

Test och validerings dokument

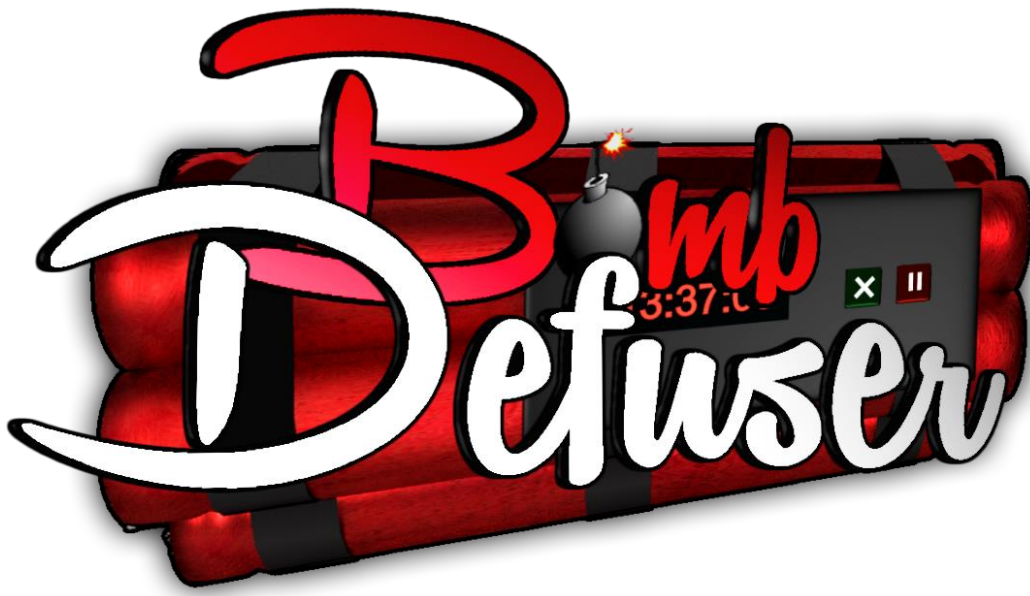
# Bomb Defuser

version: 1.0.1

2015-05-20

Handledare: Edward S. Blurock

Grupp 18



## Revision History

Date	Version	Description	Author
2015-04-16	0.0.0	Skapa dokument.	Patrik Nilsson
2015-04-16	0.1.0	Klistra in grunder.	Patrik Nilsson
2015-05-20	0.1.1	Skrivit testprocess och kategorisering.	Patrik Nilsson
2015-05-28	0.1.2	Strukturfixering, samt lagt till testfall.	Patrik Nilsson
2015-05-04	0.2.0	Lagt till testmatris.	Patrik Nilsson
2015-05-04	0.2.1	Lagt till testfall för flertal kategorier.	Patrik Nilsson
2015-05-04	0.2.2	Lagt till testfall för världen.	Patrik Nilsson
2015-05-04	0.2.3	Ai och Taser testfall.	Patrik Nilsson
2015-05-11	0.3.0	Lagt till testfall för bomben.	Patrik Nilsson
2015-05-12	0.3.1	Lagt till testfall för level editorn.	Patrik Nilsson
2015-05-17	0.3.2	Finslip på hela dokumentet.	Patrik Nilsson
2015-05-18	1.0.0	Finslip och release av fullständigt dokument.	Patrik Nilsson, Mika Lehtinen
2015-05-20	1.0.1	Lagt till text	Patrik Nilsson, Robin Andersson
2015-05-24	1.0.1	Korrekturläst och rättat	Peter Lindberg

## Innehållsförteckning

# 1 Inledning

1.1	Syfte .....	5
1.2	Omfattning .....	5
1.3	Definitioner, akronmyer och förkortningar .....	5
1.4	Referenser .....	6
1.5	Struktur .....	6

