

Technisch document

MEMORY MADNESS



**Naam:** Tevin Mulier

**Studentnummer:** 1857585.

**Klas:** 18A-OA-MINLEAR-B13-L13

Inhoud

[1 Gameplay 2](#_Toc124543)

[1.1 Wat is Memory Madness? 2](#_Toc124544)

[2 Spelregels 2](#_Toc124545)

[2.1 Spelregels Memory 2](#_Toc124546)

[2.2 Spelregels Quiz 2](#_Toc124547)

[3 Spelomgeving 3](#_Toc124548)

[4 Engine 5](#_Toc124549)

[4.1 Scenes 5](#_Toc124550)

[4.2 Code 6](#_Toc124551)

[4.2.1 Main menu 6](#_Toc124552)

[4.2.2 Memory 7](#_Toc124553)

[4.2.3 Quiz 10](#_Toc124554)

[4.2.4 Namespace 11](#_Toc124555)

[4.2.5 FallingWords 12](#_Toc124556)

[4.2.6 WordInput 14](#_Toc124557)

[4.2.7 WordSpawner 15](#_Toc124558)

[5 Wireframes 15](#_Toc124559)

[5.1.1 Startscherm 15](#_Toc124560)

[5.1.2 Opties 16](#_Toc124561)

[5.1.3 Minigame 1: Memory 16](#_Toc124562)

[5.1.4 Minigame 2: Quiz 17](#_Toc124563)

[6 Flowcharts 18](#_Toc124564)

[6.1 Flowchart schermen 18](#_Toc124565)

[7 Diagrammen 19](#_Toc124566)

[7.1 Memory Game 19](#_Toc124567)

[7.2 Quiz 20](#_Toc124568)

[7.3 FallingWords 20](#_Toc124569)

[8 Todo 0](#_Toc124570)

# Gameplay

## Wat is Memory Madness?

Het spel Memory Madness bestaat uit een divers aantal educatieve minigames. Hierbij is het de bedoeling dat de gebruiker de mogelijkheid heeft om deze educatieve minigames aan te passen naar zijn eigen behoefte. Hierdoor is het mogelijk dat er gebruik wordt gemaakt van diverse educatieve onderwerpen.

Op dit moment zijn de volgende minigames beschikbaar in Memory Madness:

* Memory
* Quiz

Er zijn plannen om nog een extra minigame toe te voegen in een latere versie. Deze minigame zal net zoals de overige mingames ook een educatieve meerwaarde hebben.

# Spelregels

De speler dient alle beschikbare minigames te behalen om het spel uit te spelen. De spelregels van de minigames zijn als volgt:

## Spelregels Memory

De speler dient alle kaarten om te draaien door deze te matchen. Het spelbord is voorzien van acht kaarten die zichtbaar zijn vanaf de achterzijde. De speler heeft de mogelijkheid om één van deze kaarten om te draaien, hierdoor wordt het gezicht van de kaart zichtbaar. Vervolgens selecteert de speler een tweede kaart, de omgedraaide kaarten worden met elkaar vergeleken. Als de twee omgedraaide kaarten niet hetzelfde zijn, worden deze weer terug omgedraaid naar hun achterzijde. Als de twee kaarten hetzelfde zijn, blijven de gezicht kanten van de kaarten zichtbaar en krijgt de speler een punt. Als de speler alle kaarten correct met elkaar heeft vergeleken, heeft hij een totaal aantal van vier punten en is de minigame uitgespeeld. De speler zal vervolgens doorverwezen worden naar het volgende spel (quiz).

## Spelregels Quiz

Tijdens het spelen van de quiz wordt de speler voorzien van een aantal vragen. Deze vragen kunnen beantwoord worden met goed of fout. Wanneer de speler het antwoord goed heeft, wordt er een punt bij de score opgeteld. Zal de speler de vraag fout hebben, dan zal er een punt van de totale score van de speler afgetrokken worden. Wanneer de speler een totaal van tien punten heeft verzameld, zal hij/zij doorverwezen worden naar het volgende spel.

# Spelomgeving

In dit wordt hoofdstuk wordt er beschreven hoe de spelomgeving er uit ziet vanuit het oogpunt van de speler.

