**Relazione PCD1718 Assigment #1 “The Game of Life”**

**Analisi del problema**

L’obiettivo dell’Assigment è l’implementazione in Java di una versione concorrene del gioco “The Game of Life” usando la programmazione concorrente multi-threaded.

Il gioco consiste nel calcolare e visualizzare l’evoluzione della matrice di celle che caratterizza il gioco, come sequenza di fotogrammi (ognuno dei quali rappresenta lo stato del mondo).

Nella matrice, ogni cella può essere in uno di due stati possibili, *live* e *dead*.

Dato lo stato s(t) della matrice, lo stato s(t+1) si computa con le seguenti regole:

* una cella m[i,j] che nello stato s(t) è *live* e ha zero o al più una cella vicina *live* (e le altre *dead)*, nello stato s(t+1) diventa *dead*(“muore di solitudine”)
* una cella m[i,j] che nello stato s(t) è *live* e ha quattro o più celle vicine *live*, nello stato s(t+1) diventa *dead*(“muore di sovrappopolamento”)
* una cella m[i,j] che nello stato s(t) è *live* e ha due o tre celle vicine *live*, nello stato s(t+1) rimane *live*(“sopravvive”)
* una cella m[i,j] che nello stato s(t) è *dead* e ha tre celle vicine *live*, nello stato s(t+1) diventa *live*

Il gioco deve presentare un’interfaccia grafica con pulsanti “start” e “stop” con cui si fa partire e fermare il gioco. Ogni stato del gioco deve essere visualizzato, insieme al numero di celle nello stato “live”.

**Indicazioni - requisiti**

* Il programma deve funzionare anche con matrici di dimensioni significative - es: (5000x5000)
* Massimizzare lo throughput, minimizzando il tempo di calcolo di ciascun fotogramma ed eventualmente anche della sequenza di fotogrammi
* Massimizzare la reattività della GUI
* Studiare e implementare meccanismi di coordinazione/sincronizzazione basati su semafori o monitor

**Descrizione della soluzione proposta**

**Architettura del sistema**

L’architettura del sistema è stata scomposta in tre livelli utilizzando il patten Model View Presenter usando il modello ad Interattori, di conseguenza vi sono tre moduli: il dominio applicativo, gli interattori e la vista.

Nel primo livello vi è una rappresentazione object oriented delle componenti del problema.

Nel secondo livello è stata incapsulata la logica di aggiornamento del gioco.

Nel terzo livello viene organizzata l’interfaccia grafica che permette la visualizzazione del gioco e l’interazione con l’utente.

**Implementazione**

**Dinamica del sistema - Reti di Petri**

**Performance**

**Java Path Finder**