

مقدمه

قبل از هر چیز هدف خود را از دید یک برنامه نویس بررسی می کنیم. برای انتقال موضوع کافی است بدانیم که برای ساخت بعد سوم از دنیایی که برای ما تشریح شده می بایست مفاهیمی ابتدایی را در این مورد بدانیم تا در نتیجه بتوان یک موتور سه بعدی طراحی و پیاده سازی کرد. در واقع هدف اصلی ما مستقل از بستر پیاده سازی کسب توانایی اداری از مدل سازی دنیایی سه بعدی است که قابلیت نمایش در صفحه مانیتور را داشته باشد. با این تفصیر به مفهومی ابتدایی از دنیای طراحی خواهیم رسید که به تصویر مجسم معروف است پس از بررسی هندسی برخی مفاهیم به بررسی برخی انواع تصویر مثل پرسپکتیو، ایزومتریک، دیمتریک خواهیم پرداخت در ادامه مباحث تغییرات حاصله از حرکت به بعد سوم را خواهیم دید و در نهایت به معرفی فضاها و مشخصات ریاضی آنها از دید گرافیک کامپیوتر و نحوه تغییر موقعیت ها در آن خواهیم رسید.

تصویر مجسم و اهمیت آن

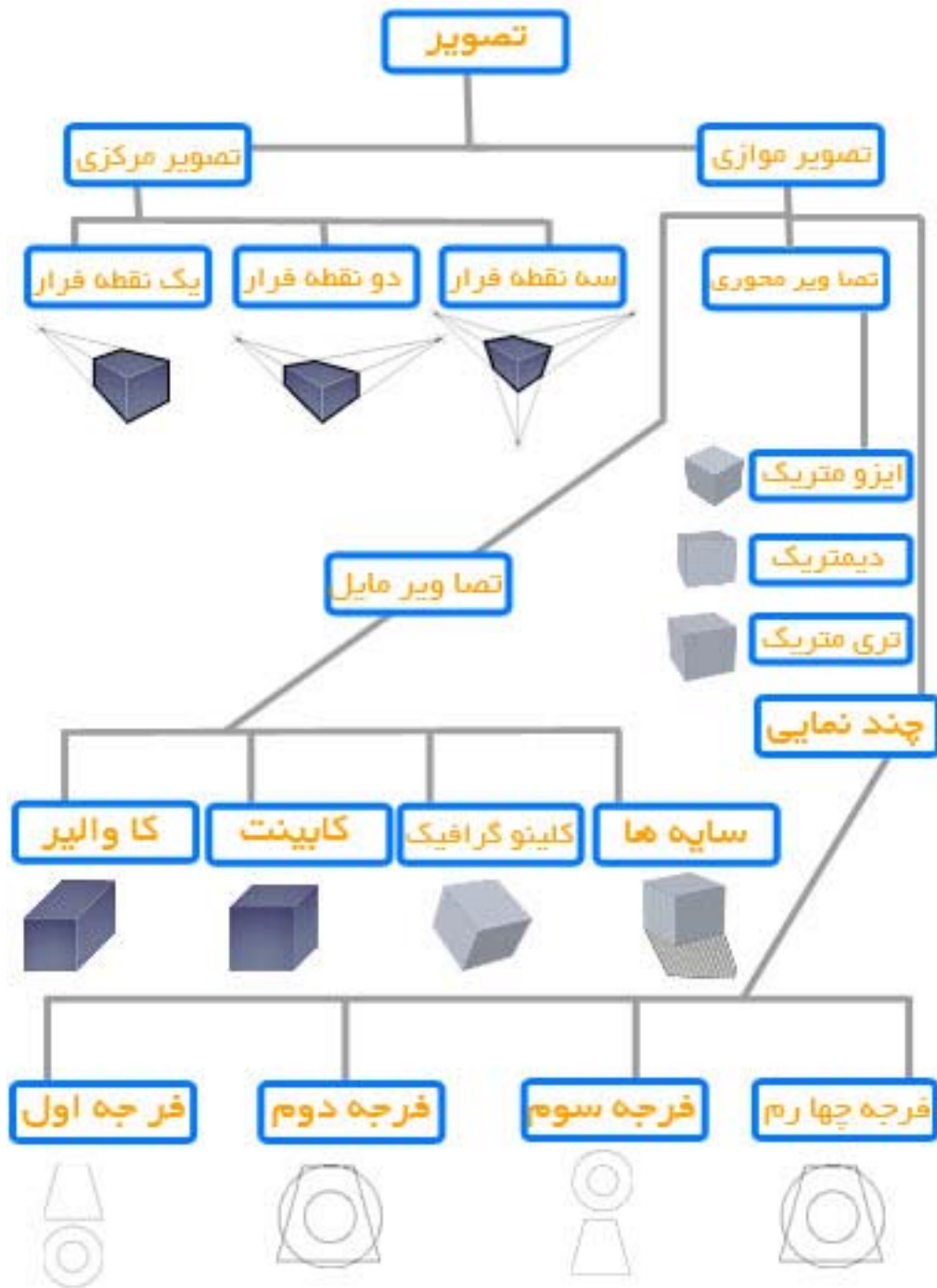
برای ساخت یک ایده تازه – غالباً صنعتی – یک طراح باید بتواند منظور خود را به نحوی مناسب و گویا بیان کند. برای این منظور روش های مختلفی خصوصاً از سوی دانشمندان نقشه کشی ابداع شده است که تصویر مجسم پایه ای ترین آنهاست. تصویر مجسم ساده ترین راه برای نمایش یک تصویر سه بعدی است خصوصاً زمانی که این تصویر دارای پیچیدگی های فراوان از قبیل خم های مختلف، جزئیات ظریف، سوراخ ها و برخورد های گوناگون و از این قبیل باشد. در واقع تصویر مجسم همان تصویری است که از یک جسم در زاویه ای خاص در ذهن طراح نقش خواهد بست.

