**Tenebris**

**Grupp 24**

**Verifiering och valideringsdokument**

**V. 1.1**

**2018-05-11**

# **Dokumenthistorik**

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Beskrivning | Författare |
| 180411 | 1.0 | Skapandet av dokument | Oscar Rehnberg |
| 180507 | 1.01 | Lagt till fler kravbaserade tester | Anton Persson |
| 180511 | 1.02 | Uppdatering enligt feedback | Oscar Rehnberg |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Innehållsförteckning

[**Dokumenthistorik** 2](#_Toc513806263)

[**Verifiering och valideringsdokument** 2](#_Toc513806264)

[**Syfte** 2](#_Toc513806265)

[**Ordlista** 2](#_Toc513806266)

[**Referenser** 3](#_Toc513806267)

[**Testprocess** 3](#_Toc513806268)

[**Testning** 3](#_Toc513806269)

[**Kravbaserad systemtestning** 3](#_Toc513806270)

[**Testfall kravbaserad systemtestning** 4](#_Toc513806271)

[**Spårningsmatris** 6](#_Toc513806272)

[**Testrapporter** 7](#_Toc513806273)

[**Testrapport kravbaserad systemtest 1** 7](#_Toc513806274)

# **Verifiering och valideringsdokument**

# **Syfte**

Syftet med detta dokument är att förklara hur testprocessen ser ut genom detta projekt. Dokumentet innehåller även resultat från testningen. Testningen är en viktigt del av mjukvaruutveckling som dels ger feedback på ens arbete men även ett annat perspektiv på hur mjukvaran ska utvecklas.

# **Ordlista**

**Kravdokument:** Ett dokument som tidigare skapats i projektets gång.

**Designdokument:** Ett dokument som tidigare skapats i projektets gång.

# **Referenser**

<https://www.tutorialspoint.com/software_testing/software_testing_documentation.htm>

<http://istqbexamcertification.com/what-is-black-box-specification-based-also-known-as-behavioral-testing-techniques/>

<http://istqbexamcertification.com/what-is-equivalence-partitioning-in-software-testing/>

# **Testprocess**

Den testprocess som kommer att användas är kravbaserad systemtestning. Testningen kommer att ske genom att en person startar spelet och provar alla funktioner så att de funkar. Testaren kommer även att använda kravdokumentet och designdokumentet som referens. Eftersom designdokumentet specificerar tydligare hur funktionerna används kommer testaren först markera huruvida kraven uppfylls och sedan huruvida designdokumentet stämmer. Det innebär att systemtestningen kommer att ske två gånger. Den mall vi utgår ifrån [[1]](#footnote-1) använder sig av en plan, ett scenario, case och sedan rapportering. I detta fall är vår plan kravdokumentet och designdokumentet. Testaren dokumenterar vilket scenario, i form av ett ID, i kravdokumentet som testas, t.ex. att få karaktären att hoppa. Nästa steg är att dokumentera vad man förväntar sig ska hända, t.ex. att karaktären kommer att hoppa. Till slut dokumenteras resultatet. Om det uppstår någon slags bugg rapporteras detta i resultatet. Eftersom projektgruppen även strävar efter en bra känsla man spelar är det viktigt att testaren rapporterar om någonting ser fel ut, t.ex. dålig animation. Testningen kommer att ske i slutet av sprint två i förberedelse om att förbättra spelet. Dionist Peci ansvarar för att utföra testningen med hjälp av författaren av detta dokument.

# **Testning**

## **Kravbaserad systemtestning**

Syftet med systemtestningen är att utvärdera om systemets funktioner stämmer överens med kravdokumentet. Man går igenom hela programmet och undersöker alla funktioner. Det som kommer att utföras är black box testing[[2]](#footnote-2). Det innebär att testaren kommer endast spela spelet och kategorisera testfallen efter spelets funktioner. För att hjälpa testaren kommer hen använda designdokument och kravdokument för att veta vilka funktioner som finns. För att förtydliga kommer testaren använda sig ekvivalent partition [[3]](#footnote-3). Det innebär att de olika funktionerna kommer att grupperas om de är lika till beteendet, t.ex. att förflytta sig till höger eller vänster. Detta är för att koden är nästintill lika.

**Prioritering**

Prioriteringen kommer att följa kravdokumentets prioritering.

# **Testfall kravbaserad systemtestning**

**1a Plattformar 1**

Krav: Skapandet av plattformar vilket innebär en textfil som specificerar placering samt fungerande kollisionshantering med alla karaktärer som spelet innehar - alltså visuellt sätt ska karaktären inte ha något problem med kollisionen.

Testfall id: 1

Testfall namn: Kollision över

Initial status: Spelet startat; inom platform mode.

Data setup: Textfilen har platformen floor markerat som “f”.

Input: Antar att användaren landar uppe en platform.

Förväntad Output: Karaktären faller inte igenom dvs stannar på platformen.

Faktisk Output:

**2a Karaktär 2**

Krav: En karaktär som spelaren styr genom att gå på de olika plattformarna. Karaktären ska styras med hjälp av “WAD” eller piltangenterna på ett tangentbord - alltså karaktären ska visuellt sätt röra sig efter användarens input. Karaktären ska även ha fungerande kollisionshantering.

