

2015

# Laith

Verifierings- och Valideringsdokument v1.0  
2015-05-23

Grupp 26

Andreas Karlsson  
Simon Johansson  
Henrik Phan  
Simon Bothén  
Tim Lindstam

Malmö Högskola

# Revisionshistorik

Datum	Namn	Aktivitet	Version
2015-03-30	Tim L.	Första draft	0.1
2015-05-12	Henrik P.	Fyllde i testprocess	0.2
2015-05-15	Henrik P.	Lade till introduktion och testfall.	0.3
2015-05-22	Tim L.	Skrev testrapporter samt kravmatris	0.4
2015-05-22	Simon J.	Rensning av text samt korrekturläsning	0.5
2015-05-22	Tim L.	Korrekturläsning	0.6
2015-05-23	Andreas K.	Färdigställde dokument med försättsblad, innehållsförteckning och revisionshistorik.	1.0

# Innehållsförteckning

Revisionshistorik.....	1
Introduktion .....	3
Syfte.....	3
Omfattning .....	3
Ordlista .....	3
Projektets testprocess.....	4
Testprocessen.....	4
Testmetoder .....	5
Integrationstestning .....	5
Enhetstestning.....	5
Betatestning .....	5
Kravbaserad Testning .....	5
Testfall .....	6
ID: 1 .....	6
ID: 2 .....	6
ID: 3 .....	6
ID: 4 .....	6
ID: 5 .....	7
ID: 6 .....	7
ID: 7 .....	7
ID: 8 .....	8
ID: 9 .....	8
ID: 10 .....	8
Testrapportering .....	9
ID: 1 .....	9
ID: 2 .....	9
ID: 3 .....	9
ID: 4 .....	9
ID: 5 .....	9
ID: 6 .....	9
ID: 7 .....	10
ID: 8 .....	10
ID: 9 .....	10
ID: 10 .....	10
Spårningsmatris.....	11

# Introduktion

## Syfte

Syftet med detta dokument är att beskriva testprocessen och testmetoderna. Den används för att få en strukturerad testningsprocess och för att dokumentera de olika testerna.

## Omfattning

Den beskriver även hur problem ska hanteras när de upptäcks genom testerna. Den innehåller information om de olika testfallen som är knutna till kraven i kravspecifikationen.

## Ordlista

Se bifogat dokument: "DA336A\_Grupp\_26\_Laith\_Ordlista"

# Projektets testprocess

## Testprocessen

Själva testen i projektet kommer att ske i flera omgångar och i olika delar av projektet. Så fort som möjligt kommer vi att börja göra interna tester. Vid skrivandet av ny kod ansvarar varje utvecklare själv för att köra egna enhetstester för att se till att varje del av koden fungerar individuellt. Testerna sker omgående då utvecklaren anser att en körbar del av koden har blivit färdig genomförs och det är upp till utvecklaren att bestämma hur dessa tester ser ut. Om ett problem uppstår i testen ska utvecklaren själv försöka lösa problemet om möjligt. Mindre problem kan lösas enkelt av utvecklaren och behöver inte dokumenteras eller rapporteras. Gäller det större buggar eller problem rapporteras detta till teamet på en gemensam bugglista så att andra kan komma med lösningar till problemet. Om buggen kvarstår tills nästa möte görs en bedömning på hur allvarlig buggen är och hur hög prioritet som borde sättas på att fixa buggen.

I slutet av varje sprint genomförs ett större kravbaserat systemtest där hela teamet är närvarande. Här testas hela alla projektets komponenter tillsammans. I dessa test görs en genomgång av vissa testfall för att se om ett krav är uppfyllt. Testet genomförs sedan genom att en eller två personer i teamet startar upp spelet och går igenom de relevanta testfallen. Teamet bedömer gemensamt om ett visst krav har uppfyllts eller inte. Testtillfället ska dokumenteras i en testrapport. Kraven som vi har måste uppfyllas med projektet för att projektet skall kunna ses som en succé.

Under de någon av de två sista sprinterna ska betatestning ske för att kunna testa de krav som inte kan täckas av testfallen. Här handlar det alltså om användbarhetstester för att se hur nya användare navigerar spelet. Testningen rapporteras och sammanställs för att sedan avgöra om eventuella ändringar till spelet.

