

Tesina sulla grafica 2D e 3D utilizzata dalle prime console e dai primi Personal Computer fino ad oggi.

Un po' di storia e di concetti base

Prima di entrare nel vivo della grafica 2D e su come <<girava>> sui primi dispositivi elettronici è meglio fare una panoramica generale sui primi strumenti che si utilizzavano per visualizzare le immagini sugli schermi.

Il primo di questi è stato il CTR.

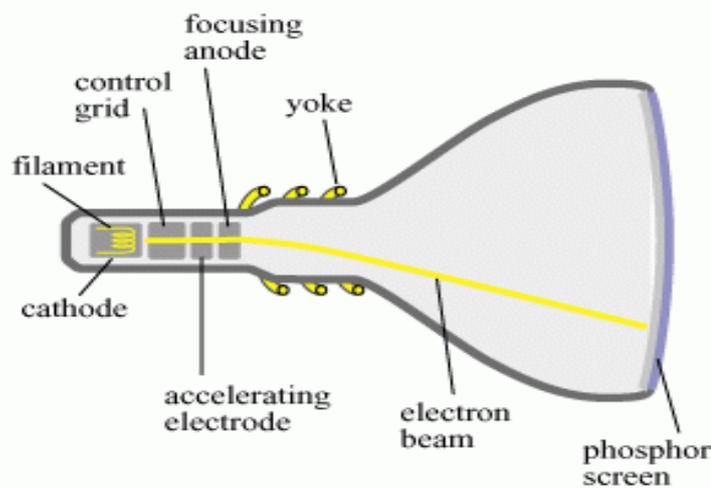


Figura 1: CTR.

Il CRT è l'acronimo di "tubi a raggi catodici" (*Cathode Ray Tube*). Questo dispositivo, come possiamo vedere dall'immagine, è composto da un emettitore di un fascio di elettroni che raggiunge uno schermo fosforescente

(monocromatico per gli oscilloscopi o tricromatico per i monitor) grazie un tubo a vuoto. Per merito di questa fosforescenza

Gruppo: Macrì Gabriele, Morchio Massimo.

30/11/2020

Classe VBinf

l'immagine non scompare quando l'emissione del fascio cessa, ma persiste per qualche frazione di secondo; in questo modo verrà visualizzata un'immagine apprezzabile all'occhio umano. Il primo gioco visualizzato su CRT fu "Tennis For Two" nel 1958...ma questo tipo di schermo presto abbandonò il suo utilizzo per la visualizzazione di immagini geometriche (tipiche dei videogiochi) sostituendole con le forme d'onda per gli output degli oscilloscopi. Questo dispositivo però ha consentito l'invenzione della **rasterizzazione** delle immagini. Infatti l'emettitore di elettroni poteva raggiungere qualunque fosforo sullo schermo e per ovviare a questo problema si sono implementati dei magneti capaci di direzionare in una posizione specifica gli elettroni, gli si facevano percorrere riga per riga tutto lo schermo, ripetendo questa azione più volte. In questo modo si facilitava la trasmissione dell'immagine all'elaboratore: nacque così la tecnica **Raster**.

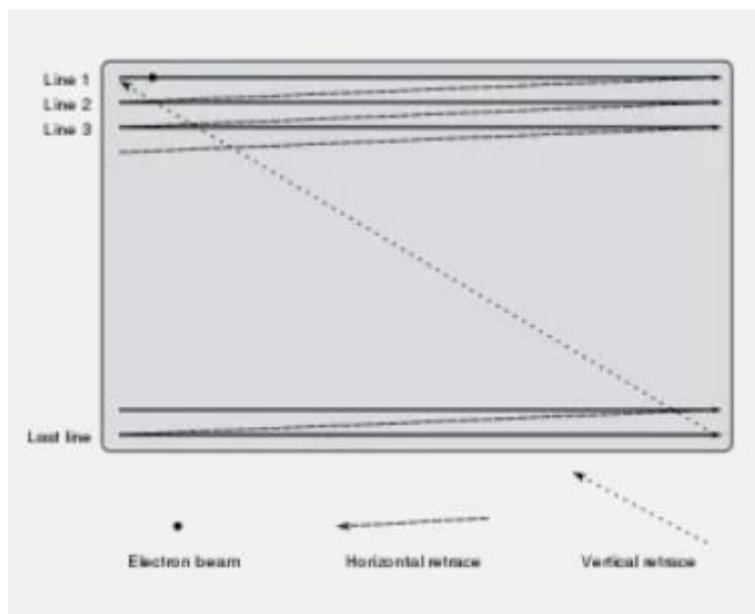


Figura 2: Output CRT.