اصول و پیدایش تصویر

شاید تا کنون به این نتیجه رسیده باشید که یک جسم در حالت های مختلف قرارگیری شما نسبت به آن و یا تابش حالات و جنس های مختلف نور، مثل نور آفتاب یا یک لامپ آویز، تصاویر مختلفی را در ذهن تداعی و زاویه جدید از خود را به ما نمایش می دهد. به طور کلی از یک جسم به دو صورت می توان تصویر تهیه کرد که عبارتند از :

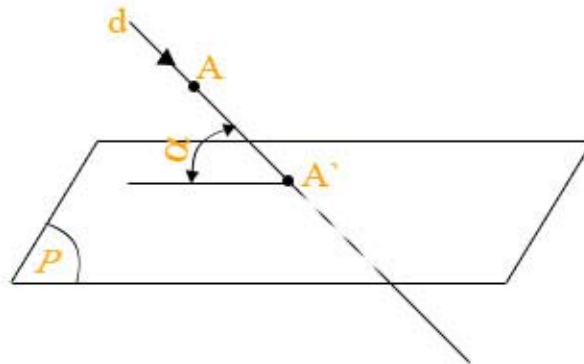
- تصویر موازی
- تصویر مرکزی

یک تقسیم بندی کاملتر از تصاویر در شکل زیر آمده است :

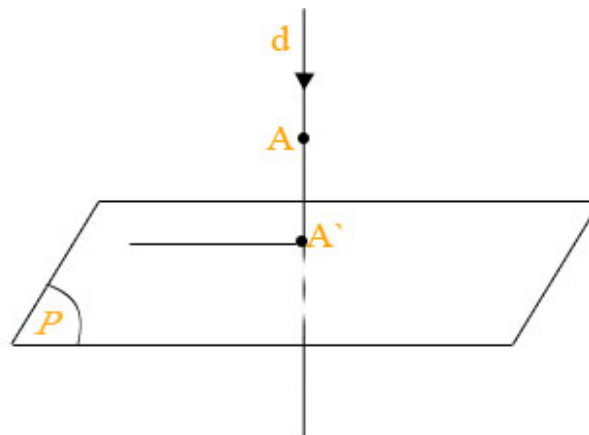


قبل از ادامه مبحث تصویر لازم می آید به این نکته اشاره شود که برای ترسیم یک نقطه، خط، سطح و یا حجم می بایست از اصول **هندسه ترسیمی** استفاده کرد. در واقع به کمک هندسه ترسیمی ما یک جسم سه بعدی را با تصاویری دو بعدی خلق خواهیم کرد این دقیقا چیزی است که یک برنامه نویس موتور سه بعدی نیاز دارد یعنی نمایش اجسام سه بعدی در صفحه نمایش دو بعدی. برای این خلق تصویر مطمئنا از سطحی بنام **صفحه**

تصویر استفاده می کنیم. به طور کلی هر گاه خط d تحت زاویه α از نقطه A گذشته و صفحه P را در نقطه A' قطع نماید، خط d را شعاع مصور، زاویه α را زاویه تصویر، نقطه A' را تصویر نقطه A و صفحه p را صفحه تصویر می نامیم :



صفحه تصویر ممکن است قائم، افقی و یا حالات دیگری داشته باشد:



چند مفهوم تعریف نشده در هندسه ترسیمی

در هندسه نقطه، خط و صفحه مفاهیم تعریف نشده اند و می بایست به صورت ذهنی آنها را پذیرفت.

نقطه

به صورت یک مفهوم ذهنی شناخته می شود. نقطه را به عنوان یک اصطلاح نخستین و یا یک مفهوم

تعریف نشده می پذیریم. چون نمی توانیم با درک، تصور یا مشاهده آنرا بشناسیم. ناگزیر بدون آنکه تعریفی

داشته باشد به عنوان کوچکترین جزء مادی که هیچ بعدی ندارد پذیرفته می شود. مثلا اگر نوک خودکار یا گچ را به سطح مناسب مثل کاغذ یا تخته بفشارید اثر به جا مانده یک نقطه است. البته نقطه را به صورت تقاطع دو خط نیز بیان کرده اند ولی قایل قبول نیست. و یا کوچکترین عنصری که طول عرض و ارتفاع ندارد. یا تعاریف گرافیکی مثل ستاره در آسمان و ... عبارات دیگری هستند که موصوف نقطه می باشند.

خط

مجموعه ای از نقاط که در یک راستای معین قرار گرفته باشند. اثر نوک قلمی بر کاغذ که بیشتر از یک نقطه باشد. برای معرفی خط کافی است حداقل دو نقطه از آن معلوم باشد. در هندسه وقتی به طور مطلق از کلمه **خط** استفاده می کنند منظور خط راست است. اگر یک کاغذ را تا کنید اثر تا خوردن کاغذ یک خط راست است. در حالت کلی یک خط را با دو نقطه متمایز می توان نمایش داد.

صفحه

مجموعه‌های از نقاط یا خطوط که به صورت مسطح واقع شده اند و یا سطحی که فضا را به دو قسمت تقسیم می کند. سطح کاغذ و یا تابلوی کلاس نمونه هایی از صفحه اند. در هنرهای تجسمی می توان سطح دریا را یک صفحه محسوب کرد.

- از برخورد دو خط یک نقطه و از برخورد دو صفحه یک خط بوجود می آید.

حجم

قسمتی از فضا که بوسیله تعدادی صفحه محدود شده باشد حجم نامیده می شود. به عبارت دیگر لازمه وجود هر حجم داشتن سه بعد طول، عرض و ارتفاع می باشد. احجام ممکن است هندسی یا غیر هندسی باشد. احجام هندسی مثل منشور، استوانه و ... و احجام غیر هندسی مانند پدیده های طبیعی کوه، درخت و ... احجام ممکن است محصور بین حجم های دیگر باشد مانند فضاهای خالی یک دره یا گودال در طبیعت یا انواع سوراخ ها در قطعات صنعتی. در کتب مختلف کلمه جسم معادل حجم در نظر گرفته شده است.

با داشتن فهمی از موارد ذکر شده می توان به ادامه بحث یعنی انواع تصاویر پرداخت. اگر به یاد داشته باشید پیش تر گفته شد که تصاویر به دودسته کلی موازی و مرکزی تقسیم بندی می شوند حال جزئی تر به آنها می نگریم.