De speler start het spel in het hoofdmenu hierbij heeft de speler de optie om het ‘opties’ menu te selecteren. In het opties menu is het op dit moment mogelijk om de volume van het spel aan te passen. Er zijn plannen om meer instellingen toe te voegen aan dit menu, denk hierbij aan het aanpassen van de resolutie en het selecteren van een achtergrondmuziek. Wanneer de speler zich weer bevindt in het hoofdmenu bestaat er ook nog een optie om het spel af te sluiten door ‘stoppen’ te selecteren. Wanneer de speler wenst om Memory Madness te spelen, dient ‘spelen’ geselecteerd te worden.



Figuur 1: Main menu Figuur 2 Opties

Wanneer de speler het spel begint komt hij/zij eerst in het introductiescherm van de memorygame. In het introductiescherm worden de spelregels uitgelecht. Daarnaast heeft de speler de optie om het spel te starten of terug te gaan naar het hoofdmenu. Wanneer de speler het spel start zal de memory game beginnen. Links boven in het scherm wordt de score bijgehouden. De speler heeft ook nog de optie om het spel te herstarten door de ‘start’ knop te selecteren.



Figuur 3: Introductiescherm Memory

Na het behalen van vier punten wordt de speler gefeliciteerd in het zogenaamde ‘gehaald’ scherm. Hierin heeft de speler de optie om de memory game opnieuw te starten of om het volgende spel te starten.

Wanneer de speler er voor kiest om het volgende spel te starten, wordt hij/zij doorverwezen naar het introductiescherm van de quiz. In dit introductiescherm worden de spelregels uitgelecht. In dit scherm heeft de speler nog de keuze om terug te gaan naar de memory game of om het spel te starten.

Als de speler er voor kiest om de quiz te starten ziet hij boven in het scherm de vraag staan. Vervolgens heeft de speler de keuze om de vraag te beantwoorden met “GOED” of “FOUT”. Wanneer de speler een van deze optie selecteert, wordt hij voorzien van actieve feedback waarin wordt weergegeven of de vraag goed of fout is. Dit wordt gedaan in de vorm van een animatie. Wanneer de speler de “GOED” knop selecteert, zal de “FOUT” knop van het scherm af schuiven. Hierbij zal vervolgens zichtbaar worden of het antwoord goed of fout is. Hetzelfde principe geld ook voor de “GOED” knop. Wanneer de speler een totaal aantal punten van tien heeft is Memory Madness op dit moment uitgespeeld.



Figuur 4: Quiz

Het is voor de speler mogelijk om de vragen in de quiz aan te passen in de Unity editor. Hierbij heeft de speler de mogelijkheid om zoveel mogelijk vragen toe te voegen waardoor de quiz aangepast kan worden naar de wens van de speler.



Figuur 5: Vragen toevoegen Unity editor

Het idee is om zoveel mogelijk educatieve minigames toe te voegen die aan te passen zijn door de gebruiker. Hierdoor ontstaat er uiteindelijk een leuke bundel van minigames die in elke klaslokaal gespeeld kan worden.

# Engine

Memory Madness is gemaakt met Unity. Unity is een game-engine voor het ontwikkelen van computerspellen diverse platformen. In dit hoofdstuk worden onderdelen die gemaakt zijn met Unity beschreven.

## Scenes

Memory Madness bestaat uit een aantal scenes. Elke scene bepaald het scherm die de gebruiker ziet tijdens het spelen van Memory Madness. de volgende scenes worden gebruikt in Memory Madness:

* Scene\_000
  + Main Menu
* Scene\_0001
  + Introductiescherm Memory
* Scene\_001
  + Memory
* Scene\_LevelGehaald
  + Gefeliciteerd scherm memory en quiz
* Scene\_0002
  + Introductiescherm Quiz
* Scene\_002
  + Quiz
* Scene\_0003
* Introductiescherm FallingWords
* Scene\_003
* FallingWords

## Code

In het vorige hoofdstuk is zichtbaar dat elke scene een bepaald onderdeel representeert in Memory Madness. Om de actieve elementen in de scenes daadwerkelijk iets te laten doen, is er gebruikt gemaakt van code. Unity maakt gebruik van C# (uitgesproken als “C sharp”), dit is een programmeertaal die origineel uitgebracht is door Microsoft. Tegenwoordig wordt C# geaccepteerd als standaard programmeertaal.

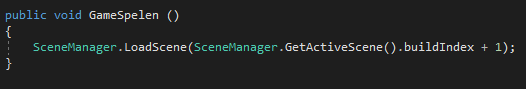
In dit hoofdstuk worden de lastige onderdelen die gecodeerd zijn in Memory Madness besproken.

