

Analisi e sviluppo di un componente Java per la generazione automatica di modelli NetLogo

Candidato: Aurel Pjetri

Relatore: Prof. Enrico Vicario

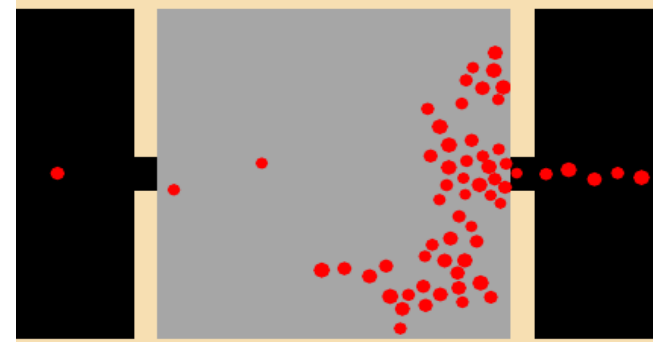
Co-Relatore: Dott. Sandro Mehic

Simulazione delle folle

Gli approcci più diffusi si dividono in due categorie:

- Microscopico:

rappresenta ogni persona come un agente, particella o macchina a stati finiti.



Simulazione di evacuazione tramite *social force model*

- Macroscopico:

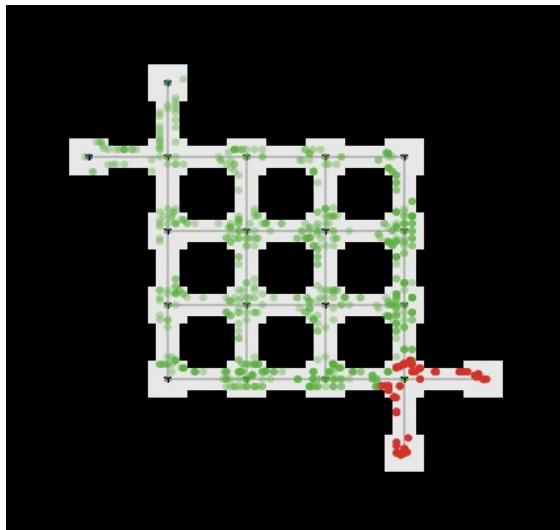
studia caratteristiche macroscopiche come densità media e velocità della folla.



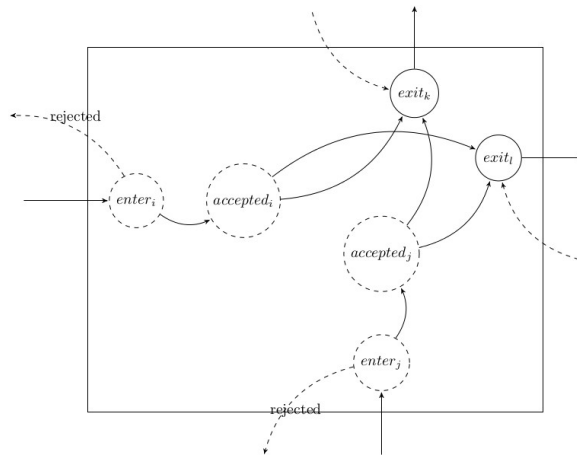
Simulazione tramite approccio *fluid dynamics*

Approccio gerarchico

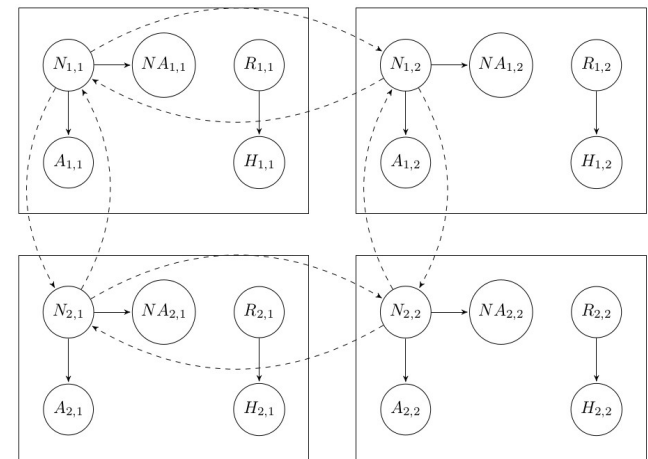
Utilizza tre livelli di scala per dare una soluzione analitica indipendente dal numero di agenti.