Dopo aver introdotto un metodo per "dividere" lo schermo in parti orizzontali, per le immagini dei monitor si è pensato un metodo più efficace: Oltre alla suddivisione orizzontale è stata

Gruppo: Macrì Gabriele, Morchio Massimo.

30/11/2020

Classe VBinf

introdotta anche quella verticale, più efficace per la scrittura a video d'immagini con coordinate cartesiane...ecco creata la **grafica raster** o **bitmap**! L'elemento unitario, ovvero la "casella" creata dall'intersezione degli assi viene definita **pixel**, a quest'ultimo gli viene associato un valore in binario che è acceso o spento per gli schermi monocromatici mentre per quelli policromatici un valore associato ad un indice della palette di colori per i vecchi sistemi a **256 colori** mentre adesso gli si associa direttamente il colore con la tecnica del **RGB**.

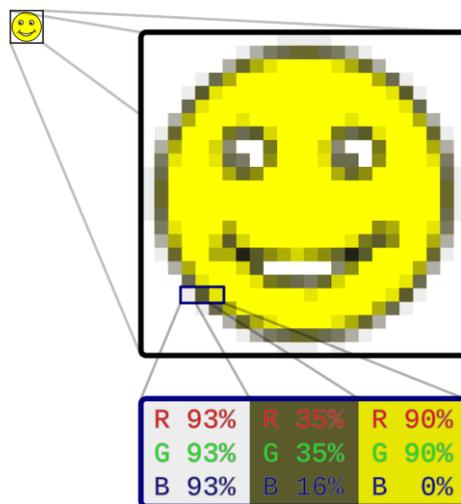


Figura 3: Rasterizzazione

Il principale inconveniente della visualizzazione dell'immagine raster semplice era l'enorme esborso in termini di memoria. Per ovviare a questo problema si è pensato ad un altro escamotage. Inizialmente, nei primi **text mode**, sullo schermo si creavano delle "caselle" di dimensioni fisse logicamente separate tra di loro in cui al loro interno si poteva scrivere un solo carattere, in questo modo non si andavano a salvare tutti i pixel che componevano il carattere, bensì si teneva in considerazione soltanto il riferimento al carattere disegnato.

L ^C	0 0 0 0 0 0 0 0	00000000	11...
	0 0 0 0 0 0 0 0	11111111	00...
	0 0 0 0 1 1 1 1	00001111	00...
	0 0 1 1 0 0 1 1	00110011	00...
	0 1 0 1 0 1 0 1	01010101	01...
	00000		
00001			
00010			
00011			
00100			
00101			
00110			
00111			
01000		
01001		
...		

Figura 3: Esempio di Text Mode 8x8.

Questo metodo viene esattamente implementato nell'elaborazione delle immagini per i videogiochi: Infatti si creano delle **"tile map"** tutte della stessa grandezza, solitamente di 8x8 o 16x16 pixel, dove ogni piastrella era



Figura 3: Esempio di tile map 8x8 utilizzata in Super Mario Bros del 1985. d'immagini, si poteva utilizzare sempre la stessa tile map memorizzata nella ROM.

composta da una porzione d'immagine memorizzata nella ROM del gioco e/o da una combinazione di colori. Le tile map potevano anche essere riciclate ad esempio per creare degli sfondi, infatti se uno sfondo era "statico", ovvero che non presentava particolari cambiamenti di colori o

Funzionamento grafica 2D

Dopo una descrizione generale su come funziona la grafica è tempo di scendere nei particolari di quella 2D. La grafica 2D, come suggerisce la parola stessa (2D: 2 Dimensioni), si basa sulla visualizzazione di due dimensioni, ovvero l'altezza e la larghezza, in termini più tecnici l'ascissa e l'ordinata del piano di visualizzazione.

I giochi a due dimensioni vengono creati su due categorie diverse di prospettive:

- **Prospettiva dall'alto:** Questo tipo di visualizzazione, utilizzata in larga scala nei primi videogiochi, si riferisce ad un'angolazione della telecamera che visualizza il giocatore e l'area circostante dall'alto, come se il mondo lo vedessimo dagli occhi di un uccello in volo.



