

# TDP005 Projekt: Objektorienterat system

## Kravspecifikation

Författare

Daniel Huber, [danhu849@student.liu.se](mailto:danhu849@student.liu.se)

Viktor Rösler, [vikro653@student.liu.se](mailto:vikro653@student.liu.se)

## Innehåll

<b>1</b>	<b>Revisionshistorik</b>	<b>2</b>
<b>2</b>	<b>Spelidé</b>	<b>2</b>
2.1	Spelets mål . . . . .	2
<b>3</b>	<b>Målgrupp</b>	<b>2</b>
<b>4</b>	<b>Spelupplevelse</b>	<b>2</b>
<b>5</b>	<b>Spelmekanik</b>	<b>2</b>
5.1	Förflyttning . . . . .	3
5.2	Fiender . . . . .	3
5.3	Vapen . . . . .	3
5.4	Spelarfönstret . . . . .	3
5.5	Spelmeny . . . . .	3
<b>6</b>	<b>Regler</b>	<b>3</b>
6.1	Single player . . . . .	3
6.2	Multiplayer . . . . .	4
6.3	Fiender . . . . .	4
6.4	Power-ups . . . . .	4
6.5	Projektiler . . . . .	4
6.6	Poäng . . . . .	4
6.7	Kollisionshantering . . . . .	4
6.7.1	Projektiler eller fiender som kolliderar med spelaren. . . . .	4
6.7.2	Fiender som kolliderar med annan/andra fiender. . . . .	4
6.7.3	Projektiler som träffar andra projektiler. . . . .	4
6.7.4	Spelaren rör vid lavan. . . . .	4
6.7.5	Projektiler träffar vägg eller plattform. . . . .	4
6.7.6	Spelare plockar upp liv power-up. . . . .	4
<b>7</b>	<b>Visualisering</b>	<b>5</b>
<b>8</b>	<b>Kravformulering</b>	<b>5</b>
8.1	Kravuppfyllelse . . . . .	5
8.1.1	<i>Spelet ska simulera en 2D-värld i realtid. (Måste vara 2-dimensioner)</i> . . . . .	5
8.1.2	<i>Minst 3 typer av objekt. Ska finnas flera instanser av minst två av dessa. Till exempel ett spelarobjekt och många instanser av två olika fiendeobjekt.</i> . . . . .	6
8.1.3	<i>Ett beteende som måste finnas med är att figurerna (Objekten) ska röra sig över skärmen. Rörelsen kan följa mönster och/eller vara slumpmässig. Minst ett objekt utöver spelaren ska ha någon typ av rörelse.</i> . . . . .	6
8.1.4	<i>Ska vara enkelt att modifiera banor i spelet. 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.</i> . . . . .	6
8.1.5	<i>Spelet ska upplevas som sammanhängande spel som går att spela.</i> . . . . .	6
8.1.6	<i>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.</i> . . . . .	6
8.1.7	<i>Världen (spelplanen) kan antas vara lika stor som fönstret (du kan göra en större spelplan med scrollning, men det blir lite krångligare).</i> . . . . .	6

8.1.8	<i>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.</i>	6
8.2	Ska-krav på projektet . . . . .	6
8.3	Bör-krav på projektet . . . . .	7
8.4	Krav på koden . . . . .	7

## 1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.2	Kravspecifikation TDP005	201118
1.1	Modifierad för att stödja xelatex och unicode	150603
1.0	Skapad för studenter att använda som mall till kommande dokumentinlämningar	140908

## 2 Spelidé

Din familj har kidnappats av Ondskan! Du vet om att Ondskan håller till på toppen av berg omgivna av lava, men du vet inte vilket! Klättra upp för bergen genom att hoppa upp på plattformarna, undvik den ökande nivån av lava, döda eller undvik fienderna, samla power-ups och besegra bossen på toppen av berget! I denna bottom-top platformscroller kan du spela ensam eller tillsammans med en vän. Klara banorna snabbare för högre highscore! Kan du rädda din familj i tid?

### 2.1 Spelets mål

Varje nivå avklaras när toppen av nivån nås av spelaren och spelet avslutas när toppen på den sista nivån nås. På vägen upp mot toppen dödas eller undviks fiender.

## 3 Målgrupp

Spelet riktar sig mot spelare som underhålls av spel med en ökande svårighetsgrad för varje nivå och där nivåer måste avklaras under en viss tid.

## 4 Spelupplevelse

Spelaren tvingas i varje nivå hitta den mest tidseffektiva vägen upp samtidigt som fiender undviks eller dödas. Vid spelets avslutande belönas spelaren med en känsla utav att ha klarat av en utmaning då spelet designats att vara svårt.

