**Systeemtestplan**

Project: Snakes

Project team: Project Snake

Teamleden: Roel Lucassen (386216)

Opdrachtgever: Cornelissen, Jeffrey J.R.

Versie: 1.0

Versiedatum: 20 februari 2019

Status: Concept

**Documenthistorie**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versie** | **Wijzigingen** | **Auteur** | **Datum** |
| 0.1 |  |  |  |

**H1 Inleiding**

Ik ga een online snake spel maken met een paar twists. Ten eerste wordt je niet groter waardoor je voor jezelf en de tegenstander moet oppassen. Ten tweede zullen er muren in het spel komen waardoor de map elke keer anders is. Ten derde zullen er power-ups in het spel komen waardoor alles interessanter wordt. Zie de requirements in het analysedocument.

Deze applicatie moet ook getest worden. In dit document laat zien hoe en waarop de applicatie getest gaat worden. Hierbij wordt de risicoanalyse, de teststrategie en de testcases behandeld.

**H2 Uitgangssituatie**

In mijn Uitgangssituatie ga ik uit van een lokale versie en een gedistribueerde versie. Hiervoor gebruik ik een lokale of online database gevuld met test data.

Op de lokale versie zal de spel functionaliteiten getest worden.

Op de gedistribueerde zullen de online functies getest worden, Hier gaat het over Inloggen, Registreren en kijken of de clients met elkaar communiceren.

**H3 Productrisicoanalyse**

*bepalen: faalkans, schade, risicoklasse en ernstklasse. De risicoklasse is gelijk aan schade \* faalkans. De ernstklasse is gebaseerd op de risicoklasse. Geef aan hoe de ernstklasse wordt bepaald. Bijvoorbeeld:*

*Schade: 1 = Laag, 2 = Midden, 3 = Hoog*

*Faalkans: 1 = Laag, 2 = Midden, 3 = Hoog*

*Risicoklasse = Schade \* Faalkans*

*Ernstklasse C: risicoklasse ≤ 2*

*Ernstklasse B: 2 < risicoklasse ≤ 6*

*Ernstklasse A: risicoklasse > 6*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Testdoel** | **Schade** | **Faalkans** | **Risicoklasse** | **Ernstklasse** |
| F1 | 3 | 1 | 6 | B |
| F2: | 2 | 2 | 4 | B |
| F3 | 3 | 3 | 9 | A |
| F4 | 1 | 1 | 2 | C |
| F5 | 1 | 1 | 1 | C |
| F6 | 1 | 2 | 2 | B |
| F7 | 2 | 1 | 2 | B |
| R1 | 3 | 1 | 3 | B |
| R2 | 3 | 2 | 6 | A |
| R3 | 1 | 1 | 1 | C |
| R4 | 1 | 1 | 1 | C |
| R5 | 3 | 2 | 6 | B |
| R6 | 2 | 2 | 4 | B |
| R7 | 2 | 1 | 2 | C |
| R8 | 1 | 1 | 1 | C |
| R9 | 3 | 2 | 6 | B |
| Kwaliteits Q1 | 2 | 3 | 6 | B |
| Kwaliteits Q2 | 2 | 1 | 2 | B |
| Kwaliteits Q3 | 1 | 1 | 1 | C |
| Kwaliteits Q4 | 2 | 1 | 2 | C |
| Kwaliteits Q5 | 3 | 2 | 6 | B |
| Kwaliteits Q6 | 1 | 1 | 1 | C |

