

Analysedocument

Individuele opdracht

Michelle Broens | S21M | 2017

Inhoudsopgave

[Inleiding 2](#_Toc477336777)

[Begrippen 3](#_Toc477336778)

[Opdrachtomschrijving 4](#_Toc477336779)

[Scenario 5](#_Toc477336780)

[Functionele eisen 6](#_Toc477336781)

[Mechanic systeem 6](#_Toc477336782)

[Level systeem 7](#_Toc477336783)

[Niet-functionele eisen 8](#_Toc477336784)

[Use case 9](#_Toc477336785)

[Use case diagram 9](#_Toc477336786)

[Use case beschrijvingen 10](#_Toc477336787)

[User interface 14](#_Toc477336788)

[User interface navigatie 14](#_Toc477336789)

[User interface schetsen 15](#_Toc477336790)

[Start screen 15](#_Toc477336791)

[Loading screen 15](#_Toc477336792)

[Main screen 16](#_Toc477336793)

[Save screen 16](#_Toc477336794)

[Journal screen 17](#_Toc477336795)

[Skill screen 17](#_Toc477336796)

[World map 18](#_Toc477336797)

[Inventory 18](#_Toc477336798)

[EER model 0](#_Toc477336799)

[EER bijlage 0](#_Toc477336800)

[Productdecompositie 0](#_Toc477336801)

# Inleiding

In dit document komen er een aantal verschillende onderwerpen aanbod die te maken hebben met de analysefasen van het maken van een applicatie over … De onderwerpen die aan bod komen zijn:

* Een lijst met begrippen: in deze lijst staan woorden die de lezer misschien niet begrijpt of misschien niet helemaal begrijpt.
* Opdrachtomschrijving: in dit stuk komt een beschrijving van wat de opdracht inhoud, dus wat het concept inhoud en wat de rest van de opdracht zou zijn.
* Het scenario: een korte beschrijving over een persoon die door de ontwikkelde applicatie loopt.
* De functionele eisen: de eisen die het programma moet kunnen, bijvoorbeeld een speler kan bewegen.
* De niet-functionele eisen: de eisen die niet te maken hebben met het programmeren van code zoals de functionele eisen, bijvoorbeeld de code moet herbruikbaar zijn.
* De use cases: de functionele eisen worden in een diagram gezet zodat het duidelijk wordt door wie deze functie uitgevoerd wordt. Deze functies worden in de use case beschrijving van start tot eind doorgelopen.
* De userinterface: hier wordt een schets gemaakt van hoe de applicatie er ongeveer uit gaat zien.
* Het EER model: een schets van hoe de structuur van het programma er in de database uit gaat zien.
* De productdecompositie: een schets van de verschillende onderdelen die opgeleverd worden.
* Planning: de planning van hoe dit project uitgewerkt wordt. Per onderdeel staat een tijd van wanneer er een onderdeel begonnen wordt en wanneer deze af is.

# Begrippen

Dit zijn begrippen die de opdrachtgever niet meteen zal begrijpen als hij/zij dit document zal lezen.

|  |  |
| --- | --- |
| **Begrip** | **Uitleg** |
| **Use case** | In software engineering is een use case een lijst van acties die een persoon (actor) en het systeem uitvoert om een doel te bereiken, bijvoorbeeld het inloggen. |
| **Userinterface** | Hier praten de gebruiker en het systeem met elkaar. |
| **Database** | Een georganiseerde collectie van data in tabellen. |
| **AFK** | Een afkorting voor het woord afkorting. |
| **MoSCoW** | Een methode om aan te geven hoe belangrijk een requirement is, met de M voor must(moet), S voor shoulds(zou moeten), C voor could(zou kunnen) en W voor will not(wordt niet geïmplmenteerd). |
| **Topdown** | Een game waarbij het perspectief van boven ligt. |
| **RPG** | Dit is een genre in videogames, hier beslist de speler over acties van de speler |
| **Quest** | Een doel die je moet behalen in de game |
| **Combat** | Het gevecht van de speler tegen een vijand of meerdere vijanden |
| **Unity** | Unity is een game development platform waar je games kan ontwikkelen voor op verschillende platforms. |
| **Sprite** | Een entiteit die zich beweegt op het computerscherm |
| **Texture** | De samenhang van een oppervlakte |
| **Slot** | Een plaats waar iets ingeschoven kan worden |
| **AI** | Artificial intelligence, dus code die een patroon uitvoert |

# Opdrachtomschrijving

De game wordt een singleplayer topdown 45º 2D game. Ik wil een RPG maken met een kleine verhaallijn en een aantal quests. Wat ik wil gaan doen voor de combat is dat een character gewoon aan kan vallen en ook aan kan vallen met skills. Ik wil ook een klein beetje een verhaal schrijven door middel van de quests, ik ben van plan alleen een main quest te maken. Ik wil zoveel mogelijk textures en sprites zelf maken, maar als ik bij het programmeren dan tijd te kort kom laat ik dat vallen. Als eerst wil ik een sprite maken of ergens van het internet halen en deze laten lopen, hierna komt het aanvallen en daarna het gebruiken van skills. Deze volgorde telt ook voor de vijand. Het verhaal wat ik hierin ga brengen word een basic verhaallijn om ervoor te zorgen dat de focus van het maken van de game hier niet op komt te liggen. Dus de spelers’ character is een held die al zijn herinneringen kwijt is, dus moet hij al zijn kracht terugkrijgen door weer vanaf het begin te beginnen met levelen en zijn herinneringen terug te krijgen. Naarmate hij meer levelt krijgt hij meer kennis terug over zijn verleden. Ik wil in het begin ongeveer vijf levels maken zodat er een stijgende moeilijkheidsgraad in zit en ik deze in het begin ook kan testen, dit kan altijd uitgebreider worden.

