

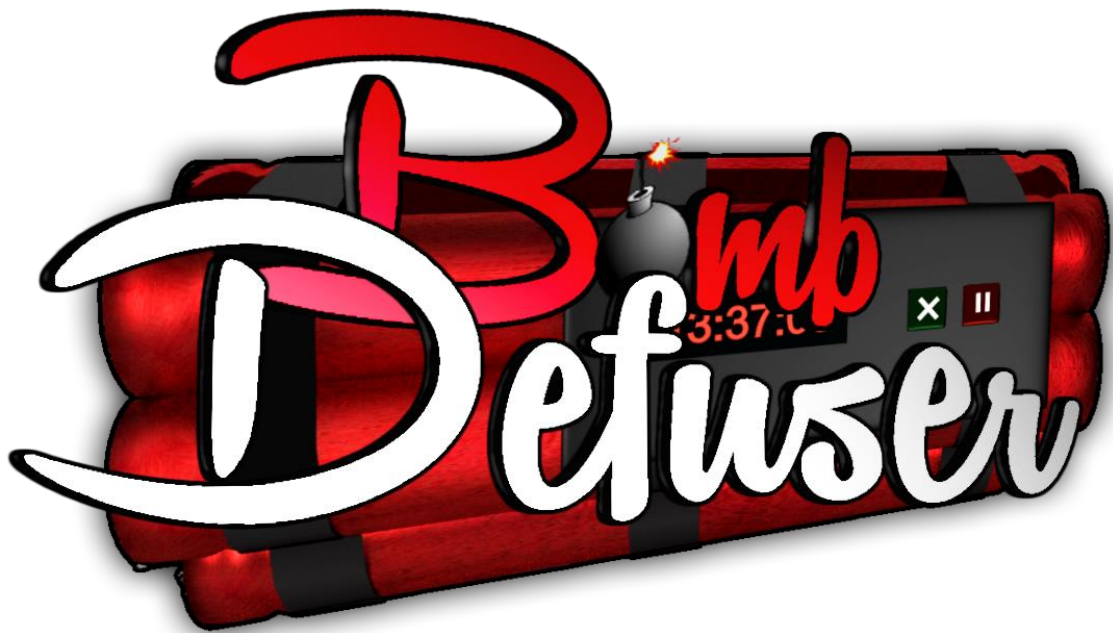
Designdokument  
**Bomb Defuser**

version: 1.0.2

2015-05-22

Handledare: Edward S. Blurock

Grupp 18



## Revision History

<b>Date</b>	<b>Version</b>	<b>Description</b>	<b>Author</b>
2015-04-07	0.0.0	First draft, fundament av dokument	Mika Lehtinen
2015-04-07	0.1.0	Skapat rubriker, fyllt på under rubriker	Mika Lehtinen
2015-04-11	0.1.1	Skapat Use-case diagram	Robin Andersson
2015-04-11	0.1.1	Skrivit om Gameplay och story	Mika Lehtinen
2015-04-12	0.1.2	Uppdaterade Use-case diagram	Mika Lehtinen
2015-04-14	0.1.3	Skrivit om upplägget av speltesterna	Mika Lehtinen
2015-04-14	0.1.3	Rättat stavning	Peter Lindberg
2015-04-16	0.2.0	Skrivit revisionshistorik, innehållsförteckning och skrivit use-case beskrivning	Mika Lehtinen
2015-05-17	0.2.1	Klass hierarki, Managers	Robin Andersson
2015-05-17	0.2.2	Skapade system-diagram	Anton Björkman
2015-05-17	0.3.0	Skrivit om Kontext-Diagram	Mika Lehtinen
2015-05-18	0.3.1	Överblick av system	Robin Andersson
2015-05-18	0.3.2	Kontrolläst, Rättat och uppdaterat samtliga bilder	Mika Lehtinen
2015-05-19	0.4.0	Kontrolläst och finslipat.	Mika Lehtinen
2015-05-19	0.4.1	Kontrolläst och rättat.	Peter Lindberg

2015-05-19	0.4.2	Klassdiagram, diagramförklaring, finslipat	Robin Andersson
2015-05-19	1.0.0	Klassdiagram, diagramförklaring, finslipad.	Patrik Nilsson
2015-05-22	1.0.1	Klassdiagram kommentar	Robin Andersson
2015-05-22	1.0.1	Kontrolläst och rättat.	Peter Lindberg
2015-05-23	1.0.2	Tidiga konceptbilder	Peter Lindberg

## Innehållsförteckning

### 1 Inledning

1.1	Syfte .....	6
1.2	Omfattning .....	6
1.3	Definitioner, akronmyr och förkortningar .....	6

1.4	Referenser .....	
...	7	
1.5	Struktur .....	
..	7	
<b>2</b>	<b>Designhistoria .....</b>	
....	8	
<b>3</b>	<b>Spelvision .....</b>	
.....	9	
<b>4</b>	<b>Gameplay</b>	
4.1	Översikt .....	
...	10	
4.2	Spelsätt .....	
...	11	
4.3	Winning conditions .....	
.	11	
4.4	Levels .....	
...	11	
4.5	Vad gör spelet roligt? .....	
..	11	
4.6	Upplägg av speltest .....	
..	12	
4.6.1	Speltest 1 .....	
...	12	
4.6.2	Speltest 2 .....	
...	12	
4.6.3	Speltest 3 .....	
...	12	

<b>5</b>	<b>Kontextdiagram.</b>	.....	
.....		13	
<b>6</b>	<b>Systemdiagram.</b>	.....	
.....		14	
<b>7</b>	<b>Användargränssnitt.</b>	.....	
.....		14	
	7.1 Startskärm.	.....	
...		14	
	7.2 Levelselect skärm.	.....	
..		15	
	7.3 Ingame skärm.	.....	
..		15	
	7.4 Cutscene skärm.	.....	
...		16	
	7.5 Vinst/förlust skärm.	.....	
.		16	
<b>8</b>	<b>Usecase</b>		
	8.1 Use-casediagram.	.....	
...		17	
	8.2 Use-case beskrivning	.....	
...		17	
<b>9</b>	<b>Story</b>	.....	
.....		18	
<b>10</b>	<b>Överblick av spelet.</b>	.....	
.....		18	

**11 Klassdigram.....**  
..... 20

# 1 Inledning

## 1.1 Syfte

Syftet med detta dokumentet är att lägga ett fundament för projektet som vi kommer följa genom projektets gång.

