

Opdracht Ontwikkelstraten

Beginsituatie

Gegeven is PHP-code voor een website waarmee het spel Hive gespeeld kan worden. De spelregels van Hive zijn te vinden op Blackboard. Op dit moment kan je het spel met twee spelers die achter dezelfde computer zitten spelen.

Deze code is te vinden op de Github-repository [hanze-hbo-ict/itvb23ows-starter-code](https://github.com/hanze-hbo-ict/itvb23ows-starter-code). Deze code bevat een aantal bugs, missende features en problemen met betrekking tot de kwaliteit van de code.

De applicatie vereist een databasekoppeling. Hiervoor heb je een MySQL-server nodig. In het bestand `database.php` vind je de instellingen hiervoor; het bijbehorende databaseschema staat in `hive.sql`.

Bugs

1. De dropdown die aangeeft welke stenen een speler kan plaatsen bevat ook stenen die de speler niet meer heeft. Bovendien bevat de dropdown die aangeeft waar een speler stenen kan plaatsen ook velden waar dat niet mogelijk is, en bevat de dropdown die aangeeft vanaf welke positie een speler een steen wil verplaatsen ook velden die stenen van de tegenstander bevatten.
2. Als wit een bijenkoningin speelt op (0, 0), en zwart op (1, 0), dan zou het een legale zet moeten zijn dat wit zijn koningin verplaatst naar (0, 1), maar dat wordt niet toegestaan.
3. Als wit drie stenen plaatst die geen bijenkoningin zijn, mag hij als vierde zet helemaal geen steen spelen. Het spel loopt dan dus vast.
4. Als je een steen verplaatst, kan je daarna geen nieuwe steen spelen op het oude veld, ook als dat volgens de regels wel zou mogen.
5. De *undo*-functionaliteit werkt nog niet goed. De oude zetten worden nog niet verwijderd, en de toestand van het bord wordt niet altijd goed hersteld. Bovendien kan je ook undo'en als er nog geen zetten gedaan zijn, en dan lijkt het erop dat je een toestand uit een ander spel ziet.

Feature requests

1. De sprinkhaan is nog niet geïmplementeerd. Implementeer de regels om deze steen te bewegen.

- a. Een sprinkhaan verplaatst zich door in een rechte lijn een sprong te maken naar een veld meteen achter een andere steen in de richting van de sprong.
 - b. Een sprinkhaan mag zich niet verplaatsen naar het veld waar hij al staat.
 - c. Een sprinkhaan moet over minimaal één steen springen.
 - d. Een sprinkhaan mag niet naar een bezet veld springen.
 - e. Een sprinkhaan mag niet over lege velden springen. Dit betekent dat alle velden tussen de start- en eindpositie bezet moeten zijn.
2. De soldatenmier is nog niet geïmplementeerd. Implementeer de regels om deze steen te bewegen.
 - a. Een soldatenmier verplaatst zich door een onbeperkt aantal keren te verschuiven
 - b. Een verschuiving is een zet zoals de bijenkoningin die mag maken.
 - c. Een soldatenmier mag zich niet verplaatsen naar het veld waar hij al staat.
 - d. Een soldatenmier mag alleen verplaatst worden over en naar lege velden.
3. De spin is nog niet geïmplementeerd. Implementeer de regels om deze steen te bewegen.
 - a. Een spin verplaatst zich door precies drie keer te verschuiven.
 - b. Een verschuiving is een zet zoals de bijenkoningin die mag maken.
 - c. Een spin mag zich niet verplaatsen naar het veld waar hij al staat.
 - d. Een spin mag alleen verplaatst worden over en naar lege velden.
 - e. Een spin mag tijdens zijn verplaatsing geen stap maken naar een veld waar hij tijdens de verplaatsing al is geweest.
4. De regels wanneer je mag passen zijn nog niet geïmplementeerd. Implementeer deze regels.
 - a. Een speler mag alleen passen als hij geen enkele steen kan spelen of verplaatsen, dus als hij geen enkele andere geldige zet heeft.
5. Het spel geeft nog niet aan wanneer iemand gewonnen heeft of wanneer er sprake is van een gelijkspel.
 - a. Een speler wint als alle zes velden naast de bijenkoningin van de tegenstander bezet zijn.
 - b. Als beide spelers tegelijk zouden winnen is het in plaats daarvan een gelijkspel.
6. Voeg de mogelijkheid toe om tegen een AI te spelen. Op de Github-repository [hanze-hbo-ict/itvb23ows-hive-ai](https://github.com/hanze-hbo-ict/itvb23ows-hive-ai) vind je een Python-implementatie van een AI. Deze AI moet in een aparte container draaien, en de PHP-applicatie moet HTTP-requests gebruiken om de AI aan te spreken. Je kan de documentatie van de API vinden in de `README.md` in repository van de API.