# Testmetoder

Vi kommer att använda oss av en rad olika metoder för att få ut så mycket av testen som möjligt. Vi kommer bland annat att använda oss utav:

## Integrationstestning

Integrationstestning där vi prövar hur pass spelet funkar i sin helhet kommer att vara extremt viktigt för oss som utvecklar spel därför att ett spel måste fungera 100% när alla delarna sitter ihop, annars finns det en stor chans att hela spelet rasar samman.

Som ett exempel, säg att hoppfunktion hos karaktärerna funkar perfekt när i kör det kontrollerat med primitiva drivare, men sen när vi kopplar ihop alla delarna tillsammans märker vi att hoppfunktionen inte längre fungerar, då kommer en stor del av hela spelet att brista samman eftersom att vårt spel har stora delar där spelet inte går att klara av om man inte kan hoppa. Nu är detta ett mycket extremt exempel, men det förklarar varför vi måste se till att fungerar ihop.

## Enhetstestning

Vi kommer att göra enhetstestningar naturligt under vårt projekt. Ett unity-spel är uppbyggt av ett samspel mellan olika scener och scriptfiler som måste kunna fungera tillsammans. Därför kommer enhetstester att ske naturligt medans arbetet fortsker.

## Betatestning

Vi kommer att använda oss utav betatestning under den fjärde sprinten. När vi gör betatestningen är det tänkt att spelet kommer att vara nära att vara en färdig produkt, och denna testningen kommer därför att användas för att få en potentiell slutanvändares åsikter om spelet, vad som bör ändras för att spelet ska blir roligare. Detta ger oss även en bild om hur en ny användare upplever spelet. Det kan handla om att en viss mekanik är förvirrande eller att någonting är otydligt eller inte syns. Vi kan då anpassa spelet för att göra det lättare för nya spelare att förstå.

Fördelen med betatestningen är att få nya synvinkar på hur användaren kommer att spela spelet samt hitta buggar som vi som utvecklare inte hade tänkt på. Därför är det viktigt att ha stora testgrupper av användare som spelar spelet vid slutet av utvecklingen som hjälper utvecklarna med att hitta buggar.

## Kravbaserad Testning

Vi kommer att göra kravbaserad testning i slutet av varje sprint för att kontrollera så att projektet klarar av våra krav för att spelet ska bli så komplett som möjligt.

# Testfall

## ID: 1

Namn: Skapa rum

Förberedelser: Uppkoppling till nätet.

Teststeg:

1. Spelaren väljer level som ska spelas.
2. Spelaren väljer karaktär.
3. Spelaren klickar "Create room"

Förväntat resultat: Spelaren skapar ett nytt rum med den valda karaktären(F-M-K-2) och leveln(F-M-K-1).

## ID: 2

Namn: Joina rum

Förberedelser: Uppkoppling till nätet. Minst två andra klienter har redan skapat rum.

Teststeg:

1. Spelaren klickar "Join room #0".

Förväntat resultat: Spelaren kommer in i rum #0 på samma level med en annan karaktär än hosten. (F-LS-K-2).

## ID: 3

Namn: Förflyttning av karaktär

Förberedelser: Uppkoppling till nätet. Spelaren är i ett rum.

Teststeg:

1. Spelaren använder "A,D,Spacebar" på tangentbordet.

Förväntat resultat: Karaktären rör sig höger när spelaren trycker "D", vänster med "A" och hoppar med "Spacebar" (F-GP-K-3). Kameran följer spelaren(F-K-K-1). Kameran ska inte roteras(F-K-K-2).

## ID: 4

Namn: Grappling hook

Förberedelser: Uppkoppling till nätet. Spelaren är i ett rum som Narissa. Det finns ett objekt inom räckvidd som Narissa kan använda "Hook" på.

Teststeg:

1. Spelaren högerklickar på hookable-objektet.
2. Spelaren använder "W,A,S,D".
3. Spelaren trycker på "Spacebar".

Förväntat resultat:

1. Narissa skjuter iväg en äntherhake mot muspekaren(F-GP-K-4) som fastnar på objektet. (F-GP-K-5)
2. "A,D" gör att Narissa svingar sig höger eller vänster, "W,S" gör att äntherhaken blir längre eller kortare. (F-GP-K-5)
3. Äntherhaken förstörs och Narissa faller ner. (F-GP-K-5)

## ID: 5

Namn: Shield

Förberedelser: Uppkoppling till nätet. Spelaren är i ett rum som Gareth. Det finns en fiende som skjuter projektiler.