**H4 Teststrategie**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Testdoel** | **PRA** | **UT** | **CT** | **IT** | **ST handm.** | **ST autom.** |
| F1 | B | ⬤ | ⬤⬤ | ⬤ |  |  |
| F2 | B | ⬤ | ⬤⬤ | ⬤ |  |  |
| F3 | B | ⬤⬤ | ⬤⬤⬤ | ⬤⬤ |  |  |
| F4 | B |  | ⬤⬤ | ⬤⬤⬤ |  |  |
| F5 | C | ⬤ | ⬤ | ⬤⬤⬤ |  |  |
| F6 | B | ⬤⬤⬤ | ⬤⬤ |  |  |  |
| F7 | C | ⬤⬤ | ⬤⬤ |  |  |  |
| F8 | B |  | ⬤ | ⬤ |  |  |
| F9 | B | ⬤ | ⬤⬤ | ⬤⬤ |  |  |
|  |  |  |  |  |  |  |
| R1 | B |  | ⬤⬤ |  |  |  |
| R2 | B | ⬤⬤ |  |  |  |  |
| R3 | C | ⬤ |  |  |  |  |
| R4 | B |  | ⬤⬤ |  |  |  |
| R5 | C | ⬤ |  |  |  |  |
| R6 | C |  | ⬤ |  |  |  |
| R7 | B |  | ⬤⬤ |  |  |  |
| R8 | B | ⬤⬤ |  |  |  |  |
| R9 | B |  | ⬤⬤ |  |  |  |
| R10 | B |  | ⬤⬤ |  |  |  |
|  |  |  |  |  |  |  |
| Q1 | B |  |  | ⬤⬤ |  |  |
| Q2 | B |  | ⬤⬤ |  |  |  |

Toelichting bij bovenstaande tabel:

|  |  |
| --- | --- |
| PRA-RK | Risicoklasse uit de productrisicoanalyse (PRA): risicotabel |
| Toetsen | Toetsing/review van de verschillende tussenproducten zoals requirements, functioneel ontwerp, technisch ontwerp |
| Ontwikkeltest | Unittest en Unitintegratietest |
| ST | Systeemtest |
| IT | Integratie test |
| CT | Component test |
| UT | Unit Test |
| ⬤ | beperkte dynamische test |
| ⬤⬤ | gemiddelde dynamische test |
| ⬤⬤⬤ | zware dynamische test |

**H5 Logische testcases**

*In dit hoofdstuk worden de logische testcases beschreven. Een logische testcase is afgeleid van de requirements uit het analyse document. In het algemeen kunnen meerdere logische testcases worden gedefinieerd voor een requirement.*

*Voorbeeld van logische testcases:*

|  |  |
| --- | --- |
| **ID** | **Omschrijving** |
| TC-1 | Een slang beweegt naar links,boven,rechts en onder. |
| TC-2 | Een slang pakt een pop-up. |
| TC-3 | Een slang raakt een muur. |
| TC-4 | Een slang raakt territory. |
| TC-5 | Een slang spawnt op niks. |
| TC-6 | Een slang spawnt op een powerup. |
| TC-7 | Een slang spawnt op een muur. |
| TC-8 | Een slang spawnt op territory. |
| TC-9 | Een slang sterft. |
| TC-10 | Een slang sterft drie keer. |
| TC-11 | Inloggen. |
| TC-12 | Verkeerd inloggen met verkeerde naam en wachtwoord. |
| TC-13 | Registreren met een al bestaande naam. |
| TC-14 |  |

**H6 Fysieke testcases**

|  |  |
| --- | --- |
|  | De test is geslaagd en er is niks op aan te merken. |
|  | De test lijkt goed te gaan, maar er is iets op het behaalde resultaat aan te merken. |
|  | De test is gefaald. |
|  | Deze test is niet van toepassing op de huidige versie van de applicatie en is ook niet uitgevoerd. |
|  | Dit is geen test maar een actie die ondernomen moet worden om daar op volgende tests uit te kunnen voeren (initialisatie actie). |

