

osgEarth

یک sdk جغرافیای است. یک engine از openscene Graph است

ویژگی ها :

1. برنامه سه بعدی جغرافیایی
2. تجسم مدل و تصاویر را مستقیماً در حد امکان فراهم می کند
3. با استاندارد ها داده های نقشه ای را فراهم میکند

نیازمندی ها

1. نقشه پایه آماده باشد
2. به خدمات استاندارد wms یا Tms دسترسی داشته باشد
3. داده های ذخیره شده محلی با تصاویر مبتنی در وب باید یکپارچه سازی شود
4. در زمان اجرا لایه های جدید اضافه شود
5. باید با داده های که در طول زمان تغییر می کند مقابله کرد.

راهنمای استفاده

ابزار های osgearth

1-osgearth_viewer

برای نمایش و خواندن فایل نقشه : این دستور یک فایل می خواند و یکسری آرگومان میگیرد

osgearth_version

ورژن استفاده شده را fffd

osgearth_cache

برای مدیریت کش نیز استفاده می شود و اصولاً غیر تعاملی می باشد که با آرگومان seed می آید

osgearthth_conv

محتویات یک tilesource در یک پکیج دیگری کپی می کند.

osgEarth_package

یک بسته مبتنی بر tms قابل توزیع مجدد از یک فایل earth تولید میکند منظور از tms همان tile map service میباشد

osgEarth_tfs

یک مجموعه داده tfs از منبع ویژگی مانند Shapefile می سازد. با پیش پردازش ویژگی های خود در ساختار شبکه ای ارائه شده توسط tfs می توانید عملکرد مجموعه داده های بزرگ را به طور قابل توجهی افزایش دهید. علاوه بر این بسته tfs تولید شده می تواند توسط هر وب سرور استاندارد دی ارائه شود

osgearth_backfill

یک ابزار تخصصی برا پردازش داده ها tms نیز می باشد. برخی از خدمات نقشه برداری وب از مجموعه داده های کاملاً متفاوتی در سطوح بزرگنمایی مختلف استفاده می کنند. به ما این امکان را می دهد که علاوه بر ساخت فایل tms با روش های معمول با این ابزار می شود میزان بزرگنمایی و لود کردن را تغییر داد.

osgearth_overlayviewer

سبب نشان دادن دو نقشه یکی همان نقشه معمولی و دیگری فرستوم های مرزی را برای همپوشانی نشان می دهد.

osgearth_boundarygen

ساختن یک مدل هندسی برای اضافه کردن به بیس نقشه استفاده می شود.

استفاده از earthfile

یک توصیف xml از یک نقشه است این روش ساده و سریعترین راه برای راه اندازی یک نقشه نیز می باشد. این فایل با پسوند **osg** می باشد
امکانات earthfile

- نوع نقشه را نیز مشخص میکند
- منابع تصویر، ارتفاع، بردار و مدل برای استفاده
- در حافظه کش ذخیره میکند

سینکس های از جمله **map** , **image** , **evaluation** , **cach** را دارد.

حافظه پنهان

با راه اندازی حافظه پنهان سبب می شود بسیار از کار ها سریعتر نیز انجام شود. این فقط برای استفاده به عنوان حافظه پنهان موقت و نه در قالب انتشار داده در نظر گرفته شده است. ساختار در هر زمان ممکن است تغییر کند

spatial Refrence

مجموعه ای از مختصات را به یک مکان واقعی متناظر روی زمین نگاشت می کند.
که به چهار روش نیز مقدار دهی نیز می شود

- [Coordinate System Type](#)
- [Horizontal Datum](#)
- [Vertical Datum](#)
- [Projection](#)

ویژگی ها و نماد ها

ویژگی های هندسی برداری هستند. بر خلاف تصاویر و داده های ارتفاعی (که شطرنجی هستند)، ویژگی وضوح نمایشگر گسسته ای ندارد. **osgearth** میتواند در هر سطحی آنها را نمایش دهد.

ویژگی ترکیبی از سه جز نیز می باشد

- ترکیبی از **vector geomerty**
- ویژگی ها مثل نام ها و مقادیر
- مرجع فضای(مختصات)

ایجاد ویژگی لایه

مدل لایه های ما به دو صورت نیز می باشد

- شطرنجی شده به عنوان یک لایه تصویر
- یک لایه مدل

شطرنجی کردن لایه بسیار ساده است و سعی می شود تصاویر را به صورت کاشی در بیاورند و از آن کاشی تصویر لایه معمولی به وجود آورند.

