

Case Study OOAD: Finch

1 Inleiding

Het bedrijf PeeWee Games wil een kennisquiz-applicatie Finch ontwikkelen die beschikbaar komt op de meest gangbare devices.

PeeWee Games vraagt aan AIM-studenten om een analyse en ontwerp te maken voor deze kennisquiz. Hieronder wordt de volledige functionaliteit van Finch beschreven.

Er zijn een paar randvoorwaarden aan deze opdracht verbonden: je moet in groepjes van twee studenten een objectgeoriënteerde analyse en ontwerp in UML maken volgens de richtlijnen gegeven in Larman, en je moet dit ontwerp richten op een implementatie van de logica in Java. Tevens bouw je (een deel van) een Java-applicatie voor het spelen van een individuele quiz.

2 Beschrijving van spel

Registreren

Om de kennisquiz Finch te kunnen spelen moet je je registreren door een gebruikersnaam en wachtwoord op te geven. Nadat je je hebt geregistreerd ontvang je een startsaldo van 100 munten en krijg je twee vragenlijsten cadeau. Verderop wordt toegelicht wat er onder een vragenlijst wordt verstaan.

Thema's en vragenlijsten

Finch kent thema's zoals *Sport* en *Muziek*. Een thema bevat diverse vragenlijsten die bestaan uit een grote serie vragen die bij een specifiek onderwerp horen binnen het betreffende thema. Zo zijn er vragenlijsten *Sport - voetbal*, *Sport – atletiek*, *Muziek - pop & rock* en *Muziek - dance*.

Quizvragen

De vragen van de quiz zijn zoals eerder aangegeven ingedeeld in vragenlijsten. Er zijn meerkeuzevragen (met steeds vier alternatieven) en open vragen (short answer). Elke vraag hoort bij precies één vragenlijst. Een voorbeeld van een meerkeuzevraag uit de vragenlijst *Sport - voetbal*:

Wie was de bondscoach van het Nederlands Elftal tijdens het WK voetbal van 2014 in Brazilië?
A. Frank de Boer
B. Louis van Gaal
C. Ronald Koeman
D. Bert van Marwijk

Een voorbeeld van een open vraag uit de vragenlijst *Muziek - pop & rock*:

Welke band stond in 2020 in de hitparade met het nummer <i>Soldier On</i> ?

Bij deze laatste vraag is in Finch vastgelegd dat meerdere antwoorden correct zijn, namelijk *Di-Rect* en *Direct*.

De eerste versie van Finch is op de Nederlands markt gericht. Dat geldt dus ook voor de vragen. In de toekomst dient Finch ook beschikbaar te komen in andere landen.

Spelverloop quiz

Als je ervoor kiest om een quiz te spelen, dan selecteer je een vragenlijst en krijg je vervolgens 10 vragen uit deze lijst voorgelegd. Dat is steeds een mix van meerkeuzevragen en open vragen. Je beantwoordt de vragen een voor een. Daarbij kun je niet terug naar een vorige vraag om het antwoord aan te passen. Per goed beantwoorde vraag verdien je punten. Verder krijg je een bonus op basis van de totale tijd die nodig was om de quiz te doorlopen. Hoe sneller je bent, hoe meer bonuspunten je verdient. Als je alle vragen goed hebt, dan krijg je extra punten en ook nog 2 munten. Voor elke speler wordt per vragenlijst zijn 'lifetime best' bijgehouden.

Het spreekt voor zich dat Finch-spelers bij het beantwoorden van de vragen geen hinderlijke vertragingen in de applicatie mogen ondervinden.

Het beheren van vragen

Voor medewerkers van PeeWee Games moet het mogelijk zijn om vragen te beheren. Nieuwe vragen, meerkeuze of open, met de bijbehorende antwoorden moeten kunnen worden toegevoegd en aan een bestaande vragenlijst worden gekoppeld. Ook het aanpassen en verwijderen van vragen (inclusief antwoorden) moet mogelijk zijn. Verder moeten ook thema's en bijbehorende vragenlijsten kunnen worden toegevoegd en gewijzigd.

