

Relazione

Descrizione della Strategia Risolutiva e dell'Architettura Proposta:

Architettura Generale:

- L'architettura è basata su un modello di simulazione multi-agente, in cui diversi agenti interagiscono all'interno di un ambiente simulato.
- L'ambiente di simulazione (AbstractEnvironment) fornisce un'interfaccia per l'inizializzazione, l'esecuzione dei passi e la gestione delle azioni degli agenti.
- Gli agenti (AbstractAgent) rappresentano entità che operano all'interno dell'ambiente di simulazione. Ogni agente ha la capacità di percepire l'ambiente e di eseguire azioni in risposta a tali percezioni.

MasterAgent:

- La classe MasterAgent agisce come il controllore principale della simulazione.
- Utilizza un approccio basato su thread per coordinare l'esecuzione dei passi della simulazione e la gestione degli agenti.
- Suddivide gli agenti tra diversi worker agent per eseguire i passi della simulazione in modo concorrente.
- Utilizza un executor per eseguire i passi dei worker agent in modo asincrono.

WorkerAgent:

- I WorkerAgent sono responsabili dell'esecuzione dei passi degli agenti assegnati.

- Ogni worker agent esegue i passi degli agenti assegnati in parallelo, consentendo una maggiore efficienza nell'esecuzione della simulazione.
- Utilizzano un flag di stop per determinare quando interrompere l'esecuzione della simulazione.

Approccio Asincrono:

- L'utilizzo di un executor e di CompletableFuture consente di eseguire i passi della simulazione in modo asincrono e concorrente.
- Questo approccio migliora le prestazioni complessive della simulazione, consentendo il parallelismo nell'esecuzione dei passi degli agenti.

Strategia Risolutiva:

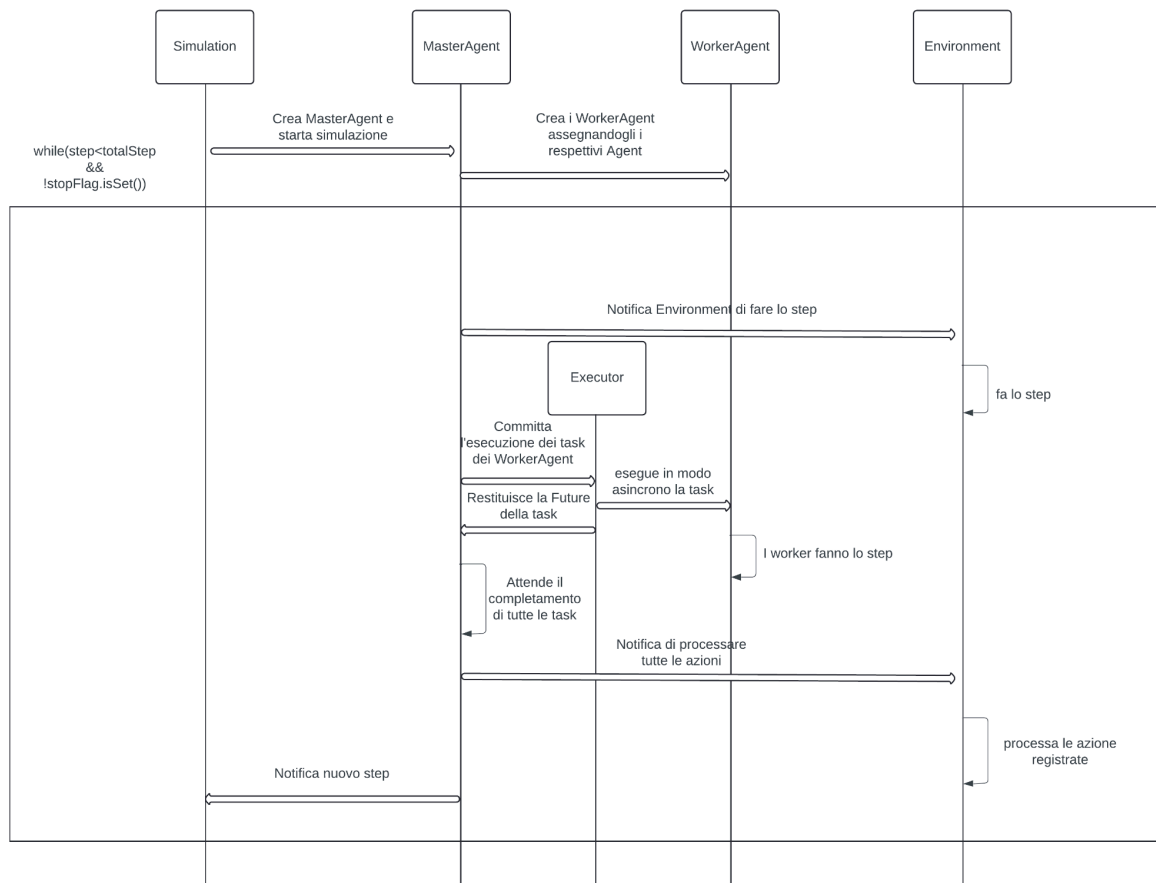
- La strategia risolutiva si basa sull'organizzazione della simulazione in passi discreti, in cui ogni passo comporta l'aggiornamento dello stato dell'ambiente e l'esecuzione delle azioni degli agenti.
- L'uso di thread e executor consente di gestire in modo efficiente l'esecuzione dei passi della simulazione e il coordinamento tra gli agenti.
- La suddivisione degli agenti tra worker agent consente di sfruttare al meglio le risorse del sistema e migliorare le prestazioni complessive della simulazione.

