
Versie 1.0 februari 2015 Start-versie

Over dit document

In dit document probeer is duidelijk te maken hoe de spellen die nu gespeeld kunnen worden op de site tot stand zijn gekomen. Dan een uitleg over hoe en waarmee ze zijn gemaakt en wat jij kan doen om nieuwe spellen te ontwikkelen. Het kan heel goed dat de uitleg hier en daar niet goed genoeg is om er iets mee te kunnen. Als dat zo blijkt te zijn zal dit document aangepast worden. Ik zal met versienummers gaan werken zodat je kunt zien wanneer je nog met een oude versie werkt en per versie aangeven wat er is veranderd.

Stukje geschiedenis

De spellen die je nu online kunt spelen, bestaan al veel langer maar dan in de vorm van executable Windows programma's die je kunt downloaden en op je computer kunt installeren. Als je op de home-pagina van de site kijkt zie je onder het hoofdstuk "Downloaden" linkjes naar uitleg over de bestaande spellen (ABC voor taal en 123 voor rekenen) en linkjes waar je die programma's kunt downloaden. Als je daar naar kijkt zal je zien dat slechts van een paar spellen nu een online-versie is gemaakt. Het ombouwen van alle aanwezige spellen naar een online-variant is een enorme klus.

Hoe en waarmee zijn de spellen gemaakt?

Alle spellen zijn gemaakt volgens het principe van "Client side scripting" in tegenstelling tot "Server side scripting". De gebruikte taal op de client (de browser van de gebruikers) is in dit geval javascript. Zie ook http://en.wikipedia.org/wiki/Client-side_scripting. Het komt er op neer dat van de server (de site van edubas) de benodigde script-files worden geladen aangevuld met andere files zoals html en css.

Om de spellen te spelen wordt weliswaar javascript-code uitgevoerd maar er wordt gebruik gemaakt van een framework waardoor het programmeren van de spellen een stuk eenvoudiger wordt en dat is angular zie <https://angularjs.org/>. Het angular framework (bibliotheek) staat al op de server. Als je gebruik gaat maken van angular (= aanbeveling) hoef je alleen eigen javascript-code te schrijven die gebruik maakt van angular.

De javascript-code haalt zelf de data (spelinhoud) op van de server. Deze databestanden zijn json-bestanden en staan in één van de mappen op de server die bij een spel horen.

Hoe verder?

In dit document wordt aan de hand van één van de spellen uitgelegd hoe de spellen in hoofdlijn in elkaar zitten. Het is de bedoeling dat op dit concept wordt ingehaakt anders wordt het een ratjetoe aan spelvormen. Daarna wordt uitgelegd wat er al is en wat jij moet doen om een spel toe te voegen aan de site.

*Opmerking: Het huidige concept is niet in beton gegoten. Als er verbeteringen (verfraaiingen) aangebracht kunnen worden is dat te overwegen. Waar het om gaat is dat het niet per se **dit** concept moet zijn maar wel een concept waar alle spellen zich aan houden.*

Opbouw scherm

Alle spelonderdelen hebben nagenoeg dezelfde opbouw maar we nemen als voorbeeld het spel "ABC – Spelling (wat is het juiste woord)". Dat spel ziet er als volgt uit.



Uitleg over de onderdelen van dit scherm.

Linksboven

Een afbeelding die het spel illustreert.

Rechtsboven

De drie niveaুকnoppen. Met een klik op een niveaুকnop worden de gegevens van dat niveau van de server gehaald (json-bestand). Standaard is niveau 1 geselecteerd en wordt het json-bestand opgehaald van niveau 1. Daaronder een afbeelding van een niet-lachende zon. Op dezelfde plek staat een afbeelding van een lachende zon die tevoorschijn komt als de opdracht is gelukt .

Midden

In dit deel wordt het spel gespeeld. In dit voorbeeld moet je op het goede woord klikken. Als er meer ruimte nodig is om je spel te spelen, kan je de afbeelding linksboven laten vervallen en die ruimte ook benutten.

Onder

Dit is het bedieningspaneel. Met de pijltoetsen kan je naar het volgende of vorige woord. Met de stopknop ga je altijd terug naar de overzichtspagina. Helemaal in de rechterhoek staat de maker van het spel. In jouw spel staat natuurlijk jouw naam.