Een speler heeft een hand-, armor-, leg-, en bootslot. Hier kan hij verschillende uitrusting aan koppelen. In het begin krijgt een speler een standaarduitrusting, naarmate hij meer quests doet zal hij ook betere uitrusting krijgen.

# Scenario

De speler start het spel en komt op het hoofdscherm, hier kan hij kiezen tussen “Start Game”, “Exit Game” en “Load Game”. Als de speler op “Exit Game” klikt sluit de applicatie. Als de speler op “Load Game” klikt wordt zijn laatste save file geladen. Als de speler op “Start Game" klikt start hij een nieuw spel. Er wordt dan eerst een klein stukje dialoog getoond. Daarna volgt een tutorial die de speler moet volgen. Hierin wordt uitgelegd hoe je aan moet vallen, met welke knoppen je moet lopen en op welke knoppen je skills zitten. Hierna komt de speler in level 1. Hier komt hij een aantal enemies tegen die verslagen moeten worden van de quest die gegeven wordt aan het begin van het gebied. Als de speler vijf quests in dit gebied heeft afgerond mag hij naar het volgende gebied, dit gaat zo door tot hij bij level vijf komt, dit gebied heeft zes quests waarvan de laatste quest een boss fight is. Als de speler deze boss verslagen heeft komt er “Victory” op het scherm te staan. Hierna wordt de speler teruggestuurd naar het hoofdscherm. En kan de speler weer vanaf het begin aan beginnen.

# Functionele eisen

Dit hoofdstuk gaat over de functionele requirements die geïmplementeerd gaan worden in de applicatie. Deze worden beschreven met een korte beschrijving, aangeduid hoe belangrijk deze requirement is met de MoSCoW analyse en een eventuele opmerking.

## Mechanic systeem

|  |  |  |  |
| --- | --- | --- | --- |
| **AFK** | **Requirement** | **MoSCoW** | **Opmerking** |
| **MSR1** | De speler moet kunnen lopen | M |  |
| **MSR2** | De speler moet kunnen aanvallen | M |  |
| **MSR3** | De vijand moet kunnen lopen | M |  |
| **MSR4** | De vijand moet kunnen aanvallen | M |  |
| **MSR5** | De speler moet skills hebben | M |  |
| **MSR6** | De speler moet skills kunnen gebruiken | S |  |
| **MSR7** | De speler heeft items | M |  |
| **MSR8** | De speler kan items gebruiken | S |  |
| **MSR8** | De vijand heeft skills | S |  |
| **MSR9** | De vijand kan skills gebruiken | S |  |
| **MSR10** | De speler kan een wapen uitrusten | S |  |
| **MSR11** | De speler kan zijn uitrusting veranderen | S |  |
| **MSR12** | De vijand kan een item laten vallen | S |  |

## Level systeem

|  |  |  |  |
| --- | --- | --- | --- |
| **AFK** | **Requirement** | **MoSCoW** | **Opmerking** |
| **LSR1** | De vijand heeft een level | M |  |
| **LSR2** | De speler heeft een level | M |  |
| **LSR3** | De speler kan levelen | S | Als deze requirement niet geïmplementeerd wordt dan wordt het level van de speler in de database bepaald |
| **LSR4** | Er moeten meerdere enemies in één level komen | S | Bij dit level wordt het level bedoelt waar de speler op speelt |
| **LSR5** | De speler moet de game kunnen opslaan | S |  |

# Niet-functionele eisen

|  |  |  |  |
| --- | --- | --- | --- |
| **AFK** | **Requirement** | **MoSCoW** | **Opmerking** |
| **NFR1** | Code moet herbruikbaar zijn | M |  |
| **NFR2** | De code moet goed leesbaar zijn | M |  |

# Use case

In dit hoofdstuk komt het use case diagram en de beschrijvingen die bij het diagram horen.

## C:\Users\Gebruiker\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Use case diagram - Page 1.pngUse case diagram

## Use case beschrijvingen

In dit hoofdstuk komen de beschrijvinging die bij de use case diagram van de vorige pagina horen. Ik heb echter de use case die de NPC heeft niet verder uitgewerkt omdat de code die hier later onder komt veel gaat lijken op de speler, alleen wordt hiervoor alles automatisch gedaan.

|  |  |
| --- | --- |
| **Naam** | De speler moet kunnen lopen |
| **Samenvatting** | De speler moet door op de WASD toetsen in de toegewezen richtingen kunnen bewegen |
| **Actor** | De speler |
| **Aanname** | De actor heeft op “Game Start” of “Load Game” geklikt en zit in de game |
| **Beschrijvingen** | 1. De speler klikt op een van de “WASD” knoppen 2. Het systeem beslist of deze knop correspondeert met een van de richtingen die de actor heeft ingevoerd 3. De sprite op het scherm loopt de richting op die correspondeert met de knop die de actor heeft ingedrukt [1] |
| **Uitzondering** | [1] Dit kan herhaald worden tot de speler bijvoorbeeld in een muur loopt |
| **Resultaat** | De actor is een kant opgelopen naar de knop die is ingedrukt |

|  |  |
| --- | --- |
| **Naam** | De speler moet kunnen aanvallen |
| **Samenvatting** | De speler moet kunnen aanvallen door op de rechtermuisknop te klikken |
| **Actor** | De speler |
| **Aanname** | De actor heeft op “Game Start” of “Load Game” geklikt en zit in de game |
| **Beschrijvingen** | 1. De actor ziet een vijand voor zich staan en klikt op de rechtermuisknop 2. Het systeem berekent het aantal schade wat de speler op de vijand doet 3. Er gaat naar het aantal schade wat de speler doet HP van de vijand af [1] |
| **Uitzondering** | [1] De speler heeft zijn aanval gemist, dus zal hij geen schade doen |
| **Resultaat** | De speler heeft een vijand succesvol aangevallen |