## 1.2 Omfattning

I detta dokumentet kommer vi att ta upp en övergripande blick på designarbetet där vi bland annat går in väldigt djupt på designarbete. Vi kommer inte gå in på djupet i funktionen av specifika klasser i systemet.

## 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.

**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.

**Icke-funktionella krav:** Är ett krav som inte är direkt relaterat till vad systemet ska göra  
**funktionella krav:** är ett krav som beskriver vad systemet ska göra

**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.

**Renderad:** Är en process som genererar en bild eller en animerad filmsekvens från en 2D eller 3D model.

**Mekaniker:** Samlingsord för funktionalitet som förekommer i spel, exempel på detta är till exempel hopp, rörelse.

**AK-47:** Ett sjukligt farligt gevär från ryssland.

**Winning Condition:** Den samling krav som måste uppfyllas för att du ska klara den angivna nivån och på så vis avancera vidare till nästa.

**Testscenario:** Ett specifikt scenario man sätter upp för att prova en viss funktion vid ett speltest.

**porta:** att porta ett spel till en plattform innebär att du gör spelet brukbart för det bestämda systemet du vill använda.

## 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 studenter 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:

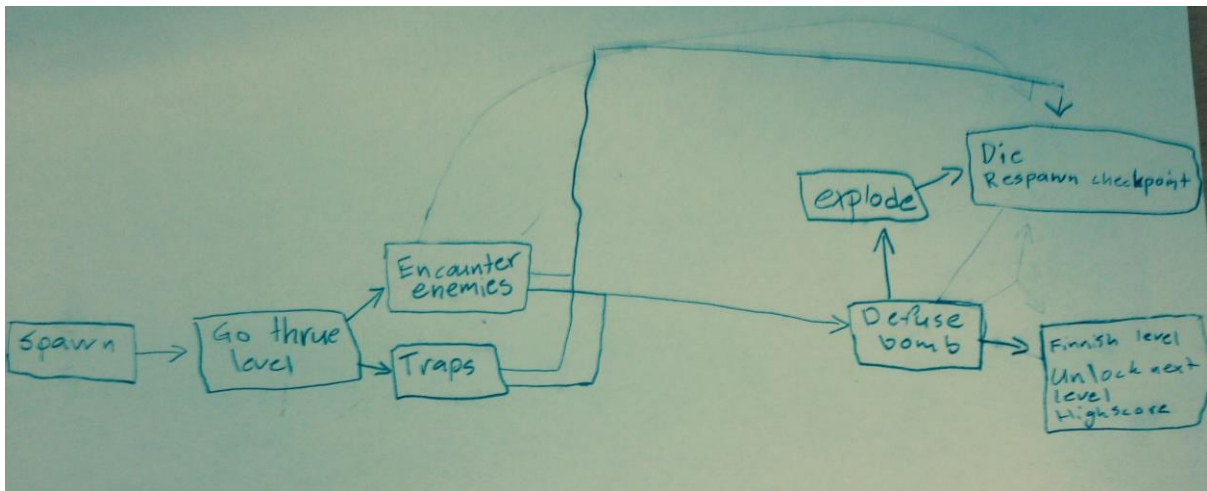
- [www.java.com](http://www.java.com)
- [www.blender.org](http://www.blender.org)
- [www.microsoft.com](http://www.microsoft.com)

## 1.5 Struktur

Dokumentet är strukturerat på så vis att det inleder med en väldigt ingående beskrivning på designhistoriken. Dokumentet går sedan över till lite mer överskådlig information kring projektet som till exempel användargränssnitt, Kontext-diagram och Use-case diagram. I slutet av dokumentet kommer det ett väldigt ingående segment som förklarar hur systemets inre struktur är uppbyggt.



## 2 Designhistoria



Vi startade designarbetet med att sätta oss ner och brainstorma. Vi försökte visualisera hur spelupplägget kunde tänkas se ut, och bilden ovan ger en liten representation av resultatet.

Innan vi påbörjade arbetet med koden så tog vi fram ett par exempeltexturer för både karaktär och miljö så vi hade en chans att bestämma oss vilken stil vi skulle använda oss av redan innan projektet började. Vi diskuterade vilken riktning vi ville gå med det grafiska arbetet och beslutade oss för att arbeta med den pixelart-stil som en av utvecklarna framställt. Vi gjorde även skisser på användargränssnittet så vi hade en god bas lagd för kodningen. Spelnivåerna utvecklade vi mycket likt traditionella plattformsspel med fällor, fallgropar och fiender och har därför inte förändrats i någon större utsträckning under utvecklingsprocessen.

I ett väldigt tidigt skede av processen var alla utvecklare i projektet överens om att spelet skulle utvecklas till plattformen Android. Detta huvudsakligen för att många i gruppen har god erfarenhet med ramverket LibGDX som vi använder oss av, detta ramverket är också väldigt likt C#'s XNA som samtliga utvecklare i projektet har erfarenhet av.

Bomben var väldigt central under hela processen och vi arbetade väldigt mycket med designen för den tidigt i processen. Vi ville ha en spännande och stilren övergång in till fasen där man desarmerar bomben, så vi bestämde oss för att ha en cutscene när bombväskan öppnas. En av de grafikansvariga utvecklarna tog fram en cutscene renderad i 3D för att ge en stark visuell effekt när man öppnar lådan. Och vid implementation så visade det sig vara en väldigt känslöförhöjande substans för stämningen.

Redan innan prototypen påbörjats beslutades att vi skulle ha med en del fällor och mekaniker som uteslöts från den slutgiltiga produkten. Vi hade en vision om att använda oss av olika typer av fällor så som spikar och plattformar i rörelse för att göra spelupplevelsen mer

dynamisk. Men allt eftersom utvecklingen fortlöpte upptäckte vi att ovannämnda mekaniker skulle bli väldigt svåra att implementera på ett naturligt sätt, då de inte passar väl in i spelets natur. Vi utvecklade istället fläktar, som man på ett strategiskt sätt kan använda både till sin för- och nackdel, vilket visade sig passa in i spelet på ett mycket mer naturligt sätt.

I samband med prototypen utvecklade vi också tre lager man kan rita texturer i; ett bakgrundslager, ett kollisionslager och ett förgrundslager. Detta gav oss möjligheten att få karaktären att gå på, framför och bakom texturer. Vilket gav oss stora möjligheter att implementera pusselmekaniker i spelet till skillnad från traditionella fällor som vi ursprungligen hade planerat.

