Computer Grafica

Applicazioni Web-based

Progettazione di applicazioni web grafiche.

"Leonardo da Vinci"
Firenze
ESAME DI STATO
2015-2016

Asefa Filmon Arefayne
V Informatica

Indice

Progetto
1. Computer Grafica
   1.1 Quindi cos'è ?
   1.2 Un accenno alla storia …
   1.3 GPU
2. Informatica
   2.1 WebGL ?
   2.2 Gli svantaggi
3. Un' applicazione Grafica
   3.1 Interattività
   3.2 Fotorealismo
4. Sistemi e reti
5. Tecnologie di Progettazione
6. Matematica
7. Inglese
8. Conclusioni

Prefazione

Ho sempre avuto la passione per la computer grafica e con questo progetto ho cercato di creare una piccola applicazione web auto espositiva. L'applicazione è una pagina statica realizzata in HTML, CSS, JS e WebGL ed è collocata sul servizio di l'hosting Paperplane.

Qui potrai trovare tutte le risorse dedicate al progetto:

Il sito:
computer-graphics.paperplane.io

documento online:
https://goo.gl/CR288Y

Computer Grafica

"La disciplina che studia le tecniche e gli algoritmi per la visualizzazione di informazioni numeriche prodotte da un elaboratore."

R. Scateni, P. Cignoni, C. Montani, R. Scopigno.
Fondamenti di Grafica Tridimensionale Interattiva,
2005, McGraw Hill

Quindi cos'è ?

Un po' tutto questo:

- Creazione immagini 2d sintetiche e animazioni;
- Modellazione 2D, 3D, anche con comportamenti fisici;
- Computer Aided Design;
- Rendering delle scene, cioè creazione delle immagini simulando la proiezione ottica delle scene sulla camera;
- Animazione;
- Interfacce grafiche dei computer;
- Realtà virtuale;
- Enhancement video televisivo;
- Visualizzazione scientifica e dell'informazione;

Un accenno alla storia ...

La computer grafica nasce innanzitutto per scopi industriali e militari nella seconda metà degli anni '60,ma era esclusivamente utilizzata da quei pochi computer dotati di grande potenza di calcolo.
Con la diffusione di personal computer con schede video di grande versatilità e dotati di grande potenza di calcolo,la grafica computerizzata è diventata dominio di tutti.
Al giorno d'oggi la grafica computerizzata è parte integrante di molti ambiti professionali e di consumo come i videogiochi, ritocco fotografico,montaggio di filmati, industria cinematografica ma anche per la visualizzazione di dati tecnico/scientifici.

GPU

La GPU(graphics processing unit) è una tipologia particolare di coprocessore che si contraddistingue per essere specializzata nel rendering di immagini grafiche.
Il suo tipico utilizzo è come coprocessore della CPU, è tipicamente implementata come microprocessore monolitico e, da alcuni anni, viene anche implementata assieme alla CPU nel medesimo circuito integrato.E' su di essa che le applicazioni grafiche fanno affidamento.

Informatica

In una pagina web statica, il contenuto della pagina viene stabilito nel momento in cui si crea e si memorizza la pagina sul server web e quindi ogni volta che un utente accede a una pagina statica, questa gli presenta le stesse informazioni, a meno che non venga aggiornata, cioè riscritta.
La pagina è stata realizzata in HTML,CSS e JAVASCRIPT ma i contenuti grafici sono stati realizzati con il framework THREE.JS.
THREE.JS è un framework che fa affidamento alle WebGL,e che le rendono più semplici riducono molto le porzioni di codice da scrivere soprattutto quelle di inizializzazione.

WebGL ?

"WebGL (Web Graphics Library) è un API(Application Programming Interface) JavaScript per rendering di grafica 3D interattiva e grafica 2D all'interno di qualsiasi browser Web compatibile , senza far utilizzo di plug-in."

Quindi le WebGL portano con se molti vantaggi ed oltre ad essere cross-platform e open source permette di utilizzare l'hardware grafico direttamente da un browser.

