Java groepsopdracht

Mini games

stuart drysdale, jorrick houtman, Iris jansen, jasper kuiper en monique maquiné

2018

# Inleiding

Dit verslag is geschreven voor de groepsopdracht van het vak Java in het 4e semester van de opleidingen Applicatie ontwikkeling en Gamedeveloper. In dit document worden het plan van aanpak, het functioneel ontwerp en het technisch ontwerp.

In het plan van aanpak worden onder andere de introductie van de mini games, de taak verdeling en de planning besproken.

In het functioneel ontwerp worden onder andere het scenario, een use-case, de functionaliteiten en een moscow analyse uitgelegd.

In het technisch ontwerp worden onder andere een klassendiagram en de technieken uitgelegd.

Inhoudsopgave

[Inleiding 1](#_Toc515537543)

[Plan van aanpak 3](#_Toc515537544)

[Introductie Mini-Games 3](#_Toc515537545)

[Taakverdeling 3](#_Toc515537546)

[Grafische planning 4](#_Toc515537547)

[Functioneel ontwerp 5](#_Toc515537548)

[Scenario 5](#_Toc515537549)

[Use- case 5](#_Toc515537550)

[Functionaliteiten 5](#_Toc515537551)

[Moscow analyze 6](#_Toc515537552)

[Schetsen 7](#_Toc515537553)

[Technisch ontwerp 9](#_Toc515537554)

[Klassendiagram 9](#_Toc515537555)

[Benodigde software 9](#_Toc515537556)

[Technieken 9](#_Toc515537557)

# Plan van aanpak

## Introductie Mini-Games

Wij hebben besloten om 4 mini-games te gaan maken in Java. Deze worden gescheiden van elkaar door middel van een hoofdmenu. Ook komt er tijdens het spelen van een game een menu bovenin waarmee je kunt wisselen tussen de games.

**Getallen onthouden:** In dit spel laat het scherm een getal zien. De speler moet het getal gedurende de 10 seconden dat het in beeld is onthouden. Daarna verdwijnt het getal en moet de speler het invoeren. Wanneer de speler het goed heeft komt er een nieuw langer getal. Wanneer de speler het fout heeft stopt het spel.

**Woorden onthouden:** Het scherm laat een woord zien. De speler moet kiezen of dit woord als eens eerder voorbij is gekomen met de knoppen “gezien” en “nieuw”. Als de speler het goed heeft komt er een nieuw random woord. Als de speler het fout heeft stopt het spel.

**Tijd raden:** Het scherm laat een random getal zien dat de tijd aangeeft. Deze tijd loopt af naar 0, maar stopt op een random moment. De speler moet dan doortellen in zijn/haar hoofd en op tijd op de knop drukken met als doel zo dicht mogelijk bij de 0 te komen.

**Reflex test:** Het scherm laat een kleur en een knop zien. Na een aantal seconden verandert het scherm van kleur. De speler moet dan zo snel mogelijk op de knop drukken. De tijd die het duurde tussen het kleur wisselen en het drukken op de knop wordt weergegeven als score.

## Taakverdeling

De functie van Scrum-master gaat Jorrick vervullen. Hij zorgt ervoor dat iedereen aan het werk kan.

Het opzetten van de git omgeving is de taak van Jasper. Hij verzorgt de master en development branches.

Het schrijven van de documentatie is de taak van Jorrick en Monique. Zij schrijven het plan van aanpak, het functioneel ontwerp en het technisch ontwerp.

Het maken van de ontwerpen is de taak van Stuart. Hij heeft de games weer in schetsen.

Het aanmaken van de klassen is de taak van Iris.

Wanneer het programmeren begint krijgt iedereen een klasse. Hij/zij is verantwoordelijk voor het functioneren van die klasse.

## Grafische planning

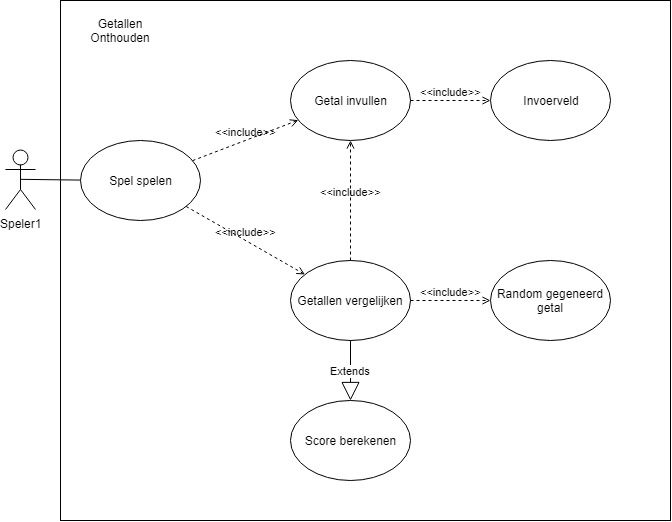
|  |  |  |  |
| --- | --- | --- | --- |
|  | Week 1 | Week 2 | Week 3 |
| Voorbereiding |  |  |  |
| Git opzetten | Jasper |  |  |
| Git en Eclipse synchroniseren | Iedereen |  |  |
| Documentatie schrijven | Jorrick, Monique en Stuart |  |  |
| Realisatie |  |  |  |
| Klassen indelen |  | Iedereen |  |
| Uitvoering |  | Iedereen |  |
| Testen |  | Iedereen |  |
| Oplevering |  |  |  |
| Opleveren |  |  | Iedereen |

