Grafické akcelerátory

Podobne ako dnes, tak ani v minulosti nebol najrozhodujúcejším aspektom počet farieb alebo rozlíšenie, ale celková plynulosť zobrazenia, o ktorú sa s príchodom trojrozmernej grafiky v 90. rokoch starali dedikované grafické karty (Graphics Processing Unit, **GPU**), medzi ktorými bola na trhu značná rivalita. Víťazne z tohto súboja vyšli značky NVIDIA a ATI/AMD, ktoré postupne porazili alebo pohltili všetku konkurenciu a vytvorili na trhu stav obmedzenej konkurencie, tzv. oligopol. Obe značky sa neustále predbiehali v pomere výkonu a ceny, pričom z dlhodobého hľadiska NVIDIA väčšinou prinášala vyšší výkon a AMD zasa nižšiu cenu.

Medzi najznámejšie a najrozšírenejšie karty možno chronologicky radiť Nvidia Riva TNT2, GeForce 256, 3Dfx Voodoo3, Matrox G400 a ATI Rage 128. Ich hlavným prínosom bolo rozšírenie schopností počítača o hardvérovú akceleráciu procesov súvisiacich s generovaním zložitejšej 3D grafiky, ako napr. práca s vysokým množstvom polygónov, aplikácia textúr na objekty a dynamické generovanie svetiel a tieňov. Karty pritom podporovali najčastejšie niektorú z vopred definovaných knižníc ako DirectX, alebo OpenGL, pričom na jej funkcie sa odvolávali tvorcovia hier.

V súčasnosti grafické karty plnia prakticky rovnaký účel, avšak ich výkon a pamäť sa zlepšovali spoločne s ďalším technologickým vývojom. Umožnili tak rozvoj nových postupov na generovanie vysoko detailnej a plynulej hernej grafiky. Za výrazný medzník možno považovať uvedenie metódy **ray-tracing** (sledovanie lúčov). Táto metóda, ktorá prostredníctvom automatizovaných výpočtov vlastností svetelných lúčov pomocou umelej inteligencie napodobňuje spôsob, akým sa svetlo odráža a láme v skutočnom svete, umožňuje realistické vykresľovanie objektov a umožňuje generovať obraz v optimálnom rozlíšení a následne ho škálovať (upscaling – zvyšovanie rozlíšenia obrazu) na potreby zariadení s vyšším zobrazovacím rozlíšením. Dopyt po čoraz výkonnejších grafických kartách časom výrazne znížil dôležitosť ostatných komponentov počítačových zostáv, keďže obsahovali vlastný procesor aj pamäť.

Samotné grafické karty tak často niekoľkonásobne presahovali výkon základnej dosky, v ktorej boli inštalované. Dodnes ide o komponent, ktorý výrazne ovplyvňuje finančnú hodnotu celých zostáv, obzvlášť tých, ktoré sú určené na spracovanie veľkého množstva dát (napr. pri ťažbe kryptomien) alebo práve na hranie.

Spočiatku na domácich počítačoch generoval grafiku hlavný procesor počítača s využitím jeho operačnej pamäte. Jeho limity nútili tvorcov podobne ako pri konzolových hrách ku kombinácii dvojrozmerných elementov a 3D modelov, ako napríklad v hre *Alone In The Dark* (1992), ktorá vykresľovala trojrozmerné postavy v statickom bitmapovom prostredí. A naopak, vesmírny simulátor *Descent* (1994) prinášal plynulý pohyb v šiestich stupňoch slobody s použitím statickej grafiky kokpitu a riadiacich panelov, ktoré zaberali takmer polovicu viditeľnej plochy. S príchodom vyspelejších grafických kariet a procesorov upravených špecificky na generovanie grafiky začali vznikať hry obsahujúce čoraz viac polygónov, ktoré boli pokryté textúrami a tieňované svetelnými zdrojmi. Prví, ktorí naplno využili potenciál tejto technológie, sú tvorcovia hry *Quake* (1996) zo

spoločnosti id Software. Vďaka predchádzajúcim skúsenostiam z ich masovo úspešných hier *Wolfenstein* a *Doom*, postavených na enginoch id Tech 0 a id Tech 1, posunuli možnosti trojrozmerného pohybu ešte o krok ďalej. Hra umožňovala pohyb vo všetkých smeroch a levely obsahovali viaceré poschodia a objekty, po ktorých sa hráč mohol pohybovať. Zdrojový kód na *Quake* bol jeden z prvých, ktoré naplno využívali potenciál grafických kariet**.** Najskôr to bol špecifický grafický akcelerátor Rendition's Vérité 1000 (VQuake z decembra 1996) a neskôr všetky karty kompatibilné s rozhraním OpenGL (GLQuake z januára roku 1997). Vďaka rozhodnutiu autora kódu Johna Carmacka zverejniť zdrojový kód a celkovej otvorenosti voči modifikáciám sa stal **herný engine** s kódovým označením id Tech 2 východiskom vývoja množstva ďalších verzií a pokračovaní tejto hry. Z historického hľadiska existovali príklady opätovného využitia častí kódu ako základu na tvorbu nových hier, no nikdy nie v takom rozsahu. Modifikácie jeho zdrojového kódu slúžili na vývoj desiatok hier ako *Hexen II* (1997), *Soldier of Fortune* (2000) alebo *Call of Duty* (2003) a separátneho herného enginu GoldSrc (neskôr Source) od Valve, ktorý bol použitý ako základ desiatok úspešných hier od *Half Life* (1998) po *Apex Legends* (2019).



