



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA DOPRAVNÍ

Jakub Culek

**POROVNÁNÍ VHODNÝCH GRAFICKÝCH ENGINŮ PRO
INTERAKTIVNÍ VOZIDLOVÉ SIMULÁTORY**

Bakalářská práce

2015



K616 **Ústav dopravních prostředků**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Jakub Culek

Kód studijního programu a studijní obor studenta:

B 3710 – AUT – Automatizace a informatika

Název tématu (česky): **Porovnání vhodných grafických enginů pro interaktivní vozidlové simulátory**

Název tématu (anglicky): Study of suitable graphical engines for interactive driving simulators

Zásady pro vypracování

Při zpracování bakalářské práce se řiďte osnovou uvedenou v následujících bodech:

- Analyzujte potřeby moderních interaktivních simulátorů dopravních prostředků z hlediska grafického enginu
- Popište a zanalyzujte nezbytné funkce současných grafických enginů
- Proveďte analýzu funkcí a možností moderních grafických enginů dostupných na trhu, a to s přihlédnutím k výsledkům analýzy potřeb
- Soustředte se především na systémy s otevřenou licencí a systémy s nízkými pořizovacími náklady
- Proveďte testovací zapojení vhodného (vhodných) kandidátů do některého ze systému vozidlových simulátorů na ústavu K616

- Rozsah grafických prací: Podle pokynů vedoucího bakalářské práce
- Rozsah průvodní zprávy: minimálně 35 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)
- Seznam odborné literatury: Peter Shirley, Steve Marschner: Fundamentals of Computer Graphics
Tomas Akenine-Moller, Eric Haines, Naty Hoffman: Real time rendering
Jason Gregory: Game Engine Architecture

Vedoucí bakalářské práce: **doc. Ing. Petr Bouchner, Ph.D.**

Datum zadání bakalářské práce:

(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání bakalářské práce:

30. listopadu 2014

- a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
- b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia

L. S.

.....
doc. Ing. Petr Bouchner, Ph.D.

vedoucí
Ústavu dopravních prostředků

.....
prof. Dr. Ing. Miroslav Svítek
děkan fakulty

Potvrzuji převzetí zadání bakalářské práce.

.....
Jakub Culek
jméno a podpis studenta

V Praze dne

Poděkování

Na tomto místě bych rád poděkoval všem, kteří mi poskytli podklady pro vypracování této práce. Zvláště pak děkuji doc. Ing. Petru Bouchnerovi Ph.D. za odborné vedení a konzultování bakalářské práce a za rady, které mi poskytoval po celou dobu mého studia a dále bych chtěl poděkovat svým rodičům a blízkým za morální a materiální podporu, které se mi dostávalo po celou dobu studia.

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na ČVUT v Praze Fakultě dopravní.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré informační zdroje v souladu s Metodickým pokynem etické přípravy vysokoškolských závěrečných prací.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb, o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 30. listopadu 2015

.....

podpis

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta dopravní

POROVNÁNÍ VHODNÝCH GRAFICKÝCH ENGINŮ PRO
INTERAKTIVNÍ VOZIDLOVÉ SIMULÁTORY

bakalářská práce

listopad 2015

Jakub Culek

ABSTRAKT

Předmětem bakalářské práce „Porovnání vhodných grafických enginů pro interaktivní vozidlové simulátory“ je zmapování historického vývoje a současných technologií v oblasti herních enginů. V první části této práce je popsán vývoj počítačové grafiky a princip fungování grafických karet. Dále jsou zde uvedeny technologie tvorby a vylepšování obrazu. Druhá část práce je zaměřena na popis funkcí a vlastností herních enginů. V závěrečné části je uvedeno několik příkladů konkrétních enginů a jejich analýza.

ABSTRACT

The subject of this bachelor thesis „Study of suitable graphical engines for interactive driving simulators“ is mapping of historical development and current technologies of game engines. In the first part of this thesis is described development of computer graphics and function of graphics cards. Furthermore there are described technologies of creating and enhancing images. Second part is about functions and properties of game engines. In the last section are provided some examples of game engines and their analysis.

KLÍČOVÁ SLOVA

Herní engine, grafická karta, vozidlový simulátor

KEY WORDS

Game engine, graphics card, vehicle simulator

Obsah

1. Úvod.....	5
2. Historie grafických karet.....	6
2.1. Období bez grafické akcelerace	6
2.2. Období 2D akceleratorů	7
2.3. 3D akcelerace, druhá polovina devadesátých let	8
2.4. Období po roce 2000	15
3. Technologie užívané v grafických kartách	19
3.1. Princip fungování moderních GPU	19
3.2. 2D Akcelerace	19
3.3. OpenGL.....	20
3.4. DirectX.....	20
3.5. Vykreslovací řetězec	21
3.6. AntiAliasing	22
3.7. Filtrování textur	24
4. Herní engine	25
4.1. Funkce herního engine	25
4.2. Audio Engine.....	25
4.3. Fyzikální engine	25
4.4. Engine pro umělou inteligenci	26
4.5. Grafický engine	26
4.6. Efekty grafických engineů	27
5. Současné enginey.....	33
5.1. Licence pro volné využití softwaru - Open source	33
5.2. Cryengine	34
5.3. Unreal Engine.....	36
5.4. Ogre3D.....	38
5.5. NeoAxis3D.....	40
5.6. Zhodnocení uvedených engineů.....	43
5.7. Další dostupné enginey	43
6. Závěr	44
7. Použité zdroje.....	45
7.1. Literatura	45
7.2. Internetové zdroje.....	45
8. Seznam obrázků	46

SEZNAM POUŽITÝCH ZKRATEK

CGA	Color Graphics Adapter
CPU	Control Processing Unit
CRT	Catode Ray Tube
DDR	Double Data Rate
DVI	Digital Video Input
EGA	Enhanced Graphics Adapter
FPS	Frame Per Second
FPS	First Person Shooter
GPL	General Public License
GPU	Graphics Processing Unit
HDR	High Dynamic Range
MDA	Mda Display Ader
MIT	Massachusetts Institute of Technology
NASA	National Aeronautics and Space Administration
PCI	Peripheral Component Interconnect
RPG	Role Playing Game
SLI	Scalable Link Interface
TTL	Transistor Transistor Logic
VGA	Video Graphics Adapter

1. Úvod

Před vyvinutím prvních herních enginů v devadesátých letech minulého století byla každá hra tvořena kompletním zdrojovým kódem, který byl vyvinut pouze pro jeden konkrétní titul. Bylo potřeba vyvíjet titul včetně kódu pro práci na konkrétním hardwaru a systému. Tituly byly tvořeny velkým množstvím kódu zajišťující samotný chod aplikace a pouze malým množstvím dat pro obsah herní funkce. Termín herní engine byl poprvé použit u 3D titulů v polovině devadesátých let. V té době se jako základ využívaly již vydané tituly, zejména populární Doom a Quake. Ostatní vývojáři poté vyvíjely pouze nový grafický obsah, charaktery, zbraně a úrovně. Postupně se modifikace oddalovaly od původních her a vznikaly nové koncepty využívající pouze jádro z původních aplikací.

Postupným vývojem vznikaly projekty, které se specializovaly přímo na vývoj jednotlivých enginů jako grafický engine pro vykreslování grafického výstupu, engine simulující fyzikální jevy nebo systémy pro umělou inteligenci. S nárůstem výpočetních kapacit hardwaru, především grafické karty, lze dosáhnout stále pokročilejších vizuálních efektů a realističtějšího podání.

Dnes existuje velké množství enginů od malých projektů specializujících se pouze na určitou část funkcí až po velké balíky obsahující kompletní nástroje pro vývoj a chod softwaru. Engine nemusí být používán pouze pro počítačové hry, ale lze jej použít pro vývoj simulací a simulátorů jako je automobil na fakultě dopravní. Některé enginy se využívají také pro tvorbu animací a celých filmů se speciálními efekty. Kromě enginů, které nejsou veřejně dostupné, existuje i velké množství zástupců k volnému využití pro osobní i komerční účely. Některé z nich jsou pak dostupné zdarma a některé jsou zpoplatněny.

V první části této práce je popsán vývoj grafických karet a princip jejich činnosti způsobu zpracování obrazu. Před analýzou vybraných enginů jsou vybrány některé pokročilé funkce, které moderní enginy podporují. Poté je v této práci vybráno několik enginů, které by mohly nahradit nebo vylepšit stávající simulátory na fakultě dopravní. Pro využití ve vozidlovém simulátoru je potřeba aby engine splňoval několik kritérií. Měl by být dostupný zdarma, se širokou podporou formátů a s kvalitním grafickým výstupem. Důležitým parametrem je možnost úprav a aplikovatelnost na již stávající systém využívaný ve vozidlovém simulátoru ústavu K616.

Zastoupeny jsou dva enginy (cryengine a unreal engine), které patří k nejkompletnějším volně dostupným enginům dnešní doby. Poté pouze grafický engine (Ogre3D), který vyniká svou možností úprav celého kódu a doplnění o nejrůznější funkce. Poslední engine (NeoAxis3D) je méně rozšířený než předchozí, ale nabízí kompletní řešení pro tvorbu her a simulátorů.

2. Historie grafických karet

2.1. Období bez grafické akcelerace

Vývoj moderních grafických karet začal okolo roku 1995 s prvními dostupnými osobními počítači. Do té doby se počítače používaly zejména pro numerické výpočty a grafický výstup byl velmi jednoduchý. Navíc také byly poměrně velké pořizovací náklady.

Dříve než byly vyvinuty grafické karty, tak obraz vytvářely speciální integrované obvody, umístěné mezi hlavním procesorem a displejem. Tyto obvody pouze převáděly data z procesoru na bitový proud (bitstream) a veškeré výpočty prováděl procesor. V roce 1977 byl použit video čip *TIA IA* v konzoli Atari 2600, který měl na starosti generování obrazu i zvuku a čtení vstupů z ovladače. O rok později přišla Motorola s čipem *MC6845*, který byl základem v počítačích od IBM. Nicméně i tento čip pouze převáděl obraz na pixely a odeslal na CRT monitor. O zpracování monochromatické grafiky se staraly jiné součásti výstupního adaptéru. O další vývoj v oblasti zobrazování se zasadila především armáda a Národní úřad pro letectví a kosmonautiku NASA.

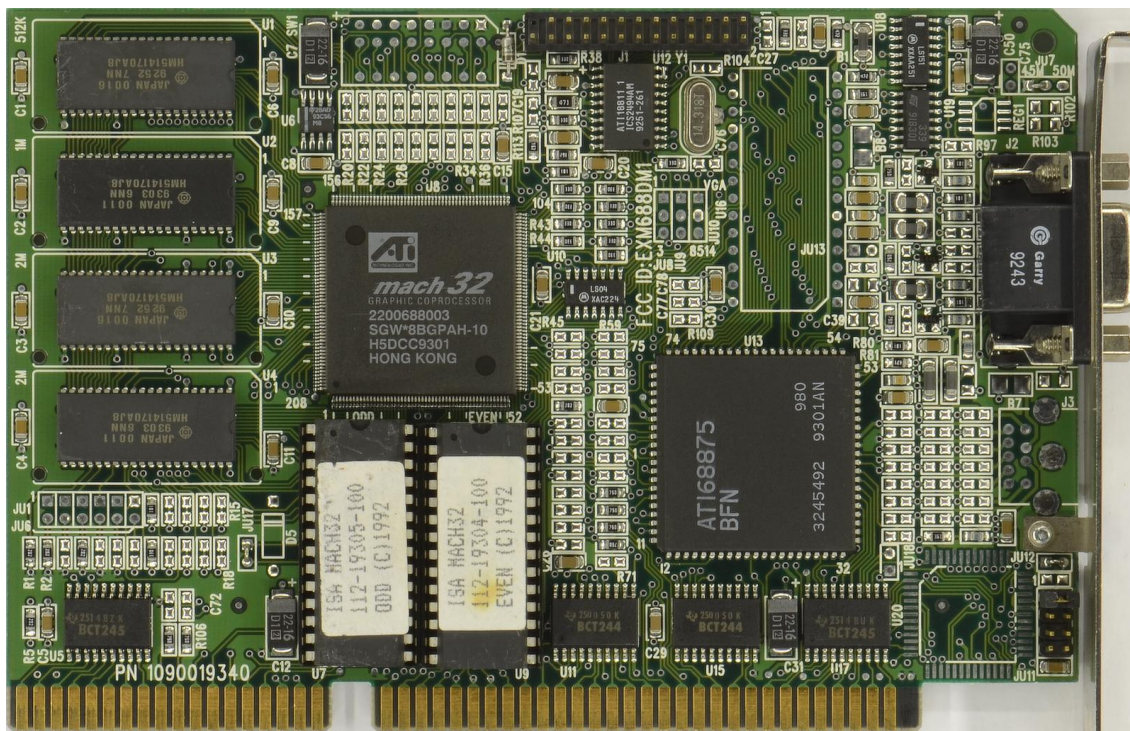
V srpnu roku 1985 byla v Kanadě založena firma ATI. Následující rok přišla na trh se svým prvním produktem. Tím byl grafický akcelerátor *Graphic Solution* pro počítače od společnosti IBM. Tato karta operovala s 16 KB paměti a byla schopná zobrazovat monochromatický text na černé pozadí TTL monitorů. Karta značně přispěla k úspěchu společnosti a jen za první rok byly zisky z prodeje deset milionů dolarů.

V březnu 1987 byla na trh uvedena řada *EGA Wonder*. První karta byla kompatibilní s MDA, CGA a EGA displeji. Obsahovala 256kb DRAM a byla schopna zobrazit 16 barev v rozlišení 640x350 pixelů. Pořizovací cena byla tehdy 399 amerických dolarů. Pozdější model 800 podporoval již šestnácti barevné VGA monitory v rozlišení 800x600 pixelů. ATI pokračovala s řadou Wonder a roku 1991 byla představena karta Wonder XL, která obsahovala zobrazovací čip Sierra RAMDAC podporující 15 bitové barvy v rozlišení až 800x600 pixelů. Do této doby sloužily grafické karty pouze k přenášení výstupu na monitor a nijak se nepodílely na urychlení výpočtu grafiky.

RAMDAC - Random Access Memory Digital-to-Analog Converter je čip pro převod digitálních dat na analogová. Je složen ze tří DA převodníků a malou paměti RAM obsahující barevnou vyhledávací tabulku. Každá barva je v paměti pomocí převodní tabulky převedena do tří složek RGB, které jsou poté jednotlivými převodníky zpracovány na analogová data. Pro režim barev truecolor nebyla kapacita paměti dostatečná a začaly se používat jiné metody tvoření výstupu.

2.2. Období 2D akceleratorů

Roku 1991 dále vyvinula společnost ATI grafický čip **Mach 8**, který sloužil jako 2D akcelerator. Byl to jeden z prvních adaptérů, který byl schopen zpracovávat grafické výpočty místo CPU. Pomocí programovacího rozhraní šlo provádět jednoduché operace jako kreslení přímků, vyplňování oblastí a práci s bitmapami. ATI později vydala kartu s novým 32 bitovým čipem **Mach 32** (obrázek 1). Karty měly kapacitu až 2 MB paměti a podporovaly rozlišení do 1280x1024 pixelů. Byl také usazen do grafické karty Graphics Ultra PRO, která jako první karta společnosti ATI používala moderní sběrnici PCI.



Obrázek 1 – Karta osazená čipem Mach32

Zdroj: <http://old.vgamuseum.info/images/stories/vlask/ATI/mach32f.jpg>

Mach 64 byl poslední a nejpobulárnější grafický 2D akcelerator. Dokonce je dodnes používán v kartách navržených pro příliš nenáročnou kancelářskou práci. Celkově byl čip vyráběn ve třech generacích. První verze, **Mach 64 CT** podporovala až 2MB paměti a za nedlouho jí vystřídala verze **Mach 64 GX**. Ta mohla být osazená v kartách s až 4 MB paměti. Poslední generace **Mach 64 VT** měla vylepšený paměťový řadič a byl výrazně zvýšen výkon čipu. Hlavní novinkou byl také video scaler, technologie díky níž lze přehrávat videa s menším rozlišením na celé obrazovce.