|  |  |  |
| --- | --- | --- |
| **ID** | **Omschrijving** | **Resultaat** |
| TC-1 | Je spawnt een slang op het bordt met de x waarde =0 en de y waarde 0. Zet de directie naar links, dan boven dan rechts en dan naar beneden.  Verwacht Resultaat: De slang maakt een vierkant met de volgorde: links, boven, rechts en dan beneden |  |
| TC-2 | Je spawnt een slang en laat hem een fast powerup pakken.  Verwacht Resultaat: de snelheid van de slang gaat van 250 naar 125 voor 8 seconden. |  |
| TC-3 | Je spawnt een slang en laat hem tegen een muur aanknallen.  Verwacht resultaat: Alle territory van de slang zal weg gaan en daarnaast zullen de levens van de slang van 3 naar 2 gaan. |  |
| TC-4 | Je spawnt een slang en laat hem tegen territory aanknallen.  Verwacht resultaat: Alle territory van de slang zal weg gaan en daarnaast zullen de levens van de slang van 3 naar 2 gaan. |  |
| TC-5 | Laat de slang op een lege grid spawnen.  Verwacht resultaat: De slang spawnt op het lege grid. |  |
| TC-6 | Laat de slang op een powerup spawnen.  Verwacht Resultaat: Slang kan niet spawnen en de gebruiker krijgt een melding. |  |
| TC-7 | Laat de slang op territory spawnen.  Verwacht Resultaat: Slang kan niet spawnen en de gebruiker krijgt een melding. |  |
| TC-8 | Laat de slang tegen een muur aan knallen.  Verwacht Resultaat: De slang zal sterven en de levens zullen van 3 naar 2 gaan.. |  |
| TC-9 | Laat de slang drie keer sterven door hem tegen een muur aan te laten knallen.  Verwacht Resultaat: Gebruiker krijgt een melding dat hij heeft verloren |  |
| TC-10 | Een gebruiker logt in op de applicatie met de volgende credentials:  Username: SnakeLover123  Password: Snake  Verwacht Resultaat:  De speler wordt ingelogd. |  |
| TC-11 | Een gebruiker registreert op de applicatie met de volgende credentials:  Username: SnakeLover123  Password: Snake  Verwacht Resultaat: Speler registreert succesvol en kan nu inloggen met deze credentials. |  |
| TC-12 | Een gebruiker logt in op de applicatie met de volgende credentials:  Username: SnakeLover123  Password: Snake1111  Verwacht Resultaat:  De speler kan niet inloggen en krijgt een melding. |  |
| TC-13 | Een gebruiker registreert op de applicatie met de credentials die al eerder zijn gebruikt:  Username: SnakeLover123 Password: Snake  Verwacht Resultaat:  De speler kan niet registreren |  |