- **Prospettiva a scorrimento laterale:**

La trattazione di questo tipo di prospettiva ci porterà via un po' di tempo, quindi tenetevi forti!

Questa tecnica di visualizzazione è una tecnica che **punta la telecamera**, e quindi il punto di vista dell'utente, **di lato** e di conseguenza i personaggi, solitamente, possono spostarsi a destra o a sinistra, saltare in alto o in basso e la telecamera si sposta continuamente per cercare di mantenere il personaggio al centro dello schermo ma non solo;

Infatti ci sono un'infinità di varianti per quanto riguarda i movimenti dei personaggi: Alcuni giochi preferiscono spostare lo schermo allo spostamento dei personaggi in modo che davanti adesso risulta più spazio sottraendolo a quello già

Gruppo: Macrì Gabriele, Morchio Massimo.

30/11/2020

Classe VBinf

percorso, in altri ancora lo schermo scorre in avanti da solo a una velocità costante e il giocatore deve tenere il passo con lo schermo, cercando di evitare gli ostacoli e raccogliere le cose prima che passino fuori dallo stesso... Insomma c'è l'imbarazzo della scelta!



Figura x: Esempio di gioco 2D, Limbo.

I primi giochi utilizzavano le **tilemap** per lo scorrimento. Difatti esse erano chiuse ai lati e quindi durante lo **scrolling** ciò che usciva da un lato dello schermo rientrava in seguito da quello opposto oppure veniva aggiornato dalla tilemap della schermata seguente, precaricata in memoria.

Inizialmente lo scorrimento si **muoveva a blocchi** utilizzando due registri di memoria e questo era un grande difetto del mondo 2D, in quanti i giochi perdevano molti punti sulla grafica. La sfida di tutti i programmatori era quella di risolvere questa problematica nota prevalentemente su PC, in quanto su console era già risolta. Un giorno però, una coppia di programmatori incalliti, composta da **Jhon Romero** e **John**

Gruppo: Macrì Gabriele, Morchio Massimo.

30/11/2020

Classe VBinf

Carmack, in particolare quest'ultimo, riuscì a risolvere il problema in una sola notte utilizzando la tecnica **Adaptive tile refresh**.

Prima di descrivere questa tecnica però mi sento in dovere di dare una definizione importante ovvero che cosa sono gli **Sprites**. Il principale problema delle *tilemap* era quello che a cavallo delle stesse non si poteva aggiungere altre immagini, in questo modo c'era solamente un'immagine statica e questo comportava a **sacrificare l'animazione!** Per ovviare a questo inconveniente si sono create appunto le "sprites": siccome gli oggetti di gioco sono completamente indipendenti dallo sfondo s'è pensato di renderli entità a parte che venivano sempre rappresentati con una combinazione di colori ed in questo modo **potevano spostarsi liberamente** attraverso l'indicazione di coordinate cartesiane. Era compito dell'hardware grafico "fondere" sulla stessa schermata gli oggetti da visualizzare e lo sfondo. Gli sprites erano proprio queste entità.

Una volta separate entità e sfondo, per **donare al personaggio un'illusione di movimento**, si assegnavano a quest'ultimo sprites diverse che si alternavano a schermo, operazione simile

Gruppo: Macrì Gabriele, Morchio Massimo.

30/11/2020

Classe VBinf

che viene anche utilizzata per la realizzazione di cartoni animati. Esempio di sprites, Dragon Ball:

Adaptive tile refresh

John Carmack per la realizzazione di questa tecnica ha



utilizzato diverse funzionalità dell'hardware EGA. Il nome di questa funzionalità prende il nome proprio dal suo funzionamento, infatti si tracciavano tutti gli elementi grafici spostati al fine di **ridurre al minimo la quantità di tempo** che si dovrebbe impiegare nel ridisegnare un intero fotogramma. Questo scopo viene raggiunto grazie all'hardware "EGA", cosa che utilizzando quello "CGA" non era possibile effettuare, siccome quest'ultimo utilizzava il software per eseguire lo scorrimento, ridisegnando lo schermo per ogni fotogramma, un esborso enorme in termini di costo computazionale della CPU.

Funziona sostanzialmente contrassegnando le **tessere bitmap** e ridisegnando solo la grafica sullo schermo che si **aggiorna** effettivamente.

Gruppo: Macri Gabriele, Morchio Massimo.