När prototypen stod färdig hade vi implementerat samt tagit bort de mekaniker som vi ansåg oss behöva och kunde då påbörja planeringen av huvudprodukten. Vi upptäckte nu att det tog väldigt lång tid att skapa nivåer med den nivå-implementeringen vi använder oss av. Så tre av utvecklarna i projektet påbörjade utvecklingen av en level-editor för att effektivisera skapandet av nivåer.

I utvecklingen av huvudprodukten så kom vi fram till att det kändes väldigt icke-originellt att döda fienderna genom att hoppa på dem. Vi bestämde oss i detta skede för att vi inte ville använda oss av ett traditionellt vapen som svärd eller pilbågar. Efter ett möte där alternativa lösningar diskuterades kom vi överens om att vi istället beväpnar huvudkaraktären med en taser, som han kan använda för att ge sina fiender en elektrisk stöt. Att beväpna karaktären med en taser kändes fräscht istället för att, som i många andra spel, använda traditionella tekniker så som hopp eller svärd. Och vi ansåg att detta passade bättre in på huvudkaraktärens personlighet som en tuff kille med solglasögon.

I samband med utvecklingen av vapnet så utvecklade vi designen på fienderna så att även dessa var färdigställda. Vi kom överens med vår grafik-ansvariga utvecklare om att vi ville ha mörkklädda fiender med valfritt vapen. Vi tog fram en prototyp av fienden och provade den, det kändes väldigt naturligt med svartklädda fiender med en AK-47 så vi städade texturen lite och implementerade den som vår slutgiltiga fiende.

## 3 Spelvision

Bomb Defuser är ett fartfyllt och actionspäckat plattformsspel i 2D-grafik som tar dig med på äventyr runt Europa, för att fullfölja ditt uppdrag att desarmera bomben i slutet av nivån.

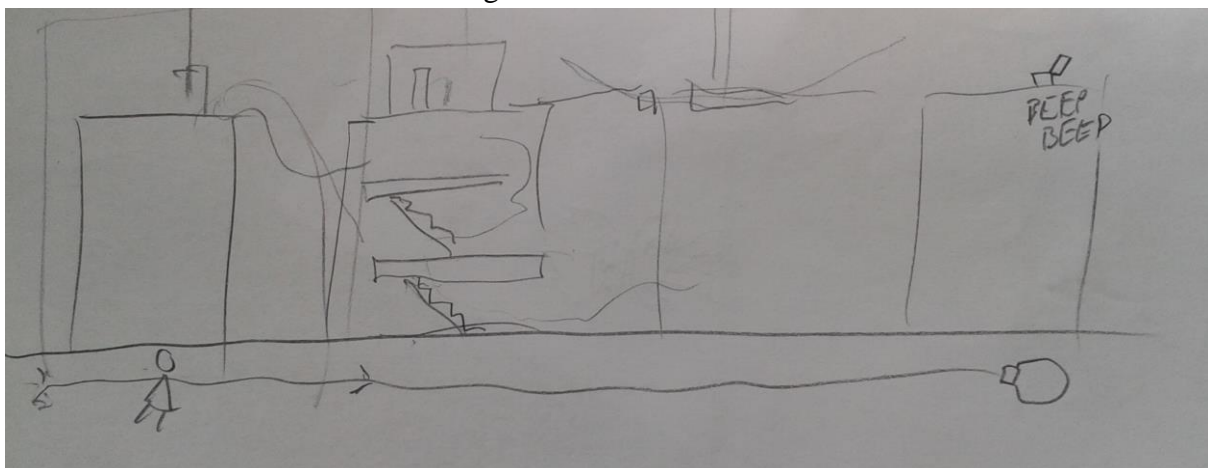
Plattformsspel är väletablerat och många välkända titlar går under denna kategorin, härifrån har vi hämtat inspiration från bland annat Super Mario's plattformssuccéer från slutet av 80-talet. Spelsättet är väldigt intensivt och går väldigt fort och nivåerna är fyllda med fiender och kluriga fällor för att göra spelet dynamiskt. Vi har en vision att skapa ett annorlunda men roligt plattformsspel som lämpar sig att spela överallt, om du så sitter på bussen eller hemma i soffan. Med den tillkommande level-editorn har du även möjlighet att skapa dina egna banor som du kan dela med dig av till dina vänner.



## 4 Gameplay

### 4.1 Översikt

Likt ett traditionellt plattformsspel kommer spelaren att ta sig genom och utforska en 2D-miljö där man stöter på en mängd hinder och fiender av olika karaktär i kampen om att nå det slutgiltiga målet. Spelet spelas offline och är ett singleplayer-spel som utspelar sig i en utomhusmiljö i olika Europeiska storstäder och mestadels ovanpå hustak. Det slutgiltiga målet är att besegra fienderna i din väg och ta sig till bomben som är placerad vid banans slut för att desarmera den innan tiden har gått ut.



### 4.2 Spelsätt

Du styr en karaktär i en 2D-miljö där du använder piltangenterna på skärmen för att röra dig höger och vänster och knapparna A och B för att hoppa och använda ditt vapen. För att klara en nivå och avancera till nästa måste du ta dig från startpositionen, genom nivån och nå det slutgiltiga målet vid nivåns slut. Under nivåns gång kommer spelaren att stöta på hinder som man tar sig förbi genom att hoppa över dem. Spelaren kommer även att stöta på fiender som kommer skjuta mot en när man kommer inom deras synfält och dessa besegrar spelaren genom att använda sitt vapen för att ge dem en elektrisk stöt.

### **4.3 Winning Condition**

För varje nivå spelaren är på så finns det en bomb i slutet av banan. Målet är att desarmera denna bomb innan tiden går ut och den exploderar. Lyckas man med detta klarar man nivån och går därmed vidare. Det slutgiltiga målet i spelet är att desarmera bomben på varje nivå som finns tillgänglig i spelet.

### **4.4 Levels**

De tillkommande nivåerna som spelaren kommer att ta sig igenom utspelar sig i utomhusmiljö, till största delen på hustak runt om i världen. Den första nivån du måste ta dig genom är Malmö, när du fullföljt ditt mål på denna nivå ges spelaren fler möjligheter att välja andra nivåer som till exempel Berlin och London.

Utöver dem förbestämda nivåerna har man också möjlighet att skapa sina egna nivåer med den tillkommande Level-Editorn och här är det enbart spelarens egen fantasi som sätter gränser.