Finch-shop

Een speler kan vragenlijsten kopen in de Finch-shop. De speler moet om dat te kunnen natuurlijk wel over voldoende munten beschikken. De aankoopbedragen (in munten) van de verschillende vragenlijsten kunnen verschillen. Het aankoopbedrag van de aangeschafte lijst wordt dan in mindering gebracht op het saldo van de speler. Van een eenmaal aangeschafte vragenlijst kun je een jaar gebruik maken.

Een speler kan ook extra munten kopen. Het afrekenen daarvan gebeurt via een online betaalservice.

Puntentelling

Hierboven is de puntentelling bij een quiz (globaal) beschreven. Hoeveel punten je precies krijgt voor een goed beantwoorde vraag en hoeveel bonuspunten bij alle vragen goed, moet nog nader worden gespecificeerd.

Bij PeeWee Games bestaat de wens om de puntentelling flexibel te maken, d.w.z. er moet makkelijk overgestapt kunnen worden op een andere systematiek.

3. Opdracht

Je wordt gevraagd om de analyse en het ontwerp voor het volledige systeem uit te voeren. Voer de onderstaande stappen uit. Beschrijf de resultaten van deze stappen in een SRS (Software Requirements Specification) en een SDD (Software Design Description), inclusief relevante toelichting bij diagrammen en onderbouwing van de belangrijkste keuzes. Templates voor SRS en SDD vind je op OnderwijsOnline.

1. Stel een use case model op dat bestaat uit een use case diagram en de use case beschrijvingen in brief format.
2. Stel de overige functionele eisen (d.w.z. aanvullend op de use cases) en de niet-functionele eisen op, genummerd en ingedeeld volgens FURPS+. In de tekst staan een paar hints voor niet-functionele eisen. Deze hints moet je nog wel wat SMARTer maken.
3. Beschrijf de use cases, die corresponderen met de beschrijving van het spel onder de kopjes *Registren*, *Spelverloop quiz* en *Finch-shop*, in fully-dressed format.
Gebruik een activity diagram om de flow van de meest complexe use cases weer te geven.
4. Maak een domeinmodel dat alles bevat dat relevant is voor het spelen van de quiz.
Ook de antwoorden die een speler geeft op de quizvragen die hij krijgt voorgelegd, dienen in het domeinmodel te worden opgenomen.
5. Werk de belangrijkste systeemoperaties van de use case(s), die je bij stap 3 hebt uitgeschreven in fully dressed format, uit met behulp van sequence diagrams.
Opmerking: Stel voor use cases met meer dan één systeemoperatie eerst een system sequence diagram op en maak vervolgens per systeemoperatie een sequence diagram.
6. Stel een design class diagram op.
7. Pas het Strategy Pattern toe voor het flexibiliseren van de puntentelling. Laat in zowel het design class diagram als ook in het betreffende sequence diagram zien hoe het Strategy Pattern wordt toegepast.
Mogelijk is er nog een ander GoF design pattern dat voor de hand ligt om te gebruiken. Denk daar eens goed over na en werk het, indien van toepassing, uit in het SDD.
8. Beschrijf aan de hand van een aantal goed gekozen voorbeelden, die zijn te zien in het design class diagrams en/of sequence diagrams, hoe je OO ontwerpprincipes (denk met name aan SOLID, GRASP en GoF design patterns) hebt toegepast.

9. Realiseer een console applicatie in Java waarmee de quiz kan worden gespeeld (zie de beschrijving hierboven onder het kopje **Spelverloop quiz**).
- Opmerking: De data voor deze applicatie mag gemockt worden. Je kunt hiervoor zelf een class maken die een methode bevat die quizvragen van voorgedefinieerde vragenlijsten retourneert.*

Enkele opmerkingen en tips:

- Mochten er onduidelijkheden zijn over het domein, neem dan contact op met de opdrachtgever. De docent zal deze rol op zich nemen.
- Vraag tijdig feedback op de diagrammen en modellen die je opstelt.
- Lever in lesweek 4 of 5 de eerste versie van het SRS in bij de docent. Je ontvangt dan feedback.

Inleveren

Het eindproduct (SRS, SDD en de Java-code) dient via iSAS te worden ingeleverd. Zie het toetsrooster in iSAS voor de inleverdeadline.