

Technisch Rapport : Kamelenrace

Matthew Demyttenaere
Ewout Merckx
Robin Geldolf

Begeleidende docent : Davy Dewinne

Inhoud

Inleiding.....	3
Meteor.js.....	3
Templates	3
Database	4
Snap.svg	5
Packages	5
Security	5,6
Structuring your application	6
Sessions.....	7
Router	7
Besluit.....	8
Bronnen	9

Inleiding

Voor het vak projecten 2 hadden we de opdracht om een multiplayer browser game te maken die gebaseerd is op het kamelenrace spel van op de kermis. Dit zouden we doen door html5,css3 en Meteor.js te gebruiken. Het doel van de game is om met kamelen tegen elkaar te racen en zo als eerste aan de finish te verschijnen. De kamelen worden vooruit gestuurd door balletjes te gooien, hoe hoger de score, hoe sneller de kamelen.

Meteor.js

Meteor.js is een open-source javascript web applicatie framework. Het gebruik het distributed data protocol(DDP) dat speciaals gemaakt is voor het meteor framework. Het protocol is een client-server protocol dat toelaat om de server up te daten en dan die updates te synchroniseren naar alle clients.

Het aanmaken van een meteor project doe je op volgende manier. Dit gaat ervoor zorgen dat een myapp project word aangemaakt en dat je ze lokaal kan testen op localhost:3000.

```
$ meteor create myapp
```

Run it locally:

```
$ cd myapp
$ meteor
=> Meteor server running on: http://localhost:3000/
```

Templates(Handlebars)

Meteor laat het toe om template talen te gebruik zoals Handlebars. Deze kan je gebruiken door een template aan te maken in je html(zie 1.). Daarna kan je de template waarde geven door de onderstaande code uit te voeren in jouw javascript.(zie2.)

```
<template name="welcomeCamel">
  <svg id="camelRunningBy">
    </svg>
</template>
```

1.

```
rendering field
+ Template.welcomeCamel.rendered = function () {...}
```

2.

Database

Meteor slaat data op in collections. Zo'n collectie maak je aan op de volgende manier.(zie 3.) De aangemaakte collecties worden opgeslaan op de mongo server.

Hier maken wij onze games en messages(chat) collecties aan.

```
Messages = new Meteor.Collection('messages');  
Games = new Meteor.Collection('games');
```

3.

Na het aanmaken van collecties kan je data toewijzen aan de collecties. Dit word gedaan op de volgende manier.(zie 4.)

Op deze manier gaan wij een game aanmaken met de nodige gegevens zoals spelers en gameld.

```
//If there are no games, create a new one with id 1  
Games.insert({  
  gameId: newGameId,  
  GameEnded: false,  
  Players: [  
    {  
      Username: $('#txtSetUsername').val(),  
      playerId: 0,  
      CurrentLocation: 0  
    },  
    {  
      playerId: 1,  
      CurrentLocation: 0  
    },  
    {  
      playerId: 2,  
      CurrentLocation: 0  
    },  
    {  
      playerId: 3,  
      CurrentLocation: 0  
    }  
  ]  
});
```

4.

Na het inserten van data in de collection is het ook mogelijk om de data weer op te vragen. Na het aanmaken van onze game moeten we de data van de game kunnen opvragen aan de hand van de gameId. Dit wordt gedaan op de volgende manier.(zie 5.)

```
Games.findOne({GameId: parseInt(Session.get("GameId"))}); 5.
```

Snap.svg

Snap.svg is een nieuwe javascript bibliotheek om te werken met scalable vector graphics(SVG). Deze svg's behouden hun kwaliteit op gelijk welke grote. Snap.svg helpt bij het animeren en plaatsen van de svg's.

Het aanmaken van een svg oppervlakte (zie 7.) en het plaatsen van svg's op deze oppervlakte (zie 8a,8b.) hebben we gedaan op de volgende manier.

```
//creating field  
s = Snap("#backgroundRaceField"); 7.
```

```
Snap.load("../img/BlueCamel.svg", onBlueCamelSVGLoaded); 8a.
```

```
function onBlueCamelSVGLoaded(f) {  
    camelBlue = s.group().transform(startLocationBlue).append(f);  
} 8b.
```

De afbeeldingen die we gebruiken in onze game zijn allemaal svg's. De kamelen hebben we zelf omgezet naar svg's met Vector Magic. Deze tool liet ons toe om gelijk welke afbeelding om te zetten naar een svg.

