Hoe werkt de scroll routine?

Globale variabelen

Variabele w en h geven aan hoeveel vakjes er op het scherm getekend moeten gaan worden

```
int w = 10;
int h = 10;
```

Variabele fw en fh geven aan hoeveel vakjes er uit het bestand gelezen moeten gaan worden
ik gebruik een bestand van 12 bij 12 vakjes

```
int fw = 12;
int fh = 12;
```

De refx en refy variabelen geven weer **vanaf welke plek** in het bestand ik de 10 bij 10 vakjes moet gaan tekenen

```
int refx = 0;
int refy = 0;
```

De grid is een twee-dimensionale array
De speldata is een array van String

```
int[][] grid;
String[] spelData;
```

De setup routine

Nadat size(640, 480) is aangeroepen zijn de breedte en de hoogte van het scherm, gemeten in **pixels** bereikbaar via de width en height variabelen.

```
void setup()
{
	int i, j;

	size(640, 480);
```

De **fh** variable (file height) staat op twaalf want mijn *game.dat* file is 12 x **12** karakters
De **fw** variable (file width) staat op twaalf want mijn *game.dat* file is **12** x 12 karakters

```
	spelData=new String[fh];
	spelData=loadStrings("game.dat");
```

Hier vraag ik processing een nieuwe twee-dimensionale array van int die is **fw** breed en **fh** hoog.

```
	grid=new int[fw][fh];

	for (i=0; i<fh; i++)
		grid[i]=new int[fw];
```

Hier vul ik de grid array met de waarden (het zijn karakters dus gebruik ik charAt) uit het *game.dat* bestand.

```
	for ( i=0; i< fw; i++) {
		for (j=0; j< fh; j++) {
			grid[i][j]=spelData[i].charAt(j);
		}
	}
}
```

De draw routine

```
void draw()
{

	background(255, 255, 255);
	fill(255, 255, 255);

	int i, j;
```

Ik gebruik een geneste **for** lus. En zet “WATER”, “GRAS” en “BOOM” op het scherm

```
	for ( i=refx; i < refx + w; i++ ) {
		for ( j=refy; j < refy + h; j++ ) {

			if ( grid[j][i] == '0' ) {
				fill(0, 0, 255);
				text( "WATER", (i - refx ) * (width/w), (j - refy) *
					(height/h)  );
			}else if ( grid[j][i] == '1' ) {
				fill(0, 255, 0);
				text(  "GRAS", (i - refx) * (width/w), (j - refy) *
					(height/h)  );
			} else if ( grid[j][i] == '2' ) {
				fill(0, 255, 0);
				text(  "BOOM", (i - refx) * (width/w), (j - refy) *
					(height/h) );
				}else {
					fill(255, 255, 255);
				}
		}
	}
}
```

Merk op deze regel:

```
text( "WATER", (i - refx ) * (width/w), (j - refy) * (height/h)  );
```

De formules
(i - refx) * (width / w)
en
(j - refy) * (height / h)

Beepaald waar ik het woord WATER moet gaan tekenen.

Ik teken rijen van **w** vakjes. Ik heb **w** op 10 gezet en de width is 640.
640 gedeeld door 10 is 64. Mijn eventuele plaatjes worden 64 pixels breed. Door **w** aan te passen kan ik ineens meer plaatjes op een rij tekenen.

Ik teken kolommen van **h** vakjes. Ik heb **h** ook op 10 gezet en de height is 480.
480 gedeeld door 10 is 48. Mijn eventuele plaatjes worden 48 pixels hoog. Door **h** aan te passen kan ik ineens meer plaatjes op een rij tekenen.

Een grid is als volgt opglagen in het geheugen:

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

Hier is **3,5** vetgedrukt, de variable grid[3][5] kan dan een '0' 1' '2' bevatten (voor water, gras of boom).

Wil ik weten waar ik op het processing venster moet gaan tekenen, is dat dus **drie posities naar rechts** en **vijf posities omlaag**.

De breedte van een vakje kan ik uitrekenen door de hoeveelheid vakjes op het scherm (de **w** variabele) te delen door de totale schermbreedte (de width)
De hoogte van een vakje kan ik uitrekenen door de hoeveelheid vakjes op het scherm (de **h** variabele) te delen door de totale schermhoogte (de height)

Vakje 3,5 moet ik dus tekenen op 3 * (width / **w**), 5 * (height / **h**)

en daarom wordt de x positie van het te tekenen plaatje 3*(640/10) = 192
en daarom wordt de y positie van het te tekenen plaatje 5*(480/10) = 240

De hele grid kan ik als volgt gaan tekenen:

```
for (i = 0; i <  w; i++ )
      for ( j= 0; j < h; j++ )
```

Dit is een geneste **for**.

De grid breedte staat nu ineens op **fw** (12) en de grid hoogte op **fh** (ook 12).

De grid is dus 12x12 vakjes en ik wil daarvan 10x10 vakjes tekenen.

Daarom bereken ik:

```
for ( i = refx; i < refx + w; i++ )
      for ( j = refy; j < refy + h; j++ )
```

Met **refx** en **refy** kan ik nu gaan bepalen vanaf welke positie in de grid ik wil gaan tekenen.

De keypressed routine

In de **keyPressed**() routine kan ik refx en refy veranderen met de a, s, d, w toetsen.

```
void keyPressed()
{

      if ( key == 'w' && refy > 0 ) {
            refy--;
      }
      if ( key == 's' && refy < fh-h ) {
            refy++;
      }
      if ( key == 'a' && refx > 0 ) {
            refx--;
      }
      if ( key == 'd' && refx < fw-w  ) {
            refx++;
      }
      if ( key == 't' ) {
            for ( int i =0; i<fh; i++)
                  println( spelData[i] );
      }
}
```

Vragen:

1. Waarom teken ik met behulp van een geneste for?
2. Kan ik de **w** en **h** variabelen zomaar op 5 en 11 zetten?
3. Kan ik een *game.dat* bestand maken van 50 x 40 vakjes en de **fw** en **fh** gewoon hier op aanpassen?