

# CPROG Rapport för Programmeringsprojektet

[Gruppnummer: ??? ]

[Gruppmedlemmar: ??? (namn och personnummer för varje medlem)]

*Skriv en kortfattad instruktion för hur programmeringsprojektet skall byggas och testas, vilka krav som måste vara uppfyllda, sökvägar till resursfiler(bildfiler/ljudfiler/typsnitt mm), samt vad en spelare förväntas göra i spelet, hur figurernas rörelser kontrolleras, mm.  
Om avsteg gjorts från kraven på Filstruktur, så måste också detta motiveras och beskrivas i rapporten.*

*Fyll i 'check-listan', så att du visar att du tagit hänsyn till respektive krav, skriv också en kort kommentar om på vilket sätt du/gruppen anser att kravet tillgodosätts, och/eller var i koden kravet uppfylls.*

*Den ifyllda Rapportmallen lämnas in tillsammans med Programmeringsprojektet. Spara rapporten som en PDF med namnet CPROG\_RAPPORT\_GRUPP\_NR.pdf (där NR är gruppnumret).*

## 1. Beskrivning

Skriv beskrivningen här...

## 2. Instruktion för att bygga och testa

Skriv din kommentar här...

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

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

Kommentar: skriv din kommentar här...

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: skriv din kommentar här...

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

Kommentar: skriv din kommentar här...

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

Kommentar: skriv din kommentar här...

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

Kommentar: skriv din kommentar här...

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

Kommentar: skriv din kommentar här...

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: skriv din kommentar här...

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

Kommentar: skriv din kommentar här...

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: skriv din kommentar här...

#### 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: skriv din kommentar här...

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: skriv din kommentar här...

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

Kommentar: skriv din kommentar här...

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

Kommentar: skriv din kommentar här...

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

Kommentar: skriv din kommentar här...

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

Kommentar: skriv din kommentar här...