Packages

Meteor bestaat uit een reeks packages, deze packages zorgen dan voor functionaliteit in meteor. Standaard bevat meteor packages zoals JQuery, coffeescript en nog vele anderen. Daarnaast kan je ofwel zelf nieuwe packages schrijven of bestaande packages importeren. Zo hebben wij de snap.svg package moeten toevoegen. Zo konden wij de functionaliteiten van de snap.svg library gebruiken in ons meteor project. Het toevoegen van deze package hebben we gedaan aan de hand van meteorite(meteor tool om packages toe te voegen).

Security

Het beveiligen van onze game kan gedaan worden door regels te definiëren op de server. Deze regels regelen wat de clients kunnen/mogen doen.

Bij het aanmaken van een meteor project zit de package “insecure” er automatisch bij. Deze package laat toe om vanaf de client query’s op de server uit te voeren. Deze package mag er staan tijdens de ontwikkeling van de game maar word daarna best verwijderd.















Dankzij volgende regels kan je het toevoegen van data wijgeren.

```
// server: don't allow client to insert a party
Parties.allow({
  insert: function (userId, party) {
    return false;
  }
});

// client: this will fail
var party = { ... };
Parties.insert(party);
```

9.

Structuring your application

- ▼  **kamelenrace** (~ / Desktop / Sc
 - ▶  **.meteor**
 - ▼  **client**
 -  **clientcode.js**
 - ▼  **collections**
 -  **databases.js**
 - ▶  **packages**
 - ▼  **public**
 - ▶  **css**
 - ▶  **img**
 - ▶  **js**
 - ▶  **webfonts**
 - ▶  **research**
 - ▼  **server**
 -  **servercode.js**
 - ▼  **views**
 -  **base.html**
 -  **game.css**
 -  **game.html**
 -  **home.css**
 -  **home.html**

Onze applicatie bestaat uit volgende folders:

client: Hierin staat alle code die op de clientside wordt uitgevoerd. Het is ook mogelijk om alle clientside code per pagina te zetten(home.js,base.js,enz..)

collections: In de collections folder worden alle collecties aangemaakt, in ons geval is dit messages en games.

packages: De packages die in ons meteor project zitten kan je hiering terug vinden.

public: Alle info die public aanwezig moet zijn zet je hierin. Zoals images.

server: De code die je hier terugvind word op de serverside uitgevoerd.

views: In de views map kan je alle html en css zetten. Hierin kan je ook de js files zetten per pagina.

Sessions

In ons meteor project kunnen we gebruik maken van sessions. Dit kan gedaan worden op de volgende manier.(zie 9.)

```
//Set the sessions vars
Session.set('Username', $('#txtSetUsername').val());
Session.set('PlayerId', '0');

// ...

{{
  Session.get('Username'),
  {{: text.
}}
```

9.

Router

Voor het navigeren in ons meteor project gebruiken we router. Router is een package die we zoals snap.svg via meteorite hebben toegevoegd aan ons project. Router werkt op de volgende manier.(zie10.) Hier maken we de mogelijke routes aan.

```
//Routing
Meteor.Router.add({
  '/': 'home',

  '/game/:_id': { to: 'game', and: function (id) {
    Session.set('GameId', id);
  }},
  '*': 'not_found'
});
```

10.

We kunnen de routing ook gebruiken om te navigeren naar diverse pagina's. Zo kan je hieronder zien dat we bij het drukken op de new game button(testing2) we worden geredirect naar de home pagina.(zie 11.)

```
testing2.click(function () {
  Meteor.Router.to('/');
});
```

11.

Besluit

Onze multiplayer browser game is aan de hand van Meteor.js en al zijn functionaliteiten zoals packages(router;snap.svg),templates en data handling(mongoDb) tot stand gekomen. We hebben het mogelijk gemaakt om een homepage weer te geven met passende animaties, die je ook toelaat om een username te kiezen. Daarna kan je op de gamepagina aan de hand van het gooien van een balletje racen tegen andere spelers online. De winnaars krijgen een boodschap die ze meedeelt dat ze hebben gewonnen en de verliezers een verliezers boodschap. Daarna krijgen ze de keuze op het spel opnieuw te spelen.

Het is ook mogelijk om tijdens de game met elkaar te chatten.

Bronnen

<http://snapsvg.io/>

<https://www.meteor.com/>

<http://docs.meteor.com/>

http://nl.wikipedia.org/wiki/Scalable_Vector_Graphics

<https://github.com/oortcloud/meteorite>

<https://groups.google.com/forum/#!topic/snapsvg/>

<http://raphaeljs.com/>