Tentamen Client Technology (101951)

Vakcode : ICT.SE.CLNT.V19(ICT.SE.CLNT.V18 en eerder) (t1)

Datum : maandag 11 mei (was: dinsdag 31 maart 2020)

Tijd : 14.30 - 16.30 uur

|  |  |  |
| --- | --- | --- |
| Klas: | Lokaal: | Aantal: |
| ICTSE2a t/m ICTSE2g, herkansers | Online: Teams | 258 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Opgesteld door : Bolt, Ernst

Docenten : BTE08

Gecontroleerd door : Swen-Peter Ekkebus, Arjen Korevaar

Rekenmachine: geen

Literatuur: **Internet**

**Niet toegestaan is het gebruik van telefoons, social media, forums, dropbox en**

**alles wat je in contact met andere personen kan brengen. Als dit toch gebeurt,**

**is er sprake van fraude en wordt dit aangegeven bij de examencommissie.**

Overige hulpmiddelen: geen

Tentamen maken op: laptop

CONTROLEER VOORAF DE VOLGENDE GEGEVENS:

Dit tentamen bevat:

4 opgaven

6 genummerde pagina's

Waarschuw Ernst Bolt via Teams als één van deze aantallen niet klopt!

# Aanwijzingen

In totaal zijn voor de vragen 8 punten te halen. Het eindcijfer wordt verkregen door de behaalde punten af te ronden naar boven en vervolgens het cijfer 4, 6 of 8 toe te kennen. Tussen de bandbreedtes wordt naar boven afgerond: dus een 5 wordt een 6. Het hoogste cijfer is een 8, het laagst te behalen cijfer is een 1 wanneer niets is ingeleverd en anders wordt er afgerond naar een 4.

**Inleverinstructie:**

1) Verwijder de map node\_modules .

2) Controleer of de code van je project in de root van de map staat. Dus

**niet:** Ictse2a ErnstBolt s1060639/studentversie/studentversie/<code>

**goed:** Ictse2a ErnstBolt s1060639/<code>

3) Lever het practicum in als zip-file op de ELO bij het inleverpunt van het practicum.

De zipfile heeft als naam je klas (a,b,c, of d), naam en studentnummer:

<klas naam studentnummer>.zip

Bijvoorbeeld:

ictse2a ErnstBolt s1060639.zip

**Instructie over package.json en dependencies**

1) Je mag **geen wijzigingen** aanbrengen in de **package.json**.

2) Het is **niet toegestaan** om **extra dependencies** te installeren.

**Startinstructie**

In de .zip waar dit toetsdocument zich in bevindt staat op de benodigde broncode als .zip.  
1) Pak het .zip-bestand met broncode uit in een lege map.

2) Installeer de packages met $ npm install.

3) Je mag **geen wijzigingen** aan brengen in de mappenstructuur van het project

4) Je mag jQuery gebruiken. Het bestand index.html bevat een link naar jQuery.

# Introductie op de applicatie

In dit practicum ontwikkel je een kleine game: op een webpagina beweegt Mario (een .png plaatje) heen en weer. Op de pagina wordt in de loop van de tijd een lijst met memes zichtbaar (een zin met een plaatje).

Zodra de gebruiker op Mario klikt wordt de score op de pagina opgehoogd en wordt er een meme aan de lijst toegevoegd.

In de eerste opdracht ontwikkel je een aantal Gulp taken om de code productieklaar te maken.  
Daarna voeg je in de tweede opdracht styling toe aan de pagina.  
In opdracht drie ontwikkel je een vendor library met de memes.  
Tenslotte ontwikkel je in de vierde opdracht de game.

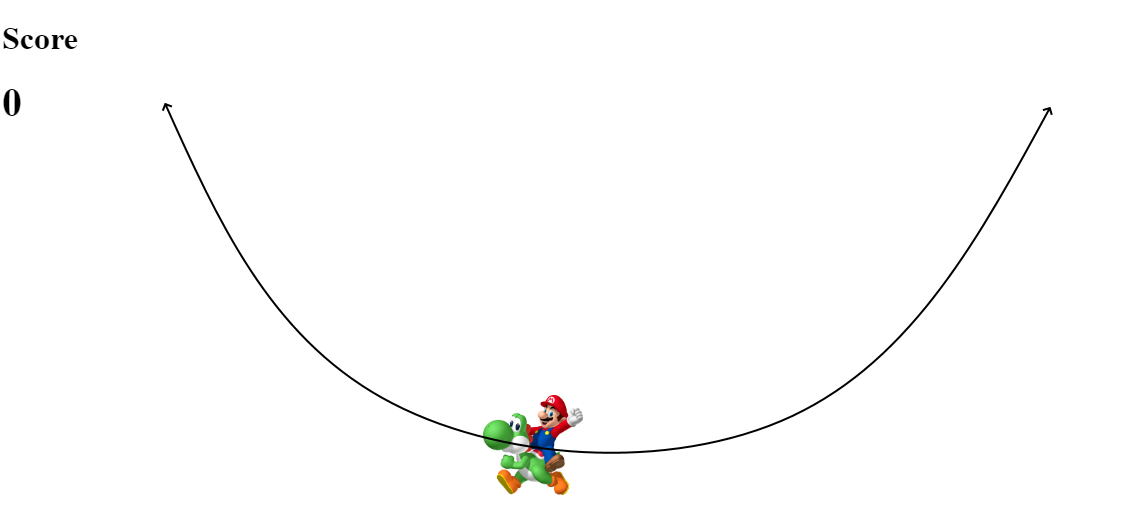
In vergelijking met de Reversi applicatie zijn de opdrachten kleiner in verband met de beperkte tijd van het practicum.