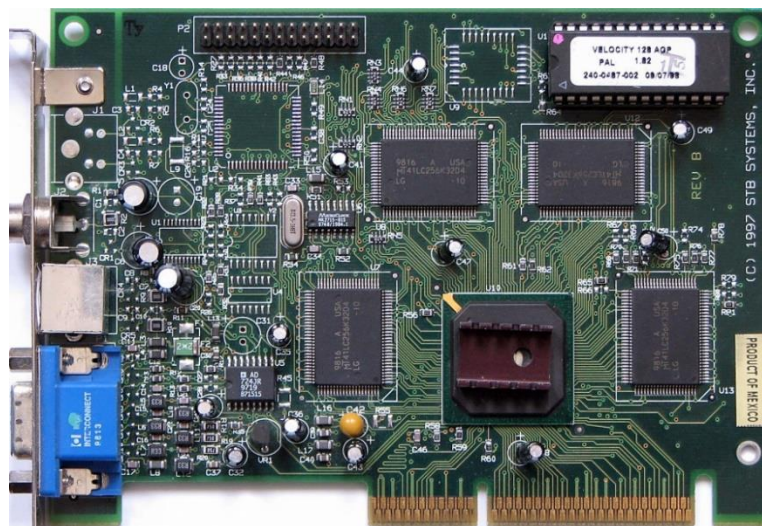
Roku 1989 se na trhu objevila další společnost vyrábějící grafické čipy a karty, S3 Graphics. Za dva roky od vzniku vyrobila grafickou kartu **S3 911** a jako první obsahovala grafický akcelerator pro grafické rozhraní Windows. Později přišla s řadou Vision kde třetí verze **968** podporovala lepší přehrávání videa a hardwarovou akceleraci bit block operací. Řada karet Trio s posledním modelem **Trio64V2** dále vylepšovala přehrávání videa a vzhledem k nízké ceně a rychlosti 2D akcelerace se tato karta stala velice populární.

2.3. 3D akcelerace, druhá polovina devadesátých let

Dalším důležitým milníkem bylo vydání roku 1992 vývojového rozhraní pro grafické karty OpenGL. Předtím měla každý hardware své specifické rozhraní a ovladače. Společnost SGI předělala své vývojové prostředí IRIS GL na nové otevřené API OpenGL. To se postupně stalo standardem pro grafické karty a byl přijat všemi výrobci grafických čipů. Knihovna se tak stala multiplatformní. Lze jej použít na mnoha systémech a u různých typů grafických akceleratorů. V případě systémů bez hardwarové akcelerace je schopna ji softwarově simulovat. V roce 1995 přišel Microsoft se svou knihovnou Direct3D, která se stala největším konkurentem OpenGL. 17. prosince 1997 byl zahájen projekt Fahrenheit, který měl spojit to nejlepší z obou knihoven, nicméně vzhledem k finančním problémům SGI a strategii Microsoftu byl roku 1999 zrušen.

2.3.1. nVidia

V druhé polovině devadesátých let byly hry vydávány především na konzolích. Běžně dostupné počítače neměli dostatečný výkon. Nově vzniklá společnost nVidia vyvinula roku 1995 grafický čip *NV1*, který byl osazován na kartu Edge 3D. Byl to první procesor schopný 3D renderování, akcelerace videa a grafického uživatelského prostředí. Karta také obsahovala port na připojení ovladače z konzole SEGA a dokonce i zvukovou kartu. Naneštěstí pro nVidii vydal Microsoft své API Direct3D, které nebylo s čipem kompatibilní. U Direct3D jsou veškeré objekty tvořeny trojúhelníkovými polygony a čip NV1 pracoval z důvodu kompatibility pro konzolové hry s polygony čtvercovými. Kvůli nekompatibilitě s novými hrami měla tedy tato karta téměř nulové prodeje. nVidia pracovala již na vylepšeném čipu NV2, ale vzhledem k malé využitelnosti čtvercových polygonů od vývoje upustila.



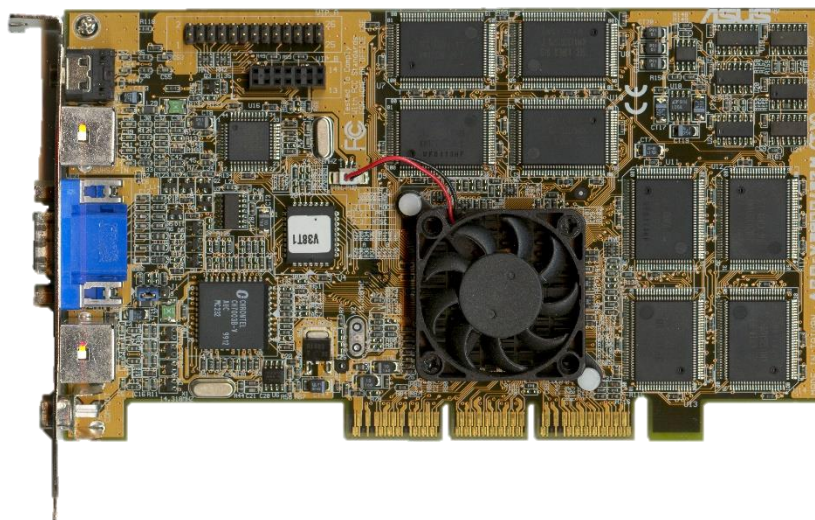
Obrázek 2 - Diamond Viper V330 osazená čipem nVidia RIVA 128

Zdroj: http://www.vgamuseum.info/images/zaatharen/nVidia/riva128_fhq.jpg

Dále bylo rozhodnuto, že nativním rozhraním pro čipy od nVidie bude Direct3D od Microsoftu. To se do budoucna ukázalo jako velmi dobrý krok. Prvním čipem byl **NV3** s označením **RIVA 128** (obrázek 2). Realtime Interactive Video and animation Accelerator přišel na trh v srpnu 1997. Jednalo se o čip, který obsahoval kompletní řešení pro zpracování obrazu. Měl integrován 2D i 3D akcelerátor, jádro VGA, RAMDAC čip a akcelerátor videa. Karta byla velice výkonná jak ve 2D tak i ve 3D. Nevýhodou byla nízká kvalita výstupu. Stala se jednou z nejrozšířenějších karet své doby a tím zachránila budoucnost celé společnosti. Za půl roku od vydání byla karta vylepšena o výkonnější RAMDAC čip, zvýšila se paměť na 8 MB a přidaly se ovladače pro podporu OpenGL.

Další čip přišel v červnu 1998 s novinkou v podobě schopnosti vykreslovat dva *texely* zároveň. Ten nesl název **RIVA TNT**. Výkonnostně byl srovnatelný s čipem Voodoo 2 společnosti 3Dfx a obraz byl kvalitnější. Další velkou výhodou bylo, že karta s tímto čipem již zvládala renderovat v 32 bitovém barevném spektru. Nevýhodou bylo pouze 16 MB paměti, které nestačilo pro vykreslování textur ve velmi vysokém rozlišení až 2048x2048 pixelů. Většina her té doby používala pouze 16 bitové textury v rozlišení do 256x256 pixelů. Karta byla kvůli vysokým nákladům na výrobu poměrně drahá a než kupovat novou kartu, tak se vyplácelo dokoupit ke stávající přídavnou Voodoo 2, která byla již levnější. Vylepšená verze **TNT 2** (obrázek 3) podporovala již 32MB paměti a procesor byl taktován na vyšší frekvenci. Díky tomu již bylo 32 bitové renderování dobře použitelné.

Texel je podobně jako pixel u obrazu elementární částí textury. U 2D grafiky je to praktický totéž, nicméně ve 3D to neplatí. Pokud je objekt blízko pozorovatele může být při nedostatečném rozlišení textury tvořen texel více pixely. Naopak pokud je objekt vzdálený, tak může být v jednom pixelu více texelů. V tomto případě se musí barva pixelu odvodit spojením barev jednotlivých texelů.

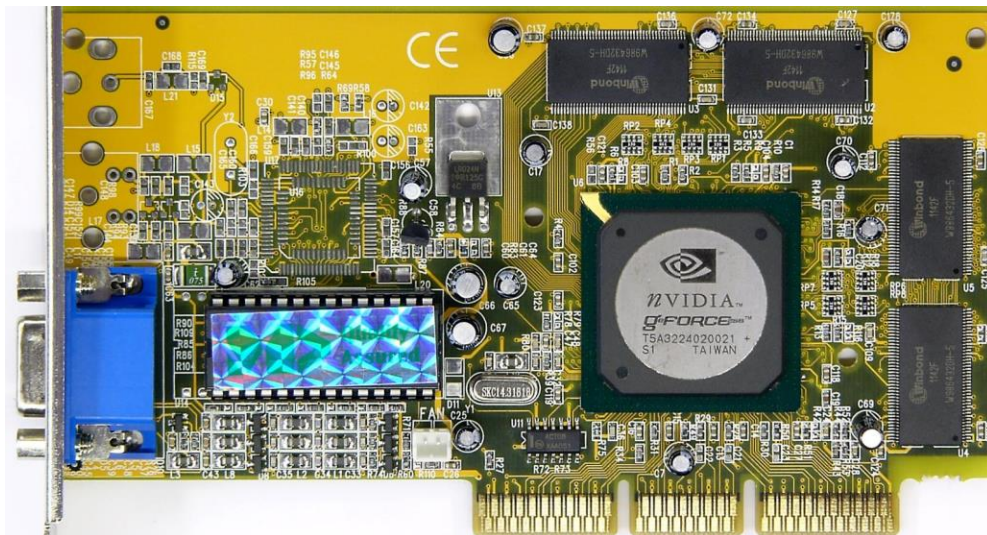


Obrázek 3 – Asus RIVA TNT 2 VIVO

Zdroj: http://hw-museum.cz/data/vga/pic/small/Asus_Riva_TNT2_VIVO_F.jpg

V srpnu 1999 začala nVidia se svou do dnes vyrábějící řadou karet GeForce. První model nesl označení 256 a obsahoval nový typ grafického čipu *NV 10* (obrázek 4). nVidia jej označila jako první GPU. Graphics Processing Unit je jedno jádrový procesor, s podporou T&L, nastavením uspořádání trojúhelníků a enginem schopných zpracovat alespoň 10 milionů polygonů za vteřinu. V porovnání s Voodoo 3 od 3Dfx a předchozí *TNT 2* poskytovala v některých titulech až 50% zlepšení *FPS*. Další novinkou byla plná podpora DirectX 7. Později byla tato karta mírně upravena pro profesionální účely a založila řadu Quadro.

FPS - Frame Per Second, česky počet snímků za sekundu udává kolikrát je zobrazen unikátní snímek během jedné sekundy. Pokud je počet snímků za sekundu nízký, jeví se hra jako neplynulá. Při snížení kvality detailů je náročnost výpočtu nižší a snímkovací frekvence se tím pádem zvýší. U videí se bere jako dostatečná hodnota, aby nebyl obraz trhaný frekvence 24 FPS. U počítačových her je minimální hodnota pro plynulou grafiku 30 FPS. 60 FPS je považován jako dostatečný i u akčnějších titulů.



Obrázek 4 - nVidia GeForce 256

Zdroj:http://www.vgamuseum.info/images/palcal/nVidia/564_manli_nVidia_geforce_256_top_hq.jpg

2.3.2. 3Dfx

Dnes již společnost 3Dfx grafické čipy nevyrobí, nicméně v druhé polovině devadesátých let přišla s revolučními inovacemi u grafických karet. Cílem společnosti bylo vytvořit dostupné grafické karty, které by umožňovaly hrát 3D hry na osobních počítačích. 6. listopadu 1995 představili čipset *Voodoo Graphics*. Novinkou bylo, že se jednalo o více samostatných čipů, které byly rozděleny na 2D a 3D a navzájem spolupracovaly. Výhodou bylo, že více čipů s menším počtem tranzistorů bylo mnohem levnější na výrobu. Nejznámější kartou s tímto čipem byla Diamond Monster 3D se 4 MB paměti. První kartou kde byl tento čip použit byla profesionální karta 3Dfx Obsidian. Ta podporovala dokonce propojení dvou karet pomocí *SLI* a dalších výpočetních jednotek. Nevýhodou bylo, že karty s čipem Voodoo obsahovali pouze 3D akcelerátor a byla potřeba ještě standartní VGA karta.

Později byl čip upraven tak, aby jej bylo možné osadit do jedné karty i s 2D čipem. Vznikla tak karta Rush, která ale neměla dostatek operační paměti a také problémy s ovladači pro propojení s 2D čipy. Proto nebyla karta příliš populární.

SLI - Scalable Link Interface je propojení dvou grafických karet patentované společností 3Dfx (obrázek 5). Propojení funguje tak, že každá karta renderuje jiný řádek. Karty by měly být od stejného výrobce a se stejným množstvím paměti. V případě, že má jedna karta kapacitu vyšší, bude mít součet hodnotu pouze dvojnásobku paměti karty s menší kapacitou. Později přišla s technologií SLI společnost nVidia, která 3Dfx odkoupila. U jejích karet lze propojit až 4 GPU, ale spojení funguje na jiném principu. Konkurenční firma AMD používá pro spojení více karet svou technologii CrossFireX.



Obrázek 5 - Propojení dvou karet Voodoo 2 do režimu SLI

Zdroj: http://www.cdrinfo.com/Sections/Articles/Sources/Asus_A8N-SLI_Deluxe/Images/Graphics/voodoo2.jpg

V listopadu roku 1997 přišla společnost s přepracovaným a vylepšeným čipem **Voodoo 2**. Ten byl díky novému výrobnímu procesu až 3x výkonnější. Nespornou výhodou u karet 3Dfx byla stále možnost propojení dvou karet a při dostatku financí šlo dosáhnout až 24 MB paměti. Byla také díky dvěma texturovacím jednotkám zpracovávat více textur najednou.

Karty Voodoo tvořily high-end tehdejší doby a tomu odpovídala i jejich cena. 2. července 1998 byla představena karta Banshee, která měla být dostupná širší veřejnosti. Základem byl čip Voodoo 2, ale pouze s jednou texturovací jednotkou. Pro kompenzování ztráty výkonu pracoval čip na o 20% vyšší frekvenci. Novinkou bylo také kvalitní 2D jádro, výstup pro TV a 3D brýle. Karta obsahovala 16 MB paměti. V několika ohledech byla karta výkonnější než původní Voodoo 2, nicméně chybějící druhá texturovací jednotka byla znatelná. Problémem byla také vysoká konkurence karty TNT od nVidie, která multitexturing podporovala. Společnost 3Dfx vedla patentové spory ohledně technologií, které byly na konkurenční kartě neoprávněně použity.

Další karta nesla název Banshee 2, vzhledem neúspěchu modelu Banshee se označoval spíše jako **Voodoo 3**. Čip byl vyráběn novým 250nm výrobním procesem a ten umožňoval taktovat na vyšší frekvenci.

Karta obsahovala opět dvě texturovací jednotky a byla vydávána v několika verzích. Nejlevnější verze Velocity disponovala oproti klasickým 16 MB pouze 8 MB paměti a měla jednu texturovací jednotku zamčenou. To se dalo ale pomocí ovladačů upravit a jednotku odemknout.

Další vydaná karta byla Voodoo s překvapivým číslem 5 (obrázek 6). Ta se brzy po vydání potýkala s konkurenční dobře výkonnou kartou od nVidie GeForce 2 GTS. Obsahovala nový procesor **VSA 100**, který již renderoval v 32 bitových barvách. Přinesla již hardwarový výpočet vyhlazování hran. Ten oproti konkurenčním řešením nevyužíval princip renderování ve vyšším rozlišení a následným zmenšení a tím byla potřeba menšího výkonu. Využívala také novou technologii T-Bufferu, kde šlo aplikovat různé efekty. V mezeře mezi další Voodoo 5 kartou byla vyvinuta výkonově slabší Voodoo 4. Vzhledem k její vysoké ceně a kvalitnější konkurenci GeForce MX nebyly její prodeje ideální. To znamenalo začátek finančních problémů.



