Testplan

Defend Your Castle

Opgesteld:

16-01-2020



Versienummer: 1.0

Opdrachtgever:

Frans de Boer

Assistent:

Richard Kingma

Door Bastiaan Kramer

I4O3A

Inhoudsopgave

[Inleiding 1](#_Toc30359262)

[Opdrachtformulering 1](#_Toc30359263)

[Rapportage 1](#_Toc30359264)

[Organisatie 1](#_Toc30359265)

[Afbakening 1](#_Toc30359266)

[Taken en verantwoordelijkheden 1](#_Toc30359267)

[Overzicht producten, kwaliteitseisen en stopcriteria 2](#_Toc30359268)

[Documentatie 2](#_Toc30359269)

[Testomgeving 3](#_Toc30359270)

[Versiebeheer 3](#_Toc30359271)

# Inleiding

Dit testplan heeft betrekking op alle hierna genoemde en (nog) relevante functionaliteiten van het product “Defend your Castle”. De resultaten zijn samengevat als volgt:  
  
De status van het product is: **CONCEPT**, wanneer de opdrachtgever tevreden is met de documentatie en de uitslagen zal het worden aangepast naar definitief.

# Opdrachtformulering

De testopdracht is als volgt: test het hele systeem aan de hand van de testscenario’s die worden aangekaart in hoofdstuk 4. De doelstellingen van de tests zijn om te bepalen of het systeem naar behoren functioneert, en dat het systeem aan alle gestelde eisen voldoet.

# Rapportage

Het rapporteren zal worden gedaan aan de opdrachtgever. De informatie die zal worden gerapporteerd is hoofdzakelijk ontstane problemen tijdens het testen, en het bespreken van de best mogelijke oplossing voor deze problemen. Het rapporteren zal worden gedaan door mij (de ontwikkelaar), het rapporteren zal twee keer worden gedaan: 1 keer na het afronden van de testresultaten met als informatie de ontdekte fouten en 1 keer nadat de fouten zijn opgelost (indien dit haalbaar is).

# Organisatie

Dit hoofdstuk heeft betrekking op het organiseren van de tests: wat wordt wel en niet getest, Waarin (en/of waarop) wordt het getest, hoe wordt het getest, wie test het en wanneer werkt het product naar behoren?

## Afbakening

Het te testen product betreft in dit geval een (simpele) videogame, gemaakt in Unity. Er zal worden getest op het doden van de vijanden die het kasteel bestormen en het aftrekken van de gezondheid van de muur wanneer ze tegen de muur staan. Ook zal er gekeken worden of de aangepaste cursor (het vizier) naar behoren werkt.

## Taken en verantwoordelijkheden

De tests zullen worden uitgevoerd door Teshale Noorman, het verwerken van deze resultaten en eventuele wijzigingsvoorstellen (opmerkingen) zal worden gedaan door Bastiaan Kramer (ontwikkelaar), de ontwikkelaar zal ook de leiding hebben in het testproces.

## Overzicht producten, kwaliteitseisen en stopcriteria

In dit overzicht worden alle aspecten genoemd die getest worden, wanneer ze voldoen aan de eisen en wanneer de tests (moeten) worden afgebroken.  
  
De lay-out van de testelementen zijn als volgt:

* Te testen element
  + Kwaliteitseis(en) aan element
    - Stopcriteria gekoppeld aan kwaliteitseis (wanneer wordt het testen afgebroken?)

Te testen elementen:

* Het genereren van vijanden
  + Vijanden worden willekeurig gegenereerd
    - Vijanden worden niet gegenereerd
* Het doodschieten van vijanden
  + Vijanden hun gezondheid wordt afgetrokken
    - De gezondheid wordt niet (juist) afgetrokken
  + Vijanden die dood zijn worden verwijderd
    - Vijanden worden niet verwijderd
    - Vijanden gaan niet dood
* Het aftrekken van de gezondheid van de muur wanneer de vijanden er tegenaan staan
  + De muur telt pas af wanneer er vijanden tegenaan staan
    - De muur telt af voordat er vijanden tegenaan staan
  + De muur stopt bij 0% gezondheid
    - De gezondheid gaat onder 0%
* Het functioneren van de aangepaste cursor (werkt het naar behoren?)
  + De cursor vervangt de normale cursor voor het spel
    - De cursor werkt niet (beweegt niet etc.)

