

TDP005 Projekt: Objektorienterat system

Kravspecifikation

Författare

Albin Dahlén, albda746@student.liu.se
Filip Ingvarsson, filin764@student.liu.se

1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.0	Första utkast	22-11-14

2 Spelidé

Spelet kommer utspela sig i en 2D-värld med ett fågelperspektiv. En spelare ska styra en karaktär som attackeras av fiender. Fienderna kommer försöka döda spelaren och spelaren i sin tur försvarar sig genom att döda fienderna. Karaktären styrs genom piltangenterna. Spelaren kommer använda olika typer av vapen som kan bytas ut och eller uppgraderas under spelets gång. Fienderna kommer anfalla i olika 'vågor' som ökar i storlek och frekvens desto längre spelet har spelats. Det finns olika typer av fienden där de olika typerna har olika hastighet, attackskada, liv och andra specialförmågor. Det finns även olika typer av banor där banans utseende kan försvåra eller förenkla spelet.

Spelet genererar även ett highscore beroende på hur väl man klarat spelet.

3 Målgrupp

Spelet riktar sig till personer som gillar shoot 'em spel och möjligheten att tävla med andra genom highscores.

4 Spelupplevelse

Spelet kommer vara utmanande och har stor återspelbarhet där man kan testa olika vapen och olika uppgraderingar. Spelet kan även spelas om flera gånger för att försöka slå sitt eget eller sina vänners highscore.

5 Spelmekanik

Spelaren kommer röra sig i spelet genom piltangenter, man kan byta dessa knappar i menyn.

Tangentknapp	Åtgärd
→	Flytta spelaren åt höger
↓	Flytta spelaren neråt
←	Flytta spelaren åt vänster
↑	Flytta spelaren uppåt
Mouse button 1	Använd vapen

6 Regler

6.1 Spelplan

- Hinder på spelplanen slumpas fram.
- Spelplanen har en fixerad storlek.

6.2 Spelare

- Spelaren är den figur som användaren styr.
- Spelaren kan använda sitt vapen för att skada fiendeobjekt.

- Spelaren kan röra sig uppåt, neråt, höger och vänster.
- Spelaren startar i mitten av spelplanen när spelet startar.
- Spelaren börjar med 100 i hälsa.
- Spelaren kan kollidera med fiender.
- Spelaren kan kollidera med hinder på spelplanen.
- Spelaren förlorar hälsa vid kollision med en fiende.
- Spelaren kan inte röra sig utanför spelplanen.
- När spelaren har 0 eller minder i hälsa avslutas spelet.

6.3 Fiendetyper

6.3.1 Fiende typ 1

- Rör sig alltid mot spelaren.
- Har 100 i hälsa.
- Skadar spelaren med 10 enheter.
- Rörelsehastighet: mellan.

6.3.2 Fiende typ 2

- Rör sig alltid mot spelaren.
- Har 50 i hälsa.
- Skadar spelaren med 10 enheter.
- Rörelsehastighet: snabb.

6.4 Poäng

Poäng erhålles vid dödandet av fiender och antal avklarade vågor.

6.5 Projektilregler

Projektiler färdas i den riktning som muspekaren är i jämfört med spelaren. Projektilen ska kunna kollidera med fiender. När projektilen kolliderar med en fiende eller når slutet av spelplanen ska projektilen försvinna. Om projektilen kolliderar med en fiende ska fienden ta skada.

7 Visualisering



Figur 1: en prototypbild av spelet

8 Kravformulering

8.1 Ska-krav

1. Det skall finnas en startmeny där spelaren kan välja att spela eller avsluta spelet.
2. Ett spelar objekt ska finnas.
3. Spelaren ska med hjälp av piltangenterna kunna styra spelarobjektet.
4. Det ska finnas 2 olika typer av fiendeobjekt.
5. Fiendeobjektet och spelarobjektet ska kunna förflytta sig.
6. Fiendeobjekten ska förflytta sig mot spelaren.
7. Spelarobjektet och fiendeobjekten ska kunna kollidera med varandra.
8. Om fiendeobjektet kolliderar med spelaren ska spelarens hälsa minska med x enheter beroende på vilken fiende typ det var.
9. Spelaren ska ha ett projektilvapen.
10. Om spelaren träffar fienden med en projektil ska fienden ta skada.
11. När ett fiendeobjekt når 0 eller mindre i hälsa ska objektet dö.
12. Om spelarens hälsa når 0 eller mindre ska spelet avslutas och game over visas.

13. En tvådimensionell spelplan ska ritas ut på skärmen och ska täcka hela spelfönstret.
14. Spelplanens uteseende slumpas fram.
15. Det ska finnas flera instanser av varje fiendeobjekt.
16. När spelaren dödar en fiende ska poäng erhållas.

8.2 Bör-krav

17. Startmenyn ska ha ett alternativ för att ändra keybindings.
18. Vapen ska kunna uppgraderas.
19. När spelaren dödar en fiende ska experience points erhållas.
20. Fiendeobjekten ska ha olika förflyttningsmönster.
21. Det ska finnas olika typer av attacker för fiendeobjekten.

9 Kravuppfyllelse

9.1 Spelet ska simulera en värld som innehåller olika typer av objekt. Objekten ska ha olika beteenden och röra sig i världen och agera på olika sätt när de möter andra objekt.

Uppfylls av krav 2, 3, 4, 5, 6, 7.

9.2 Det måste finnas minst tre olika typer av objekt och det ska finnas flera instanser av minst två av dessa. T.ex ett spelarobjekt och många instanser av två olika slags fiendeobjekt.

Uppfylls av krav 2, 4, 15.

9.3 Ett beteende som måste finnas med är att figurerna ska röra sig över skärmen. Rörelsen kan följa ett mönster och/eller vara slumpmässig. Minst ett objekt, utöver spelaren ska ha någon typ av rörelse.

Uppfylls av krav 3, 5, 6.

9.4 En figur ska styras av spelaren, antingen med tangentbordet eller med musen. Du kan även göra ett spel där man spelar två stycken genom att dela på tangentbordet (varje spelare använder olika tangenter). Då styr man var sin figur.

Uppfylls av krav 3.

9.5 Grafiken ska vara tvådimensionell.

Uppfylls av krav 13.

- 9.6** Världen (spelplanen) kan antas vara lika stor som fönstret (du kan göra en större spelplan med scrollning, men det blir lite krångligare).

Uppfylls av krav 13.

- 9.7** Det ska finnas kollisionshantering, det vill säga, det ska hända olika saker när objekten möter varandra, de ska påverka varandra på något sätt. T.ex kan ett av objekten tas bort, eller så kan objekten förvandlas på något sätt, eller så kan ett nytt objekt skapas. (Ett exempel på att skapa/ta bort objekt är när man i Space Invaders trycker på skjuta-knappen, t.ex en musknapp, då avfyras ett laserskott och skottet blir då en ny figur som skapas och placeras i världen, på en position vid laserkanonens mynning. Skottet rör sig framåt (uppåt) och om det träffar ett fiendeskepp tas både skottet och skeppet bort, om skottet kommer utanför spelplanen, dvs det missar, tas det endast bort.)

Uppfylls av krav 7, 8, 10.

- 9.8** Det ska vara enkelt att lägga till eller ändra banor i spelet. Detta kan exempelvis lösas genom att läsa in banor från en fil (lite som i Sokoban-labben i TDP002), eller genom att ha funktioner i programkoden som bygger upp en datastruktur som definierar en bana.

Uppfylls av krav 14.

- 9.9** Spelet måste upplevas som ett sammanhängande spel som går att spela!

Uppfylls av alla krav.