

CPROG Rapport för Programmeringprojektet
Gruppnummer: 13
Gruppmedlemmar: Mattias Larsson 19970113-2996

Beskrivning

Spelaren är en blå kvadrat som ska fånga krabborna som rör sig sidledes, detta görs med WASD på tangentbordet. Det finns väggar för att demonstrera kollision och man kan inte gå utanför banan. Figuren kontrolleras via "EnemyBehaviors" component objekt som man kan lägga till på objekt i scenen.

OBS. för att starta spel behöver man använda ./build/debug/play i terminalen IDE, som det är upplagt i projektmallen från ilearn.

Instruktion för att bygga och testa

Man kan använda "make" i terminalen som bygger spelet, sedan kan man använda ./build/debug/play för att testa. Alla bilder finns i resources\images.

Krav på den Generella Delen(Spelmotorn)

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

Ja

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

Ja. EntityComponentSystem använder sig av arv. Polymorfism används på flera sätt, t.ex. EnemyBehaviors kan ta emot olika koordinater för att få objekt att röra sig i spelet.

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

Ja, men jag kan ha brutit mot vissa oo regler. Vissa variabler publika, som x och y värdena Vector2D klassen, där man kan ändra i vektorn från andra klasser. Detta är för att underlätta förändringar i velocity/position via transform component, utan gets/sets.

Tillämpningsprogrammeraren har inte tillgång till något orimligt, men jag vet inte om jag följt alla regler perfekt, men jag har försökt.

[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.

Ja, "Entity" klassen i EntityComponentSystem filen

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

Ja, jag anger anledningen ifall vissa datamedlemmar är publika via kommentarer i koden.

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

Ja

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

Ja, WASD tangenterna används

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

Ja, via Collider.h

[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.

Jag kan inte testa på Mac och Linux men jag ser inte någon anledning att det inte ska fungera.

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.

Ja, det finns krabbor som rör sig och väggar som är statiska

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

Ja, det finns tre väggar och tre krabbor.

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

Ja, via EnemyBehaviors.h

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

Ja, man kan även ändra spelvärldens storlek i GameSettings.h

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

Ja man kan styra den blåa kvadrat

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

Ja, man kan vinna spelet genom att fånga alla tre krabbor. En krabba förstörs av att spelaren går in i den. Man kan kollidera med väggar.