Modellazione 3D della Via Lattea in Unity

Tutor accademici:
Prof. Francesco Isgrò
Prof. Giuseppe Longo

Tutor aziendale:
Dott. Massimo Brescia

P. D'Andrea

Idea

Realizzare una simulazione della Via Lattea:

- Navigabile in 3D, nello stile di un videogame.
- Basata su un modello configurabile.
- Da utilizzare come strumento didattico e per visualizzare (e navigare) dati provenienti da fonti esterne.
- Versatile e altamente mantenibile.

L'ambiente scelto a questo scopo, per le sue caratteristiche, è stato quello messo a disposizione da **Unity**.

Unity

"È un ambiente di sviluppo di videogames che fornisce un insieme di strumenti per la creazione interattiva di contenuti 2D e 3D"

http://www.unity.com

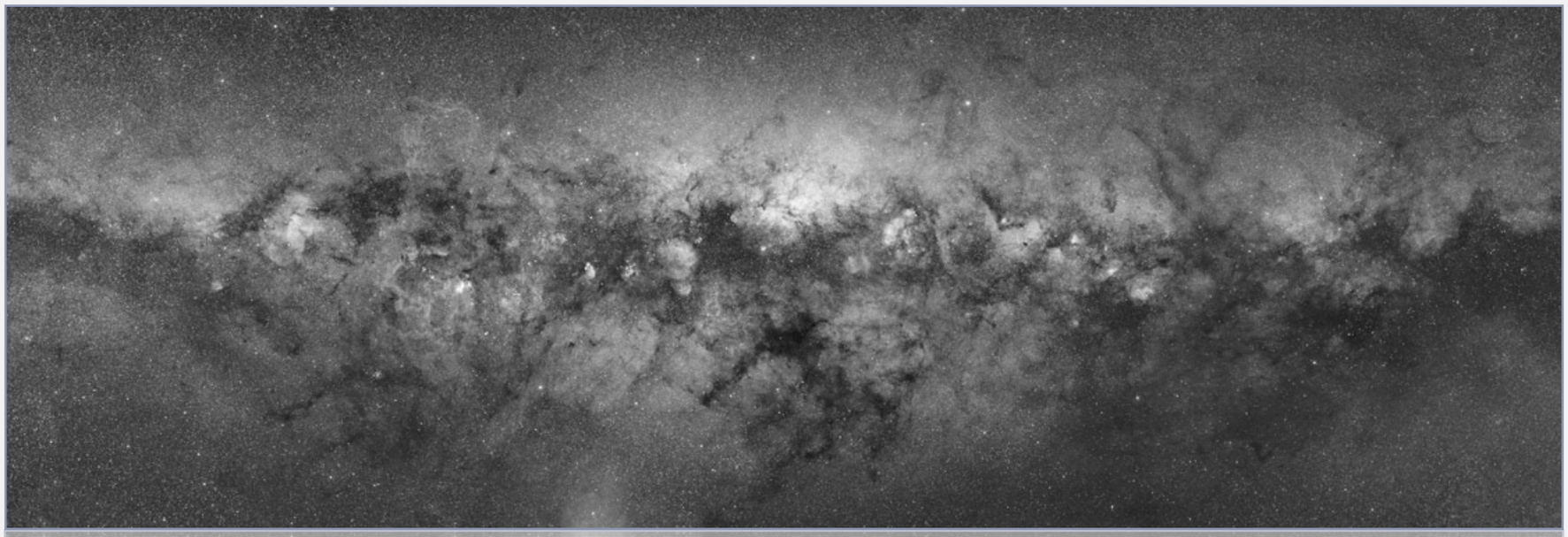
- Prototipazione rapida.
- Live debugging e profiling.
- Sviluppo in linguaggi popolari, come C# e JS.
- Motore di rendering e strumenti multi-piattaforma (Mono).
- Qualità del supporto, visto l'ampio utilizzo nello sviluppo di videogames.

La Via Lattea

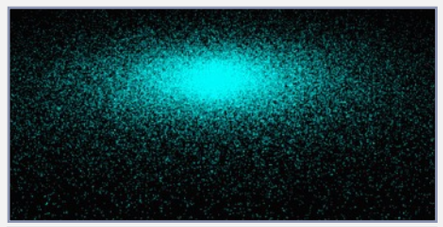
La Via Lattea è la galassia di cui fa parte il Sistema Solare insieme ad almeno altri 200 miliardi di stelle, pianeti, migliaia di ammassi e nebulose.

È una galassia di tipo spirale barrata che appartiene al Gruppo Locale, un piccolo gruppo di 3 grandi ed oltre 30 piccole galassie, nel quale occupa la seconda posizione per dimensioni ma la prima per massa.

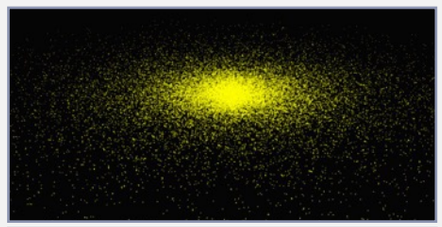
Poiché ci troviamo nelle regioni esterne di questa galassia, solo circa 20 anni luce al di sopra del piano equatoriale ma a circa 27.000 anni luce dal centro galattico, si presenta come una striscia luminosa che attraversa il cielo lungo il piano di simmetria, chiamato anche "equatore galattico".