Obr. 23: Prostredie hry Quake

Kľúčom k jeho úspechu bola optimalizácia, vďaka ktorej dokázal bežať plynulo a rýchlo na menej výkonných zostavách. Umožňoval ju predovšetkým systém **BSP** (Binary Space Partitioning), ktorý kalkuloval výpočet len tých objektov, ktoré mal hráč šancu vidieť v nasledujúcich okamihoch. Rovnako tak sa používal systém **svetelnej mapy**, ktorý šetril výpočtovú kapacitu potrebnú na generovanie tieňov. *Quake* mal tiež oproti *Doomu* silnejších a odolnejších nepriateľov v menšom počte, čím šetril počet vykreslených polygónov.

Medzi ďalšie spôsoby, akými sa v hrách pokúšali optimalizovať nároky na výkon pri zachovaní estetických kvalít vizuálnej stránky hry, patria techniky slúžiace k obmedzeniu detailov scény, ktoré nie je možné dostatočne vnímať. Tieto buď limitujú počet objektov, ktoré sú viditeľné v závislosti od vzdialenosti od hráča (level-of-detail alebo **LOD**), alebo ovplyvňujú detailnosť textúry, ktorá je na ne aplikovaná (**mip-mapping**) v rôznej vzdialenosti. Špecifickým postupom znižujúcim počet polygónov je tiež tzv. **texture-baking**. Ide o proces, pri ktorom sa prenášajú textúry renderované z modelu s vysokým počtom polygónov na zjednodušený model toho istého objektu.

Prakticky ide stále o ten istý princíp využitia vysokého výkonu na vygenerovanie detailného obrazu, jeho nasnímanie a aplikáciu v prostredí, ktoré je následne možné renderovať s nižšími požiadavkami na výkon. Rozširuje sa tak technika aplikácie statických tieňov, o čom sme hovorili pri konzolách piatej generácie.

Vplyv na náročnosť vykreslenia scény môže mať tiež priestor, ktorý zaberá renderovaná scéna. Ten je možné čiastočne zmenšiť prekrytím grafickými prvkami používateľského rozhrania (Graphic User Interface, teda **GUI)** alebo zmenšením uhla zorného poľa (Field Of View – **FOV**). Tieto techniky sa využívali najmä v minulosti, dnes už výrazne neovplyvňujú plynulosť zobrazenia na monitore (zostávajú však dôležité napríklad pri zariadeniach na virtuálnu realitu, ktoré majú limitovaný výkon a vysoké požiadavky na detailnosť a plynulosť zobrazenia). Domáce počítače dovoľovali rýchlejší nárast množstva zobrazených polygónov, ktoré sú dôležité na detailnejšie zobrazenie objektov a tiež poskytujú možnosť pokrývať jednotlivé plochy bitmapovými textúrami.

Primárny rozdiel oproti konzolám piatej generácie a grafike generovanej hlavným procesorom počítačov predstavovala možnosť dynamického renderovania obrazu s využitím tzv. **shaderov**. Postupy, ktoré využívajú, vyvinul filmový priemysel pri renderovaní počítačovo animovaných filmov a efektov (kľúčovú rolu vo vývoji zohralo štúdio Pixar), no s využitím cenovo nedostupného hardvéru a tempom, ktoré nebolo využiteľné v interaktívnych médiách, keďže renderovanie každej statickej snímky trvalo desiatky hodín.

Shadery sú procesy, ktoré vykonáva GPU pred vykreslením scény. Ovplyvňujú, ako bude vyzerať trojrozmerný objekt na konci procesu, ktorý pozostáva z viacerých krokov, súhrnne označovaných za **rendering pipeline** (vykresľovací reťazec). Rôzne postupy, ktoré sú na objekt aplikované, sú spísané v knižniciach, s ktorými je možné pracovať bez nutnosti vždy znova programovať celý proces. Najpoužívanejšie sú **DirectX** (Microsoft), **OpenGL** (šírená pod voľnou licenciou) a jej následník **Vulkan**.



Obr. 24: Príklady aplikácie rôznych shaderov v procese renderovania modelu

Grafické karty spočiatku podporovali len tzv. **pixel-shading**, teda zmenu atribútov jednotlivých pixelov, napríklad aplikovaním **normálovej mapy** určujúcej, ktoré časti textúry majú byť zobrazené svetlejšie a ktoré tmavšie na vytvorenie efektu hĺbky a detailov na rovnej ploche. Využitie 2D shaderov pri tvorbe hier (okrem iného) umožňuje vytvárať rôznu štylizáciu scén, ktorá je aplikovaná na všetky objekty.

Týmto spôsobom je možné vytvoriť napríklad dojem komiksovej kresby prostredníctvom **cel-shadingu** (nižší počet farieb, tmavé okraje). Neskôr sa do možností grafických kariet pridal **vertex-shading,** ktorý je schopný automaticky manipulovať s existujúcimi bodmi polygonálneho objektu (napr. pri generovaní pohybu vodnej hladiny) alebo **geometry-shading,** ktorý dokáže vytvárať alebo odstraňovať body modelu (napríklad prostredníctvom **teselácie**, ktorá rozdeľuje polygóny na menšie). Jednotlivé shadery boli napokon v roku 2006 integrované do spoločného štandardu označovaného **Unified Shader Model**.

Revision #2
Created 11 April 2023 07:55:37 by Admin
Updated 11 April 2023 09:52:56 by Admin