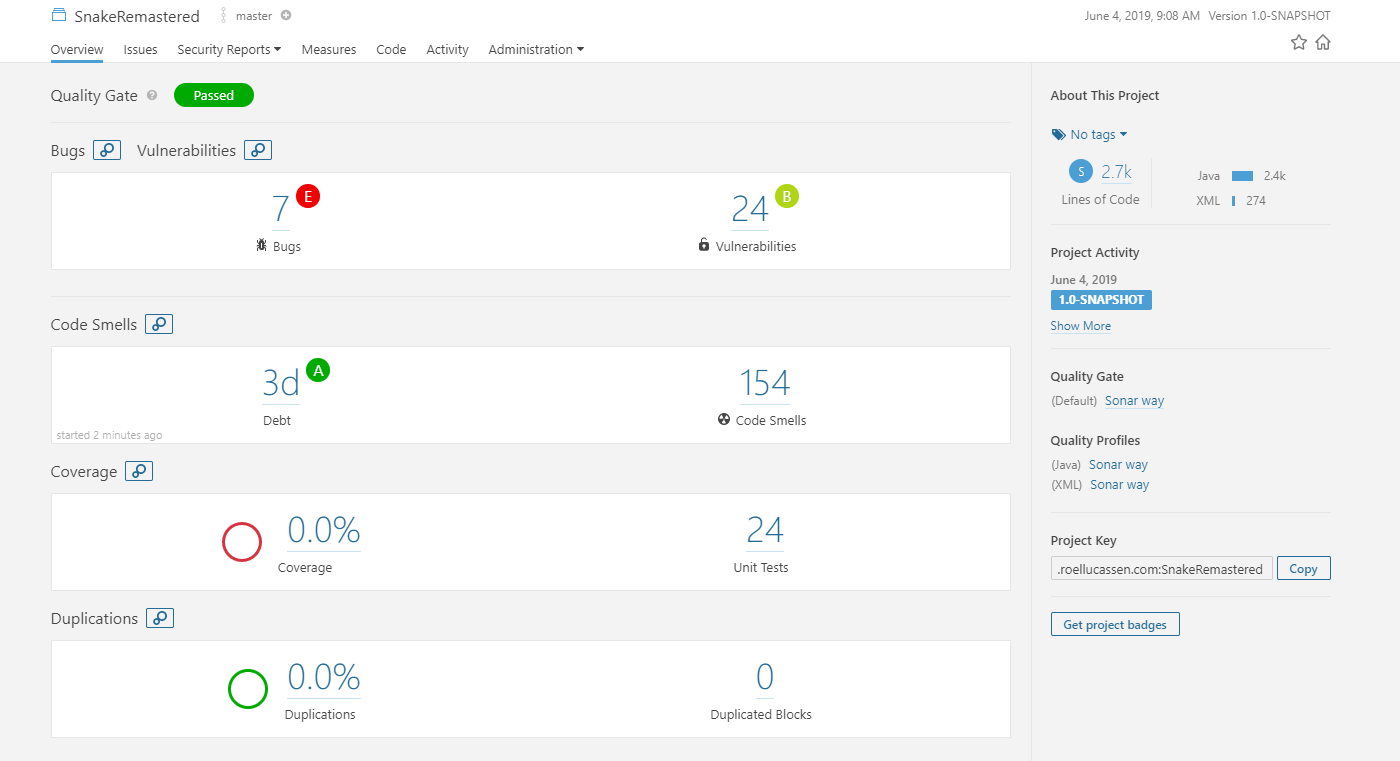
**H7 Testcoverage**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TC1 | TC2 | TC3 | TC4 | TC5 | TC6 | TC7 | TC8 | TC9 | TC10 | TC11 | TC  12 | TC  13 | TC  14 |
| F1 | X | X | X | X | X |  |  | X | X |  |  |  |  |  |
| F2 |  |  |  |  |  |  |  |  |  |  | X | X | X | X |
| F3 |  |  |  | X | X | X | X |  |  |  | X | X | X | X |
| F4 |  | X |  |  |  |  |  |  |  |  |  |  |  |  |
| F5 |  |  |  |  |  |  |  |  |  | X |  |  |  |  |
| F6 |  |  | X | X | X |  |  | X | X |  |  |  |  |  |

**H8 Unittesten en code coverage**

*In dit hoofdstuk wordt beschreven welke strategie je hebt gehanteerd bij het schrijven van de unittesten. Beschrijf voor welke klassen en methoden je de unittesten gaat schrijven en waarom je voor deze klassen/methoden hebt gekozen. Laat de testresultaten en code coverage zien in het testrapport, bijvoorbeeld door een screenshot te maken. Beschrijf ook je conclusies op basis van de testresultaten en code coverage.*

**H9 Statische code analyse**

*In dit hoofdstuk wordt beschreven hoe je statische code analyse gaat onderzoeken. Dit kan met behulp van SonarQube. Laat de resultaten van de analyse zien in het testrapport, bijvoorbeeld door een screenshot te maken. Beschrijf ook je conclusies op basis van de resultaten en geef aan op welke punten je mogelijkheden ziet om de code verder te verbeteren.*

Zoals je hierboven ziet zit er op het moment veel bugs en vulnerabilties in. De reden hiervoor is omdat dit ten eerste nog midden in het coderen is. Hierdoor is de code kwaliteit nog niet echt goed en de tweede reden ervoor is omdat de code nog niet gedubbel checkt is. Hierdoor zitten er veel vulnerabilities in de code. Echter veel van deze vulnerabilities zijn makkelijk te verwijderen. Bijvoorbeeld 9 vulnerabilities zijn er omdat ik per ongeluk ‘return’ in een finally block heb neergezet. Als ik dat zou fiksen ben ik meteen terug naar 15 vulnerabilities. Daarnaast heb ik wel veel code smells en deze zou ik goed moeten verminderen.