30/11/2020

Classe VBinf



L'hardware EGA

L'hardware utilizzato da Carmack consentiva l'aggiornamento adattivo dei riquadri per diversi motivi:

- Il buffer dello schermo può essere leggermente più ampio dello schermo e arbitrariamente alto, ovviamente con limitazioni imposte dalla memoria video .
- La posizione all'interno di questo buffer da cui viene disegnato lo schermo può essere spostata con incrementi di 1 pixel, sia orizzontalmente che verticalmente.
- Ha una memoria video sufficiente per memorizzare due di questi buffer dello schermo (descritto nel punto 1) e ha ancora altro spazio che può essere occupato dagli sprite.

Carmack ha utilizzato queste capacità **per creare un buffer di 64 pixel più largo e più alto dello schermo**, lasciando spazio per due righe e colonne aggiuntive di tessere nel buffer fuori dal

Gruppo: Macrì Gabriele, Morchio Massimo.

30/11/2020

Classe VBinf

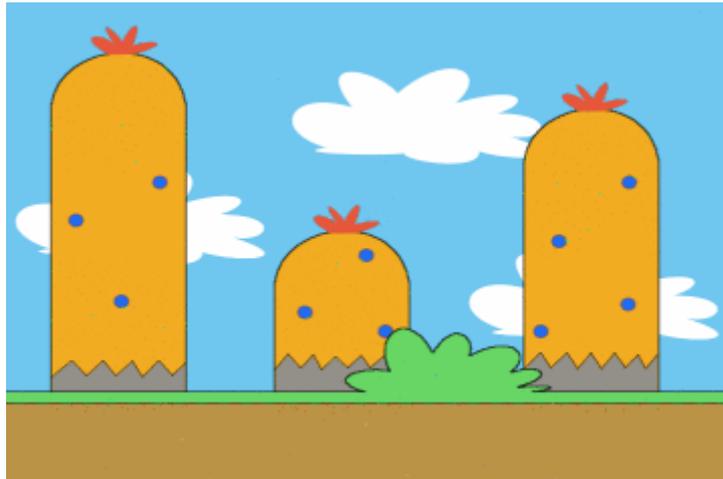
bordo dello schermo. Ha usato le capacità di offset della scheda per far scorrere lo schermo attraverso il buffer per uno scorrimento fluido, che rivela parzialmente le tessere extra.

Poiché solo i bordi dello schermo vengono ridisegnati in qualsiasi momento, gli sprite nel livello **non vengono ridisegnati** finché non raggiungono il bordo dello schermo. Per contrastare ciò, il codice calcola quali tessere uno sprite in movimento aveva precedentemente coperto e le **ridisegna per cancellare la vecchia immagine dello sprite**, quindi disegna la nuova immagine dello sprite nella sua nuova posizione. Ciò consente agli sprite di essere animati indipendentemente dallo scorrimento con uno sforzo computazionale minimo.

Poiché spostare uno sprite in questo modo implica prima cancellarlo e poi ridisegnarlo, l'immagine dello sprite cancellato può essere visibile brevemente, causando sfarfallio. **La parte finale della tecnica di Carmack** è l'uso delle stesse funzionalità hardware EGA usate per lo scorrimento per implementare anche il doppio buffering : impostare un secondo buffer in cui il codice può disegnare mentre il primo buffer viene mostrato sullo schermo, che viene quindi disattivato durante l'aggiornamento dello schermo.

Scorrimento parallasse

Una piccola variante dello scorrimento fluido ideato da Carmack è lo scorrimento parallasse. Questa tecnica consente di far muovere oltre alla telecamera le immagini di sfondo più lentamente rispetto alle immagini in primo piano, in questo modo si viene a creare **un'illusione di profondità**.