|  |  |
| --- | --- |
| **Naam** | De speler kan skills gebruiken |
| **Samenvatting** | De speler kan op de knoppen 1,2 en 3 drukken deze zitten gebonden aan skills en zullen gebruikt worden zodra de speler op de toegewezen knop drukt |
| **Actor** | De speler |
| **Aanname** | De actor heeft een skill |
| **Beschrijvingen** | 1. De actor klikt op een van de toegewezen knoppen 2. Het systeem beslist welke skill dit is en de schade wat deze zal doen op de vijand [1] 3. De vijand neemt schade naar behoren van de berekening [2] |
| **Uitzondering** | [1] De skill die de speler gebruikt is een skill die een speler zal helen, dus zal het woord schade vervangen worden met helen  [2] Als er een skills is gebruikt die een speler zal helen, wordt de speler hier geheeld worden in plaats van dat een vijand damage krijgt |
| **Resultaat** | De speler heeft een skill gebruikt tegen een vijand |

|  |  |
| --- | --- |
| **Naam** | De speler kan zijn wapen uitrusten |
| **Samenvatting** | De speler kan zijn wapen veranderen en er een uitrusten naar zijn wensen |
| **Actor** | Speler |
| **Aanname** | De speler heeft een wapen in zijn inventaris |
| **Beschrijvingen** | 1. De speler opent zijn inventaris 2. Het systeem laat voor de speler zijn voorwerpen zien waarover hij beschikt 3. De speler klikt met de linkermuisknop op het gewenste wapen wat hij wil uitrusten 4. Het systeem zorgt ervoor dat dit wapen uit de inventaris gehaald zal worden en geplaatst wordt in de spelers hand [1] |
| **Uitzondering** | [1] Het wapen wat hij wil uitrusten is te hoog level voor de speler |
| **Resultaat** | De speler heeft een ander wapen uitgerust |

|  |  |
| --- | --- |
| **Naam** | De speler kan zijn uitrusting veranderen |
| **Samenvatting** | De speler kan zijn hand slot, armor slot, leg slot en zijn boot slot veranderen. |
| **Actor** | De speler |
| **Aanname** | De speler heeft meerde uitrusting die hij een slot kan stoppen |
| **Beschrijvingen** | 1. De speler opent zijn inventaris 2. Het systeem laat voor de speler zijn voorwerpen zien waarover hij beschikt 3. De speler klikt met de linkermuisknop op het gewenste wapen wat hij wil uitrusten 4. Het systeem zorgt ervoor dat de gekozen uitrusting uit de inventaris gehaald zal worden en geplaatst wordt op de speler [1] |
| **Uitzondering** | [1] De uitrusting die hij aan wil doen is te hoog level voor de speler |
| **Resultaat** | De speler heeft zijn uitrusting veranderd |

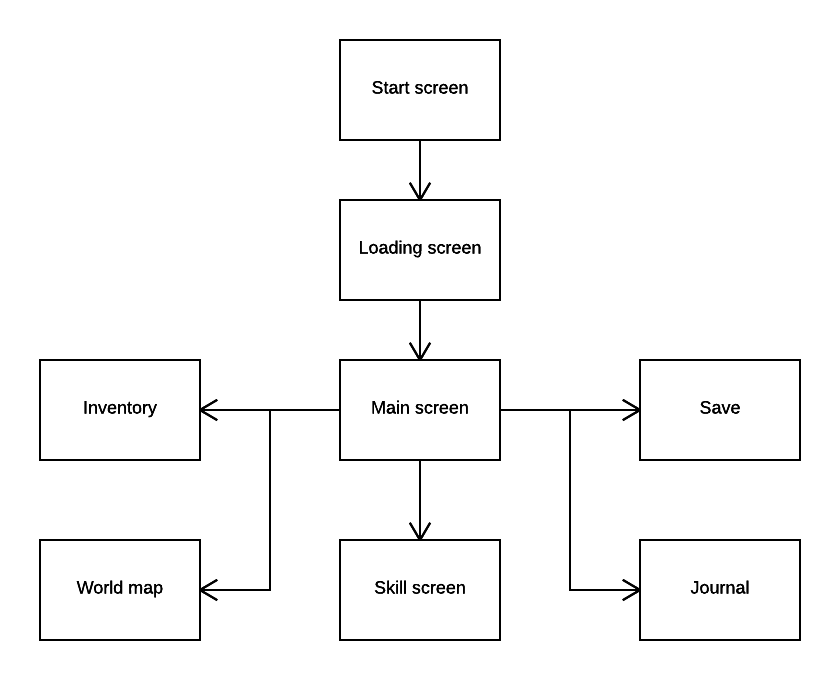
|  |  |
| --- | --- |
| **Naam** | De speler kan levelen |
| **Samenvatting** | De speler kan levels erbij krijgen naarmate hij meer vijanden heeft verslagen en meer quest heeft voltooid |
| **Actor** | De speler |
| **Aanname** | De speler heeft het tutorial voltooid |
| **Beschrijvingen** | 1. De speler verslaat een vijand of voltooid een quest 2. Het systeem verwerkt dit gekregen EXP, telt dit bij elkaar op en na een aantal EXP krijgt de speler een nieuw level erbij 3. De speler krijgt een popup met “Level up” op zijn scherm |
| **Uitzondering** | N.v.t. |
| **Resultaat** | De speler heeft er een level bijgekregen |

|  |  |
| --- | --- |
| **Naam** | De speler kan zijn game opslaan |
| **Samenvatting** | De speler heeft een bepaald punt waar hij zijn vooruitgang in het spel kan opslaan |
| **Actor** | De speler |
| **Aanname** | De speler is |
| **Beschrijvingen** | 1. De speler komt bij een save point en activeert deze door op E te drukken 2. Het systeem slaat deze data op in de database 3. De speler krijgt een popup met de tekst “Save completed” |
| **Uitzondering** | N.v.t. |
| **Resultaat** | De speler heeft zijn vooruitgang opgeslagen |