### Main menu

De main menu scene bevat de volgende script(s):

* MainMenu

In dit script wordt voornamelijk een functie gegeven aan de “SPELEN” knop waardoor de speler doorverwezen wordt naar de volgende scene, dit wordt gedaan in het volgende blok code:



Ook wordt in deze code een functionaliteit gegeven aan de “STOPPEN” knop, hierdoor heeft de gebruiker de optie om het spel te stoppen. Dit wordt gedaan in het volgende blok aan code:



### Memory

De memory scene bevat de volgende scripts:

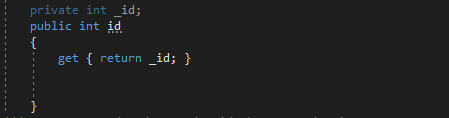
* MainCard
* SceneController
* TerugKnop
* UIButton

In dit hoofdstuk worden de lastige scripts nader toegelicht. De volledige scripts zijn beschikbaar op de repository van Memory Madness.

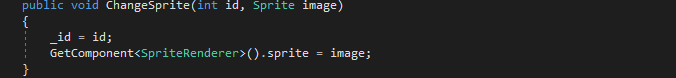
#### MainCard script

Het MainCard script houdt zich voornamelijk bezig met de zichtbaarheid van de kaarten tijdens het spelen van de memory game. De onderstaande blokken aan code waren de meest lastige en opvallende stukken op te programmeren.

In dit blok code worden de kaarten met elkaar vergeleken en er wordt gecheckt of er een match is.



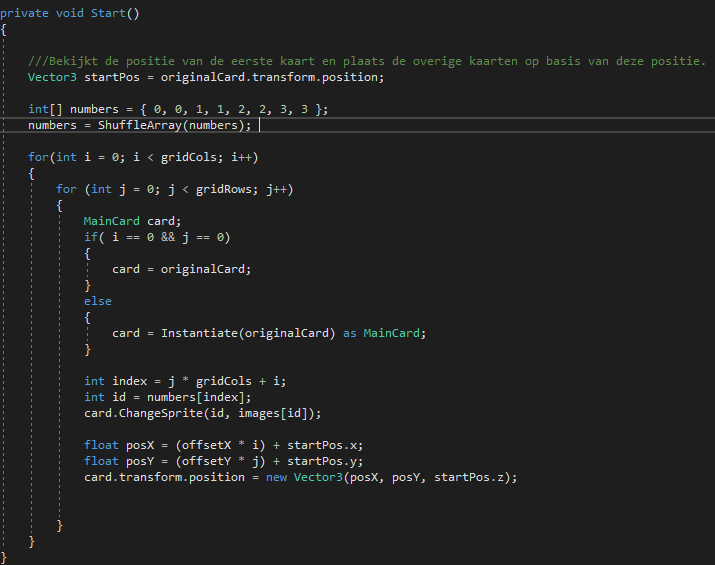
Deze blok aan code zorgt ervoor dat de omgedraaide kaart Random is.

****

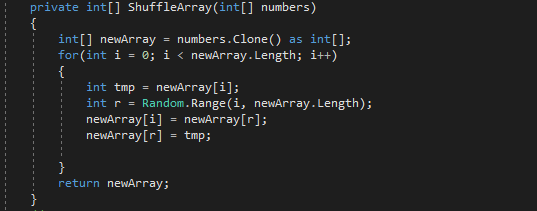
#### Scenecontroller script

Het SceneController script houdt zich voornamelijk bezig met de volgende functionaliteiten:

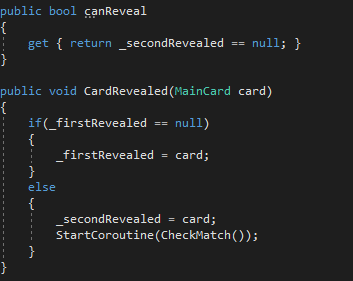
* Het genereren van de kaarten op het scherm in 2 rijen en 4 kolommen.



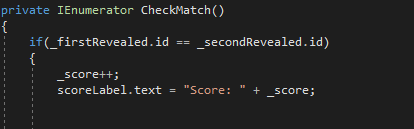
* Het schudden van een array die de lijst van kaarten bevat



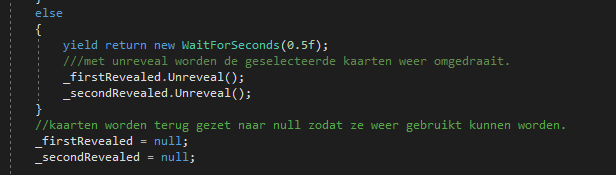
* Het checken wanneer 2 kaarten onthuld.