## 5 Spelmekanik

Nåvåerna startas med spelarkaraktern vid botten av berget med tre liv och en pistol. Efter att första plattformen nås börjar lavan stiga. Genom nivån följs spelaren av spelarfönstret. I takt med att spelaren rör sig uppåt syns fler plattformar spelaren kan hoppa upp till. På plattformar dödas eller undviks fiender av spelaren. Mellan plattformar kan flygande fiender finnas. Spelarens väg upp mot toppen försvåras av att plattformar inte kan hoppas igenom. Game over triggas om spelaren skadas till noll liv. I nivån kan power-ups plockas upp och ge spelaren tillfälliga fördelar eller återställa förlorade liv.

Tabell 1: Tangentbordskommandon

Tangent	Resultat
↑, Z	Spelaren hoppar uppåt.
←	Spelaren förflyttas åt vänster.
→	Spelaren förflyttas åt höger.
↑ + ←, Z + ←	Spelaren hoppar åt vänster.
↑ + →, Z + →	Spelaren hoppar åt höger.
X	Ett skott avfyras i den riktningen spelaren är vänd åt.

## 5.1 Förflyttning

Spelaren styrs med hjälp av tangentbordet enligt tabell:1. I luften styrs spelaren av respektive riktningstangent. Spelare förflyttas i både vänster-, höger- och höjddled med hjälp av hopp. Det finns bara en hopp höjd och denna är oföränderlig. Vid kontinuerlig rörelse åt något håll byggs momentum upp och spelarkarakteren förflyttas ytterligare en liten bit när spelarinput slutas att ges.

## 5.2 Fiender

Över nivåerna förflyttas fiender i förutbestämda mönster. De kan finnas på plattformar eller i luften mellan plattformarna. Det finns tre typer av fiender. En flygande typ, en gående typ och en hoppande typ.

## 5.3 Spelarfönstret

Spelaren följs kontinuerligt av spelarfönstret och den nedersta delen av spelarfönstret upptas till en början av lava. Vid snabb förflyttning upp kan lavan hamna ur bild och vid för långsam förflyttning upptas en allt större del av spelarskärmen av lava tills dess att spelaren träffas och game over visas på skärmen.

## 5.4 Spelmeny

Det första spelaren möts av efter programmets start är spelets meny i form av knappar. Spelaren kan manövrera över knapparna med piltangenterna och välja med mellanslag. Spelaren kan välja mellan en nivåmeny, inställningar för att se tangentkommandon samt att stänga av spelets musik och ljud och ytterligare en knapp med information över skaparna av spelet. ??

# 6 Regler

## 6.1 Single player

Spelaren ges vid varje nivå början tre liv. Dessa representeras av en health-bar i övre delen av spelarfönstret. Vid fullt liv kan extra liv plockas upp, men extra liv utöver max ges inte. Förloras alla liv visas game over i spelarfönstret. Vid kontakt med fiende skadas spelaren ett liv.

## 6.2 Multiplayer

## 6.3 Fiender

Vid fiendens död försvinner fienden. Fiender skadar spelaren ett liv.

## 6.4 Power-ups

Power-ups finns förutbestämt utlagda på plattformarna sedan innan. De är en del av nivån.

## 6.5 Projektiler

Skott som avfyras med spelarens vapen kan bara skjutas åt det håll spelaren är vänd åt, antingen höger eller vänster. Spelaren kan inte skjuta uppåt. Skotten flyger rakt och förflyttas ej i y-led. Avfyras skott från luften av att spelaren hoppar fortsätter projektilen flyga i samma y-led det sköts ifrån. Varje projektil avfyrad från spelaren som träffar en fiende åsamkar ett i skada.

## 6.6 Poäng

Varje dödad fiende i en nivå genererar poäng och en tidsbonus ges ut för snabbt avklarande.

## 6.7 Kollisionshantering

### 6.7.1 Projektiler eller fiender som kolliderar med spelaren.

Spelaren förlorar ett hp. Spelaren blir odödlig i ca 1.618 sekunder. Projektilen försvinner.

### 6.7.2 Fiender som kolliderar med annan/andra fiender.

Fienderna ändrar riktning.

### 6.7.3 Projektiler som träffar andra projektiler.

De passerar varandra och fortsätter sin bana utan att något speciellt händer.

### 6.7.4 Spelaren rör vid lavan.

Spelaren dör och texten game over visas på spelskärmen samt en knapp om att starta om nivån och en annan knapp för att gå tillbaka till nivåmenyn.

### 6.7.5 Projektiler träffar vägg eller plattform.

När projektiler träffar väggar och plattformar eller flyger utanför spelskärmen så försvinner de.

### 6.7.6 Spelare plockar upp liv power-up.

Spelaren ges ett liv i livmätaren om ett liv eller mer saknas. Annars plockas extralivet upp och försvinner från banan men ingenting händer.

