa è necessaria l'irrigazione. Dopo un certo numero di utilizzi gli irrigatori necessitano di manutenzione: durante questa fase non possono essere utilizzati. Ciascuna pianta possiede uno specifico fabbisogno di acqua, indicato in unità, che riduce la capacità d'acqua presente nel serbatoio. Nel progetto vengono seguite le seguenti specifiche:

- Il livello massimo del serbatoio è stato impostato a 4 unità d'acqua.
- L'irrigazione avviene solo la mattina: la quantità d'acqua necessaria per la singola pianta verrà definita nel problem.
- Viene simulato il processo di degradazione dell'umidità tramite apposite azioni
- La manutenzione degli irrigatori avviene dopo 1 utilizzo e deve essere eseguita tra un'irrigazione e la successiva nel pomeriggio. Solo quando l'azione "manutenzione-terminata" viene svolta, gli irrigatori ritornano in uso.
- Per garantire che la degradazione dell'umidità avvenga ogni giorno, è stata definita una funzione che conteggia il numero di degradazioni effettuate fino a quel momento.
- Non abbiamo fornito precondizioni a livello temporale nella ricarica del serbatoio.

DOMAIN

Il dominio definisce un mondo attraverso **types**, che nel nostro caso sono pianta e irrigatore.

I **predicati** descritti nel nostro mondo sono: (umidita-basso ?p - pianta, umidita-medio ?p - pianta, umidita-alto ?p - pianta) che definiscono lo stato di umidità effettiva delle piante, (in-uso ?i - irrigatore) per indicare lo stato funzionante dell'irrigatore, (manutenzione-terminata ?i - irrigatore) utilizzata per riattivare gli irrigatori a seguito della manutenzione, (collega-irrigatore-pianta ?p-pianta ?i - irrigatore) per associare un irrigatore alla rispettiva pianta e (degrado-completato ?p - pianta) utilizzato nelle azioni che gestiscono l'umidità per garantire il degrado giornaliero. Inoltre, vengono definiti i giorni e i momenti della giornata.

Le **funzioni** numeriche utilizzate sono invece: (quantita-acqua-pianta ?p - pianta) che permette di definire quanta acqua necessita una pianta, (livello-serbatoio) per rappresentare il serbatoio, (utilizzi-irrigatore ?i - irrigatore) conta quante volte un irrigatore è stato utilizzato (effettuato-degrado-giorno ?p) per garantire che il degrado avvenga tutti i giorni della settimana.

Infine, le **azioni** utilizzate sono:

Irrigazione: rappresenta l'irrigazione di una pianta utilizzando un irrigatore.

Parametri

- ?p : pianta da irrigare.
- ?i : irrigatore associato alla pianta.

Precondizioni

- L'irrigatore è collegato alla pianta (collega-irrigatore-pianta ?p ?i).
- La pianta ha un'umidità bassa (umidita-basso ?p).

- L'irrigatore è in uso (in-uso ?i).
- L'irrigatore ha meno di 1 utilizzo (<(utilizzi-irrigatore ?i) 1).
- Il livello del serbatoio è sufficiente per soddisfare il fabbisogno della pianta (>(livello-serbatoio) (quantità-acqua-pianta ?p)).
- È mattina (mattina-momento).

Effetti

- L'umidità della pianta passa a alta (umidità-alto ?p).
- L'umidità bassa e media vengono rimosse (not (umidità-basso ?p), not (umidità-medio ?p)).
- Il livello del serbatoio viene ridotto in base al fabbisogno della pianta (diminuzione (livello-serbatoio) (quantità-acqua-pianta ?p)).
- Il contatore degli utilizzi dell'irrigatore viene incrementato di 1 (increase (utilizzi-irrigatore ?i) 1).

Ricarica-serbatoio: rappresenta il riempimento completo del serbatoio d'acqua.

Precondizioni

- Il livello del serbatoio è inferiore o uguale a 1 (<=(livello-serbatoio) 1).