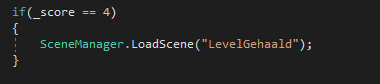
* Het bekijken of de twee onthulde kaarten een match zijn. Er voor zorgen dat de score van de speler met 1 wordt opgeteld als er een match is.



* Er voor zorgen dat de onthulde kaarten weer omgedraaid worden als er geen match is.



* De volgende scene laden wanneer de score van de speler vier is.



### Quiz

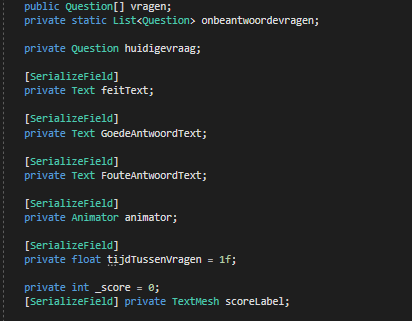
De quiz scene bevat de volgende scripts:

* GameManager
* Question

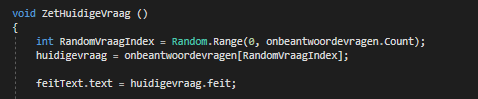
In dit hoofdstuk worden de lastige scripts nader toegelicht. De volledige scripts zijn beschikbaar op de repository van Memory Madness.

#### GameManager script

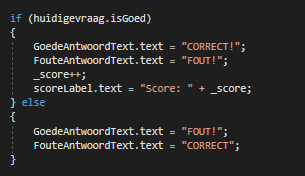
* Maak een array met de vragen die gemaakt kunnen worden in de Unity editor.



* Zorg ervoor dat de vragen zichtbaar zijn voor de speler.



* Laat zien als de vraag goed of fout is aan de speler



### Namespace

Tijdens de ontwikkeling van Memory madness is er gebruik gemaakt van de volgende libaries:

* **System.Collections**

Deze namespace is een verzameling interfaces en classes die diverse objecten bevat zoals lists, queues, bit arrays, hash tables en dictionaries.

* **System.Collections.Generic**

Deze namespace helpt de objecten om de inhoud te converteren naar het juiste data type.

* **System.Linq**

Deze namespace bevat de types die queries ondersteunen die gebruik maken van Language-Integrated Query (LINQ)

* **UnityEngine.UI**

Dit is de namespace zorgt ervoor dat er diverse lettertype en tekst data gebruikt kan worden in C#

* **UnityEngine.SceneManagement**

Deze namespace zorgt ervoor dat er van scene gewisseld kan worden.

### FallingWords

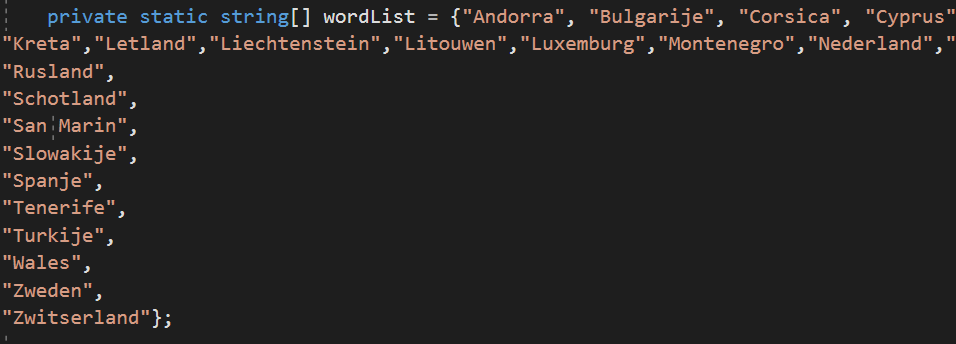
De FallingWords Scene bevat de volgende scripts:

* WordGenerator
* WordDisplay
* WordInput
* WordSpawner
* WordTimer

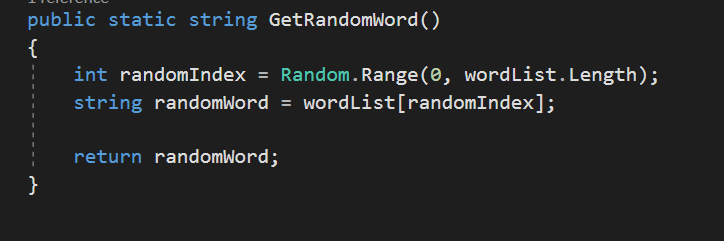
#### WordGenerator

Dit script zorgt ervoor dat de woorden die zich bevinden in een Array worden gegenereerd in Unity.