Teststeg:

1. Spelaren rör sig mot fienden.
2. Spelaren trycker och håller in högerklick på fienden.
3. Spelaren rör runt muspekaren på skärmen.

Förväntat resultat: Gareth tar upp sin sköld och riktar den mot muspekaren(F-GP-K-4). När fiendeprojektilen träffar Gareths sköld, reflekteras den i riktning mot muspekaren(F-GP-K6). När skölden är aktiverad, rör sig Gareth långsammare(F-GP-K6).

## ID: 6

Namn: Charge

Förberedelser: Uppkoppling till nätet. Spelaren är i ett rum som Gareth. Det finns en fiende. Det finns en förstörbar vägg.

Teststeg:

1. Spelaren ställer sig framför fienden och trycker "Shift".
2. När Gareth slutar sin "Charge" trycker spelaren omedelbart på "Shift" igen.
3. Spelaren väntar en stund.
4. Spelaren ställer sig framför väggen och trycker "Shift".

Förväntat resultat:

1. Gareth börjar en "Charge" mot fienden, skadar den och knuffar bakåt den när han kolliderar(F-GP-K-7).
2. Förmågan är på cooldown och aktiveras inte(F-GP-K-9).
3. Förmågan går ur cooldown.
4. Gareth chargar mot väggen och förstör den(F-GP-K-8).

## ID: 7

Namn: Swap

Förberedelser: Uppkoppling till nätet. Spelaren är i ett rum som Narissa. Det finns en annan spelare i rummet som Gareth.

Teststeg:

1. Spelaren rör sig åt höger och trycker sedan "Q".
2. Spelaren trycker omedelbart "Q" efteråt.
3. Spelaren väntar en stund.
4. Spelaren trycker "Q" igen.

Förväntat resultat: Narissa och Gareth byter plats och hastighet(F-GP-K-10). Förmågan aktiveras inte igen förrän cooldown-perioden har gått(F-GP-K-9).



## ID: 8

Namn: Narissas pilbåge och fiendens hälsomätare

Förberedelser: Uppkoppling till nätet. Spelaren är i ett rum som Narissa. Det finns en fiende. Det finns en annan spelare i rummet som Gareth mellan Narissa och fienden.

Teststeg:

1. Spelaren håller in vänsterklick.
2. Spelaren släpper vänsterklick ovanför fienden.
3. Spelaren siktar på fienden och skjuter den tills fienden dör.

Förväntat resultat:

1. Narissa börjar synligt ladda upp pilen medan vänsterklick hålls ner(F-UI-K-7).
2. När vänsterklick släpps, avfyrar Narissa en pil mot muspekaren. Pilens fart och skada beror på hur laddad pilen var. (F-GP-K-11) Pilen går igenom Gareth utan att kollidera eller skada honom(F-GP-K-1).
3. Fienden tar skada när den blir träffad av pilen. Fiendens hälsomätare blir synlig. Fienden dör och försvinner när dens hälsa är uttömd.(F-GP-K-12)

## ID: 9

Namn: Gareth svärd

Förberedelser: Uppkoppling till nätet. Spelaren är i ett rum som Gareth. Det finns en fiende.

Teststeg:

1. Spelaren ställer sig framför fienden och trycker vänsterklick.
2. Spelaren flyttar musen till andra sidan om Gareth och trycker vänsterklick.

Förväntat resultat:

1. Gareth svingar sitt svärd mot fienden. Fienden tar skada.(F-GP-K-13)
2. Gareth svingar sitt svärd i riktning bort från fienden.(F-GP-K-13)

## ID: 10

Namn: Hälsomätare och respawn

Förberedelser: Uppkoppling till nätet. Spelaren är i ett rum tillsammans med en annan spelare.

Karaktär spelar ingen roll. Det finns två fiender som kan attackera varsin spelare. Det finns en checkpoint förutom startcheckpointen.

