DATX11-23-57

'Entity Component System'-baserad spel- och simuleringsmotor

Bakgrund

En spelmotor, eller en simuleringsmotor, är vad man kallar ett program som är grunden till en spelimplementation eller en simuleringsimplementation. Motorn implementerar de mest grundläggande beståndsdelar såsom rendering, fysikhantering och ihopkoppling av komponenter.

Spel- och simuleringsmotorer har länge varit skrivna på ett imperativt och/eller objektorienterat vis. År 2007 populariserades ett annat sätt att strukturera motorer: *Entity Component System* (ECS), vilket istället är dataorienterat. Målet med ECS är att bättre utnyttja flerkärniga processorer. Ett välkänt exempel på en implementation av ECS är i spelmotorn Unity.

ECS går ut på att dela upp ett program i tre olika beståndsdelar:

- **Entities**, som inte innehåller någon data och representeras endast av ett ID och en samling *components*. Exempelvis en spelare.
- **Components**, som är dataobjekt utan någon funktionalitet. Exempelvis en komponent Position som innehåller x-, y- och z-koordinater.
- **Systems**, som utför beräkningar på *components*. Exempelvis ett MovementSystem som uppdaterar en komponent Position baserat på datan i en annan komponent Velocity.

Tack vare den här uppdelningen blir det mycket effektivare för processorer att lagra komponentdata i cacheminne, och parallelisering av systemexekvering blir möjligt.

Projektbeskrivning

Projektet går ut på att designa och implementera en parallell ECS-motor.

Fokus med projektet ligger i att utveckla det schemaläggningssystem som kommer att krävas för att effektivt kunna parallelisera arbetet som utförs av motorn. Det finns många olika aspekter att tänka på här, exempelvis vilka system som beror på samma komponenter och på vilket sätt. Det kan finnas system som bara läser från komponenter och andra som skriver. Schemaläggning av processer är ett välbeforskat problem (känt som *multiprocessor scheduling problem*), vilket innebär att det finns gott om litteratur tillgänglig kring denna delen av projektet.

Slutprodukten kommer att vara en ECS-motor med följande funktioner:

- Grundläggande ECS-funktionalitet (det går att skapa entities, components och systems och när man kör motorn exekverar systemen på komponenterna för att uppdatera deras data).
- Parallell systemexekvering. Oberoende system (system som inte utför arbete på samma typer av komponenter) ska automatiskt schemaläggas att köras parallellt.
- Det ska gå att implementera grundläggande simuleringar och spel i motorn. Exempelvis en boids-simulering.
- Simpel 2D-grafik.

Frivilliga utökningar till ECS-motorn:

- Parallelisering av individuella system. Ett system som exekverar på ett stort antal entities går att parallelisera över flera trådar.
- En eller flera program som implementerar ett spel eller en simulering ovanpå ECS-motorn.
- Entity-arketyper för mer effektiv läsning av komponentdata.
- Dynamisk exekveringsskalning beroende på nuvarande prestanda. Exempelvis körs ett system hälften så ofta fast med dubbelt tidssteg om motorn ligger efter (baserat på någon slags prioritet).
- Mer avancerad rendering, exempelvis 3D-grafik.

Litteraturförslag

Artiklar

Evolve your hierarchy
Originalblogginlägget som populariserade ECS-konceptet.

Entity Component System FAQ
En FAQ om ECS skriven av en utvecklare av en ECS-motor i C & C++. Under rubriken *Resources* finns en stor mängd resurser för vidare läsning.

Bevy - ECS
En kort programmeringsintroduktion till en specifik ECS-motor skriven i Rust. Kodexemplaren på sidan är väldigt enkla och du behöver inte kunna Rust för att förstå dem.

Lättläst StackExchange-inlägg om olika sätt att parallelisera ECS
`https://gamedev.stackexchange.com/a/189520/156728`

ECS scheduler thoughts
Ett blogginlägg om ECS-schemaläggning. Innehåller en mer exakt, matematisk formulering av schemaläggningsproblem utifrån grafteori.

Wikipedia - Identical machines scheduling
En wikipedia-artikel som beskriver *multiprocessor scheduling problem*.

Videor

Entity Component System (ECS) "Megacity" walkthrough - Unite LA 2018 Keynote
En demo (4 minuter) av hur bra prestanda man kan uppnå med ECS. (1:09 till 5:10 är mest relevant.)

Entity Component System Overview in 7 Minutes
En konceptuell genomgång (7 minuter) av hur ECS fungerar.

Converting your game to DOTS - Unite Copenhagen
En förklaring (44-minuter) för hur man konverterar objektorienterade spel till ECS.

Entity Component Systems and You: They're Not Just For Game Developers (SAConf NY 2019)
En förklaring (50 minuter) om hur ECS går att applicera på andra saker än spel.

Stageless: a complete scheduling overhaul
En RFC som beskriver en omskrivning av en existerande ECS-schemaläggare.

Förslagslämnare

Martin Jonsson

Målgrupp

IT, D och DV.

Speciella förkunskapskrav

En kurs i parallell programmering är användbar, men inte nödvändig.