## 2 Testprocess .....

... 7

## 3 Kategorisering och avgränsning .....

... 7

3.1	Unit test .....	8
3.2	Component test .....	8
3.2	Acceptanstest .....	8
3.2	Speltest .....	8

## 4 Testfall

4.1	Prototyp .....	9
4.2	Spel .....	10

4.2.1	System .....	10
4.2.2	Navigering .....	11
4.2.3	Entitet .....	12
4.2.4	Världen .....	13
4.2.5	Tiles .....	14
4.2.6	AI .....	15
4.2.7	Taser .....	16
4.2.8	Bomb .....	16
4.3	Level editor .....	17
<b>5</b>	<b>Testmatris</b> .....	
..		19

# 1 Inledning

## 1.1 Syfte

Syftet med detta dokument är att visa vilka tester som utförs på projektet. Dessa tester skall även testa de krav som finns i kravanalysen. Syftet är dessutom att få fram vad som är viktigast att testa, samt testa om det finns buggar. Allt detta skall rapporteras i detta dokument.

## 1.2 Omfattning

Omfattningen av detta dokument kommer bli tester på det som vi kan testa. Vilket kommer bli att spelet fungerar på de senaste versionerna av android. Vi kommer inte utföra test på äldre versioner. Det kommer framförallt vara test och validering på de krav som finns i kravdokumentet. Vi kommer även testa de bitar som inte kan dokumenteras och det är helhetsbilden av spelet.

## 1.3 Definitioner, akronmyr och förkortningar

**C#, Java:** Två varianter av språk man kan skriva kod i, används ofta för att utveckla system och spel.

**XNA, LibGDX:** Två väldigt snarlika ramverk till språken C# och Java som ger användaren tillgång till förbestämda funktioner istället för att du ska behöva skriva all kod manuellt.

**Cutscene:** En kort filmsekvens som ofta används som en konstpaus mellan två scener i ett spel. Det används även för att tala om för spelare/åskådare vad som händer vid ett visst ögonblick.

**Level-Editor:** Ett system utvecklat med syftet att förenkla skapandet av nivåer till ett angivet spel, utvecklas ofta för att öka effektiviteten i utvecklingen.

**Taser:** Conducted electrical weapon (CEW) är ett pistolliknande vapen som skickar iväg väldigt starka elchocker genom två pil-liknande elektroder.

**Tiles:** Ett system av rutor som vi bygger upp spelet med, ofta om man talar om "tile-based" games så är spelvärlden uppbyggd i rutor av förbestämd storlek.

**AI:** (Artificial Intelligence) Är i korta drag intelligensen som uppvisas av en maskin eller mjukvara.

**Android:** Ett populärt operativsystem för smarttelefoner, detta är systemet vi utvecklar spelet till.

**Entiteter:** översätts till enheter, ett samlingsord för dem enheter vi använder oss av i spelet, till exempel bomben eller fiender.

## 1.4 Referenser

Vi har använt oss av den vägledande dokumentation som funnits tillgänglig för oss studenter som till exempel mallar och exempel på tidigare studenters arbete. Vi har också använt oss av dem PDF-er vi fått tillgång till genom föreläsningarna genom kursens gång. Och slutligen så har vi använt oss väldigt flitigt av Roger Pressman's *Software Engineering, a practitioner's approach 8th edition*.

Programspråk, ramverk och annan mjukvara som varit av centralt värde för utvecklingsprocessen är utvecklade och tillhörande följande:

- [www.java.com](http://www.java.com)
- [www.blender.org](http://www.blender.org)
- [www.microsoft.com](http://www.microsoft.com)

## 1.5 Struktur

Strukturen i dokumentet innefattas hur testprocessen skall gå till, samt de olika typer av test som kommer utföras. Testfallen kategoriseras efter vilka krav de täcker i kravanalysen. Slutligen finns det en testmatris som visar vilka testfall som knyter samman med kraven i kravanalysen.

## 2 Testprocess

Målet med testningen är att testa om produkten fungerar som tänkt. Detta testar vi genom unit test, komponenttestning samt krav och acceptanstest. Det är viktigt att produkten uppfyller de krav som finns för att få ett fullständigt och fungerande spel. Eftersom att det är vi som både utvecklar och gör acceptanstest på produkten, kan vi genom processens gång utföra acceptanstest och ändra kraven efter produktens utformning. Att sätta upp krav tidigt kan resultera i en dålig produkt, då bättre krav och ideer kan komma längst processens väg. Dock får man inte avvika för mycket från grundidén.