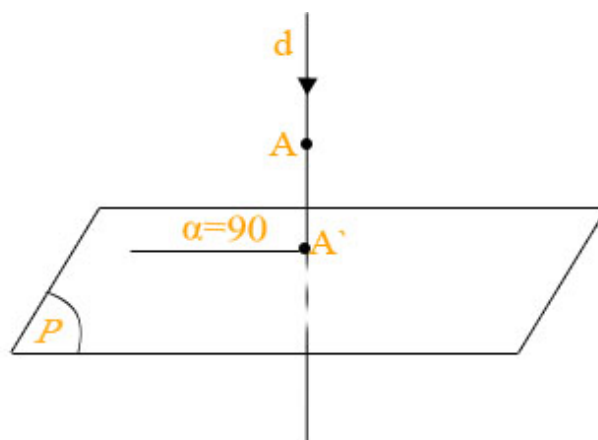
تصویر موازی

گروه بزرگی از تصاویر صنعتی را شامل می شود که خود به دو دسته تقسیم می شود:

- قائم یا عمودی
- مایل

تصویر قائم یا عمودی

دقیقا معادل با توضیحی است که برای صفحه تصویر گفته شد یعنی بنا به تعریف هر گاه خط d تحت زاویه α از نقطه A گذشته و صفحه P را در نقطه A' قطع نماید، خط d را شعاع مصور، زاویه α را زاویه تصویر، نقطه A' را تصویر نقطه A و صفحه p را صفحه تصویر می نامیم در صورتی که زاویه α برابر با 90 درجه باشد تصویر بوجود آمده تصویر قائم یا عمودی است ویژگی این تصاویر این است که شعاع های مصور آن با هم متوازیند :

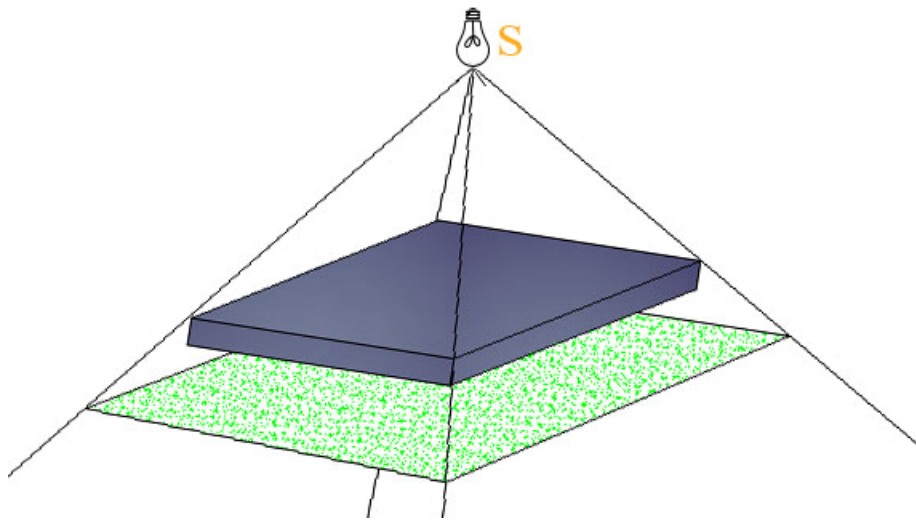


تصویر مایل

اگر مصور d با صفحه تصویر P زاویه ای مانند α که غیر از 90 درجه باشد، بسازد، تصویر مایل نام دارد. سایه های اجسام هنگام طلوع یا غروب خورشید مثال خوبی از این نوع تصویر هستند ضمنا در تصویر مایل نیز شعاع های تصویر باید با هم موازی باشند.

تصویر مرکزی

چنانچه شعاع های تصویر مجبور باشند که همه از یک نقطه گذر کنند، تصویر را مرکزی گویند.



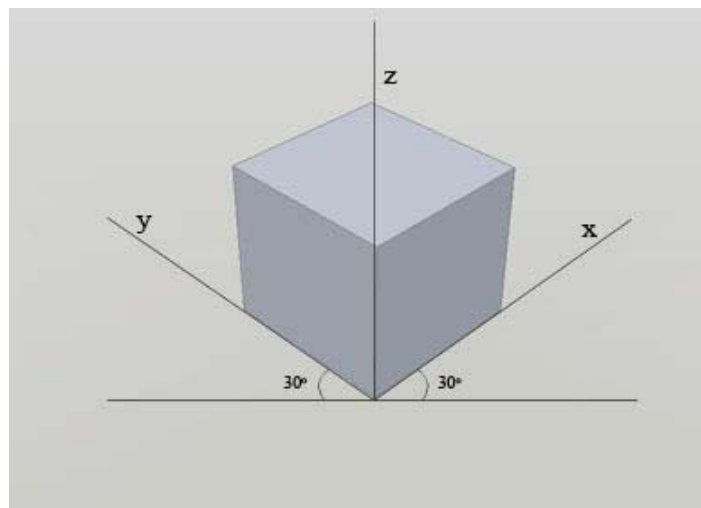
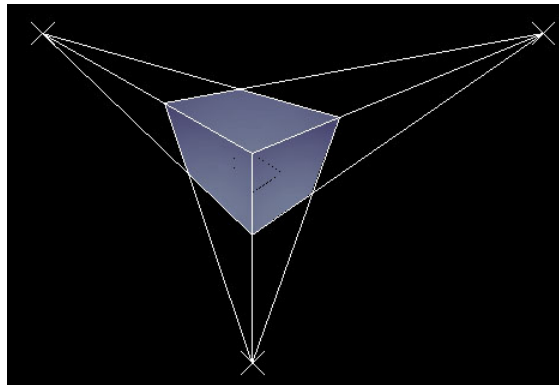
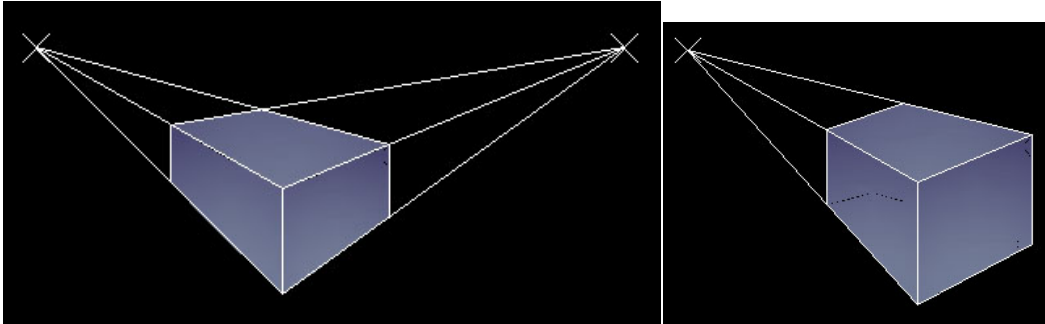
ذکر خصوصیات و تعاریف کلیه انواع تصویر علاوه بر خارج بودن از حوزه بحث نقش کاربردی مهمی را در سطح پیاده سازی موتورهای سه بعدی که هدف نهایی یک توسعه دهنده است بازی نمی کند هرچند که شایسته است برای اجرای این گونه پروژه ها از متخصصین امر بهره لازم را برد. لذا در این مقاله بیشتر به جنبه های تکنیکی مورد نیاز برای یک برنامه نویس سه بعدی می پردازیم. بنابراین لازم می آید تا مفهومی را که برنامه نویس با آن مستقیماً سروکار دارد بیشتر بررسی نماییم. در بخش بندی تصاویر به معرفی اجمالی تصویر مرکزی پرداختیم حال به معرفی پرسپکتیو به عنوان مهمترین نوع تصویر در دنیای سه بعدی کامپیوتر خواهیم پرداخت.

