Jeroen Rothbauer
1532209
18 mei 2011
Versie **1.2**

Hamaka
digitaal voor oplossing

Nebula

Platformonafhankelijk mobiel gamen

Auteur Jeroen Rothbauer
Kleine Koppel 24-13
3812 PH Amersfoort
Tel. 06 28884699

Studentnummer 1532209

Bedrijfbegeleider Jochem Bronzewijker
Hamaka B.V.
Goeman Borgesiuslaan 77
3515 ET Utrecht
Tel. 030 8903232

Eerste examinator Patrick Ubags
Hogeschool Utrecht, Faculteit Natuur en Techniek
Opleiding Mediatechnologie
Oudenoord 700
3513 EX Utrecht
Tel. 030 2308202

Versie 1.2
18-05-2011

Samenvatting

Het bedrijf Hamaka is geïnteresseerd in de nieuwe webtechnieken die onder de naam HTML5 vallen en willen hier meer kennis en ervaring over opdoen en in het bijzonder voor gebruik op mobiele apparaten. Om dit mogelijk te maken is onderzoek gedaan naar welke mobiele platformen er op dit moment populair zijn in Nederland en wat de mogelijkheden zijn voor de HTML5 technieken op deze platformen. En er is een webapplicatie ontwikkeld die gericht is op mobiele apparaten en gebruik maakt van deze technieken.

Uit het onderzoek is gebleken dat de Android en iOS (iPhone, iPod Touch, iPad) platformen de populairste zijn in Nederland en ondersteuning bieden aan veel verschillende HTML5 technieken. Ook blijkt dat de specificaties van verschillende mobiele apparaten enorm van elkaar kunnen verschillen, waardoor er buiten de verschillende platformen ook rekening gehouden moet worden met verschillende hardware beperkingen en schermresoluties. Deze laatste zijn echter wel deels gestandaardiseerd waardoor het mogelijk is om een webapplicatie te schalen aan de hand van de resolutie van de gebruiker.

Voor de webapplicatie is gekozen voor een gameportaal voor mobiele apparaten waarmee games gespeeld kunnen worden zonder dat een aparte browser-plugin nodig is. De drie games die hierbij gemaakt worden maken, net als het portaal, gebruik van verschillende HTML5 technieken. Het ontwikkelen van meerdere games biedt de mogelijkheid om verschillende nieuwe technieken te gebruiken en hiermee te experimenteren.

Zowel het gameportaal als de games moeten gebruiksvriendelijk zijn zodat er minimale uitleg op het scherm getoond hoeft te worden en zo het kleine scherm van een mobiel apparaat optimaal benut wordt.

De games kunnen met het portaal communiceren om zo gegeven over de ingelogde gebruiker op te vragen of om bijvoorbeeld een highscore op te slaan. Deze communicatie zal worden beveiligd om vals spel te voorkomen, zo zal een gevoelige aanroep bijvoorbeeld nooit twee keer hetzelfde zijn, maar kan door de aard van de gebruikte technieken nooit volledig ondoordringbaar zijn.

Op moment van schrijven is het portaal en één van de games ontwikkeld en bevinden deze zich in de testfase. De overige twee games zijn momenteel in ontwikkeling.

Inhoudsopgave

Voorwoord

Deze scriptie is een verslag van mijn afstudeerproject voor de opleiding Mediatechnologie aan de Hogeschool Utrecht. Op de site Stagemotor.nl vond ik een vacature voor een afstudeerstage bij het Utrechtse webbureau Hamaka. Na hierop gereageerd te hebben werd ik uitgenodigd voor een gesprek en heb ik samen met de twee eigenaren van Hamaka, beide ex-Mediatechnologie studenten, een afstudeeropdracht geformuleerd.

De doelgroep van deze scriptie bestaat uit mensen die inzicht willen krijgen in de ontwikkelingen van nieuwe webtechnologieën en de toepassing hiervan op een mobiel platform. Daarbij is deze scriptie bedoeld voor gebruik bij het beoordelen van mijn afstudeertraject.

Graag wil ik een aantal mensen bedanken die hebben bijgedragen aan de realisatie van dit project. Mijn afstudeerdocent Patrick Ubags voor de begeleiding vanuit school, mijn bedrijfsbegeleider Jochem Bronzewijker en zijn compagnon Bauke Zwaan voor inhoudelijke ondersteuning en begeleiding, en William Parr en Chris Simonse voor hun ideeën en hulp bij het testen van de producten.

Utrecht, 12 mei 2011

Jeroen Rothbauer

Alle afkortingen en begrippen in deze scriptie worden de eerste keer *cursief* weergegeven en zijn terug te vinden in het hoofdstuk “Verklarende woordenlijst”.

Inleiding

Het bedrijf Hamaka is geïnteresseerd in de grafische en technische mogelijkheden van webtoepassingen ontwikkeld op basis van *HTML5* en hoe deze zich verhouden onder mobiele apparaten.

Als afstudeerstudent Mediatechnologie aan de Hogeschool Utrecht ben ik aangenomen om voor Hamaka onderzoek te doen naar de mogelijkheden en beperkingen van het gebruikt van nieuwe webtechnieken, die onder de HTML5 term vallen, voor mobiele apparaten. De opdracht die hieruit is voortgekomen is het ontwikkelen van een mobiel gameportaal die gebruik maakt van HTML5 standaarden en de mogelijkheid biedt om games, ontwikkeld met HTML5 technieken, te spelen. Daarbij worden er ook drie van deze games ontwikkeld om een invulling te geven aan het platform en meer inzicht te krijgen in de technieken.

Deze opdracht is gekozen, deels vanwege mijn achtergrond in gameontwikkeling bij het keuzesemester van de opleiding, en deels omdat het ontwikkelen van meerdere games de mogelijkheid biedt om meerdere technieken effectief te kunnen gebruiken.

Deze scriptie wordt opgeleverd voor het einde van de stageperiode waardoor het project nog niet voltooid is op moment van schrijven. De overgebleven tijd zal voornamelijk besteed worden aan de realisatie van de laatste twee games en het verder testen van het gameportaal. Tijdens de afstudeerzitting zal deze fase verantwoord worden.

1 Hamaka

Hamaka is een creatief webbureau gericht op het creëren van websites en webapplicaties die eenvoudig te gebruiken en onderhouden zijn. Het bedrijf telt vijf medewerkers, allen met een technische achtergrond. De twee oprichters zijn afgestudeerd aan de Hogeschool Utrecht, opleiding Mediatechnologie.

Mijn rol binnen Hamaka is onderzoek doen naar de mogelijkheden die HTML5 bied voor mobiele platformen en hier praktische ervaring mee op doen. Hamaka wil hiermee de interne kennis op dit gebeid vergroten zodat deze in latere projecten toegepast kan worden.

2 Organisatie en werkwijze

In dit hoofdstuk wordt de organisatie en werkwijze met betrekken op opdracht beschreven.

2.1 Organisatie

De opdracht is intern voor Hamaka en wordt volledig door mij uitgevoerd. Jochem Bronzewijker, mijn bedrijfsbegeleider, treedt op als de interne opdrachtgever. Bauke Zwaan, medeoprichter van Hamaka, biedt inhoudelijke hulp en advies waar nodig.

Bij brainstormsessies en feedbackrondes is iedereen in het bedrijf betrokken om beschikking te krijgen over zo veel mogelijk verschillende meningen en ervaringen.

2.2 Werkwijze

Er wordt fulltime aan de opdracht gewerkt, wanneer niet aan deze scriptie wordt gewerkt. Wekelijks worden nieuwe voortgangen medegedeeld aan de rest van het bedrijf tijdens het weekoverleg. Verder wordt periodiek een demonstratie gegeven van de (deel)producten en de technieken die gebruikt zijn om deze te maken.

Voorafgaand aan het opstellen van een functioneel ontwerp wordt een brainstormsessie gehouden waarbij iedereen die beschikbaar is in het bedrijf aanwezig is. De ideeën die hierbij naar voren komen worden vervolgens door mij verwerkt in het functioneel ontwerp.

Wanneer een (deel)product voldoende operationeel is volgt een feedbackronde waarbij iedereen binnen Hamaka de mogelijkheid krijgt om het (deel)product uit te proberen en te testen. Ervaringen en bugs die hieruit naar voren komen worden door mij verzameld, geëvalueerd en mogelijk verwerkt in het (deel)product.

3 Opdracht

In dit hoofdstuk wordt duidelijk gemaakt wat de opdracht inhoudt en hoe deze is ontstaan.

3.1 Probleemstelling

Het spelen van games op mobiele apparaten zoals smartphones of tablets is ontzettend populair geworden, zo blijkt onder andere uit de lijst met meest gedownloade *apps* voor Apple's iOS waarvan de top 10 op moment van schrijven volledig bestaat uit games[1]. Daarnaast bestaat het app aanbod van de populairste mobiele besturingssystemen voor ongeveer 20% uit games.[2]

Om een game voor een mobiel apparaat te ontwikkelen zal de developer rekening moeten houden met de verschillende eigenschappen van het apparaat te denken aan hard- en software. Zo zijn er veel verschillende mobiele besturingssystemen, waaronder iOS, Android en Windows Phone, die allen een andere manier van programmeren vereisen.

Een game ontwikkelen voor elk van de diverse besturingssystemen is een grote investering, maar om het niet te doen zou betekenen dat men een aanzienlijke markt laat liggen. De meeste moderne besturingssystemen hebben echter een webbrowser met HTML5 ondersteuning tot hun beschikking. Dit maakt het mogelijk om games uit te voeren in de webbrowser zonder hulp van een externe plug-in of applicatie. Dit is belangrijk aangezien minstens één populair mobiel besturingssysteem, namelijk iOS, geen ondersteuning biedt aan externe plug-ins die de mogelijkheid bieden om games te draaien op eenzelfde mannier als dit op andere platformen mogelijk is. Alternatieven als Flash en Silverlight vallen dus bij voorbaat af.

Wetend dat HTML5 grotendeels besturingssysteem- en apparaatonafhankelijk is, blijft de vraag hoe en met welke technieken een uitdagende game ontwikkeld kan worden.

3.2 Doelstellingen

Het hoofddoel van het project is om een eenvoudige manier te ontwikkelen om games op meerdere mobiele besturingssystemen te laten functioneren en deze intuïtief bereikbaar te maken voor de eindgebruiker door middel van een browseronafhankelijk webportaal.

Het doel van Hamaka is kennis en ervaring op doen op het gebied van mobiele webtoepassingen. Aangezien mobiel internet steeds vaker gebruikt wordt is het belangrijk voor een ontwikkelbureau om hier intern ervaring mee op te doen zodat deze kennis later voor meerdere doeleinden gebruikt kan worden.

3.3 Hoofdvraag en deelvragen

Hoe kan een gamedeveloper, met kennis van HTML5, zo eenvoudig mogelijk, dus zonder voor ieder platform een aparte versie te moeten maken en met zo min mogelijk repetitieve code, zijn game beschikbaar maken op zoveel mogelijk mobiele besturingssystemen tegelijk op een manier dat deze game eenvoudig te vinden is door de eindgebruiker?

De deelvragen zijn:

- Wat zijn de mogelijkheden van HTML5 met betrekking tot games en hoe kunnen deze zo optimaal mogelijk worden benut?

[1] Bron: http://www.apple.com/itunes/charts/paid-apps/

[2] Bronnen: http://www.androlib.com/appstatstype.aspx (Android)
http://wp7applist.com/stats/ (Windows Phone 7)
http://148apps.biz/app-store-metrics/?mpage=catcount (iOS)

- Hoe ontwikkelen we een duidelijke en browseronafhankelijke gebruikersinterface waarmee eenvoudig genavigeerd kan worden en aansluit bij de verwachtingen van de gebruiker?
- Is het mogelijk om veelvoorkomende programmeeronderdelen beschikbaar te stellen aan alle games binnen het portaal om het ontwikkelen te vereenvoudigen?
- Hoe presenteren we de games aan de eindgebruiker zonder dat deze van te voren van het bestaan van die specifieke game af hoeft te weten?

3.4 Analyse

Om de deelvraag met betrekking tot de mogelijkheden van HTML5 te beantwoorden is een onderzoek gedaan waarvan de resultaten terug te lezen zijn in het hoofdstuk “Onderzoek”.

Door het gebruik van HTML5 hebben we de mogelijkheid om een gebruikersinterface en games te ontwikkelen die onafhankelijk zijn van de browser en dus het platform waarop deze worden weergegeven. Hoe op een eenvoudige mannier door de interface genavigeerd kan worden wordt beschreven in het hoofdstuk “Functioneel Ontwerp”.

In de derde deelvraag wordt gevraagd of het mogelijk is om veelvoorkomende programmeeronderdelen aan alle games beschikbaar te stellen. Dit is te realiseren door met het portaal een *framework* te ontwikkelen waar de games toegang tot hebben. Hierin kunnen dan functies worden geschreven die door de games met een enkele regel aan te roepen zijn waardoor deze niet keer op keer opnieuw geschreven hoeven te worden.

Door alle games in een centrale weergave te plaatsen zal de eindgebruiker worden gewezen op het bestaan van deze games. Hierdoor hoeft de gebruiker niet eerst te zoeken op de naam van een game voordat deze gevonden wordt. Hoe de games precies gepresenteerd worden wordt beschreven in het hoofdstuk “Functioneel Ontwerp”.

Het resultaat van de antwoorden op de deelvragen vormen samen het antwoord op de hoofdvraag en zijn de hoofdelementen in deze opdracht.

3.5 Eisen en randvoorwaarden

Aan de opdracht zijn enkele eisen en randvoorwaarden verbonden.

Eisen

- Er moeten zoveel mogelijk nieuwe webtechnieken worden betrokken bij het ontwikkelen van zowel het portaal als de games zodat hier ervaring mee wordt opgedaan.
- Het portaal en de games dienen te functioneren op minstens twee verschillende populaire mobiele platformen met HTML5 ondersteuning.
- Zowel het portaal als de games moeten zonder uitgebreide uitleg door de gebruiker te begrijpen zijn.
- Het aanmaken van een account mag niet verplicht zijn om games te kunnen spelen.

Randvoorwaarden

- Het project moet voldoen aan de vier competenties van de Bachelor of Engineering, en de vijf Dublin descriptoren moeten in het proces worden betrokken om te voldoen aan het hbo-niveau. Daarnaast moet erop worden toegezien dat raakvlakken worden behouden met de Mediatechnologie-gebieden.
- Het portaal en de games moeten binnen de vastgestelde stageperiode in een functionele staat worden opgeleverd.

3.6 Op te leveren producten

- Functioneel ontwerp
- Grafisch ontwerp met een intuïtieve gebruikersinterface
- Technisch ontwerp
- Werkend prototype van het portaal inclusief framework
- Ten minste 3 simpele, maar volledig functionele, games die als voorbeeld zullen dienen voor nieuwe toevoegingen, en als eerste invulling van het portaal zullen dienen

4 Proces

In dit hoofdstuk wordt beschreven hoe het project is aangepakt en gepland. Ook wordt geëvalueerd hoe dit proces uiteindelijk is verlopen.

4.1 Projectaanpak

De opdracht bestaat uit een aantal verschillende onderdelen, deze onderdelen worden uitgevoerd in de volgorde zoals afgebeeld in *Figuur 4.1*.

Figuur 4.1 – Overzicht Proces

Onderzoek
Er is begonnen met het onderzoek naar de mogelijkheden van HTML5 en hoe deze toepasbaar zijn in deze opdracht. Hierbij zijn zoveel mogelijk verschillende technieken betrokken om zo breed mogelijk de interne kennis van HTML5 te vergroten.

Omdat het portaal en de games in verschillende browsers moeten draaien en de meeste browsers niet alle onderdelen van HTML5 ondersteunen is er onderzocht welke browsers veel gebruikt worden op mobiele platformen en welke gedeelten van HTML5 deze ondersteunen.

Daarnaast is er onderzoek gedaan naar verschillende populaire mobiele platformen om erachter te komen in hoeverre er rekening gehouden moet worden met verschillende hardware specificaties. Hierin is bekeken hoe de webapplicaties het beste in verschillende resoluties kunnen worden weergeven en wat de limitaties zijn qua CPU en GPU snelheid of geheugen.

Eerste opzet Portaal
Na het onderzoek is er begonnen met de ontwikkeling van het portaal. Dit begint met de ontwerpfase waarin een functioneel, technisch en grafisch ontwerp wordt gemaakt waarin wordt beschreven of getoond hoe het portaal gaat werken, hoe het gemaakt gaat worden en hoe het eruit komt te zien.

Daarna is de eerste opzet gemaakt voor de server-kant waarbij een database wordt opgezet en ook de communicatie tussen de server en de database wordt ontwikkeld.

Vervolgens is de interactieve gebruikersinterface gemaakt op basis van het grafisch ontwerp. Daarbij wordt ook de bijbehorende code op de server gemaakt om dynamische functionaliteiten mogelijk te maken.

Tot slot is het framework ontwikkeld waarmee games veelvoorkomende acties eenvoudig kunnen uitvoeren. Bijvoorbeeld het opvragen van de schermoriëntatie van de gebruiker. Ook kan met de bijbehorende *API* data worden gestuurd naar, of opgehaald van de server. Dit kan bijvoorbeeld gebruikt worden voor een highscore lijst.

Games en doorontwikkeling Portaal
Na de eerste opzet van het portaal is er begonnen met het ontwikkelen van de games. Tijdens dit proces zijn naar verwachting dingen tegenkomen die nog nodig of gewenst zijn in het portaal. Deze elementen zijn daarop meteen geïmplementeerd in het portaal.

Er is eerst voor de eerste game een functioneel, technisch en grafisch ontwerp gemaakt, waarna de eerste game op basis hiervan ontwikkeld is. Vervolgens zijn voor de overige twee games de ontwerpen gemaakt. Hierna worden die games achter elkaar aan de hand van deze ontwerpen ontwikkeld. Dit om te zorgen dat alle documenten voltooid zijn voor deze scriptie.

Afronden Portaal en Testen
Wanneer de games gereed zijn kan het portaal worden afgemaakt zodat het een stabiel product wordt. Daarna worden het portaal en de games grondig getest en waar nodig verbeterd. Het testen bestaat uit het uitvoeren van iedere functionaliteit van het portaal en de games op ten minste één iPhone en één Android aparaat omdat dit twee van de meest populaire mobiele platformen zijn (welke HTML 5 ondersteunen) en deze binnen Hamaka beschikbaar zijn. Daarnaast wordt gebruik gemaakt van de Android Software Development Kit om meerdere Android configuraties te emuleren. Eventuele problemen die tijdens het testen optreden worden gedocumenteerd en opgelost waarna een nieuwe testronde start. Tot slot zal een medewerker van Hamaka die het systeem nog niet eerder gebruikt heeft alles doorlopen om te controleren of alles naar de verwachting van een nieuwe gebruiker functioneert.

4.2 Planning

Tijdens het project is er gebruik gemaakt van de planning uit Figuur 4.2 waarin per kalenderweek weergegeven wordt aan welk onderdeel van het project gewerkt wordt.

Week	**Activiteit**
05	Start Stage Schrijven Plan van aanpak
06	Onderzoek
07-08	Portaal - Functioneel ontwerp - Technisch ontwerp - Grafisch ontwerp
08-09	Portaal - Eerste opzet server/database
09-11	Portaal - Eerste opzet interactieve gebruikersinterface
11-12	Portaal - Eerste opzet game-API
12-13	Game #1 - Functioneel ontwerp - Technisch ontwerp - Grafisch ontwerp
13-14	Game #2 - Functioneel ontwerp - Technisch ontwerp - Grafisch ontwerp Game #3 - Functioneel ontwerp - Technisch ontwerp - Grafisch ontwerp
14-17	Ontwikkelen Game #1
18-20	Ontwikkelen Game #2
19	Scriptie opsturen naar afstudeerdocent voor goedkeuring
21	Inleveren scriptie
21-23	Ontwikkelen Game #3

Week	Activiteit
23-24	Afronden portaal en testen
24	Laatste stageweek
25/26	Examenzitting

Figuur 4.2 - Planning

4.3 Projectevaluatie

Het project is grotendeels volgens plan verlopen. Er is eerst een onderzoek gedaan waarin de mogelijkheden en grenzen voor het project zijn vastgesteld. Van daaruit is na een brainstormsessie een functioneel ontwerp opgesteld voor het gameportaal. Hierop zijn een aantal aanpassingen aan functionaliteiten en interfacebeslissingen besproken en waar nodig aangepast. Vervolgens is vanuit het uiteindelijke functioneel ontwerp een technisch ontwerp gemaakt.

Er is een grafisch ontwerp gemaakt die ter goedkeuring is gesteld bij Hamaka. Vervolgens is de database opgebouwd en de code op de server om hiermee te communiceren. Hierna is alle servercode geschreven die ervoor zorgt dat bij een aanvraag van de gebruiker de juiste gegevens worden teruggegeven. Vanwege discussies over het grafische ontwerp is eerst het framework en de API voor de games opgebouwd, waarna het grafische ontwerp is aangepast aan de wensen van Hamaka en is omgezet naar een interactieve interface.

Er is vervolgens een brainstormsessie gehouden om tot een idee te komen voor de eerste game. Hierna is voor die game een functioneel, technisch en grafisch ontwerp gemaakt. Er is besloten om in plaats van eerst dit proces te herhalen voor de tweede en derde game, eerst de eerste game te ontwikkelen zodat er ontwikkelervaring is voordat volgende games uitgedacht worden.

Tijdens de ontwikkeling van de eerste game is ook verder gewerkt aan het protaal, voornamelijk aan het framewok en de API. Na de ontwikkeling van de game volgde een feedbackronde voor zowel het portaal en de game waarna enkele van de aangedragen punten zijn verwerkt in de producten.

Er volgde een brainstormsessie waarin ideeën voor de overige twee games zijn aangedragen. Op basis hiervan zijn voor beide games een functioneel, technisch en grafisch ontwerp gemaakt.

Hierna is er begonnen met het achtereenvolgend ontwikkelen van de tweede en derde game.

5 Onderzoek

In dit hoofdstuk wordt onderzocht wat HTML5 precies inhoud en welke technieken hiervan interessant zijn voor gebruik in dit project. Daarbij wordt er gekeken welke mobiele platformen veel gebruikt zijn en in hoeverre deze de HTML5 technieken ondersteunen. Vervolgens wordt gekeken in of er sprake is van verschillen in hardwarespecificaties en resoluties en in hoeverre dit effect heeft op het project.

5.1 Wat is HTML5

HTML5 wordt steeds meer gebruikt als verzamelterm voor verschillende nieuwe webtechnieken. Hieronder valt vanzelfsprekend HTML versie 5 zelf wat onder andere nieuwe elementen biedt om webdocumenten gestructureerd te kunnen opbouwen. Daarnaast bevat het enkele nieuwe functionaliteiten zoals de canvas, waarmee een leeg vlak wordt geplaatst waar met JavaScript elementen mee gevisualiseerd kunnen worden. Een ander voorbeeld zijn de audio- en videotag waarmee respectievelijk audio en video kan worden afgespeeld zonder dat een externe plugin als Flash nodig is.
Daarnaast worden er vaak nieuwe JavaScript APIs bij de HTML5 term gerekend. Een voorbeeld is bijvoorbeeld de Geolocation API waarmee de positie van een gebruiker in lengte en breedte graden kan worden opgevraagd.
Tot slot worden de versie 3 van CSS en enkele ondersteunende technieken zoals gebruik van SVG (Scalable Vector Graphics) bestanden tot de HTML5 groep gerekend.

Ondersteuning voor de verschillende HTML5 technieken verschilt per browser. Om te controleren of een browser ondersteuning biedt voor een bepaalde techniek kan gebruik worden gemaakt van Modernizr. Modernizr is een script dat test voor ondersteuning van de meest interessante en bruikbare nieuwe webtechnieken die op het moment beschikbaar zijn en geeft op hun site een duidelijk overzicht hiervan. Op basis van dit overzicht en verschillende artikelen van voornamelijk het *W3C* en de website “Dive Into HTML5” is uitgezocht welke van deze nieuwe technieken interessant zijn voor gebruik in deze opdracht. De technieken die hieruit zijn gekomen staan hieronder met de reden waarom de betreffende techniek voor deze opdracht interessant is.

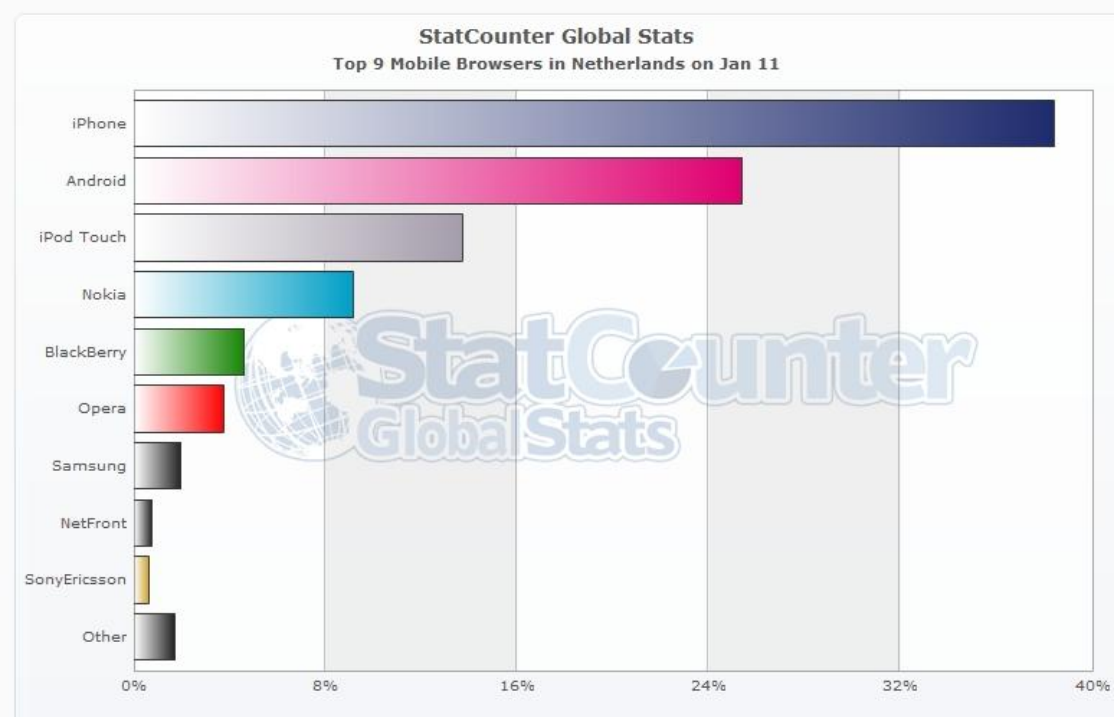
Bruikbare technieken

- **Canvas/Canvas Text**
  Het canvas maakt het mogelijk om met JavaScript op het scherm te tekenen en is dus ideaal voor het weergeven van elementen voor games.
- **HTML5 Audio**
  Als er muziek of geluidseffecten gebruikt dienen te worden in de games is er een mannier nodig om deze af te spelen zonder het gebruik van externe plugins. Het Audio element maakt dat mogelijk.
- **Touch events**
  Doordat mobiele apparaten beschikken over een touchscreen zullen de games bestuurd worden door middel van aanrakingen. Het gebruik van de nieuwe touch events maakt het mogelijk om de besturing natuurlijker aan te laten voelen.
- **SVG**
  SVG of Scalable Vector Graphics maken het mogelijk om vector afbeeldingen weer te geven binnen een website waardoor het schalen van de website geen groot probleem is, doordat er geen kwaliteitsverlies optreedt.
- **Geolocation API**
  Deze API maakt het mogelijk om de positie van de gebruiker, en de verandering daarin op te vragen als de gebruiker dat toestaat. Dit kan gebruikt worden als gameplay element of voor een gelokaliseerde dienst.

- **Web Sockets**
  Dit maakt directe communicatie tussen verschillende gebruikers mogelijk en kan gebruikt worden om multiplayer games mogelijk te maken.
- **Local-/session storage**
  Hiermee wordt een client-side alternatief geboden voor cookies. Waardoor er lokaal informatie opgeslagen en gelezen kan worden zonder dat eerst met een server contact opgenomen hoeft te worden.
- **ApplicationCache**
  Het kan mogelijk gemaakt worden om een game op het apparaat van een gebruiker op te slaan zodat deze gespeeld kan worden zonder dat er nog eens verbinding met internet nodig is.
- **Web SQL Database**
  Hierbij geldt hetzelfde als bij applicationCache alleen gaat het hier om het lokaal opslaan en opvragen van gestructureerde data door middel van *SQL* query's zonder dat een internet verbinding benodigd is.
- **DeviceOrientation**
  Dit maakt het mogelijk om de oriëntatie van een apparaat over de drie assen op te vragen zodat het mogelijk wordt om bewegingsgevoelige games te maken.
- ***Input fields***
  De nieuwe mogelijkheden van invoervelden kunnen in het portal bijvoorbeeld gebruikt worden voor het registratieformulier.
    - **Placeholder attribute**
      Om een eventueel formulier completer te maken kan dit gebruikt worden om lege velden een voorbeeld inhoud te geven.
    - **E-mail field type**
      Dit is in feite een gewoon tekstveld, maar het maakt aan de browser kenbaar dat het een e-mail vel d is en die kan zich daar op aanpassen.
    - **Form validation**
      Hierdoor hoeft er geen extra code geschreven te worden om te controleren of alle gewenste velden ingevuld zijn en of er geldige waarden in staan.
- ***CSS3***
  Al deze onderdelen maken het mogelijk om pagina's mooier op te maken of dit op een makkelijkere mannier te doen.
    - *Animatie*
        - **Transitions**
        - **Animations**
    - *Opmaak*
        - **2D Transforms**
        - **Reflections**
        - **Text Shadow**
        - **Box Shadow**
        - **Opacity**
    - *Pagina opbouw*
        - **Columns**
        - **Flexible Box Model**

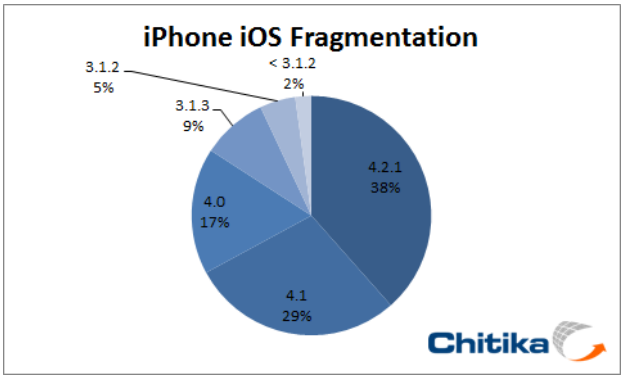
5.2 HTML5 browser ondersteuning

Omdat het allemaal nieuwe technieken betreft worden deze niet ondersteund door alle browsers. We zullen dus moeten testen welke functionaliteiten bruikbaar zijn voor de browsers waar wij ons op richten.
In Figuur 5.1, een diagram van statistiekensite StatCounter, is te zien dat de standaard-browsers van iOS (iPhone/iPod Touch) en Android verreweg het populairst zijn in Nederland. Dit zullen dan ook de browsers zijn waarop wij ons zullen richten.



Figuur 5.1 - Mobiel browser-gebruik in Nederland

Doordat de laatste update voor iedereen beschikbaar is heeft bijna iedere iPhone of iPod Touch gebruiker iOS versie 4.0 of hoger zoals ook te zien in het diagram van online-adverteerder Chitika in Figuur 5.2. Er hoeft dus niet voor veel verschillende versies getest te worden. Bij Android ligt dit anders en is niet altijd de laatste versie voor alle gebruikers van het platform beschikbaar. Hierdoor zal er op verschillende versies getest moeten worden.



Figuur 5.2 - Versiefragmentatie van iOS over januari 2011

Waar mogelijk wordt tijdens het testen gebruik gemaakt van een fysiek apparaat, anders gebeurt dit met behulp van een emulator.

Voor iedere geselecteerde functionaliteit is een bijbehorend functioneel voorbeeld gezocht zodat gekeken kan worden of deze functionaliteit naar verwachting werkt. In bijlage 1 is een overzicht te vinden waarin voor iedere geselecteerde functionaliteit de bijbehorende test staat. Daarnaast worden de tests van Modernizr als referentie gebruikt.

Hieronder volgen de resultaten met per versie per browser aangegeven of een functionaliteit naar verwachting werkt en een conclusie van deze resultaten.

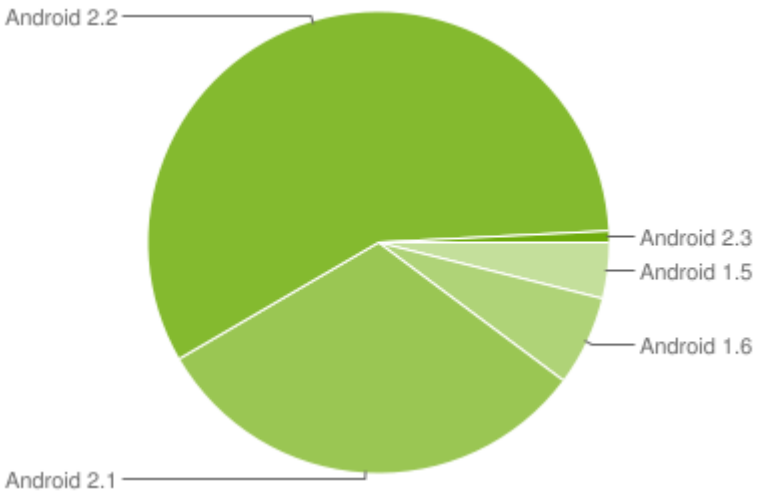
Groen	betekent dat de functionaliteit volledig naar verwachting werkt.
Oranje	betekent dat de functionaliteit gedeeltelijk werkt.
Rood	betekent dat de functionaliteit ontbreekt.

	Android Web browser				iOS Safari	
	1.6*	2.1*	2.2	2.3*	4.0	4.2
Canvas/Canvas Text						
HTML5 Audio						
Touch events						
SVG						
Geolocation API						
Web Sockets						
Local-/session storage						
ApplicationCache						
Web SQL Database						
DeviceOrientation						
Input fields						
Placeholder attribute						
E-mail field type						
Form validation						
CSS3						
Transitions						
Animations						
2D Transforms						
Reflections						
Columns						
Text Shadow						
Box Shadow						
Opacity						
Flexible Box Model						

* *Getest met emulator*

Hieruit blijkt dat de versie van het besturingssysteem / de webbrowser veel invloed heeft op de beschikbare functionaliteiten. We zullen dus een keuze moeten maken voor welke versies we willen ontwikkelen. Voor alle Nederlandse iOS gebruikers is de nieuwste versie, op moment van schrijven versie 4.2.1, van het besturingssysteem en de browser beschikbaar en in veel gevallen al in gebruik. Het is voor ons dus mogelijk om op een recente versie te richten.

Bij Android apparaten is het mogelijk dat er niet naar de laatste versie geüpdate kan worden waardoor het nodig is te kijken naar het gebruik per versie. Figuur 5.3 laat zien welk percentage gebruikers met een bepaalde versie gebruik hebben gemaakt van een centrale applicatie binnen het Android besturingssysteem, de “Android Market”, over een periode van 2 weken eindigend op 2 februari 2011. Hieruit kunnen we



Figuur 5.3 - Android versie gebruik

dus opmaken dat verreweg het grootste gedeelte van de Android gebruikers gebruik maakt van versie 2.1 of 2.2, en dit zal later vermoedelijk steeds meer verschuiven naar 2.3 of latere versies. Gezien onze testresultaten van versie 2.1 en 2.2 hetzelfde zijn, zullen we deze versies gebruiken als basis voor de ontwikkeling op en het testen van het Android platform. Versie 1.6, de laatste versie voor 2.0/2.1, beschikt over te weinig van de geselecteerde functionaliteiten om te ondersteunen en zal dus ook niet meer in het ontwikkel- of test-proces worden betrokken.

Uit de testresultaten blijkt dat enkele functionaliteiten niet aanwezig zijn op een aantal of alle geteste versies die we willen ondersteunen. Dit zijn HTML5 Audio, SVG, Web Sockets, DeviceOrientation en Form validation. Als gevolg hiervan zullen deze functionaliteiten niet gebruikt worden om compatibiliteit te behouden en zo een zo groot mogelijk bereik te krijgen.

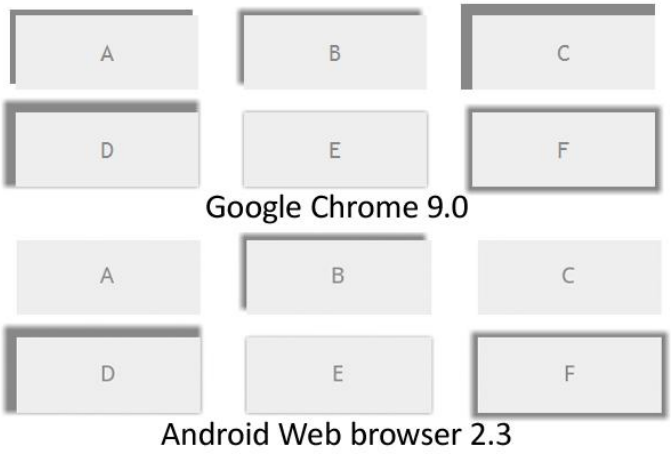
Verder zijn er een aantal functionaliteiten die gedeeltelijk ondersteund worden, deze zullen nu stuk voor stuk behandeld worden.

E-mail field type
Bij iOS apparaten past het virtuele toetsenbord zich aan aan dit veldtype om het de gebruiker makkelijker te maken. Android behandeld dit veldtype als een gewoon tekstveld en doet er verder niets mee. Het is dus wel mogelijk om gewoon te gebruiken omdat het geen negatief effect heeft op Android, maar wel een positief effect voor iOS gebruikers.

Reflections
De reflectie op Android werkt maar het mist het verloop naar transparant zoals te zien in Figuur 5.4. Doordat er zo geen mooie reflectie gemaakt kan worden op Android apparaten zal deze techniek niet worden toegepast.

Figuur 5.4 - Reflections vergelijking

Box Shadow
Zoals te zien in Figuur 5.5 missen in de Android browser schaduwen zonder een blur (voorbeeld A en C). Dit betekent dat zolang er een blur wordt meegegeven schaduwen wel gebruikt kunnen worden.



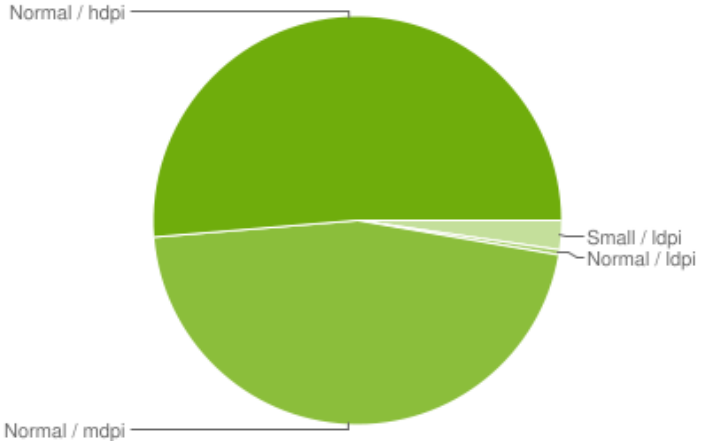
Figuur 5.5 - Box Shadow vergelijking

5.3 Hardware

Specificatie
Het iOS platform kent redelijk vaste hardware specificaties waar rekening mee gehouden kan worden en kent buiten de iPad maar 2 verschillende resoluties waar we later op terugkomen. Android draait echter op veel verschillende hardware samenstellingen. De site andro-phones.com heeft een overzicht met de specificaties van alle Android telefoons: http://www.andro-phones.com/2010-android-phones.php.

In dat overzicht is te zien dat er veel verschillende CPU en geheugen configuraties zijn binnen het Android platform. Dit betekent dat om een zo groot mogelijk publiek te bereiken de games zo min mogelijk rekenkracht en geheugen mogen gebruiken.

Schermresolutie
Ook schermresoluties kunnen verschillen bij Android. Hier zit echter wel enige regelmaat in. Er bestaan 3 categorieën schermformaten, klein, normaal en groot. De grote categorie is bedoeld voor tablets en hier richten wij ons niet op. Kleine schermen worden niet veel gebruikt zoals ook te zien in Figuur 5.6 en zijn over het algemeen niet bedoeld om games op te spelen. Wij zullen ons dus richten op schermen in de categorie normaal.



Figuur 5.6 - Verdeling van Android schermgrote en pixel-dichtheden

In deze categorie zijn 5 verschillende resoluties mogelijk die onderverdeeld zijn in verschillende pixel-dichtheden. Medium dpi (dots per inch) is het uitgangspunt voor alle andere en heeft maar 1 resolutie, 320x480. Dit is ook de resolutie van de eerste drie generaties van de iPhone en iPod Touch. Daarvan afgeleid is Low dpi met ¾ van de dichtheid en High dpi met 1,5x de dichtheid. Low dpi komt erg weinig voor en zal niet uitgebreid getest worden. Hight dpi komt wel veel voor en heeft 2 resoluties, 480x800 en 480x854. Het iOS heeft er met de introductie van de vierde generatie iPhone en iPod Touch ook een nieuwe resolutie bijgekregen, 640x960, met een pixel-dichtheid die 2x zo hoog is als die van de vorige generaties. Dit wordt door Apple het Retina-display genoemd.

Door te werken met pixel-dichtheid kunnen webpagina's eenvoudig geschaald worden. De pagina kan worden gemaakt voor een 320x480 resolutie met medium dpi en wordt op een apparaat met high dpi automatisch geschaald zodat de gebruiker hetzelfde ziet alleen scherper. Daarnaast is het ook mogelijk om bepaalde afbeeldingen alleen in te laden voor een specifieke pixel-dichtheid waardoor een apparaat met een medium dpi geen afbeeldingen hoeft te downloaden die bedoeld zijn voor een high dpi. Door de verschillende resoluties zijn er echter wel afwijkende beeldverhoudingen waardoor pagina's op het ene apparaat langer moeten zijn dan op het andere. Dit is iets waar tijdens de ontwikkeling goed rekening mee moet worden gehouden. Een overzicht van alle besproken resoluties geordend op beeldverhouding is te vinden in Figuur 5.7.

	Low (x0,75)	Medium	High (x1,5)	Retina (x2)
2:3		*320x480*		*640x960*
3:5	240x400		*480x800*	
~ 9:16			*480x854*	
5:9	240x432			

Figuur 5.7 - Resoluties per schermverhouding en pixel-dichtheid

5.4 Conclusie

Uit het onderzoek blijkt dat de meest gebruikte mobiele platformen in Nederland iOS 4.0 en hoger en Android 2.1 en 2.2 zijn dit zullen dan ook de platformen zijn waar op gericht zal worden.

Uit de geselecteerde HTML5 technieken worden de volgende ondersteund door bovengenoemde platformen en kunnen dus gebruikt worden in het project.

HTML/JavaScript	**CSS3**
Canvas/Canvas Text	Transitions
Touch events	Animations
Geolocation API	2D Transforms
Local-/session storage	Columns
ApplicationCache	Text Shadow
Web SQL Database	Box Shadow
Input Placeholder	Opacity
E-mail fieldtype	Flexible Box Model

Doordat het Android platform uit veel verschillende hardwarespecificaties bestaat, zal het cpu en geheugen gebruik laag gehouden moeten worden om zoveel mogelijk apparaten te ondersteunen. Daarnaast zijn er verschillende resoluties mogelijk waarvan er een aantal veel gebruikt zijn. Er zal uitvoerig getest worden op deze resoluties om te verzekeren dat alles er goed uit blijft zien.

6 Functioneel Ontwerp

Het functioneel ontwerp is gemaakt om voor de ontwikkeling duidelijkheid te krijgen over wat de functionaliteiten van het portaal moeten zijn en hoe de gebruiker deze moet ervaren. Door dit te documenteren is het ook mogelijk voor een ander persoon om duidelijkheid te krijgen over het project, zodat het overgenomen kan worden na afloop van de stageperiode.

6.1 Ideeën

Uit brainstormsessies en inspiratie uit andere gameportalen zoals Steam(PC/Mac)[3] en Xbox Live(Xbox 360)[4] zijn een aantal ideeën naar voren gekomen. Deze worden hier nader beschreven.

Overzicht games
Het belangrijkste element van het portaal is dat de gebruiker een duidelijk en eenvoudig navigeerbaar overzicht krijgt van de beschikbare games.

Dit kan worden uitgebreid met een sorteermogelijkheid en het selecteren van favorieten voor geregistreerde gebruikers. Dit is echter pas relevant als er een groot aantal games op het platform beschikbaar zijn.

Gebruikersaccounts
Om het voor gebruikers mogelijk te maken om persoonlijke voorkeuren op te slaan en weer op te halen moet het mogelijk zijn voor een gebruiker om zich te kunnen registreren en in te loggen.

Dit kan later worden uitgebreid met het OpenID systeem zodat gebruikers kunnen inloggen of registreren met hun bestaande account voor onder andere Facebook of Google.

Voor een gebruiker kan het interessant zijn om andere gebruikers van het portaal toe te voegen in een vriendenlijst. Zo kunnen behaalde prestaties met elkaar vergeleken worden en kan bijgehouden worden waar zijn vrienden zich mee bezig houden. Het is echter veel extra werk om dit overzichtelijk en betrouwbaar in beeld te kunnen brengen waardoor het niet in deze versie van het portaal geïmplementeerd wordt.

API
Er moet een API in het portaal worden geïntegreerd waarmee games verschillende veelgebruikte programmeertaken, zoals het berekenen van de schermoriëntatie van het apparaat, eenvoudig kunnen opvragen. Ook portaalgerelateerde zaken zoals de gebruikersnaam van de huidige gebruiker moet opgevraagd kunnen worden. Tot slot moet het mogelijk zijn voor een game om via de API bepaalde gegevens op te slaan, zoals behaalde highscores.

Highscore
Een game heeft de mogelijkheid om de score van de gebruiker bij te houden en deze op te slaan in de database van het portaal. De mogelijkheid bestaat er voor zowel geregistreerde gebruikers als gasten. Voor geregistreerde gebruikers wordt de gebruikersnaam waarmee geregistreerd is gebruikt. Gasten kunnen zelf een naam ingeven waarbij een duidelijke indicatie komt te staan dat het om een gast gaat.

Voor geregistreerde gebruikers kan een gecombineerde highscore worden bijgehouden waarbij de hoogste score van de gebruiker uit iedere game bij elkaar worden opgeteld. Deze score wordt dan op het profiel van de gebruiker weergegeven.

[3] http://steampowered.com

[4] http://www.xbox.com/live

Achievements
Achievements zijn een soort emblemen die een gebruiker kan verzamelen door bepaalde doelen gesteld door een game te behalen. Dit is een relatief simpele maar effectieve mannier om de herspeelbaarheid van een game te verhogen. Doordat mensen een game volledig willen uitspelen voor ze er mee stoppen zullen ze minder snel geneigd zijn naar een andere game over te stappen als er nog onbehaalde achievements zijn.

Offline
Dankzij de ApplicationCache techniek van HTML5 is het mogelijk om (een gedeelte van) een website lokaal op te slaan zodat deze uitgevoerd kan worden zonder internet verbinding. Voor dit project zou dit voor de games ingezet kunnen worden. Dit vereist echter wel veel extra code die moet controleren of de gebruiker online of offline is en hierop correct moet reageren.

6.2 Classificatie

De Ideeën zijn geclassificeerd naar prioriteit aan de hand van het MoSCoW-model.

Must have
- Overzicht van games
- Mogelijkheid tot registreren/inloggen
- Algemene API voor de games
- Highscore mogelijkheid voor alle gebruikers

Should have
- Achievement mogelijkheid voor geregistreerde gebruikers
- Gecombineerde highscore voor geregistreerde gebruikers

Could have
- OpenID integratie
- Downloaden van games voor offline spelen

Would have
- Friends mogelijkheden
- Favoriete games
- Status van spel opslaan zodat later verder gespeeld kan worden

6.3 Verloop

Hier wordt beschreven hoe een gebruiker het systeem zal doorlopen. Dit wordt verder weergegeven door middel van een flowchart zoals te zien in Figuur 6.1.

Startscherm
Bij het openen van de hoofdpagina van het portaal wordt gekeken of de gebruiker is ingelogd in het systeem. Als dit het geval is krijgt de gebruiker naast het overzicht van de games, inclusief een aanbevolen game, ook een compact overzicht van zijn profiel te zien. Hierin zal naast zijn gebruikersnaam en avatar ook de gecombineerde highscore die behaald is getoond worden en een verwijzing naar de laatst gespeelde game zodat deze snel en eenvoudig opgestart kan worden. Ook wordt een knop weergegeven waarmee uitgelogd kan worden. Als de gebruiker op de gebruikersnaam of avatar drukt zal een formulier verschijnen op de plaats van de aanbevolen game, waarmee het e-mailadres, de avatar en het wachtwoord van een gebruiker aangepast kan worden.

Als de gebruiker niet is ingelogd wordt in plaats van het profieloverzicht een knop weergegeven om te registreren of in te loggen. Daarnaast wordt net als bij de ingelogde gebruikers een overzicht van de games weergegeven. Inloggen of registreren gebeurt met aparte formulieren die verschijnen op de plaats van de aanbevolen game. De knop om te registreren maakt plaats voor een informerende tekst over het formulier. Het is voor een gebruiker ook mogelijk om een vergeten wachtwoord op te vragen die verstuurd zal worden naar het e-mail adres waarmee de gebruiker geregistreerd is.

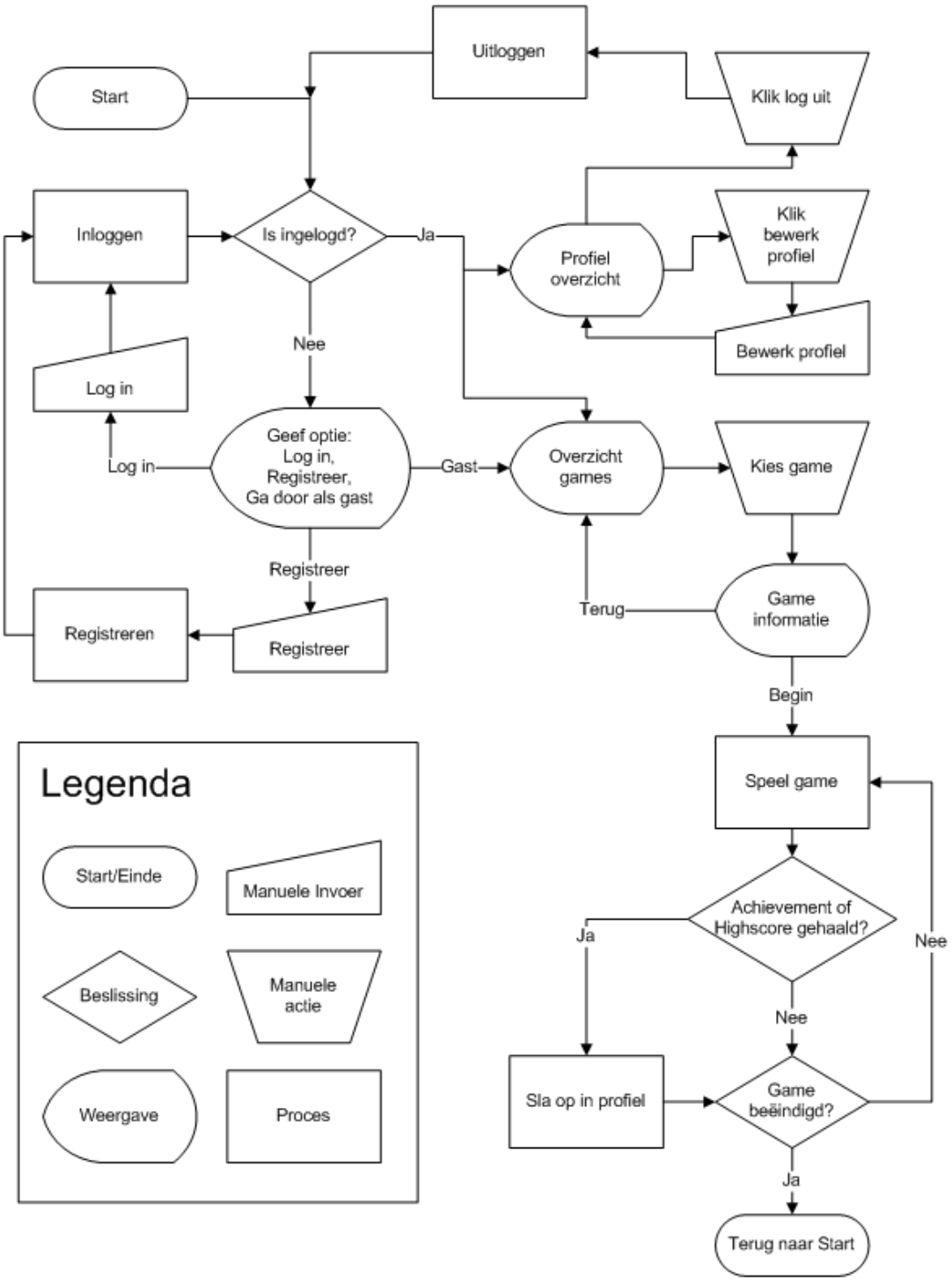
Gamescherm
Als een gebruiker op een game uit het overzicht, of de aanbevolen of laatst gespeelde game drukt verschijnt het informatie scherm voor die game. Hier wordt gedetailleerdere informatie over de game gegeven, naast de titel, bestaande uit een beschrijving en een identificerende afbeelding. Daarnaast worden enkele screenshots getoond die na een druk vergroot worden. Ook zijn hier iconen van de achievements die beschikbaar zijn in de game te zien, degene die de gebruiker al behaald heeft zijn gemarkeerd. Wanneer zo een icoon ingedrukt blijft verschijnt er de titel en een beschrijving van die achievement.

Verder krijgt de gebruiker hier de mogelijkheid om het spel te starten of terug te gaan naar het startscherm.

Game
Een game kan informatie opvragen van de huidige gebruiker, zoals zijn avatar of gebruikersnaam. Als de gebruiker een gast is zal het spel door het portaal worden gepauzeerd en zal er aan de gebruiker gevraagd worden om een naam in te voeren. Als er een achievement wordt behaald zal dit door het portaal gevisualiseerd worden. Als een gebruiker niet is ingelogd en een achievement wordt behaald wordt het spel gepauzeerd en gevraagd of de gebruiker wil inloggen of registreren of het spel verder wilt spelen zonder achievements.

Als een game wordt beëindigd zal de gebruiker terug worden gestuurd naar het startscherm.

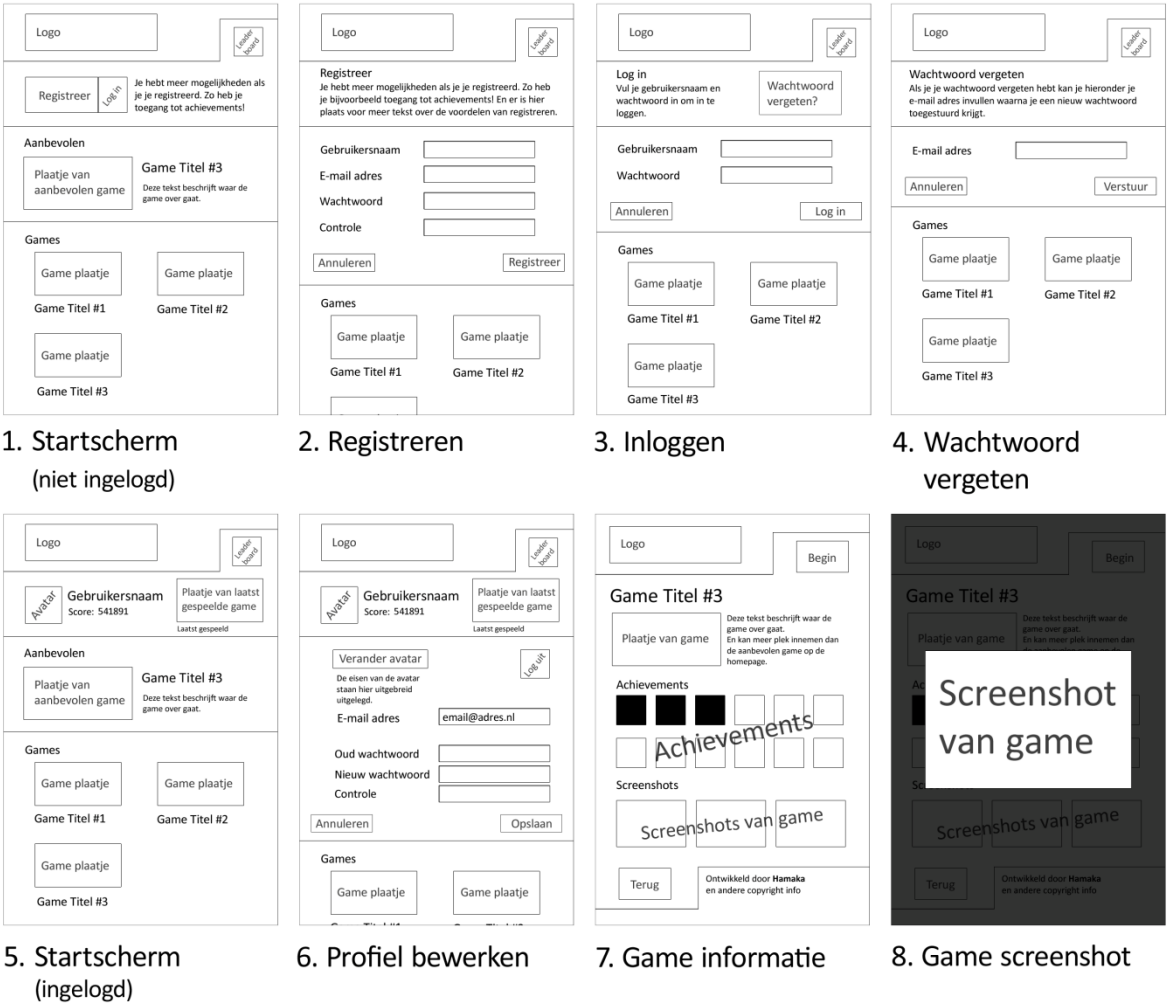
Leaderboard
Als een gebruiker op zijn gecombineerde score klikt in zijn profiel zal het leaderboard getoond worden waarin een gebruiker kan zien hoe zijn gecombineerde score zich verhoudt tegen die van andere gebruikers.



Figuur 6.1 - Flowchart

6.4 Wireframes

Omdat het erg belangrijk is dat de informatie in een duidelijke en intuïtieve mannier aan de gebruiker wordt gepresenteerd zijn er wireframes gemaakt. In een wireframe wordt aangegeven welke positie de verschillende elementen krijgen op het scherm. De wireframes uit Figuur 6.2 zijn gemaakt om te zorgen dat er een eenvoudig te begrijpen interface wordt ontwikkeld zonder dat dit wordt beïnvloed door grafische ontwerpbeslissingen. Het grafisch ontwerp zal op basis hiervan worden gemaakt.



Figuur 6.2 - Wireframes

7 Technisch Ontwerp

Op basis van het functioneel ontwerp is een technisch ontwerp gemaakt. Dit om vooraf vast te stellen met welke technieken het op te leveren product ontwikkeld zal worden, en hoe het technisch gestructureerd zal zijn. Hierdoor is het ook mogelijk voor iemand om het project over te nemen na afloop van de stageperiode.

7.1 Technieken

HTML
Bouwstenen om de visuele opbouw van het portaal te realiseren. HTML is de webstandaard voor het opbouwen van webpagina's en is recentelijk uitgebreid met nieuwe elementen. Deze maken het onder andere mogelijk om dynamische content, zoals games, weer te geven zonder gebruik van een externe browser-plugin als Flash.

CSS
Opmaaktaal voor het portaal. CSS wordt door een browser gebruikt om een opmaak te geven aan de HTML elementen van een pagina. Ook deze standaard heeft een nieuwe versie die steeds meer wordt ondersteund in browsers en waarvan ook enkele elementen gebruikt zullen worden in het portaal.

Javascript
Scripttaal die uitgevoerd wordt door de browser. Javascript geeft ons de mogelijkheid om HTML elementen te beïnvloeden zonder dat interactie van de gebruiker nodig is. Dit maakt het mogelijk om een game aan te sturen in het nieuwe canvas element, maar ook om informatie van de server te halen zonder dat een pagina helemaal ververst hoeft te worden. Dit zal dan ook gebruikt worden om de API mee te ontwikkelen waarmee de games kunnen communiceren met de server.

PHP
Scripttaal die wordt uitgevoerd door de server. Deze zorgt voor de communicatie tussen het portaal en de database. PHP is een actief ontwikkeld en breed ondersteunde taal en wordt ook door Hamaka als primaire ontwikkeltaal gebruikt.

Er zal gebruikt worden gemaakt CodeIgniter, een krachtig en snel PHP framework dat gestructureerd werken bevorderd door gebruik te maken van het Model View Controller model.

In het Model View Controller model wordt een applicatie opgedeeld in drie delen.

Controllers bevatten alle logica en worden direct aangeroepen door de gebruiker. Hier wordt gezorgd dat in de juiste situaties de juiste elementen worden aangeroepen.
In Models wordt de dataconnectie verzorgd, hier kan een Controller naar data vragen waarna de Model deze data uit de database haalt en terugstuurt aan de Controller.
Views zijn (delen van) HTML pagina's die worden aangeroepen door een Controller en waaraan data meegegeven kan worden. De vooraf gemaakte HTML pagina wordt dan samen met de dynamische data door de browser weergegeven.

MySQL
Een database systeem dat uitstekend wordt ondersteund door PHP. Hiermee zal de database worden opgebouwd die via PHP met SQL query's aangesproken zal worden.

7.2 Communicatie

Voor de API en andere elementen van het portaal die data van de server willen halen zonder de pagina te verversen is er communicatie nodig tussen de JavaScript code en de PHP code. JavaScript heeft de mogelijkheid om een webpagina op te roepen terwijl er data meegestuurd wordt en daar de uitvoer van te ontvangen zonder dat de gebruiker daar hinder van ondervindt.

Beveiliging
Er zal ook communicatie plaatsvinden waarvan niet gewenst is dat de gebruiker weet wat er gestuurd wordt omdat dit tot vals spel kan leiden, bijvoorbeeld bij het opslaan van een highscore.
Doordat JavaScript pas bij de browser gecompileerd wordt is het voor een geavanceerde gebruiker mogelijk om de broncode te lezen. Hierdoor is het helaas onmogelijk om de communicatie volledig te beveiligen. Het kan wel zo moeilijk mogelijk gemaakt worden. Door voor iedere gebruiker, iedere nieuwe browsersessie, een sleutel te genereren die meegestuurd moet worden met een server-aanroep wordt een dergelijke aanroep anders per gebruiker, per sessie. Hierdoor heeft, mocht deze gekraakt worden, de aanroep maar enkele keren een effect en maar voor 1 gebruiker. De gekraakte aanroep kan dus niet gedeeld worden met andere gebruikers.

Daarnaast zal de JavaScript code door een zogenaamde obfuscator gehaald worden. Daarmee wordt de code moeilijk leesbaar voor mensen door begrijpelijke woorden om te zetten naar enkele letters en onnodige spaties en lege regels weg te halen. Dit zorgt er meteen voor dat de bestandsgrootte verminderd wordt zodat de code sneller gedownload en ingeladen kan worden.

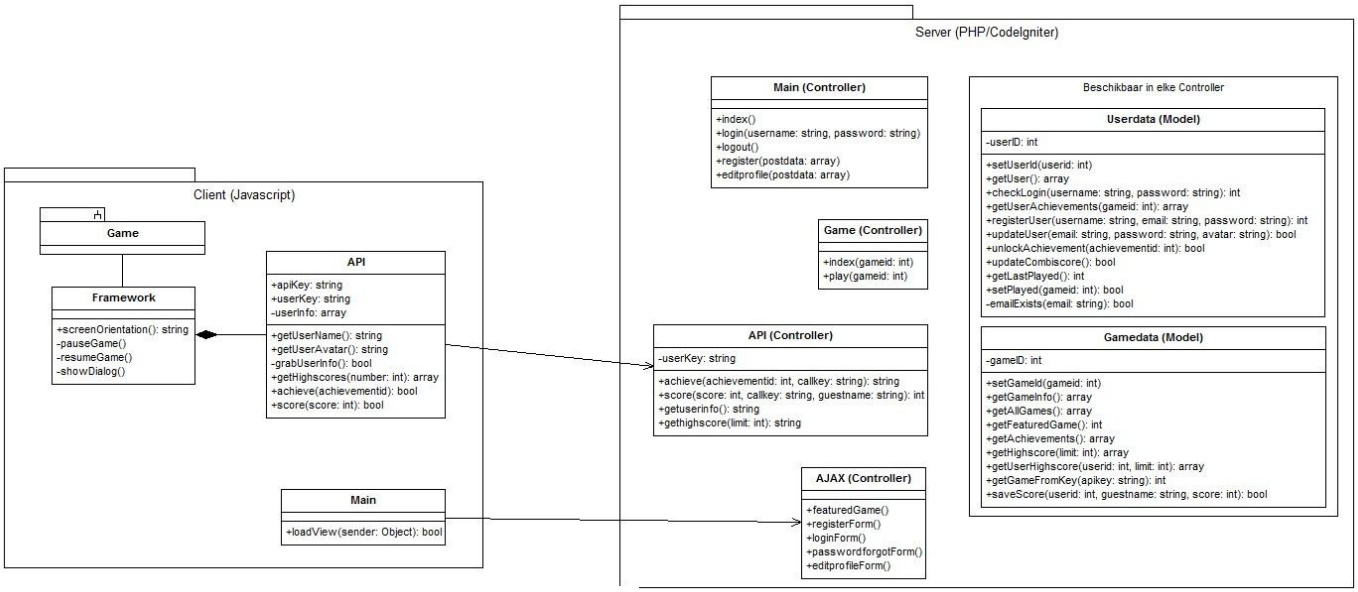
7.3 Klassendiagram

Om de technische structuur van het portaal te visualiseren en het zo overzichtelijk te maken voor het daadwerkelijke ontwikkelen is een klassendiagram ontworpen volgens de UML-standaard zoals te zien in Figuur 7.1. Hierin is ook weergegeven hoe de communicatie tussen het client-systeem en het server-systeem zal plaatsvinden.

Uitleg van de klassen
Klassennamen worden **vetgedrukt** weergegeven.

In het CodeIgniter framework kan elke controller direct worden aangesproken. Iedere controller staat dan ook voor een aparte webpagina. Als er geen specifieke controller wordt opgevraagd zal de **Main** controller worden aangeroepen. Deze geeft de juiste versie van de startpagina terug aan de browser. Verder zal deze algemene functies verwerken zoals het in- en uitloggen van een gebruiker.
Voor de formulieren voor het inloggen, registeren, etc. is het onnodig om de hele pagina opnieuw te laden omdat het grootste gedeelte hetzelfde zal blijven. Hierom zal door JavaScript het juiste gedeelte worden aangevraagd bij de **AJAX** klasse van de server.
De **Game** controller zal aangeroepen worden om de game-informatie pagina voor de aangeroepen game op te vragen. Ook het speelscherm van de game wordt hiermee opgebouwd.
Een in JavaScript geschreven game zal toegang hebben tot de JavaScript klasse **Framework** waarmee veelgebruikte berekeningen kunnen worden uitgevoerd. Bijvoorbeeld het opvragen van de schermoriëntatie van de gebruiker. Andere berekeningen worden, indien nodig, tijdens het ontwikkelen van de games toegevoegd aan het framework. Het framework bevat ook de JavaScript **API** die een game toegang geeft om een aantal server aanvragen of handelingen uit te voeren. Deze worden afgehandeld met de server klasse **API**.

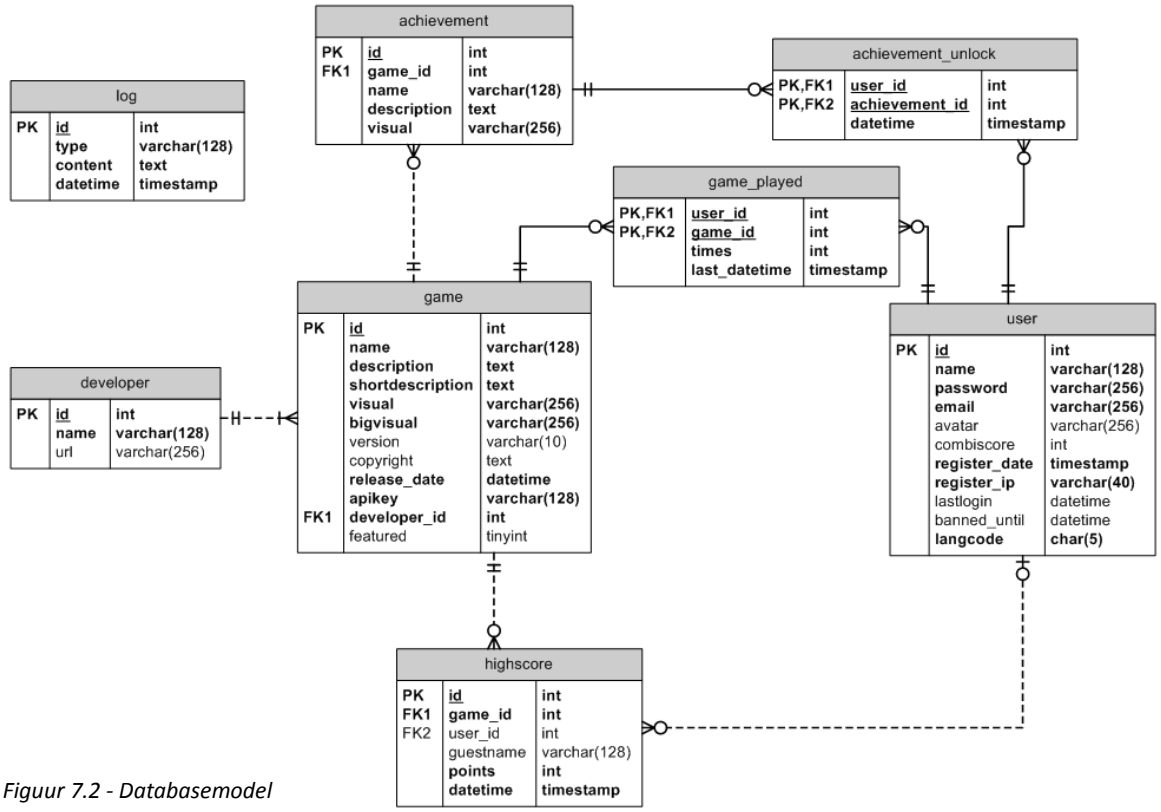
Elke Controller klasse heeft toegang tot de twee Model klassen, **Userdata** en **Gamedata**. Hieraan kan respectievelijk een userid en gameid worden meegeven om bijbehorende data op te vragen vanuit de database. Ook kan hiermee een relevant item aan de database worden toegevoegd of gewijzigd.



Figuur 7.1 - Klassendiagram

7.4 Databasemodel

Het database model, zoals te zien in Figuur 7.2, is gemaakt om een zo optimaal mogelijke datastructuur te ontwerpen. Zonder dat er redundantie optreedt en met maximale uitbreidingsmogelijkheden voor de toekomst.



Figuur 7.2 - Databasemodel

8 Realisatie

In dit hoofdstuk wordt per fase van de projectrealisatie beschreven hoe deze verliep en wat hierin gebeurde.

8.1 Globale fasering

De realisatie van het project is verlopen in enkele fases.

- Ontwerpen
- Opzetten database en server klassen
- Ontwikkelen game- framework en API
- Ontwikkelen interactieve gebruikersinterface
- Ontwerpen eerste game
- Ontwikkelen eerste game
- Ontwerpen tweede en derde game
- Ontwikkelen tweede en derde game
- Testen

8.2 Realisatie per fase

Ontwerpen
Bij het presenteren van de resultaten van het onderzoek is een brainstormsessie gehouden binnen Hamaka waarin de functionaliteiten voor het gameportaal zijn besloten. Op basis hiervan is het functioneel ontwerp gemaakt. Daarin is ook getoond hoe een gebruiker het systeem zal doorlopen en hoe de interface is opgebouwd. Dit laatste kende enkele revisies na feedback.

Vervolgens is een technisch ontwerp gemaakt waarin onder andere beschreven is hoe de communicatie tussen de JavaScript en PHP code plaatsvind en hoe de code gestructureerd is. Ook is een model voor de database ontworpen.

Daarna is een grafisch ontwerp gemaakt op basis van de wireframes uit het functioneel ontwerp. Deze is nog meerdere malen aangepast tijdens latere fases na feedback. Dit was mogelijk omdat het grafisch ontwerp pas weer noodzakelijk werd bij het ontwikkelen van de gebruikersinterface.

Opzetten database en server klassen
De volgende fase begon met het opzetten van de database structuur op basis van het ontwerp uit het technisch ontwerp. Hierna zijn alle model klassen gemaakt op de server welke voor de communicatie moeten zorgen tussen de controller klassen en de database. Dit is gebeurd op basis van het klassendiagram uit het technisch ontwerp.

Vervolgens zijn alle controller klassen op de server gemaakt volgens het technisch ontwerp.

Ontwikkelen game- framework en API
Aangezien het grafische ontwerp nog geen goedgekeurde versie had bereikt was de volgende fase het ontwikkelen van het game- framework en API in JavaScript. Hiervoor is een test-'game' gemaakt die bestond uit een JavaScript document die alle functies van het framework en de API aanriep en het resultaat ervan toonde. Bij het ontwikkelen van de API is ook de API controller op de server bijgewerkt om ondersteuning voor, onder andere, de toegevoegde beveiliging toe te voegen.

Ontwikkelen interactieve gebruikersinterface
Nadat het uiteindelijke grafisch ontwerp voltooid was is dit ontwerp omgezet naar een interactieve variant. Hierbij is gebruik gemaakt van een combinatie van HTML, CSS en JavaScript om een vloeiende interface te maken die eenvoudig is in gebruik.

De grootste uitdaging was het schalen van de interface voor de verschillende resoluties en pixeldichtheden van mobiele apparaten. Dit is uiteindelijk opgelost door voor iedere pixeldichtheid een eigen set afbeeldingen en CSS bestand te maken. Hierdoor wordt alles altijd in de goede grote weergeven en worden er geen afbeeldingen geladen die onnodig groot zijn voor het apparaat waarmee het portaal bekeken wordt.

Ontwerpen eerste game
Nadat de eerste versie van het portaal klaar was volgde een brainstormsessie om met ideeën te komen voor de eerste game. Dit resulteerde in een functioneel en technisch ontwerp voor deze game waarin voornamelijk het concept en de gameplay elementen van de game worden beschreven samen met hoe de game technisch gestructureerd is. Hierna is een grafisch ontwerp voor alle elementen in de game gemaakt.

Ontwikkelen eerste game
De ontwikkeling van de eerste game begon met het uitbreiden van het EaselJS framework. Het framework had voor mij veel voordelen voor het ontwikkelen van games op basis van de HTML canvas omdat het veel lijkt op programmeren in ActionScript 3. Het miste echter een belangrijk onderdeel voor deze game. Namelijk de ondersteuning van touchevents, welke zorgen voor ondersteuning van touchscreens. Het toevoegen van deze functionaliteit bleek uiteindelijk niet al te complex omdat het mogelijk was om de wel bestaande mouseevent functionaliteiten te kopiëren en aan te passen.

Tijdens het ontwikkelen van de game zelf waren er af en toe problemen met het feit aan dat JavaScript objectgeoriënteerd programmeren niet fatsoenlijk ondersteund. Dit zorgde er soms voor dat sommige variabelen of functies niet naar verwachting reageerde. Dit was wel altijd met een kleine omweg weer op te lossen. Tijdens de ontwikkeling van de game is ook het framework en de API van het portaal uitgebreid nu er meer zekerheid was over wat belangrijk is in de ontwikkeling van deze games en hoe ze gepresenteerd worden.

Bij vroege tests werd duidelijk dat er een fout zit in de implementatie van het HTML canvas element in de Android 2.1 browser. Deze maakt het onmogelijk om fatsoenlijk een afbeelding in het canvas te plaatsen, wat essentieel is voor deze games. Het gebruik van deze Android versie loopt echter snel af en gebruik hiervan zal naar verwachting tegen tijd van de release van het portaal miniem zijn. Vannaf Android 2.2 werkt het canvas wel weer naar behoren.

Nadat de game klaar was is deze samen met het portaal uitvoerig getest door de medewerkers van Hamaka en feedback die daarbij naar voren kwam is meegenomen bij het verbeteren van beide producten.

Ontwerpen tweede en derde game
De volgende fase begon weer met een brainstormsessie waarin over het concept van de tweede en derde game is besloten. Hierna is een functioneel en technisch ontwerp opgesteld voor beide games met dezelfde opzet als dat van de eerste game. Ook zijn er weer bijbehorende grafische ontwerpen gemaakt.

Ontwikkelen tweede en derde game
De laatste twee games worden ontwikkeld in de tijd tussen het schrijven van deze scriptie en de afstudeerzitting. Hierbij zal een zelfde structuur worden aangehouden als bij het ontwikkelen van de eerste game.

Testen
Uiteindelijk zullen alle elementen uitvoerig worden getest door de medewerkers van Hamaka en waar nodig zullen verbeteringen aan worden gebracht in de producten.

9 Eindproduct

In dit hoofdstuk wordt beschreven wat het resultaat is van het eindproduct van het project. Dit resultaat wordt vervolgens geëvalueerd copleet met een conclusie en aanbeveling voor de opdrachtgever.

9.1 Resultaat

Het uiteindelijke eindproduct bestaat uit het gameportaal “Nebula”, een game met de titel “Rise of the Penguins” en twee games die ontwikkeld worden in de rest van de stage periode genaamd “Velo City” en “Photo Slide”. Deze laatste games zullen in dit hoofdstuk buiten beschouwing worden gelaten maar worden wel behandeld tijdens de afstudeerzitting.

Nebula
Alle elementen die in het functioneel ontwerp als must have of should have zijn geclassificeerd zijn in het eindproduct geïmplementeerd zoals beschreven in het functioneel ontwerp.

De visuele elementen worden automatisch geschaald naar de juiste resolutie door gebruik te maken van de nieuwe media queries. Hiermee wordt gekeken wat de pixeldichtheid van het huidige scherm is en haalt daarbij het juiste CSS bestand op waarin de bijbehorende groten gespecificeerd staan net als verwijzingen naar de correcte afbeeldingen.

Alle deelschermen van het starscherm, zoals het inlogformulier of het leaderboard, worden ingeladen zonder dat de hele pagina wordt ververst. De nieuw ingeladen delen worden vervolgens in beeld gebracht met de nieuwe CSS animatie mogelijkheid. Als deze techniek niet wordt ondersteund door de browser wordt een JavaScript animatie gebruikt.

Bij het inloggen van een gebruiker worden de inloggegevens opgeslagen in de nieuwe localStorage component van de browser. Het wachtwoord is hierbij versleuteld. Als een gebruiker op het portaal komt en deze is niet ingelogd wordt eerst gekeken of er iets in deze localStorage staat. Indien dat zo is wordt deze informatie gebruikt om porberen in te loggen. Als dit mislukt wordt de login informatie verwijderd uit de localStorage.

Tijdens het inloggen wordt tevens een sleutel gegenereerd op de server die vervolgens wordt opgeslagen in de sessionStorage component van de browser. Dit werkt hetzelfde als de localStorage maar wordt automatisch leeggehaald als de browsersessie wordt beëindigd. Deze sleutel wordt gebruikt voor het versleutelen van beveiligde API aanroepen. Deze aanroepen worden vervolgens aan de server kant weer ontsleutelt.

Het game-framework geeft games de mogelijkheid om:

- de hoogte en breedte van het webdocument te berekenen;
- de adresbalk te verbergen in de Android of iOS browser;
- de huidige schermoriëntatie op te vragen;
- de gewenste schaal berekenen op basis van een bron pixelratio;

Daarnaast is het voor een game mogelijk om via de API

- iemands gebruikersnaam op te vragen, als de gebruiker niet is ingelogd wordt een invulveld getoond waarin de gebruiker een naam kan opgeven;
- de URL naar de avatar van de huidige gebruiker op te vragen;
- de highscore lijst van de game op te vragen;
- het behalen van een achievement op te slaan voor een gebruiker;
- een behaalde score op te slaan, gekoppeld aan de ingelogde gebruiker of aan de naam ingevuld door de niet-ingelogde gebruiker.

Bij het opslaan van het behalen van een achievement wordt gecontroleerd of de gebruiker in is gelogd of niet. Als dit het geval is zal in een hoek van het scherm voor een paar seconde verschijnen dat er een achievement behaald is. Als de gebruiker niet is ingelogd zal de pauze functie van de game worden aangeroepen en wordt de gebruiker gevraagd of deze in wilt loggen of registeren om dit op te slaan. Als de gebruiker besluit dit niet te doen zal dit voor de rest van de game niet meer worden gevraagd. Wordt wel ingelogd of geregistreerd handelt het framework dit af zonder dat de gebruiker uit de game wordt gehaald.

Rise of the Penguins
De eerste game voor het Nebula portaal maakt gebruik van het nieuwe canvas element van HTML. Hierop kunnen met behulp van JavaScript elementen worden getekend en dus geanimeerd wat belangrijk is voor veel games. Het maakt ook gebruik van de nieuwe touchevent mogelijkheid van JavaScript waarmee schermaanrakingen op een touchscreen te detecteren zijn. Voor het “afschieten” van de puck in de game wordt het verschil in tijd en afstand tussen het plaatsen van een vinger op het scherm en het loslaten gebruikt om de snelheid en richting van de puck te berekenen.

Twee pucks kunnen elkaar raken en zullen vervolgens op een natuurlijke mannier van elkaar afketsen. Ditzelfde principe geld voor een pinguïn die een puck raakt of een pinguïn die een andere pinguïn raakt.

De game probeert een constante framerate van 25 frames per seconde (fps) te halen. Als de framerate voor een volle seconde onder de 20fps valt zal een lagere resolutie voor de achtergrond afbeelding gebruikt worden zodat de browser minder berekeningen hoeft te doen om een frame weer te geven. Deze resolutie verlaging verloopt in vier stappen waarbij de laatste stap bestaat uit het helemaal vervangen van de achtergrondafbeelding door een egale kleur.

Om te voorkomen dat het hele spel wordt afgesloten als de gebruiker op de terug knop van de browser drukt, bijvoorbeeld omdat deze terug wil gaan van het highscore menu naar het hoofdmenu, wordt op iedere aparte pagina in de game de URL aangepast. Zo ziet de browser het ook als een aparte pagina en kan de terug knop werken zoals de gebruiker het verwacht.

De game bevat een aantal functies die aangeroepen kunnen worden door het Nebula framework. Als er door het framework een dialoog wordt getoond zal de pauze functie worden aangeroepen en bij het sluiten van het dialoog de hervat functie. De game heeft in deze functies alle onderdelen staan die nodig zijn om het spel te pauzeren en daarna te hervatten. Als de game wordt afgesloten door het framework krijgt wordt in de game eerst een functie aangeroepen zodat deze de kans krijgt om eerst een aantal handelingen uit te voeren.

9.2 Evaluatie

Het oorspronkelijke doel van de opdracht was het experimenteren en ervaring opdoen met nieuwe webtechnieken. Dit is zeker gelukt, er zijn al verschillende nieuwe technieken gebruikt en er zullen er nog een aantal volgen in de nog te ontwikkelen games.

Ook aan de eisen van de opdracht zelf zijn voldaan, alleen het aantal games moet op moment van schrijven nog worden uitgebreid. Het portaal en de games functioneren naar behoren op ten minste het Android en iOS platform. Verder werkt het op alle recente versies van de grote desktop browsers, waaronder Internet Explorer 9 welke later dit jaar ook verschijnt op het Windows Phone 7 platform waardoor waarschijnlijk dan ook dit platform ondersteund is.

Het portaal is gebruiksvriendelijk gebleken uit tests met verschillende mensen die het product daarvoor nog niet gebruikt hadden. Het enige dat niet voor iedereen duidelijk was is de leaderboard knop. De functie van deze knop zal dan ook nog duidelijker gemaakt moeten worden. Ook de game is gebruiksvriendelijk gebleken, iedereen heeft altijd op tijd door gehad wat de bedoeling is en hoe de game bestuurd moet worden.

Verder is het voor iedereen, geregistreerd of niet, mogelijk om de games te spelen en zich te plaatsen in de highscore lijsten van de games. Geregistreerde gebruikers hebben wel extra mogelijkheden zoals het behalen van achievements maar hebben geen voordelen die directe invloed hebben op de games waardoor iedereen gelijk wordt behandeld.

Het portaal is met een paar kleine aanpassingen eenvoudig meertalig te maken doordat er gebruik is gemaakt van losse taalbestanden die apart in te laden zijn. Ook kan de interface worden voorzien van een alternatief kleurenschema. Beide mogelijkheden zijn al functioneel en gekoppeld aan de gebruikersaccounts alleen worden deze opties nog niet met de gebruikers gecommuniceerd. Het portaal is dus ontworpen voor uitbreidbaarheid op meerdere vlakken, daarbij ook het aantal games. Omdat de games vrijwel volledig los staan van de interface van het portaal kunnen games worden toegevoegd zonder dat een aanpassing aan het portaal nodig is.

Een probleem met een systeem dat gebruik maakt van JavaScript is dat de eindgebruiker altijd toegang heeft tot de broncode en in dit geval dus de mogelijkheid heeft om vals te spelen met highscores en achievements. Er is een beveiliging opgebouwd om te zorgen dat een aanroep naar de server om dit soort informatie op te slaan maar één keer hetzelfde kan zijn in een speelsessie. Dit samen met het zo onleesbaar mogelijk maken van de code zal het voor de meeste mensen onmogelijk maken om vals te spelen. Er zijn echter altijd mensen die de code wel ontcijferd kunnen krijgen en deze kunnen dus zien hoe de beveiliging is opgebouwd waardoor het voor hun mogelijk is deze te kraken. Beveiliging tegen geautomatiseerde scripts in het registratieformulier missen ook nog in deze versie van het portaal.

Als er een groot aantal games wordt toegevoegd aan het portaal zal het huidige overzicht onoverzichtelijk worden wat nadelig is voor de eindgebruiker. Echter zolang het aantal games relatief laag blijft zouden extra overzichtselementen alleen onnodige ruimte in beslag nemen.

9.3 Conclusie

Het project is in zijn originele doel geslaagd om kennis en ervaring op te doen over nieuwe webtechnieken. Daarnaast voldoet het uiteindelijke product aan de eisen die door de opdrachtgever gesteld zijn. Het eindproduct is aan het einde van de stageperiode volledig functioneel en bruikbaar voor de huidige situatie. Bij de toevoeging van een grote hoeveelheid games zou echter het overzicht aangepast moeten worden. En als het portaal populair mocht worden zal de beveiliging verder opgeschroefd moeten worden om vals spel te voorkomen.

De gebruiksvriendelijkheid is over het algemeen goed maar kan verbeterd worden. Voornamelijk door een verduidelijking van de leaderboard knop.

Mocht het later gewenst zijn is het eenvoudig om lokalisering toe te voegen aan het portaal, en is het mogelijk om nieuwe talen toe te blijven voegen. Toevoeging van nieuwe kleurenschema's is ook een mogelijkheid en een geregistreerde gebruiker kan de optie worden gegeven om een kleurenschema te kiezen.

9.4 Aanbevelingen

Ik beveel Hamaka aan bij voortzetting van het project het registratieformulier te voorzien van een *captcha* beveiliging om te zorgen dat geautomatiseerde scripts geen accounts aan kunnen maken op het portaal.

Daarnaast kunnen de resultaten van de tests aan het einde van het project gebruikt worden om de gebruiksvriendelijkheid van het portaal waar nodig te verbeteren.

Tot slot is het aan te raden om de beveiliging van de client-server communicatie te blijven uitbreiden als hier nieuwe inzichten over komen.

10 Reflectie

In dit hoofdstuk wordt beschreven hoe ik voldaan heb aan de voorwaarden van de opleiding met behulp van de generieke competenties van de Bachelor of Engineering en de Dublin descriptoren. Daarnaast geef ik een profielschets van mezelf gebaseerd op deze stageperiode.

10.1 Competenties van de Bachelor of Engineering

Tijdens de opleiding wordt gericht op het beheersen van de vier generieke competenties van de Bachelor of Engineering. Hieronder wordt beschreven hoe ik denk dat aan deze competenties voldaan zijn in dit project.

Inzicht krijgen
Het inzicht krijgen in de opdracht heb ik gerealiseerd door eerst bij verschillende bestaande mobiele webpagina's en games te kijken hoe deze zijn opgebouwd. Daarnaast heb ik gekeken naar verschillende implementaties van HTML5 technieken om inspiratie op te doen voor het gebruik van deze technieken en meteen te zien hoe deze te implementeren zijn.

Daarnaast heb ik het onderzoek, zoals beschreven in deze scriptie, uitgevoerd. Buiten het inzicht krijgen in verschillende technieken heb ik hierbij de grenzen van het project kunnen bepalen.

Ontwerpen
In overleg met de opdrachtgever zijn de functionaliteiten en de opbouw van het project vastgelegd. Daarbij heb ik gebruik gemaakt van de verkregen inzichten om de door mij voorgestelde methodes te beargumenteren.

Op basis van dit overleg heb ik de functionaliteiten en opbouw van het project uitgewerkt in een functioneel en technisch ontwerp.

Plannen
Ik heb een plan gemaakt voor de aanpak van het project. Hierin heb ik het project opgedeeld in fases en beschreven hoe de uitvoering van die fases plaats zouden vinden. Daarnaast heb ik in een tijdsplanning de verschillende fases verdeeld over de weken van de stageperiode.

Uitvoeren
Bij de uitvoering van het project is zoveel mogelijk vast gehouden aan het plan en de ontwerpen. Er is alleen van afgeweken als hier een gegronde reden voor was. Bijvoorbeeld het vervroegen van de onwikkeling van de eerste game om zo ervaring op te doen om de haalbaarheid en limitaties beter te betrekken bij het ontwerpen van de latere games.

10.2 Dublin descriptoren

De eindtermen van de bachelor opleiding worden beschreven in de Dublin descriptoren. Hieronder wordt aangegeven hoe ik denk aan deze descriptoren voldaan te hebben in dit project.

Kennis en inzicht
Ik heb de kennis en inzicht die ik heb opgedaan tijdens de opleiding nodig gehad om een goed geïnformeerd oordeel te kunnen vormen tijdens de ontwerpfase. Ook tijdens de uitvoering heb ik me kunnen beroepen op eerder opgedane kennis en inzicht.

Daarnaast heb ik kennis en inzicht vergaart tijdens het project. Het project is gericht op het toepassen van de nieuwste ontwikkelingen op het gebied van webtechniek. Hierdoor heb ik me constant op de hoogte gehouden van ontwikkelingen en ontdekkingen die invloed zouden kunnen hebben op het project en deze waar nodig verder onderzocht en toegepast.

Toepassen kennis en inzicht
De opgedane kennis en inzicht zijn nodig geweest om tot afgewogen beslissingen te komen en deze op een goede mannier te implementeren in het project. Bij het ontwikkelen van de producten was mijn kennis in programmeren en inzicht in de gebruikte technieken belangrijk om tot oplossingen te komen bij problemen die zich voordeden.

Oordeelsvorming
Voor het vormen van een oordeel, over bijvoorbeeld de mannier van technische implementatie, heb ik de relevante gegevens verzameld door onderzoek te doen naar mogelijke oplossingen en door te experimenteren met verschillende beschikbare technieken. Door de gevonden data met elkaar te vergelijken werd het mogelijk voor de beste oplossing te kiezen voor de situatie.

Communicatie
Er hebben binnen Hamaka een aantal brainstormsessies plaatsgevonden waarin ik samen met de rest van het bedrijf tot ideeën over onder andere functionaliteiten voor de verschillende producten ben gekomen. Daarnaast is er veel overleg geweest over verschillende delen in het project. Een voorbeeld hiervan is het grafisch ontwerp waarbij uiteindelijk is besloten om een andere kleur te gebruiken dan ik oorspronkelijk in gedachte had, om zo tot een beter product te komen.

Leervaardigheden
Dit project vereiste van mij dat ik bekend raakte met nieuwe technieken op een platform (mobiel) waar ik redelijk weinig ervaring mee had. Hierdoor ben ik constant bezig geweest met het leren van nieuwe elementen hiervan. Doordat het project bestond uit meerdere producten had ik de mogelijkheid om dingen die ik geleerd had bij het ontwikkelen van het ene product, beter doordacht toe te passen in het volgende product. Hierdoor kreeg ik daarbij weer de ruimte om nieuwe dingen uit te proberen om weer van te leren.

10.3 Profielschets

In mijn werk ben ik resultaat- en doelgericht. Ik ben pas tevreden over mijn eigen resultaat wanneer dit aansluit bij het beeld dat ik ervan heb. Dit perfectionisme kan ik in de meeste gevallen wel goed relativeren aan de tijd die beschikbaar is. Ik kan goed zelfstandig werken maar vind het ook prettig om in een teamverband tot beslissingen te komen. Bij het vinden van oplossingen of het opdoen van ideeën komt het voor dat ik te lang zelfstandig focus op een oplossing. In zo een geval zou ik beter een beroep kunnen doen op de expertise of ideeën van mijn collega's.

Ik ben geïnteresseerd in het ontdekken van nieuwe technieken en het experimenteren hiermee. Daarbij ga ik als programmeur met een creatieve instelling graag de uitdaging aan om complexe systemen te ontwikkelen welke voor de eindgebruiker zo simpel mogelijk zijn in gebruik.

Genoeg variatie in werk is voor mij belangrijk. Daarbij wens ik de mogelijkheid te hebben om nieuwe ontwikkelingen in mijn vakgebeid te volgen en deze in een geschikte situatie toe te passen. Ook een hoge mate van verantwoordelijkheid en bevoegdheden binnen een project zijn voor mij belangrijk zodat ik ook zelfstandig te werk kan gaan. Een functie als ontwikkelaar in een creatief en mediagericht team past het hiermee best bij mij.

Verklarende woordenlijst

API	[*Application Programming Interface*] – Een verzameling functies die een applicatie in staat stellen te communiceren met een andere applicatie of softwareonderdeel.
Apps	Afkorting voor applicaties. Meestal gebruikt om een applicatie binnen een mobiele omgeving aan te duiden.
Capthca	[*Completely Automated Public Turing-test to tell Computers and Humans Apart*] – Een test die niet gemaakt kan worden door een geautomatiseerd script en dus kan controleren of de maker van de test menselijk is.
CSS	[*Cascading Style Sheet*] – Geeft aan hoe een webbrowser de elementen aangegeven in een HTML document moet opmaken.
Framework	(In softwareontwikkeling) Een verzameling van componenten en functies die gebruikt kunnen worden bij het programmeren van applicaties.
HTML	[*HyperText Markup Language*] – Deze taal geeft aan hoe een website is opgebouwd en welke elementen deze bevat.
HTML5	Verzamelterm voor verschillende nieuwe, of in ontwikkeling zijnde, webtechnieken en -standaarden. Hieronder vallen onder andere HTML versie 5, CSS versie 3 en verschillende nieuwe JavaScript functionaliteiten zoals de “Geolocation API”.
JavaScript	Een scripttaal die wordt uitgevoerd door de webbrowser van een gebruiker.
PHP	[*PHP: Hypertext Preprocessor*] – Een scripttaal die na aanvraag van een gebruiker wordt uitgevoerd door de server, waarna de uitvoer naar de webbrowser van de gebruiker wordt gestuurd.
SQL	[*Structured Query Language*] – Een taal voor het opvragen of aanpassen van gegevens in een database.
W3C	[*World Wide Web Consortium*] – Een organisatie verantwoordelijk voor het ontwerpen van de standaarden voor het wereldwijde web.

Bronnen

AndroLib.com. (2011). *Distribution of Apps and Games in Android Market*. Opgeroepen op 2 Februari, 2011, van AndroLib: http://www.androlib.com/appstatstype.aspx

andro-phones.com. (2011). *All Android Phones of 2010*. Opgeroepen op 10 Februari, 2011, van AndroPhones.com: http://www.andro-phones.com/2010-android-phones.php

Apple Inc. (2011). *iTunes Charts - Paid Apps*. Opgeroepen op 2 Februari, 2011, van Apple.com: http://www.apple.com/itunes/charts/paid-apps/

Google Inc. (2011). *Developer Resources*. Opgeroepen op 10 Februari, 2011, van Android Developers: http://developer.android.com/resources/index.html

Modernizr. (2011). Opgeroepen op 10 Februari, 2011, van Modernizr: http://www.modernizr.com/

Pilgrim, M. (2011). Opgeroepen op 10 Februari, 2011, van Dive Into HTML5: http://diveintohtml5.org/

Ruby, D. (2011). *61% of iPads Already Running iOS 4 – January iOS and Android OS Breakdown.* Opgeroepen op 10 Februari, 2011, van Chitika Insights: http://insights.chitika.com/2011/61-of-ipads-already-running-ios-4-january-ios-and-android-os-breakdown/

StatCounter. (2011). *Top 9 Mobile Browsers in Netherlands on Jan 11*. Opgeroepen op 10 Februari, 2011, van StatCounter Global Stats: http://gs.statcounter.com/#mobile_browser-NL-monthly-201101-201101-bar

TrouserMac Industries. (2011). *Apple iTunes App Store Metrics, Statistics and Numbers for iPhone Apps*. Opgeroepen op 2 Februari, 2011, van 148Apps: http://148apps.biz/app-store-metrics/?mpage=catcount

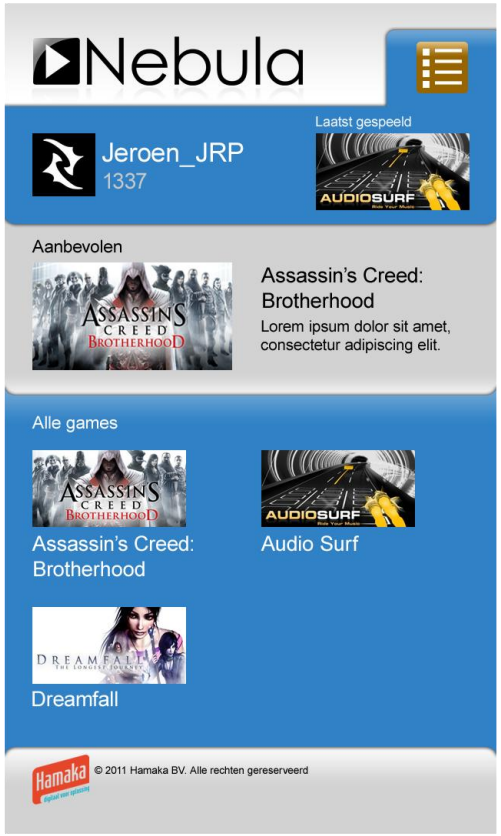
Violin, L. (2011). *Windows Phone 7 Marketplace Stats*. Opgeroepen op 2 Februari, 2011, van WP7applist: http://wp7applist.com/stats/

World Wide Web Consortium. (2011). *DeviceOrientation Event Specification.* Opgeroepen op 10 Februari, 2011, van W3C: http://dev.w3.org/geo/api/spec-source-orientation.html

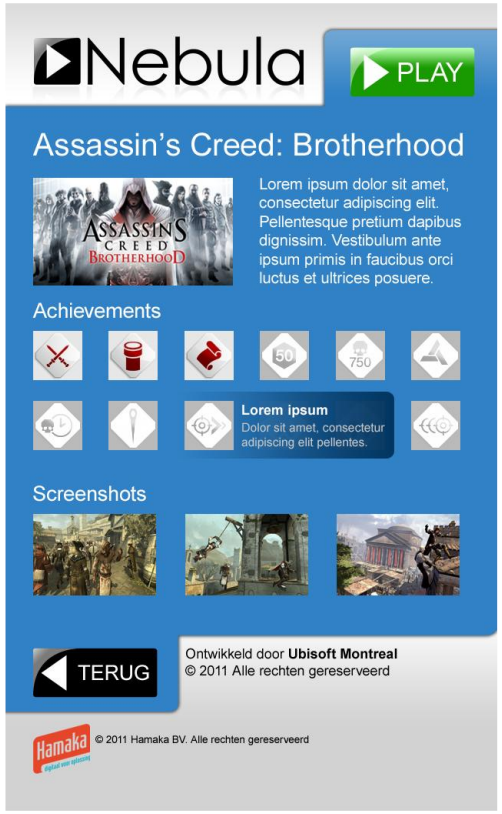
Bijlagen

Bijlage 1: Overzicht van onderzoek tests

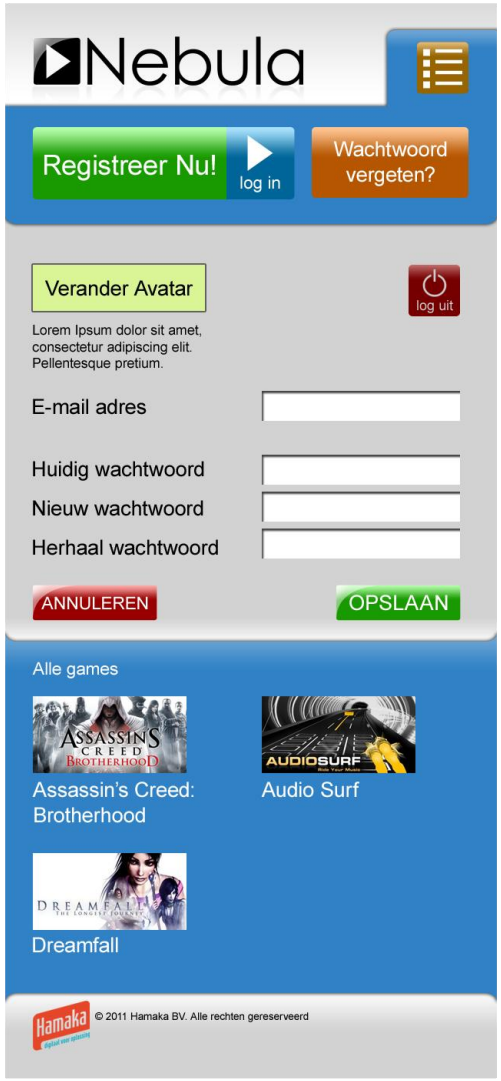
Functionaliteit	Test
Canvas/Canvas Text	http://html5demos.com/canvas
HTML5 Audio	http://demos.w3avenue.com/html5-unleashed-tips-tricks-and-techniques/sample-04-audio-demo.html
Touch events	http://o.sitepen.com/labs/code/iphone/events.html
SVG	http://www.w3schools.com/svg/tryit.asp?filename=circle1&type=svg
Geolocation API	http://html5demos.com/geo
Web Sockets	http://html5demos.com/web-socket
local-/sessionStorage	http://html5demos.com/storage
applicationCache	http://html5demos.com/offlineapp
Web SQL Database	http://html5demos.com/database
DeviceOrientation	http://ariya.github.com/js/marblebox/
Input fields	
Placeholder attribute	http://davidwalsh.name/dw-content/html5-placeholder.php
E-mail field type	http://www.w3schools.com/html5/tryit.asp?filename=tryhtml5_form_email
Form validation	http://diveintohtml5.org/examples/input-required.html
CSS3	
Transitions	http://css3.bradshawenterprises.com/
Animations	http://css3.bradshawenterprises.com/
2D Transforms	http://www.westciv.com/tools/transforms/index.html
Reflections	http://digitaalcv.nl/demos/iphone/reflection/
Columns	http://www.css3.info/preview/multi-column-layout/
Text Shadow	http://www.css3.info/preview/text-shadow/
Box Shadow	http://www.css3.info/preview/box-shadow/
Opacity	http://www.css3.info/preview/opacity/
Flexible Box Model	http://www.html5rocks.com/tutorials/flexbox/quick/



Start pagina



Game informatie



Losse elementen

Alle game en achievement afbeeldingen zijn afkomstig van Steam (http://store.steampowered.com).
Alle rechten zijn voorbehouden aan hun respectievelijke makers.

Bijlage 3: Documentatie eerste game

Rise of the Penguins

Powered by Nebula

Hamaka
digitaal voor oplossing

Inhoudsopgave

1 Functioneel Ontwerp

Dit functioneel ontwerp is gemaakt om voor de ontwikkeling duidelijkheid te krijgen over wat het idee achter de game is en in welke setting deze zich afspeelt. Daarnaast wordt beschreven uit welke gameplay functionaliteiten de game zal bestaan. Tot slot wordt een lijst met achievements opgesteld die in de game te behalen zullen zijn.

1.1 Concept

Het uitgangspunt voor deze game is dat het gebruik moet maken van de touchscreen mogelijkheden van mobiele apparaten die ondersteund zijn in HTML5.

In een brainstorm sessie ontstond het idee om de speler zich te laten verdedigen voor objecten of vijanden die van bovenaf in het scherm naar beneden komen. Dit door met de vinger iets af te schieten waarbij de richting en snelheid bepaald kan worden door de beweging die met de vinger gemaakt wordt.

De setting waar uiteindelijk voor is gekozen is die van een ijsbaan, waarbij pinguïns de vijanden zijn die van boven naar onder in het scherm lopen. De speler heeft beschikking over hockeypucks waarmee de pinguïns van de baan kunnen worden geduwd.

1.2 Gameplay

Op basis van het concept zijn de gameplay functionaliteiten uitgedacht.

De pinguïns verschijnen 1 voor 1 op een willekeurige positie aan de bovenkant van het scherm. De tijd hiertussen wordt korter naarmate het spel vordert. De pinguïns bewegen met een vaste snelheid richting de onderkant van het scherm in een rechte lijn. Als een pinguïn de onderkant bereikt verliest de speler een leven. Nadat 3 levens zijn verloren is het spel afgelopen.

Onderin het scherm bevindt zich een hockeypuck. De speler kan hier een vinger op leggen en deze in elke richting bewegen waarna de vinger weer van het scherm moet worden gehaald. De puck zal met de vinger meebewegen en de snelheid van de vinger beweging aanhouden. De vingerbeweging zal binnen een bepaalde radius moeten plaatsvinden om uitdaging in de game te houden. Tijdens de beweging van de puck over de baan zal de snelheid geleidelijk afnemen waardoor de puck stil zal komen te liggen als de beginsnelheid niet snel genoeg was. Een stilstaande puck zal vanzelf van het speelveld verdwijnen. Nadat een puck is afgeschoten verschijnt na een vaste tijdsinterval weer een nieuwe puck onderin het scherm.

Als een puck een pinguïn raakt zal de score worden opgehoogd. De totale score wordt aan het einde van het spel aan een highscore lijst toegevoegd. De pinguïn zal in het rond gaan draaien en van de baan glijden. Een mogelijke toevoeging is dat als een draaiende pinguïn een andere pinguïn raakt deze ook uitgeschakeld zal worden en een extra score wordt toegekend.

Als de speler alle pinguïns in een ronde weet af te weren wordt de volgende ronde gestart met een hogere moeilijkheidsgraad, maar een hogere score per geraakte pinguïn. Het spel is pas afgelopen als de speler al zijn levens verliest. Het doel is dus om zo lang mogelijk te overleven en een zo hoog mogelijke score te behalen.

1.3 Achievements

Achievements zijn extra doelen die in de game verwerkt zijn. Bij het behalen van zo een doel wordt de speler beloond met een embleem dat aan zijn profiel gekoppeld is. Dit zorgt voor meer uitdagingen en dus een hogere herspeelbaarheid van het spel.

Happy Feet
Raak een pinguïn met een andere pinguïn.

Strike
Raak x pinguïns binnen y seconden.*

March of the Penguins
Haal een totaalscore van x.*

Helping hand
Kaats een puck, raak een pinguïn.

Five
Voltooi level 5.

**details worden tijdens de ontwikkeling bepaald.*

2 Technisch Ontwerp

Op basis van het functioneel ontwerp is een technisch ontwerp gemaakt. Dit om vooraf vast te stellen met welke technieken het op te leveren product ontwikkeld zal worden, en hoe het technisch gestructureerd zal zijn.

2.1 Technieken

HTML + CSS
HTML wordt gebruikt voor het weergeven van het canvas waarbinnen de game gaat draaien. Ook zal het de CSS inladen die wordt gebruikt voor het bepalen van de achtergrondkleur en het inladen van benodigde lettertypen.

Javascript
Deze scripttaal zal worden gebruikt voor alle code die de game moet aandrijven. Ook wordt het gebruikt om de game elementen te tekenen op het HTML canvas. Dit laatste, samen met gebruikerinteractie met het canvas, wordt gedaan met behulp van het EaselJS framework. Dit framework maakt het werken met het canvas eenvoudiger door gecompliceerde javascript handelingen om te zetten naar functies die veel overeenkomen met die uit ActionScript 3. Ik zal het framework wel moeten aanpassen omdat het uit zichzelf alleen ondersteuning biedt voor MouseEvents waardoor alleen de muis wordt ondersteund en niet de touchscreen interface.

Daarnaast wordt gebruikgemaakt van het Nebula Framework/API die enkele functies specifiek voor het Nebula portaal beschikbaar maakt, waaronder de mogelijkheid om highscores op te slaan gekoppeld aan een Nebula gebruiker.

2.2 Klassendiagram

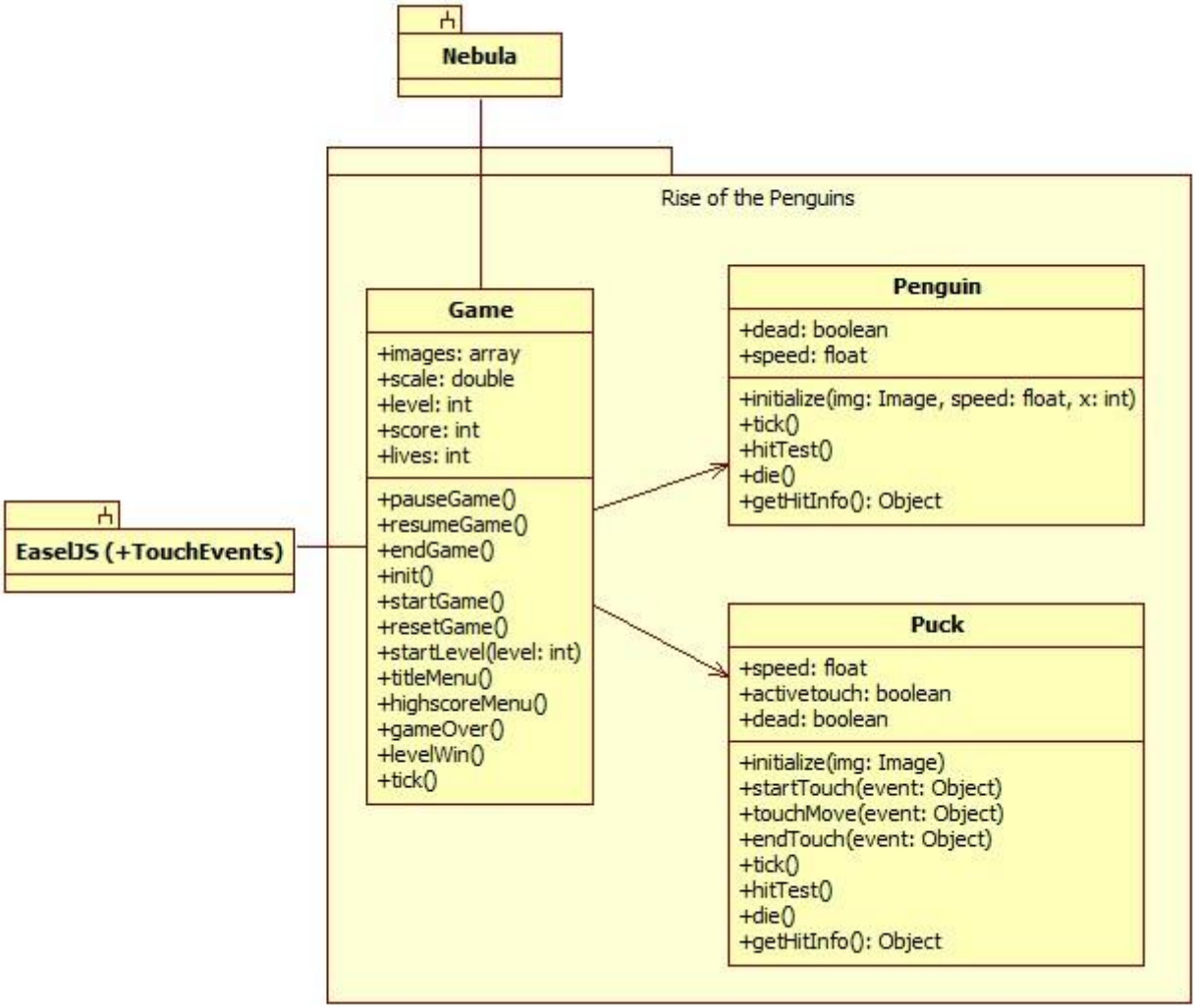
Om de technische structuur van de game te visualiseren en het zo overzichtelijk te maken voor het daadwerkelijke ontwikkelen is een klassendiagram ontworpen volgens de UML-standaard. Doordat volledig objectgeoriënteerd ontwikkelen helaas niet mogelijk is in Javascript zijn alle variabelen public.

Uitleg van de klassen
De **Game** klasse is de hoofdklasse en maakt de functies van het EaselJS en Nebula framework beschikbaar voor de hele game. De *pauseGame()*, *resumeGame()* en *endGame()* functies worden door het Nebula framework aangeroepen wanneer deze van de game verwacht dat die respectievelijk pauzeert, hervat of beëindigd.

De *tick()* functie wordt door het EaselJS framework ieder frame aangroepen en functioneert dus als de gameloop. Hierin wordt bijgehouden hoe het de speler vergaat en of deze een level heeft gewonnen of verloren. Ook wordt hier na een vastgesteld aantal frames een nieuwe instantie van de **Penguin** of **Puck** klasse gecreëerd.

Aan de **Penguin** klasse wordt naast het te gebruiken Image object een snelheid en eventueel een x positie meegegeven. Als deze laatste niet wordt meegegeven wordt een willekeurige positie bepaald. Ook hier wordt de *tick()* functie ieder frame aangeroepen. Hierin wordt aangegeven hoe de pinguïn zich moet verplaatsen ten opzichte van het vorige frame. Ook wordt hier de *hitTest()* functie aangeroepen. Hierin wordt gecontroleerd of het enig ander object van het type **Penguin** of **Puck** raakt en aangegeven hoe daarop gereageerd moet worden.

De **Puck** klasse heeft ook de *tick()* en *hitTest()* functie, maar hierin wordt alleen gecontroleerd op andere **Puck** objecten zodat het afkaatsen van twee pucks mogelijk wordt. De *startTouch()*, *touchMove()* en *endTouch()* functies worden aangeroepen als respectievelijk de vinger op het aanraakgebied wordt gelegd (of erop geklikt wordt), de vinger (of muisaanwijzer) over het scherm wordt bewogen, of de vinger van het scherm wordt gehaald (de muisknop wordt losgelaten). Hierin wordt bepaald welke beweging de puck mee moet krijgen.



Figuur 0.1 - Klassendiagram

Bijlage 4: Documentatie tweede game

Powered by Nebula

Hamaka
digitaal voor oplossing

Inhoudsopgave

1 Functioneel Ontwerp

Dit functioneel ontwerp is gemaakt om voor de ontwikkeling duidelijkheid te krijgen over wat het idee achter de game is en in welke setting deze zich afspeelt. Daarnaast wordt beschreven uit welke gameplay functionaliteiten de game zal bestaan. Tot slot wordt een lijst met achievements opgesteld die in de game te behalen zullen zijn.

1.1 Concept

Het uitgangspunt voor deze game is dat het gebruik moet maken van de geolocatie mogelijkheden van mobiele apparaten die ondersteund zijn in HTML5, hiermee is de fysieke locatie van de gebruiker te achterhalen als deze dat toestaat.

In een brainstorm sessie ontstond het idee voor een race-geïnspireerde game waarbij objecten op de weg ontweken moeten worden. Hierbij heeft de fysieke snelheid van de speler directe invloed op de gameplay waardoor de game uitdagender wordt om te spelen maar er een hogere score te behalen valt bij een hogere snelheid.

De game zal zich afspelen op een weg waarbij de speler besturing heeft over een auto.

1.2 Gameplay

Op basis van het concept zijn de gameplay functionaliteiten uitgedacht.

De snelheid waarmee de speler zich beweegt wordt gemeten en gebruikt om de basissnelheid te bepalen van de auto in de game. De auto zal zich vanzelf in een rechte lijn naar de rechterkant van het level bewegen.

Het level zal, buiten de weg waarom de auto zich bevind, bestaan uit een aantal obstakels en hulpmiddelen. Obstakels die geraakt worden met de auto zullen de snelheid van de auto verlagen. Bij bepaalde obstakels zal dit voor een tijdelijke duur zijn, bij andere geldt deze snelheidsverlaging voor de rest van het level. Ook zullen er enkele obstakels zijn die de auto volledig stoppen waarna het level opnieuw begonnen moet worden.

Hulpmiddelen zullen bestaan uit ‘snelheidsboosts’ die de snelheid van de auto zullen verhogen voor de rest van het level. Dit om te het voor de speler mogelijk te maken eerder gemaakte fouten te herstellen.

De speler kan zich door het level heen manoeuvreren door een vinger boven of onder de auto op het scherm te plaatsen. Hierop zal de auto in deze richting bewegen.

Aan het eind van elk level bevindt zich een finish waar, nadat de auto deze gepasseerd is, een score wordt berekend. Dit gebeurd op basis van de tijd die de speler over het level heeft gedaan, het level waarop gespeeld wordt en de snelheid waarmee de speler de finish passeert.

Dit betekent, aangezien de snelheid van de auto deels wordt bepaald door de fysieke snelheid van de speler, dat om een hoge score te behalen de speler in het echt ook snel moet gaan. De game wordt echter wel moeilijker bij een hoge snelheid omdat de obstakels ook sneller voorbij zullen komen waardoor deze lastiger te ontwijken zijn.

1.3 Achievements

Achievements zijn extra doelen die in de game verwerkt zijn. Bij het behalen van zo een doel wordt de speler beloond met een embleem dat aan zijn profiel gekoppeld is. Dit zorgt voor meer uitdagingen en dus een hogere herspeelbaarheid van het spel.

Watch the paint job
Voltooi level x zonder een obstakel te raken.*

Smile for the camera
Krijg de maximale snelheidsbonus bij de finish.

The Stig
Voltooi level x binnen y seconden.*

**details worden tijdens de ontwikkeling bepaald.*

2 Technisch Ontwerp

Op basis van het functioneel ontwerp is een technisch ontwerp gemaakt. Dit om vooraf vast te stellen met welke technieken het op te leveren product ontwikkeld zal worden, en hoe het technisch gestructureerd zal zijn.

2.1 Technieken

HTML + CSS
HTML wordt gebruikt voor het weergeven van het canvas waarbinnen de game gaat draaien. Ook zal het de CSS inladen die wordt gebruikt voor het bepalen van de achtergrondkleur en het inladen van benodigde lettertypen.

Javascript
Deze scripttaal zal worden gebruikt voor alle code die de game moet aandrijven. Hierbij zal gebruikt gemaakt worden van de nieuwe Geolocation functionaliteiten waarmee de lengte- en breedtegraad van de gebruiker opgevraagd kan worden.

Ook wordt de scripttaal gebruikt om de game elementen te tekenen op het HTML canvas. Dit laatste, samen met gebruikerinteractie met het canvas, wordt gedaan met behulp van het EaselJS framework (met toegevoegde touch ondersteuning). Dit framework maakt het werken met het canvas eenvoudiger door gecompliceerde Javascript handelingen om te zetten naar functies die veel overeenkomen met die uit ActionScript 3.

Daarnaast wordt gebruikgemaakt van het Nebula Framework/API die enkele functies specifiek voor het Nebula portaal beschikbaar maakt, waaronder de mogelijkheid om highscores op te slaan gekoppeld aan een Nebula gebruiker.

2.2 Klassendiagram

Om de technische structuur van de game te visualiseren en het zo overzichtelijk te maken voor het daadwerkelijke ontwikkelen is een klassendiagram ontworpen volgens de UML-standaard. Doordat volledig objectgeoriënteerd ontwikkelen helaas niet mogelijk is in Javascript zijn alle variabelen public.