## 7 Visualisering



Figur 1: LOFI-exempel för en av spelets nivåer.

## 8 Kravformulering

### 8.1 Kravuppfyllelse

Kraven nedan i *kursivt* är minimikrav som spelet minst måste uppfylla för att godkännas. Nedan specificeras också vilka av våra krav som uppfyller minimikraven.

#### 8.1.1 *Spelet ska simulera en 2D-värld i realtid. (Måste vara 2-dimensioner)*

Uppfylls av ska-krav: 1, 3

**8.1.2** *Minst 3 typer av objekt. Ska finnas flera instanser av minst två av dessa. Till exempel ett spelarobjekt och många instanser av två olika fiendeobjekt.*

Uppfylls av ska-krav: 3, 5, 11

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

Uppfylls av ska-krav: 5, 11

**8.1.4** *Ska vara enkelt att modifiera banor i spelet. 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 ska-krav: 12

**8.1.5** *Spelet ska upplevas som sammanhängande spel som går att spela.*

Uppfylls av ska-krav: Alla

**8.1.6** *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 ska-krav: 3

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

Uppfylls av ska-krav: 1, 4

**8.1.8** *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.*

Uppfylls av ska-krav: 3, 8, 9

## 8.2 Ska-krav på projektet

1. Spelet ska vara en tvådimensionell plattformsscroller.
2. Spelet ska ha en startmeny.
3. Spelaren ska kunna röra en spelarfigur sig i alla vädersträck med piltangenterna och kunna stå och hoppa på plattformar.
4. Spelet ska inkludera scrollande nivåer i y-led.
5. Spelaren ska bli jagad av stigande lava.
6. Nivån ska avslutas när man kommer till toppen.
7. När spelaren dör ska en Game Over skärm visas i spelfönstret.
8. Gravitation ska finnas som drar spelaren mot botten av spelskärmen.

9. Spelaren ska kunna avfyra projektiler som försvinner vid kollision med fiender, väggar eller åker utanför banan.
10. Hårdvarukrav: Spelet ska kunna spelas på skolans datorer.
11. Det ska finnas tre olika typer av fiendeobjekt som har olika utseende. En långsam typ, en flygande typ och en hoppande typ.
12. Nivåer ska kunna läsas in från fil.
13. Det ska finnas en livmätare som visar hur många återstående liv spelaren har.
14. Det ska finnas hp-pickups som återtillför liv till spelaren vid upplockning.
- 15.

### 8.3 Bör-krav på projektet

1. Multiplayer Två spelare kan spela spelet samtidigt. \*
2. - spelare kan vid multiplayer välja om friendly fireska vara på eller av i spelets inställningar.
3. Power-ups plockas upp av spelaren genom att spelaren med sin karaktär går in i samma område som power-up ikonerna upptar på spelskärmen. Det finns power-ups som ger spelaren tillbaka liv, möjligheten att hoppa högre samt en power-up som gör att spelaren kan hoppa högre. \*
4. Bakgrunder skiftar i utseende efterhand att spelaren tar sig till högre höjd. \*
5. Det kan finnas en Level editor som ger spelaren möjlighet att skapa banor.
6. Spelet kan spelas med handkontroll. \*
7. Animerade sprites (Explosionssprites, rörelse) \*
8. Fiender kan avfyra av projektiler. \*
9. Bosstrid, Vid toppen av berget, under striden med bossen slutar lavan att förflytta sig uppåt och stannar kvar vid spelarfönstrets botten.
10. Spelaren kan få highscore på varje nivå.
11. Ljudeffekter kan spelas upp när spelaren tar skada, avfyrrar projektiler och dör. Ljud spelas upp när fiender dör. \*
12. Det finns bakgrundsmusik och musik som spelas i nivåerna. \*

### 8.4 Krav på koden

1. Koden ska följa en accepterad kodstandard. Koden ska gå att kompilera och köra i Ubuntu.
2. Koden ska vara väldokumenterad. Doxygen ska användas.
3. "Information hidingska användas (ingen åtkomst av instansvariabler utanför klassen).
4. Designen ska vara modulär med så få beroenden mellan klasser som möjligt.
5. Det ska finnas en gemensam basklass (Sprite) för alla figurer i programmet. Övriga sprite-figurer ska ärvas från denna klass.
6. Det ska finnas abstrakta metoder, åtminstone för en Sprite-hierarki.



7. Main-funktionen ska vara liten, huvuddelen av spelet ska finnas i de klasser ni designar. Ett exempel på en lämplig main-funktion:  

```
int main() { Game game; game.run(); return 0; }
```
8. Grafiken ska realiseras med hjälp av SFML.
9. Tillsammans med koden ska en Makefile lämnas in. Handledaren ska kunna köra koden i terminalen med hjälp av denna Makefile.