پرسپکتیو Perspective

تعریف

تصویر مرکزی را در اصطلاح پرسپکتیو گویند، که گاهی هم پرسپکتیو خطی نامیده شده است. پرسپکتیو در لغت به معنی تصویری است به همان صورت که چشم انسان می بیند. اهمیت این نوع تصویر به دلیل مطابقت آن با دید انسان است و به همین دلیل نسبت به سایر تصاویر مجسم بهتر موضوع را نمایش می دهد. از این رو معماران، آرشیکت ها، نقاشان و برنامه نویسان بازی به وفور از آن استفاده می کنند علاوه بر این مزیت دیگر پرسپکتیو این است که با استفاده از آن اجسام را با هر طولی می توان ترسیم کرد. برای مثال در یک بازی برای فضا سازی جاده ها، ساختمان ها، پل ها، فضاهای داخلی و... ایده آل است. این در حالی است که تصاویر موازی برای نمایش اجسام و قطعات با ابعاد متناسب (طول و عرض و ارتفاع نزدیک به هم) مناسب است.

برای مثال جهت نمایش یک اسلحه قرار گرفته شده روی یک میز در یک بازی از نوع خاصی از تصاویر موازی به نام تصویر مجسم مایل ایزومتریک (Isometric) یا کوالیر (Cavalier) که در آن هر سه محور آن دارای یک مقیاس است استفاده می شود. دو شکل زیر به ترتیب تعدادی پرسپکتیو و محورهای ایزومتریک را نمایش می دهد :



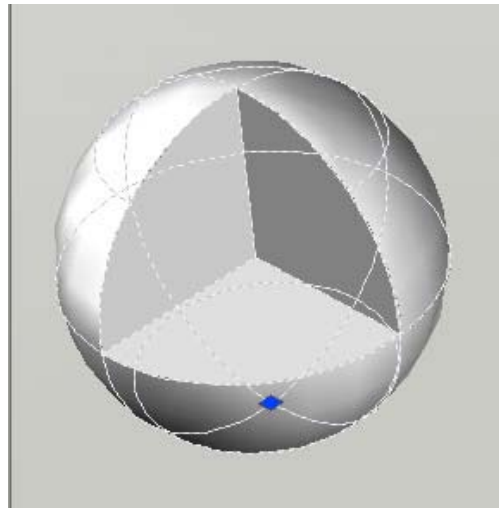
البته گونه ای دیگر از تصاویر در طراحی اجسام رایج است که به تصاویر دیمتریک یا Cabinet معروف است و در شبیه سازی سه بعدی صنعت کاربرد خاص دارد. کلیه خصوصیات آن شبیه ایزومتریک است اما مقیاس محور X ها را یک دوم بدست می آوریم که این باعث می شود تا تجسم جسم برای افراد عادی راحت تر شود.

تصویر مجسم اجسام خاص

برای آشنایی بیشتر با تصویر مجسم در طراحی مثال هایی کلاسیک در زیر آورده شده است.

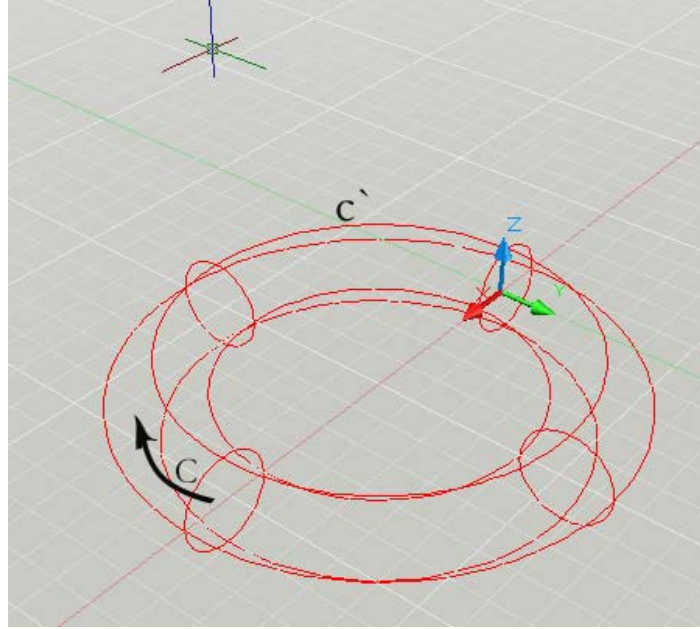
تصویر مجسم کره

کره حجمی است که از دوران یک یا نیم دایره، حول یکی از اقطارش حاصل می شود. بزرگترین دایره موجود در یک کره را که دایره ای به قطر خود کره است دایره عظیمه می نامند. یک کره دارای بینهایت دایره عظیمه و بینهایت قطر است. لذا صفحه تصویر P را به هر صورتی که در مقابل کره قرار دهیم، تصویر کره یک دایره به قطر دایره عظیمه خواهد بود. تصویر مجسم نیز از این قاعده مستثنی نیست. پس تصویر مجسم یک کره دایره ای به قطر دایره عظیمه است. اغلب اضافه یا کم کردن قسمت هایی، حالت کروی را بهتر می رساند در شکل زیر یک هشتم کره برداشته شده است :