### **4.5 Vad gör spelet roligt?**

Det som huvudsakligen gör spelet roligt är utmaningen i dem olika nivåerna. Spelet är utvecklat för att du ska ha möjligheten att fortsätta där du sist avslutade och kunna avsluta när du vill. Detta gör att du har möjlighet att spela en nivå under en bussresa eller under tiden du har rast i skolan. Att lätt kunna plocka upp och lägga ner spelet är en väldigt populär egenskap i dagsläget.

## **4.6 Upplägg av speltest**

Vi har planerat att hålla tre speltester utspritt med jämna mellanrum över arbetsprocessen för att kontrollera att tilltänkt arbete fungerar som planerat.

### **4.6.1 Speltest 1**

Det första speltestet utförs på prototypen vi skapat av spelet. Under detta speltestet fokuserar vi huvudsakligen på att testa grundmekanik och testa karaktären så rörelse och kollision fungerar som det ska, vi kommer testa följande.

- Karaktärens rörelse åt höger och vänster, hopp
- Kollision mellan karaktär och tiles

### **4.6.2 Speltest 2**

Vid detta speltestet så har vi lämnat prototypen bakom oss och arbetat en hel del med huvudprodukten. I detta speltest kommer vi fokusera väldigt mycket på världen samt objekt och fiender som är utplacerade runt banan, vi kommer även att testa spelarens vapen. Detta speltestet kommer likt det första att vara ett internt speltest där vi sätter upp olika testscenarier för de olika delarna som är aktuella för detta test. Här kommer vi testa följande.

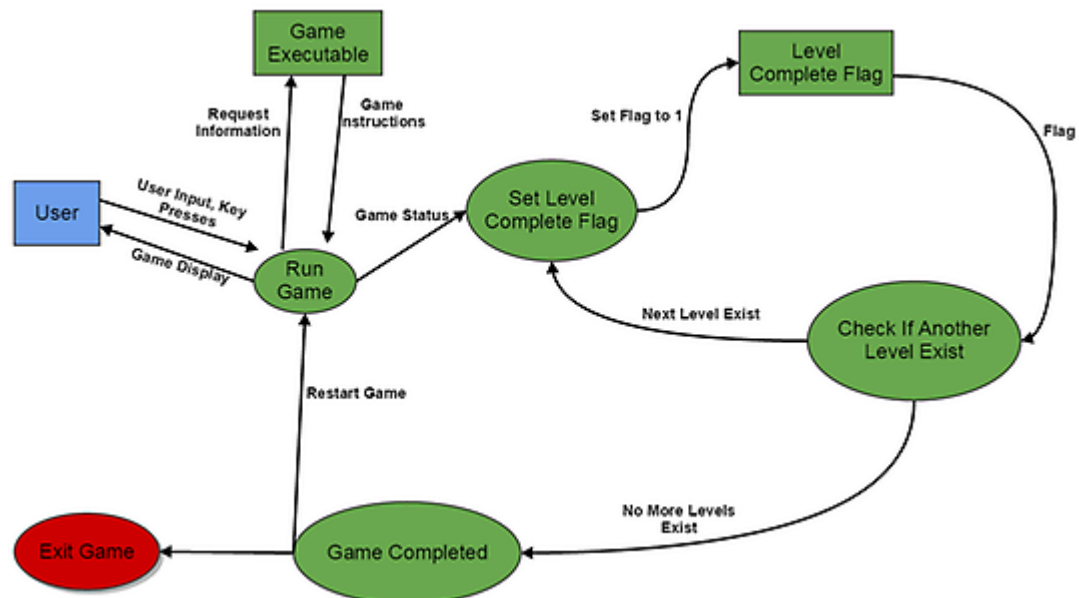
- Spelarens vapen
- Fiendernas AI (rörelse, attack, line of sight)
- Kollision mellan spelare och fiender
- Eventuell balansering av fiende
- Fläktarna (styrka, funktionalitet)
- Kollision mellan spelare och fläkt

### **4.6.3 Speltest 3**

Detta speltest kommer äga rum i slutfasen av arbetet när projektet nästan står klart, även detta är ett internt speltest. Här kommer vi på en väldigt övergriplig nivå testa spelet som helhet. Vi kommer att ha en testnivå där vi testar hur allting smälter samman och fungerar som en enhet. Här kommer vi huvudsakligen att testa följande.

- Kulhetsfaktor
- Svårighetsgrad
- Generell spelupplevelse

## **5 Context-Diagram**

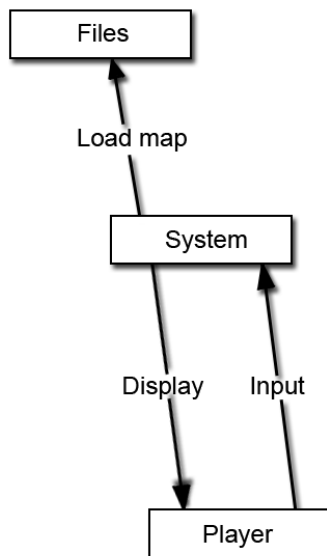


Kontext-diagrammet ovan visar med användaren som utgångspunkt hur man interagerar med spelet. Blå ruta representerar den fysiska spelaren, gröna bubblor och rutor representerar dem olika stegen du tar genom spelet, och röd bubbla representerar avslut. Den fysiska spelaren kommer genom att starta spelet med ett knapptryck och spelet kommer försöka visa spelaren en spelplan samt en kontrollbar karaktär. Spelet kommer att kontrollera att spelet är körbart, om detta är fallet kommer den begärda data att visas.

När spelnivån har nått sitt slut kommer spelet flagga för att första nivån är avklarad, därefter kontrollerar spelet om det finns fler nivåer tillgängliga. Finns det fler tillgängliga nivåer kontrollerar spelet att datan för nästa nivå är körbart och visar sedan detta på skärmen. Denna loop kommer upprepas så länge det finns tillgängliga nivåer att avancera till.

När det inte finns fler nivåer tillgängliga att ladda så är spelet avklarat och spelaren skickas därefter till slutskärmen där man väljer om man vill starta om spelet och börja från första nivå igen eller om man vill stänga av spelet.