Obrázek 6 - Voodoo 5 6000
Zdroj: <http://www.3dfx.cz/voodoo5/6000t.gif>

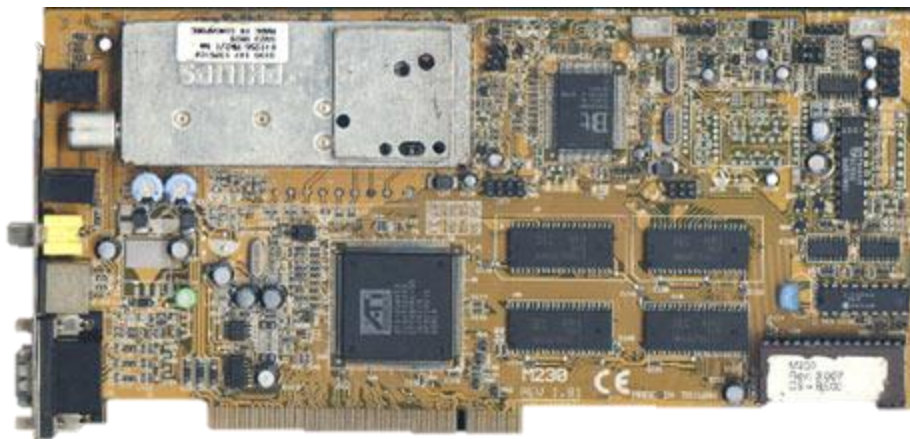
Další projekty již nebyly bohužel nikdy dotaženy až do konce. Prvním z nich byla karta Voodoo 5 600, která obsahovala dokonce čtyři procesory Voodoo 5 a byla zamýšlena jako high-end karta o velmi vysokém výkonu, ale objevil se problém s nekompatibilními řadiči. Další levnější karta Daytona s upraveným čipem **VSA-101**, obsahovala pouze 64bitový řadič, ale s možností paměti DDR. Díky 180 nm výrobní technologii byl VSA-101 schopen pracovat na vyšších frekvencích bez potřeby chlazení a tím se výrobní cena karty dostala pod 60 dolarů.

Již po vydání první generace Voodoo začal vývoj nového čipu s pracovním názvem Rampage. Vzhledem ke konkurenci bylo potřeba přijít s nástupcem Voodoo 1 dříve, než byl tento čip dokončen. Společnost 3Dfx tedy vydala pouze vylepšený původní Voodoo na verzi 2. Poté začaly práce na kartě Banshee a k dokončení Rampage opět nedošlo. Kvůli rychlé reakci na vývoj trhu byl dále vyvinut Voodoo 3 a následně VSA-100. Těsně před dokončením čipu Rampage vyhlásila 15. 12. 2000 společnost 3Dfx krach. Konec společnosti byl způsoben mnoha faktory. Prodražení sporů s nVidií kvůli patentům, absence low-end řešení a ne příliš úspěšné kombinované karty Rush a Banshee.

2.3.3. ATI

Společnost ATI přišla se svým prvním čipem 3D Rage akcelerujícím 3D grafiku v listopadu roku 1995. Jednalo se o vylepšený čip Mach64 s podporou 3D. Ten měl problémy s nedostatečnou kompatibilitou DirectX u her a byl vhodný spíše na přehrávání videí. Na jaře 1996 byl vydán na kartě 3D Xpression, která kvůli nedostatkům použitého čipu nebyla příliš výkonná.

Předělaný čip **Rage II** na kartě 3D Xpression+ již přišel s mnohem lepší podporou Direct3D a přidal také podporu přehrávání videí ve formátu MPEG2. Na prvních kartách se potýkal s nedostatkem paměti, která byla pouze ze začátku 2 MB. Později byla zvýšena na až 8 MB. Podporoval již filtraci textur a Z-Buffering. Následně se čip objevil ve verzi **Rage IIIc**, kde byl základem karty ATI All-in-Wonder (obrázek 7), která jako první kombinovala grafickou kartu s výstupem pro televizi.



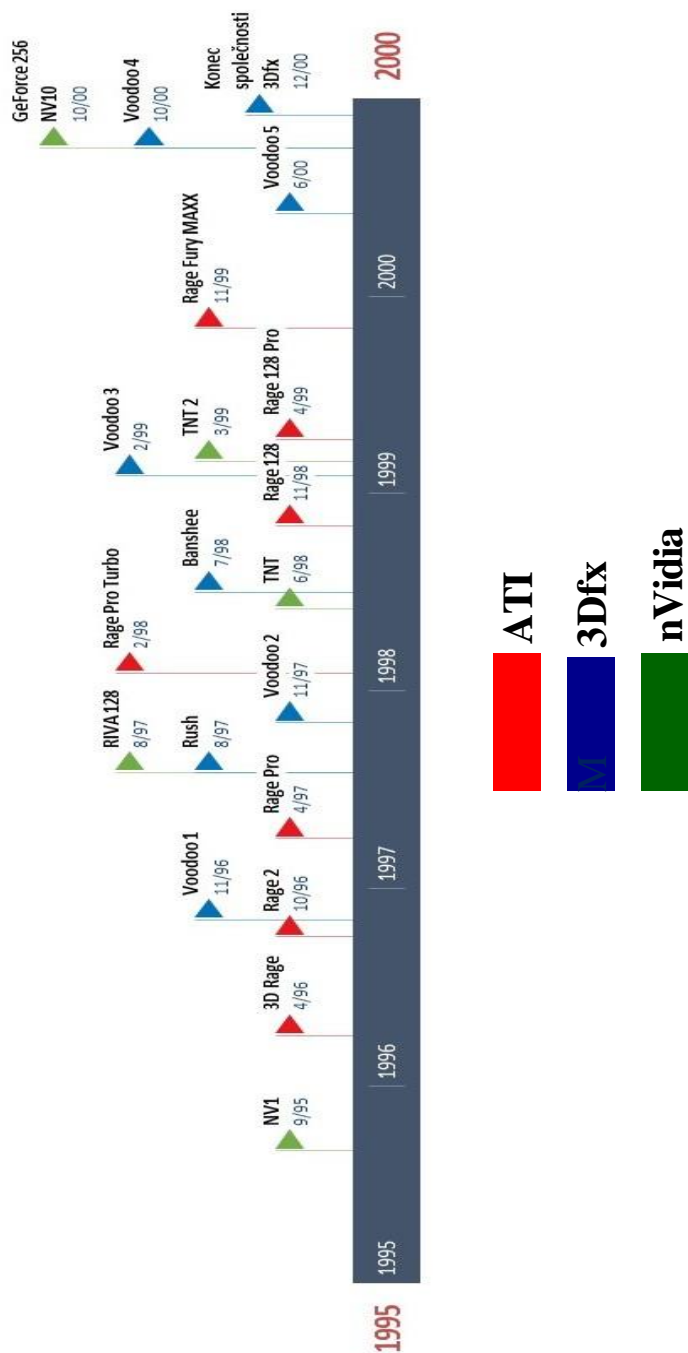
Obrázek 7 - Karta Tekram Capture TV, velmi podobné prvním kartám All in wonder
Zdroj: http://pctuning.tyden.cz/ilustrace2/soucek/3dv3_tekram_m230.jpg

Čip **Rage Pro** přišel v dubnu roku 1997 a přinesl řadu vylepšení. Nový engine pro výpočet barev, podporu mlhy a vylepšené osvětlení. Hardwarově byl čip na vysoké úrovni. Největším nedostatkem byly nekvalitní ovladače. Než byly vydány vylepšené verze ovladače, na trhu se objevily karty se srovnatelným výkonem od ostatních výrobců. S novými ovladači byla karta ještě o několik procent výkonnější a byla tedy přejmenována na Rage PRO Turbo. Tato karta se využívala především k akceleraci přehrávání DVD.

Další čip, **Rage 128** již plně podporoval OpenGL a 32bitový rendering. Opět se ale objevil problém z ovladači. Na trhu ale již byly výkonnější karty nVidia TNT a Voodoo 2. V dubnu 1999 byl představen vylepšený čip Rage 128 PRO. Ten podporoval kompresi textur, anizotropní filtrování a byl vylepšen 16bitový rendering. Problém byl již opět v ovladačích a také byl oproti konkurenci zbytečně podtaktovaný.

Jako odpověď na kartu GeForce 256 přišla Rage Fury MAXX. Ta obsahovala dva Rage 128 PRO procesory a 64 MB paměti. Každý čip měl vlastní paměť, chlazení a v prototypu i vlastní video výstup. Na trhu se bohužel objevila verze pouze s jedním společným výstupem. Více výstupů by zvyšovalo již tak vysoké výrobní náklady.

Karta tedy fungovala v režimu alternate-frame, kdy se čipy ve vykreslování střídaly po snímcích. Velkým problémem ale bylo, že nová verze Windows 2000 nepodporovala režim dvou nezávislých čipů. Tím byla karta pro spousta uživatelů nepoužitelná.



Obrázek 8 - Grafické znázornění vývoje 3D akceleratorů ve druhé polovině devadesátých let.
Zdroj: autor

2.4. Období po roce 2000

Rok 2000 byl na trhu grafických karet zlomový. Společnost 3Dfx vyhlásila 15. prosince krach a ostatní menší společnosti vyrábějící grafické čipy byly buď odkoupeny, anebo se zaměřily na jiný trh než mainstreamové 3D grafické akcelerátory.

2.4.1. nVidia

Společnost pokračovala ve vývoji a byla uvedena nová karta **GeForce 2**. Oproti předchozí verzi byla na každou výpočetní jednotku přidána ještě jedna texturovací jednotka. Díky 180 nm čipu se zvýšila taktovací frekvence na 200 MHz a tím narostl dramaticky i výkon. Ve své době byla nejvýkonnější na trhu a žádná konkurenční karta se jí nemohla příliš vyrovnat. GeForce se stala velice populární a byla vyráběna v několika variantách. Verze MX byla levnější, s menším výkonem, ale stala se jednou z nejprodávanějších.

Další řada **GeForce 3** uvedená v únoru 2001 se stala opět jednou z nejvýkonnějších na trhu. Přinesla podporu DirectX 8, pokročilý AntiAliasing a osminásobné Anizotropní filtrování. Největší novinkou ale byla programovatelná jednotka vertex shader. Dále obsahovala čtyři pixel pipeline s osmi texturovacími jednotkami. Přibyl také nový z-buffér. Nástupce **GeForce 4** (obrázek 9) obsahoval čip N25, který je považován za jeden z nejpovedenějších. Karta byla na trh uvedena v lednu 2002 a v té době neměla konkurenci.



Obrázek 9 - Karta Creative GeForce 4 Ti 4400

Zdroj: http://www.techspot.com/old-articles/reviews/hardware/3dblaster_ti4400/card.jpg

Následná generace čipu NV30 již ale tak úspěšná nebyla. Byl to první čip s podporou DirectX 9. Konstrukčně ke změnám příliš nedošlo. Obsahoval 3 vertex shadery a 4 pixel pipeline každou se dvěma texturovacími jednotkami. Ty ovšem měly problémy se stabilitou. Výkonově byl čip srovnatelný s konkurenčním ATI R300. Při aktivaci vyhlazování hran a dalších optimalizačních funkcí docházelo k razantnímu propadu výkonu. Vzrostla také energetická náročnost a potřeba výkonnějších zdrojů s přídatným napájením karet.

Přepřacovaný čip *NV40* již byl úspěšnější. Obsahoval 222 milionů tranzistorů, 6 vertex a 12 pixel pipeline. Podporoval nový DirectX 9.0c a shader model 3.0. U optimalizace obrazu již nedocházelo k tam znatelným propadům výkonu. Novinkou je více paralelních pipeline a tím je možné provádět více operací najednou. Tento čip byl usazován do karet řady GeForce 6. Ty také jako první od nVidie podporovaly spojení více karet na jedné kompatibilní základní desce. Princip vycházel od společnosti 3Dfx. Aby bylo možné pomocí SLI můstku karty propojit, musely splňovat řadu podmínek. Karty musely být naprosto stejné a od stejného výrobce. Ke správné funkci bylo nutné vytvořit pro jednotlivé hry speciální profily s instrukcemi, jak mají karty spolupracovat. Celá soustava byla také náročná na napájení.

NVidia své čipy postupně vylepšovala, zlepšoval se výrobní proces umožňující vyšší frekvence, zvyšoval se počet tranzistorů a shaderů. Jedna z nejnovějších karet od nVidie je řada **GeForce GTX 900** s novými čipy Maxwell. Zatím nejvýkonnější kartou je GeForce **GTX 980 Ti** která obsahuje 2816 shaderů a celých 6 GB paměti pracuje na frekvenci 7012 MHz.

2.4.2. ATI

Společnost ATI přišla roku 2000 také s novou generací GPU Radeon. První čip **R 100** přinesl technologii *HyperZ*, která optimalizovala celý renderovací proces. I přes slabší hardwarovou konfiguraci oproti konkurenci poskytoval dostatečný výkon. V porovnání byla karta sice méně výkonná než GeForce 2, ale měla řadu výhod. Například nové technologie v přehrávání videa, ve kterém byla společnost jedna z nejlepších na trhu. Obsahovala kvalitní paměti s kapacitou 32 a 64 MB. Karta nově podporovala programovatelné shadery, které bohužel nebyly s novou verzí DirectX kompatibilní. Podobně jako nVidia vyráběla levnější variantu karet MX, vydala ATI kartu s verzí čipu **RV 100** (obrázek 10). Ten obsahoval pouze jednu vykreslovací pipeline a nepodporoval technologii *HyperZ*. Přibyl ale výstup pro dva monitory.



Obrázek 10 - Ati Radeon R100

Zdroj: https://upload.wikimedia.org/wikipedia/commons/d/d7/Radeon_RV100_DDR.JPG

HyperZ technologie optimalizuje práci s Z-bufferem. To je část paměti, ve které jsou uloženy informace o poloze jednotlivých objektů a jejich vzdálenosti od pozorovatele. Při standardním vykreslování se vyrenderují postupně všechny objekty scény. U této technologie se nejdříve určí vzdálenosti jednotlivých objektů od pozorovatele. Ty se poté porovnaly mezi sebou, a překrývající se části již vůbec nerenderují. Součástí je také funkce pro rychlejší vyprazdňování z-bufferu. Při aktivaci těchto funkcí dochází k nárůstu výkonu karet až o 40 procent.

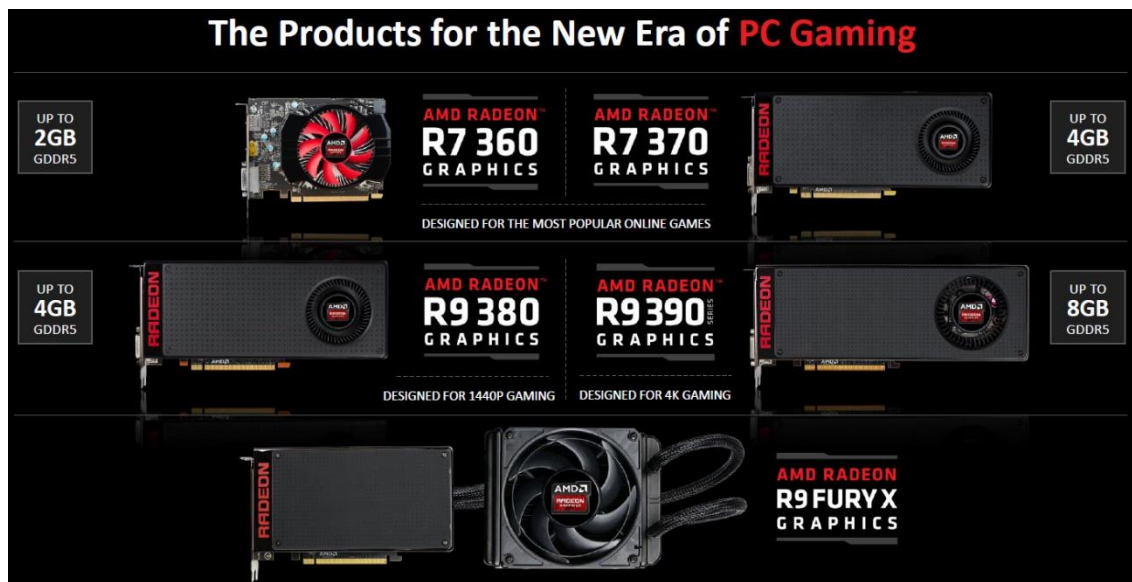
Další verze čipu **R 200** se objevila v srpnu 2001 a byla vyráběna 150 nm procesem a již podporovala DirectX 8.1 a i OpenGL 1.4. Důležitým prvkem byla plná podpora shaderu verze 1.4. Ovladače již byly odladěnější než u předchozích verzí a první karta s tímto čipem byla ATI Radeon 8500. Ta byla výkonově srovnatelná s konkurenční GeForce 3. Také tento čip byl později vydán v levnější verzi **RV 250**. Tento čip obsahoval pouze jednu texturovací jednotku a jeden vertex shader. Výkonově byl srovnatelný s verzí R 200.