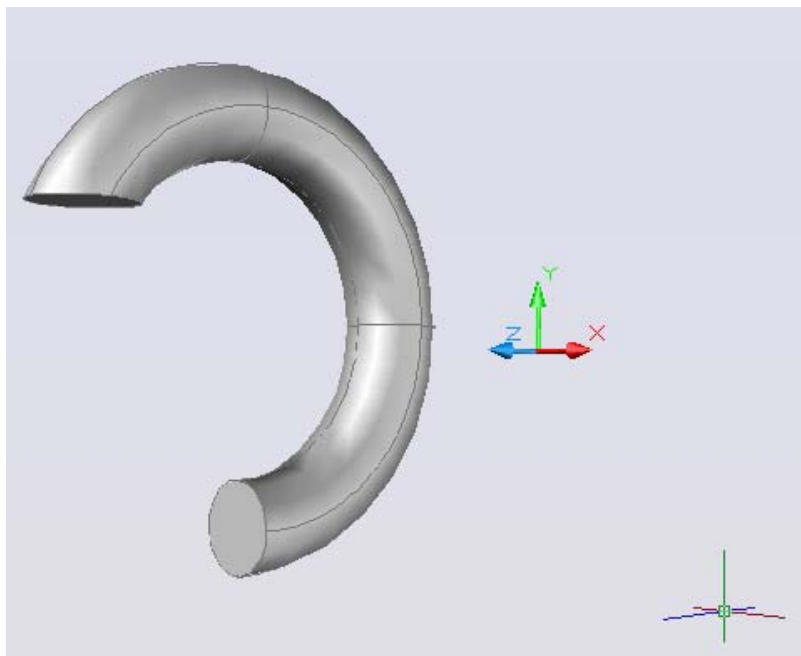
حلقه

اگر دایره C چنان چنان در فضا تغییر مکان دهد که همواره مرکز آن متکی به دایره C و سطح آن عمود بر سطح دایره C باشد، حجمی حادث می شود که حلقه نام دارد. به عبارت دقیقتر، بایتنس خط عمود بر مرکز دایره C، همواره مماس بر دایره C باقی بماند. ضمناً اگر کره در نظر بگیریم دیگر چنین شرطی لازم نیست.



زانو

اگر قسمتی از حلقه را در نظر بگیریم به آن زانو گفته می‌ود.

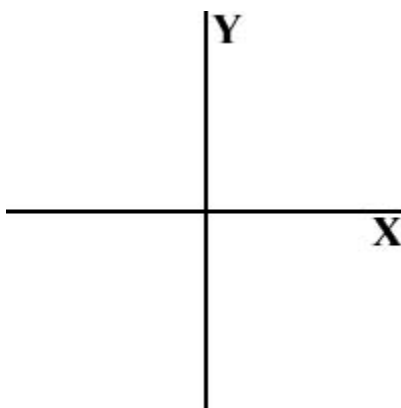


اصول ترسیم تصویر مجسم موازی

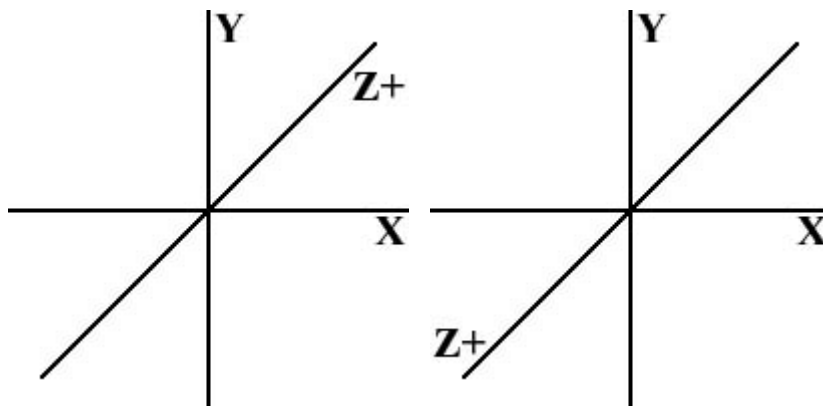
برای نمایش جسمی سه بعدی را روی سطحی دو بعدی ناچاریم که ابتدا جسم را در یک فضای سه بعدی شامل سه محور فرض و سپس جسم را تصویر کنیم. برای این منظور می بایست سیستم های دو بعدی و سه بعدی را تعدیل کنیم یا به عبارتی محور ها را **کوئوردیناتیس (Coordinate)** کنیم.

سیستم های هماهنگ

همانطور که می تواید حدس بزنید می خواهیم یک سیستم هماهنگ را برای مدل سازی هر نوع شیء سه بعدی بکار گیریم. اگر اندکی از جبر بدانید این موضوع اصلا پیچیده نخواهد بود. به شکل زیر نگاه کنید ، این یک استاندارد برای سیستم هماهنگ کارتزین است که می توان نقطه ای را در جایی از فضای دو بعدی تعریف کرد.