* Array met woorden die gegenereerd dienen te worden in de game.



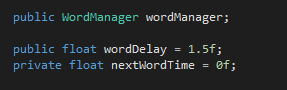
* Statische methode die ervoor zorgt dat er een willekeurige woord uit de array wordt gegenereerd.



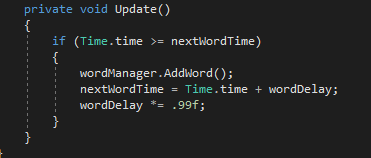
#### WordTimer

Dit script bepaald de snelheid waarin de woorden gegenereerd en getoond worden op het scherm. In het script genaamd WordTimer is er een algoritme toegevoegd waarin woorden steeds sneller gegenereerd en gespawned worden. Dit verhoogd de moeilijkheidsgraad van het spel na verloop van tijd. Omdat de spawntijd is gecodeerd als een publieke functie, kan deze aangepast worden door de speler in de Unity Editor.

* Publieke methode waarin bepaald kan worden hoe lang er gewacht moet worden met het genereren van een woord.



* Update method die een algortime bevat waardoor woorden over verloop van tijd sneller gegenereerd zullen worden.

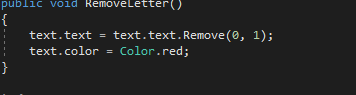


#### WordDisplay

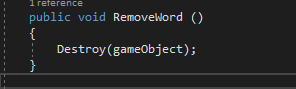
Dit script zorgt ervoor dat de speler voorzien word van feedback als de eerste letter van het woord getypt wordt.

Het script bevat een functie die herkend als de eerste letter van een woord die zich op het scherm bevindt wordt getypt. Het woord wordt vervolgens gemarkeerd doordat deze geselecteerd wordt in een rode kleur. Nadat de speler het juiste letter typt, wordt deze verwijderd van het woord. Nadat alle letters van het woord zijn getypt wordt deze verwijderd van het spel.

* Markeer tekst rood en verwijder letter.

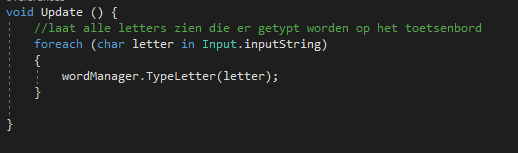


* Verwijder woord van spel.



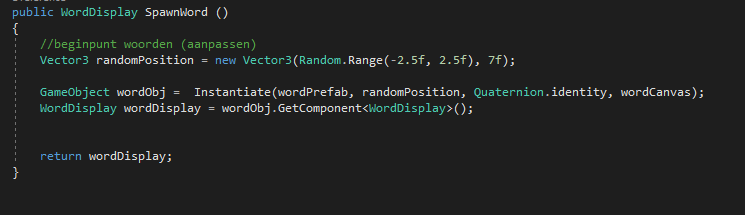
### WordInput

In het script WordInput bevindt zich een inputstring die herkend welke letters er worden getypt op het toetsenbord door de gebruiker. Deze informatie wordt uiteindelijk gebruikt door het WordDisplay script om letters van woorden te highlighten en verwijderen.



### WordSpawner

het doel van dit script is om de woorden vanaf verschillende positities naar beneneden te laten vallen over het scherm. Hierin wordt er met een RandomPosition functie een willekeurige positie gekozen op het canvas waarop de woorden zich bevinden.



# Wireframes

Dit hoofdstuk bevat de schetsen van de schermen die zichtbaar zijn tijdens Memory Madness.

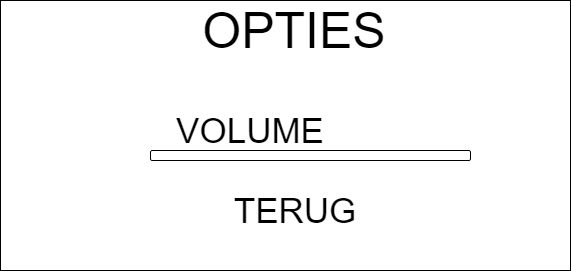
### Startscherm

De onderstaande illustratie is een schets van het main menu.