Testfall kommer utföras under hela processen och kommer främst fokusera på att uppfylla de krav som finns. Unit-test och komponenttest kommer utföras under utvecklingen. Där unit-test testar den bit kod eller del av produkten som man arbetar med. När den är klar gör man ett unit-test för att se om den fungerar. Därefter görs en komponenttest för att se hur de fungerar med de andra komponenterna i produkten. Eftersom produkten är ett spel är det viktigt att man speltestar produkten för att få en helhetsbild. Ett spel kan ses som konst, vilket inte alltid gör det lätt att göra logiska tester.

## 3 Kategorisering och avgränsning

Testerna som görs kommer vara direkt anknytna till ett krav. Anknytningen mellan krav och test kommer visas i en testmatris. Test-id kommer genereras utefter vad det är för typ av test samt vad den testar.

***Exempel:***

TC-U-V1 vilken är **T**est case, **u**nit test, **v**ärlden och att det är test **1**.

TC-C-FE1 vilken är **T**est case, **c**omponent test, **h**ur **f**läktar och **e**ntiteter funkar, samt test

**1.**

Det kommer bli ett lätt och smidigt sätt att se om det är flera komponenter som testas eller om det bara är ett.

### **3.1 Unit-test**

Unit-test är ett test som kommer utföras på en komponent i produkten. Detta kan vara att entiter skall röra sig på rätt sätt och att animationer på objekt fungerar och liknande. Ett unit-test kan vara kopplat till ett eller flera krav i kravdokumentet.

### **3.2 Component test**

Ett component test är ett test mellan flera komponenter. Detta kan vara hur entiter agerar mot andra entiter och hur entiter fungerar med kollisionslagret i världen. Det är viktigt att component test görs eftersom det oftast här man hittar buggar i spelet då olika system skall påverka varandra, vilket kan göras på flera olika sätt. Alla scenarion kommer inte att testas. Men de som är mest sannolika att inträffa efter erfarenheter i andra produkter. Component test kommer vara direkt kopplat till en eller flera krav.

### **3.3 Acceptance test**

Detta kommer vara saker som inte går under unit/component.test men ändå är kopplat till ett krav som måste testas och accepteras.

### **3.4 Speltest**

Är produkten vi producerat en hit på marknaden? Detta testas i speltet som är kopplat till ett abstrakt krav. Det man vill testa är om den konst som vi har skapat påverkar intressenterna och användarna på ett positivt sätt som uppfyller de abstrakta kraven.



## 4 Testfall

### 4.1 Prototyp

Prototypen är de viktigaste mekanikerna i spelet. Dessa vill vi testa innan vi börjar på spelet. Därför sätter vi ett antal testfall. Då dessa mekaniker måste testas och fungera felfritt.

<b>ID</b>	TC-C-PT1	
<b>Rubrik</b>	Tiles och spelare kollision testas mellan varandra.	
<b>Förberedelse r</b>	Skapa ett antal tiles på olika lager och placera dem i värden samt skapa en spelare.	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Flytta spelaren i olika riktningar		1. Spelaren skall inte kunna gå igenom tilesen på "collision lagret" 2. Spelaren skall kunna passera tiles på övre och under lager.

<b>ID</b>	TC-C-PT2	
<b>Rubrik</b>	Spelaren skall kunna hoppa på tiles	
<b>Förberedelse r</b>	Skapa ett antal tiles och placera dem i värden samt skapa en spelare.	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Flytta spelaren i olika riktningar och hoppa från olika tiles.		1. Spelaren skall endast kunna hoppa när hen är på en tile.

<b>ID</b>	TC-C-PCP1	
<b>Rubrik</b>	Kameran skall förflytta sig smidigt efter spelaren	
<b>Förberedelse</b>	Skapa en spelare i världen.	