Gli svantaggi

Non tutte le versioni dei browser supportano la libreria e bisogna avere un hardware compatibile con OpenGL 2.0 ma il problema più grande sta nel fatto che le prestazioni sono molto basse rispetto ad un'applicazione desktop(DirectX,OpenGL).

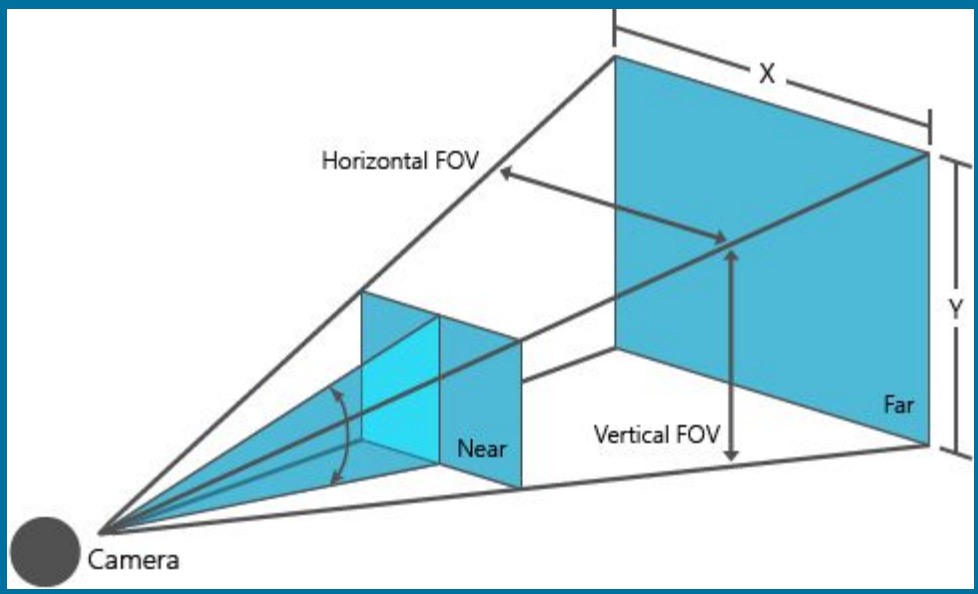
Come si include il framework in un file HTML ?

Basta scaricare il file ed includerlo come un qualunque file javascript.

```
93
94              <script src="js/three.js"></script>
95
```

Adesso bisogna definire la scena dove andremo a disegnare.

```
17      var scene = new THREE.Scene();
18      var camera = new THREE.PerspectiveCamera( 75, window.innerWidth / window.innerHeight, 1, 3000 );
19
```



camera

- Il primo parametro (75) definisce il campo visivo verticale della telecamera, in gradi (dal basso verso l'alto del campo visivo). È l'estensione di tutto ciò che si può osservare sullo schermo in ogni momento. Il campo visivo orizzontale viene calcolato usando quello verticale.
- Il secondo parametro(aspect-ratio) (window.innerWidth / window.innerHeight) definisce le proporzioni della telecamera. In genere è consigliabile usare la larghezza dell'elemento riquadro di visualizzazione divisa per l'altezza, altrimenti l'immagine ottenuta apparirà schiacciata.
- Il terzo parametro (0.1) definisce il piano frustum della telecamera vicino (cerca "Vicino" nella figura). In questo caso, il piano frustum vicino coincide quasi con il piano *xy* (cioè con lo schermo).
- L'ultimo parametro definisce (1000) il piano frustum della telecamera lontano (cerca "Lontano" nella figura). In questo caso, quando un oggetto va oltre le ±1000 unità, viene considerato esterno al mondo visibile di Three.js e viene tagliato fuori dalla visuale.