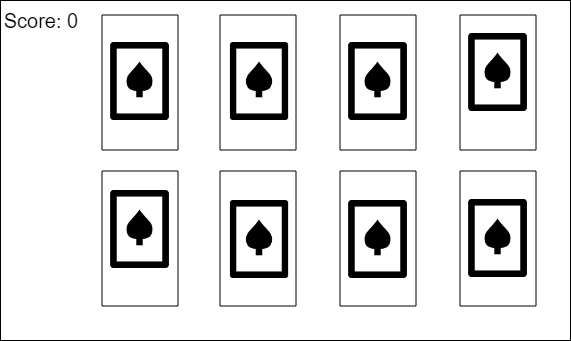
### Opties

De onderstaande illustratie is een schets van het opties menu dat beschikbaar is vanuit het main menu.



### Minigame 1: Memory

De onderstaande illustratie is een schets van de memory game.



### Minigame 2: Quiz

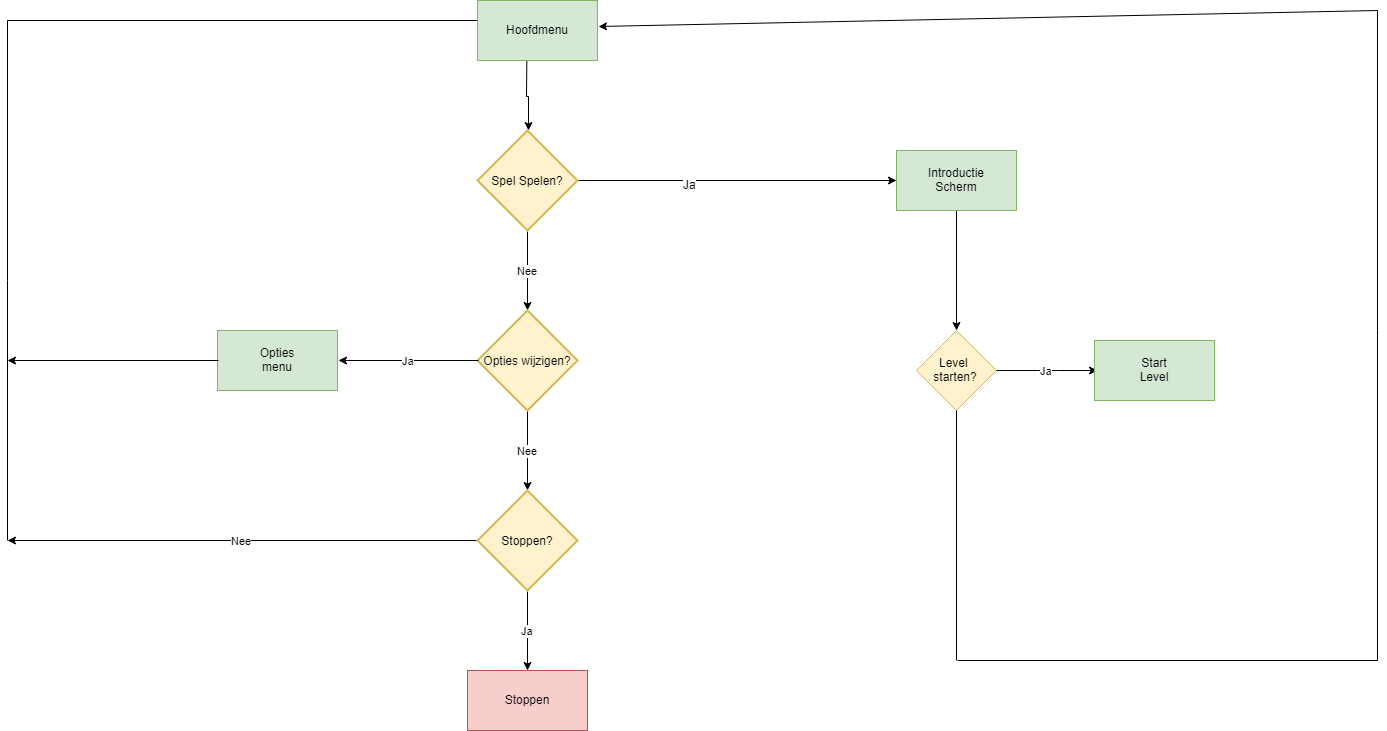
De onderstaande illustratie is een schets van de quiz.