<b>r</b>		
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Flytta spelaren i olika riktningar 2. Hoppa runt		1. Testaren för avgöra om kameran rör sig smidigt efter spelaren som är behagligt för användaren.

<b>ID</b>	TC-C-PFP1	
<b>Rubrik</b>	Fläktar skall påverka spelaren	
<b>Förberedelse r</b>	Skapa fläktar i olika riktningar, samt skapa en spelare i världen med tiles.	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Förflytta gubben på olika riktningar in i olika fläcktar. Som är riktade i olika riktningar. 2. Hoppa i fläktar.		1. Spelaren skall få en kraft som göra att spelaren "blåser" iväg. 2. Kan blåsas uppåt och "flyga" när fläkten är igång.

<b>ID</b>	TC-C-PTE1	
<b>Rubrik</b>	Fiender skall kunna gå fram och tillbaka på tiles	
<b>Förberedelse r</b>	Skapa tiles och fiender.	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Studera fiendens förflyttningar		1. Fienden skall gå fram och tillbaka. 2. Fienden skall inte trilla av tiles.

## 4.2 Spel

### 4.2.1 System

System-testfallen är de testfall som skall testa om spelet och leveleditorn fungerar på de system som de skall köras på. Spelet skall köras på Android, därför måste vi testa det på olika Android-versioner.

Leveleditorn skall fungera på windows med ett antal framework installerade.

<b>ID</b>	TC-C-SA1	
<b>Rubrik</b>	Test på Android	
<b>Förberedelse r</b>	Exportera spelet till Android.	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Starta spelet och gör en speltest för att se om spelet funkar som på datorn.		1. Spelet fungerar likvärt som på datorn.

<b>ID</b>	TC-U-S2	
<b>Rubrik</b>	Spara data.	
<b>Förberedelse r</b>	Exportera spelet till Android.	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Gör en progress genom levelarna i spelet. 2. Stäng ner spelet. 3. Starta spelet igen.		1. Samma progress skall finnas när man startar spelet igen.

## 4.2.2 Navigering

Navigering är hur man tar sig mellan olika menyer. Detta provas med hjälp av knappar. Det är viktigt att detta funkar för att kunna starta olika nivåer.

<b>ID</b>	TC-U-N1	
<b>Rubrik</b>	Level meny	
<b>Förberedelse r</b>	Starta spelet och stanna på huvodemenyn	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Tryck på "play". 2. Välj en level. 3. Repetera för att prova olika levelar.		1. Tryck på play skall visa upp flera knappar. 2. Dessa knappar tar dig sen till olika

	levlar.
--	---------

<b>ID</b>	TC-U-N2	
<b>Rubrik</b>	Level menyn	
<b>Förberedelse r</b>	Starta spelet och navigera till level menyn	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Tryck på en level som är låst.		1. Level skall inte laddas. 2. Man stannar kvar i level-menyn

<b>ID</b>	TC-U-N3	
<b>Rubrik</b>	Level-menyn	
<b>Förberedelse r</b>	Starta spelet och navigera till level-menyn	
<b>Teststeg</b>		<b>Förväntat resultat</b>
2. Tryck på en level som är låst.		3. Level skall inte laddas. 4. Man stannar kvar i level-menyn

### 4.2.3 Entitet

Entiteter är levande objekt som finns i världen. Dessa påverkas av krafter, exempelvis gravitationen och blåst från fläktar. Alla entiteter kan elimineras, samt förflytta sig i världen. I dessa testfallen provar vi om allt detta fungerar.

<b>ID</b>	TC-C-EVII	
<b>Rubrik</b>	Flytta entiteter i världen	
<b>Förberedelse r</b>	Skapa tiles, entitet, samt koppla kontroller till entiteten	

Teststeg	Förväntat resultat
<ol style="list-style-type: none"> <li>1. Använd kontrollerna för att förflytta entiten.</li> <li>2. Gå in i tiles, hoppa flytta frit runt i världen</li> </ol>	<ol style="list-style-type: none"> <li>1. Användaren skall kunna hoppa och röra entitets sidleds</li> <li>2. Entiteten skall endast kunna hoppa när den är på marken/står på en tile</li> <li>3. Entiteten skall påverkas av gravitation</li> <li>4. Entiteten skall endast kollidera med tiles på kollisionslagret.</li> </ol>

