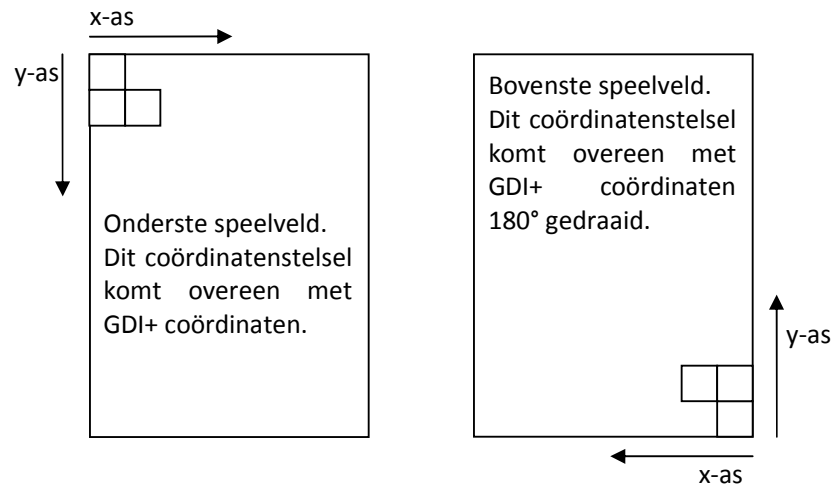


# 1 Algoritmiëk Etaris

## 1.1 Speelvelden



Figuur 1 Coördinatensysteem speelvelden

Deze definities gelden altijd, zo zijn het speelveld en het Grid altijd gedefinieerd:

Omlaag bewegen (vallen): in negatieve y-richting bewegen  
 Naar rechts bewegen: in positieve x-richting bewegen (zie opmerking hieronder)  
 Naar links bewegen: in negatieve x-richting bewegen (zie opmerking hieronder)  
 Omhoog bewegen: in positieve y-richting bewegen  
 Het veld is gedefinieerd in:  $0 \leq x \leq (width - 1) \wedge 0 \leq y \leq (height + offset - 1)$  met *width* de breedte in blokken, *height* de hoogte in blokken en *offset* hoogte van blokken buiten het zichtbare tekengebied van het speelveld.

Om het onderste speelveld te tekenen zoals weergegeven hoeven geen transformaties gedaan te worden, daar het gebruikte coördinatensysteem overeenkomt met het coördinatenstelsel dat de GDI+ (.NET) hanteert.

Het bovenste speelveld wordt in feite getekend zoals het onderste speelveld. Met exact dezelfde functies en dezelfde soort coördinaten. In feite komt het speelveld er dan uit te zien zoals het onderste speelveld. Echter, op het .NET object waarmee getekend wordt, wordt een transformatiematrix ingesteld. Deze matrix draait automatisch het hele coördinatensysteem. Kortom, de code hoeft op de meeste plaatsen geen rekening te houden met onder/boven of bovenste speelveld/onderste speelveld. Dit gaat automatisch en is transparant.

Een 180° rotatie is gemakkelijker daar nu geen speelstukken gespiegeld hoeven te worden. Immers, bij een spiegeling zouden gespiegelde speelstukken op het scherm kunnen verschijnen: speelstukken die niet in het configuratiebestand gedefinieerd zijn. Deze methode is simpeler en voorkomt dat probleem.

Echter, de Game klasse moet rekening houden met het bovenste speelveld. Wanneer de gebruiker een speelstuk naar links wil verplaatsen, moet het speelstuk in de positieve x-richting verplaatst worden, naar rechts dus. Voor een verplaatsing naar rechts geldt het omgekeerde.