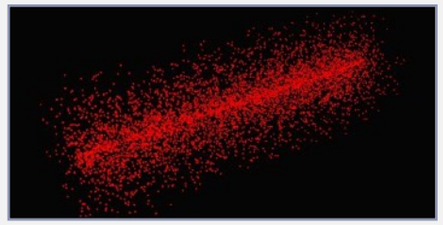
La Simulazione



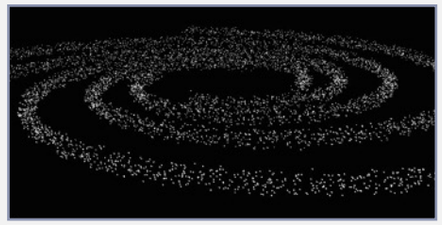
ThickDisk



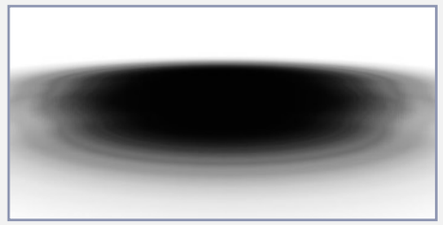
ThinDisk



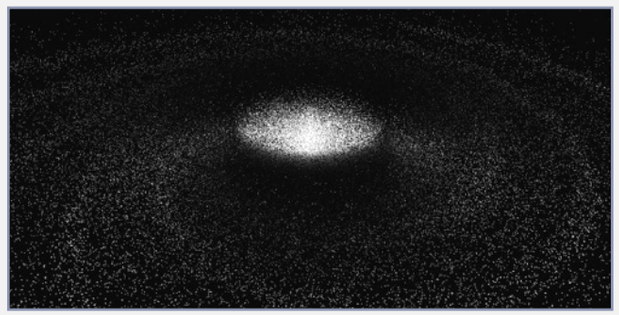
Bar



SpiralArms



Dust



MilkyWay

ProceduralMeshes

Mesh

GameObject
osservatore + Camera

Dati da file (CSVParser)

Modello

Parametri del modello
e della simulazione

UI

Utente

Modello

Disco

Regione principale della Galassia, caratterizzata da un'intensa formazione stellare.
È costituito da 2 componenti: **disco sottile** e **disco spesso**, distinte sia in termini cinematici che di popolazione stellare.

Diametro: tra 40 e 50 kpc

Altezza: 1.5 kpc

Densità di stelle:

$$\rho(R,Z) = \rho_{\oplus} \times e^{\left(-\frac{Z-Z_{\oplus}}{h_{Z,\rho}} - \frac{R-R_{\oplus}}{h_{R,\rho}}\right)}$$

Disco sottile:

$$h_Z = 0.30 \pm 0.10 \text{ kpc}$$
$$h_R = 2.50 \pm 0.30 \text{ kpc}$$

Disco spesso:

$$h_Z = 0.90 \pm 0.08 \text{ kpc}$$
$$h_R = 3.80 \pm 0.20 \text{ kpc}$$

Barra

Zona di dimensioni ridotte e molto popolata, localizzata nel centro della Via Lattea e caratterizzata da una quasi totale assenza di polveri che sono vaporizzate dalla radiazione uv emessa dalle stelle presenti nella barra stessa.

Rapporto assiale: 10 : 4 : 4

Lunghezza: 4.5 kpc

Densità di stelle:

$$\rho_{stars}(d) = \left(1 - \frac{d}{R}\right) a$$

Modello

Bracci di spirale

Utilizzando come traccianti gli oggetti più giovani del disco, è possibile individuare la presenza di bracci che conferiscono alla Via Lattea una particolare struttura a spirale. Lungo la struttura dei bracci è inoltre presente un volume di polveri che contribuisce con le polveri presenti nel disco al fenomeno di **estinzione**.

Curva descritta: $$\theta = \frac{1}{a}\log\left(\frac{r}{b}\right) + \theta_0$$

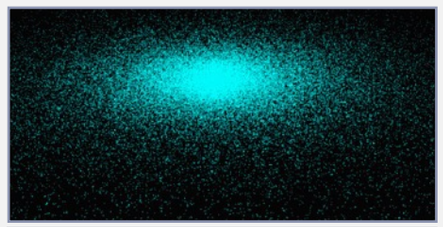
Densità di polveri: $$\rho(R) = 1 - \frac{R}{R_{\max}}$$

Polveri

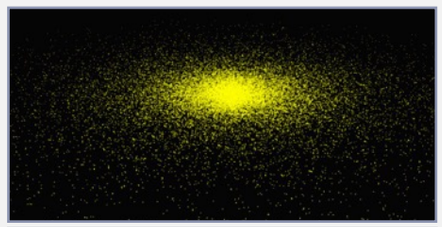
Presenti da una distanza di circa 1.5 kpc dal centro della Galassia, fino ai confini esterni. Determinano il fenomeno di estinzione, ossia l'assorbimento e la dispersione di parte della radiazione elettromagnetica che le attraversa e per questo influiscono in maniera significativa sulla capacità di osservare, nello spettro visibile, i copri celesti che si trovano oltre le immediate vicinanze rispetto alla propria posizione nella Via Lattea.

