

# Glava 1

## OSNOVNE KOMPONENTE GRAFIČKE OBRADNE

Računarska grafika nije potpuno originalna tema, jer za definisanje i rešavanje problema koristi neke već uspostavljene tehnike kao što su geometrija, algebra, optika i ljudska psihologija. Geometrija se upotrebljava da obezbedi okvir za opisivanje 2D i 3D prostora, dok se algebarske tehnike koriste za definisanje i evaluiranje jednakosti vezanih za određeni prostor. Nauka o optici postavlja modele za opisivanje ponašanja svetlosti, dok ljudska psihologija nudi modele za ljudsku viziju i percepciju boja.

Iako pomenuti opseg tema navodi korisnika da pomisli da je reč o kompleksnoj nauci, računarsku grafiku je, u stvari, relativno lako razumeti, zahvaljujući primeni posebnih tehnika koje se koriste za simulaciju virtuelnog sveta oblika, za skaliranje, boje i pokrete iako se te tehnike zasnivaju na nekim osnovnim tehničkim principima. Drugačije se i ne može očekivati kada se ima u vidu da računarska grafika koristi u tako specifičnim oblastima kao što su modelovanje molekula ili medicinska dijagnostika mada se u ovakvim slučajevima pomenuti osnovni principi unapređuju posebnim tehnikama. Postoje aspekti animacije na računaru koji zahtevaju posebno matematičko i naučno znanje, ali to ne bi trebalo korisnike da spreči da eksperimentišu sa takvim sistemom. Interfejs između korisnika i računara se neprekidno razvija u pravcu izolovanja korisnika od irelevantnih unutrašnjih kompleksnosti samog sistema.

### 1.1 Hijerarhija grafičkog softvera

Postoje raznorazne podele grafičkog softvera, od kojih je svaka podela priča za sebe. Podela grafičkih alata koja ovde sledi je jedna od mogućih, koja se rukovodi kriterijumima postojećih aplikacija.

#### 1.1.1 Jednostavnije grafičke biblioteke

Grafičke biblioteke omogućavaju pristup grafičkim komandama ili funkcijama. Grafičke komande su definisane u nekom programskom jeziku (C, Java, C++, ili neki drugi), a

mogu da se primenjuju na različitim platformama. Jednostavnije grafičke biblioteke, odnosno grafičke biblioteke nižeg nivoa predstavljaju softverski interfejs za hardver. Treba napomenuti da svaki grafički alat pripada nekoj jednostavnijoj grafičkoj biblioteci. U ove jednostavnije grafičke biblioteke spadaju:

- **Direct3D (DirectX)** predstavlja set nezavisnih API interfejsa za Windows platforme. API je akronim od engleskog naziva za programski interfejs aplikacije. Biblioteka DirectX predstavlja grupu API interfejsa, s tim što treba napomenuti da tu pripada i biblioteka Direct3D.
- **GKS (GKS-3D)** je biblioteka za 2D grafiku sa ANSI/ISO standardom koji definiše metode za pravljenje slika na rasterskim ili vektorskim izlaznim uređajima. Biblioteka GKS-3D je sastavni deo biblioteke GKS koja omogućava izradu 3D objekata.
- **Mesa** je 3D biblioteka sa API interfejsom koji je sličan API interfejsu biblioteke OpenGL. Njeno zaduženje je da emulira OpenGL komande na platformama koje ne podržavaju OpenGL. Mesa može da se upotrebljava kao direktna zamena za OpenGL.
- **MicroSystem 3D Graphic Tools** predstavlja 3D biblioteku koja može da se upotrebi za razvoj programskih jezika Visual Basic i Visual C++.
- **OpenGL** predstavlja primarno okruženje za razvoj interaktivnih 2D i 3D grafičkih aplikacija. OpenGL je akronim od engleskog naziva za otvoreni grafički jezik.
- **OpenGL for Java (GL4Java)** mapira kompletan OpenGL 1.2 API i kompletan GLU 1.2 API za Javu i integriše sve njihove funkcije u Javu kako bi mogle da se koriste u ovom okruženju.
- **PHIGS** je standardni 3D grafički API. Trebalo bi napomenuti da su OpenGL i Direct3D grafički standard danas, koji dolaze sa većom podrškom i hardvera i softvera. PHIGS se koristi za kompleksnije 3D aplikacije u tehničkim i komercijalnim oblastima, uključujući proizvodnju, industrijski dizajn, tehničku analizu i naučnu vizualizaciju.
- **QuickDraw3D** je biblioteka novijeg datuma koja je ugrađena u QuickTime od strane firme Apple Computer.
- **XGL** je biblioteka koju je razvila firma SUN Microsystems. Ova biblioteka podržava veliki broj API interfejsa, ali i biblioteke, kao što su GKS i PHIGS.