در روش دوم از طریق یک فرایند مدل را به نقاط و اشکال هندسی تبدیل میشود این روش در **osgEarth** بیشتر کاربرد دارد.

ایجاد کامپوننت

ممکن است برای یک لایه به ویژگی های احتیاج داشته باشید که به صورت کامپوننت استفاده شود .

یک نمونه از کامپوننت نیز `style` نیز می باشد که به صورت `css` می توان به آن مقداری کرد

ویژگی های زمین

یک سری از ویژگی ها با زمین نقشه نیز ارتباط دارند از جمله مواردی که میتوان درباره زمین نام برد:

- خیابان ها که از خطوط زمین پیروی میکنند
- درختان زمین
- و نقشه برداری موضوعی مانند رنگ آمیزی منطقه یک کشور بر اساس جمعیت آن

این ابزار انواع مختلفی از زمین را ارائه می دهد.

بسته نقشه `map clampin`

این روش ساده ترین روش می باشد هنگام تهیه ویژگی های صفحه نقشه لایه های ارتفاع را در نقشه نمونه می کند

این روش سبب بهبود کیفیت می شود :

- بسته به وضوح انتخاب شده ، می تواند نمونه ای از داده های ارتفاع در نقشه باشد. برای تعداد زیادی از ویژگی ها ، می تواند CPU فشرده و وقت گیر باشد.
- نمونه برداری دقیق است و برای هر نقطه از هندسه انجام می شود. برای بهبود سرعت تدوین می توانید در سانتر وئید هر ویژگی نمونه برداری کنید. بسته به وضوح هندسه ویژگی ، ممکن است برای دستیابی به کیفیت بهتر نیاز به کاهش داده های خود داشته باشید.
- کیفیت رندر در مقایسه با سایر روشها خوب است.

با دستور زیر نیز فعال می شود

```
altitude-clamping: terrain; // terrain-following on
```

```
altitude-technique: map; // clamp features to the map data
```

```
altitude-resolution: 0.005; // [optional] resolution of map data to clamp to
```

روش Draping

این روش زمانی استفاده می شود که زمین های ناهمواری در نقشه نیز وجود داشته باشد و `osgearth` این بافت را به عنوان یک ویژگی را به یک بافت RTT تبدیل می کند و سپس آن را روی زمین قرار می دهد.

این روش باعث بهبود عملکرد و کیفیت نیز می شود.

- ویژگی هایی را کاملاً با زمین مطابقت خواهد داد. هیچ نگرانی در مورد وضوح وجود ندارد.
- ممکن است هنگام ارائه خطوط یا لبه های چند ضلعی ، آثار باستانی را بدست آورید. بافت پیش بینی شده از نظر اندازه محدود است و هرچه بیشتر منطقه ای که باید آن را بپوشاند ، وضوح تصویر پیش بینی شده کمتر است. این بدان معنی است که در عمل کشیدن برای چند ضلعی ها برای خطوط مفیدتر است.
- مصنوعات مخلوط غیر منتظره ممکن است ناشی از کشیدن بسیاری از هندسه های شفاف در بالای یکدیگر باشد.

فعال کردن این روش

altitude-clamping: terrain; // terrain-following on

altitude-technique: drape; // drape features with a projective texture

روش gpu clamping

این روش از یک تکنیک دو فاز استفاده می کند: ابتدا از نمونه برداری از عمق استفاده می کند تا هر راس را به پوست زمین در یک سایه بان راس ببندد. ثانیاً ، این یک الگوریتم عمیق در سایه قطعه قطعه برای کاهش z-fighting اعمال می شود

- این برای خطوط مناسب است ، اما کمتر از آن برای چند ضلعی است زیرا برای انجام یک بسته بندی تقریبی خوب نیاز به کاهش فضای داخلی یک ضلعی دارد.
- سریع است ، کاملاً در زمان اجرا اتفاق می افتد و از پردازش gpu یز استفاده میکند
- هیچ اثر لبه ای وجود ندارد

داده های بزرگ

زمانی که یک سری ویژگی برای نقشه بیش اندازه بزرگ نیز باشند سبب می شود منابع عملکردشان نیز ضعیف شود. راه حل برای آن کاشی کاری و صفحه بندی است که روش های مختلفی نیز دارد که با هم مورد بررسی نیز قرار می دهیم

ویژگی های لایه ای

در این روش از یک تگ layout نیز استفاده می شود می توان با استفاده از **cropping features** مقدار مورد نیاز از کاشی را برداشت و ویژگی را به آن نسبت داد. و با استفاده از **level** میشود سطح های مختلف را نیز مشخص نمود

استفاده osfearth در کد

```
osg::Node* globe = osgDB::readNodeFile("myglobe.earth");
```

برای گرفتن یک صحنه از earth فایل و ذخیره در یک نود

ایجاد یک map در لینک زیر <https://docs.osgearth.org/en/2.10/developer/maps.html>

نود ها نقشه میتواند به شکل های مختلفی داده را نشان دهد می تواند به صورت آنلاین داده بگیرد میتواند از فایل earth استفاده کند یا میتواند صحنه را بسازد و سپس به آن لایه های مختلف نیز اضافه کند .