Densità: $$\rho(R) = e^{-\frac{R}{R_{\min}}}$$

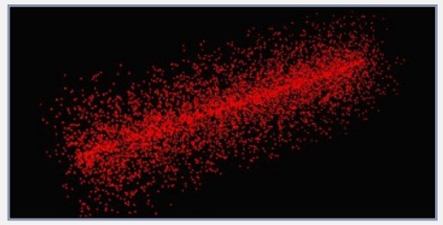
La Simulazione



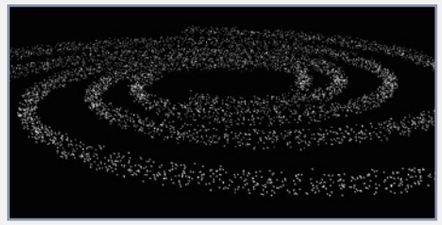
ThickDisk



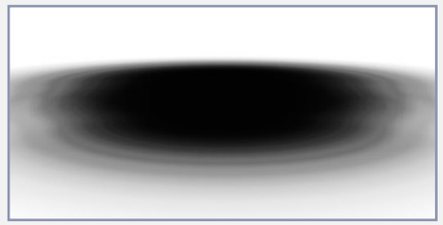
ThinDisk



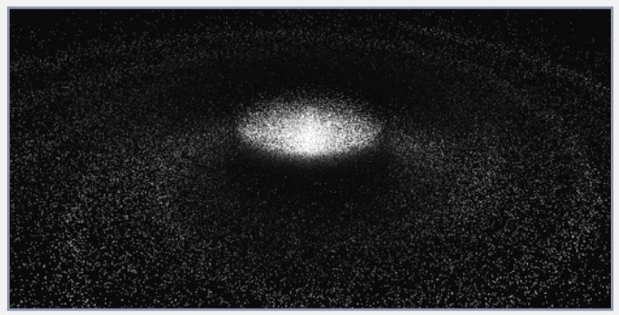
Bar



SpiralArms



Dust



MilkyWay

ProceduralMeshes

Mesh

GameObject osservatore + Camera

Dati da file (CSVParser)

Modello

Parametri del modello e della simulazione

UI

Utente

GameObject

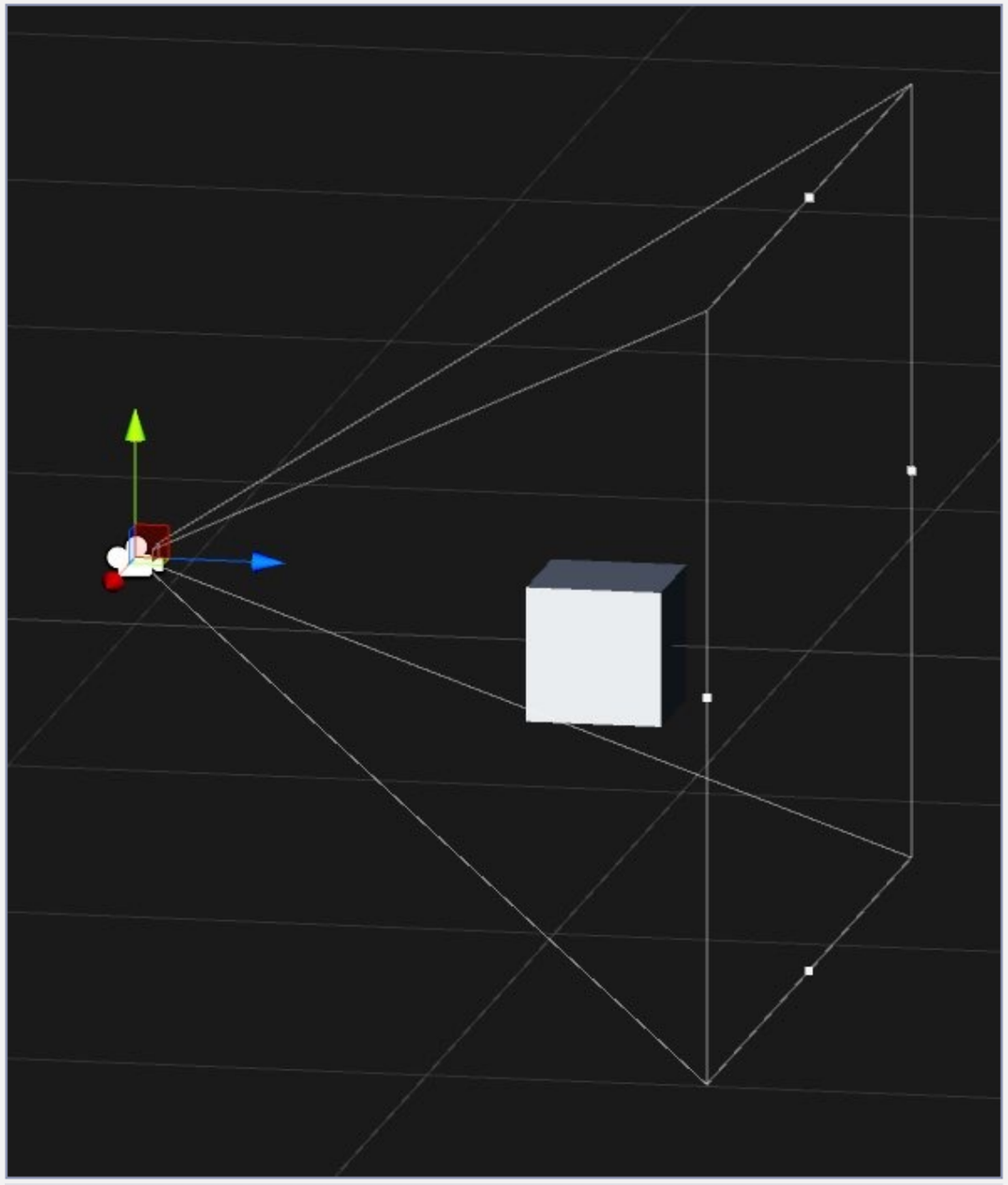
È la **base comune ad ogni elemento facente fisicamente parte di una scena**.
Ogni *GameObject* è **caratterizzato da componenti** che vi possono essere associati ad ampliarne le proprietà e determinarne il comportamento, come:

- Transform
- Camera
- MeshFilter e MeshRenderer
- Script

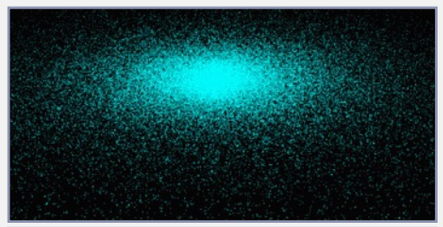
Camera

Il componente ***Camera*** può essere immaginato come un'inquadratura sulla scena. Si occupa di **definire cosa** e con quali modalità **viene visualizzato in fase di rendering**.

