

Amstelhaege - Documentatie JavaScript,

Met JavaScript, vanaf nu JS genoemd kan gebruikt worden voor vele gemakkelijke visualisaties in de browser. Dit is natuurlijk handig om het gemakkelijk en snel aan andere te kunnen laten zien. We zouden het namelijk zelfs online kunnen zetten. Denk bij een visualisatie aan die van Jasper in het hoorcollege afgelopen dinsdagavond. Alleen dan voor onze eigen opdracht.

Ik heb even snel rond gekeken, en ik weet niet of jullie JSFiddle kennen maar op die site kun je zowel HTML/CSS in combinatie met JS realtime runnen met een preview. Daar heb ik na snel wat variabelen te veranderen al een grid verkregen (<http://jsfiddle.net/h2yJn/2541/>).

Om een rect te plaatsen in JS kun je deze documentatie gebruiken (http://www.w3schools.com/tags/canvas_rect.asp). In deze kleine update kun je zien dat ik een “eengezinswoning” heb geplaatst op het canvas, het mooiste zou natuurlijk zijn als de zwarte lijnen achter de grid vallen, maar daar zal ik binnenkort naar kijken. <http://jsfiddle.net/h2yJn/2542/>

Als we er op de een of andere manier ervoor kunnen zorgen dat hij de beste oplossing gaat zoeken en dit gaat tekenen met de 4 waardes die we hebben: De eengezinswoning, de bungalow, de villa en het water, dan lijkt mij dit een prima manier om dit te visualiseren.

JS is een gemakkelijke taal, als dit runt bijvoorbeeld in Chrome kun je op f12 drukken om de console te openen en daar je errors in te zien. Daarnaast als je werkt met canvas heb je vrij snel door wanneer iets niet werkt, omdat dan het gehele programma niet werkt. Secuur zijn is natuurlijk belangrijk. Waar je op moet letten bij JS is ongeveer hetzelfde als in C, alleen hoeft je met JS niet druk te maken om geheugen en al die andere dingen die malloc etc gebruikt. Als je bovenaan een variable maakt met **var** zijn deze globaal en kunnen ze in een functie gebruikt worden, let wel op als je in die functie die je benoemd met **function** dat je je variabele overschrijft. Als je bovenaan niks definieert en in je functie bijvoorbeeld `x = 1` zet dan is deze niet global en bestaat deze alleen in de functie. Jasper vertelde in Data Processing dat het anders zat, maar zo heb ik het geleerd en dat hou ik dus aan.

Canvas

Het canvas wordt eigenlijk met maar twee gemakkelijke variabelen gemaakt, en dit zit standaard in JS.

`var canvas = $('<canvas/>').attr({width: cw, height: ch}).appendTo('body');` verteld waar de canvas in komt, in dit geval de body van de website

`var context = canvas.get(0).getContext("2d");` is voor JS om dingen op het canvas te zetten in wat voor stijl (3D werkt wel/niet daar heb ik mezelf nog nooit in verdiept). Dus voor nu houden we voor gemakkelijke 2D aan.

Verder kun je zien dat er gewoon **for** loops worden gebruikt om dingen op het canvas te zetten, dit kan in ons voordeel werken bij het maken van ons algoritme.

Als jullie nog vragen hebben stuur mij even een berichtje!