Microscopico



Mesoscopico

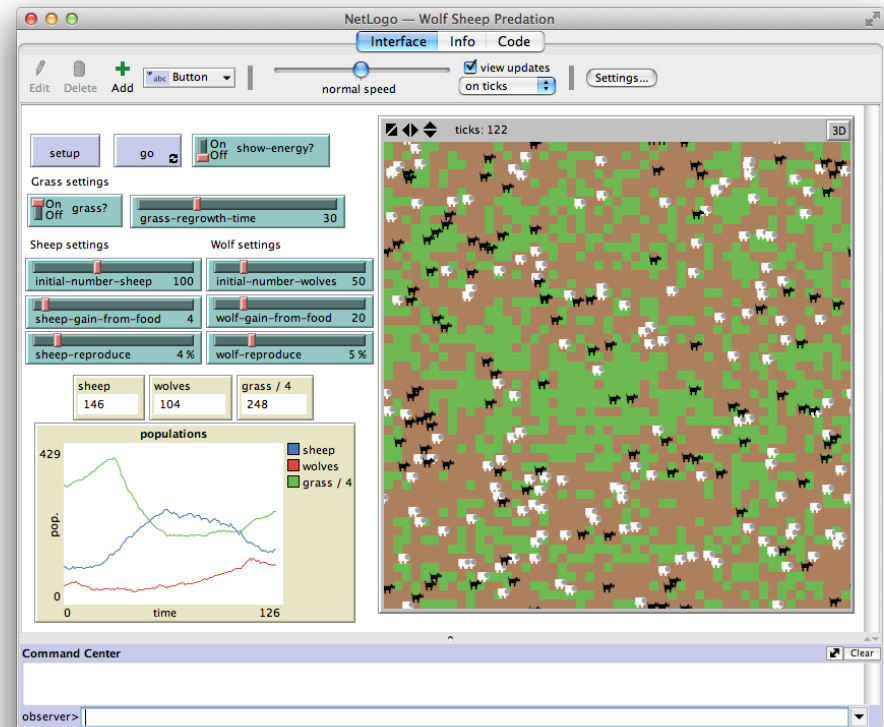


Macroscopico

NetLogo

Ambiente di modellazione di sistemi complessi e
linguaggio di programmazione agent-based

- Gli **agenti** hanno uno **stato** proprio e agiscono in modo **indipendente** e **concorrenziale**.
- Il mondo è suddiviso in **patches** interamente programmabili.
- Ampiamente accettato dalla comunità.
- É lo strumento più diffuso per la modellazione agent-based.
- Linguaggio di programmazione **procedurale** poco flessibile e di difficile gestione.

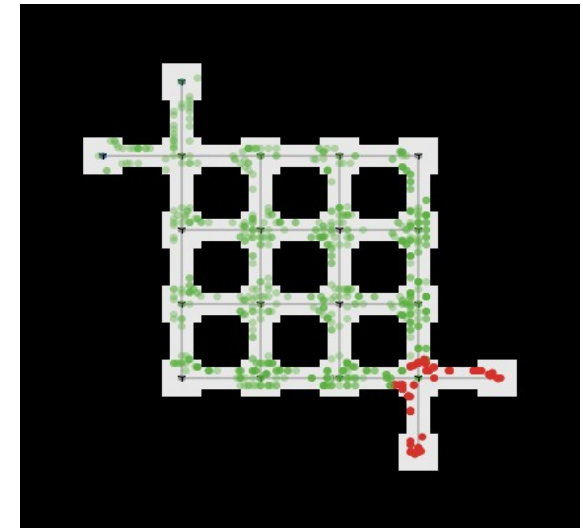
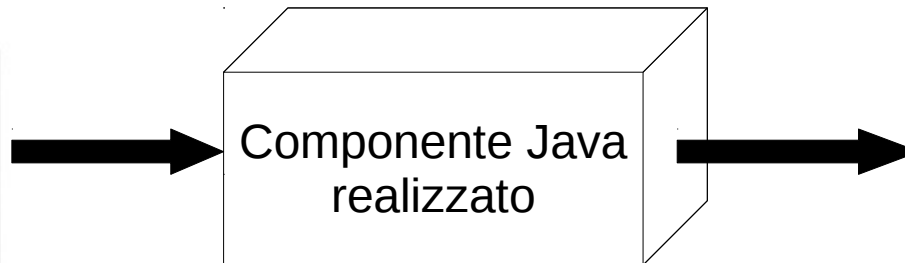