## Documentatie

De documenten die worden opgenomen in dit testplan zijn:

* Het projectplan
  + Dit zal worden geraadpleegd om de must haves vast te stellen en te testen (werken alle functionaliteiten die erin moeten zitten naar behoren?)
* Het technisch ontwerp
  + Dit zal worden geraadpleegd om alle testcases/testscenario’s vast te stellen en uit te werken

Niet opgenomen in dit testplan zal worden:

* Het functioneel ontwerp
  + Zal niet worden opgenomen omdat het technisch ontwerp de benodigde vragen al beantwoordt.

## Testomgeving

De testomgeving is het Unity programma op de laptop van de ontwikkelaar, dit omdat het een kleinschalig product omvat. Klein genoeg waardoor er geen webserver of andere computers/apparaten nodig zijn. Ook zal er gekeken worden of er een appversie van het spel kan worden uitgebracht (via Unity) zodat het zelfs zonder internetverbinding gespeeld kan worden.

## Versiebeheer

Onder dit kopje zal worden uitgelegd waar de verschillende versies van de applicatie zich bevinden en welke versies er bestaan.

* Welke versies bestaan er van de specificaties waarop getest wordt?

Er bestaan officieel 2 versies van alles: een lokale versie en een versie op GitHub. Elke versie kan worden teruggedraaid als dat nodig is, maar het wordt altijd opgeslagen als meest recente (en werkende) versie.

* Welke versies bestaan er van de te testen producten?

Er bestaat van alles momenteel 2 versies: 1 origineel en een kopie van het origineel. De tweede wordt altijd bijgewerkt zodat deze up to date is. Het origineel dient als “rollback”.

* Weke versies bestaan er van de testproducten?

Zie antwoorden hierboven gegeven.

* Waar bevinden zich de verschillende versies?

De lokale (originele) versie bevindt zich op de computer van de ontwikkelaar. De online versie (die constant wordt bijgewerkt) bevindt zich op GitHub zoals hierboven vermeld is.

* Welke versie van het te testen product is getest, en met welk testproduct?

Er wordt getest op de meest recente versie, aan de hand van testscenario’s.

## Planning

De planning voor het testproces is als volgt: de hoofdcomponenten die worden getest zijn het doodmaken van de vijanden en het testen van de gezondheid van de muur (wordt de gezondheid er vanaf gehaald etc.) Ook wordt er rekening gehouden met de samenhang van deze functies, ze zijn allebei essentieel en samen maken ze het product tot wat het moet zijn. Zonder vijanden is het geen defense game, en zonder muur is er geen doel. Er wordt verwacht dat er minimaal 1 hertest per onderdeel komt. Dit zal worden ingecalculeerd en zo snel mogelijk verwerkt worden. Er is geen prioriteit, het ene deel is niet belangrijker dan het andere deel. Samen maken ze het spel, het een is nutteloos zonder het ander.

# Testscenario’s

Hier worden alle testscenario’s opgesomd en beschreven. De tester zal worden benoemd en de scenario’s zullen worden uitgeschreven.

|  |
| --- |
| **Naam te testen onderdeel**  Vijanden doden  **Tester**  **Datum test** |
| **Testscenario**  De tester klikt op de play knop in de Unity omgeving. Hierna worden er kubussen ingeladen die vijanden moeten voorstellen. Klik op deze kubussen en test of ze doodgaan. Groene kubussen moeten in 1 klik doodgaan, gele in 2, rode in 3 en zwarte (tanks) in 10. |
| **Testresultaat** (verbeterpunten, ervaring) |

|  |
| --- |
| **Naam te testen onderdeel**  De kasteelmuur  **Tester**  **Datum test** |
| **Testscenario**  De tester klikt op de play knop in Unity. Laat de kubussen naar de kasteelmuur lopen. Als het goed is, gaat de gezondheid van het kasteel naar beneden totdat deze 0% is. Test ook of de gezondheid op peil blijft wanneer er geen kubussen het kasteel bereiken |
| **Testresultaat** |

|  |
| --- |
| **Naam te testen onderdeel**  Het vizier  **Tester**  **Datum test** |
| **Testscenario**  (Samenhangend met de test voor het doden van vijanden)  De tester gebruikt de aangepaste cursor om de vijanden dood te maken. De cursor is nu een vizier (controleer dit, experimenteer met klikken en het navigeren in de omgeving). Schiet vijanden dood met deze nieuwe cursor en kijk hoe/of dit geregistreerd wordt. |
| **Testresultaat** |