

CPROG Rapport för Programmeringsprojektet

[Gruppnummer: 11]

[Gruppmedlemmar:

Alexander Jaxgård: 20031111-4078

Elias Bennaceur: 20001125-5437

1. Beskrivning

Vårt spel vi har skapat med denna game engine är målet att skjuta ner utomjordingar i sitt rymdskepp som man kan styra. Träffar en enemy dig så är det game over, exakt som Space Invaders.

2. Instruktion för att bygga och testa

Länk till pdf instruktioner till att bygga och testa -

https://drive.google.com/file/d/1awUe6mkFsBRBirUz0ZPE9rJ_t-T_KwaZ/view?usp=drive_link

3. Krav på den Generella Delen(Spelmotorn)

- 3.1. [Ja/Nej/Delvis] Programmet kodas i C++ och grafikbiblioteket SDL2 används.
Kommentar: allt som behöver det använder sig av SDL2
- 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: allting har ett eget klass och använder oo tekniker
- 3.3. [Ja/Nej/Delvis] Tillämpningsprogrammeraren skyddas mot att använda värdesemantik för objekt av polymorfa klasser.
Kommentar: Genom virtuella destruktorer och att se till att kopiering av objekt inte tillåts uppfylls detta krav.
- 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: vår "Component" klass uppfyller detta kravet.
- 3.5. [Ja/Nej/Delvis] Inkapsling: datamedlemmar är privata, om inte ange skäl.
Kommentar: alla datamedlemmar som ska vara privata är det,
- 3.6. [Ja/Nej/Delvis] Det finns inte något minnesläckage, dvs. jag har testat och sett till att dynamiskt allokerat minne städas bort.

Kommentar: Allt som bör försvinna anropar destruktorn.

- 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: dessa metoder visas i vår "Component" klass och kan överwriteas av alla subklasser.

- 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: Vi kan kolla kollisioner baserade på vilket tag ett objekt har fått.

- 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: Vi har testat att köra kod på både mac och windows och det borde också funka på linux

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: Vi har ett objekt för spelaren, objekt för missiler och objekt för enemies som rör sig på olika sätt.

- 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: där finns flera "enemies" som attackerar spelaren åt gången och där kan vara flera "Missiles" åt gången också.

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

Kommentar: Alla objekt vi har rör sig.

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

Kommentar: Spelplanen är 700X600px och man kan se tydligt att karaktären rör sig.

- 4.5. [Ja/Nej/Delvis] En spelare kan styra en figur, med tangentbordet eller med musen.
Kommentar: Vi har testat att det går att röra på våran karaktär med musen men vi tog bort det för att vi tyckte det passade bättre att använda pilarna.
- 4.6. [Ja/Nej/Delvis] Det händer olika saker när objekten möter varandra, de påverkar varandra på något sätt.
Kommentar: När ett "Missile" objekt kolliderar med en "Enemy" objekt så spelar en explosion-animation och ett ljud spelas.