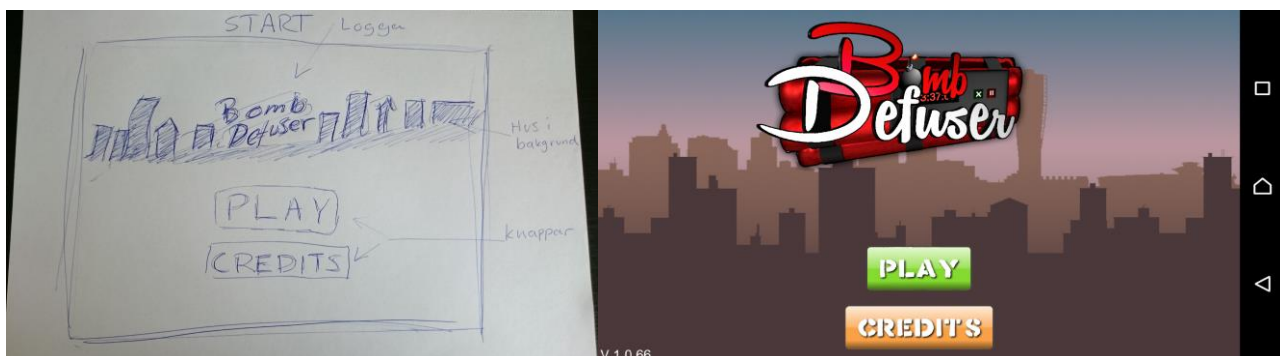
## 6 System-Diagram



Detta är system-diagrammet för huvudprodukten som på ett väldigt övergripligt sätt representerar hur de olika systemen kommunicerar med varandra. Systemet kommer i första skedet ladda den angivna nivån som du vill spela. När detta är genomfört så kommer spelaren med ett knapptryck att begära att få spela nivån, när systemet har mottagit denna information så kontrolleras det att nivån finns tillgänglig att ladda, och visar sedan den på skärmen.

## 7 Användargränssnitt

Innan vi påbörjade kodningen av projektet så arbetade vi fram ett par skisser av användargränssnittet som vi utgick ifrån under projektets gång.



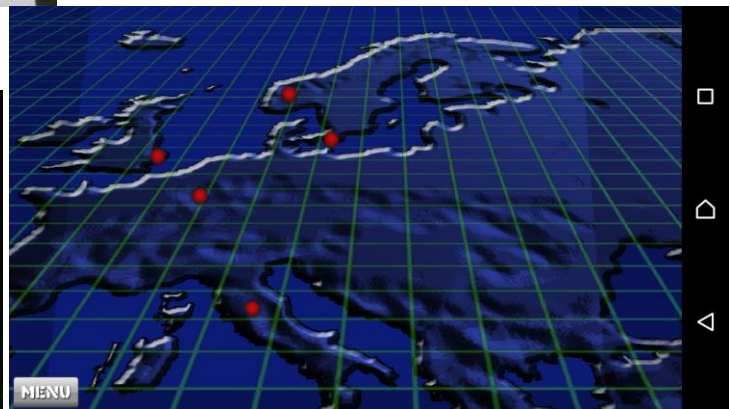
### 7.1 Startskärm



Startskärmen är tänkt att bestå av en bakgrundsbild på staden med den officiella Bomb Defuser Loggan framför. Följt av en Play knapp som tar dig till Level-Select skärmen och en Credits knapp som visar dig namnen på deltagande i projektet.

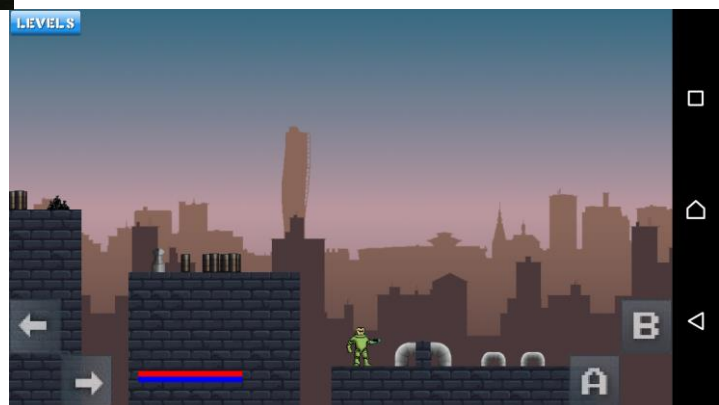
## 7.2 Level-Select skärmen

Level-Select skärmen är tänkt att bestå av en Europakarta med markeringar över städer som Malmö och London. Klickar du på en markering så tar spelet dig till den angivna nivån.

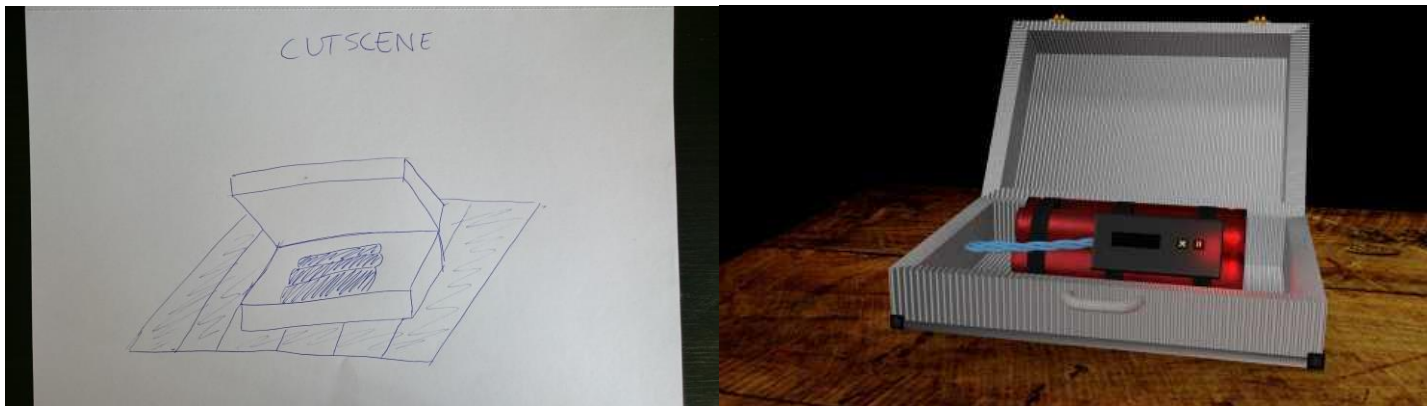


## 7.3 Ingame-skärmen

En tidig skiss på hur nivån kan tänkas se ut. Det är meningen att du ska ta dig till slutmålet som är bombväskan i slutet av nivån. När du öppnar den spelas en cutscene upp.







