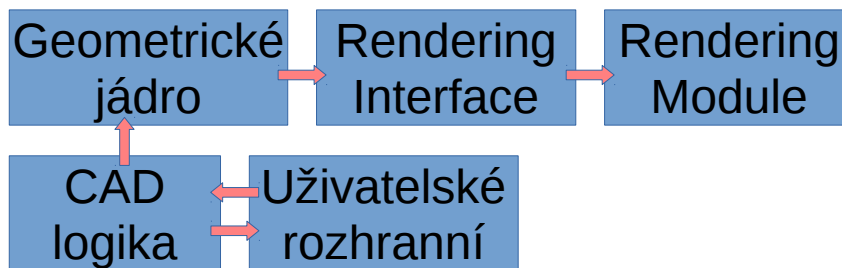


Souhrnná výzkumná zpráva za rok 2017 k projektu Vývoj softwaru v oblasti CAD systémů, 3D grafiky a vizualizace grafických scén

Ing. Jan Pečiva, Ph.D.

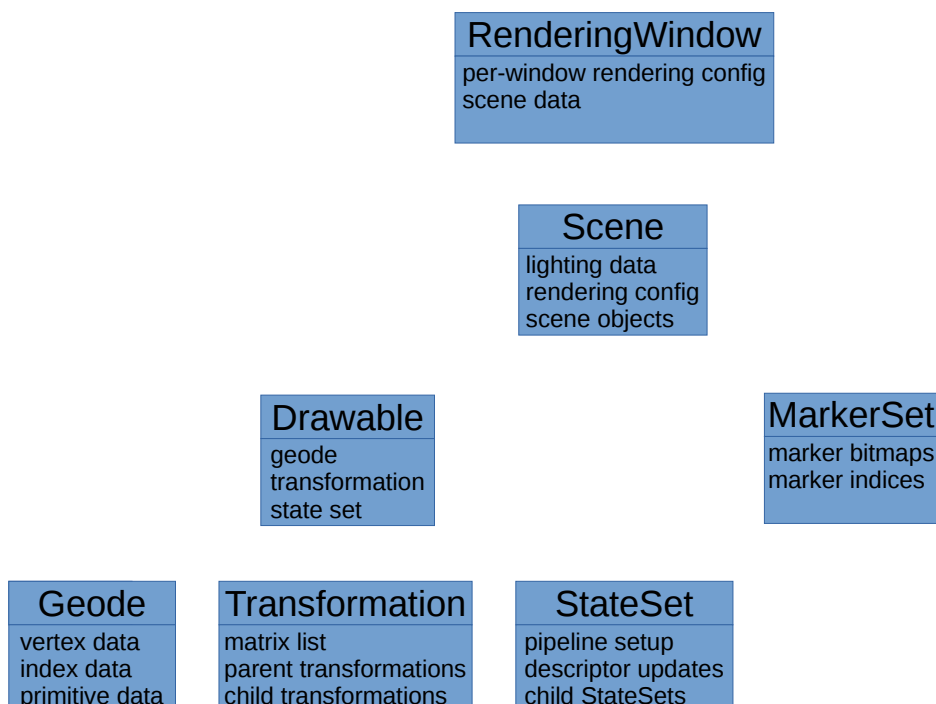
Fakulta informačních technologií, Vysoké učení technické v Brně

Výzkum a vývoj projektu se v roce 2017 soustředil na problémy výkonu a návrh rozhraní, které by umožňovalo abstrahovat datové struktury a algoritmy za vizualizační rozhraní, které by CAD aplikace používaly. Pro toto rozhraní budeme používat název Rendering interface odvozený z angličtiny a samotné datové struktury a vizualizační algoritmy budeme nazývat Rendering modul. Toto rozhraní by umožnilo modularitu návrhu CAD aplikací a znovupoužitelnost Rendering modulu, snižující tak náklady na vývoj CAD aplikací a zjednodušující jejich vývoj. Typická CAD aplikace proto vypadá následovně:



Hlavní priorita pro CAD aplikace je schopnost rychle zobrazovat obrovské množství samostatných objektů. Proto návrh Rendovacího modulu počítá se zpracováním objektů na grafickém procesoru (GPU), který díky možnosti zpracovávat velké množství úloh paralelně, dosahuje podstatně vyššího výkonu, než tradiční přístup založený na zpracování v procesoru počítače (CPU).

Po analýze požadavků a provedení množství experimentů ověřující naše předpoklady jsme navrhli tuto architekturu:



Centrem je třída Scene, která zapouzdřuje obecné nastavení zobrazené scény. RenderingWindow je pak třída odpovědná za vykreslení scény. Drawable je objekt vykreslitelný na obrazovku. Scéna jich může obsahovat i statisíce. Aplikace je musí být schopna velmi rychle zpracovávat a vykreslit na obrazovku. Každý drawable v sobě obsahuje odkaz na tři další třídy: Geode, Transformation a StateSet.

Geode (GEOmetry noDE) je třída nesoucí data geometrie objektu. Jako příklad můžeme uvést krychli, kouli, čtverec či křivku. Třída Transformation obsahuje transformační matici či více transformačních matic, které udávají umístění daného Geode ve scéně. Jednoduše řečeno, jeho souřadnice. Transformace jsou typicky organizovány do stromových struktur, což umožňuje vytvářet hierarchické transformace. StateSet v sobě zapouzdřuje rendrovací stav. Jedná se o nastavení rendrovací pipeline, materiálové vlastnosti objektů, textury, přepínání bufferů, atd. Umožňuje také vytváření tzv. render pasů. Poslední třída je MarkerSet, umožňující zobrazování "markerů", česky "značek", které jsou implementovány za pomoci bitmap.

Rozhraní je v současné době v procesu implementace a ověřování. Paralelně se vytváří pro tři rendrovací "back-endy" či implementace. Standardní implementace bude využívat rozhraní OpenGL a high-level vizualizační knihovnu Coin. Důvod výběru knihovny Coin je v jejím relativně častém použití ve vědecké a simulační komunitě a díky jejímu využití alespoň jedním z našich průmyslových partnerů.

Druhá implementace je založena na moderním rozhraní Vulkan, široce používaném jak v profesionální tak herní oblasti. Tato implementace by měla být poskytovat maximální výkon a měla být k dispozici na veškerém moderním hardware.

Poslední implementace by měla soužit pro serverové a takzvané "headless" aplikace, kde není potřeba grafického výstupu. Při aktivaci tohoto rozhraní tedy nebude aplikace nic zobrazovat, ale může sloužit jako serverová aplikace k provádění výpočtů, které obdrží například od vzdáleného uživatele na internetu.