Testfall id: 2

Testfall namn: Styra karaktär med “WAD”.

Initial status: Spelet start, inom platform mode.

Data setup: Textfilen har platformen floor markerat som “f”.

Input: Att användaren trycker på WAD.

Förväntad Output: W = Karaktären hoppar, A = Karaktären går till vänster, D = Karaktären går till höger.

Faktisk Output:

Testfall id: 3

Testfall namn: Styra karaktär med piltangenterna.

Initial status: Spelet startat, inom platform mode.

Data setup: Textfilen har platformen floor markerat som “f”.

Input: Att användaren trycker på piltangenterna.

Förväntad Output: Uppåtpilen = Karaktären hoppar, Vänsterpilen = Karaktären går till vänster, Högerpilen = Karaktären går till höger.

Faktisk Output:

**3a Kamera 3**

Krav: En kamera som förflyttas efter karaktären.

Testfall id: 4

Testfall namn: Kameraförflyttning platform.

Initial status: Spelet startat, inom platform mode.

Data setup: Textfilen har platformen floor markerat som “f”.

Input: Att användaren trycker på piltangenterna för att flytta sig till höger, och vänster.

Förväntad Output: Kameran följer.

Faktisk Output:

Testfall id: 5

Testfall namn: Kameraförflyttning combat.

Initial status: Spelet startat, kolliderat med monster, inom combat mode..

Data setup: Textfilen har platformen floor markerat som “f”, och “g”.

Input: Att användaren trycker på piltangenterna för att flytta sig till höger, eller vänster, och kolliderar med monster.

Förväntad Output: Kameran zoomar in.

Faktisk Output:

**4b Hoppa 4**

Krav: Spelaren ska kunna hoppa mellan olika plattformar och därför behöver vi även implementera gravitation.

Testfall id: 6

Testfall namn: Gravitation.

Initial status: Spelet startat, inom platform mode.

Data setup: Textfilen har platformen floor markerat som “f”.

Input: Att användaren trycker på piltangenterna för att flytta sig till höger, eller vänster, och kommer utanför platformen. Alternativt att hoppa och komma utanför platformen.

Förväntad Output: Karaktären faller.

Faktisk Output:

**3c Skills 3**

Krav: Spelare ska ha en möjlighet att använda skills. Detta innebär en implementation av cooldowns eller mana.

Testfall id: 7

Testfall namn: Mana

Initial status: Spelet startat, kolliderat med monster, inom combat mode.

Data setup: Textfilen har platformen floor markerat som “f”, och “g”.

Input: Att användaren trycker på piltangenterna för att flytta sig till höger, eller vänster, och kolliderar med monster. Därefter väljer användaren att använda en spell.

Förväntad Output: Mana reduceras.

Faktisk Output:

Testfall id: 8

Testfall namn: Mana

Initial status: Spelet startat, kolliderat med monster, inom combat mode.

Data setup: Textfilen har platformen floor markerat som “f”, och “g”.

Input: Att användaren trycker på piltangenterna för att flytta sig till höger, eller vänster, och kolliderar med monster. Därefter väljer användaren att använda en spell.

Förväntad Output: Mana reduceras.

Faktisk Output:

**5c Fällor 5**

Krav: Nivåerna ska ha fällor utplacerade för att skapa hinder som spelaren måste navigera runt.

Testfall id: 8

Testfall namn: Spike trap

Initial status: Spelet startat, inom platform mode.

Data setup: Textfilen har platformen floor markerat som “f”.

Input: Att användaren trycker på piltangenterna för att flytta sig över ett spike trap.

Förväntad Output: Game Over skärm visas.

Faktisk Output:

**1c Items 1**

Krav: Spelaren ska kunna plocka items för att göra sin karaktär starkare.

Testfall id: 9

Testfall namn: Rune Of Wind

Initial status: Spelet startat, inom platform mode.

Data setup: Textfilen har platformen floor markerat som “f”, och chest som “c”.

Input: Att användaren trycker på piltangenterna för att flytta sig över ett chest.

Förväntad Output: Karaktären springer snabbare.

Faktisk Output:

## **Spårningsmatris**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 2a | 3a | 4b | <id krav> |  |
| 1 | X | X | X |  |  |
| <id testfall> |  |  |  |  |  |
| <id testfall> |  |  |  |  |  |
| <id testfall> |  |  |  |  |  |
| <id testfall> |  |  |  |  |  |

# **Testrapporter**

## **Testrapport kravbaserad systemtest 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id testfall** | **datum** | **Kod/dokument version** | **Utfört av** | **Resultat** |
| 1 | 180411 | 1.0 | Dionist Peci | Förväntat |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. <https://www.tutorialspoint.com/software_testing/software_testing_documentation.htm> 2018-04-11 [↑](#footnote-ref-1)
2. <http://istqbexamcertification.com/what-is-black-box-specification-based-also-known-as-behavioral-testing-techniques/> 2018-04-11 [↑](#footnote-ref-2)
3. <http://istqbexamcertification.com/what-is-equivalence-partitioning-in-software-testing/> 2018-04-11 [↑](#footnote-ref-3)