

CPROG Rapport för Programmeringsprojektet

[Gruppnummer: 75]

[Gruppmedlemmar: Simon Hessling Oscarson 19980829-0077

vad en spelare förväntas göra i spelet:

målet för spelaren är att ta de stillastående figurerna som finns på kartan. Dessa kallas Nests. Varje Nest ger en poäng. Det finns även örnar som flyger emot spelaren. Blir spelaren träffad av ett av dessa sätts poängen till noll och spelaren får börja om från platsen den befinner sig.

Inget annat än spelaren själv kan kontrolleras utav spelaren.

hur programmeringsprojektet skall byggas och testas:

1. Högerklicka på "Makefile".
2. Klicka "open in integrated terminal"
3. skriv "make" i terminalen, klicka på enter.
4. när den har laddat klart skriv "./build/debug/play" i terminalen, klicka på enter.

hur figurernas rörelser kontrolleras:

Spelaren kontrolleras genom W A S D.

Avsteg som gjorts från filstrukturen:

istället för resources har jag en assets folder och ingen filstruktur inom den. Det som finns där i är textfiler för att bygga maps och bilder. Anledningen till detta är för att jag inte ville ha bilder i två olika mappar. Istället la jag textobjekt i assets.

sökvägar:

Resurs filer

[project/assets/proj.png](#)

[project/assets/EagleEnemy.png](#)

[project/assets/SeagulPlayer.png](#)

[project/assets/terrain_ss.png](#)

[project/assets/map.map](#)

Header filer

project/include/ECS/Animation.h
project/include/ECS/ColliderComponent.h
project/include/ECS/Components.h
project/include/ECS/ECS.h
project/include/ECS/KeyboardController.h
project/include/ECS/ProjectileComponent.h
project/include/ECS/SpriteComponent.h
project/include/ECS/TileComponent.h
project/include/ECS/TransformComponent.h
project/include/AssetManager.h
project/include/Collision.h
project/include/Game.h
project/include/GameObject.h
project/include/Map.h
project/include/TextureManager.h
project/include/Vector2D.h

C++ filer

project/src/AssetManager.cpp
project/src/Collision.cpp
project/src/Game.cpp
project/src/GameObject.cpp
project/src/Map.cpp
project/src/TextureManage.cpp
project/src/Vector2D.cpp
project/src/main.cpp
project/src/ECS.cpp

Makefil

project/Makefile/

Övriga

project/.vscode/
project/build/debug/play
project/doc/CPROG_RAPPORT_GRUPP_75

- Krav på den Generella Delen(Spelmotorn)

- [Ja/Nej/Delvis] Programmet kodas i C++ och grafikbiblioteket SDL2 används.
Kommentar: Ja
Jag har använt SDL2 och C++
- [Ja/Nej/Delvis] Objektorienterad programmering används, dvs. programmet är uppdelat i klasser och använder av oo-tekniker som inkapsling, arv och polymorfism.
Kommentar: Ja
Programmet är gjort objektorienterat. Programmet använder sig av ett ECS för att komma åt olika "Components" som gör olika saker som man sedan kallar på.
- [Ja/Nej/Delvis] Tillämpningsprogrammeraren skyddas mot att använda värdesemantik för objekt av polymorfa klasser.
Kommentar: Ja
- [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
I Game() klassen. Där kallar man på alla figurer. Man kan även göra detta i AssetManager() men dem spawnar först i Game().
- [Ja/Nej/Delvis] Inkapsling: datamedlemmar är privata, om inte ange skäl.
Kommentar: Ja
- [Ja/Nej/Delvis] Det finns inte något minnesläckage, dvs. jag har testat och sett till att dynamiskt allokerat minne städas bort.
Kommentar: Ja
- [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
- [Ja/Nej/Delvis] Spelmotorn har stöd för kollisionsdetektering: dvs. det går att kolla om en Sprite har kolliderat med en annan Sprite.
Kommentar: Ja

Går spelaren mot en kant så tar det stopp.

- [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

Inte gjort något plattformsspecifikt.

- **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. Objekt har olika beteenden och rör sig i världen och agerar på olika sätt när de möter andra objekt.

Kommentar: Ja

Du har bakgrund spelare och andra objekt som rör sig.

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

Kommentar: Ja

Det finns i nuläget 3 olika typer av objekt

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

Kommentar: Ja

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

Kommentar: Ja

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

Kommentar: Ja

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

Kommentar: Ja