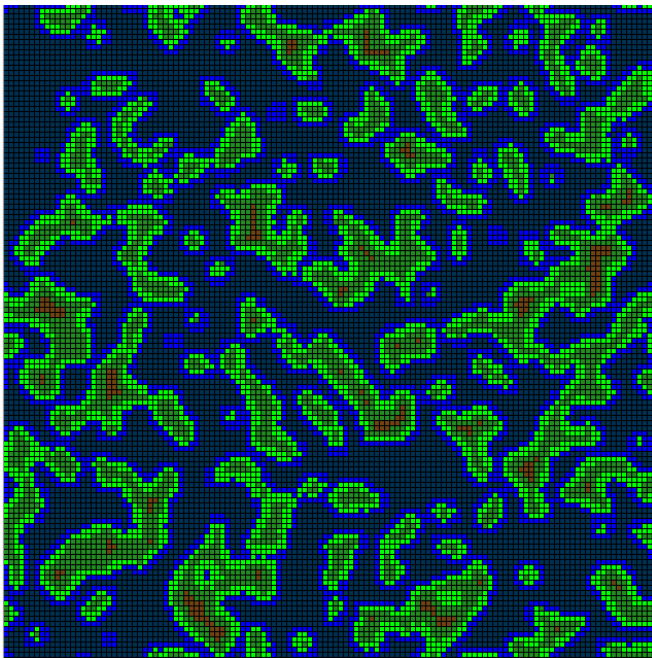
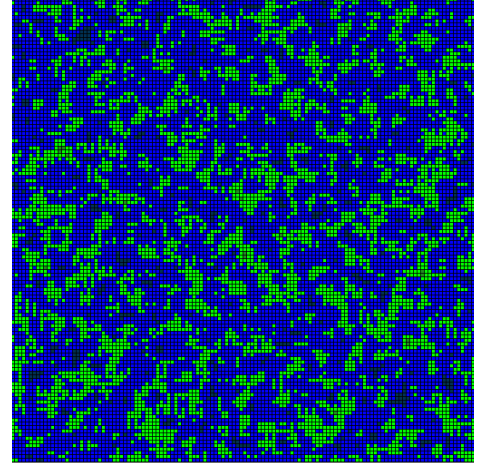


Map Generator

L'automa implementato è un *generatore di mappe* casuali, completamente creato da me.

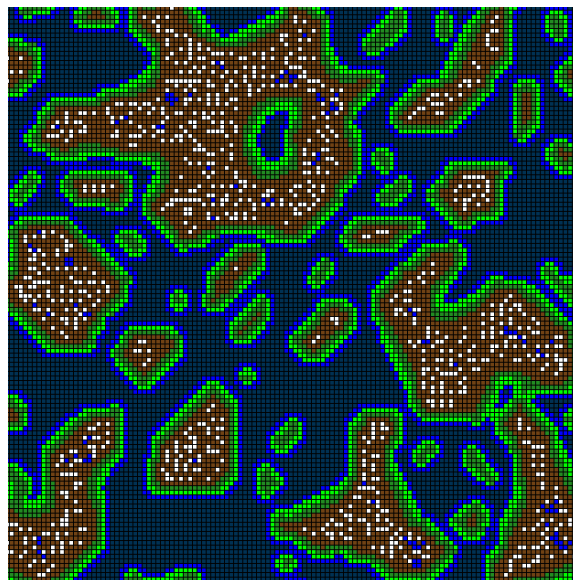
Lo stato iniziale di questo automa è una mappa con i 2/3 ricoperta da acqua e il restante di terra.

Con l'evolversi della mappa si possono notare più stati che ogni cella può assumere nel corso del tempo in base ai propri vicini.



L'automa genera tante piccole isole in mezzo al "mare" che possono avere due stati di profondità (*water*, *deep water*) in base alla vicinanza con le isole, queste isole hanno anche loro diversi stati (*land*, *forest*, *mountain*) che cambiano in base alla loro grandezza.

Oltre a generare isole, se essa è composta da un numero di montagne elevato, c'è la possibilità che si generi neve e successivamente si sciogla in acqua.



Sviluppo

Versione seriale: è rappresentata da una matrice, allocata dinamicamente, inizialmente riempita con valori. Tuttavia, sono necessarie due matrici per rappresentare il *passaggio corrente* e quello *successivo* e alla fine dello step si aggiorna la matrice di lettura scambiandola con quella di scrittura appena aggiornata.

Versione Parallela: come la versione seriale, i processi hanno due matrici, per lo stato corrente e successivo, che però in questo caso sono *sottomatrici* della matrice principale che utilizzeremo per la stampa. Poiché lo stato di una cella si basa su tutti i vicini, tutti i processi inviano la *prima riga al processo precedente* e *l'ultima riga al processo successivo*. Quindi procedono con il calcolo della nuova sottomatrice.

Al termine della fase di calcolo, il thread 0 riceve tutte le sottomatrici calcolate in un'operazione di raccolta (gather), in cui la matrice principale viene riempita da tutte le sottomatrici compresa quella del thread 0. Il thread 0 quindi procede a disegnare la matrice.

