

TDP005 Projekt: Objektorienterat system

Kravspecifikation

Författare

Daniel Huber, danhu849@student.liu.se

Viktor Rösler, vikro653@student.liu.se

Innehåll

1	Revisionshistorik	1
2	Spelidé	1
2.1	Spelets mål	2
3	Målgrupp	2
4	Spelupplevelse	2
5	Spelmekanik	2
5.1	Förflyttning	2
5.2	Fiender	2
5.2.1	Flygande fiende 1	3
5.2.2	Gående fiende 1	3
5.2.3	Hoppande fiende 1	3
5.3	Spelarfönstret	3
5.4	Spelmeny	4
6	Regler	4
6.1	Single player	4
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 träffas av andra projektiler.	5
6.7.4	Lavan rörs av spelaren.	5
6.7.5	Vägg eller plattform träffas av projektiler.	5
6.7.6	Liv power-up plockas upp av spelaren.	5
7	Visualisering	6
7.1	Ska-krav på projektet	6
7.2	Bör-krav på projektet	7
7.3	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 ta dig uppför berget så snabbt som möjligt! I denna bottom-top plattformsscroller 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 avklaras när toppen på den sista nivån nås. På vägen upp mot toppen dödas eller undviks fiender.

3 Målgrupp

Spelet riktas 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 avklarande 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.

5.1 Förflyttning

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.

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 spelarkaraktern förflyttas ytterligare en liten bit när spelarinput slutas att ges.

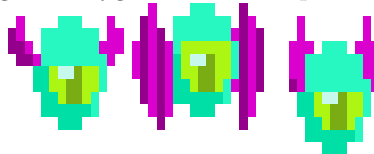
5.2 Fiender

Över nivåerna förflyttas fiender i förutbestämda mönster. De kan finnas på plattformar eller i luften mellan plattformarna. Det ges ingen visuell representation i spelet av hur många återstående liv en fiende har. Tre

typer av fiender finns.

5.2.1 Flygande fiende 1

Figur 1: Flygande fiende 1 sprite-sheet



Tabell 2: Egenskaper: Flygande fiende 1

Attribut	Värde
Liv	1
Skada	1
Speciell förmåga	Är inte bunden till plattformar.

5.2.2 Gående fiende 1



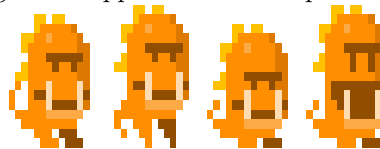
Figur 2: Gående fiende 1 sprite-sheet

Tabell 3: Egenskaper: Gående fiende 1

Attribut	Värde
Liv	3
Skada	1
Speciell förmåga	Har mer liv.

5.2.3 Hoppande fiende 1

Figur 3: Hoppande fiende 1 sprite-sheet



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.

Tabell 4: Egenskaper: Hoppande fiende 1

Attribut	Värde
Liv	2
Skada	1
Speciell förmåga	Kan hoppa och skjuta projektiler.

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. 4

6 Regler

6.1 Single player

Spelaren ges vid varje nivåns 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 laddas in med nivån.

6.5 Projektiler

Skott som avfyras med spelarens vapen kan bara skjutas i den riktning spelarkaraktern är vänd åt, antingen höger eller vänster. Projektiler kan inte avfyras uppåt. Skotten färdas alltid i samma höjdled de sköts ifrån så om projektiler avfyras i luften fortsätter projektilen i samma y-led till kollision. Fiender träffade av spelarens projektiler åsamkas ett i skada.

6.6 Poäng

Poäng ges vid varje dödad fiende och snabbt avklarande av nivån.

6.7 Kollisionshantering

6.7.1 Projektiler eller fiender som kolliderar med spelaren.

Ett hp av spelarens liv förloras och spelaren ges odödlighet i 1.618 sekunder. Projektilen upplöses.

6.7.2 Fiender som kolliderar med annan/andra fiender.

Riktningen på fiendernas banor ändras till motsatt riktning.