# Opgave 1 Gulp (1 punt) In deze opdracht ontwikkel je een aantal Gulp taken die allemaal als resultaat hebben dat er bestanden aan de ‘dist’ map zijn toegevoegd. Let op: het is niet toegestaan om gulpfile.js/index.js aan te passen.

In het project is een map gulpfile.js. Beschikbaar. De volgende taken moeten de bijbehorende bestanden kopiëren en/of concateneren naar de ‘dist’ map.   
Maak de taken werkend aan de hand van de onderstaande informatie.  
  
De taak ‘html’, **kopieert** het index.html bestand naar dist/index.html.  
De taak ‘image’, **kopieert** het bestand src/img/mario.png naar dist/img/mario.png.  
  
De taak ‘json’, **kopieert** het bestand vendor/memes/memes-list.json naar dist/memes/meme-list.json.

De taak ‘css’: de css bestanden van src/css/../...css en vendor/../.... .css **concateneren en wegschrijven** naar dist/style.css.  
  
De taak ‘js’: de js bestanden van src/js../...js en vendor/../...js **concateneren en wegschrijven** naar dist/app.js.

# Opgave 2 HTML & CSS (1 punt) In deze opdracht ontwikkel je een animatie en past die toe op Mario, zodat hij in een halve cirkel heen en weer beweegt:



Daarnaast wijzig je de animatie als de applicatie responsive is.

1. Ontwikkel een css animatie die ervoor zorgt dat Mario heen en weer beweegt op een denkbeeldige halve cirkel. Breid de bestaande css uit en gebruik de selectors die al aanwezig zijn.
2. Voeg responsive design toe aan Mario. Bij een smal scherm (600px) moet de animatie gewijzigd worden dat Mario alleen verticaal heen en weer gaat (van de bovenzijde van het scherm tot aan de onderkant en weer terug).

**Opgave 3 OO** (3 punten)  
In deze opdracht ontwikkel je een klasse Meme die een meme teruggeeft als daar om gevraagd wordt. Ook kan de klasse de lijst in de DOM updaten.  
De bron van de memes is een array met memes die beschikbaar is in de map.   
De klasse is al beschikbaar in het bestand vendor/memes/memes.js. De lijst met memes is beschikbaar in het bestand vendor/memes/meme-list.json.

1. De klasse Meme heeft twee properties: memeList, cursor. De memeList bevat alle memes die de klasse kan teruggeven. De cursor houdt de index bij van de meme die als eerste wordt teruggegeven. Geef de klasse Meme de properties: memeList, cursor. De cursor is na instantie 0.
2. De klasse Meme heeft een constructor met één parameter: memeList. De parameter wordt toegewezen aan de gelijknamige property.

Geef de klasse de constructor met de bijbehorende parameter en zorg ervoor dat de parameter wordt toegekend aan de property. Wanneer de ontvangen lijst memes leeg is dan moet er een error worden gegooid.

1. Implementeer een methode getNextMeme(). Deze methode geeft de volgende meme uit de property memeList terug. Wanneer de laatste meme is bereikt dan wordt de cursor weer op 0 gezet.
2. Implementeer een methode searchMeme(search). Deze methode geeft een lijst met memes terug die voldoen aan de meegegeven zoekterm. Wanneer er geen zoekresultaten zijn dan wordt een lege lijst teruggegeven.
3. Er is in het project een werkende testpagina spec/test.html aanwezig, met werkende testsuites. Gebruik deze testsuites om je testen toe te voegen. De testpagina maakt gebruik van Jasmine. Er staat een testsuite (die beginnen met de functieaanroep describe) klaar. Schrijf een test in de testsuite ’Meme’ die controleert of methoden van de klasse Meme beschikbaar (defined) zijn en of de constructor een error gooit als de parameter niet meegegeven is.

# Opgave 4 Module pattern, Promise (3 punt) In deze opdracht ontwikkel je de SPA module die de game aanstuurt en die gebruik maakt van de Meme klasse. De SPA is verantwoordelijk voor de aansturing van het geheel en heeft een property Game die ook weer een module is.

1. De SPA-applicatie bestaat uit twee modules: SPA en Game. Van de modules is alleen de SPA module in de global scope beschikbaar. Maak de twee modules volgens het module pattern.   
   Geef bij beide modules een publieke init methode.   
   Start de SPA module door in document ready in index.html SPA.init aan te roepen.  
   Start de Game module door in SPA.init de methode SPA.Game.init aan te roepen.
2. In de methode SPA.Game.init wordt een instantie van de klasse Meme geïnstantieerd. Maar eerst moet de lijst met memes worden opgehaald. Dit moet gedaan worden met een GET request.   
   Vraag met bijvoorbeeld jQuery het bestand meme-list.json op in de init. Het is toegestaan om een extra methode te maken in de module.
3. Nu de lijst met memes ontvangen is, kan de klasse Meme geïnstantieerd worden in SPA.Game.init.  
   Instantieer de klasse Meme en geeft de lijst met memes mee als argument.
4. Tijdens de initialisatie van de game wordt aan een klik op Mario een eventhandler gekoppeld. Zodra er op Mario geklikt wordt moet de score met één worden opgehoogd.  
   Maak het verhogen van de score werkend.
5. Zodra er op Mario geklikt wordt moet er ook een meme aan de lijst worden toegevoegd. Breid de eventhandler uit en maak gebruik van de instantie van de klasse Meme.
6. Schrijf een test om te controleren of na SPA.Game.init, gevolgd door een klik op Mario leidt tot een nieuwe meme in de lijst.

**Einde Tentamen**

Gebruik de inleverinstructie aan het begin van dit document