ملزومات برای راحتی و آسانی کار

در برنامه های کل زمین معمول است که شما می خواهید چیزی را از نزدیک ببینید (مانند هواپیما در ارتفاع) در حالی که زمین و افق آن را از فاصله دور می بینید. این مسئله برای سخت افزار گرافیکی مدرن مشکلی ایجاد می کند زیرا دقت بافر عمق استاندارد به شدت از اشیاء نزدیکتر به دوربین حمایت می کند. **LogarithmicDepthBuffer** یکی از راه حل های این مشکل است. برای تنظیم مجدد مقادیر بافر عمق GPU استفاده می کند تا بتواند در این نوع سناریو بهتر استفاده شود.

```
LogarithmicDepthBuffer logdepth;  
logdepth->install( view->getCamera() );
```

فرمت قالب بندی مختصات

از فرمت ها برای قالب بندی مختصات جغرافیایی به عنوان یک رشته استفاده می شود .

دو نوع فرمت نیز وجود دارد: **MGRSFormatter** و **LatLongFormatter** که یک **GEOPoint** نیز می گیرد و یک **string** نیز برمیگرداند. هر کدام از این فرمت ها قالب های مختلفی را میگیرد.

ابزار **Mousecoordstool**

این ابزار باعث می شود بتوان مختصات نقشه را گزارش دهد

بخش **coordinate system**

بین **osg**, **opengl**, **osgearth** چندین سیستم مختصات مختلف و فریم های مرجع در حال استفاده وجود دارد و می تواند گاهی گیج کننده باشد که این است. در اینجا برخی از اصول اولیه ذکر میکنیم

آشنایی با **opengl** را میتوانید در لینک http://www.songho.ca/opengl/gl_transform.html مشاهده کنید

فضای مدل (یا شی) به مختصات واقعی در هندسه (مانند کاشی های زمین، مدل هواپیما و غیره) اشاره دارد. در **osg** مختصات ممکن است مطلق باشند و یا از تبدیل **osg** استفاده بکنند مختصات به دو صورت جهانی و محلی تعریف میشوند

مختصات جهان به صورت مطلق بیان می شود. آنها عوض نمی شوند. مختصات محلی تبدیل شده اند تا آنها را به یک نقطه مرجع (در مختصات جهانی) نسبت دهند.

چرا از مختصات محلی استفاده کنیم؟ زیرا سخت افزار **OpenGL** فقط می تواند مقادیر 32 بیتی را برای مکان های رأس مدیریت کند. اما در سیستمی مانند **osgEarth**، ما باید مکان هایی را با مقادیر بزرگ نشان دهیم و نمیتوانیم این کار را بدون فراتر رفتن از محدودیت های دقت 32 بیتی انجام دهیم. راه حل استفاده از مختصات محلی است.

نمایش مختصات

فضای دید (که گاهی اوقات به آن فضای دوربین یا چشم می گویند) موقعیت شکل هندسی را نسبت به خود دوربین بیان می کند. دوربین در مبدا (0,0,0) قرار دارد و محورهای مختصات عبارتند از:

+X : Right

+Y : Up

-Z : Forward (direction the camera is looking)

فضای **view** روی **cpu** کار میکند و اصولا نمایش و دوربین در نقطه 0,0,0 می باشد

مختصات کلیپ

همان چیزی است که پس از اعمال حجم نمایش به دست می آورید فضاهای کلیپ از مختصات همگن 4 بعدی استفاده می کند. محدوده مقادیر در فضای کلیپ، شکست دوربین را در بر می گیرد و به این صورت بیان می شود:

X : [-w..w] (-w = left, +w = right)

Y : [-w..w] (-w = bottom, +w = top)

Z : [-w..w] (-w = near, +w = far)

W : perspective divisor

وقتی نیاز به نمونه برداری یا دستکاری اطلاعات عمق در صحنه دارید، فضای کلیپ مفید است.

مراجع

<https://docs.osgearth.org/en/2.10/about.html>