Čip **R 300** byl vydán v srpnu 2002 a jako první plně podporoval nový DirectX 9 se shardery verze 2.0. Nechyběla ani 2D akcelerace pro GUI, video akcelerace a podpora více monitorů. První kartou s tímto čipem byla Radeon 9700. Čip obsahoval přes 100 milionů tranzistorů 4 vertex a 8 pixel pipeline. Začátkem roku 2003 byl vydán ještě ve vylepšené verzi **R 350** s podporou paměti DDR2. I z něj byla vyrobena levnější **RV 350** s polovičním počtem pipelin. Byl vyroben 130 nm procesem a to umožňovalo taktovat na vyšší frekvence. Tím pak byly vyráběny i pozdější verze **R 360** a **RV 360**.

Na jaře 2004 přišla společnost ATI s novou řadou high-endových karet X. Na první Radeon X800 PRO byl usazen čip **R 420**, který nově pracoval s rychlou pamětí typu GDDR 3. Obsahoval 6 vertex a 16 pixel pipeline a také podporoval DirectX 9.0 b. Čip byl do budoucna několikrát upraven a byla vydána i slabší verze. U dalších čipů se zlepšovala výrobní technologie a tím bylo možné taktovat čipy na vyšší frekvenci, zvyšoval se počet vertex a pixel pipeline a zvyšovala se i paměť grafických karet.

V polovině roku 2005 ATI představila obdobu SLI, technologii CrossFire. Ta umožňovala propojení dvou karet za účelem zvýšení výkonu. Nevýhodou bylo nutnost pořízení karty z edice CrossFire, která fungovala v režimu master. K té bylo možné připojit libovolnou kartu pomocí speciálního kabelu DVI. Koncem roku 2006 byla technologie vylepšena a již nebyla potřeba jedna master karta, nýbrž mohly být spojeny dvě libovolné karty podporující tuto technologii. Změny se dočkalo i samotné propojení a to bylo již pomocí interní propojky.

V červenci 2006 byla společnost ATI odkoupena firmou AMD a grafické karty se začaly vydávat pod značkou ATI Radeon HD. Roku 2013 došlo opět ke změně značení a nyní se karty označují podle výkonosti ve formátu RX a generací trojmístným číslem (obrázek 11).



Obrázek 11 - Příklad číslování řady 300 grafických karet AMD

Zdroj: http://www.kitguru.net/wp-content/uploads/2015/06/amd_radeon_300_promo.jpg

Jeden z posledních modelů karty od AMD Radeon R9 Fury X přišel na trh 24. června 2015. Obsahuje čip AMD Fiji a nové paměti HBM které jsou schopny dosáhnout propustnosti až 512GB za sekundu. Čip pracuje na frekvenci až 1050 Mhz a obsahuje 8,9 miliard tranzistorů, které jsou sestaveny do 4096 shaderů. Kapacita paměti dosahuje GB a lze ji použít na hry i v 4K rozlišení.

3. Technologie užívané v grafických kartách

3.1. Princip fungování moderních GPU

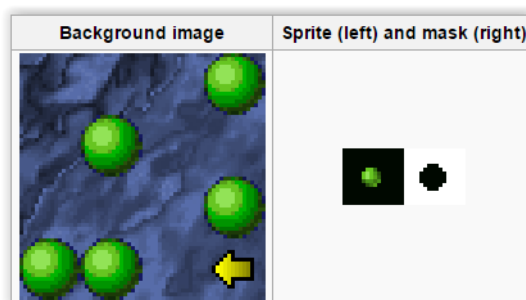
GPU jsou speciální procesory určené ke zpracování videa a výpočet grafiky. Svou strukturou jsou navrženy k výpočtu velkých objemů dat a pomáhají menší zátěži CPU. V dnešní době je většina výkonu grafických karet použita na zpracování 3D grafiky. Karty pracují ve vykreslovacích řetězcích, což je princip vykreslení výstupu ze 3D scén. Dříve byl tento postup fixně dán konfigurací procesoru. V roce 2001 přinesla karta GeForce 3 možnost tyto řetězce softwarově upravovat.

3.2. 2D Akcelerace

2D akcelerátory grafiky slouží pro zrychlení práce s dvourozměrnou grafikou. Dříve se využíval u 2D počítačových her a při akceleraci grafického uživatelského rozhraní operačních systémů. Účelem bylo zejména snížení zátěže CPU. K tomu se využívalo několika operací, viz níže. Jedny z nejlepších akcelerátorů vyráběla společnost ATI. V dnešní době jsou již CPU dostatečně výkonné a také moderní GPU jsou schopny dřívější speciální čipy plně nahradit. Od zavedení shaderů v grafických kartách se tento způsob akcelerace přestal využívat.

3.2.1. Sprity

Sprity jsou malé dvourozměrné obrázky či animace, které jsou zasazeny do scény (obrázek 12). V paměti je Sprite od okolního obrazu oddělen. To umožňovalo snadnější vykreslování pohybu, protože není potřeba vykreslovat celou scénu pokaždé zvlášť. Vykreslování probíhalo ve speciálním hardwaru Blitteru. Spritem je například i kurzor myši.



Obrázek 12 - Ukázka aplikování Spritů do scény
Zdroj: https://en.wikipedia.org/wiki/Bit_blit

3.2.2. Blitter, BitBlit

Blitter je speciální procesor, který dokáže velmi rychle přesouvat data v paměti počítače nezávisle na CPU. Typická funkce je přesun částí obrazu na neměnicím se pozadí scény.

BitBlit, neboli bit block transfer je operace, při které je kombinováno více bitmap do jedné. Ke sloučení je využíváno booleanských funkcí AND, OR, XOR a NOT.

3.3. OpenGL

OpenGL je programovací rozhraní pro počítačovou grafiku. Vznikl v roce 1992 a s mnoha aktualizacemi je jedním z grafických standardů dodnes. Celá knihovna je navržena tak, aby byla použitelná v téměř jakémkoliv programovacím jazyce. Nejčastěji se programuje v C++, případně C. Knihovna je opensource a je tedy veřejně dostupný celý zdrojový kód. Hlavní princip spočívá tom, že se obraz před výstupem tvoří v tzv. frame bufferu. Viz. Kapitola vykreslovací řetězec níže.

3.4. DirectX

Před vydáním Windows se vyvíjely hry pro systém MS-DOS. Ten poskytoval přímý přístup k jednotlivým komponentům, jako jsou grafická karta, vstupní zařízení a zvuková karta. Nový operační systém Windows 95 již nedovoloval přímý přístup k operační paměti a kontrolu jednotlivých komponentů. To byl jeden z důvodů, proč bylo nutné přijít s novým vývojovým rozhráním, které by přístup pod Windows umožňoval. To bylo vyřešeno vývojem knihovny DirectX. První verze byla vydána v září 1995. Následovalo dlouhé testování kompatibility na všemožném hardwaru a softwaru. Dále dlouho trvalo, než vývojáři her na API DirectX začaly vyvíjet tituly ve velkém. Dnes je i vzhledem k rozšíření Windows využívány prakticky ve všech titulech vydaných na PC. Tento soubor knihoven se skládá z několika částí:

- DirectDraw - poskytuje přímý přístup ke grafické kartě, slouží zejména pro vykreslování 2D grafiky
- Direct3D - API pro práci se 3D grafikou. Díky této části lze využívat a ovládat grafické akcelerátory v daném PC
- DirectSound - část umožňující práci se zvuky ve vysoké kvalitě
- DirectMusic - společně s DirectSound poskytuje kompletní nástroje k ovládání zvuku a hudby
- DirectInput - dovoluje čtení vstupů z externích zařízení včetně ovládání funkce Force Feedback
- DirectPlay - rozhraní k ovládání komunikace po síti
- DirectShow - slouží pro streamování médií v systému Windows. Umožňuje záznam i přehrávání ve vysoké kvalitě.

Na rozdíl od OpenGL není DirectX open source a nelze tedy upravovat základní zdrojový kód. Stěženi verzi DirectX byla 8.0, která přinesla v roce 2000 podporu vertex a pixel shaderů, mlhy a mapování textur. V současné době je již verze 12.0, která slibuje zvýšení výkonu díky asynchronním shaderům a lepší podpoře více jader.

3.5. Vykreslovací řetězec

3.5.1. Určení vrcholů

Na scéně jsou určeny jednotlivé vrcholy (vertexy). Ke každému vrcholu je přiřazena matice dat obsahující jeho polohu, barvu, normálu a texturovací souřadnici.

3.5.2. Transformace vrcholů

V dalším kroku se pomocí vertex shaderu jednotlivá data upravují. Dále lze aplikovat geometry shader, který může vrcholy přidávat nebo naopak vypouštět a tím upravovat výslednou geometrii.

3.5.3. Sestavení primitiv

Data o jednotlivých vrcholech z předchozích shaderů jsou zpracována a z vrcholů se sestaví síť tvořené trojúhelníky.

3.5.4. Rasterizace

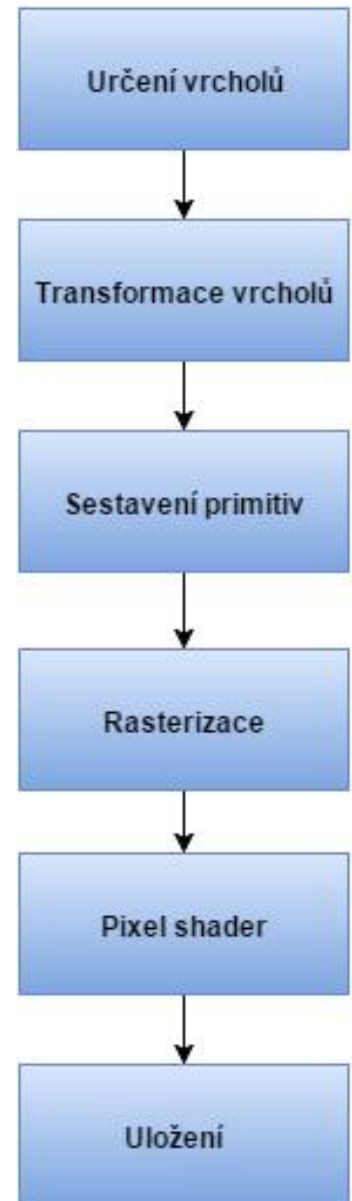
V této části se převádí 3D scéna na 2D obraz. Primitiva se interpolují do rastrovací mřížky a převedou se na jednotlivé pixely. Data o hloubce se uloží do Z-bufferu.

3.5.5. Pixel shader

Zde se pomocí pixel shaderu upravují data o jednotlivých pixelech. Výstupem jsou informace o barvě a hloubce jednotlivých pixelů. Dále lze aplikovat nastavení osvětlení a mapovat textury. Jelikož se zde zpracovává pixel po pixelu, je tato část na výkon nejnáročnější.

3.5.6. Uložení do bufferu

Nakonec je výsledný render uložený do tzv. framebufferu odkud je odeslán na obrazovku. Dále jsou v bufferu uloženy informace o odstínu a barvě jednotlivých pixelů. Také lze aplikovat test hloubky, kde se překrývající fragmenty vypouští. Data z framebufferu lze použít pro vytvoření textur použitelných v dalších procesech.

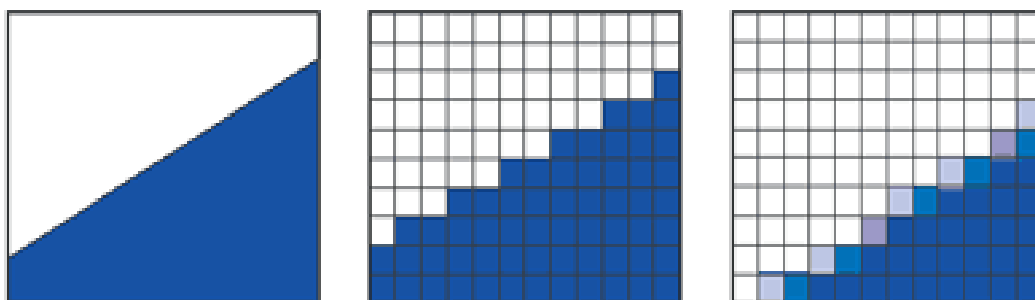


Obrázek 13- Schéma vykreslovacího řetězce moderních GPU.

Zdroj: autor

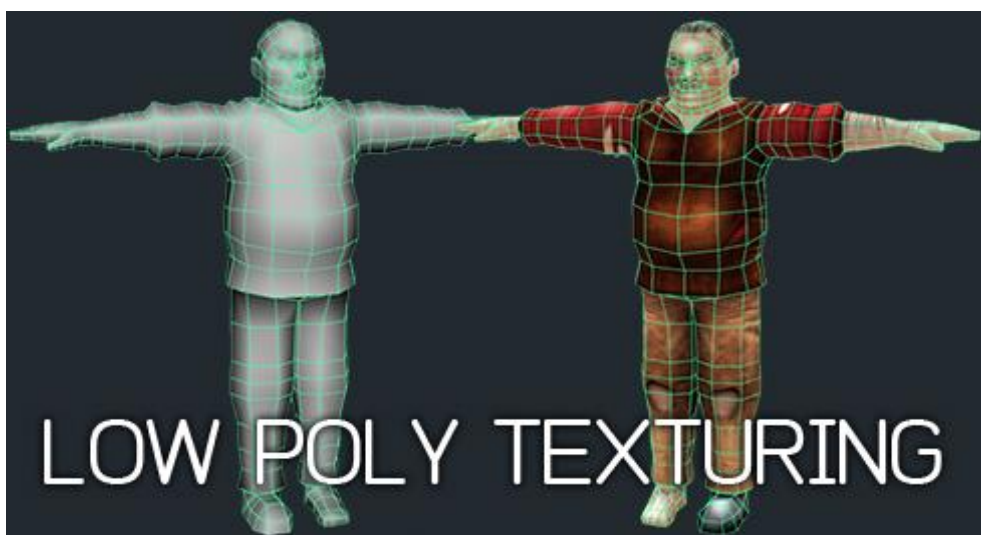
3.6. AntiAliasing

Anti Aliasing neboli vyhlazování hran je způsob jak předejít „zubatým“ hranám a celkově zvýšit kvalitu obrazu (obrázek 14). Všechny modely jsou tvořeny sítí polygonů, na které je pak nanесena textura (obrázek 15). Při rasterizaci je scéna převedena do jednotlivých pixelů. Na rozhraní dvou barev se určuje výsledný odstín pixelu podle majoritního zastoupení dané barvy. Problémy vznikají v případě, kde jsou barvy zastoupeny v podobném poměru. Výsledná barva může být pouze jedna a tím vznikají barevné schody. Čím více polygonů síť tvoří, tak tím jsou objekty podrobnější a hrany hladší. S rostoucím rozlišením však klesá hodnota FPS. Proto se začaly aplikovat způsoby vyhlazování hran.



Obrázek 14 - Zobrazení funkce Anti aliasing. Nejdříve je ideální hrana (vlevo), na které vzniknou při rasterizaci zuby (uprostřed). Pomocí Anti aliasingu je dosaženo částečného vyhlazení (vpravo).

