11/8/2018

Albert-Jan Scherrenburg

Student Nr: 1684118

Datum: 08-11-2018

Versie: 1.0.1

Pick a Pair

Technisch Verslag

# Versiebeheer

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Datum | Naam | Aanpassingen |
| 1.0.0 | 06-11-2018 | Albert | Document aangemaakt en technisch ontwerp gemaakt |
| 1.0.1 | 07-11-2018 | Albert | Complexe functies beschreven en document compleet gemaakt. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Inleiding

Dit is het Technisch Verslag document van het memory spel Pick a Pair. Het spel is ontworpen en gemaakt door mijzelf, Albert-Jan Scherrenburg, in opdracht van het vak Educatieve Games Programmeren.

Dit document begint met een technisch ontwerp, waar het spel en al de functies van het spel schematisch worden weergegeven.

Vervolgens worden de complexere functies van dit spel een voor een behandeld en beschreven, zodat ze door iedereen begrepen en overgenomen kunnen worden, mocht dit nodig zijn.

Aan het einde van dit document wordt nog een korte conclusie gegeven over de technische aspecten van dit spel, en enkele tips voor developers die dit spel willen overnemen om het vervolgens zelf verder te ontwikkelen.

# Inhoudsopgave

[Versiebeheer 1](#_Toc529631792)

[Inleiding 2](#_Toc529631793)

[Inhoudsopgave 3](#_Toc529631794)

[Technisch Ontwerp 4](#_Toc529631795)

[Functie Beschrijving 5](#_Toc529631796)

[GameManager Script 5](#_Toc529631797)

[Initialize Cards 5](#_Toc529631798)

[Check Cards 6](#_Toc529631799)

[Card Comparison 6](#_Toc529631800)

[Card Script 7](#_Toc529631801)

[Setup Graphics 7](#_Toc529631802)

[Flip Card 7](#_Toc529631803)

[Conclusie 8](#_Toc529631804)

# Technisch Ontwerp

Hieronder staat het technisch ontwerp voor het spel Pick a Pair – Memory Game. Ik heb er voor gekozen om een class diagram te maken voor het technische ontwerp. In dit diagram staat elke class uit de scripts voor dit spel beschreven. Verder geeft dit diagram de connecties tussen de classes ook weer.



# Functie Beschrijving

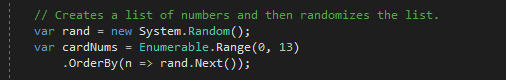
In dit hoofdstuk worden de complexe onderdelen van de scripts die dit spel laten draaien beschreven. Het game manager script en het card script zijn de twee grootste scripts die wat complex zijn. Deze twee scripts zal ik dan hier ook behandelen. De andere scripts spreken voor zich en zijn met behulp van het commentaar in de scripts te begrijpen. Om dit zo overzichtelijk mogelijk te doen heb ik er voor gekozen om per script een beschrijving te geven, en per script de meest complexe onderdelen te behandelen.

## GameManager Script

Het Game manager script is het belangrijkste script van het spel Pick a Pair – Memory Game. Dit script start wanneer de speler vanuit het menu scherm op de start knop klikt, en zorgt ervoor dat de kaarten op het scherm worden ingeladen. Daarnaast wordt dit script ook gebruikt om kaarten om te draaien en vervolgens te kijken of de omgedraaide kaarten een match vormen.

### Initialize Cards

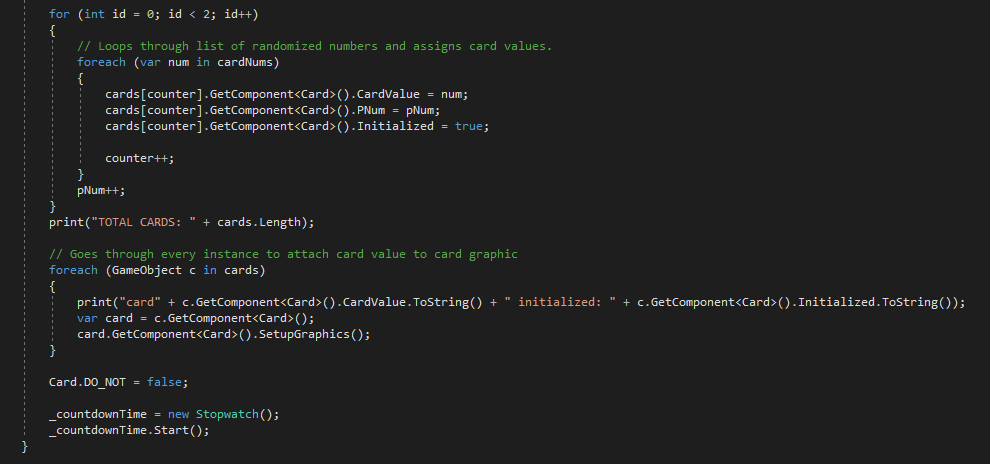
In deze Void worden de kaarten ingeladen. Dit is het eerste wat het script doet als het spel wordt geladen. Als eerste worden er een paar integers en variabelen opgezet. Deze integers worden gebruikt om alle kaarten op de juiste manier in te laden. De rand variabele wordt gebruikt om met behulp van de cardNums variabele een willekeurige nummer volgorde te creëren.



Nadat de willekeurige volgorde is gegenereerd worden de getallen middels een loop aan de kaart waarden vastgemaakt.

