

- **Krav på den Generella Delen(Spelmotorn)**

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

Kommentar: Ja, Filerna är utav .cpp och .h vilket är C++ filer samt i koden syns det tydligt att grafiska användargränssnittet har byggts med hjälp av SDL.

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

Kommentar: Ja, Sprite klassen är byggt som en rotklass, och det finns även subklasser "EnemyCharacter" och "PlayerCharacter".

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

Kommentar: Ja, med hjälp av Static getInstance metoder.

[ 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: Ja, min sprite klass fungerar så.

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

Kommentar: Ja, all data ligger inom "private:" delen av klassen

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

Kommentar: Ja, det finns implementerade destruktorer för de klasser som dynamisk allokerar minne.

[ 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, finns att se i min GameEngine.cpp fil genom en switch sats

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

Kommentar: Ja, genom medlemsfunktion isCollided som finns i GameEnginge klassen

[ Ja/Nej/Delvis ] Programmet är kompillerbart 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, inga plattformspecifika konstruktioner har gjorts.

- **Krav på den Specifika Delen(Spelet som använder sig av Spelmotorn)**

[ 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, finns att läsa mer om i beskrivningen

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

Kommentar: Ja, en player object och flera enemy object.

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

Kommentar: JA, enemy objekten åker neråt i banan och player objektet kan flyttas i sidled med tangentnedtryckningar.

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

Kommentar: Ja, förflyttning bland objekten syns tydligt under spelets omgång.

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

Kommentar: Ja, PlayerCharacter går att styra med tangentnedtryckningar.

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

Kommentar: Ja, vid kollision försvinner objekt.