

Labo Webtechnologieën

Reeks 1: JavaScript

2 + 9 oktober 2019

Doel: Een eenvoudig spel implementeren in Javascript (ES6). Leren werken met het HTML5 canvas element en met de localStorage.

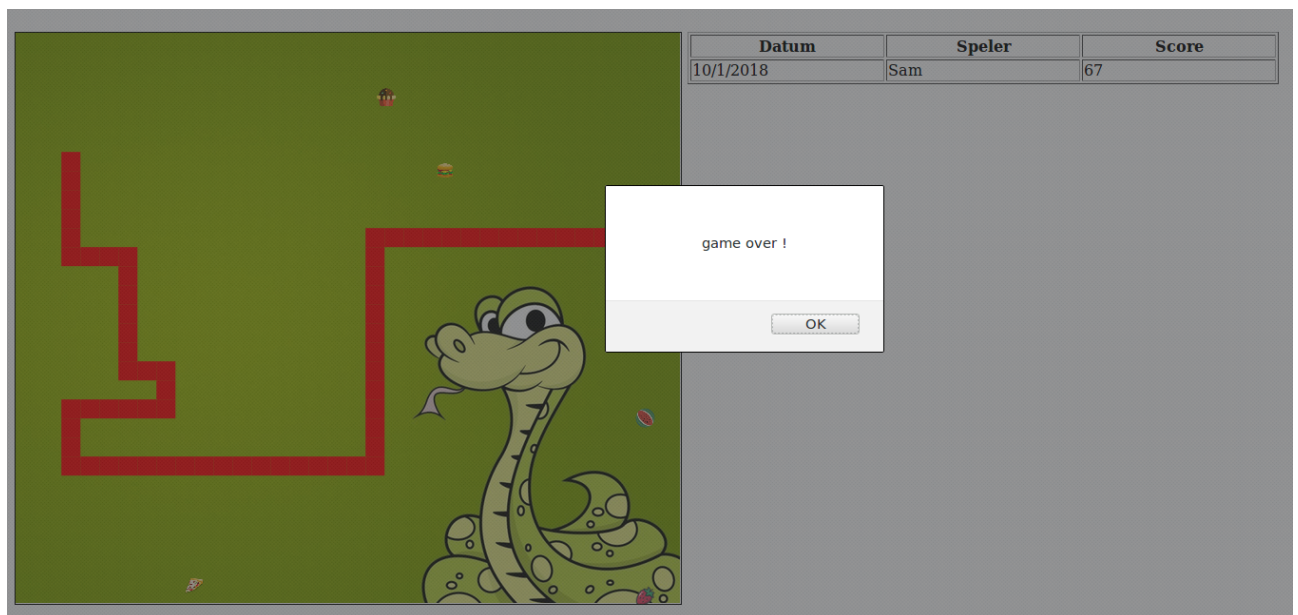
Voorkennis: Basis HTML en OOP.

1 Inleiding

HTML5 introduceert een aantal nieuwe elementen en functionaliteiten zoals semantic tags en form validation. Eén van de belangrijkste vernieuwingen is het canvas-element. Dit element laat toe om op een eenvoudige manier te tekenen binnen een webpagina. Een andere nieuwe functionaliteit is de “local storage” die het mogelijk maakt om gegevens lokaal (client side) te bewaren.

HTML bepaalt de inhoud van de pagina, de opmaak wordt beschreven door CSS. In dit labo gaan we intensief gebruikmaken van JavaScript. JavaScript is een scripttaal die toelaat om logica toe te voegen aan de webpagina.

Het doel van dit labo is om te leren werken met het HTML5 canvas en met de localStorage en om een complexe programmalogica in JavaScript te implementeren. In deze reeks zullen we een eenvoudige versie van het bekende spel “Snake” maken dat volledig binnen de browser draait.



2 ES6

Tijdens dit labo schrijf je ES6 Javascript code (let variabelen, arrow functions, classes, modules).

- Bekijk hoofdstuk 1 en 2 uit het boek “Advanced Game Design with HTML5 and Javascript” (zie theorie op Ufora).
- Maak een nieuw “Empty project” aan in Webstorm/IntelliJ of een andere editor naar keuze.
- ES6 is vrij nieuw, om het correct werkend te krijgen moeten we nog een aantal opties aanpassen in Webstorm/IntelliJ:

- Configureer Webstorm zodat ES6 gebruikt wordt: File -> Settings -> Languages & Frameworks -> Javascript
- Vink “Allow unsigned requests” aan in File -> Settings -> Build, Execution, Deployment -> Debugger.

3 Het HTML5 canvas

Het canvas biedt de ruimte aan waarbinnen het spel gespeeld zal worden.

- ◆ Maak een HTML5 pagina met een canvas van 700x600 pixels.
- ◆ Voeg de achtergrond (zie Ufora toe aan de pagina).

4 Experimenteren met het canvas

Het tekenen op het canvas gaat via JavaScript. Op internet vind je vele voorbeelden van hoe dit te doen. De officiële API is door W3C gedefinieerd: <http://www.w3.org/TR/2dcontext/>.