Effetti

- Il livello del serbatoio viene riportato al massimo (assign (livello-serbatoio) 4).

Eseguì-manutenzione: rappresenta l'inizio della manutenzione di un irrigatore.

Parametri

- ?i : irrigatore da sottoporre a manutenzione.

Precondizioni

- L'irrigatore ha raggiunto 1 utilizzo (= (utilizzi-irrigatore ?i) 1).
- È pomeriggio (pomeriggio-momento).

Effetti

- Il contatore degli utilizzi dell'irrigatore viene azzerato (assign (utilizzi-irrigatore ?i) 0).
- L'irrigatore non è più in uso (not (in-uso ?i)).
- La manutenzione dell'irrigatore è terminata (manutenzione-terminata ?i).

Manutenzione-finita: rappresenta il completamento della manutenzione di un irrigatore.

Parametri

- ?i : irrigatore che ha completato la manutenzione.

Precondizioni

- La manutenzione dell'irrigatore è terminata (manutenzione-terminata ?i).

Effetti

- L'irrigatore torna in uso (in-uso ?i).

Avanzamento giorni: rappresenta il passaggio da un giorno al successivo. Ogni azione specifica il giorno corrente e il momento della giornata. E' mostrato un esempio con martedì mattina.

Precondizioni

- È lunedì sera (lunedì-giorno, sera-momento).

Effetti

- Rimuove lo stato di lunedì (not (lunedì-giorno)).
- Rimuove lo stato di sera (not (sera-momento)).
- Imposta il giorno a martedì (martedì-giorno).
- Imposta il momento a mattina (mattina-momento).

Avanzamento momenti della giornata: rappresenta il passaggio tra i momenti della giornata: mattina, pomeriggio e sera. E' riportato un esempio con il giorno martedì.

Precondizioni

- È mattina (mattina-momento).

Effetti

- Rimuove lo stato di mattina (not (mattina-momento)).
- Imposta il momento a pomeriggio (pomeriggio-momento).

Umidità media: rappresenta il degrado dell'umidità di una pianta da alta a media.

Parametri

- **?p** : La pianta il cui livello di umidità deve essere degradata.

Precondizioni

- Il degrado non è stato completato (not (degrado-completato ?p)).
- È sera (sera-momento).
- La pianta ha un'umidità alta (umidita-alto ?p).

Effetti

- L'umidità della pianta passa a media (umidita-medio ?p).
- L'umidità alta viene rimossa (not (umidita-alto ?p)).
- Il degrado viene completato (degrado-completato ?p).
- Incrementa il contatore del degrado giornaliero (incremento (effettuato-degrado-giorno ?p) 1).

→ In modo analogo viene gestita l'azione umidità bassa

Resetta-umidità: rappresenta il reset dello stato di degrado di una pianta.

Parametri

- **?p** : La pianta il cui stato di degrado deve essere resettato.

Precondizioni

- È pomeriggio (pomeriggio-momento).

Effetti

- Rimuove lo stato di degrado completato (not (degrado-completato ?p)).

PROBLEMS

La nostra scelta progettuale è stata quella di testare il domain attraverso tre istanze di problem generici differenti, che utilizzano un quantitativo diverso di piante (una, due o tre) e un livello di umidità iniziale di umidità diverso (alto o basso). Il livello di umidità medio viene utilizzato esclusivamente per rendere il progetto più realistico simulando una reale degradazione di umidità delle piante.

È composto da una prima parte di dichiarazione degli objects, l'inizializzazione, in cui viene indicato lo stato iniziale del mondo specifico e da una dichiarazione dei goal attesi.

Il goal è raggiunto quando ciascuna pianta ha umidità alta, sono state eseguite 6 degradazioni dell'umidità nel corso dei 7 giorni e si raggiunge domenica sera. Al termine del plan il serbatoio non deve necessariamente essere ricaricato così come la manutenzione degli irrigatori.