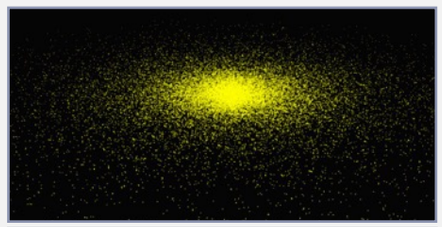
La visuale in prima persona con la quale si esplora la simulazione è dovuta all'associazione del componente Camera principale al GameObject che svolge il compito di osservatore e del quale l'utente controlla i movimenti.



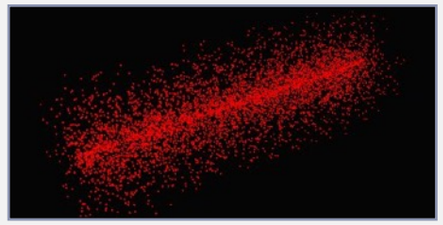
La Simulazione



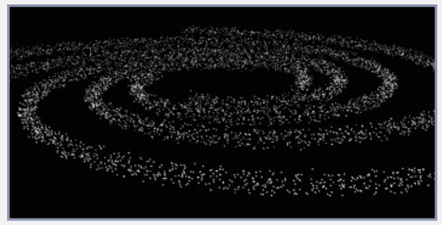
ThickDisk



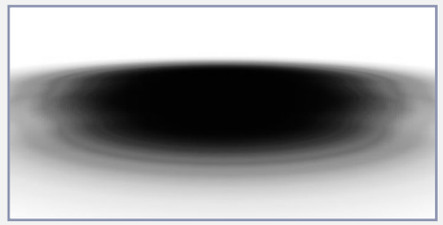
ThinDisk



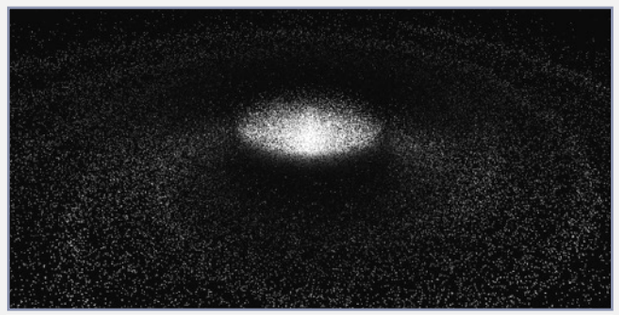
Bar



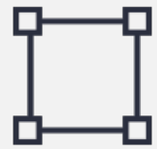
SpiralArms



Dust



MilkyWay



ProceduralMeshes

Mesh

GameObject
osservatore + Camera

Dati da file (CSVParser)