# User interface

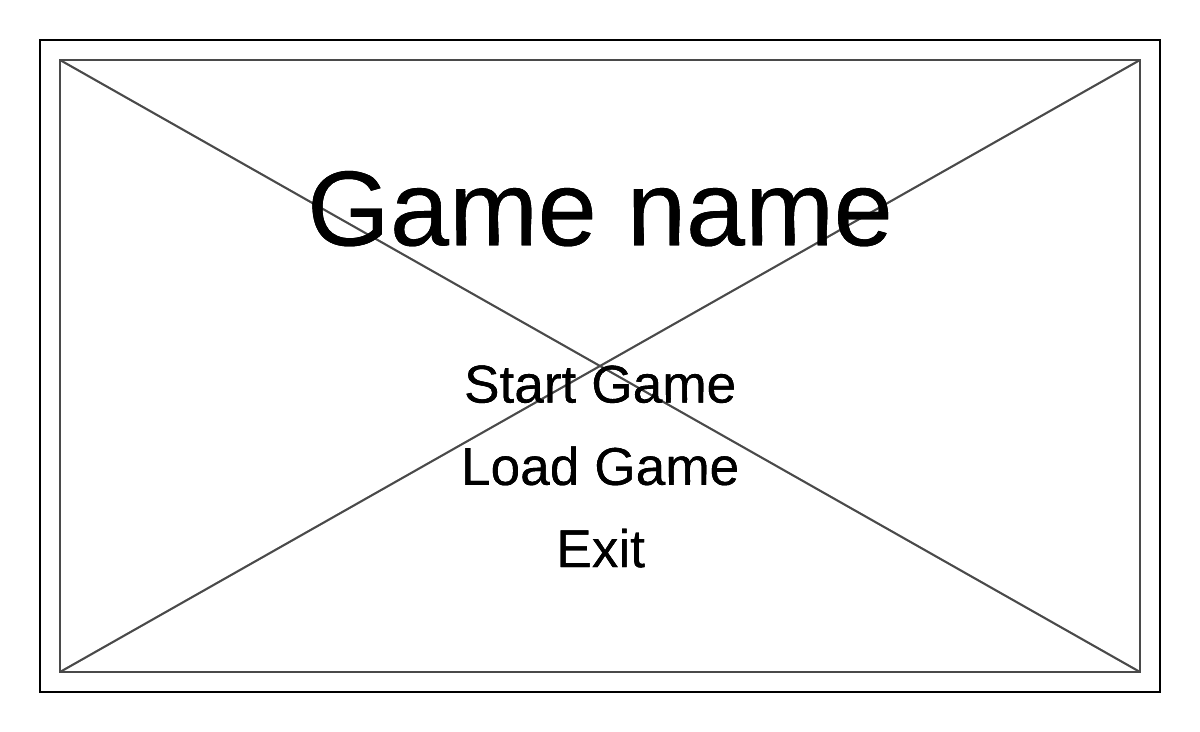
In dit hoofdstuk komen een paar designs voor hoe de applicatie er van de buitenkant uit komt te zien.