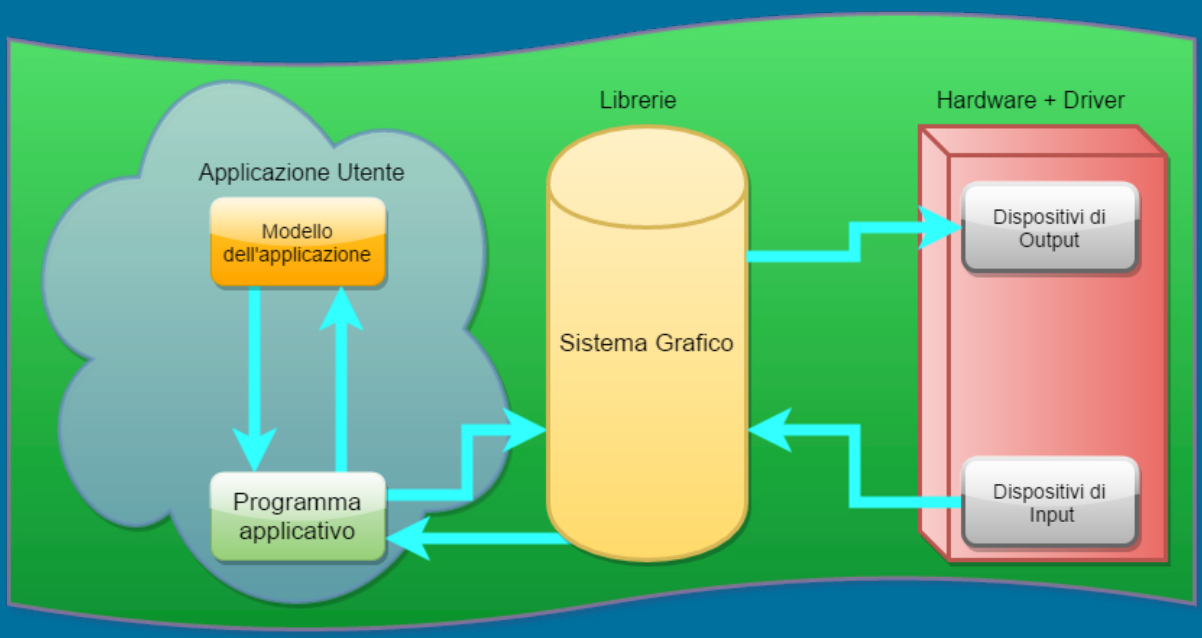
Dopo aver definito la camera si potranno aggiungere alla scena modelli,luci,effetti speciali,etc.

Adesso la nostra applicazione grafica è quasi completa basterà creare un ciclo dove disegneremo la nostra scena.
Per semplicità mostrerò un esempio di una semplice applicazione in THREE.JS.

```html
<html>
    <head>
        <title>My first Three.js app</title>
        <style>
            body { margin: 0; }
            canvas { width: 100%; height: 100% }
        </style>
    </head>
    <body>
        <script src="js/three.js"></script>
        <script>
            var scene = new THREE.Scene();
            var camera = new THREE.PerspectiveCamera( 75, window.innerWidth/window.innerHeight, 0.1, 1000 );

            var renderer = new THREE.WebGLRenderer();
            renderer.setSize( window.innerWidth, window.innerHeight );
            document.body.appendChild( renderer.domElement );

            var geometry = new THREE.BoxGeometry( 1, 1, 1 );
            var material = new THREE.MeshBasicMaterial( { color: 0x00ff00 } );
            var cube = new THREE.Mesh( geometry, material );
            scene.add( cube );

            camera.position.z = 5;

            var render = function () {
                requestAnimationFrame( render );

                cube.rotation.x += 0.1;
                cube.rotation.y += 0.1;

                renderer.render(scene, camera);
            };

            render();
        </script>
    </body>
</html>
```

Un’ applicazione Grafica

Vi è una descrizione di qualche tipo (procedurale o meno) del mondo che deve essere rappresentato. La produzione di tale descrizione (modello) prende il nome di modellazione.
Per le descrivere modelli complessi vengono utilizzati programmi software ad esempio 3DSMAX,Maya, AutoCAD.
Da tale descrizione si ottiene una immagine bidimensionale, questo processo è chiamato globalmente rendering.
La sequenza di procedure ed algoritmi che implementano il rendering prende il nome di pipeline grafica (ogni API ne utilizza una sua).
L’immagine ottenuta viene quindi visualizzata sullo schermo (in applicazioni interattive, per esempio) o salvata su file.