Ophalen data

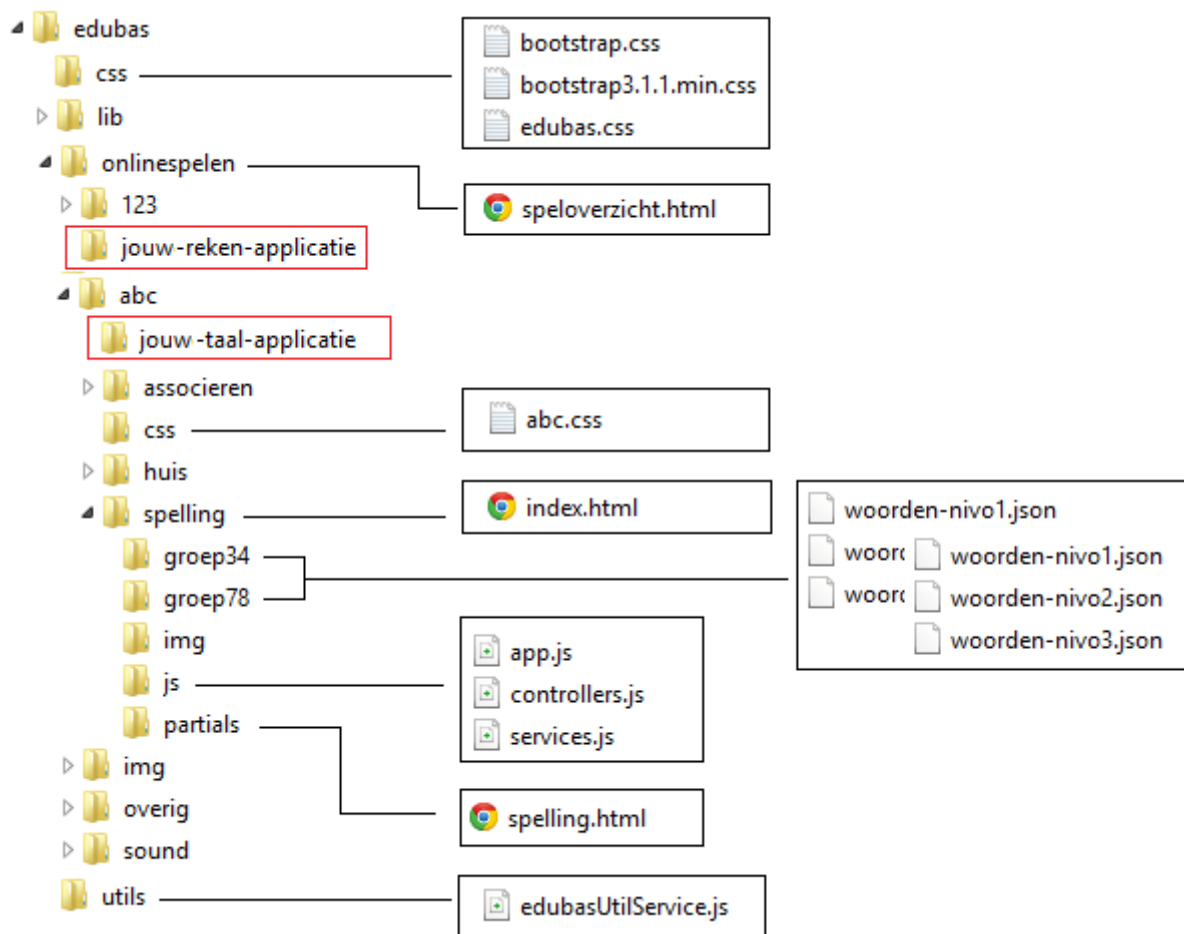
Alle spelonderdelen halen hun gegevens uit een json-bestand. Het json-bestand bevat alle “name-value pairs”, arrays of wat dan ook om het spel te kunnen spelen. Het json-bestand van een niveau wordt in zijn geheel ingelezen en in een array (van objecten) gezet. De array bevat dan de te spelen opdrachten (in dit voorbeeld de te raden woorden).

Om niet telkens dezelfde serie opdrachten voorgeschoteld te krijgen als je het onderdeel start, wordt de array “gehusselt”. Dit om te voorkomen dat je steeds dezelfde serie opdrachten te zien krijgt.

De mappenstructuur van de spellen

Hieronder een afbeelding van de mappenstructuur op de site van edubas. Aan de hand van dit plaatje leg ik uit hoe het voorbeeldspel (ABC – Spelling) in de mappenstructuur is terug te vinden en hoe jouw spellen daar in passen. Het is handig de voorbeeldzip uit te pakken, dan zie je het gelijk staan en kan je een kijkje nemen in de bestanden.

