Functioneel ontwerp Memory groep 7

# Voorwoord

Dit memory project voeren wij uit voor NHL hogenscholen, waarin Martin Molema onze productowner “klant” is. Door middel van dit project voeren wij het tweede deel van het thema software ontwikkeling uit. Wij zijn Groep 8 van de eerste klas ICT-HBO, we bestaan uit 6 studenten.

# Inhoudsopgave

Inhoud

[Voorwoord 2](#_Toc494968725)

[Inhoudsopgave 3](#_Toc494968726)

[Functionele eisen en wensen 4](#_Toc494968727)

[Analyse gewenste situatie 5](#_Toc494968728)

[Informatieverwerking 5](#_Toc494968729)

[Applicaties 6](#_Toc494968730)

[Infrastructuur 6](#_Toc494968731)

[Consequenties 7](#_Toc494968732)

[Organisatorische consequenties 7](#_Toc494968733)

[Planning 8](#_Toc494968734)

# Functionele eisen en wensen

De functionele eisen en wensen specificeer ik volgens de MoSCow methode.

Must Have:

* Speelveld bestaande uit 16 memory kaarten. 4x4 kaarten dus.
* Achterkant van de kaarten mag zelf gekozen worden.
* Voorkant van de kaarten mag zelf gekozen worden.
* Het scherm moet dynamisch (form onload) opgebouwd worden (niet 16 kaarten vooraf op het windows form plaatsen), maar eerst een hoofdmenu o.i.d.
* Memory moet met 2 spelers gespeeld kunnen worden, beide spelers zijn ‘mens’ en spelen op hetzelfde bord/scherm (geen meerdere geopende instanties van het memory spel.
* Bij het starten van het spel ziet de speler alle kaarten ‘op de kop’ in een 4 bij 4 matrix.
* De spelers kunnen hun namen ingeven in het scherm.
* Er moet een knop zijn om het spel te resetten (herstarten).
* Wanneer het spel is afgelopen, moeten de scores en namen van beide spelers worden opgeslagen en of de speler heeft verloren of gewonnen.
* Vanuit het hoofdmenu moet er genavigeerd kunnen worden naar een pagina waar de highscores op te zien zijn.
* De huidige status van het spel moet worden vastgelegd in het tekstbestand ‘memory.sav’ (het spel wordt als het ware bevroren). De volgende aspecten van het spel moeten worden vastgelegd: status van de 16 kaarten, namen van de spelers, scores van de spelers, welke speler aan de beurt is.
* Terug laden van de opgeslagen bevroren status van het spel. Bij het terug laden van de bevroren status worden de kaarten teruggezet, de namen ingevuld, de scores ingevuld en welke speler aan de beurt was.

Should have:

* Leaderboards.
* Unlockable cardbacks.

Could have:

* Het versleutelen van het memory.sav bestand.
* Achievements.
* In-game currency.
* Soundtrack.
* Difficulty, hardcore, brutal, nightmare mode (8x16 max).
* Card animation.
* Full screen.

Won’t Have:

* Mobile device support.
* Cloud saving.
* Online scoreboard.
* Het toevoegen van sockets voor multiplayer tussen twee computers/laptops

# Requirement analyse

**Must have:**

Dit zijn de must have’s, omdat zonder deze eisen het product niet zal voldoen aan de eisen van de Product Owner. Zonder deze eisen is het geen volledige en functionele game. Door het toevoegen van deze aspecten zal het een werkende game worden.

**Should have:**

De reden dat wij voor deze should have’s hebben gekozen, is vanwege het feit dat Leaderboards en Card Packs twee onderdelen waren waarbij we erg veel interesse er in hadden om dit op een praktische en creatieve wijze te verwezenlijken.

**Could have:**

Een aantal van onze could have’s zijn ook onderdelen die er we er heel graag in willen hebben, maar hebben minder prioriteit omdat we eerst aan de base game, Unlockable card packs en Leaderboards willen werken. Nadat dat gebeurd is, werken we aan onderdelen zoals de Soundtrack, Achievements, In-Game Currency en Full screen. Daarna gaan we kijken hoe we bepaalde elementen samen kunnen laten werken, zoals de Unlockable card packs met de Currency in de Store.

