OpenGL را بهتر بشناسیم

گرافیک کامپیوتری همواره از جمله جذابترین جنبه های کامپیوترها بوده است. بازیها، برنامه های طراحی گرافیکی دو بعدی و سه بعدی و شبیه سازیها همگی به نوعی از قابلیتهای گرافیکی یک کامپیوتر بهره میبرند. مهمترین نکته در این زمینه، برقراری تعادلی منطقی بین کیفیت تصاویر نمایش داده شده بر روی صحنه و سرعت اجرای برنامه میباشد. یکی از راه هایی که در دو دهه اخیر همواره برای افزایش سرعت مورد استفاده قرار گرفته است، حذف موانع غیرضروری بوده است. در این راه، سعی شده است تا ارتباط برنامه نویس به ساده ترین صورت با سخت افزار گرافیکی برقرار شود. تقریبا همزمان با عرضه اولین نسخه های ویندوز، محصولاتی نظیر Glide و WinG هم به بازار آمدند که کم و بیش از حمایت مایکروسافت هم برخوردار بودند. ولیکن به جرات می توان گفت شرکت Silicon Graphicsاولین محصول استاندارد و همه منظوره را برای پیاده سازی گرافیک در سیستمهای ویندوز ارائه نمود. محصول این شرکت که OpenGL نام دارد، در طی حدود یک دهه، علی رغم ضعفهای فراوان توانسته است به حیات خود ادامه دهد. (بد نیست بدانید که OpenGL از روی بسته نرم افزاری بنام Iris GL ساخته شد. این کتابخانه در دهه ۸۰ در سیستمهای UNIX کاربرد فراوانی داشت.) شاید این مسئله بیش از هر چیز دیگری به سادگی OpenGL مربوط شود. این سادگی به حدی است که میتوان گفت OpenGL کتابخانه استاندارد صنعتی گرافیک میباشد . امروزه با عرضه نسخه های جدیدDirectX ، شاید حداقل در صنعت ساخت بازی کمتر به OpenGL توجه شود ولی این کتابخانه هنوز هم علاقمندان خاص خود را دارا میباشد و اکثر نرم افزارهای کاربردی طراحی همچون برنامه هایCAD/CAM ، انیمیشن یا ویرایش تصویر بطور پیش فرض از آن بهره میبرند. در این مقاله سعی خواهیم کرد با هم نگاهی هر چند گذرا به ویژگیها و نقاط ضعف این مجموعه بیاندازیم.

اگر بخواهیم از دید مهندسی نرم افزار به OpenGL نگاه کنیم، شاید سه خصوصیت بارز برای آن بیابیم OpenGL .به شیوه ای طراحی شده است که دستورات در یک محل صادر شده و در محل دیگر اجرا شوند، به همین سادگی. به این معماری، ساختار مشتری/خدمات دهنده (client/server) گفته میشود. بدین شیوه OpenGL میتواند حتی بر روی شبکه های کامپیوتری هم بخوبی کار کند . خصوصیت دیگر استفاده از یک محل مجتمع برای نمایش تصاویر میباشد. این محل که میتواند در هر جایی از حافظه قرار بگیرد (از جمله حافظه گرافیکی) بافر فریم (frame buffer) نامیده میشود.

مورد آخر اینکه OpenGL معمولا خیلی در مورد محدوده پارامترها وسواس به خرج میدهد و اجازه نمیدهد مقادیر خارج از حوزه ارسال شوند. این عمل که parameter validation نام دارد گاهی اوقات باعث پایین آمدن سرعت رندر میگردد.

شاید خصوصیت اول یکی از مهمترین دلایلی باشد که خیلی ها OpenGL را کتابخانه ای خوب (یا بهتر بگوییم آسان) توصیف میکنند. وقتی این حالت را با DirectX مقایسه میکنیم، میبینیم که حضور تکنولوژیCOM ، اغلب باعث پیچیدگی کار برنامه نویسی با آن میشود. پس شاید بتوان گفت OpenGLاز این لحاظ برتری دارد؛ زیرا بصورت یک سری فایلهای سرباره و کتابخانه ای ارائه میشود، حالتی که برنامه نویسها سالهاست با آن آشنا هستند.

از جمله ویژگیهای مهمOpenGL ، که باعث جذب طرفداران زیادی برای آن شده است، قابلیت اجرای آن در انواع سیستم عاملها میباشد. یعنی این کتابخانه، برخلاف بسیاری از رقبای خودش تنها به سیستم ویندوز محدود نمیباشد و به گفته خیلی از کاربرانش، هر جا که یک کامپیوتر وجود داشته باشد میتوان از OpenGL هم استفاده کرد.