I problemi realizzati sono 3 per verificare il funzionamento del programma con più piante (e irrigatori) in condizioni diverse.

PROBLEM 1: un irrigatore - una pianta.

- **Dichiarazione objects**: vengono dichiarati gli elementi lattuga (oggetto pianta) e irrigatore lattuga (oggetto irrigatore), collegato alla pianta lattuga.
- **Inizializzazione**: viene indicato lo stato iniziale:
 - La pianta lattuga è collegata all'irrigatore irrigatore-lattuga.
 - È lunedì mattina.

- La pianta ha umidità bassa (deve pertanto essere irrigata).
- Serve 1 unità d'acqua per irrigare la lattuga.
- L'irrigatore non è ancora stato utilizzato.
- Il serbatoio contiene 4 unità d'acqua.
- L'irrigatore è in uso.
- La pianta non ha ancora subito degrado giornaliero.

PROBLEM 2: due irrigatori - due piante ciascuna a livelli di umidità diversa ma con stesso quantitativo d'acqua richiesto.

- **Dichiarazione objects:** sono stati aggiunti gli elementi pomodori (oggetto pianta) e irrigatore pomodori (oggetto irrigatore), collegato alla pianta pomodori.
- **Inizializzazione:** viene indicato lo stato iniziale:
 - Entrambe le piante sono collegate ai rispettivi irrigatori.
 - È lunedì mattina.
 - La lattuga inizia con umidità bassa (deve essere irrigata) mentre i pomodori partono con umidità alta.
 - Ora ogni pianta richiede 3 unità d'acqua per irrigazione.
 - Gli irrigatori non sono ancora stati utilizzati.
 - Il serbatoio contiene 4 unità d'acqua.
 - Gli irrigatori sono in uso.
 - Non si è ancora verificata alcuna degradazione.

PROBLEM 3: tre irrigatori e tre piante a livelli di umidità diversa e con diverso quantitativo d'acqua richiesto.

- **Dichiarazione objects:** si ha ulteriore aggiunta degli elementi carote (type pianta) e irrigatore carote (type irrigatore), collegato alla pianta carote.
- **Inizializzazione:** viene indicato lo stato iniziale:
 - Ogni pianta è collegata al rispettivo irrigatore.
 - È lunedì mattina.
 - La lattuga ha umidità bassa così come i pomodori (devono essere irrigate) mentre le carote partono con umidità alta.
 - La lattuga chiede un'unità di acqua per l'irrigazione, i pomodori 2 e le carote 3. Gli irrigatori non sono ancora stati utilizzati.
 - Il serbatoio contiene 4 unità d'acqua.
 - Gli irrigatori sono in uso.
 - Non si è ancora verificata alcuna degradazione.

GUI - INTERFACCIA GRAFICA DI VISUALIZZAZIONE

Il programma simula la gestione di una serra utilizzando un piano generato da un planner PDDL (ENHSP). L'interfaccia grafica, realizzata con Pygame, consente di visualizzare lo stato delle piante, del serbatoio d'acqua e degli irrigatori, oltre a navigare tra le azioni del piano.

L'obiettivo è mantenere le piante in salute (umidità alta) fino alla fine della simulazione, gestendo l'irrigazione, la manutenzione degli irrigatori e il livello del serbatoio.