## 7.4 Cutscene skärmen

Cutscenen kommer vara en 3D-renderad scen på tre sekunder som visar hur väskan innehållande bomben öppnas, och det är nu spelarens uppgift att desarmera bomben.

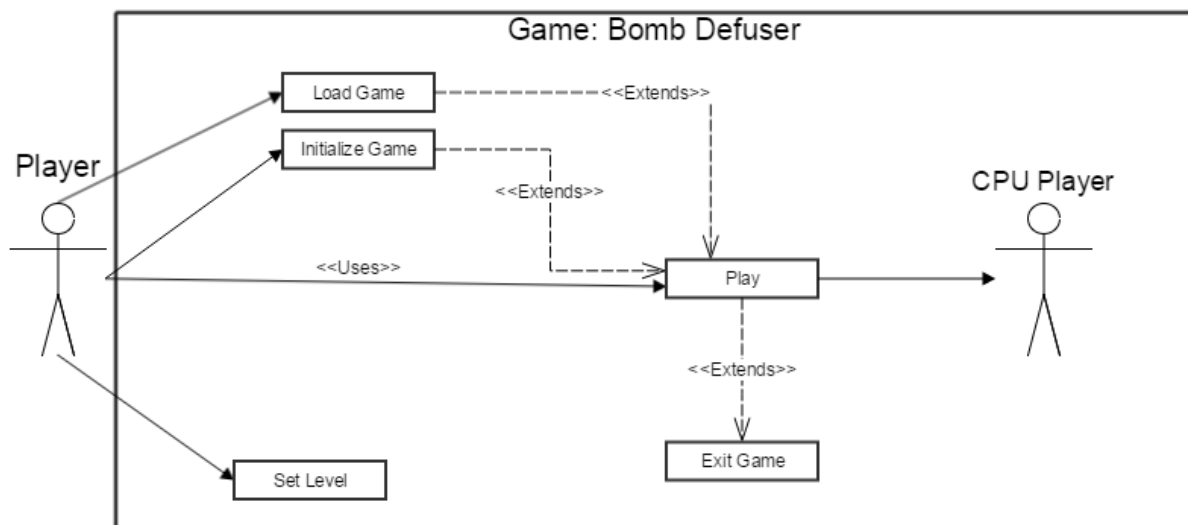


## 7.5 Winning/förlust skärmen

Slutskärmen både för lyckad och misslyckad bombdesarmering är tänkt att se likadana ut fast med olika text. Du kommer få två knappar för att välja vad du vill göra med exploderande byggnader i bakgrunden.

## 8 Use Cases

### 8.1 Use case-Diagram



### 8.2 Use case-beskrivning

Spelaren som i detta use-case-diagrammet är vår huvudaktör som har som mål att spela spelet. För att detta ska kunna gå att genomföra är det en förutsättning att spelaren har en Android-telefon av nyare modell (4.0 eller nyare) med spelet Bomb Defuser installerat.

Den huvudsakliga uppgiften vår huvudaktör har för att nå sitt mål är att trycka på knappar på skärmen. Vid uppstart av applikationen kommer aktören att ha möjlighet att trycka på knappen 'Play' som i sin tur initierar och laddar spelet för att du ska ha möjlighet att välja vilken bana du vill spela. När huvudaktören tryckt på den bana som ska spelas kommer nivån att ladda och ge en bild på telefonens skärm.

Systemet kommer väl i spelläge att producera information beroende på hur huvudaktören interagerar med spelet, där beroende på vilka knappar man använder så agerar karaktären på olika sätt. I samma skede kommer aktören också att mottaga information beroende på hur banorna ser ut och hur fiender beter sig.

Vid eventuell förlust kommer spelet abruptt att gå över till ett Game Over läge där du med hjälp av knapptryck väljer om du vill spela om nivån eller stänga av spelet.

## 9 Story

Året är 2146 och du spelar en ung äventyrshungrig herre av okänd nationalitet. En dag drar en ond ännu okänd makt sprider sig över Europa med ett enda syfte, att eliminera och överta alla storstäder i Europas länder. Denna ännu okända skara maktmissbrukande individer har stationerat ut högteknologiska bomber i storstäderna för att fullfölja sina ondskefulla mål, och de måste desarmeras till varje pris för att rädda världen från fullständig katastrof.

Eftersom du är högtbildad bombtekniker så har Svea rikets konung erbjudit dig uppdraget att leta reda på och desarmera bomberna för att rädda Europa från detta ondskefulla dåd. Du godtar naturligtvis uppdraget och tar genast på dig din bombdräkt och ger dig ut mot den första destinationen att rädda, Malmö.

Vad konungen inte var medveten om är att det placerats ut fällor runt bomben för att göra den väldigt svåråtkomlig, även att det är beväpnade vakter som patrullerar kring bomben för att förhindra att någon desarmerar den. Du hade en aning om att du skulle stöta på hinder under ditt uppdrag och beväpnade dig med din taser och din listighet för att tackla de hinder som kan tänkas komma i din väg för att nå slutmålet.

## 10 Överblick av spelet

Klassdiagrammet nedan är en representation över spelets uppbyggnad.

**BombMain** är klassen som initierar spelet. Den sköter även uppdatering och rendering av olika typer av objekt och moduler. Moduler innebär en större mängd klasser eller objekt som interagerar med varandra och ger nödvändig funktionalitet.

**Globals** är globala variabler som används på flertal ställen i koden. Vi har valt att göra detta för enkelhetens skull.

**StateManager** är en modul som befinner sig i BombMain. Modulen ger funktionalitet till övergångar av scener.

### *Exempel 1*

LevelScreen → MenuScreen

### *Exempel 2*

LevelScreen → GameScreen

**Load** är algoritmen som laddar in nivåer via en text fil. Detta underlättar oss att kunna skapa flera nivåer utan att behöva "hårdkåda" nivåerna.

**Particles** är ett partikelsystem som används till fläktarna av spelet.

**SoundManager** är en modul som har hand om musik och ljud i spelet. Vi kan lätt anropa denna modul för att spela upp ljud.

**Utilities** är en grupp klasser som används som en verktygslåda. Exempel knappar, rullande bakgrund och filmsekvenser.

**World** representerar värden i spelet och innehåller saker som plattformar, spelaren, fiender etc.

**Bomb** är objekt i världen som spelaren skall ta sig till. Denna måste finnas i alla nivåer.

**Entity** är objekt i världen som påverkas av krafter och kollisioner av tiles.

**Enemy** är en AI som kan hantera sig runt i världen utan att verka dum. Detta är även den huvudsakliga motsåndet i spelet.