درست است که OpenGL ابتدا توسط شرکت SGI ارائه شد ولی در حال حاضر تقریبا متولی مشخصی ندارد و هر کسی میتواند یک نسخه اختصاصی از OpenGL را نوشته و بر روی اینترنت قرار دهد تا دیگران هم از آن بهره ببرند؛ کاری که الان شرکتهای بزرگی همچون مایکروسافت و اپل انجام داده اند و نسخه اختصاصی از این کتابخانه را برای سیستم عاملهای خود عرضه کرده اند .در واقع OpenGL، وصله های زیادی دارد (این وصله ها extension نامیده میشوند). الان مدتهاست که گفته میشود قرار است نسخه ۲,۰ OpenGL به بازار بیاید، ولی هنوز چنین چیزی در دنیای واقعی وجود ندارد.

در ضمن OpenGL یک نوع ماشین حالت (state machine)به شمار می آید. یعنی اکثر خصوصیات و قابلیتهای آن بصورت یک تابع حالت عرضه میشوند. بدین ترتیب که اگر مثلا خصوصیت نورپردازی آن را فعال نمایید، تا زمانی که بار دیگر این خصوصیت را غیر فعال نکرده باشید صحنه شما تحت نورپردازی قرار میگیرد.

اگر شما از آن دسته افرادی هستید که از اینترنت برای یافتن پاسخ سوالهای برنامه نویسی خود استفاده میکنید، با خیال راحت به سراغ OpenGL بروید. چون منابع زیادی در این زمینه وجود دارد و میتوانید مطمئن باشید که هیچ یک از سوالهای شما بی پاسخ نخواهد ماند OpenGL .بر خلاف اکثر کتابخانه های مشابه دارای مستندات کاملا استانداردی میباشد. شرکت SGI سالهاست کتابی تحت عنوان OpenGL Red Book را عرضه کرده است که به توضیح و تشریح OpenGL اختصاص

داده شده است. همچنین کتاب OpenGL Blue Book هم بعنوان مرجع دستورات این کتابخانه میباشد. نکته دیگر اینکه OpenGL یک کتابخانه شیء گرا نمیباشد و همانطور که گفتیم بعنوان یک سیستم مشتری/خدمات دهنده همه امور خود را با یک سری تابع انجام میدهد. این مسئله میتواند در مورد برنامه های بزرگ مشکل آفرین باشد. (البته خود OpenGL بصورت مرحله ای طراحی شده است، ولی میتوان به گونه ای برنامه را طراحی کرد که از تکنولوژی شیء گرا استفاده کند؛ هر چند این کار خیلی راحت نمیباشد)

از جمله امکانات خوبی که در این کتابخانه تعبیه شده است، توابع خود فراخوان (callback) میباشد. این توابع میتوانند همانند یک متغیر بعنوان آرگومان به یک تابع دیگر ارسال شده و در موقع مقتضی فراخوانی شوند. از این توابع بیشتر برای اعلام خطاهای داخلی دستورات، شکستن سطوح منحنی و اعمال مورد نیاز در حین تغییرات پنجره برنامه استفاده میشود.
حالا اجازه دهید کمی هم راجع به معماری OpenGL بحث کنیم و ببینیم OpenGL چطور یک صحنه را رسم میکند. در پایین ترین سطح گرافیک سه بعدی، معمولا سه شیء اولیه نقطه، خط و چند ضلعی وجود دارند OpenGL .قادر به رسم تک تک این اشیاء میباشد. هر چند در حوزه گرافیک کامپیوتری، عملا چیزی بنام چندضلعی وجود ندارد و اشکالی که بیش از سه ضلع داشته باشند به مثلث تبدیل شده و سپس رندر میگردند. (فرآیند شکستن سطوح tessellation نامیده میشود)
OpenGLبطور پیش فرض همزمان با صادر شدن هر دستوری، عملیات مربوطه را انجام میدهد؛ یعنی دستورات و داده ها در جایی ذخیره نمیشوند تا بعدا همگی با هم اجرا گردند. این مسئله خود از دلائل سادگی کار با OpenGL میباشد. ولی هنگامی که نیاز به رسم صحنه ای پیچیده داشته باشید، سرعت اجرای برنامه خیلی اهمیت می یابد، و سادگی کار نمیتواند تنها عامل موثر باشد. در نتیجه نیاز به بافری خواهید داشت که یک سری دستورات و داده ها را در خود ذخیره کرده و سپس در زمان مورد نیاز همگی آنها را به یکباره اجرا کند. چنین مکانیسمی در خیلی از کتابخانه های مشابه هم وجود دارد. مثلا در DirectX به این مکان ذخیره گر، بافر گره یا بافر اندیس گفته میشود. در OpenGL هم مکانی از حافظه بنام لیست نمایشی به این عمل اختصاص داده شده است. از این لیستها میتوان برای انجام اموری مانند نمایش فونتها، بافتها، انیمیشن، ذخیره اطلاعات گره ها و ... استفاده کرد. همچنین امکان ذخیره اکثر متغیرهای حالت هم در این لیستها وجود دارد. از این نظر میتوان آنها را با بلاکهای حالت (state block) در DirectX مقایسه کرد.
OpenGLمیتواند سه حالت عملیاتی داشته باشد. حالت معمولی که همان حالت رندر نام دارد و دو حالت دیگر حالتهای انتخاب (selection) و بازخورد (feedback) میباشند که میتوانند جهت انتخاب اشیاء در صفحه توسط کاربر (با وسایلی مثل ماوس) و یا ارسال اطلاعات اشیاء رندر شده به جایی غیر از صفحه (مثلا به یک فایل یا چاپگر) استفاده شوند.