Flusso del Programma

L'utente inserisce il nome del file PDDL del problema e sceglie la modalità del planner (sat-hadd per soluzioni subottimali o opt-hrmax per soluzioni ottimali). Il programma verifica la presenza del file e mostra un elenco di file pddl disponibili in caso di errore (attenzione a non inserire il domain). All'utente verrà poi chiesto con quale modalità verrà eseguito il planner ENSHP: sat-hadd (suboptimal), opt-hrmax (optimal).

```
C:\Users\Utente\Desktop\ESAME>serraguiirrigazionefinalev12.py
pygame 2.6.1 (SDL 2.28.4, Python 3.13.3)
Hello from the pygame community. https://www.pygame.org/contribute.html
Inserisci il nome del file problem (ATTENZIONE! Va scritto con estensione .pddl): e
File 'e' non trovato! File disponibili:
- serrairrigazionev1_domain.pddl
- serrairrigazionev1_problem.pddl
- serrairrigazionev2_domain.pddl
- serrairrigazionev2_problem.pddl
- serrairrigazionev3_domain.pddl
- serrairrigazionev3_problem.pddl
- serrairrigazionev4_domain.pddl
- serrairrigazionev4_problem.pddl
- serrairrigazionev5_domain.pddl
- serrairrigazionev5_problem1.pddl
- serrairrigazionev5_problem2.pddl
- serrairrigazionev5_problem3.pddl

Inserisci il nome corretto del file problem (ATTENZIONE:Va scritto con estensione .pddl): serrairrigazionev5_problem3.pddl
Inserisci la modalità di enshp (Suboptimal: digitare 'sat-hadd' Optimal: digitare 'opt-hrmax'): opt-hrmax
```

immagine 1

Il planner ENHSP viene eseguito per generare un piano di azioni basato sul dominio e sul problema forniti. Il piano viene analizzato e convertito in una lista di azioni. Lo stato iniziale del problema fornito viene costruito leggendo il file PDDL: include informazioni come umidità delle piante (bassa, media, alta), livello del serbatoio, stato di manutenzione degli irrigatori, contatori di utilizzi e manutenzioni per ogni irrigatore.

L'interfaccia grafica mostra lo stato corrente della serra: piante con colori che rappresentano il livello di umidità, serbatoio con il livello d'acqua attuale, irrigatori con informazioni sugli utilizzi e le manutenzioni.

L'utente può navigare tra le azioni del piano utilizzando i pulsanti "Avanti", "Indietro", "Inizio" e "Fine".

Ogni azione del piano aggiorna lo stato dell'interfaccia, modificando l'umidità delle piante, il livello del serbatoio e lo stato degli irrigatori.

COMPONENTI PRINCIPALI:

1. Funzione **run_planner**: esegue il planner ENHSP utilizzando il file di dominio e il file di problema forniti. Restituisce l'output del planner, che include il piano generato.
2. Funzione **parse_plan**: analizza l'output del planner per estrarre il piano, ovvero la sequenza di azioni da eseguire. Ogni azione viene aggiunta a una lista.

3. Funzione **estrai_piante_da_problem** :legge il file PDDL del problema per estrarre i nomi delle piante definite nella sezione :objects. Questi nomi vengono utilizzati per inizializzare lo stato delle piante nella simulazione.

4. Funzione **parse_initial_state**: costruisce lo stato iniziale della simulazione leggendo il file PDDL. Inizializza l'umidità delle piante, lo stato degli irrigatori, il livello del serbatoio, i contatori di utilizzi e manutenzioni per ogni irrigatore. E' bene precisare che nell'interfaccia l'utilizzo degli irrigatori non corrisponde alla funzione utilizzata in PDDL: nel caso della Gui serve per contare il numero di utilizzi effettivamente svolti alla fine del plan, da confrontare con il numero di manutenzioni eseguite. Il numero di manutenzioni eseguite dovrà essere sempre di uno inferiore al numero di utilizzi poiché nel goal dei vari problemi proposti non è stata richiesta una manutenzione finale. Sarà l'irrigazione che gestirà lo stato degli irrigatori come precondition.

5. Funzione **update_state**: aggiorna lo stato della simulazione in base all'azione eseguita. Gestisce diverse azioni, tra cui:

- irrigazione: aggiorna l'umidità della pianta, riduce il livello del serbatoio e incrementa il contatore degli utilizzi dell'irrigatore.
- ricarica-serbatoio: riempie il serbatoio al livello massimo
- esegui-manutenzione: imposta lo stato di manutenzione di un irrigatore (True se in manutenzione, False se non lo è).
- manutenzione-finita: incrementa il contatore delle manutenzioni.
- umidità-media e umidità-bassa: definiscono il degrado dell'umidità delle piante.