6.7.3 Projektiler träffas av andra projektiler.

Projektilernas bana fortsätts utan att något speciellt händer.

6.7.4 Lavan rörs av spelaren.

Spelkaraktären dödas, texten game oversamt en knapp om att starta om nivån och en knapp till nivåmenyn visas på spelskärmen.

6.7.5 Vagg eller plattform träffas av projektiler.

När väggar och plattformar träffas av projektiler upplöses projektilerna.

6.7.6 Liv power-up plockas upp av spelaren.

Spelaren ges ett liv i livmätaren om ett liv eller mer saknas.

7 Visualisering

Figur 4: LOFI-exempel Nivå start.



I övre högra hörnet representeras spelarens återstående liv av gröna kuber i en health-bar. Varje nivå 4 startas med spelkaraktären i nedre delen av spelarfönstret alldeles ovanför lavan. En eller flera utstakade vägar av plattformar i olika höjd placeras över nivån.

7.1 Ska-krav på projektet

1. Spelet ska vara en tvådimensionell plattformsscroller där nivån konsekutivt visas desto högre upp spelaren förflyttas.
2. Två spelare ska kunna spela spelet samtidigt. Spelkaraktärerna ha olika färg.
3. Spelet ska ha en startmeny, med möjlighet att ta sig till en nivåmeny, en inställningsmeny genom att

klicka på knappar.

4. Spelaren ska kunna röra en spelarfigur i alla vädersträck med piltangenterna och kunna stå och hoppa på plattformar.
5. Spelaren ska efter första hoppet börja bli jagad av stigande lava. Nudas lavan av spelaren ska spelaren dö.
6. Nivån ska avslutas när toppen nås av spelaren eller spelaren avlider.
7. Vid spelarens död ska en Game Over vy visas i spelfönstret.
8. Spelaren ska i luften efter maximal hopphöjd nåtts alltid föras nedåt mot botten av spelskärmen.
9. Projektiler ska kunna avfyras mot den sida spelaren är vänd åt från spelarens position.
10. Projektilerna ska vid kollision med fiender eller väggar upplösas.
11. Spelet ska kunna spelas på datorer.
12. Det ska finnas tre olika typer av fiendeobjekt som har olika utseende. En flygande typ 1, en hoppande typ 2 och en gående typ3.
13. Det ska finnas en livmätare som visar hur många återstående liv spelaren har.
14. Spelaren ska kunna plocka upp power-ups genom att gå in i samma område som power-up ikonerna upptar på spelskärmen. Det ska finnas hp-power-ups som återtillför liv till spelaren??. Liv-power-ups plockas upp även om spelaren har fullt liv.

7.2 Bör-krav på projektet

1. Spelare ska vid multiplayer kunna välja om friendly fireska vara på eller av i spelets inställningar.
2. Det ska finnas power-ups som ger spelaren möjligheten att hoppa högre och en power-up som dödar alla synliga fiender på skärmen. *
3. Spelet ska också kunna spelas med handkontroll. *
4. Fienders död, spelarens rörelse och projektilers kollision ska animeras med hjälp av sprites.
5. Fiender ska kunna avfyras projektiler som färdas i rak bana. *
6. I slutet på banan ska det finnas en boss. Vid toppen av berget, under striden med bossen slutar lavan att förflytta sig uppåt och stannar kvar vid spelarfönstrets botten.
7. Ljudeffekter ska spelas upp när spelaren skadas, när spelarens projektiler avfyras, när spelare dör samt när fiender dör.
8. Spelet ska spela bakgrundsmusik.
9. Spelaren ska kunna välja att ändra om sina tangentbordskommandon i spelets inställningar.

7.3 Krav på koden

1. Koden ska följa en accepterad kodstandard. Spelkoden ska gå att kompilera och köra i Ubuntu 20.04.
2. Koden ska vara väldokumenterad. Doxygen ska användas.
3. "Information hidingska användas (ingen åtkomst av instansvariabler utanför klassen).
4. Designen ska vara modular 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.