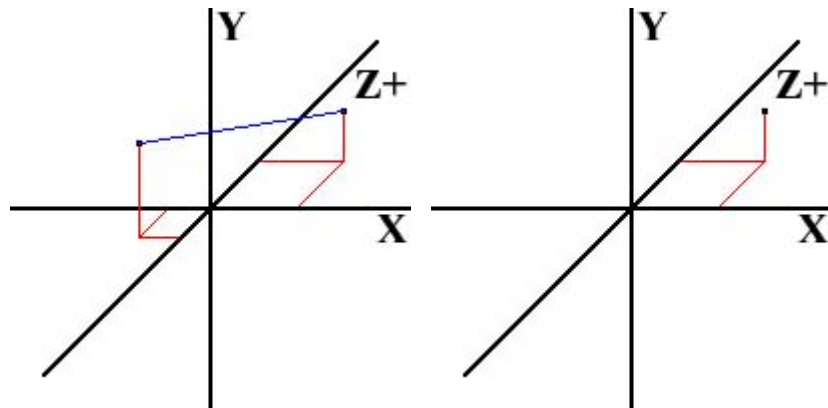
حال تمام کاری که ما خواهیم کرد افزودن محور Z به این سیستم است که به ما امکان تعریف نقاط در فضای سه بعدی را می دهد. این به نظر کافی میرسد اما یک مشکل کوچک دارد، قسمت مثبت محور Z را می توان هم به داخل و هم به بیرون اشاره داد. وقتی محور Z را به داخل اشاره دهیم سیستم **چپ دست** و اگر آنرا به بیرون اشاره دهیم سیستم **راست دست** نامیده می شود.



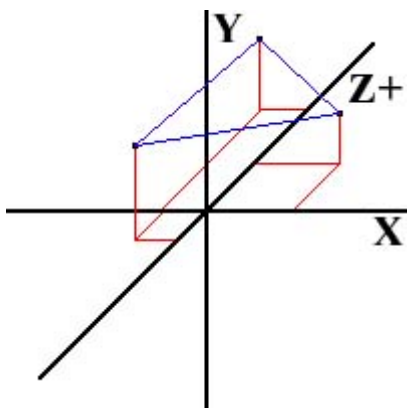
این که برای کار کدام را در نظر بگیرید مهم نیست اما به خاطر داشته باشید که این انتخاب در محاسبات ریاضی شما تاثیر خواهد داشت. اغلب برنامه نویسان بازی سیستم چپ دست را ترجیح می دهند زیرا اینکه محور Z به درون مانیتور اشاره کند قابلیت فهم بیشتری دارد. در هندسه به این سه محور **کنج سه قائمه** گویند در ادامه اشکال هندسی ابتدایی را روی این سیستم بررسی می نماییم.

اشکال هندسی پایه

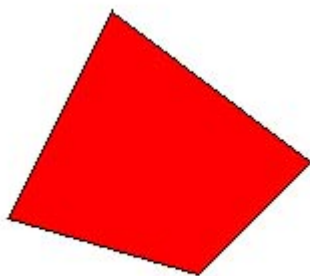
حال که ما یک سیستم هماهنگ داریم اجازه بدهید که با آن کمی بیشتر کار کنیم. در ابتدا ببینیم که در فضای سه بعدی نقطه و خط چگونه تعریف می شوند :



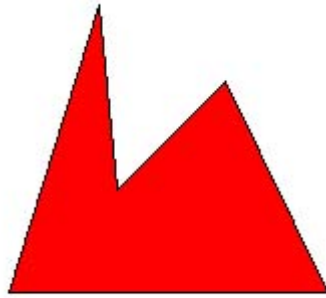
هر موقعیت در فضای سه بعدی می تواند با یک نقطه که از سه مقدار (X, Y, Z) استفاده می کند تعریف شود. با اینکه این بدیهی به نظر می رسد ولی قبل از ادامه مطمئن شوید که این موضوع را حس می کنید. مثل فضای دو بعدی در فضای سه بعدی نیز یک خط توسط دو نقطه واحد تعریف می شود. یک مفهوم مهم در فضای سه بعدی این است که ما می توانیم هر چیز را با تغییر مقادیر ذکر شده نقاط تغییر دهیم و یا جابجا کنیم. آخرین شکل پایه هندسی از دید برنامه نویسان سه بعدی مثلث است. مثلث قلب و روح موتور سه بعدی ما خواهد بود چراکه تقریباً تمام اشیائی که خواهید ساخت از مثلث مشتق می شود.



دلیل استفاده ما از مثلث ها به جای سایر اشکال هندسی پیچیده تر این است که مثلث ها تعداد زیادی خصوصیات ضمنی دارند که سایر چند ضلعی ها ندارند. اول اینکه مثلث ها ضمانت هم صفحه ای (coplanar) بودن خود را به عهده می گیرند. این موضوع بدیهی است در صورتی که قبول کنیم یک صفحه با استفاده از سه نقطه بوجود می آید. یعنی خصوصیت یک صفحه در فضا آن است که مستوی باشد لذا اگر کمتر از سه نقطه برای صفحه انتخاب کنیم صفحه تشکیل نمی شود و اگر تعداد نقاط بیشتر از سه شود ضمانت مستوی بودن آن منوط به محاسبات فوق العاده پیچیده ریاضی است که گهگاه غیر ممکن می نماید. دوم اینکه یک مثلث ضمانت می کند که محدب (convex) باشد. اگر با این اصطلاح آشنا نیستید به شکل زیر نگاه کنید که یک چند ضلعی بدون هیچ فرورفتگی را تداعی می کند :



بر خلاف این شکل که یک چند ضلعی فرو رفته را تداعی می کند :



ممکن است فکر کنید استفاده مطلق از مثلث محدودیت آور است اما اینگونه نیست. هر چند ضلعی هم صفحه قابل تجزیه به تعدادی مثلث است. در واقع اکثریت کارت های شتاب دهنده سه بعدی (3D Accelerator) با مثلث ها کار می کنند. بنابراین استفاده کردن از مثلث در موتور سه بعدی کیفیت بهتری را در خروجی روی این کارت ها به ارمغان می آورد.

فضاها در بعد سوم

تا کنون با مفهوم فضا آشنا شده اید، درست است منظور ما از فضا در این متن فضای سه بعدی اقلیدسی است که با سه محور X, Y و Z تعریف شده می باشد، که پیش تر به اجمال این سیستم را بررسی کرده ایم.

تعریف فضاها سه بعدی مختلف در برخورد با گرافیک سه بعدی بسیار مفید است. تعداد زیادی از این فضاها تعریف و استاندارد شده است در زیر سه مورد از اشتراکات آنها آمده است که در موتور Frog نیز استفاده شده است.