6. Funzione **draw**: disegna lo stato corrente della simulazione nella finestra Pygame. E' presente il titolo, il giorno, il momento della giornata (es. "Lunedì - mattina", stato iniziale) e l'azione successiva mostrata in grafica. Vengono disegnati anche il serbatoio e i pulsanti per navigare il plan. Ogni pianta viene rappresentata come un cerchio colorato: rosso per umidità bassa, arancione per umidità media, verde per umidità alta. Per gli irrigatori, rappresentati da un rettangolo grigio, viene mostrato lo stato di manutenzione (MAN se in manutenzione, OK! se funzionante), il numero di utilizzi e il numero di manutenzioni effettuate.

7. Funzione **animate_serbatoio**: anima il cambiamento del livello del serbatoio durante la simulazione. Quando il livello viene modificato, la funzione crea un'animazione fluida per rappresentare visivamente il cambiamento.

8. Funzione **main**: è la funzione principale che gestisce l'intero ciclo della simulazione. Si occupa di inizializzare Pygame e configurare la finestra grafica, gestire gli eventi dell'utente (come clic sui pulsanti o chiusura della finestra), avviare il planner e caricare il piano generato. Gestisce anche la navigazione nel piano, aggiornando lo stato e ridisegnando la finestra grafica in base all'azione corrente.

Navigazione del Piano

Dopo che l'utente ha fornito problem e modalità da prompt dei comandi (**immagine 1 precedente**), può navigare tra le azioni del piano a partire da Avvio simulazione (**immagine 2**): il click porta a una schermata di caricamento in attesa del plan fornito da ENHSP (**immagine 3**).

Segue il passaggio a un'altra finestra che permette la visualizzazione dello stato iniziale che varierà a seconda del problem selezionato (**immagine 4**). Da questo punto in poi, si può navigare tra le azioni del piano utilizzando i seguenti pulsanti: Avanti (esegue l'azione successiva), Indietro (torna all'azione precedente.), Inizio (riporta lo stato all'inizio del piano) e Fine (porta lo stato alla fine del piano).

L'interfaccia termina la sua visualizzazione del plan la domenica-pomeriggio poiché l'azione successiva "sera" indica la terminazione del goal (cioè domenica sera). Tutte le piante saranno contraddistinte dal cerchio verde (**immagine 5**)