Il confine tra grafica 2D e 3D: La grafica vettoriale

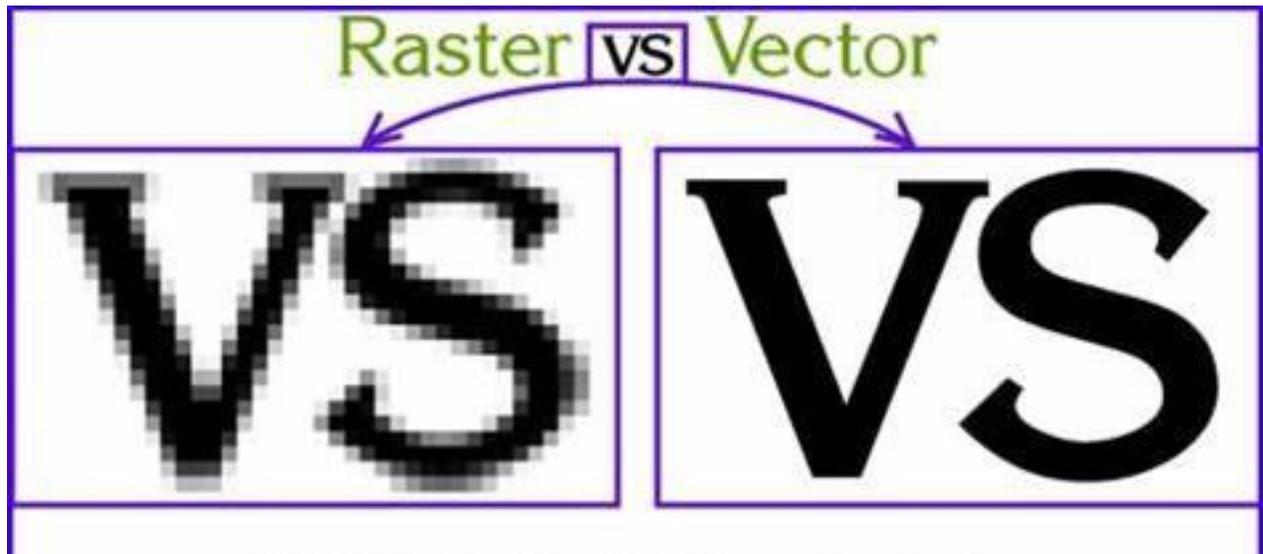
Prima di parlare della grafica 3D nel dettaglio bisogna prima parlare di un punto intermedio tra la grafica 3D e quella 2D: la **grafica vettoriale**.

La grafica vettoriale è una tecnica per la rappresentazione di un'immagine, in cui le sue caratteristiche sono tante primitive geometriche messe insieme per rappresentare un modello e alle quali possono essere associate anche dei colori o sfumature.

Questa tecnica quindi si basa su delle operazioni matematiche su queste primitive che ne descrivono i valori di spessore e forma ad esempio, ma anche su operazioni come **transizione, rotazione o ridimensionamenti**.

Per rappresentare un oggetto, e quindi inserire una primitiva geometrica, vengono stabilite delle "coordinate" o dei punti che dettano il percorso della stessa.

Ciò rende il tutto completamente diverso dalla grafica raster che prevedeva la rappresentazione delle immagini come una griglia di pixel.



(Differenza tra immagine raster e immagine vettoriale)

Come mostrato nell'immagine che mette a confronto un'immagine raster e un'immagine vettoriale, si nota una notevole differenza quando si ingrandisce la figura.

Nell'immagine **raster**, essendo formata da pixel, quando si ingrandisce **perde di qualità** mentre l'immagine vettoriale no essendo composta da punti e funzioni.

I principali vantaggi della grafica vettoriale sono:

- Anche su notevoli dimensioni d'immagine, quest'ultime occupano poco spazio su disco;
- E' la scelta migliore per rappresentare loghi o disegni tecnici;
- Le immagini possono essere ridimensionate senza perdere risoluzione.

Esistono anche degli svantaggi però, ad esempio:

- I formati delle fotografie rimangono in raster;
- Non è adatta alle mescolanze complesse di colori.

Diciamo che la grafica vettoriale è il confine tra le due grafiche (quella 2D e quella 3D) perchè grazie a questa si poteva donare all'immagine un effetto **tridimensionale**. Infatti alcuni

Gruppo: Macrì Gabriele, Morchio Massimo.

30/11/2020

Classe VBinf

sviluppatori di videogiochi preferirono rinunciare alla grafica raster per tornare ad utilizzare il CRT, sostanzialmente utilizzavano la posizione del pannello elettronico come **punto di fuga prospettico** per dare all'immagine appunto un'illusione tridimensionale.

Esempio di grafica vettoriale utilizzata nel gioco **Vertex**.

Funzionamento grafica 3D



La grafica 3D è un insieme di modelli tridimensionali basata sull'elaborazione di algoritmi in modo da riprodurre al meglio una verosimiglianza fotografica nell'immagine finale.