دستورات اصلی OpenGL با پیشوند gl آغاز میشوند، مثلا دستور glEnable() مسئول فعال کردن متغیرهای حالت میباشد glu .هم پیشوندی است که قبل از نام دستورات کتابخانه کمکی OpenGL ظاهر میشود. در OpenGL بسیاری از دستورات دارای چندین قالب میباشند و میتوان آرگومانهای آنها را هم به شکل مقداری و هم به شکل آرایه ای) برداری) به تابع ارسال کرد، که البته نوع آرایه ای دستورات علی رغم پیچیده تر بودن اغلب سریعتر میباشد. برای اینکه بتوانید اطلاعاتی هم نسبت به قابلیتهای سیستمی که برنامه روی آن اجرا میشود بدست آورید از توابعی مثلglGetInteger ، glGetFloatیا glGetBoolean استفاده کنید. همچنین از آنجاییکه تفاوتهایی بین نسخه های مختلف OpenGL وجود دارد، میتوانید به کمک همین توابع از قابلیتهای دقیق نسخه خود آگاه شوید. توابع کتابخانه کمکی هم در اصل از روی خود توابع اصلی ساخته شده اند .تنظیم دوربین و محوطه دید، رسم اشکال سه بعدی همچون کره و استوانه، رسم سطوح منحنی، آماده سازی تصاویر و کمک به اشکالزدایی برنامه از جمله قابلیتهای این کتابخانه میباشند.

یکی از مزایای OpenGL در مقایسه با رقبایش، قابلیت رندر کردن اشیاء منحنی و سطوح درجه دو و چند جمله ای میباشد. ارزیاب (evaluator) که یک مفهوم ریاضی میباشد در OpenGL هم جهت رندر کردن منحنی های درجه دوم بکار میرود. همچنین در اینجا میتوانید سطوح و منحنیهای NURBSرا توسط کتابخانه کمکی رسم کنید.

در OpenGL میتوانید دو حالت رنگی را تنظیم کنید، یکی حالت رنگی حقیقی یا RGBA که در آن رنگها بصورت مولفه های قرمز، سبز، آبی و آلفا مشخص میشوند و حالت اندیس رنگی که در آن کلیه رنگهای موجود درون یک پالت (جدول رنگی) قرار میگیرند و برای استفاده از هر یک باید اندیس معادل آن رنگ را فراخوانی کرد. این روزها استفاده از حالت پالتی کمتر رایج میباشد ولی هنوز هم جهت حفظ سازگاری، این قابلیت در OpenGL قرار داده شده است .در واقع زمانی که OpenGL طراحی میشد وضعیت تقریبا برعکس بود چون کارتهای گرافیکی قابلیت نمایش تعداد کمی رنگ را داشتند، همچنین قابلیتهایی مثل) dithering مات سازی (و) halftoning سایه سازی) که در دهه ۹۰ برای ایجاد سایه های رنگی (و در نتیجه افزایش تعداد رنگهای موجود) بکار گرفته میشدند در این کتابخانه گنجانده شده اند.

