

# Particelle creative

Seminario sull'uso delle particelle  
a fini artistici e creativi

Relatore: Lion Igaly

**Il seminario è rivolto a tutti coloro che, anche se principianti, vorrebbero utilizzare le particelle nella creazione di proprie opere espressive: da una semplice fontana ad un quadro in movimento.**

**O arricchendole con effetti speciali: nebbia, fuoco, fumo, luccichii e bagliori, sabbia, acqua in movimento, fuochi pirotecnici, vapore e molto altro**

## **Gli argomenti:**

**Descrizione delle particelle e loro impiego**

**Gli emettitori**

**Le particelle e l'Arte**

**Creatività con le particelle**

**Esempi pratici di opere**

**Elementi che influiscono sulla visione delle particelle**

**Influenza delle particelle sull'ambiente circostante**

**Miti sul lag delle particelle**

**Generarle dove non c'è build**

**Come vengono generate dagli script:**

**(funzione `IParticleSystem`)**

**Strumenti che possono essere impiegati per produrre  
script di particelle:**

**gli hud - una panoramica tra gratuiti e non**

## Descrizione delle particelle e loro impiego

**Le particelle sono:**

**-texture che fluttuano nell'aria.**

**-completamente piatte. Esse rivolgono la loro unica faccia all'osservatore, a tutti gli osservatori contemporaneamente, la visione delle particelle è, cioè, soggettiva.**

**-Temporanee, ogni particella non vive più di trenta secondi.**

**- Definite dalla funzione `IIParticleSystem()`**

**Inoltre le particelle:**

**-non sono prim (non potete cliccarle, ne spingerle ne sedervici sopra ne interagire con esse.**

**-Sono generate direttamente nel computer che le osserva e non dal simulatore della regione in cui sono emesse.**

**-Non sono viste allo stesso modo da due computer distinti.**

**- Hanno colore e trasparenza come le facce di un prim.**

**- Possono essere corte e strette da 0,04 m fino a lunghezze di 4 m**

## Inoltre le particelle:

Ci sono 28 parametri che incidono sul loro funzionamento attraverso la funzione

```
ParticleSystem( [ ,,elenco parametri,, ] ) ;
```

I parametri non impostati restano al loro valore di default.

Se lasciamo tutti questi parametri al loro valore di default otteniamo un sistema di particelle spento.

In pratica, per azzerare un emettitore, si usa questa chiamata alla funzione `ParticleSystem`:

```
ParticleSystem( [ ] ) ;
```

Nota bene: rispettate maiuscole e minuscole nelle parole riservate del linguaggio LSL.

Le parentesi quadre racchiudono una lista di parametri vuota.

## **Inoltre le particelle:**

**Il destino delle particelle è predeterminato alla nascita. Dove vanno, quanto a lungo vivono, a cosa assomigliano ecc. sono tutti definiti quando viene chiamata la funzione `IParticleSystem`.**

**Ci sono tre eccezioni a questo:**

**Il movimento del loro prim emettitore**

**Il movimento di un eventuale prim target o avatar target**

**Un cambiamento dei venti nel simulatore (quando si dichiara che il sistema di particelle prodotto è sensibile al vento).**

## **Gli emettitori**

**- Le particelle per essere prodotte hanno bisogno di un emettitore.**

**- Un emettitore è un prim con uno script per particelle all'interno (nel content).**

**Il prim può essere anche phantom e totalmente trasparente o può corrispondere ad un qualsiasi elemento di un gruppo linkato della nostra opera.**

**Qualsiasi prim può emettere particelle, ma solo un effetto alla volta per singolo prim.**

**Influenzano come avverrà l'emissione: il tipo di prim, le torsioni che subisce e le rotazioni. ( e come già accennato, anche il suo eventuale moto).**



## **Gli emettitori**

**-Si può scegliere qualsiasi tipo di prim (cubo, sfera, torus, cono ecc.) ma è bene sapere che il tipo di prim influenza l'emissione delle particelle.**

**-Il cubo ha un comportamento perfetto e predicibile, aderente a quanto si progetta scrivendo lo script per l'emettitore.**

**-La sfera differisce dal cubo solo perché quando la si crea nasce con una rotazione di 90 gradi sull'asse Y.**

**E quindi un getto per fontane, ad esempio, verrebbe emesso coricato se non si elimina questa rotazione di nascita.**

**- L'orientamento è quindi un fattore che incide sull'emissione di particelle**

**- Anche la dimensione del prim può introdurre variazioni al progetto. Di solito si usa un prim grosso per effetti di grandi dimensioni e piccolo per piccole dimensioni.**

## **Le Texture**

- Si può scegliere qualsiasi tipo di texture tenendo presente che:**
- La dimensione della texture comporta un tempo di caricamento iniziale proporzionale.**
- Le texture con canale alpha ( trasparenze) risentono del bug SL delle sovrapposizioni.**
- Non essendo stesa su un prim che ne limita il contorno la texture apparirà nell'aria per come è: rettangolare. A meno di non sfumare nelle trasparenze al contorno.**
- Un singolo script per particelle non può emettere 2 texture diverse contemporaneamente, ma può alternarle con soluzioni timer o rispondenti ad eventi.**