Al contrario della grafica 2D su cui erano basati precedentemente i giochi su console e pc (dove la visuale e lo scorrimento erano solamente in due direzioni), con la grafica 3D si ottiene una visuale in tre dimensioni aggiungendo la profondità avendo una percezione del tutto diversa da quella precedente (due dimensioni).



Il metodo di produzione della grafica 3D si basa su due caratteristiche:

- una descrizione della scena, ovvero ciò che si vuole rappresentare;
- un motore di rendering che si fa carico di tutti gli algoritmi per la creazione di un'immagine 2D dalla scena.

Un motore di rendering è un componente hardware o software che analizza le informazioni codificate in entrata e le rielabora creandone una rappresentazione grafica.

Per la creazione di modelli tridimensionali semplici, come può essere una sfera, si opera su un sistema di riferimento cartesiano tridimensionale: un esempio è la l'equazione $x^2+y^2+z^2=r^2$ che rappresenta una sfera di raggio r .

Per la creazione di modelli 3D più complessi vengono usate due tecniche:

- **geometria solida costruttiva**: vengono eseguite delle operazioni con dei modelli tridimensionali semplici (un

Gruppo: Macrì Gabriele, Morchio Massimo.

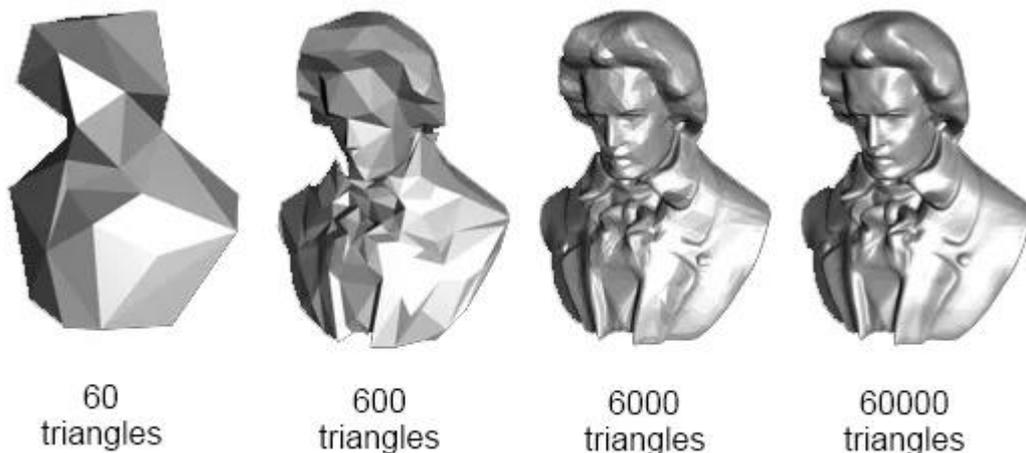
30/11/2020

Classe VBinf

esempio è la classica creazione di un tubo, dove vengono eseguite delle operazioni booleane con dei cilindri semplici e diametri differenti);

(tubo 3D formato con le operazioni tra due cilindri)

- **superfici di suddivisione**: viene introdotta questa tecnica al fine di raffinare una figura grezza che inizialmente è “sfaccettata”. Si può allora definire l’immagine con l’aggiunta di algoritmi rendendo la figura densa di poligoni e ottenendo quindi un’immagine più raffinata.



(Da notare però come il passaggio da 6000 a 60000 triangoli non apporti una notevole migliona all’occhio umano, mentre la differenza la notiamo nell’occupazione della memoria occupata.)

Tuttavia per la creazione di modelli 3D complessi non è sufficiente la geometria solida costruttiva e la superfici di suddivisione. Vengono quindi usate delle “spline” (un insieme di

Gruppo: Macrì Gabriele, Morchio Massimo.