**Wont have:**

Mobile device support zal ons programma niet gaan ondersteunen, we bouwen het namelijk op met Windows Forms. Ook zal onze game geen gebruik maken van online services zoals cloud saving en online scoreboards. We hebben bewust gekozen voor geen connectiviteit met online services, zodat we konden concentreren op de voor ons belangrijkere aspecten van onze game.

# Analyse gewenste situatie

## Informatieverwerking

|  |  |
| --- | --- |
| Name | Spel starten. |
| Actors | Speler. |
| Pre | Speler heeft toegang tot computer. |
| Normal Flow | 1. Speler start het spel op. 2. Speler Selecteert de “Game modus”. 3. Speler selecteert de moeilijkheidsgraat. 4. Speler start het spel. 5. Speler speelt het memory spel. |
| Results | Speler heeft plezier. |
| Alternative Flow | Spel start het spel niet. [1] |

|  |  |
| --- | --- |
| Name | Spel spelen |
| Actors | Speler |
| Pre | Speler heeft het spel gestart. |
| Normal Flow | 1. Speler kiest 2 kaarten. 2. Als kaarten het zelfde zijn, + 1 punt. 3. Speler speelt speelbord leegspelen. 4. Indien de speler klaar is in de tijd, spel gewonnen. |
| Results | Speler krijgt een score. |
| Alternative Flow | Als tijd voorbij is, spel verloren.[4] |

|  |  |
| --- | --- |
| Name | Cardbacks veranderen |
| Actors | Speler |
| Pre | Er zijn Cardbacks vrijgespeeld |
| Normal Flow | 1. Speler gaat naar het hoofdmenu. 2. Speler gaat naar het menu “Shop”. 3. Speler selecteert de gewenste cardback. |
| Results | Speler heeft een alternatieve cardback |
| Alternative Flow | De speler selecteert geen andere cardback.[3] |

|  |  |
| --- | --- |
| Name | Scores bekijken |
| Actors | Speler |
| Pre | Speler heeft een keer eerder een spel gespeeld. |
| Normal Flow | * 1. Speler opent applicatie.   2. Speler navigeert naar scoreboard.   3. Speler kan kijken wat alle (high)scores zijn. |
| Results | Speler is blij, want hij heeft de highscore. |
| Alternative Flow | Speler is droevig, want hij heeft 1 van de laatste scores. [3] |

|  |  |
| --- | --- |
| Name | Opnieuw starten |
| Actors | Speler |
| Pre | Speler is bezig in een spel |
| Normal Flow | * 1. Speler beslist opnieuw te beginnen.   2. Speler klikt op reset button. |
| Results | Spel is opnieuw begonnen |
| Alternative Flow | Speler bedenkt zich. [1] |

|  |  |
| --- | --- |
| Name | Opslaan |
| Actors | Speler |
| Pre | Speler speelt een spel |
| Normal Flow | 1. Speler wil stoppen. 2. Speler drukt op save. 3. Het spel wordt opgeslagen. 4. Speler kan het spel afsluiten. |
| Results | Speler is gestopt met het spel en kan ieder moment verder gaan |
| Alternative Flow | Speler speelt door [4] |

## Applicaties

Het memory spel is een windows forms application.  
Deze applications maken gebruik van .NET Framework, Omdat wij gebruik maken van het nieuwste .net framework moet de computer die het memory spel wil spelen deze ook geïnstalleerd zijn.  
Dit kan alleen op windows 7,8 en 10. Hierom is Windows 7+ noodzakelijk.  
Hierop moet natuurlijk de nieuwste versie van .NET geïnstalleerd zijn(4.6.2).

## Infrastructuur

Indien de applicatie Multieplayer support krijgt voor meerdere computers dan zal er een netwerk nodig zijn. Hierbij zal een standaard Peer to Peer verbinding moeten voldoen.  
verder is er een werkende computer nodig.

# Consequenties

## Organisatorische consequenties

Indien groep 8 het project op tijd en voldoende inlevert en presenteert krijgen wij een voldoende op dit project. Met die voldoende verdienen wij EC’s waardoor we een stap dichter bij ons diploma zijn.  
Hiernaast komen wij gelukkiger door de opleiding heen, omdat we ingeval van stress of depressie memory kunnen spelen.

# Planning

De deadline voor het afronden van het memory spel, is week 44. Het proces hiervan word verdeeld in 5 sprints (inclusief sprint 0) van 1 week per sprint.