Documentatie Project
Memory Game

Jan Hoekstra
26 oktober 2018

0 Inhoudsopgave

1 Voorwoord

1.1 Opdracht

korte omschrijving opdracht

1.2 Groep

- Hoekstra, Jan Sietze

- Meijer, Joost
- Hooft, Emiel
- Masselink, Tjibbe
- Raven, Niek
- B.A. Hensbergen Productowner

2 Managementsamenvatting

probleemschets, hoofdvraag, belang van de hoofdvraag methode en werkwijze resultaten conclusies en aanbevelingen

3 Inleiding

4 Projectomschrijving

De bedoeling van dit project is het gaan opzetten van een Memory Game met de eisen en wensen van de klant. Het project wordt in een groep van 5 man groot uitgevoerd.

4.1 Huidige situatie

De projectgroep heeft de opdracht gekregen een memory spel te maken. Ze hebben toegang tot een tutor en een product owner voor hulp.

4.2 Probleemstelling

Er is behoefte aan een digitaal memory spel, ontwikkeld door de projectgroep. Zie [Eisen] voor alle verplichte eigenschappen van het memory spel.

4.3 Werkwijze

Voor het project is 1 periode vastgesteld. Week 110 van het eerste schooljaar. De eerste 5 weken kunnen worden gebruikt voor planning etcetera en de 5 opvolgende weken voor het programmeren.

We gaan werken met een scrum methode, elke week een gedeelte. De eisen worden elke week opgesplitst. Elke week is er een demo voor de productowner.

4.4 Definition of Done

Een item is 'done' wanneer:

- het door een teamgenoot gereviewed is
- het getest is
- het gedane werk omschreven staat in een user story
- het gedocumenteerd is

5 Functioneel Ontwerp

5.1 MoSCoW analyse

Must Have

- 4 x 4 matrix van kaarten
- Gameveld moet dynamisch geladen worden
- 2 Spelers moeten tegen elkaar kunnen spelen in 1 game
- De spelers kunnen hun namen ingeven in het scherm;
- Resetknop
- Persistent scoreboard met namen met win/verlies status
- Vanuit het hoofdmenu moet er genavigeerd kunnen worden naar een pagina waar de highscores op te zien zijn
- Status van spel bevriezen en later hervatten d.m.v. een tekstbestand

Should Have

- Kaartenthema

Could Have

- Spelers kunnen een avatar kiezen.

Won't have

- Een optie in-game om kaartenthema te selecteren
- Artificial Intelligence

6 Technisch Ontwerp

6.1 Achtergrond

6.1.1 Te gebruiken technieken & talen

Om het memory project te realiseren worden de volgende technologieën gebruikt;

- Het project wordt geschreven in C# en XAML
- Er wordt gebruik gemaakt van Microsoft Visual Studio als ontwikkelomgeving
- De widget toolkit die gebruikt wordt is Microsoft's eigen Windows Presentation Foundation
- Voor versiebeheer wordt gemaakt van Git in combinatie met GitHub