## 1.2 Grafički API interfejs

Postavlja se pitanje šta je API. API predstavlja skup rutina koje aplikativni program koristi da upravlja izvršavanjem procedura od strane operativnog sistema računara. Podela programskih interfejsa aplikacija biće obavljena prema proizvođačima.

### 1.2.1 Jednostavni API

U grupu jednostavnih API interfejsa spadaju: **IrisGL**, **OpenGL** i **MesaGL**.

**IrisGL** je jednostavniji API koji je ranije bio poznat pod nazivom **Iris 3D API**. Ovaj grafički interfejs sadrži u sebi 390 funkcija koje su implementirane u grafičkom modu. Naslednik ovog API-ja je OpenGL.

**OpenGL** je možda najpoznatiji grafički API. Uvela ga je 1992. godine firma Silicon Graphics, kao opšti sistem za CAD i 3D API za X-terminale zasnovane na operativnom sistemu Unix. OpenGL se razvio iz SGI-jeve vlasničke grafičke biblioteke IrisGL. U prvo vreme, upotreba OpenGL bila je ograničena na poslovne primene, kao što su industrijsko, interno i mehaničko projektovanje, kao i na statističke i naučne analize. Međutim, API se utemeljio u oblasti računarskih igara. Svi glavni PC paketi za 3D animaciju, pa čak i neki od jeftinijih proizvoda te vrste, sada podržavaju OpenGL.

OpenGL API je projektovan da može da podrži napredne grafičke tehnike za realističan prikaz, kao što su preslikavanje tekstura (sposobnost da se primeni slika na grafičku površinu), uklanjanje nazubljenosti, transparentnost, zamagljivanje, osvetljavanje (sposobnost da se proračuna obojenost površine kada su na nju primenjeni različiti modeli osvetljavanja iz jednog ili više izvora svetlosti), glatko senčenje (sposobnost da se proračunaju efekti senčenja kada svetlost pada na površinu pod uglom, što rezultuje finim razlikama u nijansama boje na površini), zamagljivanje pokreta i transformacija za modelovanje (sposobnost promene mesta, veličine i perspektive objekta u trodimenzionalnom koordinatnom prostoru).

Skup njegovih osobina je sličan onome koji ima Direct3D, ali je OpenGL API nižeg nivoa od svog rivala, obezbeđujući vrlo fino upravljanje osnovnim elementima 3D scene, kao što su informacije o tačkama i trouglovima. Aplikacija u OpenGL mora da obezbedi sve geometrijske informacije za svaku od primitiva (tačka, linija ili trougao) u sceni, kao i za efekte koji će se primeniti na primitive (boja, transparentnost, zamagljivanje

itd.). Nivo upravljanja koji se daje programerima je glavni činilac koji stoji iza tvrdnje da je sa OpenGL API interfejsom mnogo lakše praviti aplikacije nego sa Direct3D, kao i da je to mnogo pouzdaniji proizvod na različitim hardverskim platformama.

U suštini, postoje dva nivoa za hardverski ubranu podršku OpenGL-u. To su ICD (instalabilni klijentski upravljački programi) koji ubrzavaju osvetljavanje, transformacije i rasterizaciju i MCD (mali klijentski upravljački programi) koji podržavaju samo rasterizaciju. Dok su MCD lakši za prodavce hardvera, ICD nude bolju performansu.

**MesaGL** je grafička biblioteka sa API interfejsom i veoma je slična OpenGL-u. Ovaj API koristi sintaksu OpenGL-a, s tim što je za to bila potrebna dozvola firme Silicon Graphics, Inc. (SGI).