- فضای شیء **Object space** : در این سیستم یک شیء مجموعه ای از چند ضلعی ها فرض شده. اگر نیاز دارید که تعداد زیادی از نمونه های یک شیء را در مکان های مختلف دنیای خود قرار دهید به فضای شیء نیاز مندید. همچنین در مدل سازی نیز کاربرد خواهد داشت برای مثال طراحی شبکه در **3D Studio** در **object space** تعریف شده است.
- فضای جهانی **World space** : مهمترین سیستم تعدیل می باشد. در واقع جایی است که اشیاء قرار می گیرند و جایی که می بایست فیزیک، حرکات، نور ها و برخوردها شناسایی شوند. به این فضا به عنوان دنیای بازی خود نگاه کنید.

- فضای دید **View space**: فضای دیدی یا فضای چشم و یا همان فضای دوربین این سیستم مستقیماً به دوربین اشاره دارد. اشیاء در فضای جهانی جهت مشخص شدن اینکه از زاویه دید کدام قسمت های آنها در صفحه نمایش قابل دیدن است به حرکت در می آیند. البته برخی این فضا را فضای صفحه نمایش **Screen space** می نامند ولی قطعاً اینگونه نیست چرا که هنوز صفحات نمایش سه بعدی وجود ندارد.

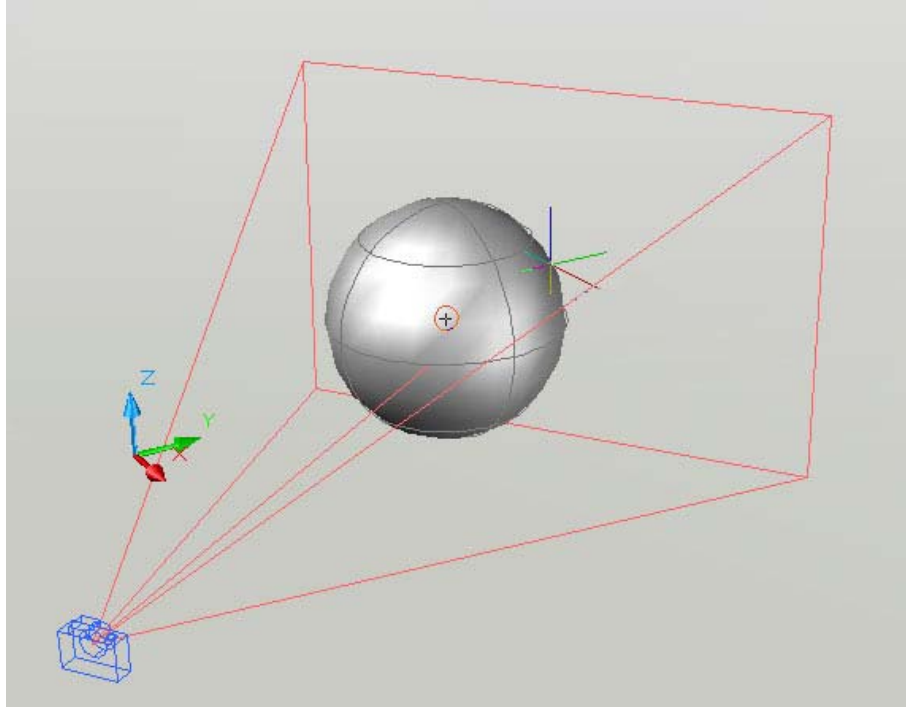
- فضای دو بعدی صفحه نمایش **Screen space**: چیزی که ما فضای صفحه نمایش می نامیم در واقع یک نمایه اجزاء همگون از صفحه نمایش است. منظور از اجزاء به طور مطلق پیکسل نیست **درگاه دید** وظیفه نگاشت فضای دید را به فضای صفحه نمایش دارد و همچنین مبدا در فضای صفحه نمایش وسط صفحه است (به منظور پشتیبانی از پرسپکتیو).

تعریف درگاه دید **View Port**: فرایندی که به ما اجازه می دهد تا هر عکس انتخاب شده را در محل معینی روی یک صفحه نمایش قرار دهند.

نکته مهم این که فضاها توسط ماتریس ها در کامپیوتر نمایش داده می شوند. هر یک سه محور ذکر شده یک جهت را نمایش می دهد: **X** مثبت راست، **Y** مثبت چپ و **Z** مثبت جلو یا عقب (بسته به مدل مورد استفاده چپ دست یا راست دست).

دوربین چیست

یک دوربین یک موجودیت ساده است که در **world space** جای گرفته است. در کوچکترین نقطه دید فراگرفته شده توسط یک مخروط است که نشان دهنده اجسام دیدنی **visible** در دنیای ما می باشد. معمولاً جای ما را در دنیا نشان می دهد.



همچنین دوربین مشخص می کند که شما در حال مشاهده چه چیزی هستید و بالا و پایین را برای شما مشخص می کند. تمام اطلاعات موجود در دوربین در **world space** تعریف شده است. دم دست ترین راه پیاده سازی استفاده از یک ماتریس 3×3 و بردار انتقال برای نمایش اطلاعات دوربین می باشد. اولین بردار نمایانگر سمت راست دوربین، دومی نمایانگر سمت بالا و سومی جهتی که شما در حال نگرستن به آن هستید، می باشد.

$$\begin{vmatrix} U_x & V_x & N_x \\ U_y & V_y & N_y \\ U_z & V_z & N_z \end{vmatrix}$$

بردار انتقال محل دوربین را نمایش می دهد. برای مثال شما در $(3, 10, 5)$ می باشید، راست در امتداد بردار $(1, 0, 0)$ ، بالا در امتداد بردار $(0, 0, 1)$ و شما در حال نظاره کردن چیزی در نقطه $(3, 15, 5)$ می باشید. نمایش ماتریسی آن به شرح زیر است :

$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{vmatrix}$$

بردار انتقال یا محل دوربین عبارت است از $(3, 10, 5)$

از یک فضا به فضایی دیگر

همچنان که دیدیم ما چهار فضای متفاوت را معرفی کردیم. حال انتقالات بین این فضاها را بررسی خواهیم کرد. انتقال از یک فضا به فضای دیگر مشابه انتقال دوربین تعریف می شود. تنها نکته مهم در این انتقالات این است که برای انتقال از **world space** به **view space** از معکوس انتقال دوربین استفاده می کنیم. برای مثال تصور کنید ما دوربین را به راست می بریم اما اشیاء به راست نمی روند بلکه به چپ می روند.