<b>ID</b>	TC-C-EB2	
<b>Rubrik</b>	Action knappen	
<b>Förberedelse r</b>	Skapa en bomb i världen.	
<b>Teststeg</b>		<b>Förväntat resultat</b>
<ol style="list-style-type: none"> <li>1. Gå fram till bomben</li> <li>2. Använd "action" knappen</li> </ol>		<ol style="list-style-type: none"> <li>1. Om entiteten står på bomben skall en ny scene visas</li> </ol>

<b>ID</b>	TC-C-EE3	
<b>Rubrik</b>	Döda Entitet	
<b>Förberedelse r</b>	Skapa fiender och en spelare (entitet)	
<b>Teststeg</b>		<b>Förväntat resultat</b>
<ol style="list-style-type: none"> <li>1. Låt fienden döda dig</li> </ol>		<ol style="list-style-type: none"> <li>1. Spelaren tar skada, syns på en mätare.</li> <li>2. Spelaren dör.</li> <li>3. Gameover screen kommer upp.</li> </ol>

<b>ID</b>	TC-C-EF4	
<b>Rubrik</b>	Entiteter och yttre krafter	
<b>Förberedelse r</b>	Skapa fläktar i olika riktningar, samt skapa en spelare i världen med tiles.	

Teststeg	Förväntat resultat
<ol style="list-style-type: none"> <li>1. Förflytta gubben på olika riktningar in i olika fläcktar. Som är riktade i olika riktningar.</li> <li>2. Hoppa i fläcktar.</li> </ol>	<ol style="list-style-type: none"> <li>1. Spelaren skall få en kraft som göra att spelaren "blåser" iväg.</li> <li>2. Kan blåsas uppåt och "flyga" när fläkten är igång.</li> </ol>

## 4.2.4 Världen

Världen är där entiteter kan röra sig fritt. Världen består av olika objekt som tiles och olika detaljer. I dessa testfallen provar vi om världen laddas korrekt från en fil. Vi testar även om kameran förljer spelaren korrekt.

<b>ID</b>	TC-U-V1	
<b>Rubrik</b>	Ladda levels från fil	
<b>Förberedelse r</b>	Skapa en level i ett .bdmap format	
<b>Teststeg</b>		<b>Förväntat resultat</b>
<ol style="list-style-type: none"> <li>1. Skapa en knapp på kartan som laddar filen.</li> <li>2. Se om där är fel i algoritmen och om banan är som den ska?</li> </ol>		<ol style="list-style-type: none"> <li>1. Banan laddas in helt korrekt och fungerar som den ska.</li> </ol>

<b>ID</b>	TC-C-VC2	
<b>Rubrik</b>	Kameran skall förflytta sig smidigt efter spelaren	
<b>Förberedelse r</b>	Skapa en spelare i världen.	
<b>Teststeg</b>		<b>Förväntat resultat</b>
<ol style="list-style-type: none"> <li>3. Flytta spelaren i olika riktningar</li> <li>4. Hoppa runt</li> </ol>		<ol style="list-style-type: none"> <li>2. Testaren för avgöra om kameran rör sig smidigt efter spelaren som är behagligt för användaren.</li> </ol>

## 4.2.5 Tiles

Tiles är huvudsakligen rektanglar som kan placeras i tre lager. De kan placeras i bakgrunden samt framför spelaren. Sedan finns det ett lager som spelaren befinner sig på. Detta lager hanterar kollision mellan entiteter och rektanglarna. I följande testfall testar vi om de ritas ut korrekt med rätt texturer.

