

2017-06-15

Cíle práce

- Návrh a implementace entitního systému
- Priority:
 - Paralelní přístup
 - Modularita a přizpůsobitelnost
 - Optimalizace

Cílem této bakalářské práce byl návrh a implementace entitního systému založeného na kompozici, se zaměřením na využití v herních grafických enginech.

- Návrh a implementace entitního systému = Založeného na kompozici, zaměřením na použití v herních grafických enginech.
- Priority: = V práci jsem se dále soustředil na
- #Paralelní přístup = Umožnění paralelního přístupu k výslednému systému.
- #Modularita a přizpůsobitelnost = Protože jsou herní enginy téměř vždy specializované na určitý typ her, bylo důležité Další prioritou byla modularita a přizpůsobitelnost.
- #Optimalizace využití RVP = Důležitou částí byly také optimalizace, primárně ve směru efektivního využití rychlých vyrovnávacích pamětí

1 minuta

└ Význam

- Jádru herního enginu
 - Komunikace mezi moduly
 - Omezení faktor
- Nedostatky aktuálních řešení
 - Akumulace nechtěného stavu
 - Komunikace mezi entitami
 - Omezení množství typů
- Entity-Component-System paradigma



Dále bych přešel k důvodům, proč jsem si toto téma vybral a proč je důležité tuto problematiku řešit.

- Jádru grafického enginu = Entitní systém je jádrem herního enginu, který zprostředkovává komunikaci mezi jednotlivými moduly. Tímto však na něm vzniká de facto závislost, která může být v dalším vývoji problematická. Je tedy nutné navrhnout entitní systém tak, aby zbytečně zbytek herního enginu neomezoval.
- Nedostatky aktuálních řešení = Tím se dostávám k dalšímu bodu, čímž jsou nedostatky aktuálních řešení založených na dědičnosti. Mezi ně patří například:
 - -Akumulace stavu a chování = Při průchodu stromu dědičnosti
 - -Komunikace mezi entitami = Dalším problémem je komunikace
 - -Neohebnost typů = Omezená množina typů, vytvořená programátorem ve zdrojovém kódu.
- Entity-Component-System paradigma = Dalším bodem je ECS paradigma, na kterém je tato práce založena. ECS je programovací paradigma založené na kompozici a vzniká z objektové orientace. Mezi základní pojmy patří Entita, Komponent a Systém. ECS lze přirovnat k tabulce v databázi, kde sloupce reprezentují jednotlivé komponenty a řádky entity.

2 minuty

Řešení I

- Dekompozice problému
- Entita = Identifikátor + metadata
- Komponenty
 - Pasivní datová struktura
 - Individuální nosiče
- Systémy
 - Skupiny entit
- Paralelní zpracování
 - Entitní
 - Systémový
 - Množinový

Tímto se dostávám k samotnému návrhu entitního systému. Předpoklady - množina se kterou se pracuje, specifika podle zamerení enginu

- Dekompozice problému = Základem byla dekompozice problému do pěti modulů, o jejichž funkci se nyní ve zkratce zmíním.
- Entita = Identifikátor + metadata = První z nich je správa entit a jejich metadat. Entity jsou reprezentovány celočíselným identifikátorem a mají přidělený řádek v tabulce metadat.
- Komponenty = Další doménou jsou komponenty a jejich nosiče. Komponentou může být libovolná pasivní datová struktura, přičemž by neměla obsahovat žádné výkonné operace. Zajímavou částí jsou nosiče komponent. Nosičem je datová struktura, která udržuje mapování komponenty daného typu na entitu. Každá komponenta může mít definovaný svůj typ nosiče, což umožňuje vyšší specializaci entitního systému.
- -Pasivní datová struktura
- -Individuální nosiče
- Systémy = Doména systémů je ve skutečnosti rozdělena do dvou - samotné systémy a entitní skupiny. Každý systém specifikuje komponenty o které má zájem. Každému systému je přiřazena skupina entit, která zaručeně obsahuje pouze takové entity, o které má systém zájem. Tímto je umožněna nepřerušovaná iterace nad entitami, bez nutnosti opakovaného testování.
- -Specifikace komponent
- -Skupiny entit
- Paralelní zpracování = Poslední částí je modul paralelního zpracování. Součástí návrhu jsou tři způsoby paralelizmu - entitní, systémový a pomocí množin změn. Cílem tohoto modulu je údržba a následná aplikace množin změn.

Řešení II

- Fáze operace
 - Inicializace
 - Iterace
 - Obnova
- Implementační jazyk C++
 - Generické programování
 - Generování kódu

- Fáze operace = Dalším bodem této prezentace jsou jednotlivé fáze, ve kterých se může entitní systém nacházet.
- -Inicializace = První z nich je fáze inicializace, ve které jsou registrovány komponenty, které budou následně používány.
- -Iterace = Následuje fáze iterace, která umožňuje plný přístup k entitnímu systému
- -Obnova = Poslední fází, do které lze přejít po dokončení iterace, je obnova. Součástí obnovy je dokončení zpožděných operací a uvedení entitního systému do konzistentního stavu.
- Implementační jazyk C++ = Pro implementaci byl zvolen programovací jazyk C++, který je standardem při vývoji herních engineů. Kromě jeho základních vlastností bylo také využito generického programování za pomoci šablon. Dále bylo také využito metaprogramování ke generování kódu.
- -Generické programování
- -Generování kódu
- -Data Oriented Design

└─ Výsledky

- Multiplatformní knihovna **Entropy**
- Optimalizace
- Podpora paralelismu
- Třídní rozvrh
- Možnosti rozšíření
 - Paralelizace obnovovací fáze
 - Serializace entit
 - Vazba na smluvová rozhraní
 - Předlohy entit
 - Dynamické komponenty

Na závěr bych rád znovu shrnul hlavní výsledky této bakalářské práce.