## 1.2.2 Kompleksniji API

U ovu grupu grafičkih programskih interfejsa aplikacija (API-ja) spadaju: **Open Inventor**, **OpenGL Optimizer**, **Cosmo3D**, **OpenGL++** i **Simple Scene Graph**.

**Open Inventor** je objektno orijentisan API koji nudi rešenja problema koji se javljaju prilikom programiranja interaktivne grafike. Predstavlja programski modul koji se zasniva na bazama podataka 3D scena što dramatično pojednostavljuje programiranje grafike. U sebi sadrži bogat komplet objekata kao što su kocke, poligoni, tekst, materijali, kamere, svetla, 3D prikaze i editor koji povećava brzinu programiranja i povećava programske sposobnosti samog korisnika.

**OpenGL Optimizer** poboljšava vizuelizaciju kompleksnih 3D modela. Na taj način, programeru je omogućeno da pravi aplikaciju koju može vizuelno da kontroliše i gde može interaktivno da utiče na bazu podataka. Ovom API-ju su dodate tehnologije koje se zovu *ClearCoat* i *RealityMapping*. Tehnologija *ClearCoat* je revolucionarna nova tehnologija koja proizvodi prirodan prikaz odličnih materijala kao što su boja, plastika i staklo. Tehnologija *RealityMapping* je revolucionarno nova tehnologija 3D softverskog renderovanja kako bi se sjajne površine prikazale kako treba.

**Cosmo3D** je kratko vreme bio samostalan API, a onda je integrisan u OpenGL Optimizer.

**OpenGL++** predstavlja skup funkcija, koje su pisane u programskom jeziku C++, koji omogućava 3D grafičkom sistemu OpenGL-a da podrži objektno orijentisane strukture podataka. Ovaj projekat je rezultat saradnje firmi SGI, IBM i Intel, koje su rešile da naprave kompleksni API koji će podržavati OpenGL. Rad na ovom API interfejsu prekinut je kada je firma SGI prekinula ovu saradnju i sa firmom Microsoft počela da razvija API pod nazivom Fahrenheit, koji je, takođe, propao.

**Simple Scene Graph (SSG)** je API koji je trebalo da postane sastavni deo OpenGL-a, koji je koristio programski jezik C++ i koji je mogao da radi sa GLUT-om (*OpenGL Utility Toolkit*). SSG je sadržao biblioteku jednostavnih matrica i vektora i podržavao je neka testiranja.

## 1.2.3 Cosmo API

Treba napomenuti da **Cosmo** nije grafički API. Pitanje je da li Cosmo može da se nazove grafičkim API-jem u pravom smislu zato što je Silicon Graphics, u nastojanju

da iskoristi prednosti Java, VRLM 3D okruženja i raznih multimedijalnih elemenata za razvoj prestižnih aplikacija i prezentacija, napravio Cosmo. grafički konglomerat koji se, u stvari, sastoji od vizuelizacije kompleksnih naučnih i poslovnih podataka, poboljšane vizuelne komunikacije, kao i mogućnosti da razvijaju igrice ili interfejse za igrice. Sastavni deo ovog API-ja predstavljaju sledeći proizvodi:

- **Cosmo Create** je sistem koji kombinuje 2D slike, 3D sekvence virtuelne realnosti, tekst i ostale elemente kako bi napravio multimedijalnu stranicu za Internet ili proizvod.
- **Cosmo Player** omogućava prikazivanje ili pretraživanje prezentacije koja je razvijena upotrebom Cosmo API-ja. Treba napomenuti da ovaj proizvod radi na svim operativnim sistemima.
- **Cosmo Code** omogućava vizuelno okruženje za već postojeće Java kodove prilikom stvaranja multimedijalne prezentacije.
- **Cosmo Worlds** omogućava uključivanje sledećih stvari u multimedijalnu prezentaciju: svetlosne efekte, više izvora svetlosti, zvuk, ogromnu biblioteku boja, materijala i tekstura, kao i primene slika čiji je autor sâm programer.
- **Cosmo MediaBase** omogućava programeru lagano upravljanje fajlovima koji učestvuju u prezentaciji.