Zdroj: <http://www.svethardware.cz/3d-technologie-anti-aliasing/8871/img/body-0.5FC8.jpg>

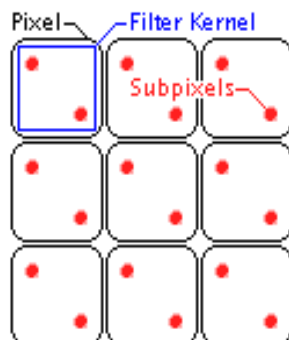


Obrázek 15 - Ukázka modelu složeného ze sítě polygonů a nanесenou texturou

Zdroj: http://blog.damienfate.com/wp-content/uploads/blog.damienfate.com/2011/04/mesh02_header1.jpg

3.6.1. Supersampling

Tato metoda byla poprvé aplikována v období karet GeForce 256, Radeon 7000 a Voodoo 5. Základním principem je vnitřní vykreslování scény ve větším rozlišení a rozdělení pixelů na více subpixelů. Ty jsou pak pomocí filtrů sloučeny zpět do jednoho pixelu a jejich barvy jsou zprůměrovány (obrázek 16). N-násobný supersampling tedy znamená, že je každý jednotlivý pixel rozdělen na n subpixelů a tím se i n -krát zvýší pracovní rozlišení. Nevýhodou tohoto způsobu je, že renderování ve vyšším rozlišení a následné převzorkování je velmi náročné na výpočetní výkon grafických karet.



Obrázek 16 - Metoda určování pixelu pomocí rozdělení na jednotlivé subpixely

Zdroj: <http://www.svethardware.cz/3d-technologie-anti-aliasing/8871-2/img/body-0.3E0.jpg>

3.6.2. Multisampling

Multisampling je vylepšená metoda vyhlazování se kterou přišly již karty GeForce 3. Princip je stejný jako u supersamplingu, ale rozdělení na subpixely a výpočet barev je aplikován pouze na rozhraní polygonů. Tím se nemění kvalita, ale ušetří se výkon, jelikož není třeba přepočítávat celou scénu. Tato metoda je v dnešní době s menšími rozdíly nejběžnější.

3.6.3. Další metody specifické pro výrobce

Výrobce čipů nVidia přišel s technologií Quincunx. U této metody jsou pro výpočet barvy zahrnuty i subpixely okolních pixelů. Výhodou je, že pro výpočet je použito více subpixelů bez potřeby vyššího rozlišení.

Společnost ATI přinesla technologii SmoothVision. U té se jednotlivé subpixely v pixelech vybírají náhodně. Tento proces je aplikován jako u supersamplingu na celou scénu, ale díky technologii HyperZ není vliv na výkon tak razantní.

3.6.4. Chyby při vyhlazování

Problém při vyhlazování může nastat u objektů, které jsou velké pouze v řádech pixelů. Například kabely, dráty, či pletiva plotů. Při průměrování barev z jednotlivých subpixelů se tak může stát, že ani jeden subpixel nebude obsahovat barvu objektu a ten se tedy pak nevykreslí. Další problém může nastat, pokud je hrana natočena tak, že neprochází danými subpixely a tím bude výsledná barva pixelu shodná s tou původní. Těmito kritickými úhly při vyhlazování se 4 subpixely jsou násobky 90 stupňů.

3.7. Filtrování textur

Filtrování textur je forma antialiasingu, při které se zvyšuje kvalita textur nanesených na jednotlivé polygonové síť. Při filtrování se vyhlazují spoje jednotlivých textur.

3.7.1. Mapování bez filtrování

Při této metodě se určí barva pixelu podle nejbližšího texelu. Tato metoda je málo náročná na výkon grafických karet, ale dochází zde často ke zkreslení a nežádoucím efektům.

3.7.2. Bilineární filtrování

Metoda, při které se určuje výsledná barva pixelu na základě odpovídajících texelů z nanesené textury. U textur lze při zachování poměru stran měnit velikost. Například pokud je objekt od pozorovatele vzdálený, stačí, aby byla textura menší, měla menší rozlišení a byla tak méně náročná na zpracování. Naopak blízké textury jsou větší a obsahují více detailů. Tento proces změny velikosti textur se nazývá mipmapování. Ve scéně lze tedy opakovat jednu texturu vícekrát s rozdílným rozlišením a velikostí. Barva pixelu je poté určena jako průměr barev čtyř nejbližších texelů dané textury.

3.7.3. Trilineární filtrování

Jelikož je scéna složena z textury o různém rozlišení, tak na rozhraních jednotlivých velikostí se mohou objevit znatelné hrany. Při trilineárním filtrování se tyto hrany vyhladí tím, že se v jejich okolí použije znovu bilineární filtrování a tím se opět zprůměrují barvy nejbližších pixelů.

3.7.4. Anizotropní filtrování

Problémem zůstávají textury, které jsou na okraji zorného pole. Předchozí metody pracují se čtvercovými mipmapami, jenže ve skutečnosti je potřeba zohlednit, že se jednotlivé hrany vůči pozorovateli nejsou kolmé. Dobrým příkladem je dlážděná podlaha (obrázek 17). Při anizotropním filtrování se již mění i poměr stran textur tak, že čím více je textura k okraji obrazovky, tím je šířka textury proti délce menší. To zajišťuje, že nedochází k rozmazávání okrajů zorného pole.



Obrázek 17 - Ukázka rozdílu mezi trilineárním a anizotropním filtrováním. V levé části je vidět rozmazání obrazu dané pevným poměrem stran textury.

Zdroj: https://upload.wikimedia.org/wikipedia/commons/8/80/Anisotropic_filtering_en.png

4. Herní engine

4.1. Funkce herního engine

Herní engine tvoří základ pro vývoj a chod nejen počítačových her, ale například i virtuálních simulátorů a dalších aplikací operujících v grafickém prostředí. Větší enginey se skládají z více dílčích engineů, které řeší jednotlivé části hry. Většina engineů obsahuje své vlastní prostředí pro vývoj.

4.2. Audio Engine

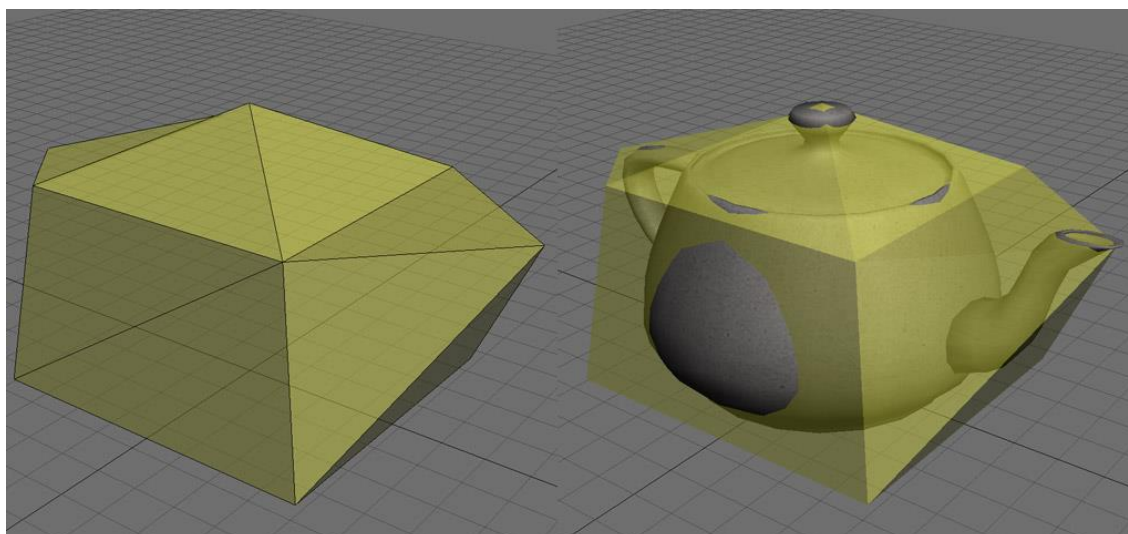
Engine pro generaci zvuků a přehrávání hudby. Zejména u 3D scén zaručuje, aby byl zvuk prostorový a v co nejlepší kvalitě.

4.3. Fyzikální engine

Tento engine počítá a simuluje veškeré fyzikální procesy ve hře. V dnešní době je již nedílnou součástí většiny herních engineů. Jednou z jeho hlavních činností je výpočet pohybu objektů a jejich interakce. Základem jsou knihovny obsahující komplexní algoritmy pro fyzikální výpočty a také obecné pohybové rovnice pro simulaci gravitace a mechaniku pevných i kapalných těles. Fyzikální enginey se dají rozdělit do dvou kategorií. Na real-time computing, kdy je potřeba simulovat fyziku v reálném čase a high-precision computing, kde je naopak potřeba dosáhnout velice přesných výsledků.

4.3.1. Real-time computing

U počítačových her je kladen důraz více na rychlost výpočtu než na absolutní přesnost. Pro tyto účely je každá entita zastoupena dvěma objekty. Jeden objekt, který je ve hře vykreslen detailně a druhý objekt je jednodušší a ten slouží pro výpočty fyzikálního engine (obrázek 18). Důležitou funkcí je výpočet kolizí a interakcí mezi entitami, při kterém je použita zjednodušená síť objektů. Mezi náročné efekty patří například exploze, déšť nebo věrohodné zobrazení mlhy a kouře.



Obrázek 18 - Zobrazení entity ve dvou objektech. Renderovaný model (vpravo) a model používaný pro výpočet kolizí (vlevo)

Zdroj: http://wiki.splashdamage.com/upload/b/b8/Teapot_vehicleplayerclip.jpg

S postupným vývojem technologií jsou výpočty stále rychlejší a přesnější. V dnešní době se fyzika nejčastěji počítá na grafické kartě pomocí tzv. Physics Processing Unit. Jeden z největších výrobců grafických čipů nVidia začal v roce 2008 využívat technologii PhysX, která se zaměřuje výhradně na výpočet fyziky na grafických kartách. Tím snižuje výpočetní nároky na CPU.

4.3.2. High-precision computing

Fyzikální enginey slouží nejen při vývoji a hraní her, ale i pro vědecké výpočty a různé simulace, kde je potřeba zahrnout větší rozsah proměnných a dosáhnout velmi přesných výsledků na úkor delší doby výpočtu. Například jeden z prvních počítačů ENIAC sloužil k výpočtu balistických křivek střel americké armády. Bylo potřeba počítat nejen s typem a hmotností střel, ale také s odporem vzduchu a vlivem povětrnostních podmínek.

4.4. Engine pro umělou inteligenci

Engine simulující chování ostatních entit. Obsahuje předem naprogramované pohyby a činnosti přímo neovládaných postav a systémů. Jejich interakci mezi sebou a reagování na vnější podněty.

4.5. Grafický engine

Grafický engine vykresluje veškerý vizuální výstup. Jednotlivé objekty vytvořené v externích programech se přidávají do scény, nastaví se osvětlení a další efekty. Poté se scéna vyrenderuje. Stejně jako u fyzikálního engineu se i grafické enginey dělí na dvě části, non real-time rendering a real-time rendering .

4.5.1. Non real-time rendering

Vykreslování neprobíhá v reálném čase, ale vytvořená scéna je vykreslována postupně a každý pixel je vypočítáván zvlášť. Tím se dosahuje vyšší kvality renderu. Toto vykreslování se využívá při výstupu z grafických programů nebo při tvorbě animací a filmů.

4.5.2. Real-time rendering

Při renderování v reálném čase se tvoří výstup několikrát za vteřinu. Díky tomu lze vytvářet plynulý výstup, na který může uživatel okamžitě reagovat. Tento postup je výpočetně náročnější a nemusí být dosaženo tolik detailů jako u předchozího způsobu renderování.

4.6. Efekty grafických enginů

4.6.1. HDR

High Dynamic Range rendering (obrázek 19) neboli vysoce dynamický rozsah zobrazení je jednou z pokročilých funkcí grafických enginů. Každý pixel je zastoupen intenzitou tří základních barev, červené, zelené a modré. Při základním rozsahu je hodnota intenzity definována od nuly do 255 stupňů. Pro HDR je barevný pracovní prostor větší tím, že každý stupeň podrobněji definován. Při tomto způsobu renderingu se počítá barva jednotlivých pixelů v závislosti na okolním osvětlení a tím vzniká realističtější obraz. Díky HDR lze tedy na výstupu dosáhnout věrohodnějších barev a více světelných efektů. U HDR videí a fotografií se aplikuje metoda pořizování více verzí záznamu. Jeden podexponovaný a jeden přeexponovaný a ten se poté softwarově spojí. Tuto technologii podporují grafické karty výrobce ATI od řady R300 a u výrobce nVidia od řady GeForce 5000. Nevýhodou je vyšší náročnost na výpočet výstupu.



Obrázek 19 - Porovnání renderu s technologií HDR

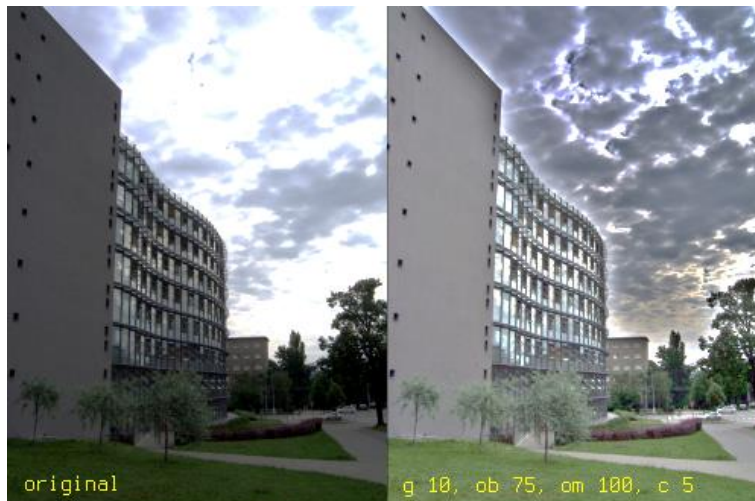
Zdroj: http://images.bit-tech.net/content_images/2004/11/farcry_patch13_eval/6800u-5-hdr0-hdr7-1.jpg

4.6.2. Tone mapping

Tone mapping je proces, při kterém se převádí HDR výstup zpět na standardní rozsah (obrázek 20). Zachovávají s při tom výhody vysokého rozsahu. Obraz zůstává kvalitnější a barvy zřetelnější i přes to, že monitor neumí HDR zobrazit. Existují dvě metody převodu:

- *Globální:* Obraz se bere jako celek a pixely jsou převedeny nezávisle na svém okolí. Tato metoda je rychlejší a méně náročná, ale výsledek nemusí být kvalitní.

- *Lokální:* Při převodu se bere obraz po částech a každý pixel je převeden v závislosti na svém okolí. Tento způsob je pro výpočet náročnější, dosahuje se ale kvalitnějšího výstupu než při metodě globální. Někdy však výsledek může působit nepřirozeně a mohou se objevit nežádoucí efekty, například stínování okolo tmavých objektů.



Obrázek 20 - Nepřirozený efekt při použití tone mappingu.

Zdroj: <http://blog.tasuki.org/wp-content/uploads/2008/tone-mapping-1.png>

4.6.3. Motion Blur

Motion blur je efekt rozmazání k umocnění pocitu pohybu (obrázek 21). Čím se objekt pohybuje rychleji, tím je efekt mocnější. Typicky se využívá u závodních her. Pokud se efekt použije kvalitně, obraz vypadá pro lidské oko přirozeněji a dynamičtěji. U některých her se aplikuje rozmazání krajů obrazovky k navození pocitu tunelu a vyšší rychlosti, pro některé uživatele může být efekt nepřirozený.