# Flowcharts

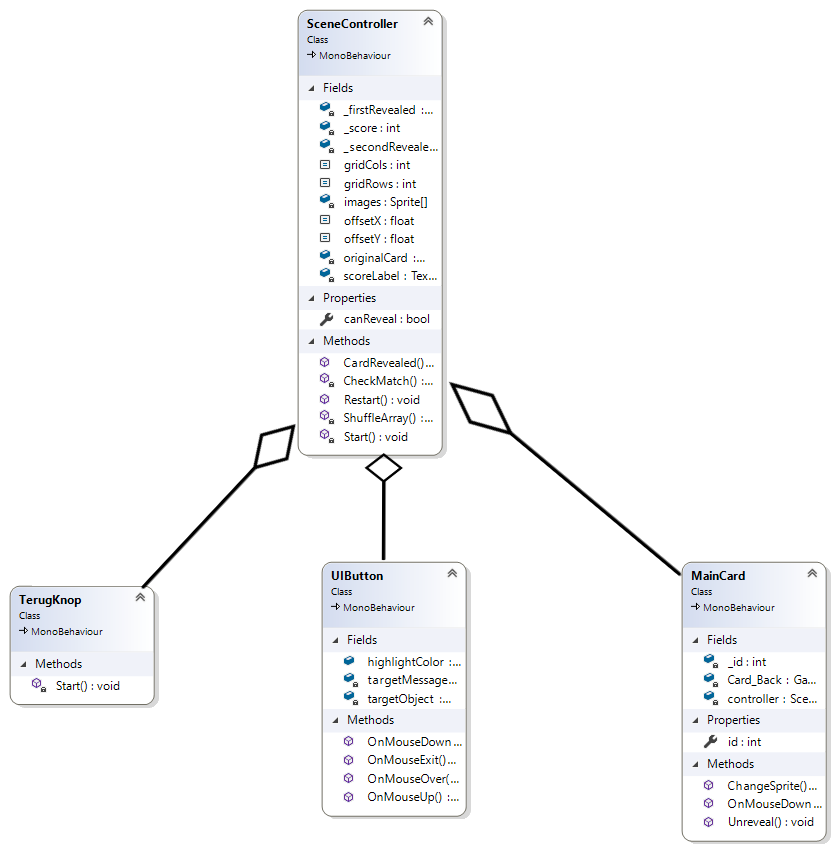
In dit hoofdstuk worden er flowcharts getoond die relevant zijn voor Memory Madness.