Obiettivo

Automatizzare il processo di scrittura del codice NetLogo
che esegue le simulazioni

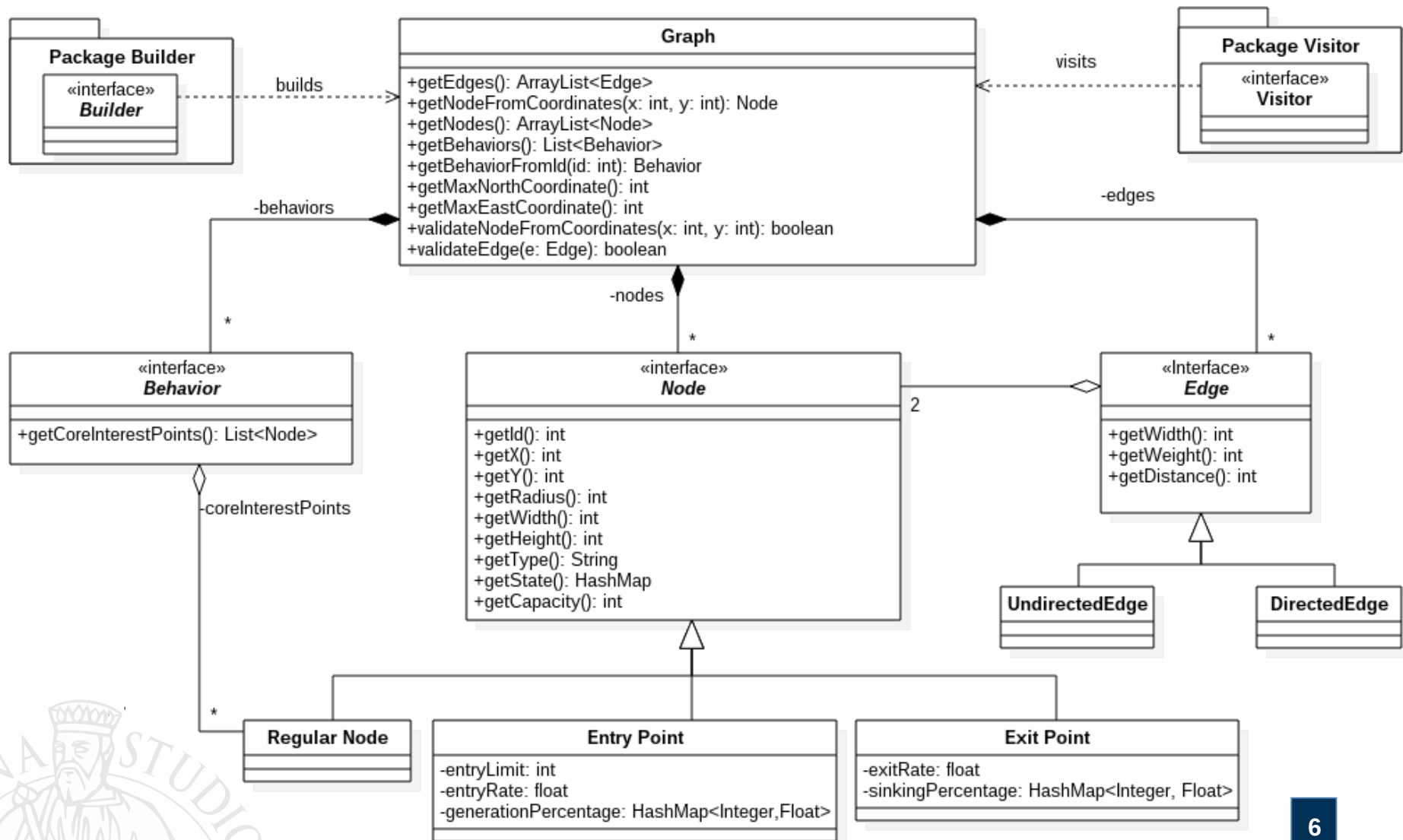


Documento
XML



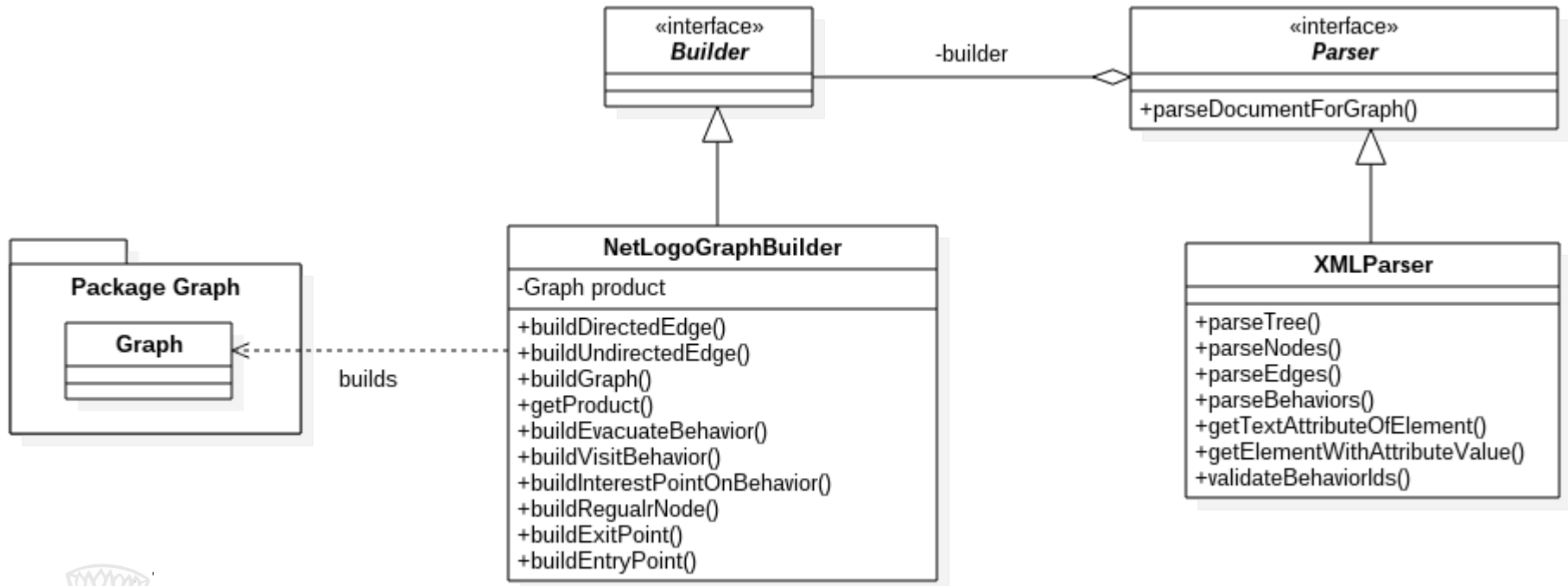
Modello NetLogo

Logica di dominio



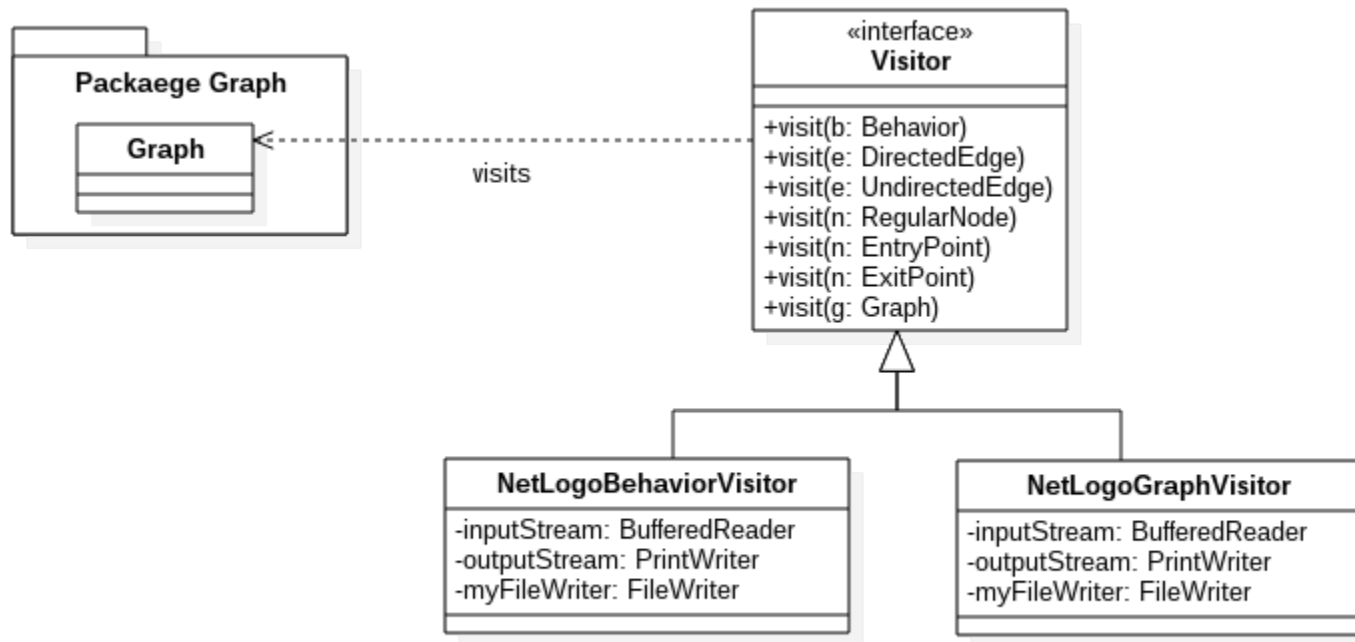
Builder

- **XMLParser** analizza il documento ed estrae le informazioni di interesse.
- **NetLogoGraphBuilder** costruisce e compone gli oggetti della struttura.



Visitor

- Usano le interfacce degli oggetti della struttura per estrarre le informazioni.
- Scrivono il codice NetLogo.



Workflow

```
<?xml version="1.0" encoding="UTF-8" ?>
<grafoxml>
  <!-- attributi per i nodi -->
  <key id="nType" for="nType" value="normal" />
  </key>
  <key id="nx" for="nod" value="x" />
  <key id="ny" for="nod" value="y" />
  <key id="nWidth" for="nod" value="width" />
  <key id="nHeight" for="nod" value="height" />
  <key id="nRadius" for="nod" value="radius" />
  <!-- attributi per le linee -->
  <key id="moverQuantit" for="mover" value="1" />
  </key>
</grafoxml>
```