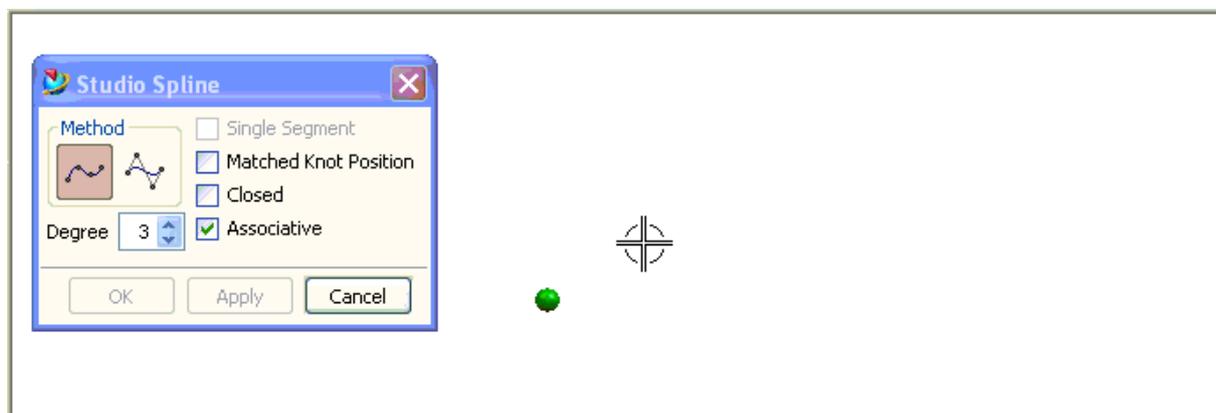
30/11/2020

Classe VBin

spline viene chiamata “patch”) che approssimano le curve continue sulle tre dimensioni.

Le **spline** sono la base della grafica 3D e sono fondamentalmente delle funzioni costituite da polinomi che uniscono un insieme di punti (anche detti nodi) in modo tale da rendere la funzione continua e non “spezzata”.

Le **NURBS** (Non Uniform Rational Basis-Splines) sono una classe di spline più comunemente usate nel mondo tridimensionale e sono quindi un insieme di curve geometriche che vengono usate appunto per la creazione e miglioramento delle superfici e curve nel mondo delle tre dimensioni.



Un po' di storia

Per quanto riguarda la nascita della grafica 3D su console, bisogna concentrarsi sull'azienda Nintendo in Giappone e un giovane ragazzo appena uscito dal liceo Dylan Cuthbert, fondatore di Q-Games 2001.

Questo ragazzo lavorava in un'azienda di giochi 3D a Londra nel 1988: la Argonaut Software.

Gruppo: Macrì Gabriele, Morchio Massimo.

30/11/2020

Classe VBinf

La loro intenzione era inizialmente quella di cercare di capire il funzionamento del gameboy Nintendo, ma nel modo più duro: decodificare il sistema.

Andando avanti col progetto, Cuthbert afferma di aver preso una cartuccia (in parole povere una “scheda” contenente il gioco), averla svitata ed aver collegato i fili ad un circuito che avevano alla **Argonaut**. Riuscì così a visualizzare il codice e poté riscriverlo e programmarlo.

Cuthbert aveva sviluppato la prima demo 3D su gameboy, e ottenne subito l’attenzione di Nintendo. Spedì subito una Rom del gioco e venne assunto dalla casa giapponese per lavorare al progetto 3D su console.



Shigeru Miyamoto, dipendente Nintendo, vuole implementare il 3D nella console Super Nintendo e grazie all’aiuto di Dylan riescono a sviluppare il chip Super FX.



(Chip Super FX integrato nella Super Nintendo)

Il super FX viene utilizzato per il rendering della grafica avanzata nel 2D e 3D nella Super Nintendo.

Agisce sulla grafica disegnando i poligoni su un framebuffer della RAM, ovvero una porzione di memoria dove vengono memorizzati i pixel di un'immagine. Questi dati vengono poi trasferiti alla memoria video principale utilizzando la funzionalità DMA (Direct Memory Access) che consente ai sottosistemi hardware di accedere direttamente alla memoria.

Successivamente, Dylan & co., riescono a sviluppare una demo di volo in 3D, ma a Shigeru Miyamoto non piacciono varie cose: ad esempio il problema della visuale che, se settata troppo in centro durante i movimenti o fissa ad un'ala dell'aereo, non si otteneva un particolare effetto che coinvolgeva a pieno i giocatori.

Gruppo: Macrì Gabriele, Morchio Massimo.

30/11/2020

Classe VBinf

A questi problemi di grafica e visuale ci ha pensato l'addetto alla grafica Giles Goddard, collega di Cuthbert ai tempi dell'Argonaut Software.

Una volta quindi sistemati gli ultimi ritocchi la Nintendo potè far uscire il primo gioco in 3D su console: Star Fox.

Con Star Fox la Nintendo si lancia praticamente nel mondo dei giochi in 3D.

