

Björnligan

Kravspecifikation

Project TDDI02

Skapades datum: 11-09-2014

Senast ändrad: 18-09-2014

Introduktion

Denna kravspecifikation är ett dokument som ska exakt förklara hur Björnligans projekt i programmering 2014 har planerats och skall utföras. Detta dokument kommer agera som underlag vid vidare beslutstagning då projektet börjar utvecklas.

Definitioner och förkortningar

I denna kravspecifikation använda följande definitioner:

- Bana/spelplan/karta/map/level – Syftar på den yta som spelaren kommer att röra sig på.
- Spelare – Syftar på den karaktär användaren kommer kontrollerar på spelplanen.
- Plattform – Syftar på terräng på spelplanen.
- Kollision – Syftar på då objekt i spelet krockar.
- Fiende – Syftar på de datorkontrollerade karaktärer som jobbar för att döda spelaren.
- Liv/dör – Syftar på spelarens tillstånd i spelet. Spelaren dör då den träffas av t.ex. en projektil från en fiende.
- Hjälpmedel – Syftar på föremål på kartan som ger speciella effekter när spelaren plockar upp dem. T.ex. hjärta ger liv och skor ökar hastigheten.

Projektets syfte

Målet med detta projekt är att skapa ett fungerande spel i samma anda som de gamla Super Mario-spelen. Då alla inblandade i projektet spelat dessa spel då de var yngre ansågs det som ett bra val att få förnya detta spel i vår egen bild.

Målen med att göra detta spel är:

- Få en djupare förståelse för C++.
- Få en större bild av hur det är att jobba inom mjukvara-utveckling.
- Lära oss jobba mot ett gemensamt mål som projektgrupp.
- Lära oss om processen för projektutveckling.

Krav

Skall-krav

S1. Fysikmotor (Box2D)

S1.1. Spelaren skall kunna röra sig framåt, bakåt, hoppa.

S1.2. Kollision skall existera i spelet. Detta innebär att kollision med plattformar, fiender etc. skall hanteras av motorn.

S2. Det ska finnas olika typer av objekt på kartan

S2.1. Fiender som patrullerar

S2.2. Hinder

S2.3. Hjälpmedel som underlättar för spelaren

S3. Level-editor. (Level-skapare)

S3.1. Ett separat läge då användaren skapar egna banor.

S3.2. Spelaren skall kunna placera ut objekt på kartan, så som fiender, hinder, hjälpmedel och plattformar.

S3.3. Banorna skall kunna sparas och laddas från en separat fil med eget filformat.

S4. Ett användargränssnitt som innehåller:

S4.1. Startmeny där man väljer karta att spela

S4.2. Pausmeny.

S4.3. Level-editor.

S5. Då spelaren ”dör” i spelet så skall banan laddas om från start. Spelaren har endast 1 liv.

S6. En komplett och spelbar bana skall finnas från spelstart.

S7. Bakgrundsmusik skall finnas från spelstart.

S8. Spelet ska vara presenteras i 2D grafik med hjälp av grafikbiblioteket SDL.

S9. Spelet ska ha en smart kamera som följer spelarens rörelse utan att spelaren behöver styra denna.

S10. Spelobjekt med olika egenskaper skall finnas på banan för att ge spelaren hjälpmedel/hinder.

S11. När spelaren klarar en karta låses en ny karta upp.

Bör-krav

- B1. High-scores ska sparas efter avslutat spelsession.
- B2. Fiender ska få mer funktionalitet.
 - B2.1. Fiender ska kunna skjuta projektiler.
 - B2.2. Fiender ska kunna ha mer komplex rörlighet.
- B3. Tillägg av ljudeffekter.
 - B3.1. Ljudeffekt vid projektiler.
 - B3.2. Ljudeffekt vid kollisioner.
- B4. Lägga till fler funktioner för spelaren.
 - B4.1. Spelaren bör kunna ducka inuti spelet.
 - B4.2. Spelaren bör kunna använda vapen inuti spelet.
- B5. Destruktion av objekt (hoppa sönder plattformar underifrån).
- B6. Flytta runt vissa objekt (lådor och liknande).
- B7. Fusk-koder.
- B8. Plattformar som automatiskt rör på sig.
- B9. Aggressiva fiender som jagar spelaren.
- B10. Stegar som man klättrar vertikalt på.
- B11. Trampoliner.

Osäkerheter och risker

1. Inte lyckas integrera de externa biblioteken på det sättet som planerat.
2. Brist på tid till att implementera alla funktioner.
3. En dåligt strukturerad kod bas som gör det svårt att bygga vidare på den.
4. Att spelet inte kan köras tillräckligt snabbt på alla maskiner för att få det spelbart.