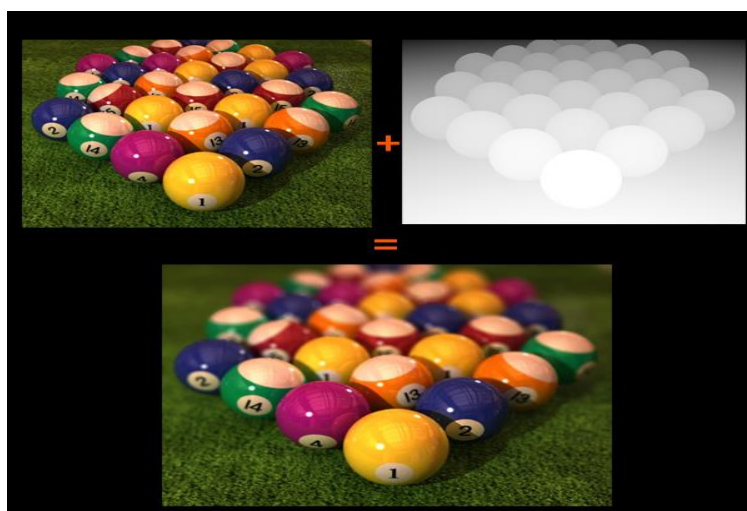
Obrázek 21 - Ukázka efektu motion blur

Zdroj: http://www.fx fx.net/wp-content/uploads/2010/02/motion_blur_effect_covwer1.jpg

4.6.4. Depth Of Field

V překladu hloubka ostrosti je efekt rozostření obrazu. Při snímání skutečného světa objektivem je část obrazu ostrá a zbytek rozmazaný. U počítačové grafiky se rozmazání aplikuje uměle kvůli realističtějšímu výsledku a většího pocitu hloubky obrazu. V dnešní době se využívají dva způsoby vytvoření efektu:

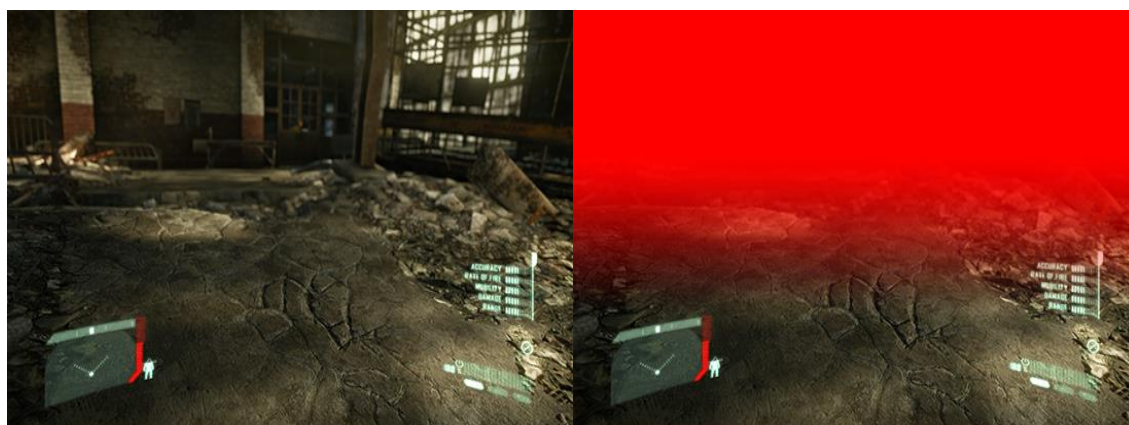
- *Z-Depth mapa*, která udává vzdálenost konkrétního objektu od kamery a míru jeho rozostření. Většinou je ve stupních šedi. Čím je barva tmavší, tím bude pixel více rozmazaný (obrázek 22). Výhodou je, že lze vytvořit postupné rozmazání. Problém nastává u průhledných objektů a u odrazů.



Obrázek 22 - Použití Z-Depth mapy pro vytvoření efektu hloubky ostrosti

Zdroj: <https://cms-assets.tutsplus.com/uploads/users/169/posts/22640/image/5.JPG>

- Druhým způsobem je přidat do scény desku z průhledného materiálu v určité vzdálenosti od kamery a na ní aplikovat efekt rozmazání. Rozmaže vše za deskou. Tento způsob je na provedení jednodušší a i na výkon méně náročný, ale má také nevýhody. Rozmazání za deskou je konstantní a vzniká ostrý přechod mezi rozmazaným a nerozmazaným prostředím. U pokročilých enginů lze použít gradientní přechod a tím zmenšit ostrost okraje desky (obrázek 23).

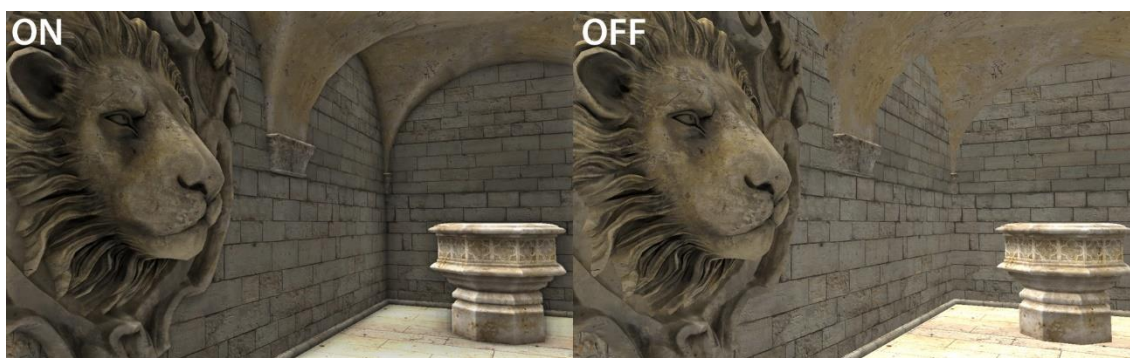


Obrázek 23 - Vytvoření efektu hloubky ostrosti pomocí desky s efektem rozmazání.

Zdroj: <http://www.svethardware.cz/graficke-enginy-her-a-realny-svet/18297-5/img/body-16.2B48.jpg>

4.6.5. Ambient Occulsion

Ambient Occulsion je způsob stínování, které přidává na realističnosti. Aplikuje se při osvětlení objektů globálním světlem a výpočtu zastíněním okolními objekty. Díky tomuto stínování lze lépe rozpoznat tvar jednotlivých předmětů. Výpočet funguje na principu určení několika bodů, ze kterých se vyšlou paprsky do okolí. Pokud paprsek narazí na některý z objektů, mírně změní jeho osvětlení. Objekt je pak tmavší nebo světlejší v závislosti na počtu paprsků, které na něj narazí. Tento efekt je poměrně náročný na výpočet a je za potřebí výkonnější grafický adaptér. Pokročilejší metodou výpočtu je technologie SSAO, která byla použita poprvé v roce 2007 ve hře Crysis. Pro každý pixel se v pixel shaderu zjistí hloubka okolních pixelů a spočítá se množství zastínění nejbližším okolím (obrázek 24).



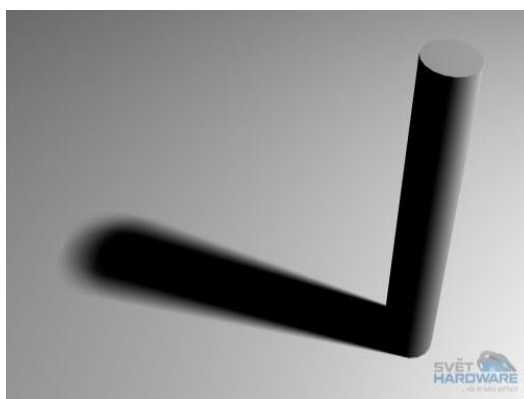
Obrázek 24 - Porovnání se zapnutým SSAO (vlevo) a bez SSAO (vpravo)

Zdroj: <http://media.moddb.com/images/engines/1/1/33/ssao1.jpg>

4.6.6. Výpočet stínů

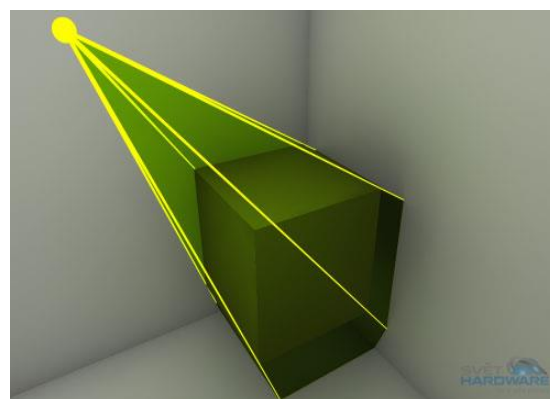
Pro výpočet základních stínů objektů se dnes využívají dva způsoby:

- *Shadow Map*: nejprve se scéna vyrendruje z pohledu světla. Tím vznikne textura stínů do níž se následně přidá objekt na který je stín vržen. K složitějším tělesům se vytváří stíny, které používají zjednodušenou geometrii. Na výsledné textury stínů lze aplikovat různé filtry a efekty (obrázek 26).



Obrázek 26 - Vytvoření stínů pomocí Shadow Map

Zdroj: <http://www.svethardware.cz/graficke-enginy-her-a-realny-svet/18297-2/img/body-6.2918.jpg>



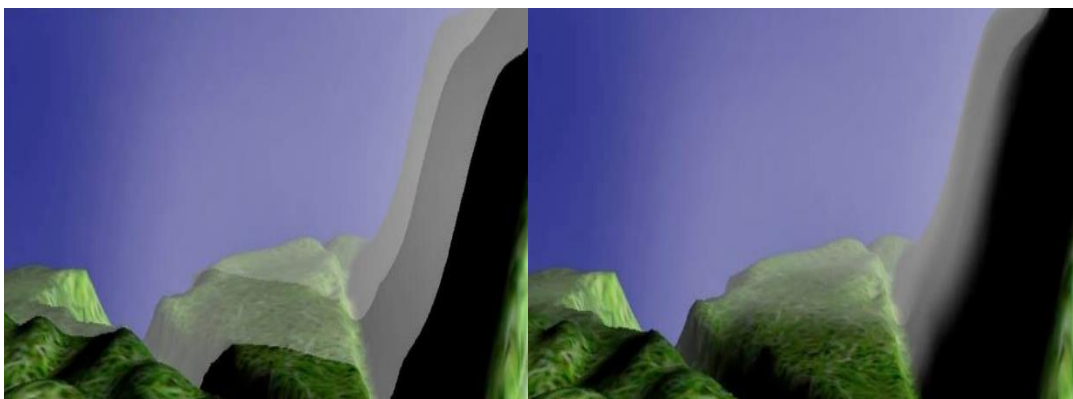
Obrázek 25 - Metoda stencil Shadow

Zdroj: <http://www.svethardware.cz/graficke-enginy-her-a-realny-svet/18297-2/img/body-8.489A.jpg>

- *Stencil Shadow*: pro vytvoření stínu se určí průsečíky mezi zdrojem a objektem. Poprvé byla tato metoda aplikována v enginu hry Doom3. Vykreslení stínů je dáno pomocí paprsků, které dopadají ze zdroje světla na hranu osvětleného objektu. Koncové body těchto paprsků tvoří plochu, která se vyplní a tím vznikne ostrý stín. U této metody se musí vykreslované plochy počítat pro každý objekt a zdroj světla zvlášť. To dělá tuto metodu náročnější. Lepší realističností se dá dosáhnout mírným posunutím zdroje světla a složením výsledných stínů (obrázek 25).

4.6.7. Vykreslování částicových systémů

Částice jsou malé objekty, které se vyskytují ve velkém množství na scéně a tím vytváří částicový systém. Tento systém se využívá například pro simulování kouře nebo u efektů explozí a jisker. Každá částice má danou dobu životnosti, která začíná v určeném místě a chová se dle předem definovaných instrukcí. Na každou částici se může mapovat textura a aplikovat další filtry. Částice může být ovlivněna různými dynamickými silami a gravitací. V roce 2007 přišla společnost nVidia s pokročilou technologií *Soft particle* pro vytváření efektu mlhy a kouře. Odstraňuje problém s ostrými přechody na okrajích systému (obrázek 27).



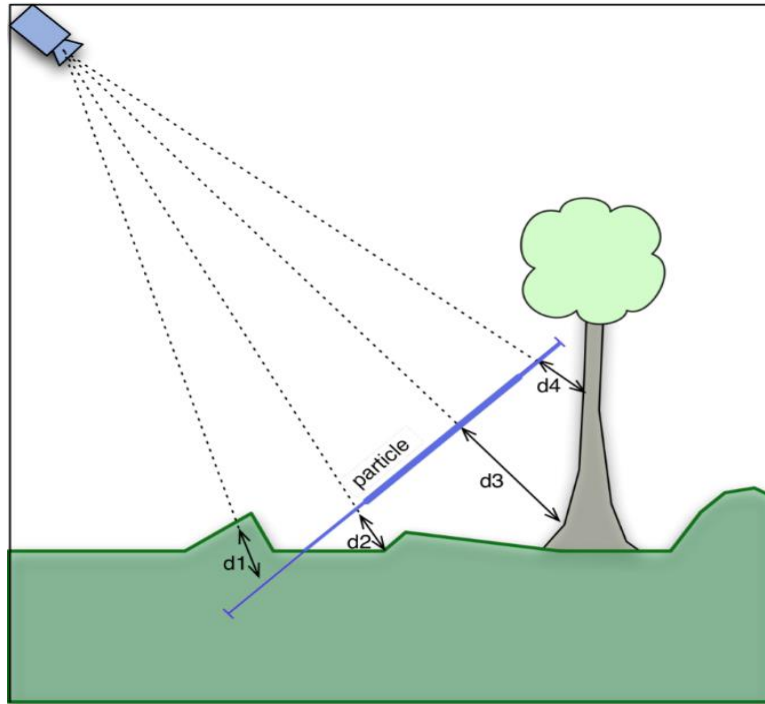
Obrázek 27 - Aplikování technologie *Soft particles*

Zdroj: Tristan Lorach. *Soft Particles* [online]. Leden 2007. [cit. 2015-11-28]. Dostupné z: http://developer.download.nvidia.com/whitepapers/2007/SDK10/SoftParticles_hi.pdf

Základem je měření vzdálenosti paprsků vyslaných z kamery za částicemi a určení míry průhlednosti následujícím vzorcem:

$$D = \text{saturate} (Z_{\text{scény}} - Z_{\text{částice}})$$

Přičemž *saturate* je funkce pro určení míry plnosti částic, čím je hodnota rozdílu vyšší, tím bude částice méně průhledná. $Z_{\text{scény}}$ znamená vzdálenost kamery od scény a $Z_{\text{částice}}$ vzdálenost částice od kamery (obrázek 28).



Obrázek 28 - Výpočet průhlednosti částic

Zdroj: Tristan Lorach. *Soft Particles* [online]. Leden 2007. [cit. 2015-11-28]. ISBN .

Dostupné z: http://developer.download.nvidia.com/whitepapers/2007/SDK10/SoftParticles_hi.pdf

Pokud paprsek za částicí pokračuje dál do nekonečna, zobrazí se částice standartním způsobem. Pokud se částice nachází za objektem, není zobrazena vůbec. V případě že se částice nachází před objektem, závisí míra průhlednosti na vzdálenosti od daného objektu. Čím je částice blíže objektu, tím je míra průhlednosti větší, až jsou částice přímo na objektu úplně průhledné.

5. Současné enginy

5.1. Licence pro volné využití softwaru - Open source

Open source je druh licencí pro software s volně dostupným zdrojovým kódem. Licence nesmí omezovat volné poskytování softwaru. Nesmí být vyžadován autorský honorář, ani žádný jiný poplatek. Program musí umožňovat distribuci ve zdrojovém kódu i zkompilevané formě. V případě, že je nějaká část bez zdrojového kódu, musí být tento kód k dispozici volně dostupný na internetu. Licence musí povolovat úpravy, tvorbu odvozených děl a musí umožňovat distribuci pod stejnými podmínkami jako u licence původního softwaru. Také by licence neměla mít omezení pro žádnou osobu či skupinu osob a pro oblast využití softwaru. Práva spojená se softwarem se musí stahovat na všechny osoby, jimiž je program dále šířen bez nutnosti provedení úprav v licenci.

