

# FaculTime: een spel met een uitroepteken!

Daan de Groot, Joris den Elzen, Joris Reigâh, Bas Vogelzang

June 2016

## Abstract

Onderweg op wintersportvakantie was er een probleem: Laurens en Daan, studenten Natuurkunde en Informatica, verveelden zich. Helaas was een pakje kaarten niet voorradig, dus moesten ze zelf maar iets verzinnen. Ze zochten hun toevlucht tot de digitale klok die in de bus hing. De opdracht was simpel: maak 24 van de afzonderlijke getallen door wiskundige operatoren te gebruiken. Eenmaal aangekomen op wintersport bleken ze dit spel meer te willen spelen. Daarom breidden ze het spel uit naar dobbelstenen. De naam? FaculTime!

## 1 Beschrijving

### 1.1 Inleiding

De applicatie die we ontwikkeld hebben heet FaculTime. De naam is afgeleid van de Engelse termen *factorial* en *time*, welke centraal staan in onze app. De gebruiker kiest een aantal dobbelstenen, zeg  $n$ , en moet door elke dobbelsteen exact één keer te gebruiken, met behulp van wiskundige operatoren  $n$  faculteit maken. Dit uiteraard in een zo kort mogelijke tijd.

### 1.2 Productverantwoording

Een spel voor een smartphone moet volgens ons aan een aantal voorwaarden voldoen: het moet laagdrempelig zijn, het moet niet te lang duren, het moet makkelijk zijn om te spelen, maar moeilijk om er goed in te zijn. Het concept van FaculTime voldoet naar onze mening volledig aan deze beschrijving. Verder wilden (of beter: willen) we er een competitie-element aan toevoegen door er een multiplayer spel van te maken. Hierin moet elke speler een vijftal puzzels zo snel mogelijk oplossen. Degene die het snelst is wint.

### 1.3 Specificaties

Bij het openen van het spel kun je in het startmenu kiezen om te spelen met een aantal dobbelstenen van 3 tot en met 8. Na deze keuze wordt het spel geopend, waar met een animatie het gekozen aantal dobbelstenen geworpen en getoond

wordt. Nu moet de speler deze dobbelstenen en de benodigde operatoren zo neer zetten dat je het juiste resultaat krijgt. De game zelf kan gezien worden in de figuren 1 en 2.

## 2 Ontwerp

### 2.1 Globaal ontwerp

Ons product kan opgesplitst worden in verschillende componenten. Ten eerste heb je natuurlijk alles wat de gebruiker ziet. De layout, de imageviews en alle andere visuele dingen. De imageviews worden aangepast door de verschillende handlers voor de dobbelstenen of operatoren. Alle berekeningen worden achter de schermen bijgehouden. Door het samenwerken van alle componenten weet het spel wat er precies gebeurt en wanneer het klaar is.

### 2.2 Detailontwerp

#### 2.2.1 Dobbelsteen

Uiteraard hebben we het object dobbelsteen. Deze heeft twee constructoren: een constructor waarbij het aantal ogen wordt gekozen, en een constructor waarbij het aantal ogen random wordt bepaald. Verder kan je op elk moment het aantal ogen opvragen en het aantal ogen zetten als dat nodig is. Ook kan je op elk moment de dobbelsteen opnieuw rollen. Tot slot kun je de bijbehorende drawable opvragen.

#### 2.2.2 Operator

Ook hebben we een object operator. Een operator is een enumeratie met een methode om de operatie uit te voeren. Als de operator bijvoorbeeld plus is, zal  $a+b$  worden uitgerekend. Verder kan je ook de bijbehorende drawable opvragen.

#### 2.2.3 Game

Deze klasse is de interface voor het spel. Hierin worden de dobbelstenen opgevraagd, evenals de operatoren. Ook wordt hier gecontroleerd of er al een juiste oplossing is gevonden en in welke tijd dat is gebeurd. Ook worden hier de swaps van dobbelstenen en operatoren uitgevoerd.

### 2.3 Ontwerpverantwoording

We zijn helaas aangelopen tegen het feit dat het scherm te smal is voor de UI die we oorspronkelijk in gedachte hadden. Daarom hebben we gekozen om dit compleet op de schop te nemen, en hierbij hebben we zelfs een nieuwe manier van rekenen geïmplementeerd. Hierbij worden de operatoren van links naar rechts uitgelezen waardoor haakjes niet meer nodig zijn. Hierdoor kunnen we alles toch in de breedte van het scherm kwijt. We zijn van mening dat dit niet

de mooiste manier is om dit te realiseren, liever zouden we een boomstructuur gebruiken, maar vanwege de tijd hebben we gekozen om toch deze implementatie te gebruiken. In de toekomst zijn we van plan te bekijken om alsnog de boomstructuur te implementeren.

### **3 Reflectie**

We zijn, net als vele anderen, aangelopen tegen het feit dat tijd schaarser is dan we zouden willen. We hebben daarom nog een hoop dingen die we graag zouden implementeren, maar die vanwege de deadline niet mogelijk bleken. Voorbeelden hiervan zijn de boomstructuur, maar ook een volledig werkende multiplayer. Omdat we geen halve features wilden brengen hebben we die daarom volledig uit de app gelaten, maar zeker de multiplayer is bijna goed genoeg om gepubliceerd te worden. Daarom voelt het dubbel om te stellen dat we tevreden zijn over onze app. Voor een eerste app is deze zeker niet verkeerd, maar er missen simpelweg een aantal dingen die met wat extra tijd prima geïmplementeerd hadden kunnen zijn.



Figure 1: Start screen

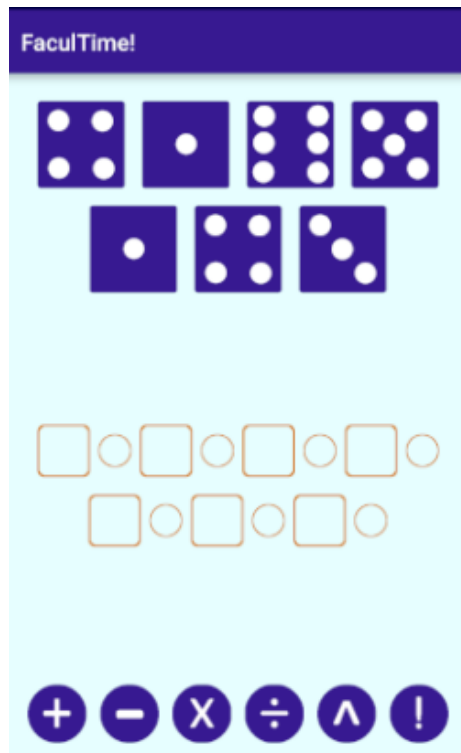


Figure 2: Gamelayout