file XML

Analisi del
documento XML
e costruzione
degli oggetti
Java

Visita
degli oggetti
e scrittura
del codice NetLogo

```
extensions [nw table]

;; Beacons are nodes
;; the shortest path
;; reach their destination
breed [beacons beacon]
;; These turtles are
breed [street-drawers]

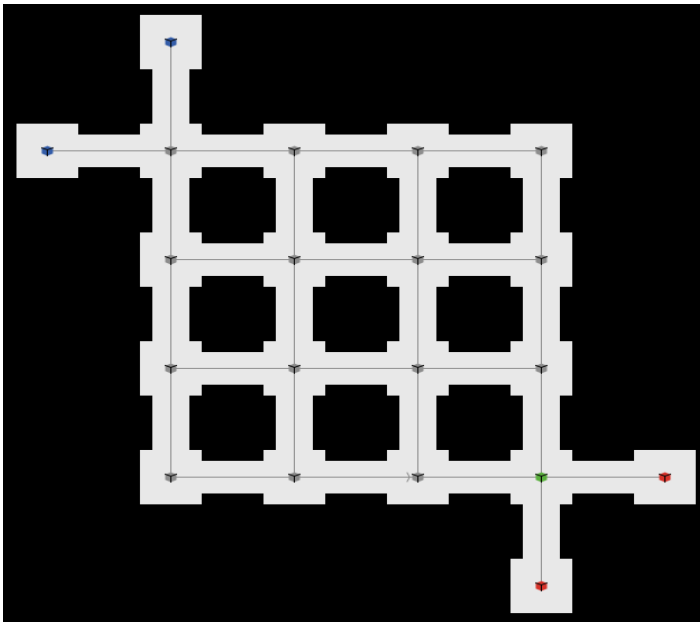
;; Movers are turtles
breed [movers mover]

;; Streets are links
undirected-link-breed

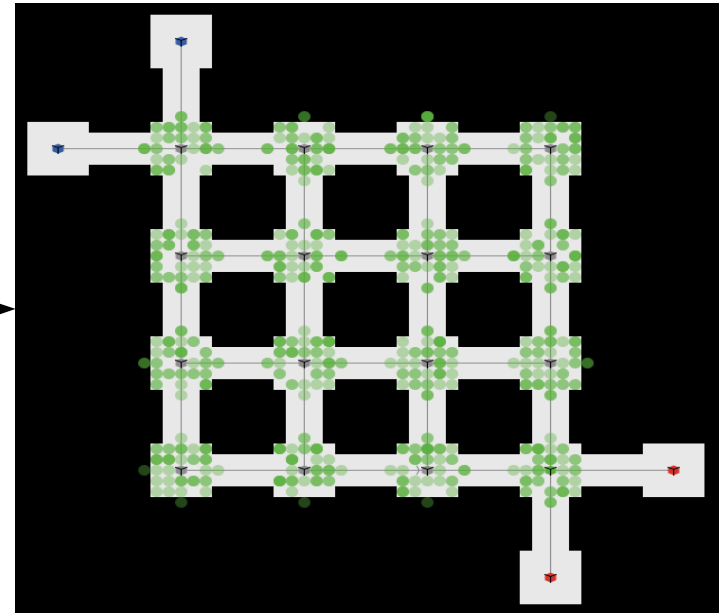
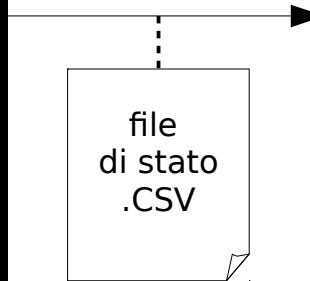
;; Streets are links
undirected-link-breed
```