Interattività

Real Time

I motori 3D in tempo reale (real time engine) sono impiegati laddove sia necessario produrre immagini tridimensionali interattive. Con questa espressione si indica la capacità di calcolare, e quindi di mostrare a schermo (oppure tramite altri dispositivi ottici, ad esempio occhiali per la realtà virtuale) le immagini in brevissimo tempo, tale da ottenere un certo numero di immagini al secondo, tipicamente 30-60 immagini al secondo per avere un effetto di movimento realistico(abbreviato spesso come fps,"Frames Per Second). Questi requisiti di velocità possono essere raggiunti con varie tecniche, evolutesi nel tempo grazie soprattutto all'invenzione di dispositivi hardware dedicati allo scopo: gli acceleratori grafici 3D. Questi dispositivi, costituiti in sostanza da un coprocessore matematico e da una certa quantità di memoria RAM, svolgono certe funzioni matematiche estremamente ottimizzate e consentono allo sviluppatore di sgravare la CPU da un'enorme quantità di calcoli, permettendo quindi di realizzare motori grafici più raffinati e più veloci(GPU). L'introduzione degli acceleratori grafici ha decretato la formazione di due sottocategorie di motori in tempo reale: i motori grafici software e i motori accelerati in hardware. Nonostante il nome possa trarre in inganno, si parla in entrambi i casi di software. La differenza consiste nel fatto che i primi sfruttano esclusivamente la CPU (ed eventualmente la FPU e le istruzioni SIMD come MMX, SSE, 3DNow!, ecc.) per effettuare i calcoli geometrici necessari, mentre i secondi relegano molte delle funzioni primarie (come la trasformazione, l'illuminazione, l'applicazione delle texture, ecc.) all'acceleratore hardware. Ovviamente entrambi gli approcci portano dei vantaggi e degli svantaggi: i motori software renderizzano le immagini esattamente nel modo previsto dal programmatore ma risultano lenti, quindi non possono produrre immagini di elevata qualità per l'eccessiva quantità di calcoli necessari; i motori accelerati, invece, sono estremamente veloci e producono immagini di elevata qualità, ma richiedono la presenza di hardware dedicato e l'accuratezza delle immagini è soggetta al particolare acceleratore utilizzato. I motori 3D in tempo reale trovano largo impiego nella realizzazione di videogames, simulatori, interfacce grafiche, realtà virtuale.

Fotorealismo

pre-rendering

E’ così che vengono chiamate le applicazioni che producono immagini di qualità prossima o addirittura paragonabile a immagini di scene reali. I motori grafici di questa categoria sono esclusivamente di tipo software, cioè non si appoggiano su hardware di accelerazione 3D. Nelle applicazioni in cui vengono sfruttati i motori 3D fotorealistici, la precisione e la qualità delle immagini renderizzate è prioritaria rispetto alla velocità di calcolo. I motori grafici di questa categoria sfruttano algoritmi molto sofisticati per simulare fedelmente gli effetti ottici di diffusione, rifrazione, riflessione, pulviscolo, proiezione di ombre, aberrazioni cromatiche e altri effetti che contribuiscono a rendere la scena estremamente realistica. Molti di questi

algoritmi non possono essere implementati nei motori in tempo reale per la loro estrema complessità, oppure vengono implementati in forma semplificata e approssimativa. Alcuni di questi algoritmi sono il Ray Tracing, il Photon Mapping, e altri. Data la mole di calcoli necessaria, generalmente i motori 3D fotorealistici sono progettati per essere eseguiti su macchine multiprocessore e su cluster(render farm). Questi motori grafici sono usati per la realizzazione di opere artistiche, progettazione architettonica e meccanica, design, produzioni cinematografiche (per effetti speciali o per interi film d'animazione). Inoltre i modelli presi in questione sono molto dettagliati e composti da molti poligoni(milioni).