- ◆ Maak een nieuwe JavaScript file aan en voeg de verwijzing ernaar toe in je HTML. Indien je gebruik maakt van ES6 modules dien je de javascript als volgt te includeren: `<script src="script.js" type="module"></script>`
- ◆ Zoek op hoe je vanuit JavaScript toegang krijgt tot het canvas.
- ◆ Teken een eenvoudige figuur (bvb een rechthoek) op het canvas. Hoe zorg je ervoor dat je zeker bent dat de pagina volledig geladen is en de DOM volledig opgesteld is, vooraleer je begint te manipuleren?

5 Een animatie op het canvas

Om een animatie te verkrijgen zal je periodiek de positie van de elementen op het canvas moeten herberekenen en ze op de juiste plaats opnieuw tekenen.

- ◆ Zoek op hoe vanuit je vanuit JavaScript bepaalde code periodiek kan laten uitvoeren.
- ◆ Teken een enkel vierkantje op het canvas dat continu verspringt naar een willekeurige plek op het canvas (elke 100 ms). Bedenk dat het canvas hiervoor telkens hertekend wordt en er eigenlijk telkens een nieuw vierkant getekend wordt.
- ◆ Zorg nu dat dit vierkantje op het canvas continu verschuift in plaats van verspringt. Als het aan de rand komt moet het omkeren. De beginpositie initialiseer je met een random waarde maar je zorgt dat je een afstand van minstens 1/4 breedte en 1/4 hoogte van de rand blijft om te starten. De bewegingsrichting initialiseer je ook met een random waarde: ofwel horizontaal starten (random links of rechts), ofwel verticaal starten (random omhoog of omlaag) .

6 Luisteren naar keyevents

Zorg ervoor dat je de bewegingsrichting van het vierkantje kan veranderen met de pijltjestoetsen.

- ◆ Zoek op hoe je kan luisteren naar events van het toetsenbord.
- ◆ Achterhaal welke codes overeenkomen met de pijltjestoetsen. TIP: gebruik de developer tools in firefox en chrome, vooral het consolevenster en console.log kunnen je helpen.
- ◆ Implementeer het veranderen van de bewegingsrichting van het vierkantje met de pijltjestoetsen.
- ◆ Voeg ook de mogelijkheid toe om het spel te pauzeren door op “p” te drukken.

7 Voorstelling van de slang en programmalogica

Telkens de slang het voedsel gevonden heeft moet ze langer worden.

- ◆ Hoe teken je het voedsel op het canvas? Op Ufora staan een aantal figuren die je kan gebruiken. Zorg dat de figuur eerst ingeladen is vooraleer je deze probeert te tekenen (<http://www.jefclaes.be/2010/12/html5-drawing-images-to-canvas-gotcha.html>).
- ◆ Hoe plaats je tekst op het canvas (om de score weer te geven)?
- ◆ Hoe stel je de slang voor in code ? Maak gebruik van de nieuwe ES6 functionaliteit om je code te groeperen in klassen en modules. Stel een klassendiagram op.
- ◆ Beschrijf de logica van het programma.
- ◆ Implementeer deze logica.

8 Weergave van high scores

Dankzij de nieuwe local storage functionaliteit kunnen we gegevens lokaal (client side) bewaren. Deze gegevens zijn ook de volgende keer dat de pagina geopend wordt nog beschikbaar.

- ◆ Zoek informatie op over deze local storage api.
- ◆ Er bestaan verschillende mogelijkheden om de gegevens op te slaan, welke zijn dit? Welke is het meest geschikt om high scores in op te slaan?
- ◆ Voeg een tabel toe aan de pagina. In deze tabel zal er een overzicht komen van wie wanneer welke score behaald heeft.
- ◆ Bij het openen van de pagina moet de naam van de gebruiker gevraagd worden (gebruik hiervoor de prompt functionaliteit van JavaScript).
- ◆ Op het einde van het spel moet de score tevoorschijn komen in de tabel, samen met de naam. Dit kan je implementeren door vanuit JavaScript de tabel aan te passen (nieuwe “tr” maken en toevoegen).

9 Uitbreidingen (optioneel)

- ◆ Voorzie verschillende soorten voedsel die verschillende punten opleveren.
- ◆ De verschillende soorten voedsel gedragen zich ook anders, sommige soorten blijven altijd op de zelfde plek, andere bewegen, anderen verdwijnen na verloop van tijd.
- ◆ Laat de gebruiker toe om de parameters van het spel aan te passen, voeg een HTML5 form toe waarin de gebruiker o.a. de snelheid van het spel kan instellen. Maak gebruik van geschikte HTML5 input velden (range, ...).
- ◆ Zorg ervoor dat de gebruiker een snapshot kan nemen van het spel door op 's' te drukken. Sla alle informatie over het spel (positie van de objecten, slang, score, ...) op in de localStorage en voorzie een knop om een bewaard spel opnieuw in te laden.