## User interface navigatie



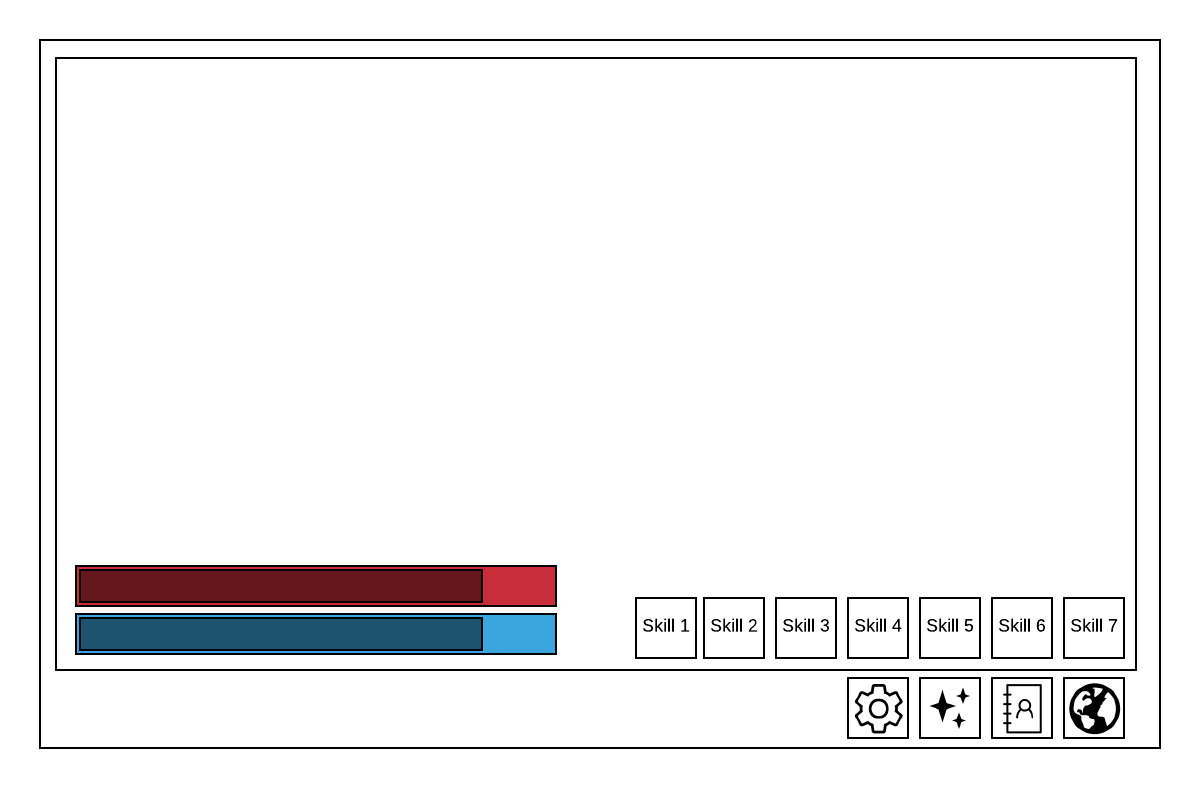
## User interface schetsen

## Start screen

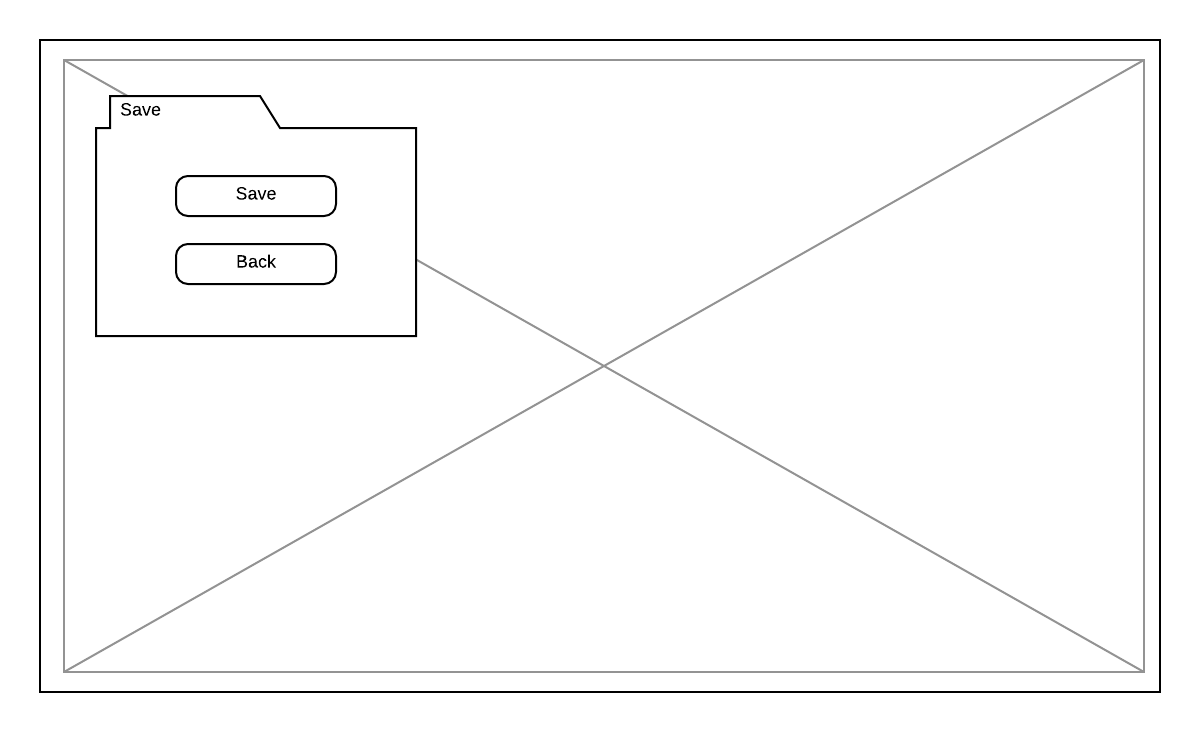


## C:\Users\miche\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Loading