#### 1.2.4 Fahrenheit API

Prvobitno najavljen početkom 1998. godine, **Fahrenheit** je trebalo da se sastoji od tri komponente. Prva je *Fahrenheit Scene Graph*, koja je po funkciji slična režimu *Retained Mode* iz *Direct3D*. Ona omogućava da se prilikom razvoja programer koncentriše na pravljenje ukupne scene, a ne na realistično prikazivanje pojedinačnih poligona. Gotovo u isto vreme bio je uveden **FLM** (*Fahrenheit Large Model Visualisation*, za vizuelizaciju velikih modela), što predstavlja kompleksniji API ili API vrlo visokog nivoa, projektovan za CAD i profesionalne primene koje podrazumevaju strukture kao što su zakrivljene površine. Najznačajnija komponenta, međutim, biće uvedena malo kasnije. Ta komponenta je **FLL** (*Fahrenheit* nižeg nivoa), što predstavlja jednostavniji API ili API niskog nivoa. Bavi se osnovnim upravljanjem geometrijom ali, što je najvažnije, zameniće *Direct3D Immediate Mode* i radiće zajedno sa OpenGL-om.

#### 1.2.5 Microsoftov API

Grafička tehnologija je oblast industrije PC računara koja se posebno brzo razvija, sa novim skupovima čipova, novim revizijama skupova čipova i čak potpuno novim tehnologijama koje se pojavljuju tempom koji zvonu na uzbunu. Ovo predstavlja problem za primene u oblastima gde treba iskoristiti prednosti najnovijeg 3D hardvera, jer je apsolutno nemoguće za bilo koga ko radi na razvoju aplikacija da piše softver u originalnom kodu za svaki grafički procesor.

Rešenje je u programskom interfejsu aplikacije, API-ju. API radi kao posrednik

između aplikacionog softvera i hardvera na kome se on izvršava. Programer piše kôd koji na izlazu daje podatke API drajveru putem standardizovanih komandi, a ne direktno preko hardvera. Drajver, napisan od strane proizvođača hardvera, onda prevodi taj standardni kôd u originalni format koji poseban, dati model hardvera može da razume.

Prvi put predstavljen 1995. godine, **DirectX** je integrisani skup alata za programiranje, projektovan tako da pomogne programerima da naprave niz multimedijjskih aplikacija za platformu operativnog sistema Windows. On pokriva gotovo sve aspekte multimedije i u vreme DirectX 7.0 obuhvatao je sledeće glavne komponente:

- **Direct3D** za 3D grafiku u realnom vremenu;
- **DirectDraw** za 2D grafiku;
- **DirectSound** za audio reprodukciju;
- **DirectPlay** za mrežno povezivanje (posebno za računarske igre za više učesnika preko interneta);
- **DirectInput** za komandne palice i druge slične uređaje;
- **DirectMusic** za muzičke podatke zanovane na porukama.

DirectX 8.0 je uveden krajem 2000. godine i spojio je DirectSound i DirectMusic u komponentu DirectX Audio, dok su do tada razdvojene funkcionalnosti za 2D i 3D grafiku, DirectDraw i Direct3D spojene u komponentu DirectX Graphics. Tada je, takođe, komponenta DirectShow, ranije implementirana kao poseban API, postala zvanična komponenta DirectX.

Microsoft je uspeo da napravi i DirectX (verzija 9.0) kako bi zadovoljio većinu korisnika koji su tražili poboljšanje performansi svojih računara u igranju akcionih igrica i korisnika koji su želeli da gledaju video prezentacije i animacije kod kuće na svojim personalnim računarima.

Ključ za način na koji **Direct3D** dozvoljava razvijaoocu računarske igre da radi nezavisno od hardvera PC računara je njegov HAL (*Hardware Abstraction Layer*), sloj za apstrakciju hardvera koji kao efekat ima nezavisno pisanje softvera. HAL obezbeđuje spregu sa pogodnostima koje su široko implementirane u 3D grafičkom hardveru i dozvoljava proizvođačima da naprave drajvere koji povezuju HAL sa hardverom. To omogućava Direct3D aplikacijama da iskoriste osobine većine hardverskih uređaja hardvera bez potrebe da se pišu za svaki posebno.