Modello

Parametri del modello
e della simulazione

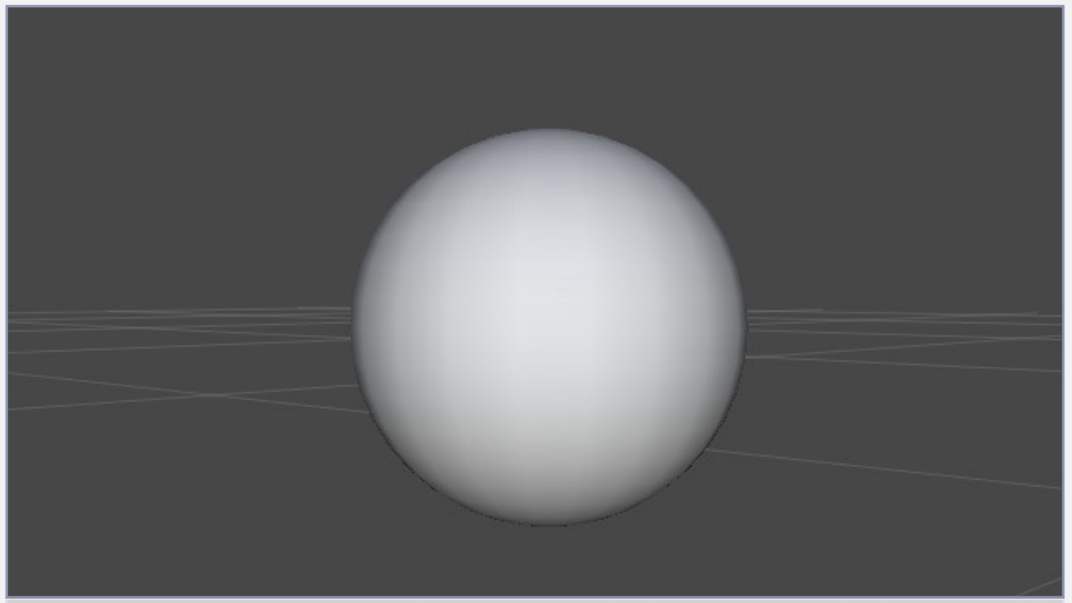
UI

Utente

Mesh e Shader

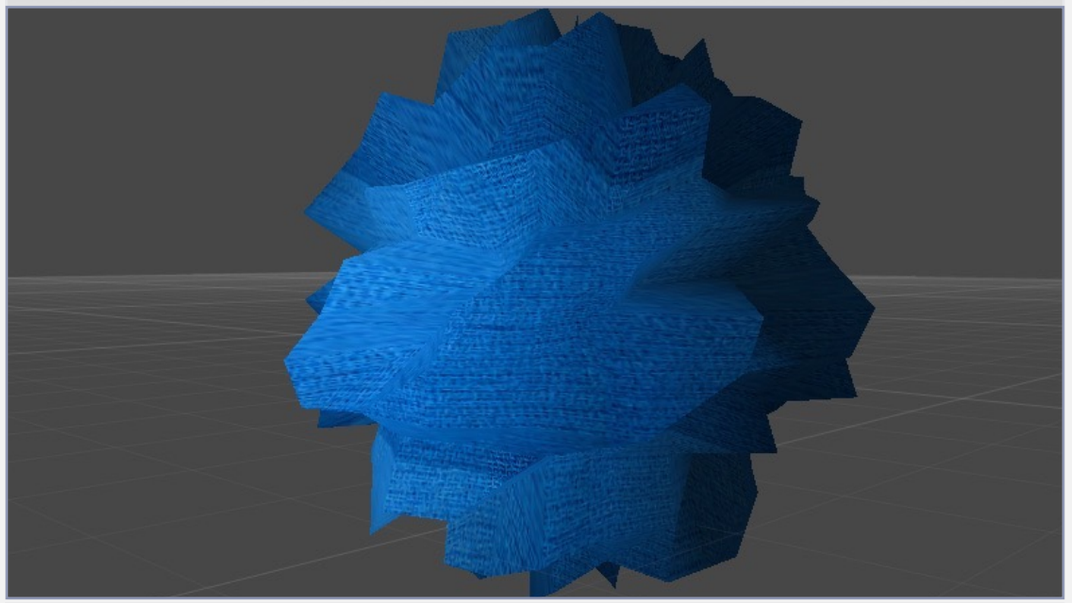
Mesh

Collezione di vertici e triangoli che definiscono la forma superficiale di un poliedro in un ambiente 3D.

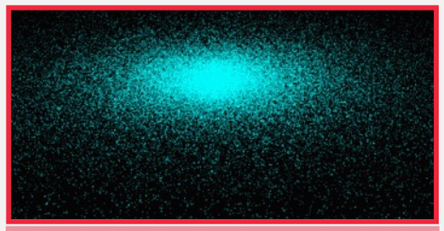


Shader

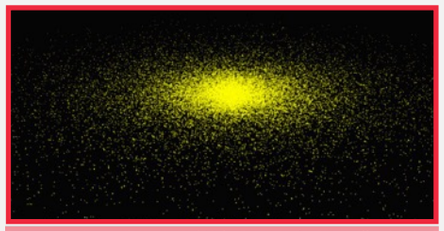
Script eseguiti dalla GPU e associati alle mesh che consentono di determinare l'aspetto finale di un oggetto variandone la forma, la posizione, l'illuminazione, la texture o i dettagli dei singoli pixels rispetto a quelli definiti inizialmente.



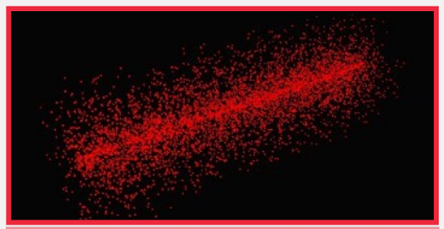
La Simulazione



ThickDisk



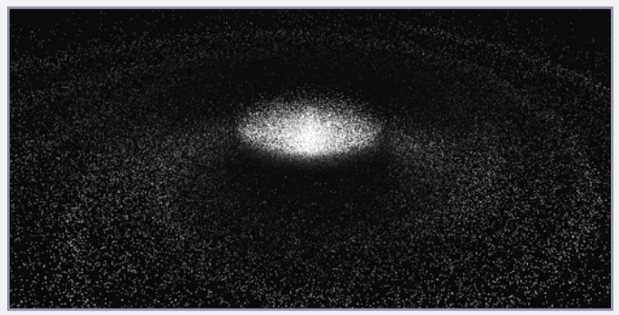
ThinDisk



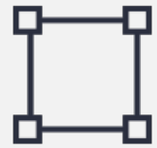
Bar

SpiralArms

Dust



MilkyWay



ProceduralMeshes

Mesh

GameObject
osservatore + Camera

Dati da file (CSVParser)