Descrizione del Comportamento del Sistema:

Il sistema inizia con la configurazione dell'ambiente di simulazione e l'aggiunta degli agenti ad esso.

Successivamente, il MasterAgent avvia l'esecuzione della simulazione, suddividendo gli agenti tra i worker agent e coordinando l'esecuzione dei passi della simulazione. Durante ogni passo della simulazione, l'ambiente viene aggiornato e vengono eseguite le azioni degli agenti, il tutto in modo

asincrono e concorrente. Alla fine della simulazione, viene fornito un report sullo stato finale del sistema e sulle prestazioni complessive della simulazione.



Report sulle Prestazioni:

- I dati forniti indicano che la simulazione è stata completata in un totale di 12577 millisecondi, eseguendo 100 passi e coinvolgendo 500 veicoli.
- Tempo per Passo: Il tempo medio per ciascun passo è stato di circa 118 millisecondi. Questo suggerisce che il sistema è in grado di gestire in modo efficiente le operazioni richieste in ciascun passo della simulazione.
- Utilizzo della CPU: L'utilizzo della CPU è stato misurato al 70%. Questo valore indica un carico di lavoro

significativo sulla CPU durante l'esecuzione della simulazione. Potrebbe essere necessario valutare se ci sono opportunità per ottimizzare l'efficienza dell'algoritmo o l'utilizzo delle risorse della CPU.

- Thread Pool: Sono stati utilizzati due thread pool distinti durante l'esecuzione della simulazione:
 - Un thread pool è stato utilizzato per il controller della GUI in modo da consentire l'esecuzione di più simulazioni in contemporanea. Questo è un approccio comune per gestire le interazioni dell'interfaccia utente in modo reattivo e fluido.
 - Un thread pool con 9 thread è stato utilizzato per i worker agent. Questi thread sono responsabili dell'esecuzione delle operazioni asincrone all'interno della simulazione. L'uso di più thread per i worker agent consente di elaborare più operazioni in parallelo, migliorando così le prestazioni complessive della simulazione.

In generale, l'uso della programmazione asincrona basata su FrameworkExecutor ha consentito di gestire in modo efficiente la simulazione, consentendo l'esecuzione di 500 veicoli su 100 passi con un tempo di completamento accettabile e un utilizzo efficiente delle risorse della CPU.