Opmerking: Het spel “ABC-Spelling” is een erg eenvoudig spel maar kan goed dienst doen als voorbeeld. Het principe blijft voor moeilijke(re) spellen hetzelfde.



edubas

Dit is de hoofdmap waar alles onder valt

edubas/css

Twee versies van angular bootstrap.css en een edubas.css

edubas/lib

Hierin staat de code van angular (versie 1.2.10) en jquery (versie 1.10.2). In het voorbeeldzipje is deze leeggelaten maar op de site is die aanwezig.

edubas/onlinespelen

In deze map staat "speloverzicht.html". Dit is een (statische) pagina met een overzicht van alle online te spelen spellen van edubas. De spellen worden hier gestart.

edubas/onlinespelen/123

Hier staan alle rekenmodules

edubas/onlinespelen/abc

Hier staan alle taalmodules. De module "spelling" wordt verder toegelicht.

[edubas/onlinespelen/abc/css](#)

In deze map is een css-bestand aanwezig bedoeld voor gebruik in alle spellen binnen abc. Je kunt in je eigen map ook een css-bestand maken die zaken “overruled”. Voor 123 is een vergelijkbaar bestand voor de spellen binnen 123.

[edubas/onlinespelen/abc/spelling](#)

In deze map staat alles wat nodig is om het spel “ABC- Spelling” te spelen. Je kunt deze opzet aanhouden voor jouw spellen. Deze zijn als voorbeeld al met een rood kader aangegeven.

In “index.html” staat eigenlijk alleen maar welke bestanden (javascript, css enz.) nodig zijn om dit spel te spelen.

[edubas/onlinespelen/abc/spelling/groep34 en groep78](#)

Dit zijn de mappen waar de data staat voor dit spel. Omdat dit spel voor twee groepen is bedoeld (groep 3/4 en groep 3/8) zijn er twee mappen met data. Bij het starten van het spel wordt meegegeven of het voor groep34 of groep78 is bedoeld zodat de data uit de juiste map wordt gehaald.

[edubas/onlinespelen/abc/spelling/img](#)

Hier staat de afbeelding die in het spel ter illustratie (linksboven) wordt gebruikt (spelling.png).

[edubas/onlinespelen/abc/spelling/js](#)

Hierin staat de javascriptcode voor het spel verdeeld over de drie bestanden app.js, controllers.js en services.js.

[edubas/onlinespelen/abc/spelling/partials](#)

In deze map staat de html van het eigenlijke spel. Omdat dit spel slechts uit één pagina bestaat, is er maar één html-bestand.

[edubas/ onlinespelen /img](#)

Hier staan afbeeldingen van spellen die in “edubasonline.html” worden getoond (150 pixels breed en 166 pixels hoog).

[edubas/ onlinespelen /overig](#)

Hierin staan de modules die niet onder taal of rekenen vallen. B.v het spel “Match”

[edubas/ onlinespelen /sound](#)

In deze map staan geluidsfragmenten. Uitspraak van letters en woorden die gebruikt worden in het spel ABC-Huis (zet de letters in de goede volgorde). Staat los van het spel ABC – Spelling maar je zou er gebruik van kunnen maken in jouw spel als je iets met geluid wilt doen.

[edubas/ utils](#)

Hier staat “edubasUtilService.js”. In dit bestand staan een aantal functies die vaak voorkomen in alle spellen. Als je zelf functies schrijft die ook nuttig zijn voor andere spellen, zet die dan daar in. Graag even markeren (met commentaar) zodat te zien is wat nieuw is.

Wat moet je opleveren voor een nieuw spel?

Om een nieuw spel toe te voegen aan de bestaande structuur moet je een paar zaken maken en aanleveren die ik op de site kan plaatsen

1. Een map met de naam van jouw spel met daarin alle code (javascript, html, css enz.) en overige spullen (data, plaatjes enz.) voor jouw spel (net zoiets als ABC –spelling). Alle angular spullen staan al op de site.
2. Een afbeelding (150 x 166 pixels) met jouw spel erop en een korte uitleg over het spel (zie in “edubasonline.html” voor voorbeelden).
3. Eventueel een nieuwe versie van “eddubasUtilService.js” als je daarin nieuwe functies hebt gemaakt die anderen ook kunnen gebruiken.

Wat doe ik met jouw aangeleverde spullen?

Ik neem jouw spel op in een testomgeving om te zien of alles werkt. Als dat zo is wordt het overgebracht naar de site en is de site een spel rijker.

Eerst uitproberen?

Wil je eerst uitproberen of dit concept gaat werken dan kan je een klein testprogrammaatje maken of een eerste conceptversie van je spel. Als die werkt, komt die ook op de site. Ik maak dan op de overzichtspagina een hoofdstuk met spellen in de maak (of zoiets) met daarop een link naar jouw spel. Kunnen de gebruikers alvast zien wat er aan komt. Als je al een naam voor je spel hebt, wordt dat getoond.