Mogelijk zal de AI zetten doen die ongeldig zijn op grond van de interpretatie van de regels zoals je applicatie die heeft, maar dit mag je negeren. Je mag gewoon de zet uitvoeren die de AI voorstelt, ook als deze niet geldig is.

Opdracht

Lever een portfolio in, als ZIP- of RAR-archief, waarin je aantoont dat je de onderstaande opdrachten hebt uitgevoerd. De portfolio bevat in ieder geval een bundle van je eigen repository met alle branches, relevante configuratiefiles, screenshots die aantonen dat je de opdrachten hebt gemaakt en argumentatie voor gemaakte keuzes.

Planning

Onderstaande weekplanning is indicatief; deze sluit aan bij de onderwerpen van de hoorcolleges en werkcolleges, maar het is toegestaan om een ander tempo te hanteren.

Week 1

Maak op Github een eigen fork van de gegeven repository. Zet de issues en features in de issue tracker van je eigen repository. Als de issue tracker niet aanstaat in je fork, kan je die aanzetten in de settings van je repository, onder het kopje *Features*.

Bewijsmiddelen: Screenshots van de Github repository, de issue tracker en een aantal issues, en een Git bundle met repository. Gebruik om de bundle te maken het commando `git bundle create hive.bundle --all`. Maak deze bundle pas als alle onderstaande opdrachten zijn afgerond. Vermeld ook de link naar de repository op Github bij het inleveren.

Week 2

Maak Dockerfiles en bijbehorende configuratie zodat je de applicatie en de bijbehorende database als Docker containers kan draaien.

Bewijsmiddelen: De Dockerfiles en andere benodigde configuratiebestanden voor de containerconfiguratie en screenshots van de werkende containeromgeving.

Week 3

Configureer een pipeline in Jenkins zodat je continuous integration kan toepassen bij de verdere ontwikkeling van de applicatie. Beschrijf ook welke branchingstrategie je wilt toepassen en beargumenteer dit.

Bewijsmiddelen: De Jenkinsfile, screenshots van de werkende Jenkinsomgeving en een PDF of Word-document met de uitleg en verantwoording van de gekozen branchingstrategie.

Week 4

Pas de pipeline in Jenkins aan zodat deze ook SonarQube gebruikt om de kwaliteit van de code te controleren. Maak een rapport met SonarQube en beschrijf aan de hand hiervan de kwaliteitsproblemen in de gegeven code en mogelijke oplossingen daarvoor. Beargumenteer

je oplossingen. Pas de gegeven code aan zodat deze kwalitatief beter wordt, zowel op microniveau (de individuele regels code) als op macroniveau (de verdeling van code over verschillende modules).

Bewijsmiddelen: De aangepaste Jenkinsfile, de uitvoer van SonarQube en een PDF of Word-document met de geconstateerde kwaliteitsproblemen en de mogelijke oplossingen. De verbeterde code dient onderdeel te zijn van de ingeleverde repository.

Week 5

Los bugs 1 tot en met 4 in de codebase op. Schrijf ook unit tests om te voorkomen dat deze bugs later weer optreden.

Bewijsmiddelen: De unit tests en aangepaste code dienen onderdeel te zijn van de ingeleverde repository.

Week 6

Implementeer features 1 tot en met 5. Gebruik hiervoor test-driven development.

Bewijsmiddelen: De unit tests en aangepaste code dienen onderdeel te zijn van de ingeleverde repository.

Week 7

Implementeer feature 6. Gebruik hiervoor test-driven development en test doubles. Schrijf ook een Dockerfile om de voor deze feature benodigde container te configureren. Los daarnaast bug 5 op. Schrijf hierbij unit tests met test doubles om te voorkomen dat deze bugs later weer optreden.

Bewijsmiddelen: De Dockerfile en andere benodigde configuratiebestanden voor de containerconfiguratie en screenshots van de werkende containeromgeving. De unit tests en aangepaste code dienen onderdeel te zijn van de ingeleverde repository.