U protočnoj obradi za realističan prikaz Direct3D, geometriju 3D objekata obrađuje centralna procesorska jedinica glavnog računara pre nego što 3D akcelerator počne da prikazuje scenu na ekranu. Tokom 1998. godine - kada su se proizvođači čipova sumanuto utrkivali za pravo da se mogu pohvaliti da imaju najbrži 3D akcelerator - DirectX (verzija 5) je brzo postao usko grlo u procesu 3D realističnog prikaza. Kako je situacija postala još gora sa pojavom 3D čipova treće generacije, Microsoft je pažljivo preispitao API u DirectX za transformaciju i osvetljavanje da bi poboljšao efikasnost geometrijske obrade i uravnotežio arhitekturu sistema. DirectX 6.0 (u stvari, njegovo peto izdanje), bio je uveden u leto 1998. godine i ponudio je nove osobine za poboljšanje realističnog

prikaza kod računarskih igara i modelovanja.

Šesta verzija komponente Direct3D podržava mogućnost novijih grafičkih kartica da izvode renderovanje sa više tekstura u jednom prolazu, što značajno smanjuje vreme potrebno da bi se primenile mape tekstura. Ona, takođe, uključuje novije tehnike za dodavanje realnosti 3D scenama, kao što su anizotropno filtriranje, koje dodaje element dubine trilinearnom filtriranju, i *Bump* preslikavanje, koje stvara iluziju "realnih" tekstura i izvora svetlosti na ravnim površinama. Trend za objedinjavanje osobina OpenGL-a nastavio se sa operacijama postavljanja slika na 3D scenu, a ne tekstura na pojedinačne 3D objekte.

U borbi za prevlast u areni 3D grafike za PC računare, postoji Direct3D u jednom uglu, takmičara koji se oslanja na Microsoftovu veliku industrijsku snagu, i OpenGL u drugom, sa manje gušenja na platformi operativnog sistema Windows, ali sa očigledno većom podrškom ljudi iz razvoja. U fazama uobličavanja DirectX (verzija 7.0) izgledalo je da će se te suprotnosti usmeriti ka razumnom rešenju, sa vestima da su se Microsoft i SGI udružili da naprave nešto što se zove Fahrenheit. Međutim, kada je DirectX (verzija 7.0) došao na tržište 1999. godine, uprkos nastavljanju trenda objedinjavanja osobina OpenGL-a, izgledalo je da su se izgledi za Fahrenheit smanjili.

Pored toga što je bio optimizovan da radi 20% brže od svog prethodnika, DirectX (verzija 7.0) je uključio i izvestan broj novih osobina. Najvažnija među njima je podrška za hardverski ubranu transformaciju i osvetljavanje koje podržava većina 3D grafičkih kartica najnovije generacije, a posebno one zasnovane na skupovima čipova *GeForce 256* firme *nVidia* i *S3* firme *Savage*. S obzirom na to da transformacija i osvetljavanje predstavljaju zadatke u savremenim računarskim igrama koji zahtevaju najviše rada centralne procesorske jedinice, preusmeravanje tog posla na namenski 3D akcelerator oslobađa značajnu količinu procesorovog kapaciteta za druge zadatke - što dozvoljava da se u razvoju ugradi više detalja u realistično prikazivanje i više specijalnih efekata koji traže intenzivno angažovanje procesora.

DirectX (verzija 8), koji se pojavio krajem 2000. godine, uveo je veći broj važnih promena u sledećim oblastima:

- dalje pojednostavljenje inicijalizacije i upotreba Direct3D API-ja, prilagođavanje modela koje nudi Direct3D mogućnostima najnovijeg hardvera i uklanjanje podrške nasleđenim spregama;
- proširenje sprege Direct3D u nove 3D akcelerske tehnologije, kao što su volumetrijske teksture, realistično prikazivanje sa više uzoraka (uključujući i podršku za *T-bafer*) i tako dalje;
- uvođenje novih principa obrade podataka: senčenja, kako na nivou piksela, tako i na nivou geometrijskih podataka.