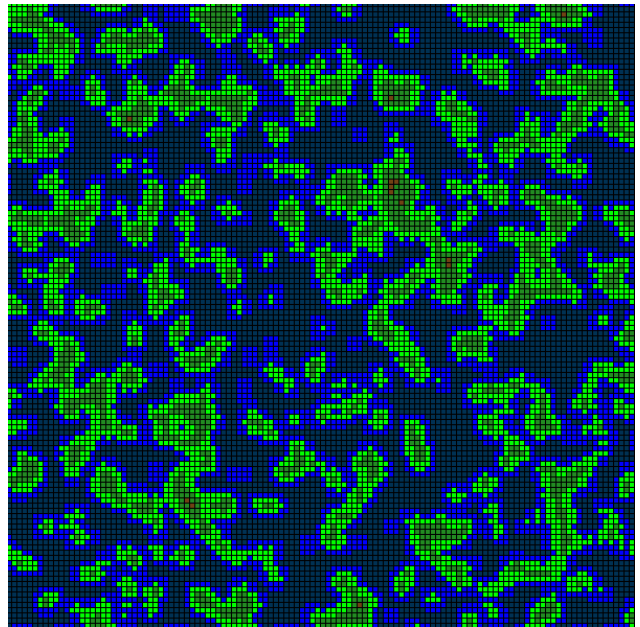
Stati celle

I stati delle celle sono: **Water**, **Deep water**, **Land**, **Forest**, **Mountain**, **Snow**.

Regole Celle

Water:

- Se non ha come vicine celle di **land**, **mountain** e **forest** cambia lo stato in **deep water**
- Se ha più di tre celle **land** vicine diventa anche lei **land**
- Se ha almeno tre celle di **forest** o **mountain** oppure almeno una cella di **snow** diventa **mountain**
- Se non ha rispettato nessuna delle condizioni allora rimane nello stato di **water**

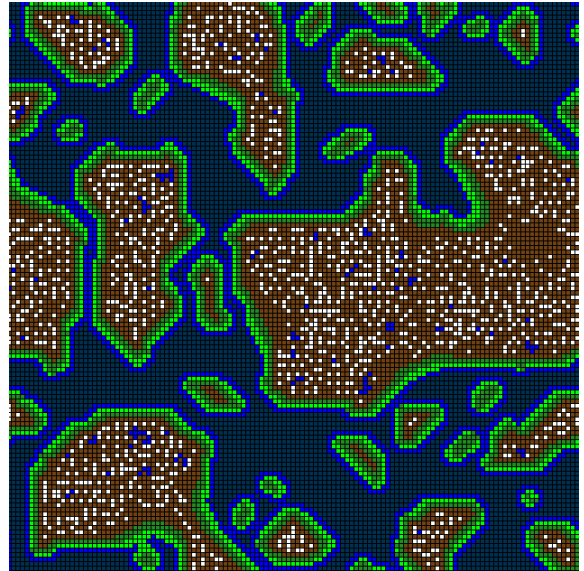


Land:

- Se la somma delle **land** e **forest** vicino è pari a tre o quattro (numero generato casualmente)
 - controlla se ha almeno sette o più **land** vicini o se la somma di **land**, **forest**, e **mountain** è pari a 8 e non ha celle vicine ne di **water** ne di **deep water** allora in questo caso la cella diventa **forest**.
 - Altrimenti rimane **land**
- Se la somma **land** e **forest** non è pari a tre o quattro allora la cella diventa **water**

Mountain:

- Se ha tutti vicini **mountain** oppure la somma di **mountain** e **snow** è pari a otto allora ha la possibilità del 5% di generare snow altrimenti rimane mountain



Deep Water:

- Se ha più di tre vicini **land** diventa anche lei **land**
- Se ha non ha ne vicini **land** , **forest** e **mountain** resta deep water
- Se non ha rispettato nessuna delle condizioni diventa water

Snow:

- Se ci sono più di due vicini **snow** allora diventa water
- Se non ha rispettato la condizione rimane nello stato di snow

Forest:

- Se la somma di celle **snow**, **mountain** e **forest** è pari a 8 allora diventa mountain
- Se ha otto celle **water** vicine diventa water
- Se ha quattro o più celle **water** vicine diventa **land**
- Se non ha rispettato nessuna delle condizioni rimane forest

Tempi

Processore: *AMD Ryzen 5 4600H (6 core) 3.00 GHz*

Tempo in millisecondi (*ms*) preso tramite la media di 10 misurazioni su ogni singolo test case.

Step eseguiti: *100*

Test dimensioni matrice: *132, 264, 1056*

Dimensione matrice (132 x 132)			
Threads	Time	Speedup	Efficienza
1	255,62058	1	1
2	145,55442	1,75618	0.87809
4	85,88670	2,97625	0.76255
6	66,02393	3.87163	0.64527

Dimensione matrice (264 x 264)			
Threads	Time	Speedup	Efficienza
1	1011,50784	1	1
2	575,56465	1,75741	0.87879
4	330,21054	3,06322	0.76580
6	241,97042	4,18002	0.69671

Dimensione matrice (1056 x 1056)			
Threads	Time	Speedup	Efficienza
1	17278,71277	1	1
2	9817,45505	1.75999	0.87999

4	5420,03974	3,18793	0.79698
6	3710,65725	4,65651	0.77608

NB: Questi tempi sono calcolati *senza stampa* e senza l'istruzione *Gather* che serve per far comunicare la propria matrice a chi deve stampare il risultato finale, questa operazione risulta pesante soprattutto quanto si utilizzano più thread.

(con Gather) 4385,38750 ms (6 Threads, matrice 1056 x 1056)

(senza Gather) 3710,65725 ms (6 Threads, matrice 1056 x 1056)

Progetto sviluppato per il corso "Algoritmi Paralleli e Sistemi Distribuiti"

Università della Calabria 2021/2022.

Andrea Tocci