5.1.1. MIT Licence

Licence vznikla na Messeschutském institutu. Software pod touto licencí lze libovolně šířit, upravovat i komerčně využívat pod podmínkou, že vždy musí být připojen text této licence, který pojednává o volném šíření softwaru:

„The MIT License (MIT)

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. “

Open Source Initiative. . [online]. 27.11.2015 [cit. 2015-11-27]. Dostupné z: <https://opensource.org/licenses/MIT>

5.1.2. GPL

General Public Licence je obecná licence pro software distribuovaný zdarma. Již třetí verze této licence byla vydána v roce 2007. Jakékoliv dílo odvozené z obsahu vydaného pod touto licencí musí být opět pod licencí GPL. Software lze upravovat, kopírovat a dál šířit jak zdarma tak za libovolnou cenu. Šedesát procent veškerého open source softwaru je vydáváno právě s touto licencí. Její celé znění se nachází na adrese: www.gnu.org/licenses/gpl.html

5.2. Cryengine

5.2.1. Licencování

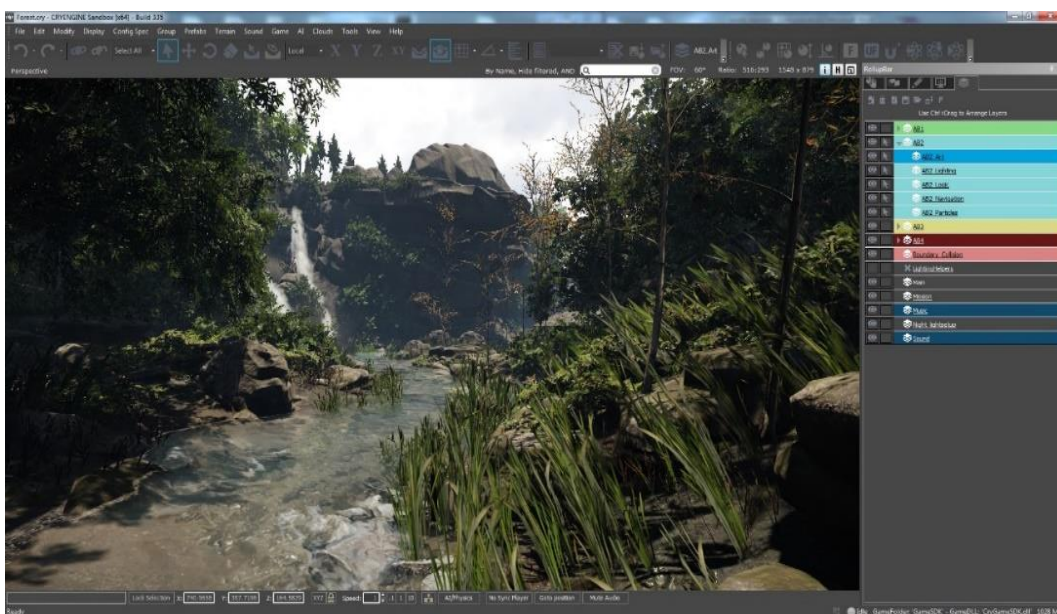
Pro nekomerční účely je engine vydáván ve speciální verzi zdarma. Pro menší komerční projekty je cena stanovena na 9,90 Euro měsíčně. Ta se platí přes internetový obchod Steam formou předplatného. Pro velké tituly je možnost si pořídit licenci na dobu neurčitou kde cena není zveřejněna. Pohybuje se však okolo jednoho milionu dolarů. Zdrojový kód není veřejný a nelze jej upravovat.

5.2.2. Historie a vývoj

První verze tohoto enginu byla využita ve hře Far Cry, která byla vydána 23. března 2004. Následně byly vydány ještě další čtyři hlavní verze. Engine byl dříve určen pouze pro vývoj komerčních titulů a v srpnu 2011 byla vydána verze pro nekomerční účely, která je doposud poskytována zdarma. Ta byla naposledy aktualizována v prosinci 2014. Komerční verze je pravidelně aktualizována.

5.2.3. Programovací jazyk, vývojové prostředí

Při programování obsahu lze použít jazyk C++ a pro skripty jazyk LUA. Součástí vývojového balíku je řada editorů, ve kterých lze tituly vytvářet bez nutnosti psaní zdrojového kódu. Pro vytváření scén slouží Sandbox Editor (obrázek 29). K dispozici jsou také nástroje pro tvorbu skriptů, objektů a animací. Editory jsou navrženy ve stylu WYSIWYP, What You See Is What You Play a hra je tak aktivní již při vytváření a lze vše hned vyzkoušet v praxi.



Obrázek 29 - Ukázka editoru scén v CryEnginu

Zdroj: http://cryengine.com/assets/images/license-detail/CE_02.jpg

5.2.4. Podporované systémy

Kromě OS Windows a Mac podporuje vývoj pro Android a iOS. Cryengine je také oficiálně licencovaný pro Xbox ONE a PlayStation 4. Součástí podpory jsou všechny verze DirectX včetně poslední verze 12.0 a OpenGL.

5.2.5. Komplexnost

Cryengine je kompletní balík obsahující vlastní fyzikální engine simulující vítr, exploze, gravitace i kolize. Součástí jsou také modely chování nehratelných postav a editor pro tvorbu audia.

5.2.6. Podporované formáty

Vlastní modely lze vytvářet v programech 3dmax a Maya. K dispozici je rozšíření, které umožňuje exportování v kompatibilních formátech cgf (geometrické modely), chr (modely postav) a skin (kůže). Editor sandbox pracuje s materiály ve vlastním formátu mtl. Pro textury lze použít formát tif a dds.

5.2.7. Vlastnosti

Cryengine je jeden z nejlepších a nejpokročilejších mezi současnými enginey. Díky pokročilé práci se světlem a stíny, realistické vodě, vegetaci a dalším efektům renderovaný obraz vypadá téměř realisticky.

5.2.8. Podpora vývojářů a komunita

V případě zakoupení plné licence na dobu neurčitou garantují vývojáři plnou podporu. Existuje také řada aktivních komunitních fór, tutoriálů a video návodů.

5.2.9. Tituly využívající enginey

Tento engine je převážně využíván u her žánru FPS, kdy je hra zobrazena z pohledu hráče. Doposud ale nebyla na tomto enginu vydána žádná větší hra s automobilovou tematikou. Mezi nejznámější hry patří série Crysis. Každý díl je vždy brán jako zátěžový test na výkon PC a nabízí využití nejnovějších technologií a funkcí v oblasti grafiky. V současné době se vyvíjí česká hra Kingdom Come: Deliverance (obrázek 30).



Obrázek 30 - Ukázka ze hry Kingdom Come: Deliverance

Zdroj: <http://www.play3.de/wp-content/gallery/kingdom-come-deliverance-060715/kingdom-come-deliverance-bild-10.jpg>

5.2.10. Odkaz na stránky engineu:

<http://www.cryengine.com/>

5.3. Unreal Engine

5.3.1. Licencování

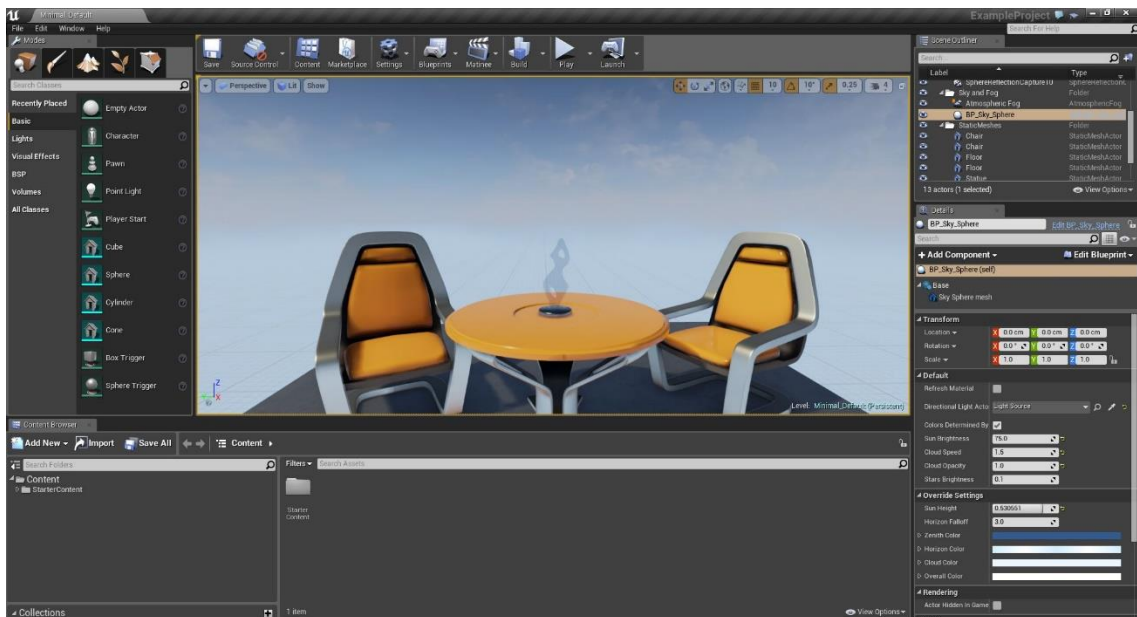
Tento engine je kompletně zdarma i pro komerční účely. Pokud ovšem příjmy z vyvinutého softwaru přesáhnou tři tisíce dolarů (zhruba 70 tisíc korun), je vývojář povinen ze zisku odvádět 5% společnosti Epic Games. Zdrojový kód engine je po registraci na stránkách Unreal engine ke stažení zdarma.

5.3.2. Historie a vývoj

Engine byl poprvé využit roku 1998 ve hře Unreal. Přestože byl vyvíjen primárně pro žánr FPS (hra z pohledu hráče), tak se využil i v dalších žánrech. Zejména u RPG (role playing game) kde hráč ovládá hrdinu, která interaguje s okolním světem a prochází příběhem.

5.3.3. Programovací jazyk, vývojové prostředí

Unreal engine pracuje v jazyku C++, součástí vývojového balíku jsou také editory ve kterých je možné většinu věcí vytvořit bez nutnosti přímého psaní zdrojového kódu. Vývojové prostředí obsahuje editory scény (obrázek 31), modelů a materiálů. K dispozici je také blueprint editor, ve kterém lze vytvářet skripty. Dále také nástroje pro tvorbu chování postav, editaci uživatelského prostředí a ovládání audia. K dispozici je obchod, odkud lze získávat již vytvořený obsah (<https://www.unrealengine.com/marketplace>).



Obrázek 31 - Level editor Unreal engine. Při jeho používání lze vytvářet scény bez nutnosti psaní zdrojového kódu. Zdroj: <https://docs.unrealengine.com/latest/images/GettingStarted/SubEditors/LevelEditorImage.jpg>

5.3.4. Podporované systémy

Vyvíjet lze pro systémy Windows, konzole Playstation 4 a Xbox One, Android a iOS. Engine také podporuje vývoj pro virtuální reality, OpenGL 3.3, DirectX 10.0 a vyšší.

5.3.5. Komplexnost

Unreal engine je kompletní engine včetně fyzikálního a audio engine. K dispozici jsou i modely chování pro umělou inteligenci.

5.3.6. Podporované formáty

Engine podporuje importování modelů ve formátu obj a fbx. Pro importování textur lze využít bmp, float, pcx, png, psd, tga, jpg, exr, dds a hdr.

5.3.7. Vlastnosti

Poslední verze 4 podporuje DirectX 11.0 a 12.0. Nabízí pokročilé HDR (high dynamic rendering), pokročilý výpočet stínů a spoustu vizuálních efektů (obrázek 32). Součástí je i editor VFX ve kterém lze vytvářet detailní efekty ohně, kouře, prachu a dalších částicových efektů. Jako novinka je, že kromě široké možnosti nastavení efektů, umí částice přijímat a emitovat světlo do okolní scény.



Obrázek 32 - Ukázka vizuálních efektů Unreal engineu

Zdroj: <https://docs.unrealengine.com/latest/images/Resources/Showcases/Effects/EffectsIntro.jpg>

5.3.8. Podpora vývojářů a komunita

K dispozici je podrobná dokumentace a také video návody. Komunita je na oficiálních fórech aktivní. (<https://docs.unrealengine.com/latest/INT/index.html>)

5.3.9. Tituly využívající engine

Tento engine využívá řada známých titulů. Například série Mass Effect, Bioshock, či Borderlands .

5.3.10. Odkaz na stránky engineu

<https://www.unrealengine.com/what-is-unreal-engine-4/>

5.4. Ogre3D

5.4.1. Licencování

Engine je pod licenci MIT (kapitola 4.1.1) a lze jej používat bez omezení i ke komerčním účelům.

5.4.2. Historie a vývoj

Již v roce 1999 vznikl projekt DIMClass, který byl navržen jako nástroj pro snadnější práci s knihovnami Direct3D a postupným vývojem se stal multiplatformním. V únoru 2000 byl projekt oficiálně zaregistrován na serveru Sourceforge a byl pojmenován jako OGRE (object graphics rendering engine). Pro OS Windows poslední stabilní verze byla vydána v listopadu 2013.

5.4.3. Programovací jazyk, vývojové prostředí

Pro programování se používá jazyk C++ a engine nemá vlastní vývojové prostředí. Lze využít jakýkoliv prostředí schopného pracovat s jazykem C++. Sami vývojáři doporučují využívat Microsoft Visual Studio, případně volně dostupný Codeblocks s MinGW kompilátorem.

5.4.4. Podporované systémy

Ogre3D je multiplatformní a lze vyvíjet pro všechny verze OS Windows, Linux i Mac OS podporuje i mobilní zařízení Android, iOS i Windows Phone. Je kompatibilní s knihovnami DirectX i OpenGL.

5.4.5. Komplexnost

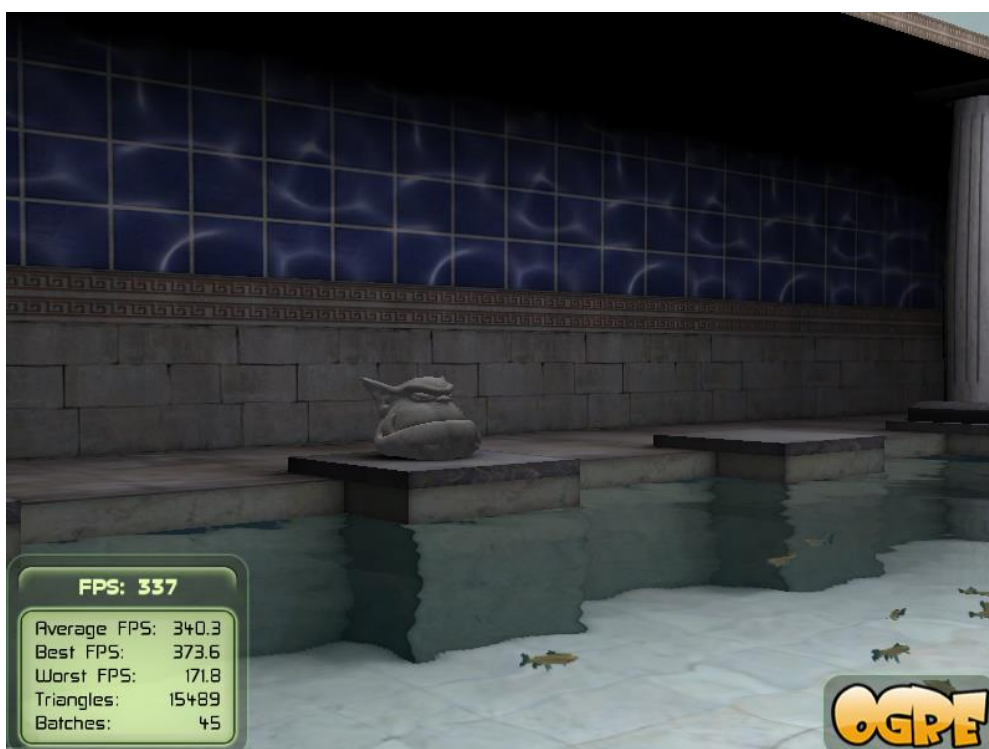
V základní části je Ogre pouze grafický engine. Díky otevřenosti lze další součásti snadno implementovat. Sami vývojáři nabízejí spoustu rozšíření, díky kterým lze vytvořit komplexní herní engine s podporou fyziky a umělé inteligence. Navíc lze použít části vytvořené ostatními uživateli, které jsou dostupné přímo na diskuzních fórech i jinde na webu (např: <http://www.ogre3d.org/developers/addons>). Dále je možné přidávat různé rozšiřující funkce. Například editor terénu, nebo externí aplikace, jako je program pro export modelů *3ds2mesh* a scén *3dssceneexporter* z 3DMax do formátů podporovaných tímto enginem.