تحركات در فضاها

برش سه بعدی

برش در واقع اجتناب از رسم چند ضلعی ها بیرون از صفحه نمایش و همچنین شکستن آنها بطوریکه از مرز صفحه نمایش عبور نکنند می باشد. عملیات برش زیاد سخت نیست کافی است صفحه ای تعریف کنیم، میانه آنرا پیدا کرده و آنرا با هر نقطه ضرب داخلی کنید نتیجه فاصله هر نقطه تا صفحه خواهد بود. مجموع این ضرب های داخلی با هر نقطه مجموع فاصله را خواهد داد. با تقسیم اولین فاصله بر مجموع فواصل نسبتی را که می بایست صفحه را قطع کند بدست می آورید. حال که نسبت ها را دارید از معادله پارامتری خط استفاده کرده تا نقاطی را که در آنجا صفحه برخورد دارد را بدست آورید

معادله پارامتری خط :

$$X = (\text{initial point.x}) + (\text{ratio}) * (\text{direction.x})$$

$$Y = (\text{initial point.y}) + (\text{ratio}) * (\text{direction.y})$$

$$Z = (\text{initial point.z}) + (\text{ratio}) * (\text{direction.z})$$

اعداد را جایگزین کنید تا نقاط جدیدی بدست آورید. مشکل ترین قسمت اضافه کردن راس جدید به چند ضلعی است. راس جدید به این علت اضافه می شود چون یک یک چند راس برش خورده. برای این منظور شما باید راس جدید را در مکان اولین راس وارد کرده و باقی رئوس را بعد از آن قرار دهید.

چرخش سه بعدی

چرخش سه بعدی خیلی شبیه به چرخش در سیستم دو بعدی است فقط می بایست چرخش را برای هر صفحه انجام داد. از آنجایی که سه بعدی ها سه صفحه دارند چرخش می باست سه بار انجام شود. قبل از ادامه با چند اصطلاح آشنا شوید : roll به صفحه X-Y گویند، pitch به صفحه Y-Z گویند، و yaw که به صفحه X-Z گویند. برای فهم این اصطلاحات به این مثال ها توجه کنید : وقتی سر خود را به بالا و پایین حرکت

می دهید در واقع عمل pitch اتفاق می افتد. گوش چپ خود را روی شانه چپ و سپس گوش راست خود را روی شانه راست قرار دهید به این حرکت roll گویند. سر خود را به صورت افقی از سمتی به سمت دیگر ببرید در این حال حرکت yaw را داریم. بازی هایی نظیر doom یا wolfenstein تنها از چرخش yaw استفاده می کنند چون شما نمی توانید بالا یا پایین را ببینید. در زیر معدلات مربوط به چرخش یک نقطه در فضای سه بعدی را می بینید :

```
x'=z*sin(yaw)+x*cos(yaw)
y'=y
z'=z*cos(yaw)-x*sin(yaw)

x"=x'
y"=y'*cos(pitch)-z'*sin(pitch)
z"=y'*sin(pitch)+z'*cos(pitch)

x"'=y"*sin(roll)+x"*cos(roll)
y"'=y"*cos(roll)-x"*sin(roll)
z"'=z"
```

اگر فرم ماتریسی را می پسندید در زیر به آنها نیز اشاره شده :

ماتریس برای چرخش حول محور X :

$$\begin{bmatrix} \cos(a) & -\sin(a) & 0 & 0 \\ \sin(a) & \cos(a) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ماتریس برای چرخش حول محور Y :

$$\begin{bmatrix} \cos(a) & 0 & -\sin(a) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(a) & 0 & \cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ماتریس برای چرخش حول محور Z :

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(a) & -\sin(a) & 0 \\ 0 & \sin(a) & \cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

نکته : وقتی از توابع \sin و \cos استفاده می نمایید می بایست زوایا را بر حسب رادیان وارد کنید.

دید پرسپکتیو

دید پرسپکتیو از دو قسمت تشکیل شده : تبدیل سیستم سه بعدی به دو بعدی (صفحه نمایش) و ساخت نمای پرسپکتیو. برای قسمت اول یک معادله ساده وجود دارد:

$$X = x / z$$
$$Y = y / z$$

X, Y, Z هم یک نقطه در فضای سه بعدی است.

برای تبدیل به نمای پرسپکتیو کافی است از فرمول زیر استفاده کنید :

$$X = 800 * x / z$$
$$Y = 800 * y / z$$

این با فرض این است که وضوح صفحه نمایش شما 800X600 باشد. دلیل اینکه از عرض صفحه نمایش برای بدست آوردن نسبت ها استفاده شده این است که این باعث می شود تصویر واقعی به نظر برسد و فاصله سر تا مانیتور که حدود یک فوت برآورد شده ناخودآگاه حفظ شود. البته تکنیک های دیگری نیز نظیر Z-Sort و Z-Buffering برای بهینه سازی دید وجود دارد که در این مجال نمی گنجد..

محمد ملک مکان

منابع :

3D graphic Fundamentals – HCC Project Brochure

3D Graphics Modeling and Rendering mini-HOWTO Dave Jarvis v1.1, 27 March 2001

AutoCAD 200k Help

http://www.fastgraph.com/help/3D_geometry_system.html

http://www.fastgraph.com/help/object_space_and_world_space.html

<http://www.amazon.com/exec/obidos/ASIN/1576104001/gamedev>

<http://www.amazon.com/exec/obidos/ASIN/1883577705/gamedev>

<http://www.msi.com>

نقشه کشی 1 – فنی و حرفه ای 1382

هندسه ترسیمی – فنی حرفه ای 1385

آموزش گام به گام هندسه ترسیمی صنعتی – احمد ملکی، احمد قربانی