<b>ID</b>	TC-U-T1	
<b>Rubrik</b>	Tiles i värden ritas ut rätt	
<b>Förberedelse r</b>	Skapa ett antal tiles och placera dem i värden för se om det ritas ut rätt	
<b>Teststeg</b>		<b>Förväntat resultat</b>
2. placera ut slumpmässigt tiles i världen på olika lager. 3. Start och se om de ritas ut i rätt ordning.		3. Tilesen ritas ut på det ställe man placerat dem 4. Tilesen har samma storlek som angetts

<b>ID</b>	TC-U-T2	
<b>Rubrik</b>	Tiles med texture algoritm	
<b>Förberedelse r</b>	Skapa ett antal tiles och placera dem i värden för se om det ritas ut rätt	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Testa med en tile om algoritmen fungerar. 2. Testa med flera olika texturer på tilesen och se om det ritas ut rätt.		5. Texturerna lägger sig på ett snyggt sätt så de passar ihop. 6. Tilesen "klipper" texturer som är utanför tilen.

## 4.2.6 AI

Ai är fienderna i spelet. Dessa kräver någon typ av kunskap om världen för att kunna navigera på tilesen. I dessa testfall provar vi om fienderna kan skjuta, dö samt patrullera utan att verka dumma.

<b>ID</b>	TC-C-AV1	
<b>Rubrik</b>	AI patrullering	
<b>Förberedelse r</b>	Skapa ett antal tiles och placera ut entiteter med Ai	

Teststeg	Förväntat resultat
<ol style="list-style-type: none"> <li>1. Studera hur entiteten rör sig.</li> <li>2. Hur den rör sig vid kanter.</li> <li>3. Hur den rör sig när den går in i en väg.</li> </ol>	<ol style="list-style-type: none"> <li>1. Går fram och tillbaka på "plan" tiles</li> <li>2. Vänder sig om när den når en vägg eller kant</li> <li>3. Stannar upp då och då.</li> </ol>

<b>ID</b>	TC-C-AE2	
<b>Rubrik</b>	AI skjuter entitet (spelare)	
<b>Förberedelse r</b>	Skapa antal fiender samt en spelar entitet.	
<b>Teststeg</b>		<b>Förväntat resultat</b>
<ol style="list-style-type: none"> <li>1. Studera om AI:n skjuter spelaren.</li> <li>2. Studera om AI:n skjuter genom vägar.</li> </ol>		<ol style="list-style-type: none"> <li>1. AI:n skall skjuta mot spelaren, men spelaren skall kunna undvika.</li> <li>2. AI:n skall inte kunna skjuta genom vägar. Den ska inte kunna skjuta om där är en väg emellan. Raycast mellan AI:n och spelaren.</li> </ol>

<b>ID</b>	TC-C-AT3	
<b>Rubrik</b>	AI:n skall dö	
<b>Förberedelse r</b>	Skapa antal fiender samt en spelar entitet.	
<b>Teststeg</b>		<b>Förväntat resultat</b>
<ol style="list-style-type: none"> <li>1. Skjut AI med tasern</li> <li>2. Studera hur AI:n reagerar</li> <li>3. Dör AI:n?</li> </ol>		<ol style="list-style-type: none"> <li>1. AI:n skall skaka när den blir skjuten av tasern.</li> <li>2. AI:n skall försvinna efter en stund.</li> <li>3. AI:n skall inte skjuta under tiden den blir skjuten.</li> </ol>

## 4.2.7 Taser

Taser är vapnet som spelaren styr i spelet. Det vi testar i detta testfall är ifall spelaren kan skjuta. Samt att skotten funkar korrekt genom att det skapas på rätt plats och går i rätt riktning.

<b>ID</b>	TC-U-T1
-----------	---------



<b>Rubrik</b>	Taser skjuter rätt	
<b>Förberedelse r</b>	Skapa en entitet (spelare).	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Tryck på "actionknappen". 2. Studera taserns "skott". 3. Studera taserns mätare.		1. Om spelaren inte är på en bomb skall tasern kunna skjutas. 2. Tasern ska skjuta år de håll som spelaren tittar. 3. Tasern ska dra från mätaren när man håller inne. 4. Är mätaren tom ska det ej gå att skjuta.

## 4.2.8 Bomb

Bomben är det slutgiltiga målet i spelet. Det ska kunna gå att deaktivera genom att trycka på 'action knappen'. Det ska även kunna gå att explodera genom att tiden har tagit slut. Samt att det ska gå att komma in i en annan fas, 'sladdfasen' där man ska klippa sladdar i rätt ordning.

