

# CPROG Rapport för Programmeringsprojektet

[Gruppnummer: 33 ]

[Gruppmedlemmar: Amanda Sjöberg 980603-8346, Jessica Borg 980503-0120]

## 1. Beskrivning

**En spelare förväntas göra följande i spelet:**

I detta spel ska spelaren använda tangentbordet (höger och vänster piltangenter + SPACE) för att styra spelobjektets rörelser i en värld där skärmen hela tiden scrollar vertikalt eller horisontellt. Plattformarna scrollar tillsammans med spelet vilket gör att du som spelare hela tiden måste vara alert på plattformarnas rörelse.

Uppgiften i spelet är att hela tiden hålla sig vid liv genom att hoppa mellan plattformarna och undvika de farliga fienderna som dyker upp på plattformarna då och då. Genom att hoppa på fiendernas huvud kan du döda dem utan att skada dig själv. Om du hoppar på dem från sidan så dör de men du förlorar ett liv! Rör du vid en fiende på annat sätt förlorar du ett liv och när dina liv är slut förlorar du spelet. Om du hamnar för långt utanför spelfönstret förlorar du också.

Så var redo för ett äventyr! Var beredd på att kämpa dig genom en svår och farlig värld där varje hopp är en kamp för överlevnad...

## 2. Instruktion för att bygga och testa

**För att bygga och testa detta programmeringsprojekt, måste följande krav vara uppfyllda:**

- Version av C++: C++17
- En installation av SDL- och SDL2-bibliotek på datorn, inklusive alla nödvändiga komponenter (SDL2, SDL2\_image, SDL2\_ttf)
- Projektet är skapat i VSCode med kompilatormiljö GCC(g++ standard C++17) men bör kunna köras i valfri programmeringsmiljö
- Alla h-filer ska ligga i mappen include och alla cpp-filer i mappen src
- Alla bilder/font/ljud ska läggas i motsvarande mapp i resources.  
När resources används behövs endast namnet (ej sökväg)
- Objekt skapas med en statisk getInstance()-metod

- Börja med att skapa ett Session-objekt (Session ses;)
- Skapa en Actor med getInstance() för att representera spelaren.
- Kalla på initPlatforms("bildnamn") och initEnemies("bildnamn") för att skapa plattformar och fiender
- Skicka med bildens namn (ej sökvägen)
- Lägg till alla skapade objekt med metoden add()
- add(Actor) ska göras efter initPlatforms() och initEnemies()
- Skapa inte Background med getInstance(), använd istället setBackground("bildnamn")
- Default är vertikal scroll, om man vill ändra detta så används set\_scroll\_horizontal(true)

#### **För att bygga projektet:**

1. Öppna projektet i den valda programmeringsmiljön.
2. Skriv in kommandot: "make" i terminalen, då utförs det som står i Makefile
3. Skriv in kommandot: "./build/debug/play.exe (BUILD\_DIR/OBJ\_NAME i Makefile)

### 3. Krav på den Generella Delen(Spelmotorn)

3.1. [ Ja/Nej/Delvis ] Programmet kodas i C++ och grafikbiblioteket SDL2 används.

**Kommentar:** Ja.

3.2. [ Ja/Nej/Delvis ] Objektorienterad programmering används, dvs. programmet är uppdelat i klasser och använder av oo-tekniker som inkapsling, arv och polymorfism.

**Kommentar:** Ja.

3.3. [ Ja/Nej/Delvis ] Tillämpningsprogrammeraren skyddas mot att använda värdesemantik för objekt av polymorfa klasser.

**Kommentar:** Ja, de har statiska getInstance()-metoder.

3.4. [ Ja/Nej/Delvis ] Det finns en gemensam basklass för alla figurer(rörliga objekt), och denna basklass är förberedd för att vara en rotclass i en klasshierarki.

**Kommentar:** Ja, rotklassen Component förbjuder både tilldelning och kopiering, samt att destruktorn deklarerats som virtuell.

3.5.[ Ja/Nej/Delvis ] Inkapsling: datamedlemmar är privata, om inte ange skäl.

**Kommentar:** Ja.

3.6. [ Ja/Nej/Delvis ] Det finns inte något minnesläckage, dvs. jag har testat och sett till att dynamiskt allokerat minne städas bort.

**Kommentar:** Delvis. Vi vet att reset()-metoden i Session orsakar minnesläckage. Vi har försökt att lösa det men hann inte helt innan deadline, så vi återgick till en tidigare version som i alla fall fungerade utan att krasha.

3.7. [ Ja/Nej/Delvis ] Spelmotorn kan ta emot input (tangentbordshändelser, mushändelser) och reagera på dem enligt tillämpningsprogrammets önskemål, eller vidarebefordra dem till tillämpningens objekt.

**Kommentar:** Ja. I rotklassen deklarerar mushändelser- och tangentbordshändelse-metoder som virtuella.

3.8. [ Ja/Nej/Delvis ] Spelmotorn har stöd för kollisionsdetektering: dvs. det går att kolla om en Sprite har kolliderat med en annan Sprite.

**Kommentar:** Ja, Collision::collision()-metoden tar in två Component-objekt och kontrollerar om rektanglarna överlappar varandra på något vis.

3.9. [ Ja/Nej/Delvis ] Programmet är kompilerbart och körbart på en dator under både Mac, Linux och MS Windows (alltså inga plattformspecifika konstruktioner) med SDL 2 och SDL2\_ttf, SDL2\_image och SDL2\_mixer.

**Kommentar:** Ja.

## 4. Krav på den Specifika Delen(Spelet som använder sig av Spelmotorn)

4.1. [ Ja/Nej/Delvis ] Spelet simulerar en värld som innehåller olika typer av visuella objekt. Objekten har olika beteenden och rör sig i världen och agerar på olika sätt när de möter andra objekt.

**Kommentar:** Ja. De flesta objekten scrollar. En Enemy rör sig fram och tillbaka på plattformen. Plattformarna genereras på slumpmässiga positioner

inom spelfönstret.

4.2. [ Ja/Nej/Delvis ] Det finns minst två olika typer av objekt, och det finns flera instanser av minst ett av dessa objekt.

**Kommentar:** Ja. Actor, Platform(flera), Enemy(flera).

4.3. [ Ja/Nej/Delvis ] Figurerna kan röra sig över skärmen.

**Kommentar:** Ja. Actor kan kontrolleras av spelaren.

4.4. [ Ja/Nej/Delvis ] Världen (spelplanen) är tillräckligt stor för att den som spelar skall uppleva att figurerna förflyttar sig i världen.

Kommentar: Ja.

4.5. [ Ja/Nej/Delvis ] En spelare kan styra en figur, med tangentbordet eller med musen.

**Kommentar:** Ja.

4.6. [ Ja/Nej/Delvis ] Det händer olika saker när objekten möter varandra, de påverkar varandra på något sätt.

**Kommentar:** Ja. En Actor kan stå på en plattform och kan döda en Enemy genom att hoppa på den (från något håll). Om en Actor står sida vid sida med en Enemy så kan Enemy knuffa Actor. Antal liv visas på skärmen och när Actor blir skadad av en Enemy så ändras siffran som visar antal liv. Actor kan bli skadad av Enemy vid kontakt annat än att hoppa på dem.