

Progetto Assembly

Generatore procedurale di labirinti e pathfinder

Pensavo di implementare in assembly un semplice algoritmo per la generazione di labirinti perfetti.

Un labirinto perfetto è un labirinto in cui non ci sono loop o aree inaccessibili.

Una volta che l'algoritmo passa per un punto del labirinto (immaginabile come un array bidimensionale) quel punto si definisce "non più esplorabile", quindi "esplorato".

L'algoritmo che vorrei implementare è il "Recursive backtracker":

1. Se non ci sono più punti esplorabili l'esecuzione termina
2. Viene scelto un punto adiacente in una direzione (nord,sud,est o ovest)
3. Se tutti i 4 punti adiacenti sono stati già esplorati allora si torna indietro di un passo e si torna all'istruzione 2. (uso della stack)
4. Se il punto è già stato esplorato o non esiste (è fuori dai bordi) allora si controlla nella direzione successiva (si ripete l'istruzione con una direzione diversa).
5. Quando è stata trovata una direzione valida si fa un passo in quella direzione e si riparte dall'istruzione 1.

Le scelte di direzione e di posizione iniziale saranno definite proceduralmente da una funzione che genera numeri pseudo-casuali così definita:

1. shift a destra di 2 sul seed
2. xor tra il seed shiftato e il seed originale
3. shift a sinistra di 6 sul risultato del punto 2
4. xor tra il risultato del punto 3 e quello del punto 2

Inizialmente il programma richiede all'utente di inserire il numero (intero) del labirinto che si vuole generare e risolvere.

Questo intero farà da seed per la funzione di generazione di numeri casuali.

A questo punto il programma genera proceduralmente il labirinto.

Quando l'algoritmo termina, mostra graficamente il labirinto risolto e chiede all'utente se vuole generarne uno nuovo o se vuole uscire.

Esempio di visualizzazione grafica di un labirinto:

```
#####  
# #A#  #  
# #.### #  
#...#...#  
#...#...#  
#...#...#  
#####  
#B...#  
#####
```