**Taser** innehåller när spelarens vapen skall laddas upp, när den skall skjuta, samt hur den skall skada fienderna (Enemy)

**Fans** är en del av tiles som har andra egenskaper. Denna "tile" kan ge spelaren en kraft åt olika riktningar.

**Hud** är den klass som hanterar objekt som skall följa med kameran. Så som spelarens liv och ammunition på tasern.

**Tiles** är den huvudsakliga objekten som förekommer i världen. Detta är de objekt som spelaren kan hoppa på och gör världen till en värld.

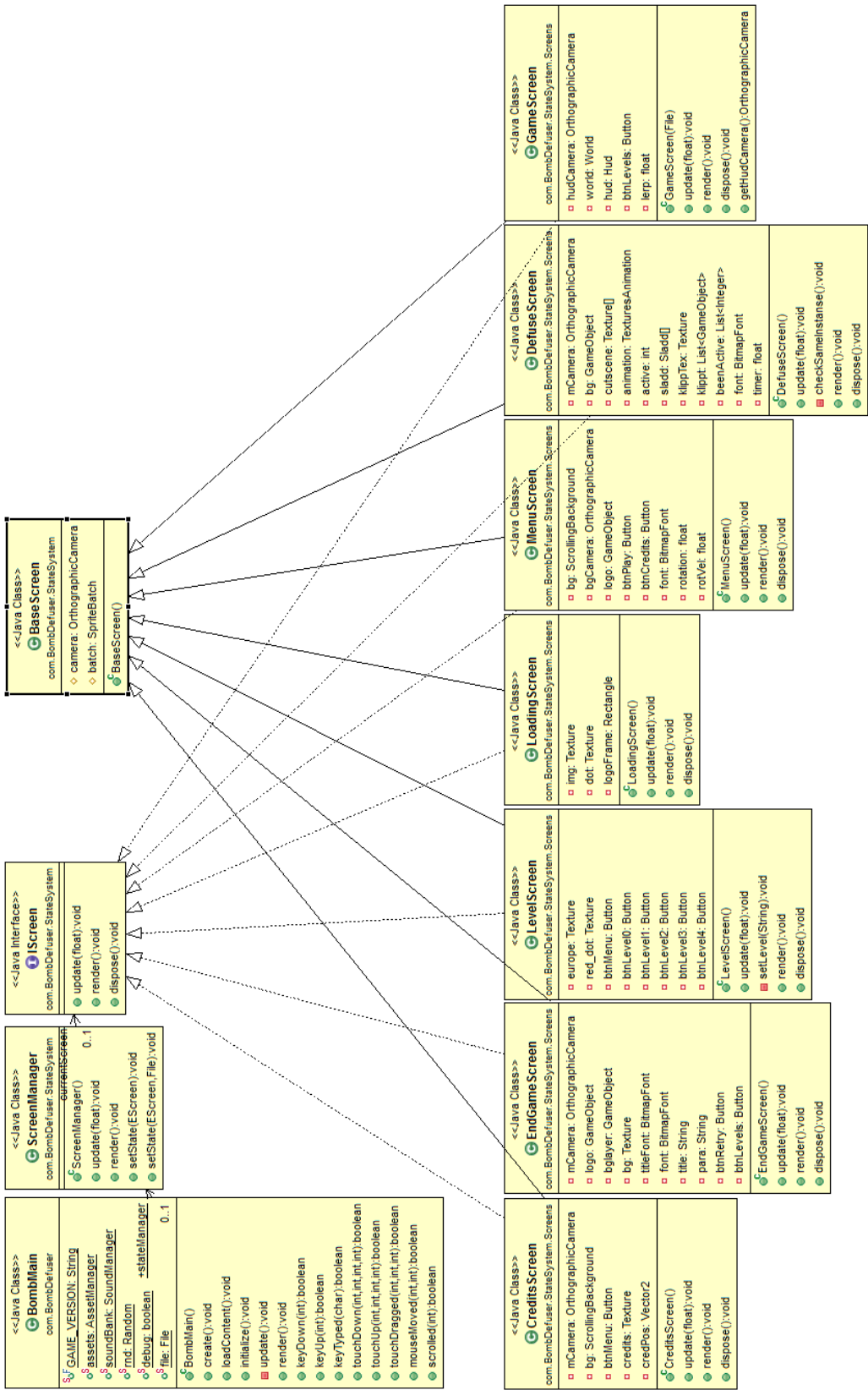
# 11 Klassdiagram

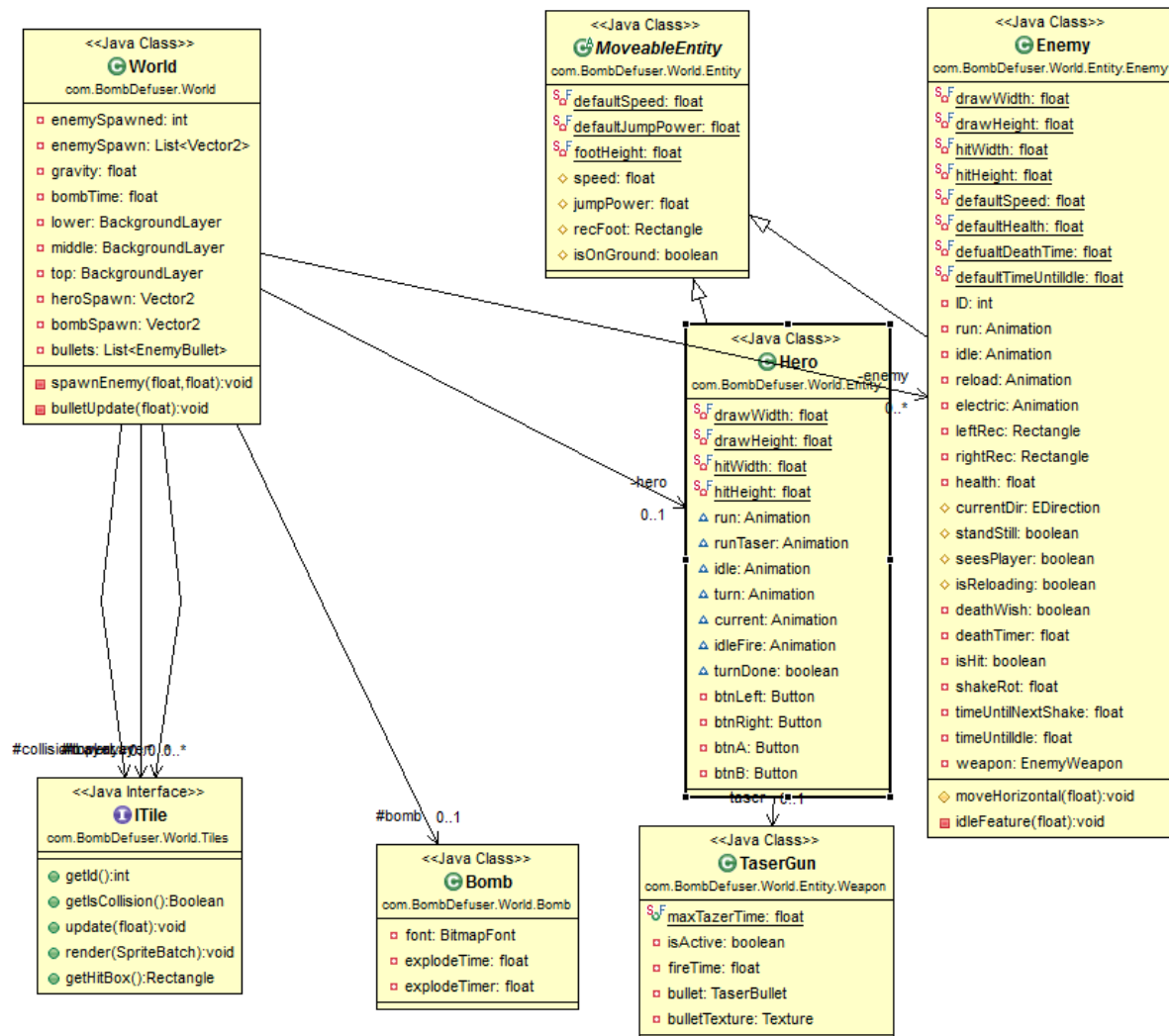