screen.pngLoading screen

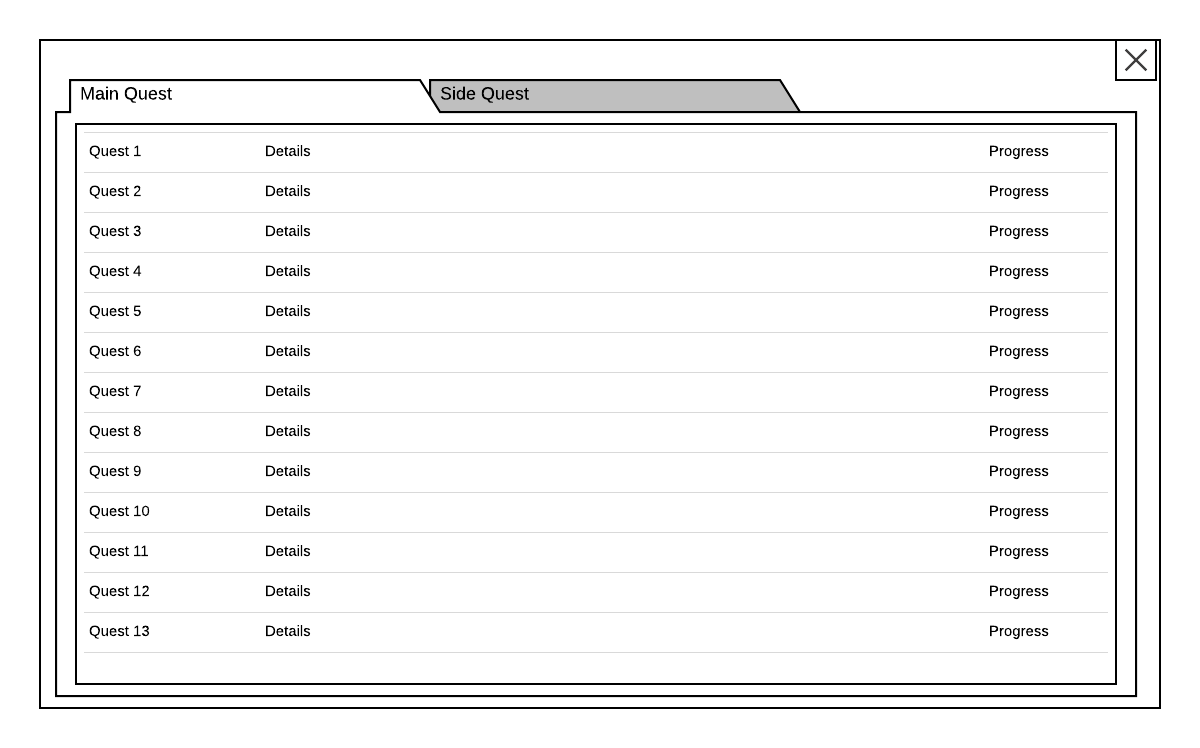
## Main screen



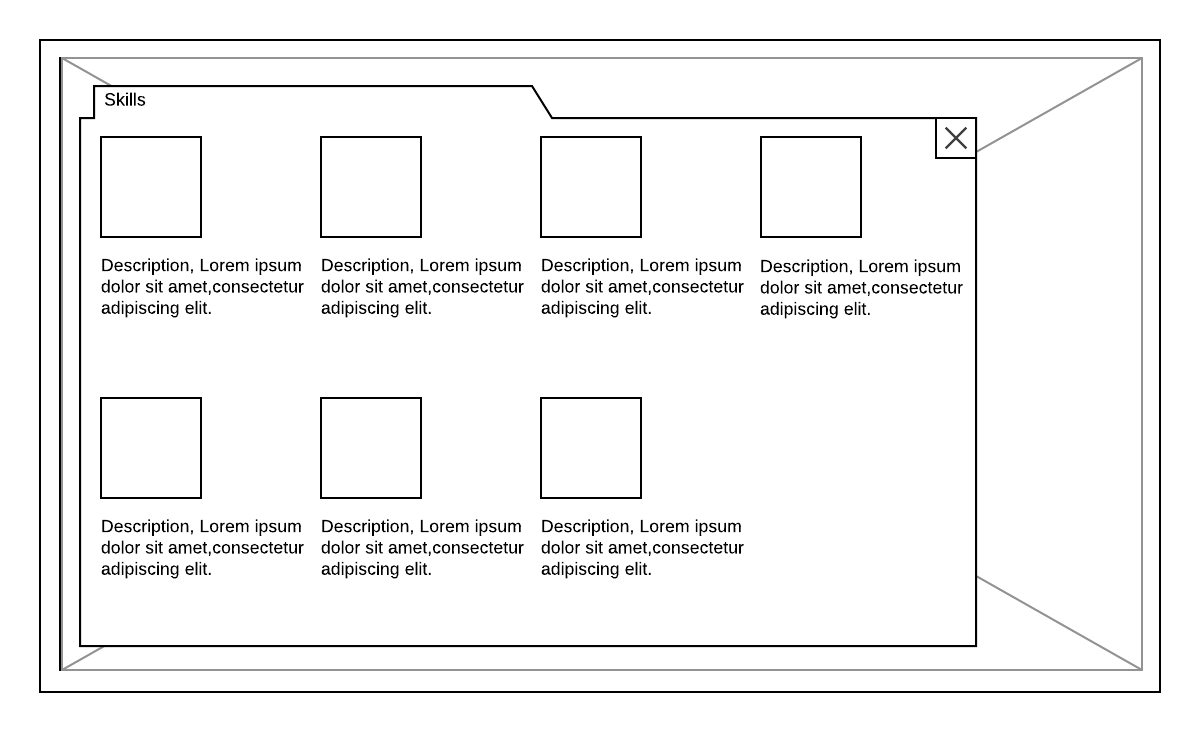
## Save screen



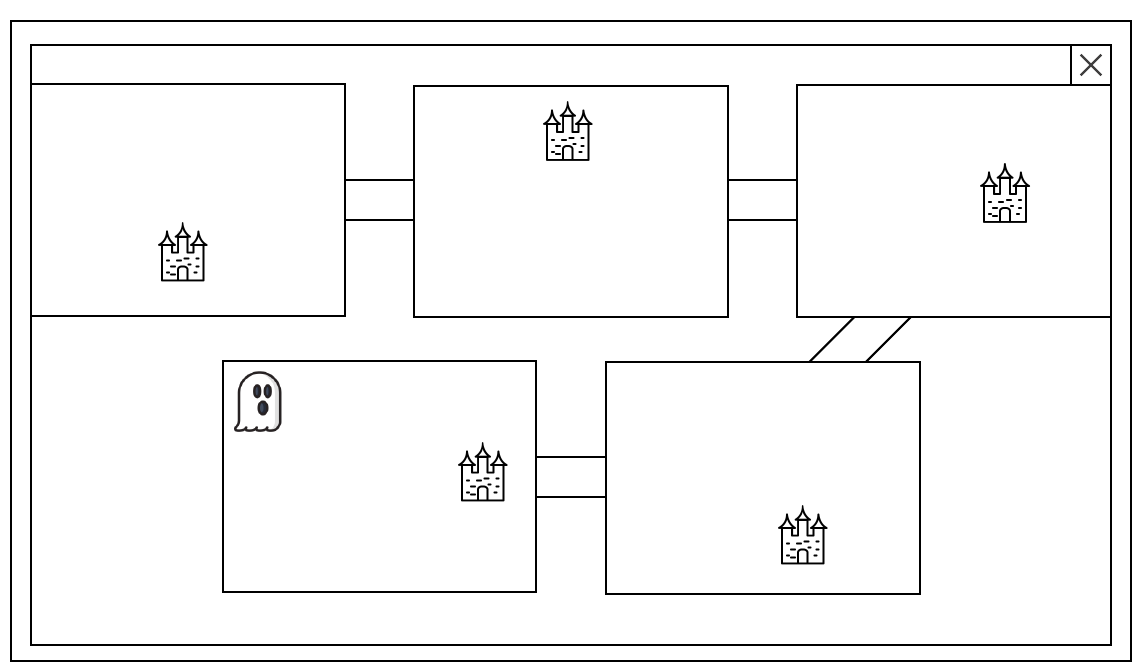
## Journal screen



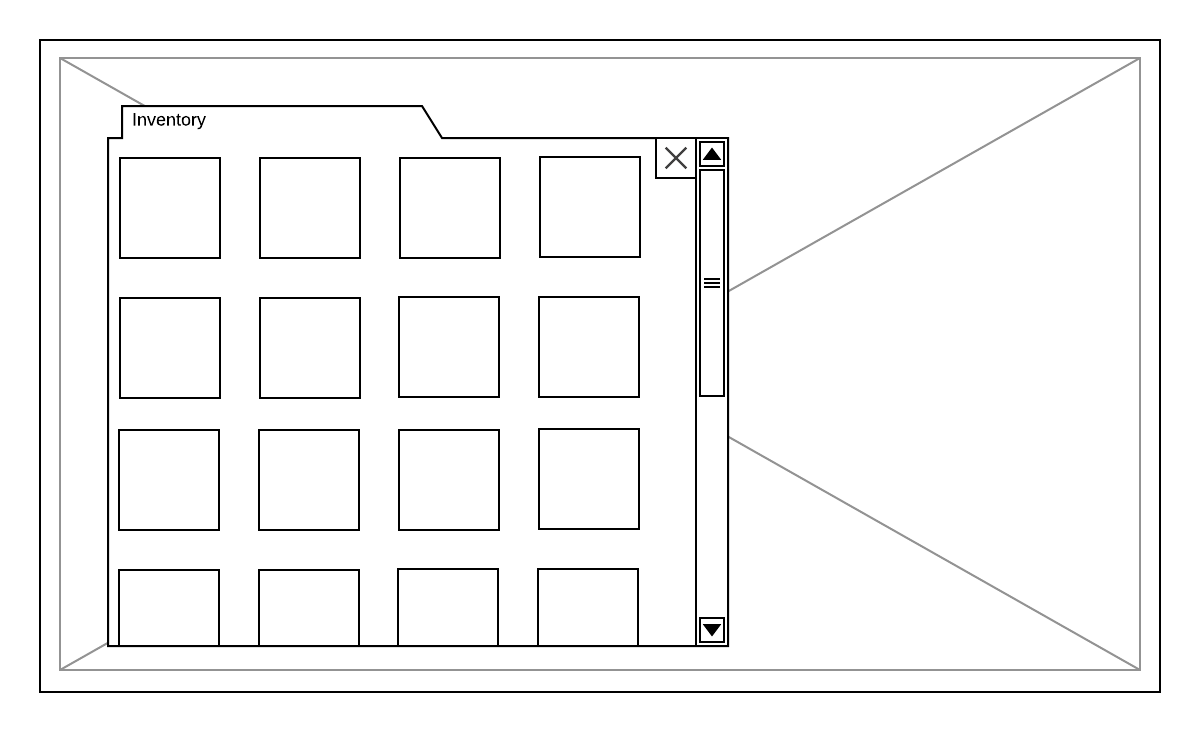