5.4.6. Podporované formáty

Engine podporuje širokou škálu operací pro tvorbu textur a práci s materiály. Textury mohou být ve formátu PNG, JPG, TGA, BMP a DDS. Ty mohou být navíc v reálném čase upravovány.

5.4.7. Vlastnosti

Součástí je také RTSS (Run Time Shader Systém). Tento systém automaticky ovládá shadery na základě jednoduchého programu. Výhodou je, že program pro shadery tvoří samostatnou část kódu a lze jej tedy používat vícekrát. Lze použít programy již vytvořené, které jsou k dispozici na fórech Ogre3D. Výstup na více monitorů není v základu podporován, ale na fórech lze nalézt několik řešení, kdy lze pomocí rozšíření jak výstup rozšířit. Podle ohlasů je problém se stabilitou a kompatibilitou. Další možností je využití technologie od společnosti ATI, Eyefinity. Ta podporuje výstup na více obrazovek přímo v grafické kartě. Podmínkou je ale kompatibilita karty AMD.



Obrázek 33 - Ukázka z engine Ogre 3D
Zdroj: autor

5.4.8. Podpora vývojářů a komunita

Díky své otevřenosti je engine vytvářen s pomocí prostředí komunity. Ta je poměrně aktivní a přímo na stránkách engine je celkem rozsáhlé diskuzní fórum. Je také k dispozici portál Wiki obsahující spoustu informací s návody. Z portálu BitBucket (<https://bitbucket.org/sinbad/ogre/>) lze získat kompletní zdrojový kód, který se neustále upravuje a lze ho poté zkompileovat pomocí programu CMake.

5.4.9. Tituly využívající engine

Ogre lze použít v simulátorech nebo jako součást herních engineů. Mezi nejznámější herní tituly patří RPG Torchlight a Venetica nebo automobilové závody Stunt Rally (obrázek 34).



Obrázek 34 - Ukázka ze hry Stunt Rally

Zdroj: <http://stuntrally.tuxfamily.org/uploads/images/ver26.jpg>

5.4.10. Odkaz na stránky engineu

<http://www.ogre3d.org/>

5.5. NeoAxis3D

5.5.1. Licencování

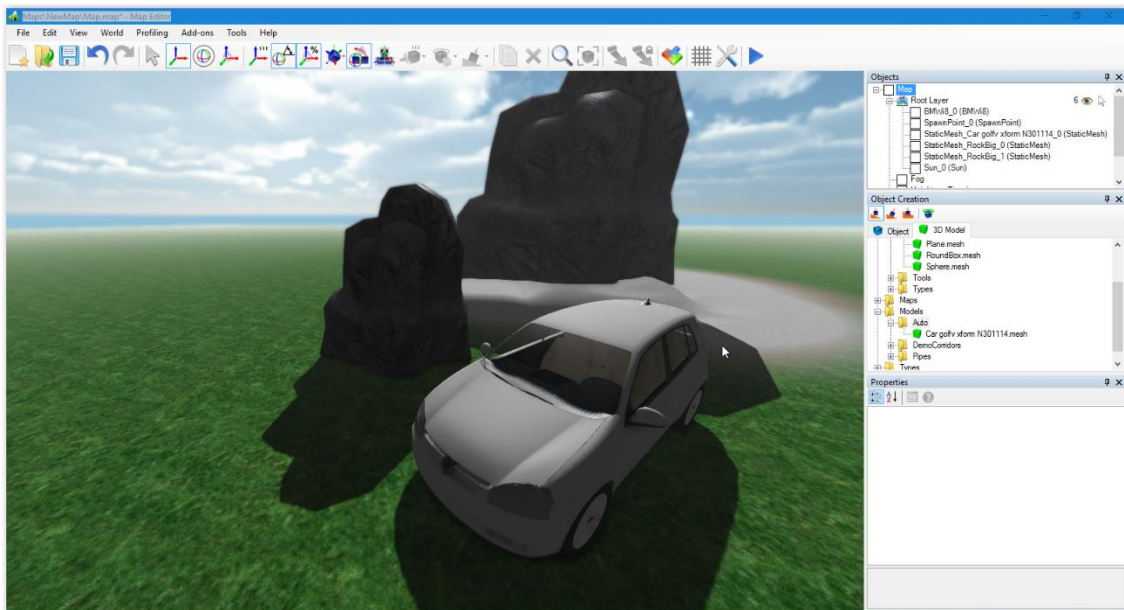
Engine je k dispozici ve třech verzích. Verzi zdarma lze použít i ke komerčním účelům a obsahuje všechny nástroje pro využívání všech funkcí engineu. Verze professional stojí kolem 700 dolarů a oproti verzi zdarma obsahuje navíc zdrojové kódy nástrojů pro editaci. Autoři zaručují přímou podporu po dobu jednoho měsíce. Za 2900 dolarů lze pořídit verzi s přímou podporou po dobu jednoho roku a kompletním zdrojovým kódem.

5.5.2. Historie a vývoj

První stabilní verze byla vydána po šesti letech vývoje v červnu roku 2011. V současné době je nejnovější verze 3.3 z dubna 2015.

5.5.3. Programovací jazyk, vývojové prostředí

Obsahuje všechny potřebné nástroje pro tvorbu pokročilých 3D projektů. Základem je editor zdrojů, jako jsou 3D modely, textury nebo materiály. Editor je dobře přehledný a lze vytvářet i komplexní modely včetně materiálů a deformačního modelu. K dispozici je editor map, ve kterém se vytvářejí scény a mapy včetně editace terénu (obrázek 35). Engine lze editovat přímo na úrovni zdrojového kódu v jazyce C++.



Obrázek 35 - Ukázka z editoru scén NeoAxis engine
Zdroj: Autor

5.5.4. Podporované systémy

Vyvíjet lze aplikace pro OS Windows, Mac, a Linux. Engine podporuje jak OpenGL, tak Direct3D.

5.5.5. Komplexnost

Engine obsahuje fyzikální systém pracující na moderním enginu nVidia PhysX a také realistický jízdní model pro vozidla a pohyb postav.

5.5.6. Podporované formáty

Pro modely se doporučuje formát FBX a collada. Součástí engine je plugin do softwaru 3Dmax a Maya pro přímé exportování modelů, materiálů, animací i scén do formátů používaných engine.

5.5.7. Vlastnosti

Neoaxis je moderní engine, ve kterém lze vyvíjet jak hry různých žánrů, tak i vědecké simulace. Obsahuje spoustu vytvořených efektů a pomocí programovatelných shaderů lze vytvořit vlastní. Součástí balíku je již vytvořené demo, kde lze funkce otestovat.

5.5.8. Podpora vývojářů a komunita

Na stránkách engine je k dispozici dokumentace s podrobnými návody jak engine používat a vytvářet vlastní projekty. Komunita na oficiálním fóru je aktivní. Díky návodům lze jednoduše vytvářet základní scény s použitím již připravených modelů. Importace vlastních modelů je otázka několika kroků. Dále lze k modelům přidat vlastní deformační modely či animaci pohyblivých částí.

5.5.9. Tituly využívající engine

Na tomto engine nebyly zatím vydány velké komeční tituly. Byl však použit například již pro simulátor motocyklu (obrázek 36) a jako výukový program péče o postižené lidi.



Obrázek 36 - Simulátor motocyklu v autoškole Yamaha, Thajsko
Zdroj: <http://www.neoaxis.com/files/emvideo-youtube-fohtBCDs0Q.jpg>

5.5.10. Odkaz na stránky engine

<http://www.neoaxis.com/>

5.6. Zhodnocení uvedených enginů

Při testování jednotlivých zástupců nejvíce vyhovoval engine *NeoAxis3D*. Při použití v praktické části této práce splňoval potřebné požadavky. Je k dispozici zdarma a podporuje pokročilé grafické efekty. Nabízí intuitivní editor, ve kterém lze snadno vytvářet obsah použitelný pro další využití. Další grafický engine *Ogre3D* rovněž vytváří kvalitní grafický výstup s pokročilými efekty, nicméně mne odrazovala absence vlastního vývojového prostředí a tím byla tvorba náročnější. Výhodou tohoto enginu je možnost implementace do stávajícího enginu, který se ve vozidlovém simulátoru využívá. U *Cryengine* je nevýhodou zpoplatnění v případě využití pro komerční účely. Oproti tomu nabízí velmi kvalitní grafický výstup a pokročilé funkce. Další výhodou je podpora virtuální reality, která by se dala u simulátorů dobře využít.

5.7. Další dostupné enginy

Níže jsou uvedeny další enginy, které jsem v této práci podrobněji netestoval, protože jsem je shledal jako méně vhodné pro použití ve vozidlovém simulátoru.

5.7.1. Irrlicht engine

Irrlicht engine je s otevřeným zdrojovým kódem. Je napsaný v jazyce C++ a existují i neoficiální verze pro další jazyky jako je Java, Python, Lua a další. Lze jej využívat pod OS Windows, Mac OSx i Linux. Jedná se pouze o grafický engine a neobsahuje vlastní vývojové prostředí. Odkaz na stránky enginu: <http://irrlicht.sourceforge.net/>

5.7.2. Panda3D

Panda3D je herní engine pro vývoj her a softwaru v jazyce Python a C++. Má otevřený zdrojový kód a je volný pro osobní i komerční užití. Neobsahuje vlastní vývojový editor. Software se vytváří ve vývojovém rozhraní Visual Studio 2010. Panda3D obsahuje vlastní engine pro simulaci fyziky, základní umělou inteligenci a lepší tvorbu částicových systémů. Tento engine vyvinula společnost Disney a využívá jej převážně pro tvorbu svých titulů. V porovnání s ostatními enginy byla kvalita renderování nižší. Odkaz na stránky enginu: <https://www.panda3d.org/>

5.7.3. Horde3D

Horde3D je malý grafický 3D engine s otevřeným kódem. Vývojáři se zaměřují, aby byl engine co nejjednodušší a zároveň umožňoval využití pokročilých efektů při renderování. S enginem lze pracovat v libovolném editoru pro programovací jazyk C#. Mezi další neoficiální jazyky ve kterých lze vyvíjet patří java, pascal a lua. Poslední novinky na internetových stránkách enginu jsou z roku 2013 a naposledy byl engine upravován více než před rokem. Odkaz na stránky enginu: <http://www.horde3d.org/>

6.Závěr

Za posledních 30 let vývoje grafických karet došlo k velkému pokroku. Na konci osmdesátých let byly počítače schopny vykreslit pouze text a několik základních monochromatických tvarů. S příchodem prvních grafických akceleratorů bylo možné začít vyvíjet pokročilejší hry a zlepšovat grafický software. Důležitým milníkem pro vývoj her byl grafický čip akcelerující 3D grafiku. V druhé polovině devadesátých let byly na trhu s těmito čipy tři hlavní výrobci. Dnes většina grafických karet obsahuje procesor od jednoho ze dvou výrobců. Společnost AMD a nVidia spolu soupeří o co největší výkon procesoru a nejnižší cenu. Díky moderní výrobní technologii, lze očekávat další nárůst výkonu a tím stálé zlepšování grafického výstupu, který je pro simulátory přibližující realitu velmi potřebný.

Při psaní této práce jsem získal spoustu nových informací a zkušeností, které bych v budoucnu rád využil v další fázi studia a profesi.

7. Použité zdroje

7.1. Literatura

- [1.] TOMAS AKENINE-MÖLLER, Eric Haines. Real-time rendering. 3rd ed. Wellesley: A K Peters, 2008. ISBN 9781439865293.
- [2.] KOČIČKA, Pavel. *CMS - Color Managment System*. IV. přepracované vydání. GRAFIE CZ, 2004.

7.2. Internetové zdroje

- [1.] Historie společností vyrábějících 3D čipy - díl III.: ATi Technologies. *PCTuning* [online]. [cit. 2015-11-30]. Dostupné z: <http://pctuning.tyden.cz/component/content/4900/4900?task=view&start=1>
- [2.] Historie společností vyrábějících 3D čipy - díl I. : 3Dfx. *PCTuning* [online]. [cit. 2015-11-30]. Dostupné z: <http://pctuning.tyden.cz/hardware/graficke-karty/4743-historie-spolecnosti-vyrabejicich-3d-cipy-dil-i-3dfx?start=1>
- [3.] Historie společností vyrábějících 3D čipy - díl IV.: nVidia. *PCTuning* [online]. [cit. 2015-11-30]. Dostupné z: <http://pctuning.tyden.cz/hardware/graficke-karty/5240-historie-spolecnosti-vyrabejicich-3d-cipy-dil-iv-nvidia>
- [4.] The History of the Modern Graphics Processor. *Techspot* [online]. [cit. 2015-11-30]. Dostupné z: <http://www.techspot.com/article/650-history-of-the-gpu/>
- [5.] 3D technologie: Anti-Aliasing. *Svět Hardware* [online]. [cit. 2015-11-30]. Dostupné z: <http://www.svethardware.cz/3d-technologie-anti-aliasing/8871>
- [6.] Grafické enginy her a reálný svět. *Svět Hardware* [online]. [cit. 2015-11-30]. Dostupné z: <http://www.svethardware.cz/graficke-enginy-her-a-realny-svet/18297>
- [7.] Tristan Lorach. Soft Particles [online]. Leden 2007. [cit. 2015-11-28]. ISBN Dostupné z: http://developer.download.nvidia.com/whitepapers/2007/SDK10/SoftParticles_hi.pdf
- [8.] The Open Source Definition. *Open Source Initiative* [online]. [cit. 2015-11-30]. Dostupné z: <https://opensource.org/osd>

8. Seznam obrázků

Obrázek 1 – Karta osazená čipem Mach32	7
Obrázek 2 - Diamond Viper V330 osazená čipem nVidia RIVA 128	8
Obrázek 3 – Asus RIVA TNT 2 VIVO	9
Obrázek 4 - nVidia GeForce 256	10
Obrázek 5 - Propojení dvou karet Voodoo 2 do režimu SLI	11
Obrázek 6 - Voodoo 5 6000	12
Obrázek 7 - Karta Tekram Capture TV	13
Obrázek 8 - Grafické znázornění vývoje 3D akceleratorů	14
Obrázek 9 - Karta Creative GeForce 4 Ti 4400	15
Obrázek 10 - Ati Radeon R100	16
Obrázek 11 - Příklad číslování řady 300 grafických karet AMD	18
Obrázek 12 - Ukázka aplikování Spritů do scény	19
Obrázek 13- Schéma vykreslovacího řetězce moderních GPU.	21
Obrázek 14 - Zobrazení funkce Anti aliasing.	22
Obrázek 15 - Ukázka modelu složeného ze sítě polygonů a nanesenou texturou	22
Obrázek 16 - Metoda určování pixelu pomocí	23
Obrázek 17 - Ukázka rozdílu mezi trilineárním a anizotropním filtrováním.	24
Obrázek 18 - Zobrazení entity ve dvou objektech	25
Obrázek 19 - Porovnání renderu s technologií HDR	27
Obrázek 20 - Nepřirozený efekt při použití tone mappingu.	28
Obrázek 21 - Ukázka efektu motion blur	28
Obrázek 22 - Použití Z-Depth mapy pro vytvoření efektu hloubky ostrosti.	29
Obrázek 23 - Vytvoření efektu hloubky ostrosti pomocí desky s efektem rozmazání ...	29
Obrázek 24 - Porovnání se zapnutým SSAO (vlevo) a bez SSAO (vpravo)	30
Obrázek 25 - Metoda stencil Shadow	30
Obrázek 26 - Vytvoření stínů pomocí Shadow Map	30
Obrázek 27 - Aplikování technologie Soft particles	31
Obrázek 28 - Výpočet průhlednosti částic	32
Obrázek 29 - Ukázka editoru scén v CryEnginu	34
Obrázek 30 - Ukázka ze hry Kingdom Come: Deliverance	35
Obrázek 31 - Level editor Unreal enginu	36
Obrázek 32 - Ukázka vizuálních efektů Unreal enginu	37
Obrázek 33 - Ukázka z enginu Ogre 3D	39
Obrázek 34 - Ukázka ze hry Stunt Rally	40
Obrázek 35 - Ukázka z editoru scén NeoAxis engine	41
Obrázek 36 - Simulátor motocyklu v autoškole Yamaha, Thajsko	42