اگر بخواهیم OpenGL را از نقطه نظر جلوه های گرافیکی بررسی کنیم، درخواهیم یافت که این کتابخانه توانسته است متناسب با زمان، اکثر قابلیتهای پیشرفته گرافیکی را پشتیبانی کند. اعمالی همچون ترکیب رنگ(blending) ، تست و ترکیب آلفا، مه و پشته ماتریس به شکل خوبی در OpenGLپیاده سازی شده اند. همچنین در حال حاضر سایه زنها (shader) هم در این کتابخانه قابل استفاده میباشند و زبانهایی همچون Cg و GLSL برای کاربرد سایه زنهای گره و پیکسل در

OpenGLعرضه شده اند تا دیگر نیازی به کدهای اسمبلی برای برنامه نویسی ریزپردازنده های گرافیکی نباشد. یکی از اصطلاحات خاصی که بیشتر در برنامه نویسی OpenGL رایج میباشد واژه تکه (fragment) میباشد که به گروهی از پیکسلها اطلاق میشود. گاهی اوقات به سایه زنهای پیکسل در دنیایOpenGL ، سایه زن تکه گفته میشود.

اموری مثل سایه زنی اشیاء(shading) ، نورپردازی و بافت زنی هم با امکانات خوبی در OpenGL پیاده سازی شده اند و در این زمینه برنامه نویس گرافیکی مشکل زیادی را حس نمیکند. پیشتر گفتیم که دستوری با نام glEnable برای فعال سازی متغیرهای حالت بکار میرود؛ جهت غیر فعال نمودن متغیرهای حالت هم میتوان از دستور glDisable استفاده نمود. فرآیند رندر کردن هم در OpenGL خیلی پیچیده نمیباشد. برای مثال به کد نمونه زیر توجه کنید:

```
glBegin(GL_TRIANGLE);
glNormal۳f(۰،۱،۰);
glVertex۳f(۱۰۰،۰،۰);
glNormal۳f(۰،۱،۰);
glVertex۳f(-۱۰۰،۰،۰);
glNormal۳f(۰،۱،۰);
glVertex۳f(۰،۰،۱۰۰);
glEnd();
```

آرگومان دستور glBegin مشخص میکند که قرار است یک مثلث رسم شود، دستور glNormal۳f هم مختصات بردار نرمال گره ای را میگیرد که مختصات آن در اولین دستور glVertex۳f بعدی خواهد آمد. بدین ترتیب یک مثلث با اضلاع ۱۰۰ ساخته میشود. در نهایت glEnd تعیین کننده پایان عملیات رندر میباشد. توجه داشته باشید که در اینجا بلافاصله پس از صادر شدن دستور glVertex گره مورد نظر بر روی تصویر ظاهر میشود .میتوان رنگ گره ها را هم توسط دستور glColor تعیین کرد.

در OpenGL همچنین امکانات خوبی برای خواندن، نوشتن، کپی و رسم مستقیم پیکسلها وجود دارد. اصولا OpenGL کتابخانه کوچکی میباشد و تعداد دستوراتش معمولا از ۱۲۰ تجاوز نمیکند)دستوراتی که خیلی اوقات کارهای مشابه را با دریافت آرگومانهای متفاوت انجام میدهند.(از جمله راه های کاربردOpenGL ، استفاده از آن به موازات سایرAPI ها همچون DirectInput و DirectSoundمیباشد زیرا این کتابخانه بطور ذاتی هیچ نوع پشتیبانی از ادوات صوتی و ورودی ارائه نمیکند. از جمله معایبOpenGL ، عدم پشتیبانی کامل سازندگان سخت افزار میباشد. یعنی هیچ

ضمانتی وجود ندارد که یک ویژگی پیشرفته گرافیکی بر روی برنامه های **OpenGL** قابل پیاده سازی باشد، اما **OpenGL** خود این قابلیت را دارد که کلیه اعمال سخت افزاری را توسط **CPU** شبیه سازی نرم افزاری کند (البته با سرعتی پایینتر.)

یادآور میشوم که کتابخانه های **OpenAL** و **OpenIL** هم چند سالی است که به موازات **OpenGL** به ترتیب برای انجام امور صوتی و ورودی برنامه ها به بازار عرضه شده اند.

امیدوارم شما عزیزان از این مقاله نهایت بهره را برده باشید، دنیای گرافیک کامپیوتری، بسیار جذاب و پرهیجان میباشد، ان شاء ا... این مقاله نقطه شروع شما برای ورود به این صنعت باشد.

منبع :**galaxyroad.com**

انتخاب مفاله : ملیکا امامی