## Skill screen



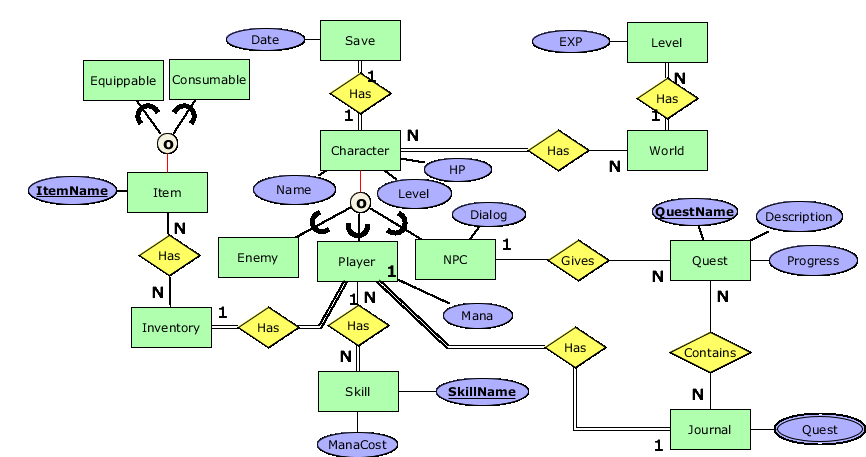
## World map



## Inventory



# EER model

In dit hoofdstuk komt het EER model samen met een bijlage om het model verder uit te leggen.