Modello

Parametri del modello
e della simulazione

UI

Utente

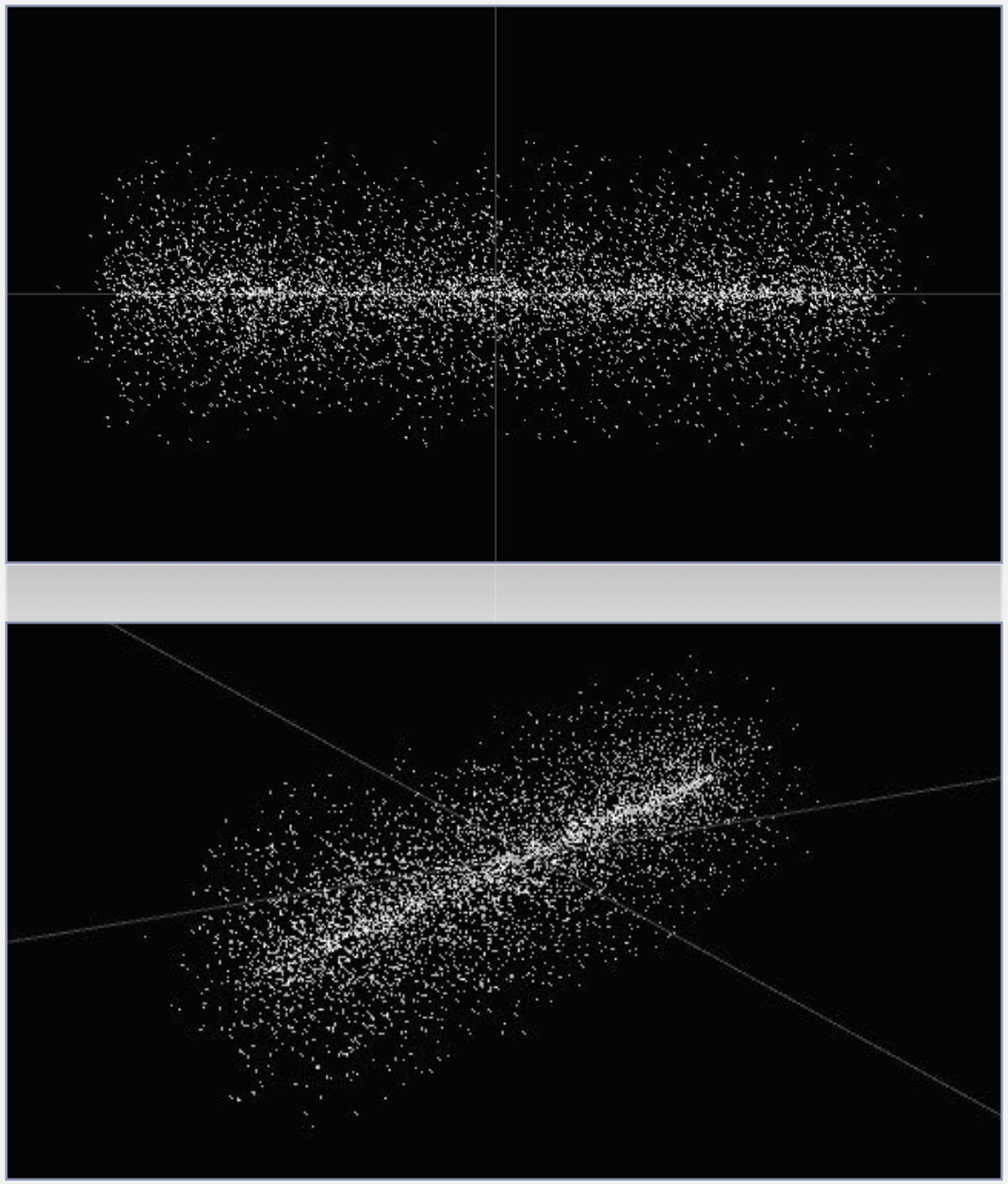
Barra

Essendo approssimabile con un cilindro, è simulata distribuendo dei punti (in numero variabile) nel volume di cilindri di raggio progressivamente crescente.
Tali punti sono disposti casualmente e seguendo una distribuzione linearmente decrescente sulla distanza dall'asse di simmetria dei cilindri:

$$\rho_{stars}(d) = \left(1 - \frac{d}{R}\right) a$$

```
for (int r = 1; r <= segments; r++) {
    //  sezione di raggio da considerare in questo loop
    float segment = (float)r / segments;
    float currentRadius = radius * segment;
    //  numero di stelle da simulare
    int stars = Mathf.roundToInt(
        (1 - segment) * totalStars);

    for (int s = 1; s <= stars; s++) {
        //  calcolo di un punto casuale
        Vector3 position = Rnd.pointOnCylinder(
            height: barLength,
            radius: currentRadius);
    }
}
```