### 1.2.6 Intelov API

Kada se pomene firma Intel, prvo na šta se pomisli su procesori. Ova firma se bavi i kreiranjem API-ja, a ovde će biti pomenuta dva: 3DRender i Intel Scene Manager.

**3DRender (3DR)** je Intelova 3D grafička biblioteka koja je optimizirana za rad

sa Pentium procesorima. Ta biblioteka ima sledeće osobine: podržava više od šesnaest svetlosnih izvora, omogućava raznorazno mapiranje, omogućava primenu atmosferskih uslova i padavina, omogućava primenu transparentnosti, podržava operativne sisteme Windows 95 i Windows NT, besplatna je.

**Intel Scene Manager (ISM)** je API koji je još uvek u razvojnoj fazi i sličan je OpenGL-u.

### 1.2.7 Sunov API

Internetove tehnologije i programski jezik Java doveli su do fundamentalnih promena u načinu dizajniranja i razvoja aplikacija. Java ima sličan izgled kao jezici C ili C++. Programeri koji imaju iskustvo sa ovim jezicima neće imati problema da nauče Javu. Naredbe kontrole toka i funkcija operatora su skoro identične. Međutim, za razliku od C-a, Java je objektno orijentisan programski jezik koji je redukovao neke funkcije C-a pa je zato relativno mali. **Java3D** je API koji se koristi za pisanje 3D grafičkih aplikacija ili 3D apleta za veb. Ovaj API daje korisniku veliku slobodu pri kreiranju i manipulisanju 3D geometrijom i omogućava primenu mnogih alata za definisanje potrebnih struktura. Java3D API je nastao kao posledica saradnje sledećih firmi: Silicon Graphics, Intel Corporation, Apple Computers i Sun Microsystems.

### 1.2.8 Appleov API

Firma Apple Computers se bavi računarima, ali određeni njeni segmenti se bave i razvojem softvera. Jedan segment je stvorio API pod nazivom **QuickDraw3D (QD3D)**. Ovaj API omogućava kreiranje 3D grafike na *Apple* računarima i renderovanje u realnom vremenu. On podržava skoro sve matične ploče sa svim hardverskim dodacima koje proizvodi pomenuta firma.

### 1.2.9 HP-ov API

Što se tiče firme HP i njenih API-ja koje razvija, vrlo je malo informacija. Najosnovnije informacije postoje o sledeća četiri API-ja:

- **Starbase** je originalni 3D API firme HP i sličan je OpenGL-u, ali je urađen na višem nivou;
- **PHIGS** je API koji je podržavao stariji industrijski standard (ISO) i ovaj API se više ne koristi;
- **PEX** je varijacija gore pomenutog API-ja pod nazivom PHIGS koji je predviđen da radi na serverima;
- **DirectModel** je sličan OpenGL Optimizeru, ali je, u međuvremenu, otkazan i biće integrisan u već pomenuti FLM za vizualizaciju velikih objekata.



### 1.2.10 Ostali API interfejsi

Ovde će biti pomenuti još neki programski interfejsi koji se pominju u literaturi i u raznim člancima, a proizvođači su različiti. O nekima ima više, a o nekima ima manje informacija, ali ovde će biti samo nabrojani oni koji se najčešće pominju, uz osnovne informacije:

- **Heidi** je API firme Autodesk koji omogućava veću brzinu i efikasnost programa da prikaže i odštampa željene informacije. Ovaj API je primarna komponenta programa: AutoCAD, 3DS MAX, 3D Studio VIZ, Volo View, Volo View Express, a sadržan je i u DWF formatu i HDI razvojnom sistemu za kvalitetniju štampu.
- **RenderMan** je API firme Pixar koji omogućava fotorealistično renderovanje i u kome se nalazi moćan programski jezik koji je senčenje doveo do savršenstva.
- **RenderWare**, **BRender** i **GKS** su stariji API-ji koji su omogućavali renderovanje, s tim što više nisu u upotrebi.
- **RealityLab** je API koji omogućava softversko renderovanje. Razvio ga je Microsoft i implementirao u Direct3D.
- **MultiGen's GameGen** je ogroman i krajnje kompleksan API koji se koristi za razvoj igrice, s tim što bi trebalo napomenuti da je i za današnje uslove preskup.