

Cíl

Dříve než se dostanu k cíli této práce, chtěl bych se krátce zmínit o tom, co to vlastně je ECS a proč by ho někdo chtěl používat.

- Návrhový vzor, využívá kompozici místo dědičnosti.
- Entita – ID v systému, Komponent – obsahuje data, Systém – obsahuje logiku.
- Výhody – vyhnout se “super” třídám, problémy s dědičností. Možnost měnit chování Entity za běhu, pomocí přidávání/odebírání Komponent.

Konečným cílem je vytvořit knihovnu v jazyce C++, která bude jednoduše použitelná jako součást jiných projektů – Cmake.

Součástí programování je tvorba Unit testů, které slouží pro potvrzení funkcionality a následně jako regresní testy.

Poslední důležitou částí je dokumentace kódu a příklady použití.

Návrh řešení(1)

Veškerá funkcionality knihovny je pro uživatele přístupná skrz třídu Universe. Požadované funkce deleguje na jednu z manager tříd

Návrh řešení(2)

Mezi oblastmi, o které se manager třídy starají jsou:

Systémy – jejich registrace a inicializace. Systém musí dědit z třídy System.

Skupiny entit – každý systém definuje, o které Entity (podle Komponent) má zájem. Skupiny obsahují seznam Entit, které tomu odpovídají

Komponenty – registrace, přístup, Komponentou může být jakákoliv POD.

Entity – ID + generace (zamezení referencí na neexistující entity)

Návrh řešení(3)

Na obrázku lze vidět návrh toku řízení při používání ECS.

Ze začátku je nejdříve potřeba registrovat typy Komponentů, následuje přidání Systémů a inicializace celého ECS.

Toto je provedeno jednou pro každý Universe, který bude v programu použit. Další akce se již odehrávají ve smyčce -

- 1) Systémy iterují přes seznamy Entit
 - a) Přidávání/odebírání Entit
 - b) Přidávání/odebírání Komponent
- 2) Po dokončení práce, kdy Systémy již nepracují s přiřazenými skupinami Entit je zavolán refresh.

3) Během refresh se dokončí akce přidávání a odebírání, a obnoví se skupiny Entit.

Při návrhu jsem se zaměřil na to, aby bylo možné k systému přistupovat paralelně, toto vyžadovalo, aby některé akce byly provedeny na dvakrát (přidávání, odebírání).

Aktuální stav

Při návrhu jsem pro jednotlivé části vytvářel prototypy, ověření funkčnosti návrhu.

Vytvořen jednoduchý nástroj pro Unit testy a profilování.

Podle návrhu jsou zatím implementované třídy filtr, bitset, entita + id a generátor ID.

Plán řešení

První částí bude dokončení implementace podle dříve uvedeného návrhu, prozatím bez plné podpory paralelizmu. Součástí bude také psaní testů.

Následovat bude implementace paralelizmu, pokročilejší typy Systémů a úložišť Komponentů.

Dále je tvorba příkladů použití – pravděpodobně drobná hra a nakonec optimalizace datových struktur.