











Scanline Sweep Algoritm för MineSweep Spel

Som en del i en programmeringsuppgift för att skapa ett MineSweep spel så behövs en algoritm för att rensa sammanhängande fält som inte innehåller minor.

När spelaren klickar på en ruta  utan som innehåller en mina eller inte ligger angränsande en mina så ska hela det sammanhängande fältet rensas.

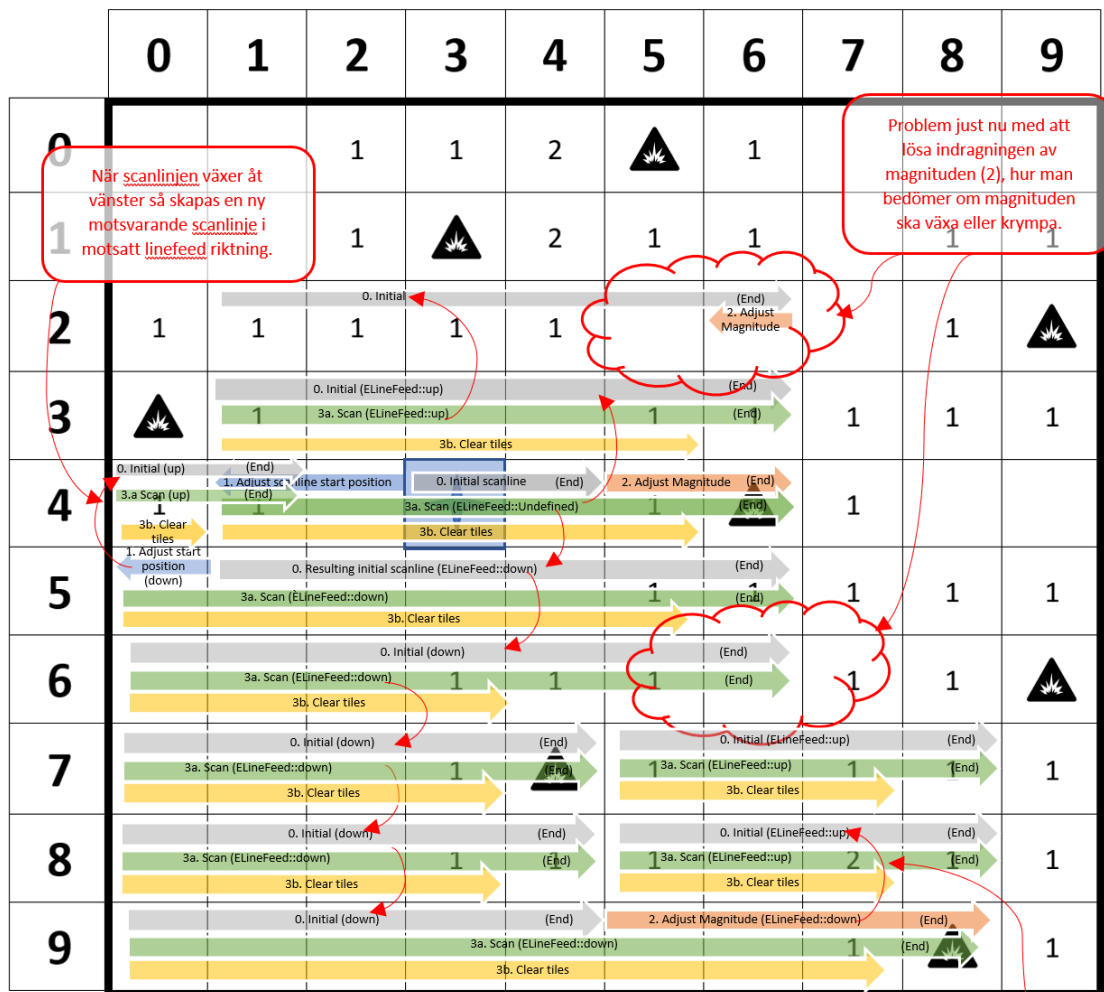
	0	1	2	3	4	5	6	7	8	9
0			1	1	2		1			
1			1		2	1	1		1	1
2	1	1	1	1	1				1	
3		1				1	1	1	1	1
4	1	1				1		1		
5						1	1	1	1	1
6				1	1	1		1	1	
7				1		1		1	1	1
8				1	1	1		2	1	1
9								1		1



En enklare algoritm som kan användas är en Flood Fill algoritm, men den algoritmen berör rutor på flera rader samtidig som den berör rutor på samma rad, vilket gör att algoritmen hoppar fram och tillbaka i minnet.

Denna scanline algoritmen är tänkt rensa en rad åt gången vilket förhoppningsvis är effektivare ur minnessynpunkt.

Jag har försökt att visualisera min scanline algoritm i nästa bild. I skrivandets stund så har jag några problem kvar att lösa. Bland annat så måste ibland scanlinen krympa för att inte "läcka" in i andra ytor som inte ska rensas.



Karl Hansson 2020

Algorithm:

1. Spelaren klickar (eller väljer) en rutan som inte innehåller en mina eller inte ligger nära en mina, alltså en ruta utan mina eller nummer.
2. En initial (0.) scanline som har en start position, en magnitud och ett line feed värde. Första scanlinen har en start position som motsvarar rutan som spelare klickade på, och har en magnitud = 1 (en ruta) och en line feed värde på ELineFeed::undefined. Efterföljande scanlines har andra värden för dessa men aldrig igen ELineFeed::undefined.
3. Om scanlinen inte börjar på 0 och ruta vid start positionen inte innehåller en mina eller nummer; expandera (1.) scanlinen åt vänster tills första rutan med ett nummer eller till början av raden (vilket som kommer först). (OBS: denna del av algoritmen behöver mer arbete eftersom den gör att fälten ibland blöder in till intilliggande fält som inte ska beröras).
4. Om sista rutan i scanlinen är tom expandera (2.) åt scanlinen åt höger tills första rutan med nummer eller tills slutet av raden (vilket som kommer först). (OBS: denna del av algoritmen behöver mer arbete eftersom den gör att fälten ibland blöder in till intilliggande fält som inte ska beröras).
5. Därefter skanna scanlinen:

- a. Skapa en ny scanline för nästa rad* för varje område i scanlinen som börjar med antingen första rutan i raden eller en ruta med nummer, och som slutar med antingen en ruta med ett nummer eller sista rutan i raden. Spara de nya scanlinen i en vector för senare hantering. *Nästa rad beror på vilket line feed värde scanlinen har (ELineFeed::up eller ELineFeed::down). Om line feed värdet är ELineFeed::undefined så skapa scanlines för både raden ovanför (som därefter har ELineFeed::up) och raden under (som därefter har ELineFeed::down).
 - b. Rensa alla rutor som berörs. Om en ruta redan är rensad så avbryt skapandet av den nästa scanlinen, skanning av den nuvarande scanlinen fortsätter dock enligt ovan till slutet.
6. Om det finns några sparade scanlines (som sparade i 5.a.) kvar att hantera; repetera steg 2 – 6 igen med en av de sparade scanlinens.

Jag har som sagt några saker kvar att lösa för att algoritmen ska fungera men jag ville ändå beskriva den för er. Jag ber om ursäkt för Svenelskan ovan, jag skriver hellre sådant här på engelska. Jag hoppas att ni förstår min beskrivning ovan och min kod går att hitta på:

https://github.com/Firelark/Karls_Minesweeper

Jag tar gärna kritik och kommentarer på min både min algoritm och min kod.

Med vänlig hälsning

Karl Hansson

karlhansson@outlook.com

2020-03-07