## Flowchart schermen

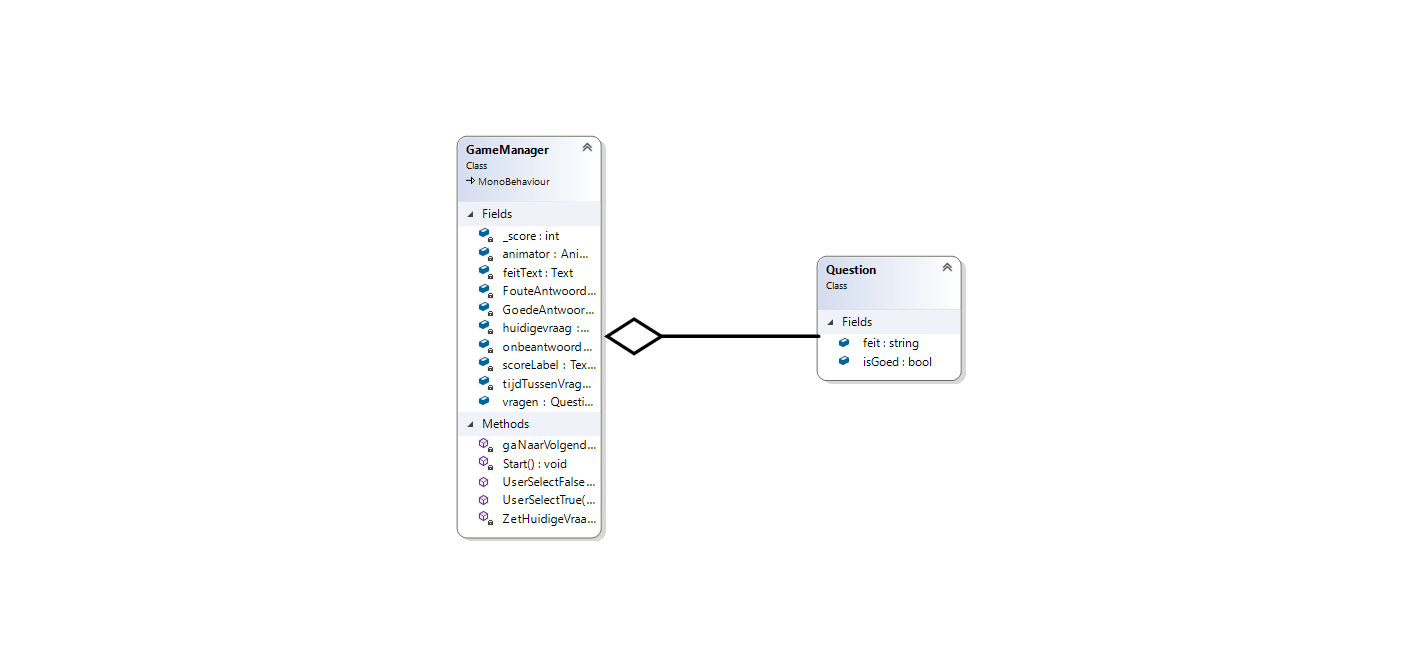


# Diagrammen

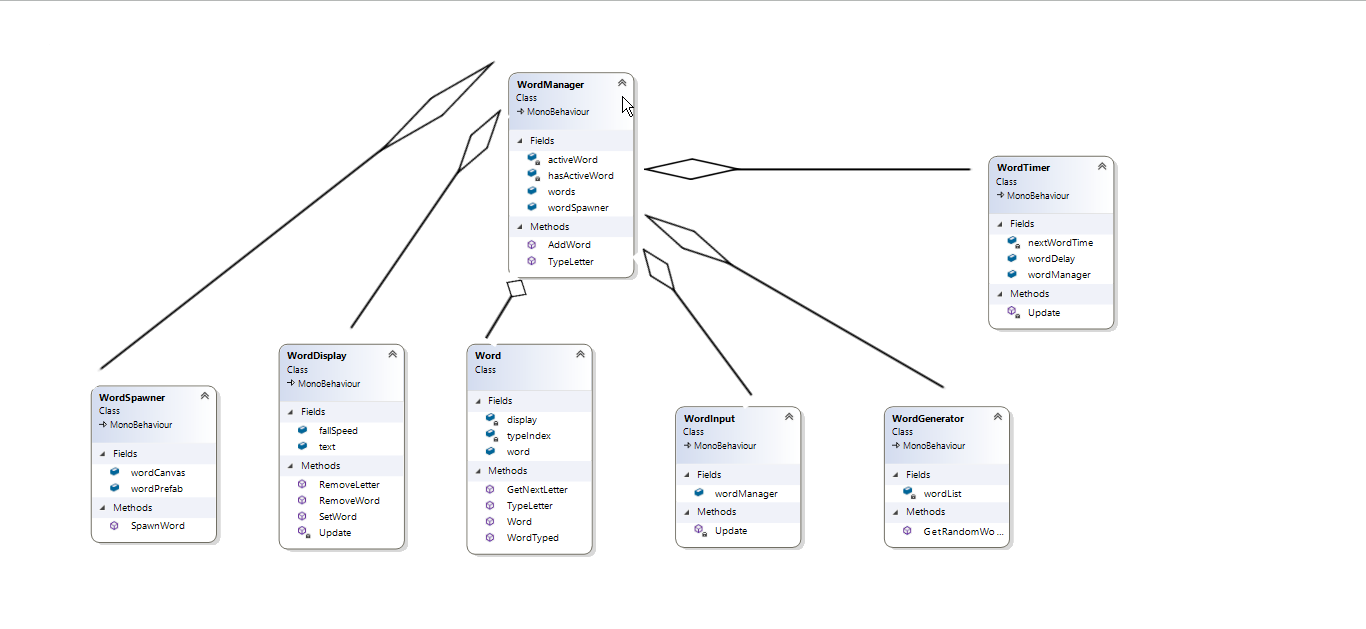
## Memory Game

****

## Quiz



## FallingWords



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Task Name | Story | Sprint Ready | Priority | Status | Story Points | Assigned to Sprint |
| Backlog | Yes | Yes | Low | Not Started | 0 | Yes |
| Pauze scherm toevoegen | Yes | No | Low | Not Started |  | No |
| "victory" geluidseffect toevoegen | Yes | No | Low | Not Started |  | No |
| High-score scherm toevoegen | Yes | No | Low | Not Started |  | No |
| Speler data opslaan | Yes | No | Low | Not Started |  | No |
| Memory kaarten aanpasbaar maken | Yes | No | High | Not Started |  | No |

# Todo

De volgende taken staan in backlog of worden eventueel op een later moment in de ontwikkeling van Memory Madness toegevoegd: