

# TDP005 Projekt: Objektorienterat system

## Dokumentmall

Författare

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

Jens hrnell, [jenoh242@student.liu.se](mailto:jenoh242@student.liu.se)

# Innehåll

## 1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.2	Kravspecifikation TDP005	201101
1.1	Modifierad för att stödja xelatex och unicode	150603
1.0	Skapad för studenter att använda som mall till kommande dokumentinlämningar	140908

## 2 Spelide´

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 platforms scroller kan du spela ensam eller tillsammans med en vän. Klara banorna snabbare för högre highscore. Kan du rädda din familj i tid?

Spelaren/spelarna börjar på botten av berget vid varje nivås början med fullt liv och en pistol. Efter att ha hoppat upp på första platformen börjar lavan stiga och spelarfönstret börjar förflytta sig uppåt. I takt med att spelarfönstret och spelaren rör sig uppåt så uppkommer fler platformar spelaren kan hoppa upp till. Fiender kan finnas och röra sig över platformarna samt skjuta i bestämda tidsintervall eller när spelaren befinner sig på samma höjd som dem. Fiender kan också flyga i utrymmet mellan plattformarna. Spelaren kan inte röra sig utanför skärmen och kan inte hoppa igenom plattformar. Får spelaren noll liv eller rör lavan så är det game over. På plattformarna kan också power-ups finnas som aktiveras när spelaren går i samma ruta som dem. Dessa kan ge antingen permanenta eller temporära fördelar i nivån, men förs inte över till nästa nivå.

### 2.1 Spelets mål

Döda fiender medan du klättrar uppför berget och klättra upp för berget snabbt för en högre andel poäng. Besegra bossarna på toppen av varje berg och rädda din familj i tid.

## 3 Målgrupp

Vårt spel riktar sig till

## 4 Spelupplevelse

## 5 Spelmekanik

### 5.1 Storlek

Allting i spelet upptar lika mycket plats, en spelruta. Detta gäller spelare, fiender, power-ups och enskilda bitar av nivån. Endast bossen är flera spelrutor stor. Det minsta något kan vara är en spelruta.

### 5.2 Förflyttning

Spelaren styrs med hjälp av tangentbord enligt tabellen ?? nedan och kan förflyttas åt höger eller vänster både på marken och i luften. Finns stegen möjliggör detta att spelaren kan förflytta sig rakt upp. Spelare kan

välja att hoppa åt både vänster, höger och rakt upp. Det finns bara en hopphöjd och denna är oföränderlig. Fiender är antingen stationära eller rör sig i förutbestämda mönster. De kan finnas på plattformar eller i luften mellan plattformarna.

Kommandon:

Tangent	Resultat	
W	Spelaren hoppar rakt upp.	
A	Spelaren förflyttas åt vänster.	
D	Spelaren förflyttas åt höger.	
W + A	Spelaren hoppar åt vänster.	
W + D	Spelaren hoppar åt höger.	
K	Ett skott avfyras i den riktningen spelaren är vänd åt.	

## 5.3 Projektiler

Skott som avfyras från pistolen spelaren börjar med kan bara skjutas åt det håll spelaren är vänd åt, antingen höger eller vänster. Spelaren kan inte skjuta uppåt. Skotten flyger rakt i x-led och förflyttas ej i y-led. De exploderar när de träffar en vägg eller fiende. Flyger de utanför skärmen utan att träffa något så försvinner de. Om fiendetyper kan avfyras projektiler gör de så under regelbundna tidsintervall. Dör en fiende efter att ha avfyrat ett skott så fortsätter fiendens skott tills det träffar något eller åker utanför banan. När fiender, spelaren eller väggar träffas av skott visas skottets explosionssprite. Spelaren kan inte skada sig själv med sina egna skott, men spelare kan vid multiplayer välja om friendly fireska vara på eller av.

## 5.4 Power-ups

Power-ups finns förutbestämt utlagda på plattformarna, men kan också släppas av fiender när de dör. Om de släpps av fiender eller inte beror på ett slumptal som genererats för varje fiende vid banans inläsning. För spelaren kommer det framstå som slumpartat, men är egentligen förutbestämt.

## 5.5 Spelarfönstret

Spelarfönstret följer hela tiden spelaren så om spelaren rör sig snabbare uppåt än lavan kan lavan hamna ur bild. Stannar spelaren för länge på samma höjd tar lavan över ett allt större område på spelskärmen.

Den nedersta delen av spelarfönstret upptas av lava. Spelare och fiender som rör vid lavan dör genast. Vid toppen av berget, under striden med bossen slutar spelarfönstret att förflytta sig uppåt. Lavan stannar kvar vid spelarfönstrets botten.

# 6 Regler

## 6.1 Single player

Spelaren har vid varje nivåns början tre liv som finns representerade i en health-bar i övre delen av spelarfönstret. Vid fullt liv kan extra liv plockas upp,

## 6.2 Multiplayer

## 6.3 Fiender

## 6.4 Power-ups

## 6.5 Vapen

## 6.6 Poäng(?)

## 6.7 Kollisionshantering

Kollisionshantering

## 7 Visualisering

## 8 Kravformulering

## 9 Kravuppfyllelse

## 10

## 11 Kravformulering

### 11.1 Minimikrav på projektet

1. Spelet ska simulera en 2D-värld i realtid. (Måste vara 2-dimensioner)
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.
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.
4. Kollisionshantering ska finnas.
5. 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.
6. Spelet ska upplevas som sammanhängande spel som går att spela.
7. 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.
8. Världen (spelplanen) kan antas vara lika stor som fönstret (du kan göra en större spelplan med scrollning, men det blir lite krångligare).
9. 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.)

## 11.2 Bör-krav på projektet

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.

## 11.3 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 ärva från denna klass.
6. Det ska finnas abstrakta metoder, åtminstone för er 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.