# Functioneel ontwerp

## Scenario

Speler opent het programma. Speler kiest een spel om te spelen in het hoofdmenu. Het gekozen spel opent. In dit geval is dat Getallen onthouden. Op het scherm ziet de speler een getal en een invoervak. Dit getal verdwijnt na 10 seconden waarna de speler de kans krijgt om dat getal in te vullen in het invoervak. Het invoerGetal wordt vergeleken met het teRadenGetal. Indien dit juist is wordt er een nieuw langer getal, met meer cijfers, weergegeven. Het aantal keer dat de speler juist heeft geraden wordt bijgehouden in een score. Dit wordt weergegeven op het scherm.

## Use- case



## Functionaliteiten

De speler kan een spel kiezen in het hoofdmenu door middel van knoppen.

De speler kan, tijdens het spelen van het spel Getallen onthouden, in het invoervak een getal invoeren.

De speler kan, tijdens het spelen van het spel Getallen onthouden, op de invoerKnop drukken om het ingevoerde getal te laten vergelijken met het teRadenGetal.

## Moscow analyze

**Musts**

Een werkend spel, Getallen onthouden

**Shoulds**

Mooie opmaak

**Coulds**

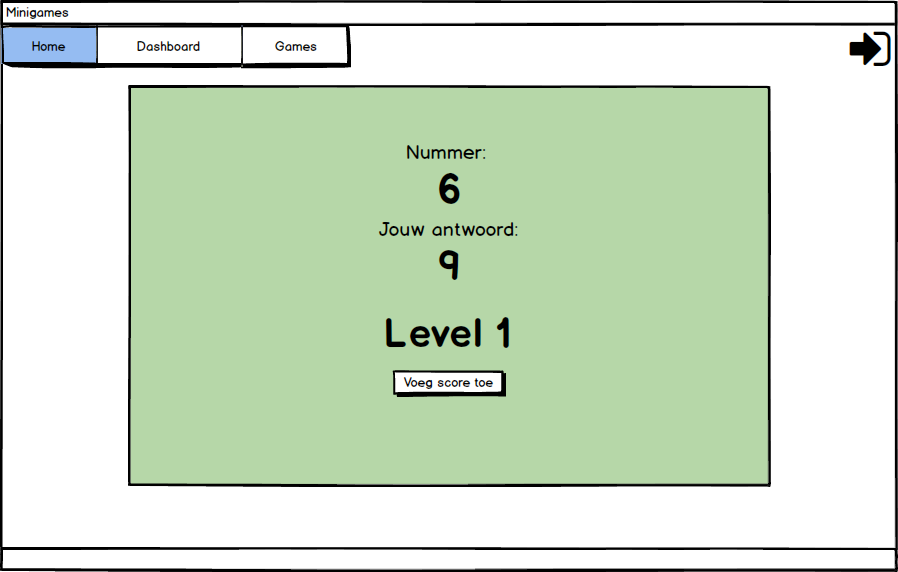
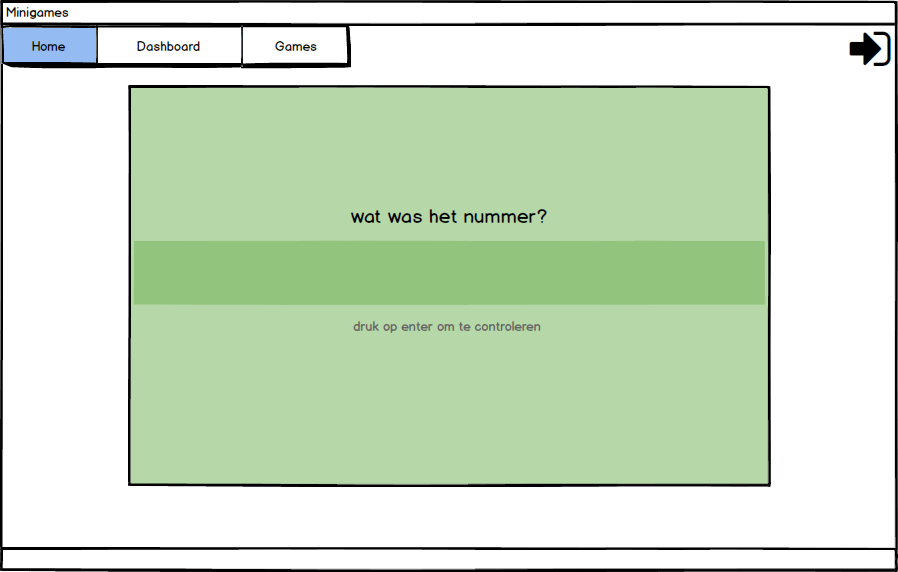
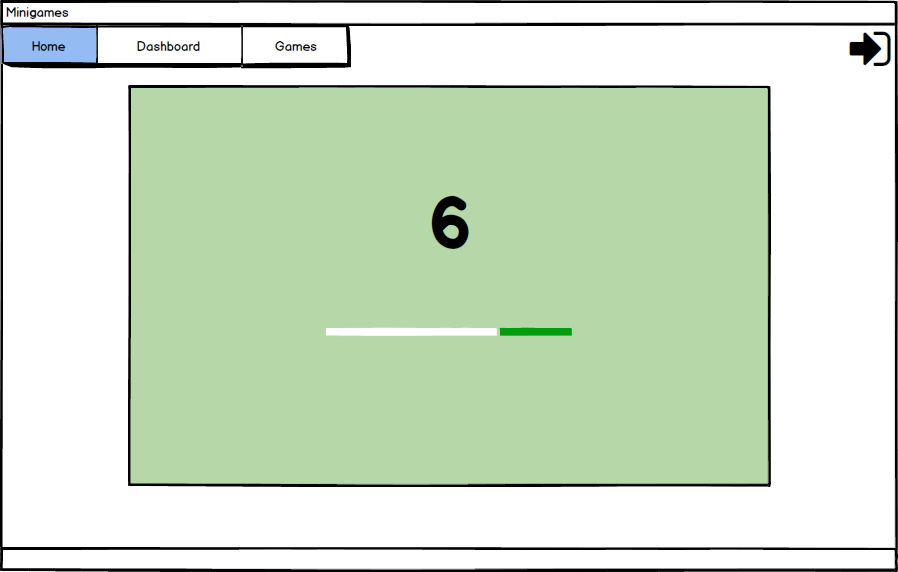
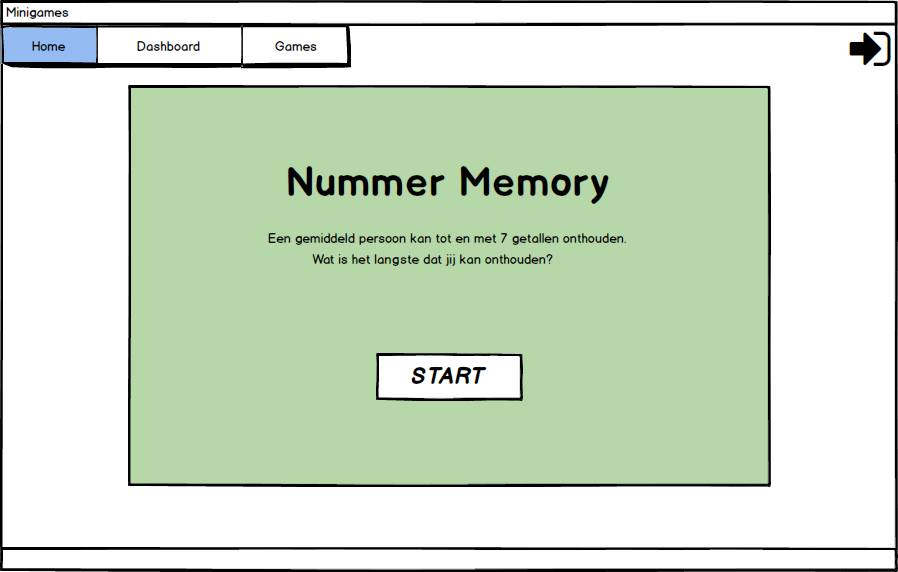
Een extra spel

**Won’t**

Muziek

3d effecten

## Schetsen



# Technisch ontwerp

## Klassendiagram

C:\Users\MoniqueMaquiné\Documents\Java\Hoofdstuk 10\klassendiagram.png

## Benodigde software

Om de Mini-games te kunnen maken hebben we de IDE Eclipse nodig om de Java code mee te schrijven.

Om de code met elkaar te kunnen delen gebruiken we Git. Hierin maken wij een Master branch, daar mag niemand aankomen, alleen Jasper de Git master, mag hier iets aanpassen. Dit doet hij wanneer er iets echt af is. Ook maken we een Development branch waar iedereen in kan werken.

## Technieken

Om de spellen te maken gebruiken wij de MVC methode. We maken models, views en controllers. Verder gebruiken wij de normale ButtonHandler en ActionHandlers die de acties van de speler behandelen. We gebruiken tekstvakken en labels om waarden op te halen en weer te geven.