Sistemi e Reti

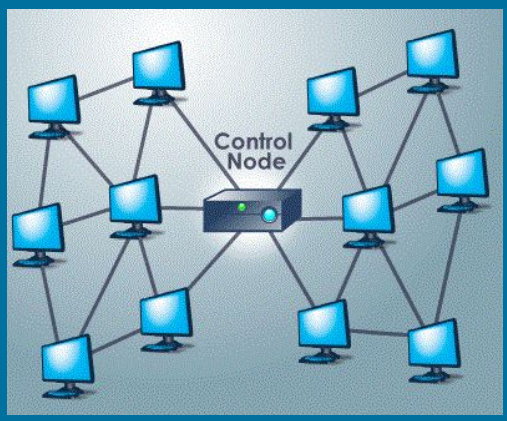
grid computing(renderfarm)

L'uso di renderfarm nell'industria dell'intrattenimento può essere vista come una delle prime applicazioni del grid computing.

Definizione

“Una renderfarm è un insieme di calcolatori collegati tra loro, chiamati nodi, allo scopo di elaborare le immagini di computer grafica.”

Molto spesso il cloud computing viene confuso con il grid computing ma la principale differenza e che nel grid abbiamo poche istanze che vengono eseguite contemporaneamente, però queste normalmente sono estremamente esose di risorse e possono richiedere l’utilizzo esclusivo dell’hardware del sistema costringendo le nuove istanze a rimanere in attesa.
Un sistema grid permette ai suoi utenti di poter massimizzare le risorse non utilizzate all'interno della loro rete ed è generalmente usato per risolvere problemi di larga scala(matematici e scientifici) che richiedono una grande potenza computazionale.



la grandezza di un sistema grid può variare:

- Piccolo: (internal network) computer con lo stesso sistema operativo (sistema omogeneo)
- Grande: sistema complesso di computer con diversi sistemi operativi (sistema eterogeneo)

Le reti sono realizzate con l'aiuto di librerie software (“middleware”) che permettono ai computer di eseguire processi o una serie di applicazioni attraverso l'intera rete di macchine.
Senza di esse la comunicazione attraverso il sistema sarebbe impossibile.
Inoltre vi è almeno un host, chiamato “control node,” che ha il compito di privilegiare e pianificare i compiti di tutta la rete.Il "control node" determina per ogni attività a quali risorse sarà acconsentito accedere e monitora il sistema per fare in modo che non venga sovraccaricato. Nel caso delle renderfarm i “nodi” vengono chiamati “rendernodes”.
Molti film famosi fanno utilizzo delle renderfarm (es. Toy story,King Kong etc)… In media per renderizzare i singoli fotogrammi del film avatar ci sono volute 24 ore e per alcuni anche 48. La renderfam utilizzata per realizzare questo film disponeva di una rete con collegamento 10 Gigabit Ethernet, 34 rack che costituivano il nucleo di calcolo,40 mila processori e 104 terabyte di memoria il tutto raffreddato ad acqua.Un flusso di quasi 18 gb al minuto per un film di 166 minuti.

Tecnologie di Progettazione
SISTEMI DISTRIBUITI

Definizione

“Un sistema distribuito è costituito da un insieme di applicazioni logicamente indipendenti che collaborano per il perseguimento di obiettivi comuni attraverso una infrastruttura di comunicazione hardware e software.”

Tra le tante definizione questa è meno precisa ma più corretta.

Un sistema informatico distribuito possiede almeno:

- elaborazione distribuita: le applicazioni risiedono su più host che collaborano tra loro;
- base di dati distribuita: il patrimonio informativo è ospitato su più host.

Internet e il Web sono due esempi di sistemi distribuiti, ma non tutti i sistemi distribuiti sono collegati a Internet e Web e sono utilizzabili da tutti gli utenti come appunto le render farm.

Come trasparenza si intende il concetto di “vedere” il sistema distribuito non come un insieme di componenti ma come un unico sistema di elaborazione: l’utente non deve accorgersi che di interagire con un sistema distribuito ma deve avere la percezione di utilizzare un singolo elaboratore.

vantaggi

sistemi distribuiti offrono spesso un miglior rapporto prezzo/qualità rispetto ai sistemi centralizzati.

svantaggi

I sistemi distribuiti sono più complessi e richiedono strumenti per l’interconnessione degli host e tecniche per l’instradamento corretto dei messaggi e dei dati.