6.2 Architectuur

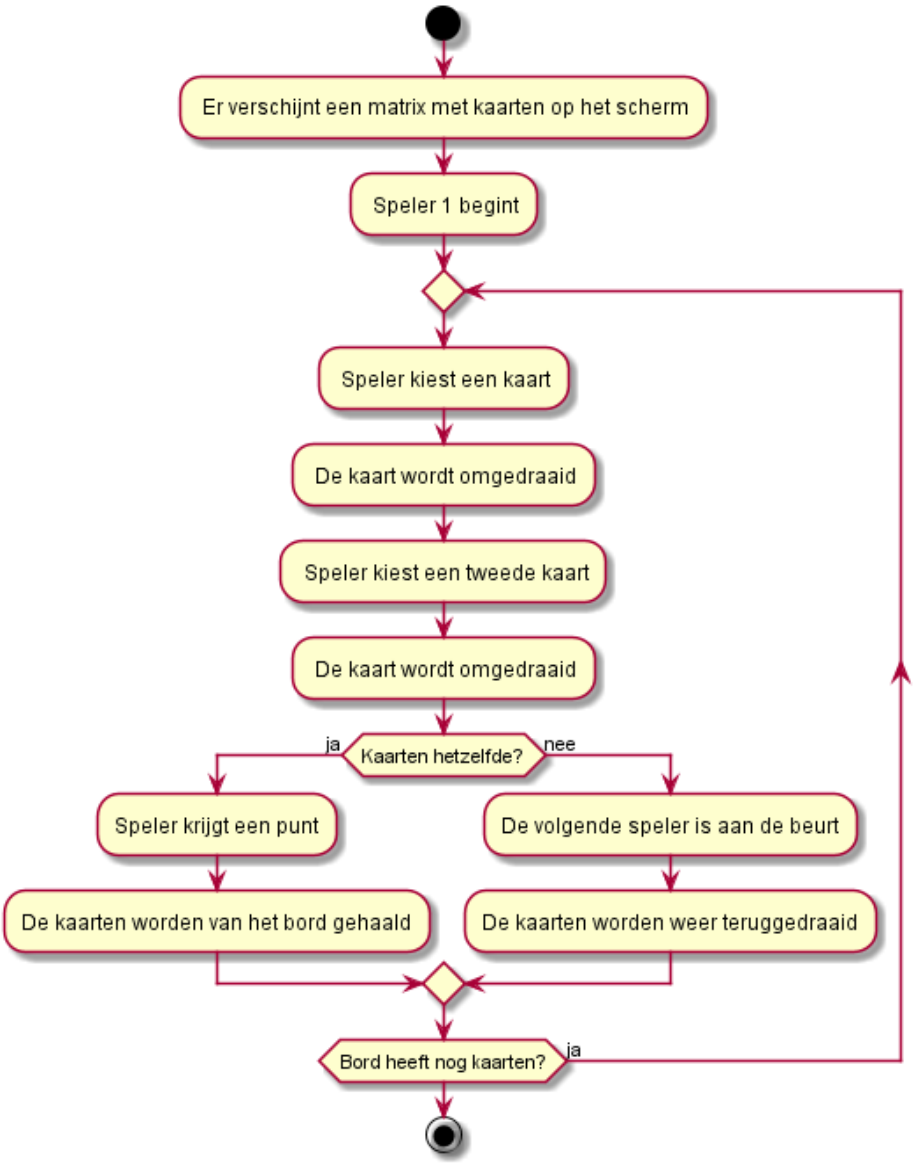
6.2.1 Logica

Waarom dit ontwerp

Omdat de programmeurs nog niet veel ervaring hebben met ontwikkelen, laat staan game state machines etc, wordt het ontwerp opzettelijk simpel gehouden.

Globaal

De gamelogica wordt uitgevoerd in een loop. In het diagram hieronder is het belangrijkste gedeelte van de high-level logica weergegeven.



Gedetailleerd

1. Matrix

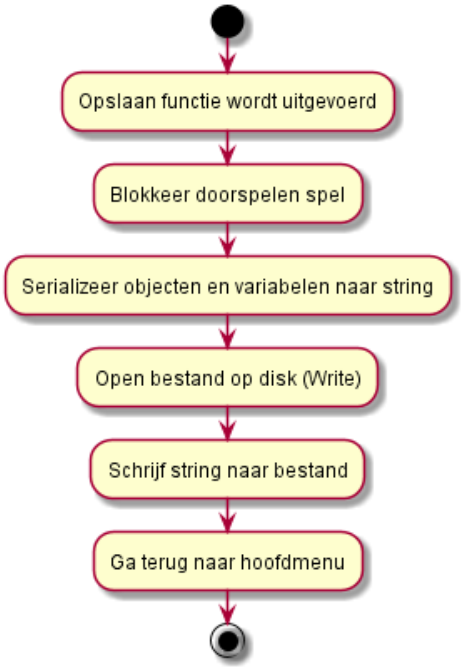
   De matrix wordt als een object aangemaakt. Dit object bevat alle benodigde informatie voor het functioneren hiervan.

2. Kaarten De kaarten worden als objecten aangemaakt. Deze objecten bevatten informatie zoals:

   - Kaart ID (verwijst naar plaatje d.m.v. objectfunctie)

6.2.2 Opslaan/laden van savestates

Omdat de matrix(incl. de kaarten) een object is, kan dit makkelijk worden opgeslagen(geserialiseerd). Variabelen kunnen ook makkelijk geserialiseerd worden.



6.2.3 User Interface

De User Interface interactie wordt geabstraheerd doormiddel van een omkoepelende class. Deze class is de enige route om wijzigingen in de UI uit te voeren. Het voert geen gamelogica uit.

6.3 Volgorde van onderdelen

De volgorde waarin het programma wordt geschreven is:

1.

6.4 Gemaakte keuzes architectuur

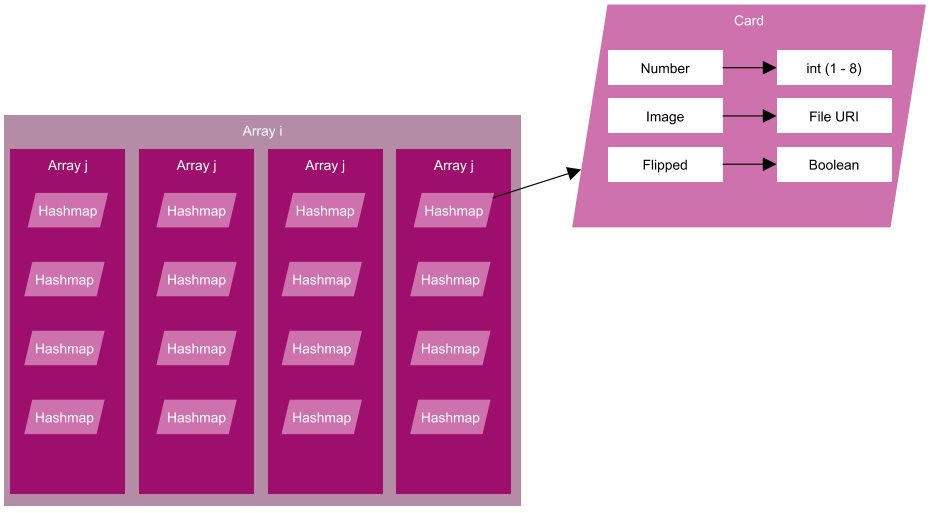
In deze subsectie wordt omschreven welke keuzes tijdens het project zijn gemaakt, deze zijn gedocumenteerd ter referentie.

6.4.1 Logica

Generatorfuncties: Om globale 'state' te voorkomen, zijn zo veel mogelijk functies geschreven met het doel een nieuw object terug te geven in plaats van een object van buiten te bewerken.

6.4.2 Data

Kaarten: De kaarten moeten worden opgeslagen in het programma, bij voorkeur onafhankelijk van de gebruikersinterface. Er is gekozen voor een 2D array met voor elke kaart een hashtable.



Figuur 1: Datastructuur