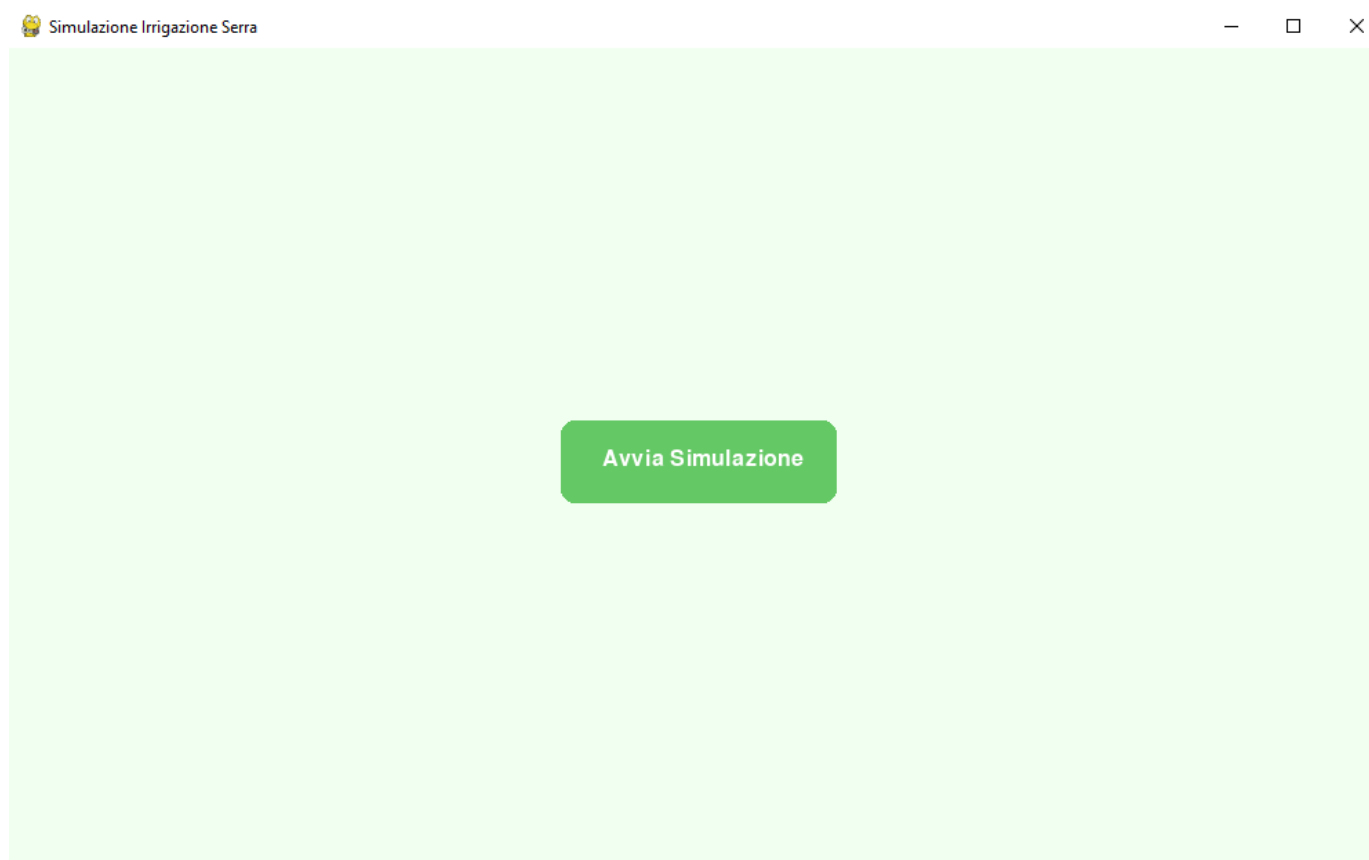


immagine 2






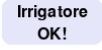


immagine 3

Simulazione Irrigazione Serra

Giorno: Lunedì - Mattina



Acqua
4/4

 Lattuga Serve irrigazione Acqua richiesta: 1	 Pomodori Serve irrigazione Acqua richiesta: 2	 Carote In salute Acqua richiesta: 3
 N.utilizzi: 0 N.manutenzioni: 0	 N.utilizzi: 0 N.manutenzioni: 0	 N.utilizzi: 0 N.manutenzioni: 0

VERDE = Livello di umidità alto
ARANCIONE = Livello di umidità medio
ROSSO = Livello di umidità basso

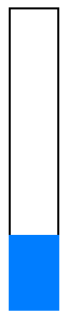
Azione successiva mostrata: irrigazione lattuga irrigatore-lattuga

Inizio Indietro Avanti Fine




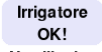
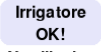
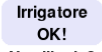
immagine 4

Simulazione Irrigazione Serra

Giorno: Domenica - Pomeriggio



Acqua
1/4

 Lattuga In salute Acqua richiesta: 1	 Pomodori In salute Acqua richiesta: 2	 Carote In salute Acqua richiesta: 3
 N.utilizzi: 4 N.manutenzioni: 3	 N.utilizzi: 4 N.manutenzioni: 3	 N.utilizzi: 3 N.manutenzioni: 2

VERDE = Livello di umidità alto
ARANCIONE = Livello di umidità medio
ROSSO = Livello di umidità basso

Azione successiva mostrata: sera

Inizio Indietro Avanti Fine

immagine 5

Ambiente di sviluppo PDDL e Python: Visual Studio Code