2 file .nlogo

Modelli NetLogo



Modellazione dell'ambiente



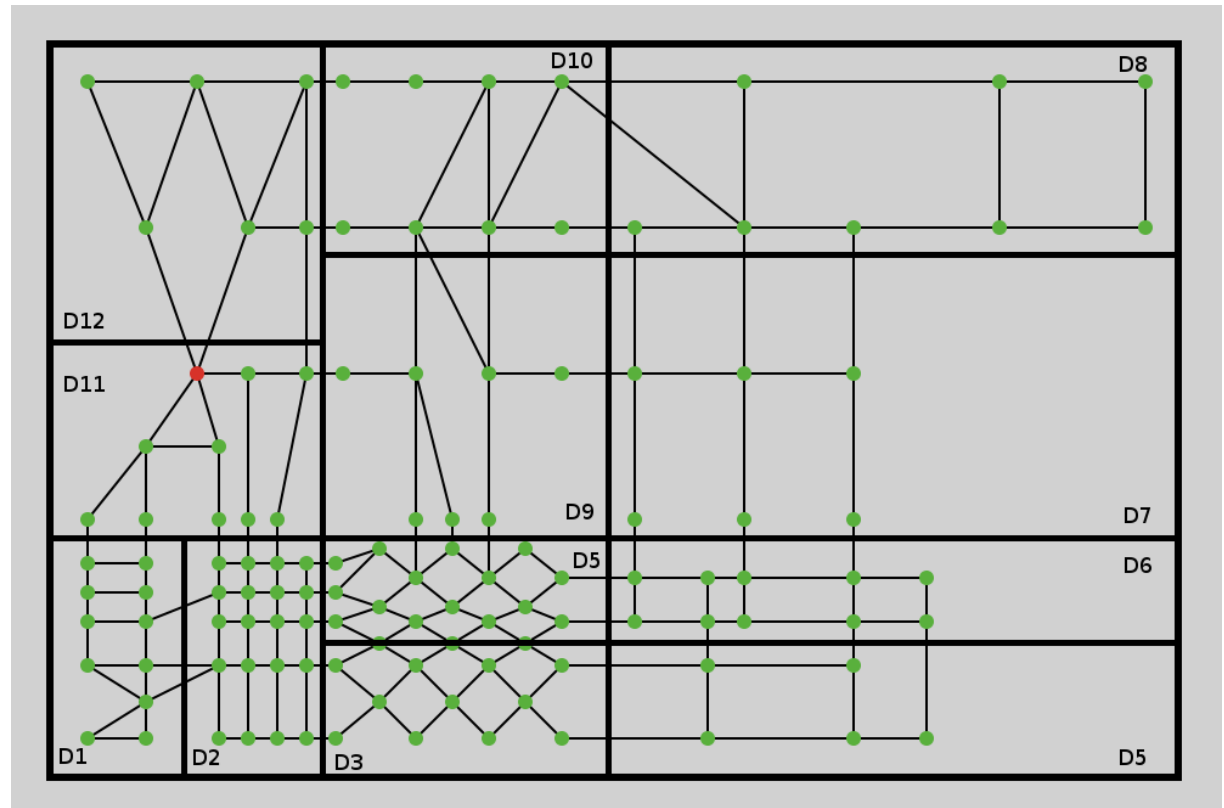
Modellazione del movimento degli attori e raccolta dei dati

Esperimenti

Simuliamo l'evacuazione su un modello ispirato alla città di Firenze

Per gli scenari A e D:

- Due modalità di simulazione: con **densità costante** e **transitoria**.
- Tre stati iniziali diversi: **alta**, **media** e **bassa densità di affollamento**.



Risultati

- Grande differenza di tempi tra modelli a densità costante e i transitori
- Lo scenario A impiega più tempo del monolitico, mentre lo scenario D la metà
- In caso di modifiche possiamo simulare solo le regioni interessate
- Nel modello monolitico impiego sempre lo stesso tempo

Scenario	densità costante	transitorio
Monolitico		9:19:40
A	21:33:36	0:21:51
D	4:49:27	0:23:13

Tempi totali di simulazione

Regione	Densità costante	Transitorio
A1	3:16:25	0:03:06
A2	2:13:11	0:02:40
A3	5:12:48	0:03:40
A4	6:20:39	0:06:08
A5	2:36:25	0:03:29
A6	1:46:20	0:02:46

Scenario A

Regione	Densità costante	Transitorio
D1	0:27:26	0:01:56
D2	0:29:15	0:02:03
D3	0:49:46	0:01:58
D4	0:29:48	0:02:00
D5	0:42:43	0:01:51
D6	0:37:23	0:01:40
D7	0:12:58	0:01:58
D8	0:11:20	0:02:07
D9	0:23:57	0:01:58
D10	0:21:52	0:02:06
D11	0:02:08	0:01:43
D12	0:0:52	0:01:48

Scenario D

Conclusioni e sviluppi futuri

Abbiamo illustrato il funzionamento del componente Java sviluppato e il contesto di ricerca in cui questo viene utilizzato.

Possibili sviluppi futuri potrebbero essere:

- Estendere la logica di rappresentazione dei comportamenti includendo anche aspetti **sociali** degli agenti come altruismo e conformismo
- Implementare una interfaccia che faciliti l'utilizzo
- Implementare nuovi parsers per formati come il CSV