<b>ID</b>	TC-U-B1	
<b>Rubrik</b>	Bomb animation	
<b>Förberedelse r</b>	Skapa en värld med en bomb.	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Ta sig till bomben med spelaren 2. Använda "action knappen" på bomben 3. Studera animationen		1. Animationen hackar inte. 2. Animationen täcker hela skärmen. 3. Animationen animeras korrekt.

<b>ID</b>	TC-U-B2	
<b>Rubrik</b>	Bomben sprängs	
<b>Förberedelse r</b>	Skapa en värld med en bomb.	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Spela spelet fram till bomben. 2. Låt tiden ta slut.		1. Spelet går till "Game Over" 2. Det står att bomben sprängdes.

<b>ID</b>	TC-U-B3	
<b>Rubrik</b>	Bomben sladdar	
<b>Förberedelse r</b>	Skapa en värld med en bomb.	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Spela spelet fram till bomben. 2. Studera om där är fem sladdar varav en lyser		1. Fem sladdar existerar på skärmen

<b>ID</b>	TC-U-B4	
<b>Rubrik</b>	Sladdarnas funktionalitet	
<b>Förberedelse r</b>	Skapa en värld med en bomb.	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Spela spelet fram till bomben. 2. kapp lys kabel. 3. kapp olys kabel.		1. Lys kabel → Annan oklippt kabel lyser. 2. Olys kabel → Bomben sprängs och du förs till "Game Over".

### 4.3 Level editor

Level editor är ett eget program som vi skapat för syftet att skapa olika nivåer snabbare och effektivare. Vi provar på att ladda och spara filer i programmet.

<b>ID</b>	TC-U-L1	
<b>Rubrik</b>	Skapa objekt	
<b>Förberedelse r</b>	Starta leveleditorn.	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Tryck på de olika knapparna i leveleditorn. 2. Studera om rätt objekt skapas. 3. Skapa objekt i olika lister genom att välja olika lager.		1. Korrekt objekt skapas. 2. De skall skapas i angivet lager. 3. Markering sker endast av aktuellt lager.

<b>ID</b>	TC-U-L2	
<b>Rubrik</b>	Objekt funktionalitet	
<b>Förberedelse r</b>	Skapa ett objekt	
<b>Teststeg</b>		<b>Förväntat resultat</b>
<ol style="list-style-type: none"> <li>1. Markera objektet och tryck på "Properties".</li> <li>2. Modifera värden och spara.</li> <li>3. Studera skillnaden.</li> </ol>		<ol style="list-style-type: none"> <li>1. Modifering i properties skall påverka objektet. (storlek och position)</li> <li>2. Öppna Properties igen ska visa samma värden.</li> </ol>

<b>ID</b>	TC-U-L3	
<b>Rubrik</b>	Spara och ladda	
<b>Förberedelse r</b>	Skapa en level med olika objekt	
<b>Teststeg</b>		<b>Förväntat resultat</b>
<ol style="list-style-type: none"> <li>1. Spara leveln med spara knappen</li> <li>2. Studera om filen har sparats</li> <li>3. Skapa ny level.</li> <li>4. Ladda den sparade leveln.</li> </ol>		<ol style="list-style-type: none"> <li>1. Filen skall sparas som .bdmap</li> <li>2. Leveln skall vara identisk med den du spara.</li> </ol>

<b>ID</b>	TC-U-L4	
<b>Rubrik</b>	Kamera	
<b>Förberedelse r</b>	Starta leveleditorn	
<b>Teststeg</b>		<b>Förväntat resultat</b>
<ol style="list-style-type: none"> <li>1. Använd pilarna för att flytta kameran.</li> <li>2. Använd ljuset på musen för att zooma.</li> </ol>		<ol style="list-style-type: none"> <li>1. Kameran skall flyttas i sidled</li> <li>2. Kameran skall zooma in och ut.</li> </ol>

<b>ID</b>	TC-U-L5	
<b>Rubrik</b>	Testkörning	

<b>Förberedelse r</b>	Skapa en level med olika objekt	
<b>Teststeg</b>		<b>Förväntat resultat</b>
1. Använd setup för att markera spelet (.jar) 2. Tryck "Play" 3. Studera om den level du gjort är i spelet.		1. En temp.bdmap skall ha skapats bredvid spelplanen. 2. Spelet skall visa den level du gjort. 3. Alla objekt finns med i leveln.