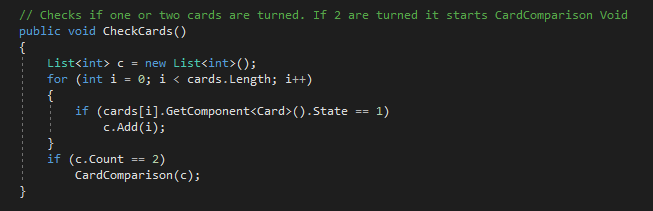
Nadat deze loop voltooid is wordt met behulp van een “foreach” loop (/controlled statement) de kaart waarden aan de objecten vastgemaakt. Deze kaart waarden worden later weer gebruikt om de juiste kop van een bepaalde kaart te laten zien als de speler op een kaart klikt.

Verder wordt de stopwatch hierna gedefinieerd en gestart, die gebruikt wordt voor de match timer.



### Check Cards

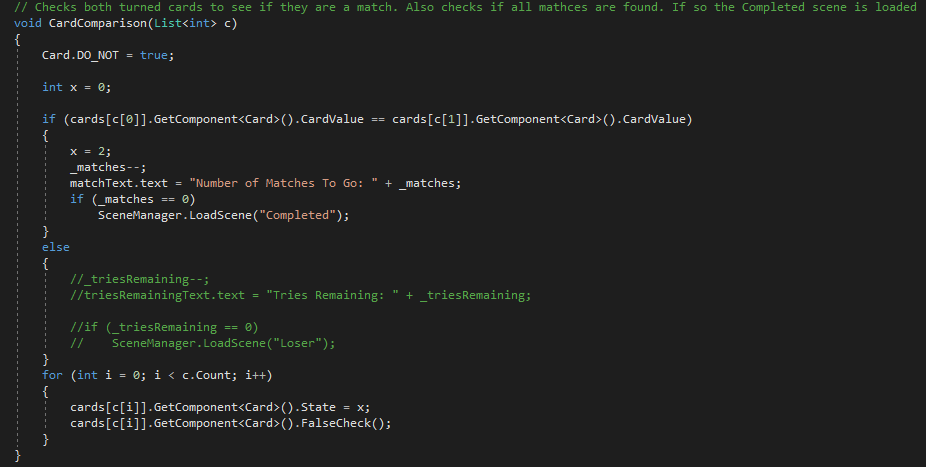
In de check cards void wordt er gekeken hoeveel kaarten er momenteel omgedraaid staan op het scherm. Als c.Count op 2 staat, wat betekend dat er 2 kaarten door de speler omgedraaid zijn, wordt de void CardComparison gestart en de values van de twee kaarten meegegeven.



### Card Comparison

In de card comparison void wordt er door het script gekeken of de twee momenteel omgedraaide kaarten een set vormen. Met behulp van een if statement wordt er gekeken of de twee waarden van de omgedraaide kaarten hetzelfde zijn, waarna als die het geval is de kaarten zullen blijven staan en de match tekst zal aangeven dat er een setje minder gevonden hoeft te worden.

Als het aantal setjes dat gevonden moet worden de 0 bereikt heeft, zal het spel het felicitatie scherm weergeven. Dit wordt gedaan doormiddel van het “SceneManager.Loadscene” statement.



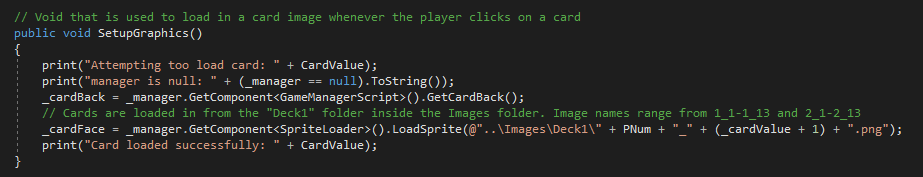
## Card Script

Het Card script is het script dat gebruikt wordt door het game manager script om de plaatjes van de kaarten in te laden, en kaarten “om te draaien”.

### Setup Graphics

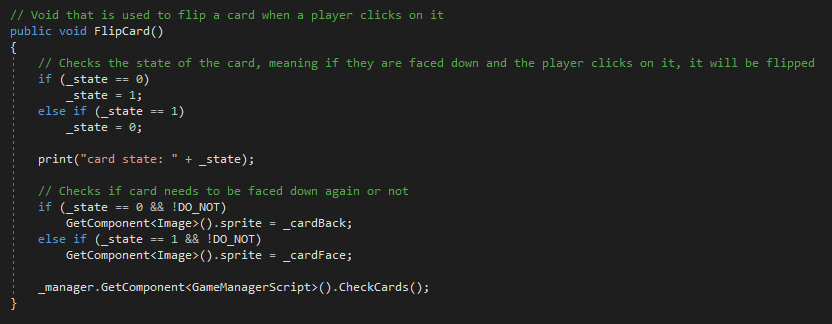
De setup graphics void wordt zoals de naam al verklapt gebruikt om de plaatjes in het spel te laden.

Als de speler op een kaart klikt, moet door het game manager script de kop van de kaart opgehaald worden. Dit gebeurt door deze void. De variabel “\_CardFace” zal worden aangehaald, die vervolgens met behulp van de waarde die aan de door de speler geklikte kaart vast gemaakt zit het juiste plaatje uit de afbeeldingen map op halen.



### Flip Card

De flip card void wordt gebruikt om kaarten om te draaien. Dit is mogelijk door kaart objecten een state te geven. Als een kaart met de achterkant naar boven ligt, zal hij een state=0 hebben. Als de speler er dan op klikt, zal de state veranderen in state=1.



# Conclusie

Na het lezen van dit technisch verslag bent U volledig voorbereid om dit project op te pakken en hier eventueel aan verder te werken. Om aan dit project te werken is het aanbevolen om de laatste versie van Unity te gebruiken, en de scripts te bewerken in een IDE naar keuze. Zelf heb ik voor dit project Microsoft Visual Studio 2017 gebruikt.