Inhoud

[USERSTORIES 1](#_Toc414969406)

[Spel instructies 2](#_Toc414969407)

[Verslag 3](#_Toc414969408)

# Userstories

**Must have**

* Als speler wil ik het karakter kunnen bewegen met de pijltjes toetsen.
* Als speler wil ik kunnen richten en schieten met de muis.
* Als speler wil ik vijanden kunnen neerschieten.
* Als speler wil ik een hi-score kunnen halen.

**Should have**

* Als speler wil ik een karakter kunnen spelen.
* Als speler wil ik dat mijn vijanden er gevarieerd uitzien.

**Could have**

* Als speler wil ik uit een karakter kunnen kiezen.
* Als speler wil ik kunnen kiezen uit meerdere wapens.
* Als speler wil ik meerdere levels kunnen kiezen.
* Als speler wil ik mijn karakter kunnen aanpassen.

# Spel instructies

Beweeg met de pijltjes toetsen, of met de WASD toetsen.

Links/A: Beweeg de speler naar links.

Rechts/D: Beweeg de speler naar rechts.

Boven/W: Beweeg de speler naar boven.

Onder/S: Beweeg de speler naar onder.

Linkermuisknop: Schiet op de plek van de cursor.

Schiet de vijanden af door op ze te klikken met de muis.

# Verslag

**1 Toelichting op het framework:**

Backbone is een Javascript library en gebaseerd op het Model–view–presenter ontwerppatroon.

**2 De architectuur van het framework. Is het traditionele MVC? MVVM? Leg uit met behulp van een schema**

De architectuur is gebaseerd op het MVP (Model–view–presenter ontwerppatroon). Het verschilt van het MVC (Model View Controller) ontwerppatroon. Het MVP model gebruikt een presenter in plaats van een controller. In Models staat data en hoe het moet reageren in de gebruikers interface. De View is een passive interface dat data weergeeft en gebruikersacties door stuurt naar de Presenter. Een presenter update de View en handelt events van de gebruiker af. Hij update ook het model en ontvangt er events van.



**3 Een beschrijving van de game die jullie willen gaan uitwerken.**

Wij willen een top down schietspel gaan maken. Je kan het karakter besturen met de pijltjes of de WASD toetsen. Je kan in 4 richtingen bewegen. Je kan schieten door met je muis ergens te klikken. Het karakter schiet dan een lijn vanuit zijn positie naar de positie waar je klikt met je muis. Je moet de vijand raken met je muis om hem neer te schieten.



Het is de bedoeling dat je tegenstanders neer schiet. De vijanden blijven spawnen en worden steeds sneller. Als je geen levens meer hebt ben je dood.

**4 Een globale uitleg van een mogelijke aanpak om deze game te ontwikkelen binnen dit framework.**

We maken een gameobject dat door elk gameobject wordt overgeërfd bijvoorbeeld door de speler of de vijand. De game object een is een model. Elk model krijgt een view toegewezen die verantwoordelijk is het voor het tekenen van dit model. Deze view wordt gegenereerd bij het eerste verzoek voor het renderen en is dus uniek aan de model.

**5 Laat zien hoe je hierbij de Model, View, Controllers en Viewmodels gebruikt. Geef code voorbeelden! Illustreer hoe je onderdelen aan kan pakken zoals:**

Elk game object in het spel is een Model in ons spel. Deze hebben eigenschappen als hun breedte, hoogte en locatie. Deze gameobjecten worden aan een collectie met boxes gezet. Deze collectie kan dan gebruikt worden om te iteraten door alle aanwezige objecten.

De view is verantwoordelijk voor het tekenen van de objecten, per model wordt een view gegenereerd door de hoofd View. De Hoofd View regelt het renderen van alle objecten, en subscribed ook aan eventhandlers van collecties zodat hij zichzelf update zodra er iets veranderd in één van de collecties.

Er is ook een Model gemaakt voor de GameController. Deze is verantwoordelijk voor het bijhouden van de loop van het spel, en hier kunnen andere objecten zich aan koppelen om hun eigen functies uit te voeren per game tick.

Tot slot wordt de userinput apart geregeld, er is gesubscribed aan de ‘keyup’, ‘keydown’ en ‘click’ events. Deze events geven ons de informatie als de user input invoert. Onze code toggled het bewegen van de speler met behulp van keyup en keydown. Keyup zet de beweging aan, en keydown zet de beweging uit. De GameController regelt dan dat de speler daadwerkelijk gaat bewegen.

**6 Wat kan er nog meer met het framework?**

Buiten de al eerder aangegeven Model/View/Collections bied Backbone ook nog Routers. Deze regelen op basis van ingevoerde URLs wat er ingeladen moet worden op de pagina. Het is vergelijkbaar met routers in het MVC model van ASP.NET, maar dan voor Javascript. Ook bied Backbone een groot aantal events aan die automatisch getriggerd worden bij bepaalde actie. Zo wordt er bij een Model bijvoorbeeld een event getriggerd als er een attribuut veranderd. Maar als dit Model ook onderdeel is van een collectie, wordt er bij die Collectie ook gelijk een event getriggerd. Hierdoor wordt de Backbone library heel efficiënt in zijn gebruik en kun je heel makkelijk code schrijven voor meerdere models met behulp van Collecties.

**7 Jullie persoonlijke mening**

Wij zijn beiden erg fan van talen als C# en Java. Het OO-programeren spreekt ons heel erg aan ook door het typeren van je variabelen. Deze restricties vinden wij prettig omdat je voor variabelen bijvoorbeeld direct weet dat het óf null is, óf de goede waarde bevat. Deze zekerheid heb je niet in talen als Javascript en PHP.

Het is wel prettig dat talen als Javascript proberen om ook modellen uit OO-concepten te implementeren. Maar wij zijn van mening dat je beter het echte OO kan hebben dan de ‘namaak’ versies.

Het MVC-model implementeren was in de opdracht ook niet echt relevant, de game was zo klein van schaal dat het maken van een MVC model meer werk was dan het spel zelf. We vonden het echter wel erg fijn dat Backbone allemaal events implementeerde waar wij simpelweg onze functie aan konden koppelen zodat we niet zelf een groot aantal koppelingen moesten maken. Zo hebben wij zo’n koppeling gemaakt dat elke keer als een collectie geüpdatet werd, dat de View zichzelf opnieuw ging tekenen. Dat was wel hele efficiënte code voor ons.

**8 De bronnen die je hebt geraadpleegd**

[**http://todomvc.com/**](http://todomvc.com/)

[**http://backbonejs.org/**](http://backbonejs.org/)

[**http://www.w3schools.com/**](http://www.w3schools.com/)

[**https://simon.html5.org/dump/html5-canvas-cheat-sheet.html**](https://simon.html5.org/dump/html5-canvas-cheat-sheet.html)

[**http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93presenter**](http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93presenter)

[**http://arunspdreamz.blogspot.nl/2010/03/design-patterns-for-aspnet-developers.html**](http://arunspdreamz.blogspot.nl/2010/03/design-patterns-for-aspnet-developers.html)

[**http://stackoverflow.com/questions/10716478/making-a-backbone-js-view-to-draw-objects-on-a-canvas**](http://stackoverflow.com/questions/10716478/making-a-backbone-js-view-to-draw-objects-on-a-canvas)

[**http://backbonetutorials.com**](http://backbonetutorials.com)