## **Le particelle e l'Arte**

**-Si possono impiegare con profitto le particelle in campo artistico.**

**-Allo stato attuale sono ancora poco utilizzate e si prestano a notevoli sperimentazioni.**

**-E' importante pesarne l'introduzione: una bella scultura può essere rovinata dall'uso smodato di particelle. Una scultura senza pregio particolare può assumere nuova vita con piccoli accorgimenti basati sulle particelle.**

**-Le particelle possono essere attore primario di espressione artistica.**

## **Creatività con le particelle**

- Le combinazioni ottenibili con le particelle sono pressochè infinite.**
- In SL una fontana, un falò, un evento atmosferico hanno maggiore efficacia con le particelle.**
- Sfruttando le peculiarità delle particelle rispetto ai prim si possono creare scenari innovativi e fantasiosi.**

# Fattori esterni che influiscono sulla visione delle particelle

- Comando a menu View-->Nascondi particelle
- Comando a menu Advanced→Rendering→Types→Particles (CTR Alt Shift =)
- Impostazioni grafiche (Particle count /draw distance)
- Dimensioni ridotte del prim emettitore (LOD (*level-of-detail* )
- Orientamento delle particelle e luce del giorno
- Bug del canale alpha
- Inquadratura della propria cam
- Lag

## **Influenza delle particelle sull'ambiente circostante**

### **•Coesistenza di più fonti vicine**

**–Puoi vedere particelle fino al numero massimo impostato sul tuo client. Oltre quel numero tutte le fonti visibili diminuiscono in equilibrio bilanciato.**

### **•Miti sul lag delle particelle**

**–Non si può causare lag ad altri con dei semplici bling sui propri gioielli. Semmai è come è fatto il gioiello che lo può.**

**–Anche se dovessimo produrre particelle in abbondanza nell'ambiente l'unico effetto sarebbe quello di diminuire in proporzione altre fonti di particelle presenti**

**–Per schede video un po' datate la soluzione è sempre quella di ridurre il numero massimo di particelle visualizzate e non di imporre ad altri di non produrne.**

### **•Generarle dove non c'è build**

**–Vesti l'emettitore !**

## Come le particelle vengono generate dagli script

**Hallo, avatar!**

**La funzione IIPasticleSystem**

<pre>default {   state_entry()   {     ISay(0, "Hello, Avatar!");   }   touch_start(integer total_number)   {     ISay(0, "Touched.");   } }</pre>	<pre>default {   state_entry()   {     <b>IIParticleSystem([ ]);</b>   }   touch_start(integer total_number)   {     <b>IIParticleSystem([ ]);</b>   } }</pre>
--	--

**La pagina di riferimento per la funzione:**

**<http://rpgstats.com/wiki/index.php?title=LIParticleSystem>**

**Un esempio di script ben documentato:**

**[http://wiki.secondlife.com/wiki/Example\\_Particle\\_Script](http://wiki.secondlife.com/wiki/Example_Particle_Script)**

Come le particelle vengono generate dagli script

## Premesse all'uso della funzione

- Le parentesi quadre della funzione `ParticleSystem([ ...parametri...])` racchiudono la lista di parametri atta a definire l'emissione di particelle.

- Un prim può avere solo un sistema di particelle attivo per volta. Ogni ridefinizione del sistema cancella gli effetti del precedente.

- Per cancellare una emissione di particelle si deve definire l'insieme vuoto di parametri chiamando la funzione `ParticleSystem([ ])`;

- Le visualizzazioni direzionali delle particelle sono relative all'asse Z del prim ed emanate dal centro del prim.

- Per le impostazioni LOD, se un prim emettitore non è grande abbastanza da poter essere visto (renderizzato) anche lo script di particelle non sarà attivo.

- Il numero di particelle emesso è condizionato dalle impostazioni del client Particle count e Draw distance.

- Per calcolare il numero di particelle emesso:

**Max\_Age \* Count / Rate** che sono parametri della funzione creatrice.



Come le particelle vengono generate dagli script

## Un esempio di parametri della funzione

**// Questo crea una esplosione di particelle rosse ,che segue il vento e mette la texture di default in fullbright (glow)**

```
IIParticleSystem([PSYS_PART_FLAGS, PSYS_PART_WIND_MASK |  
PSYS_PART_EMISSIVE_MASK,  
    PSYS_SRC_PATTERN, PSYS_SRC_PATTERN_EXPLODE,  
    PSYS_PART_START_COLOR, <1,0,0>]);
```

**•Il tempo di vita delle particelle è il default = 10 sec., se volessimo allungarlo a 20 secondi potremmo aggiungere il relativo parametro e la funzione diventa**

```
IIParticleSystem([PSYS_PART_FLAGS, PSYS_PART_WIND_MASK |  
PSYS_PART_EMISSIVE_MASK,  
    PSYS_SRC_PATTERN, PSYS_SRC_PATTERN_EXPLODE,  
    PSYS_PART_START_COLOR, <1,0,0>,  
    PSYS_PART_MAX_AGE, 20]);
```

**Ma c'è una buona notizia!**

**Esistono degli strumenti che compilano per noi i parametri della funzione *IParticleSystem* facendoci scegliere i valori in modo semplice e intuitivo.**

**Questi strumenti sono gli hud generatori di script per particelle.**

**Anzi c'è una seconda buona notizia: qualcuno è addirittura free.**

**Ad esempio questo:**

**<https://www.xstreetsl.com/modules.php?name=Marketplace&file=item&ItemID=674134>**

**Grazie a tutti per l'ascolto**

**Domande?**