Teststeg:

1. Spelarna passerar genom en checkpoint som inte är startcheckpointen.
2. Spelaren placeras så att varje spelare blir attackerad av en fiende.
3. Spelarna tar skada och blir dödade av fienderna.

Förväntat resultat: Spelarens hälsomätare går ner när dess karaktär blir träffad av en fiende (F-UI-K-4). Detsamma gäller för den andra spelarens hälsomätare(F-UI-K-5). När en spelare dör, återuppstår denna på den senaste checkpointen(F-GP-K-2)(F-E-K-3).

# Testrapportering

## ID: 1

Datum för testningen: 2015-05-22

Vem som utfört testerna: Tim Lindstam

På vilken kodversion testerna utförts: 0.9b

Vilka testfall som genomförts: ID: 1, Skapa Rum

Resultat av genomförandet av testfallen: Resultaten blev som förväntat.

## ID: 2

Datum för testningen: 2015-05-22

Vem som utfört testerna: Tim Lindstam

På vilken kodversion testerna utförts: 0.9b

Vilka testfall som genomförts: ID: 2, Joina rum

Resultat av genomförandet av testfallen: Resultaten blev som förväntat.

## ID: 3

Datum för testningen: 2015-05-22

Vem som utfört testerna: Tim Lindstam

På vilken kodversion testerna utförts: 0.9b

Vilka testfall som genomförts: ID: 3, Förflyttning av karaktär

Resultat av genomförandet av testfallen: Resultaten blev som förväntat.

## ID: 4

Datum för testningen: 2015-05-22

Vem som utfört testerna: Tim Lindstam

På vilken kodversion testerna utförts: 0.9b

Vilka testfall som genomförts: ID: 4, Grappling hook

Resultat av genomförandet av testfallen: Resultaten blev som förväntat.

## ID: 5

Datum för testningen: 2015-05-22

Vem som utfört testerna: Tim Lindstam

På vilken kodversion testerna utförts: 0.9b

Vilka testfall som genomförts: ID: 5, Shield

Resultat av genomförandet av testfallen: Resultaten blev som förväntat.

## ID: 6

Datum för testningen: 2015-05-22

Vem som utfört testerna: Tim Lindstam

På vilken kodversion testerna utförts: 0.9b

Vilka testfall som genomförts: ID: 6, Charge

Resultat av genomförandet av testfallen: Resultaten blev som förväntat.

## ID: 7

Datum för testningen: 2015-05-22

Vem som utfört testerna: Tim Lindstam

På vilken kodversion testerna utförts: 0.9b

Vilka testfall som genomförts: ID: 7, Swap

Resultat av genomförandet av testfallen: Resultaten blev som förväntat.

## ID: 8

Datum för testningen: 2015-05-22

Vem som utfört testerna: Tim Lindstam

På vilken kodversion testerna utförts: 0.9b

Vilka testfall som genomförts: ID: 8, Narissas pilbåge och fiendens hälsomätare

Resultat av genomförandet av testfallen: Resultaten blev som förväntat.

## ID: 9

Datum för testningen: 2015-05-22

Vem som utfört testerna: Tim Lindstam

På vilken kodversion testerna utförts: 0.9b

Vilka testfall som genomförts: ID: 9, Gareth svärd

Resultat av genomförandet av testfallen: Resultaten blev som förväntat.

## ID: 10

Datum för testningen: 2015-05-22

Vem som utfört testerna: Tim Lindstam

På vilken kodversion testerna utförts: 0.9b

Vilka testfall som genomförts: ID: 10, Hälsomätare och respawn

Resultat av genomförandet av testfallen: Resultaten blev som förväntat.

## Spårningsmatris

Testfall	Krav
1	F-M-K-1 F-M-K-2
2	F-LS-K-2
3	F-K-K-1 F-K-K-2 F-GP-K-3
4	<u>F-GP-K-4</u> F-GP-K-5
5	<u>F-GP-K-4</u> F-GP-K-6
6	F-GP-K-7 F-GP-K-8 <u>F-GP-K-9</u>
7	<u>F-GP-K-9</u> F-GP-K-10
8	F-GP-K-1 F-GP-K-11 F-GP-K-12 F-UI-K-7
9	F-GP-K-13
10	F-GP-K-2 F-UI-K-4 F-UI-K-5 F-E-K-3