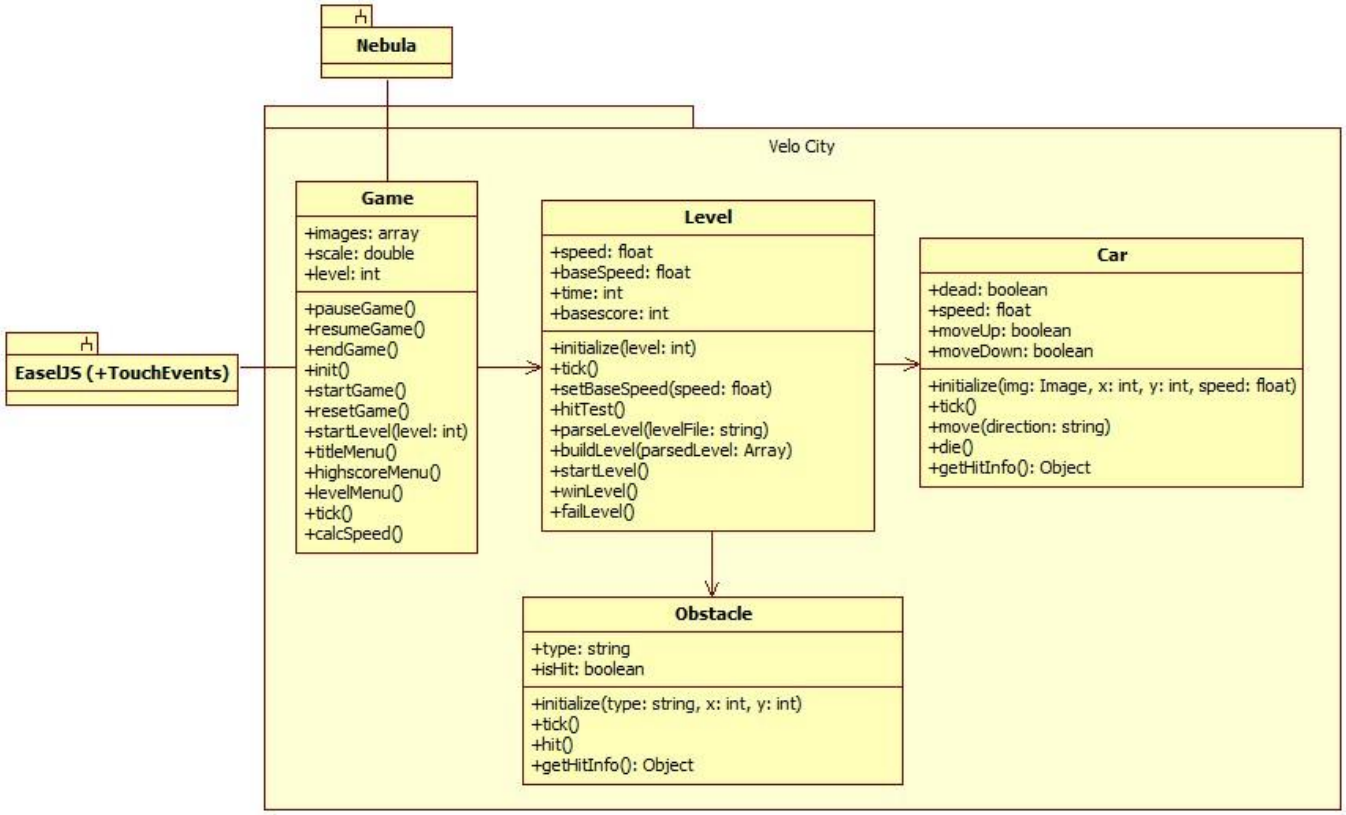
Uitleg van de klassen
De **Game** klasse is de hoofdklasse en maakt de functies van het EaselJS en Nebula framework beschikbaar voor de hele game. De *pauseGame()*, *resumeGame()* en *endGame()* functies worden door het Nebula framework aangeroepen wanneer deze van de game verwacht dat die respectievelijk pauzeert, hervat of beëindigd.

De *tick()* functie wordt door het EaselJS framework ieder frame aangroepen en functioneert dus als de gameloop. *calcSpeed()* zal, als de gebruiker het toestaat, positieverranderingen opvragen, waarna wordt berekend wat de snelheid van de gebruiker is.

In *startLevel()* wordt een nieuw **Level** object aangemaakt. In deze klasse wordt een bestand geladen die de informatie van het betreffende level bevat. Met *parseLevel()* wordt dit bestand omgezet in een bruikbaar formaat waarna dit wordt gebruikt in *buildLevel()* om het level op te bouwen. Hierbij wordt ook een **Car** object en de benodigde **Obstacle** objecten aangemaakt. Ook hier wordt de *tick()* functie ieder frame aangeroepen. Daarin wordt bijgehouden hoe de speler er voorstaat en of deze het level uitgespeeld heeft, of dat het level vroegtijdig beëindigd moet worden. Ook wordt vanuit hier de *hitTest()* functie aangeroepen. Deze controleert of het **Car** object van het level een van de andere objecten raakt en roept de juiste vervolgstappen aan.

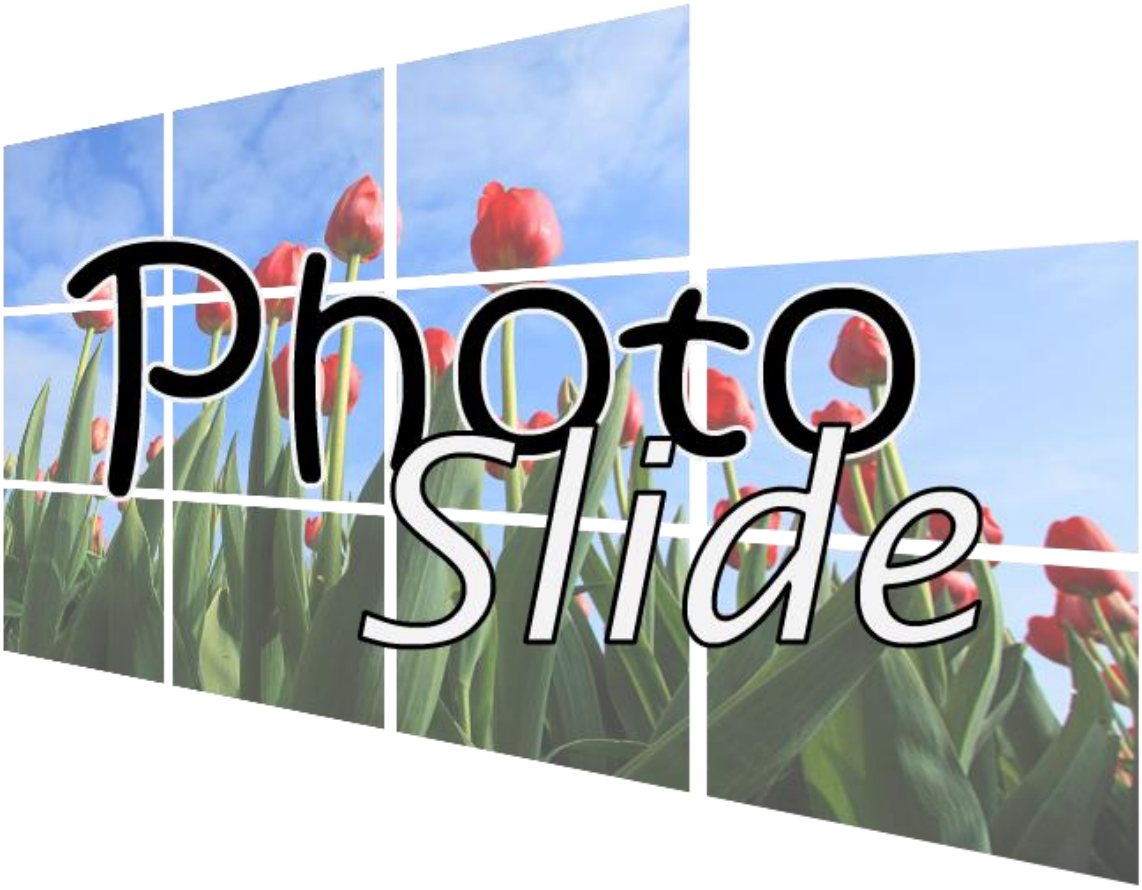
De **Car** klasse bevat de functie *move()* waarmee de speler het object naar de boven- of onderkant van het scherm kan bewegen. De *tick()* functie houdt zich ieder frame bezig met het tekenen van het object op de juiste positie.

De **Obstacle** klasse krijgt een type mee, waarmee bepaald wordt wat de gevolgen zullen zijn als het object wordt geraakt en hoe het object weegegeven moet worden. Verder heeft het ook een *tick()* functie die zorgt voor de visualisatie van het object.



Figuur 0.2 - Klassendiagram

Bijlage 5: Documentatie derde game



Powered by Nebula

Hamaka
digitaal voor oplossing

Inhoudsopgave

1 Functioneel Ontwerp

Dit functioneel ontwerp is gemaakt om voor de ontwikkeling duidelijkheid te krijgen over wat het idee achter de game is en in welke setting deze zich afspeelt. Daarnaast wordt beschreven uit welke gameplay functionaliteiten de game zal bestaan. Tot slot wordt een lijst met achievements opgesteld die in de game te behalen zullen zijn.

1.1 Concept

Tijdens een brainstormsessie is het idee ontstaan om een schuifpuzzel te ontwikkelen waarbij het voor de gebruiker mogelijk is om zelf een foto te uploaden. Deze foto zal dan opgedeeld worden in vakjes welke vervolgens gebruikt kunnen worden in de schuifpuzzel.

1.2 Gameplay

Op basis van het concept zijn de gameplay functionaliteiten uitgedacht.

De speler krijgt eerst de mogelijkheid om een afbeelding te uploaden of een standaard afbeelding te kiezen. Daarna kan gekozen worden tussen drie moeilijkheidsgraden. Vervolgens wordt de gekozen afbeelding opgedeeld in 9, 16 of 25 vlakken, afhankelijk van de moeilijkheidsgraad.

Hiervan wordt het vlak rechtsonder weggehaald en worden de overige vlakken in willekeurige volgorde geplaatst. De speler moet vervolgens de vlakken zo schuiven zodat de originele afbeelding weer gevormd wordt. Dit gebeurd op de zelfde manier als bij een standaard schuifpuzzel.

Iedere moeilijkheidsgraad heeft een vastgesteld maximum aantal punten. Na het minimum aantal stappen die nodig zijn om de puzzel op te lossen wordt voor iedere gemaakte stap punten van dit maximum afgetrokken. Als de puzzel is opgelost wordt de behaalde score getoond.

Het doel voor de speler is dus om een puzzel op te lossen in zo min mogelijk stappen.

1.3 Achievements

Achievements zijn extra doelen die in de game verwerkt zijn. Bij het behalen van zo een doel wordt de speler beloond met een embleem dat aan zijn profiel gekoppeld is. Dit zorgt voor meer uitdagingen en dus een hogere herspeelbaarheid van het spel.

A*
De maximale score behaald van een moeilijkheidsgraad.

Time passer
Los een moeilijke puzzel op.

**details worden tijdens de ontwikkeling bepaald.*

2 Technisch Ontwerp

Op basis van het functioneel ontwerp is een technisch ontwerp gemaakt. Dit om vooraf vast te stellen met welke technieken het op te leveren product ontwikkeld zal worden, en hoe het technisch gestructureerd zal zijn.

2.1 Technieken

HTML + CSS
HTML wordt gebruikt voor het weergeven van het canvas waarbinnen de game gaat draaien. Ook zal het de CSS inladen die wordt gebruikt voor het bepalen van de achtergrondkleur en het inladen van benodigde lettertypen.

Javascript
Deze scripttaal zal worden gebruikt voor alle code die de game moet aandrijven.

Ook wordt de scripttaal gebruikt om de game elementen te tekenen op het HTML canvas. Dit laatste, samen met gebruikerinteractie met het canvas, wordt gedaan met behulp van het EaselJS framework (met toegevoegde touch ondersteuning). Dit framework maakt het werken met het canvas eenvoudiger door gecompliceerde Javascript handelingen om te zetten naar functies die veel overeenkomen met die uit ActionScript 3.

Daarnaast wordt gebruikgemaakt van het Nebula Framework/API die enkele functies specifiek voor het Nebula portaal beschikbaar maakt, waaronder de mogelijkheid om highscores op te slaan gekoppeld aan een Nebula gebruiker.

PHP
PHP zal gebruikt worden om afbeeldingen van een gebruiker naar de server te uploaden zodat deze in de game gebruikt kunnen worden.

2.2 Klassendiagram

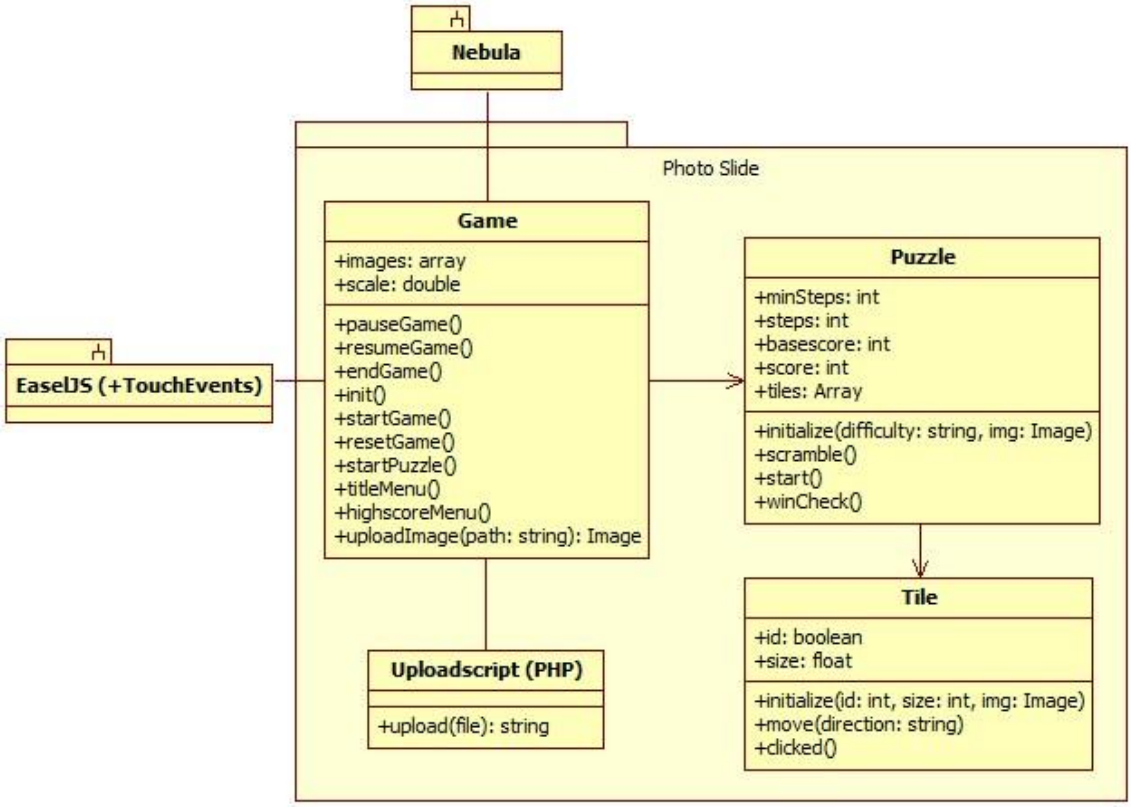
Om de technische structuur van de game te visualiseren en het zo overzichtelijk te maken voor het daadwerkelijke ontwikkelen is een klassendiagram ontworpen volgens de UML-standaard. Doordat volledig objectgeoriënteerd ontwikkelen helaas niet mogelijk is in Javascript zijn alle variabelen public.

Uitleg van de klassen
De **Game** klasse is de hoofdklasse en maakt de functies van het EaselJS en Nebula framework beschikbaar voor de hele game. De *pauseGame()*, *resumeGame()* en *endGame()* functies worden door het Nebula framework aangeroepen wanneer deze van de game verwacht dat die respectievelijk pauzeert, hervat of beëindigd.

In de *startPuzzle()* functie wordt de *uploadImage()* functie aangeroepen die een door de speler gekozen afbeelding zal uploaden met behulp van een PHP script. De functie zal met het nieuwe serverpad naar de afbeelding een Image object maken en deze teruggeven. Deze wordt samen met de gekozen moeilijkheidsgraad meegegeven bij het aanroepen van de **Puzzle** klasse.

In de **Puzzle** klasse worden de benodigde **Tile** objecten aangemaakt. Deze worden na plaatsing met de *scramble()* functie in willekeurige volgorde gezet. Na iedere stap die de speler neemt wordt met de *winCheck()* functie bekeken of de huidige positie van de **Tile** objecten de juiste is. Als dit het geval is wordt dit weergegeven samen met de behaalde score.

De **Tile** klasse wordt geïnitialiseerd met een id, grote en Image object. Het id geeft samen met de grote meteen aan wat de positie van de tile hoort te zijn (bijv. 0 is altijd linksboven, 1 daarnaast, etc.). Met deze informatie kan worden bepaald welk gedeelte van het Image object zichtbaar moet zijn voor dit vlak. De *clicked()* functie wordt aangeroepen als een speler het vlak aanklikt en bepaald of een verplaatsing mogelijk is en zo ja in welke richting. Hierop wordt de *move()* functie aangeroepen die met een animatie het valk zal verplaatsen in de meegegeven richting.



Figuur 0.3 - Klassendiagram