Klassdiagram över ScreenManager som hanterar övergångar av scener. Allt fokus ligger på IScreen som renderas och uppdateras. Varsin scen ärver ifrån en grundscen som heter

BaseScreen.

Klassdiagram  
över  
StateManager  
n.

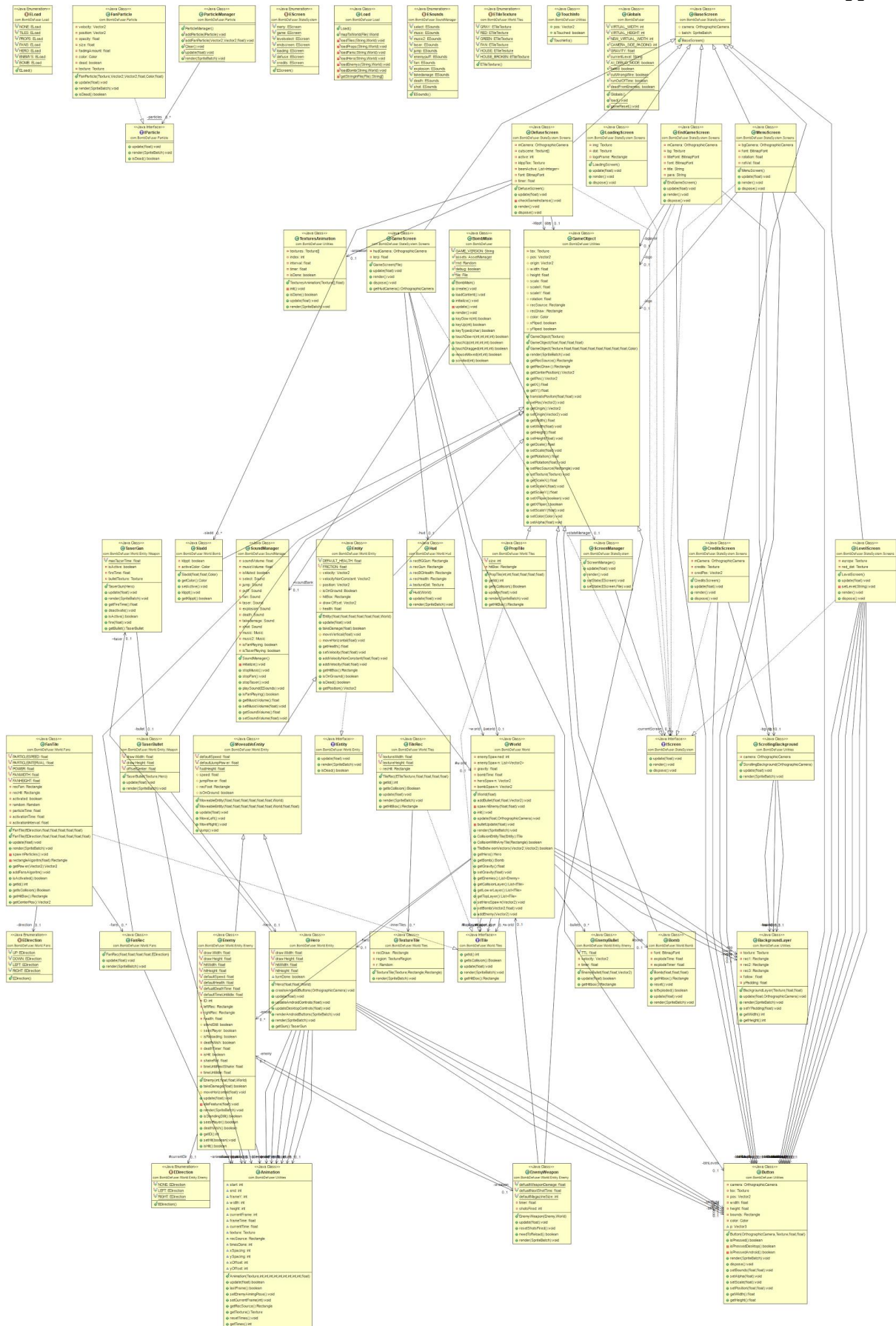
Klassdiagram över värden World hanterar hela värden så som listor på ITile som är rektanglar och kan ses som väggar, golv o tak. Värden innehåller också entiteter som hero (spelaren), fiender etc.





(Notera diagrammet visar bara de största delarna för systemet.)

**Designdokument**  
**Bomb Defuser**  
**Grupp 18**



*Klassdiagram över helheten av spelet <https://www.dropbox.com/s/ql2n3xuqxe10vm1/hel.png?dl=0>*