## EER bijlage

De entiteit **Character** heeft drie andere entiteiten die van de hoofd entiteit afleiden, de **Enemy**, **Player** en de **NPC**. De **Enemy** is een vijand in het spel de **Player** ofwel speler moet verslaan. De speler doet dit door middel van een aanval te gebruiken of een speciale aanval ofwel een **Skill** te gebruiken. Om een skill te gebruiken moet de speler **Mana** gebruiken, per **skill** verschilt hoeveel **mana** ervoor gebruikt moet worden, de sterkere skills zullen dan ook meer **mana** kosten. De **enemy** kan hetzelfde als de speler alleen wordt de **enemy** bestuurd door AI.

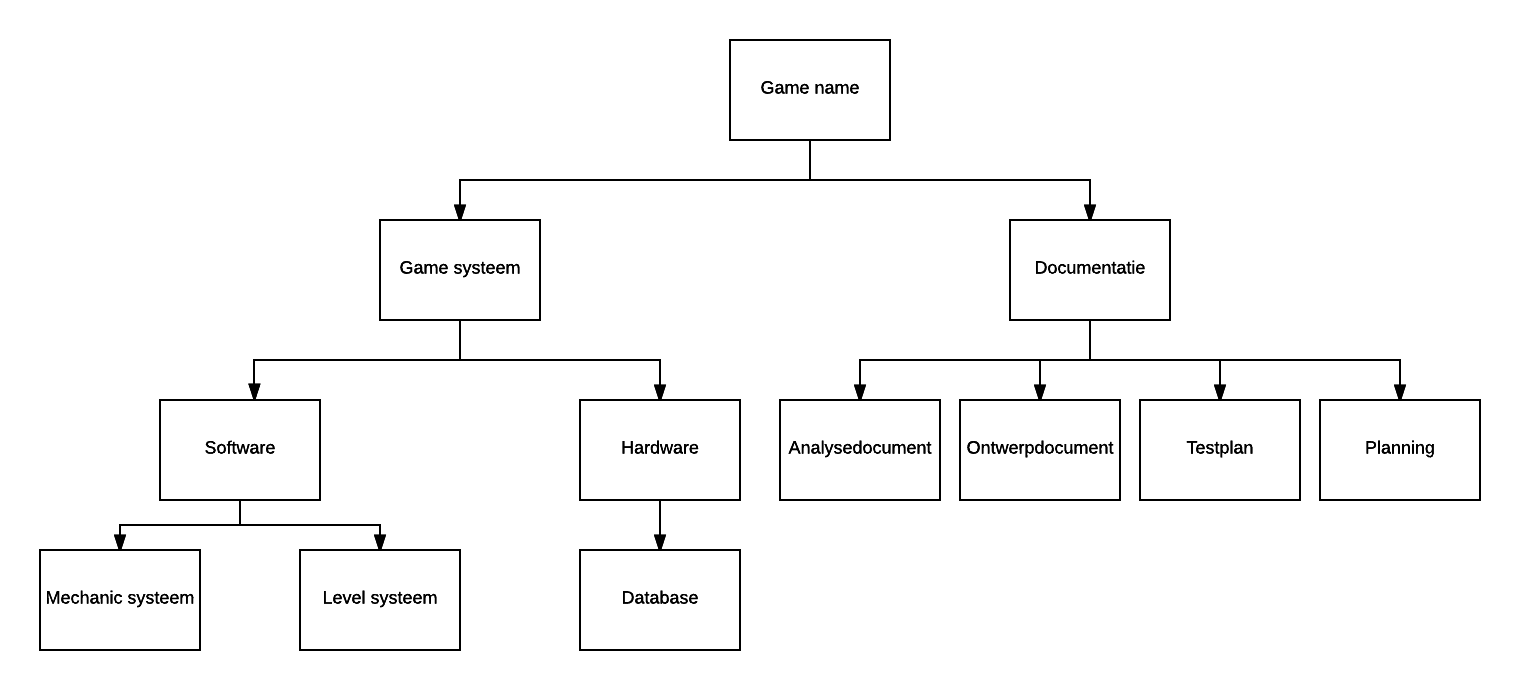
Een **NPC** is eigenlijk een karakter waarmee de speler kan praten, maar er is met dit ontwerp nog wel de mogelijkheid dat er bijvoorbeeld een quest is waar een **NPC** samen met de speler vecht. Een **NPC** is nu voornamelijk iemand waar je **Items** van kan kopen en aan kan verkopen of van wie je een **Quest** krijgt.

De **quest** die een **NPC** geeft wordt gekoppeld aan de speler door deze in de **Journal** te zetten. In de **journal** ziet de speler welke **quest** hij heeft en welke hij al voltooid heeft of hoe ver hij met een bepaalde **quest** is.

Een **Character** is gekoppeld aan de **World** entiteit, dit is de wereld waarop de **characters** zich bevinden, deze **world** is opgesplitst in verschillende **Levels**. De **levels** zijn de verschillende **levels** die beschikbaar zijn voor de speler, er kan dus altijd een nieuwe **level** toegevoegd worden. Er worden een aantal **enemies** en **NPC’s** toegevoegd aan de wereld en deze worden dan weer verdeeld over de verschillende **levels**.

Een speler heeft ook een **Inventory** waar **Items** in opgeslagen worden. De uitrusting van de speler (wapen en de verschillende slots) hoort bij de **Equippable** **items**. En een item zoals een potion wat mana punten hersteld hoort bij de **Consumables**.

# Productdecompositie

Dit hoofdstuk geeft je een beeld in welke delen het project opgedeeld wordt, bijvoorbeeld wat er allemaal opgeleverd moet worden.