## 5 Testmatris

\* TestfallID kan skrivas som TC-U-T då omfattar detta både TC-U-T1 och TC-U-T2

Prototyp			
KravID	TestfallID	Datum	Förkortad Beskrivning
FK-G-S-1 FK-G-S-2 FK-G-V-2 FK-G-V-3 FK-G-V-4	TC-C-PT*	2015-03-04	Om tilesen fungerar rätt.
FK-G-V-7	TC-C-PCP1	2015-03-05	Kameran rör sig smidigt.
FK-G-V-5	TC-C-PFP1	2015-03-10	Påverkas spelaren av fläktar.
FK-G-A-1 FK-G-A-4	TC-C-PTE1	2015-03-11	Rör fienderna sig rätt.

Spel			
KravID	TestfallID*	Datum	Förkortad Beskrivning
IFK-S-1	TC-C-SA1	2015-04-14	Android systemkrav
IFK-S-2	TC-U-S2		Framgång genom

			spelet
FK-N-1 FK-N-2	TC-U-N1	2015-03-27	Level menyn
FK-N-3	TC-U-N2		Låsta levelar
FK-N-4 FK-N-5	TC-U-N3	2015-04-23	Gamover menyn
FK-G-I-1 FK-G-I-2 FK-G-S-1 FK-G-S-2 FK-G-V-2 FK-G-V-3 FK-G-V-4	TC-U-EV11	2015-03-26	Entiter kan röra sig fritt i världen samt integrera med Tiles
FK-G-I-3 FK-G-S-4	TC-U-EB2	2015-03-30	Actionknappen mot bomb.
FK-G-S-5	TC-U-EE3	2015-04-23	Spelaren skall kunna dö.
FK-G-V-5	TC-C-EF4	2015-04-17	Entiter och fläktar
FK-G-V-1	TC-U-V1	2015-05-06	Ladd levelar från fil
FK-G-V-7	TC-U-VC2	2015-03-24	Kameran förföljer spelaren i världen
FK-G-V-8	TC-U-T1	2015-03-24	Tilesen ritas ut på rätt ställen
FK-G-V-9	TC-U-T2	2015-03-31	Algoritmen för tilesen ritas ut rätt.
FK-G-A-1	TC-C-AV1	2015-04-11	Hur AIn patrullerar
FK-G-A-2	TC-C-AE2	2015-04-26	AIn skadar spelaren.
FK-G-A-3 FK-G-T-3	TC-C-AT3	2015-04-26	Spelaren dödar AIn.
FK-G-T-1 FK-T-T-2 FK-G-S-3	TC-U-T1	2015-04-13	Skjuta med Tasern.

FK-G-B-1	TC-U-B1	2015-04-13	Bomb animation.
FK-G-B-2	TC-U-B2	2015-04-21	Bomb tiden tar slut.
FK-G-B-3	TC-U-B3	2015-04-21	Bombsladdarna existerar.
FK-G-B-4 FK-G-B-5	TC-U-B4	2015-04-21	Bombsladdarna funktionalitet.

Leveleditor			
KravID	TestfallID*	Datum	Förkortad Beskrivning
FK-G-L-1 FK-G-L-3	TC-U-L1	2015-04-18	Kunna skapa objekt.
FK-G-L-2 FK-G-L-6	TC-U-L2	2015-04-24	Test funktionaliteten i objekt.
FK-G-L-4 FK-G-L-5	TC-U-L3	2015-05-07	Spara och ladda levelar.
FK-G-L-7	TC-U-L4	2015-04-20	Kamerans funktionalitet.
FK-G-L-8	TC-U-L5	2015-05-05	Man skall testköra levelar.