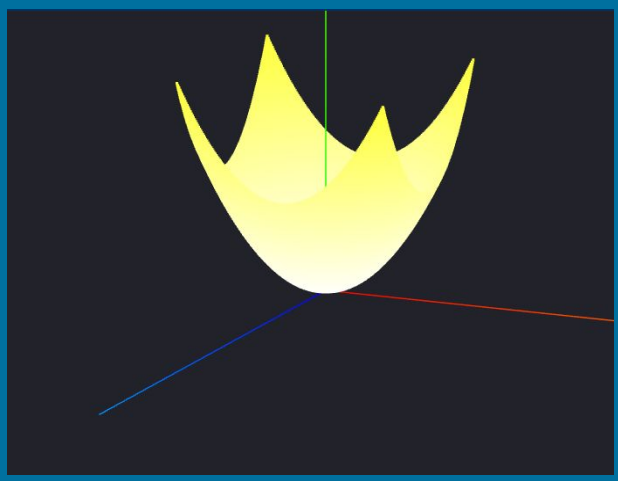
Matematica

RAPPRESENTAZIONE GRAFICA DI FUNZIONI A DUE VARIABILI

definizione

Si dice funzione reale di due variabili reali una relazione che associa a ogni coppia ordinata di numeri reali(x;y),appartenente a un sottoinsieme S di $R\,x\,R$ *,uno e un solo numero reale z.*

Grafico per Punti:



Mentre le funzioni reali ad una variabile reale (es: y=f(x), con x,y reali) erano rappresentabili efficacemente su un piano cartesiano bi-dimensionale, ora questo piano non sarà più sufficiente.

Costruire il grafico di una funzione in due variabili a mano può essere complicato, il metodo di creazione del grafico per punti è veramente efficace soprattutto se svolto da un elaboratore(computer).

Prendiamo in esempio la funzione $z = x^2 + y^2$

Per rappresentare una funzione mediante il precedente metodo bisogna prima calcolare il dominio della funzione che in questo caso è tutto $R\,x\,R$ o R^2 ovviamente non potendo disegnarlo dobbiamo studiare il suo grafico in un dominio molto più ristretto.
Costruiamo un reticolo sul piano xy e per ogni punto del reticolo calcoliamo i corrispondenti valori di z ad esempio con i punti $x = 1$ $y = -2$ avremo una $z = 5$.

nella figura è rappresentata la funzione $z = x^2 + y^2$ su un sistema di coordinate cartesiane tridimensionale (lo spazio), il risultato è un paraboloide.

Inglese

web apps

definition

“App is shorthand for ‘application’.In the online world of web browsers and smartphones,apps are usually much lighter programs focused on a single more specific task.”

Computer graphics application allow the creation,transformation and display of images. The operator can work with an original image, produced on the computer,or use a photograph or picture that has been digitized and entered into the computer. The dimensions,layout,colours and other features of the image can then be changed in a veriety of ways.Some software works with motion pictures,enabling the creation of spectacular special effects in films as well as higly realistic animation.
Web apps, in particular, run these tasks inside the web browser and often provide a rich,Interactive experience.Alll the information you need is pulled into the web app dynamically every time you ask for it(you make the request).
Web apps offer four major benefits.

advantages

1. I can access my data from anywhere.I can get to it on a web browser from any computer that’s connected to the internet.
2. I’ll always get the latest version of my app. No need to manually upgrade to a new version every time.
3. It works on every device with a web browser .Anyone can reach it from a browser on any web-connected device,whether it’s a desktop computer,laptop or mobile phone.
4. it’s safer because web apps run in the browser and i never have to download them onto my computer.

disadvantages

1. It’s need a fast Internet connection, especially in case I have to load large files.
2. The performance of a web apps are the worst for this run only certain tasks,in a web-based graphics application you need to limit number of objects in the scene.

Conclusioni

Curiosità:
Lines of codes(LOC, righe di codice) = 800+

Desidero ringraziare tutti coloro che mi hanno aiutato nella stesura della tesi con suggerimenti critiche ed osservazioni.