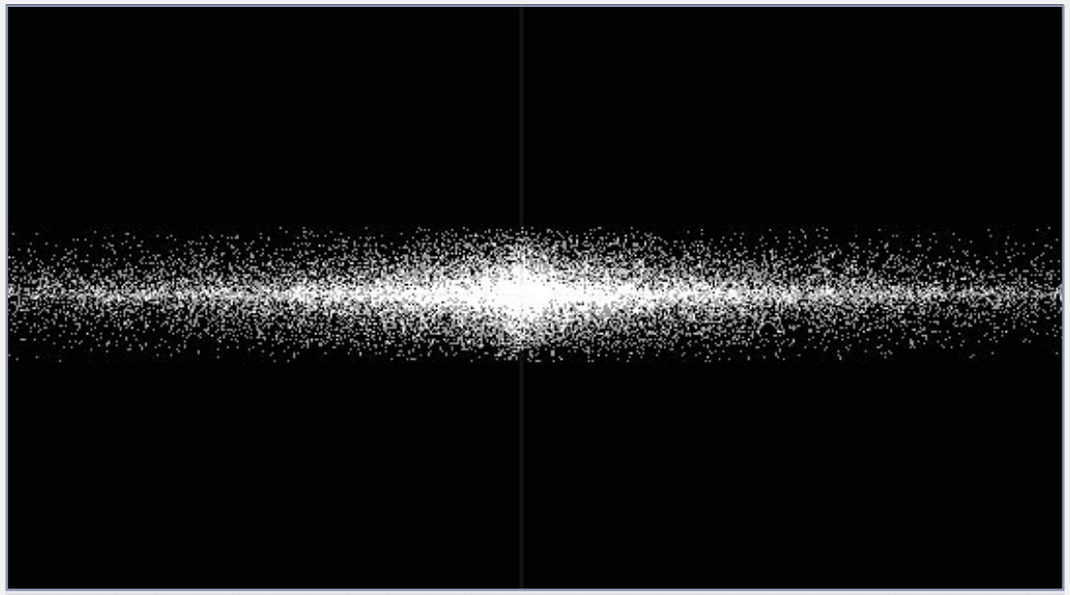


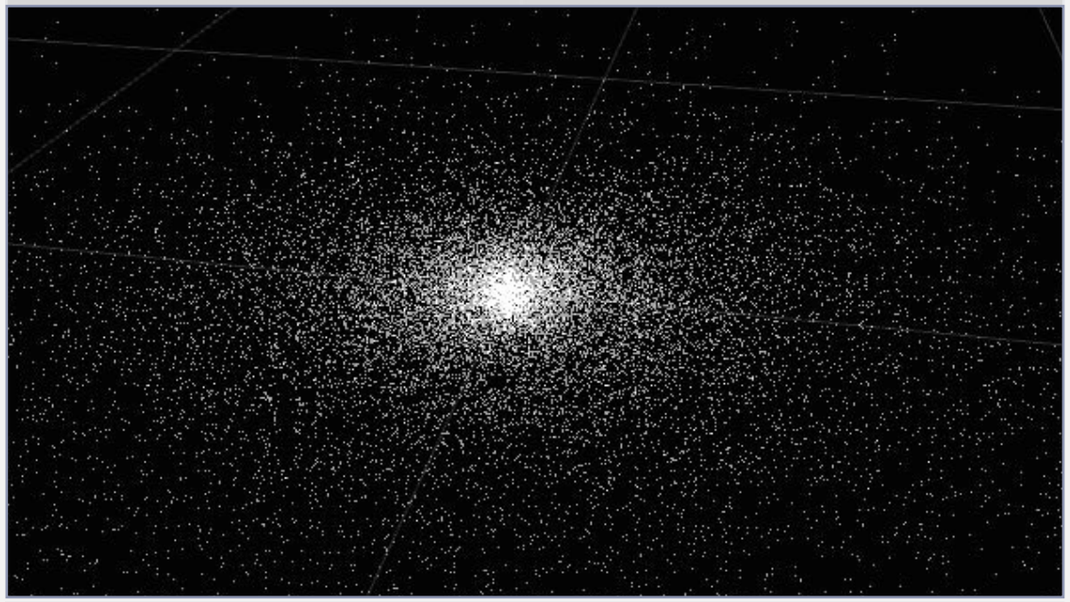
Dischi

Da un algoritmo simile a quello utilizzato per la barra, sono simulati distribuendo casualmente dei punti nel volume di dischi di raggio e altezza progressivamente crescenti.
Tali punti sono disposti casualmente e seguendo una distribuzione esponenzialmente decrescente dal centro dei dischi verso i confini esterni:

$$\rho(R,Z) = \rho_{\oplus} \times e^{\left(-\frac{Z-Z_{\oplus}}{h_{Z,\rho}} - \frac{R-R_{\oplus}}{h_{R,\rho}}\right)}$$

```
// densità di stelle massima
float maxDensity = starsDensity(0, 0);
// iterazione sul numero di sezioni del raggio
for (int Ir = 1; Ir <= segmentsR; Ir++) {
    // raggio interno della sezione considerata
    float Ri = R * (Ir - 1) / segmentsR;
    // raggio esterno della sezione considerata
    float Re = R * I / segmentsR;
    // raggio medio tra esterno e interno
    float Rmid = Re - (Re - Ri) / 2f;
    // iterazione sul numero di sezioni dell'altezza
    for (int Iz = 1; Iz <= segmentsZ; Iz++) {
        // altezza da considerare in questo loop
        float H = Z * Iz / segmentsZ;
        // numero di punti da calcolare
        int P = Mathf.roundToInt(starsDensity(Rmid, H) / maxDensity);
        for (int p = 0; p < P; p++) {
            // calcolo di un punto casuale nel disco
            Vector3 position = Rnd.pointInDisk(
                innerRadius: Ri, outerRadius: Re, height: H);
        }
    }
}
```





Bracci di spirale

Spirali logaritmiche che giacciono nel piano equatoriale della Galassia e che si diramano dalle estremità della barra.

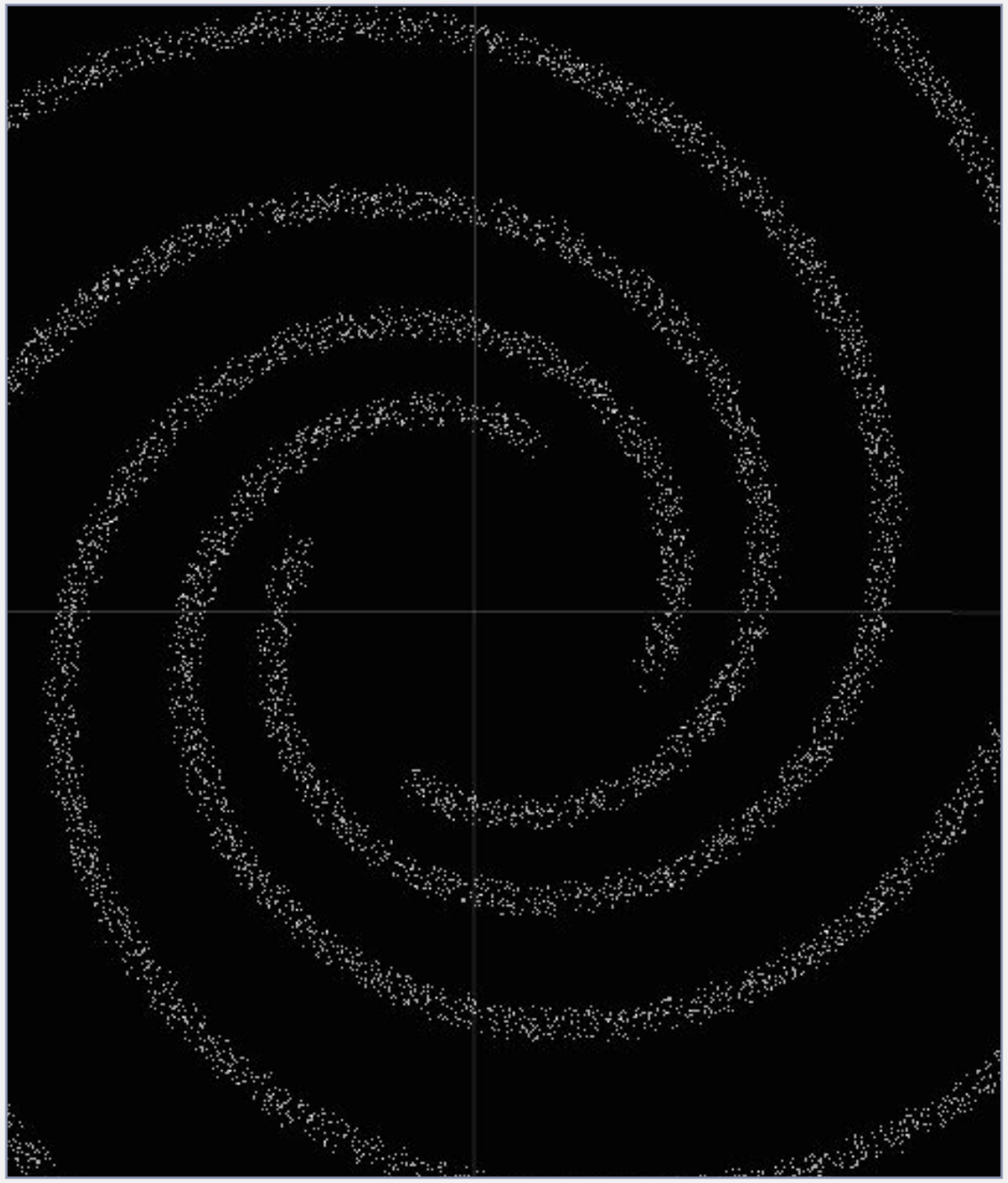
Struttura a spirale

Ognuno dei quattro bracci descrive una curva di equazione:

$$\theta = k \log\left(\frac{r}{b}\right) + \theta_0$$

```
//  costanti definite dal modello
float[]  theta0 = new float[] { -20f, 70f, 160f, 250f };
float          k = 1f / Mathf.Tan(0.2234);
float minRadius = 2.1f;

for (int arm = 0; arm < 4; arm++) {
    for (float r = Rmin; r < Rmax; r += Rstep) {
        //  calcolo coordinata θ (radiale) del punto guida
        float theta0Curr = Mathf.Deg2Rad * theta0[arm];
        float theta = k * Mathf.Log(r / minRadius) +
            theta0Curr;
        //  conversione in un sistema di riferimento 3D
        Vector3 point = new Vector3(
            r * Mathf.Cos(theta), 0, r * Mathf.Sin(theta));
    }
}
```



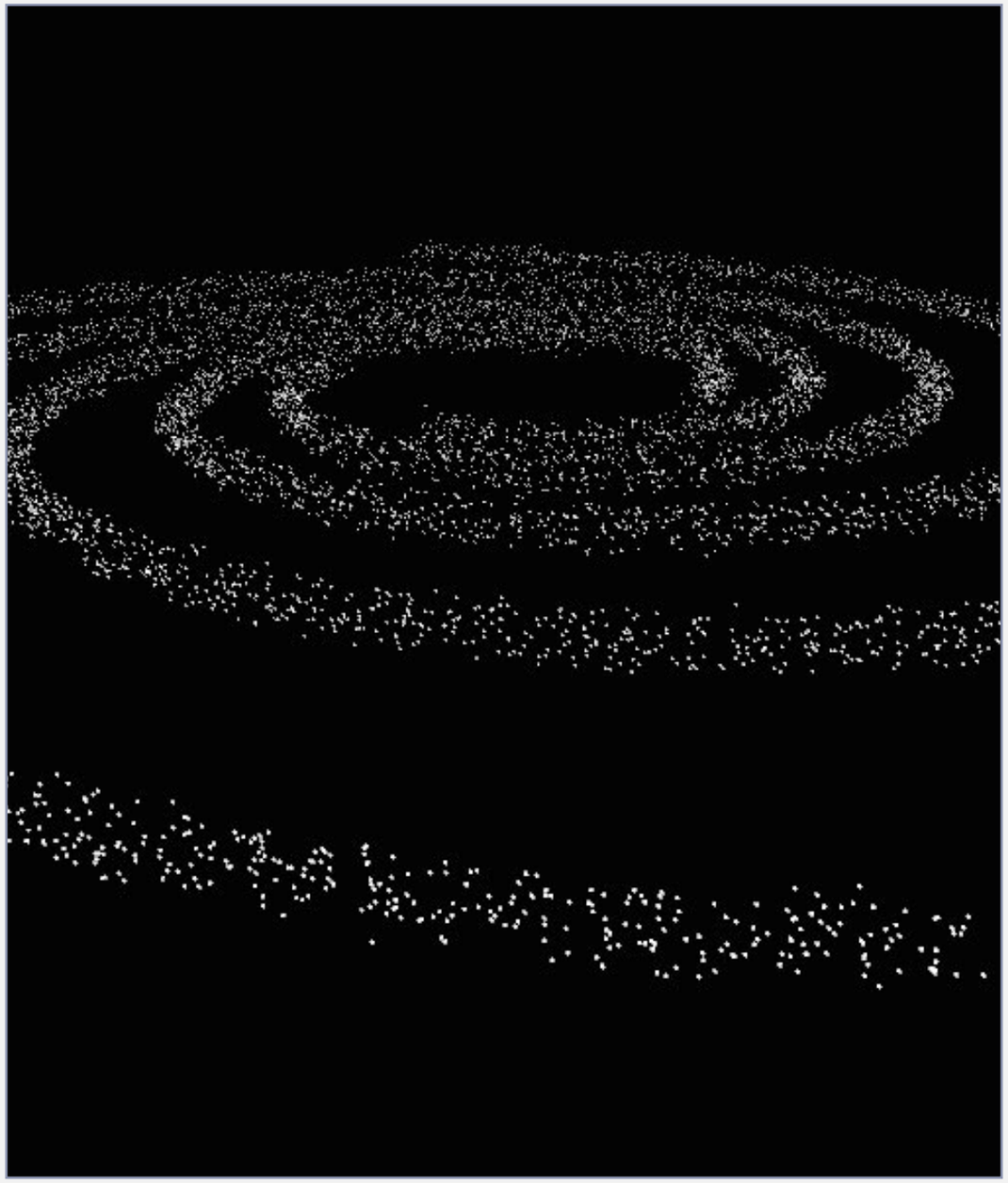
Bracci di spirale

Polveri

I punti calcolati dall'algoritmo sono utilizzati come ***guida*** per la costruzione di un certo numero di mesh tubolari di raggio crescente.
A queste mesh sono assegnati valori costante ti trasparenza che, sommandosi, simulano l'effetto di estinzione causato dalle polveri.

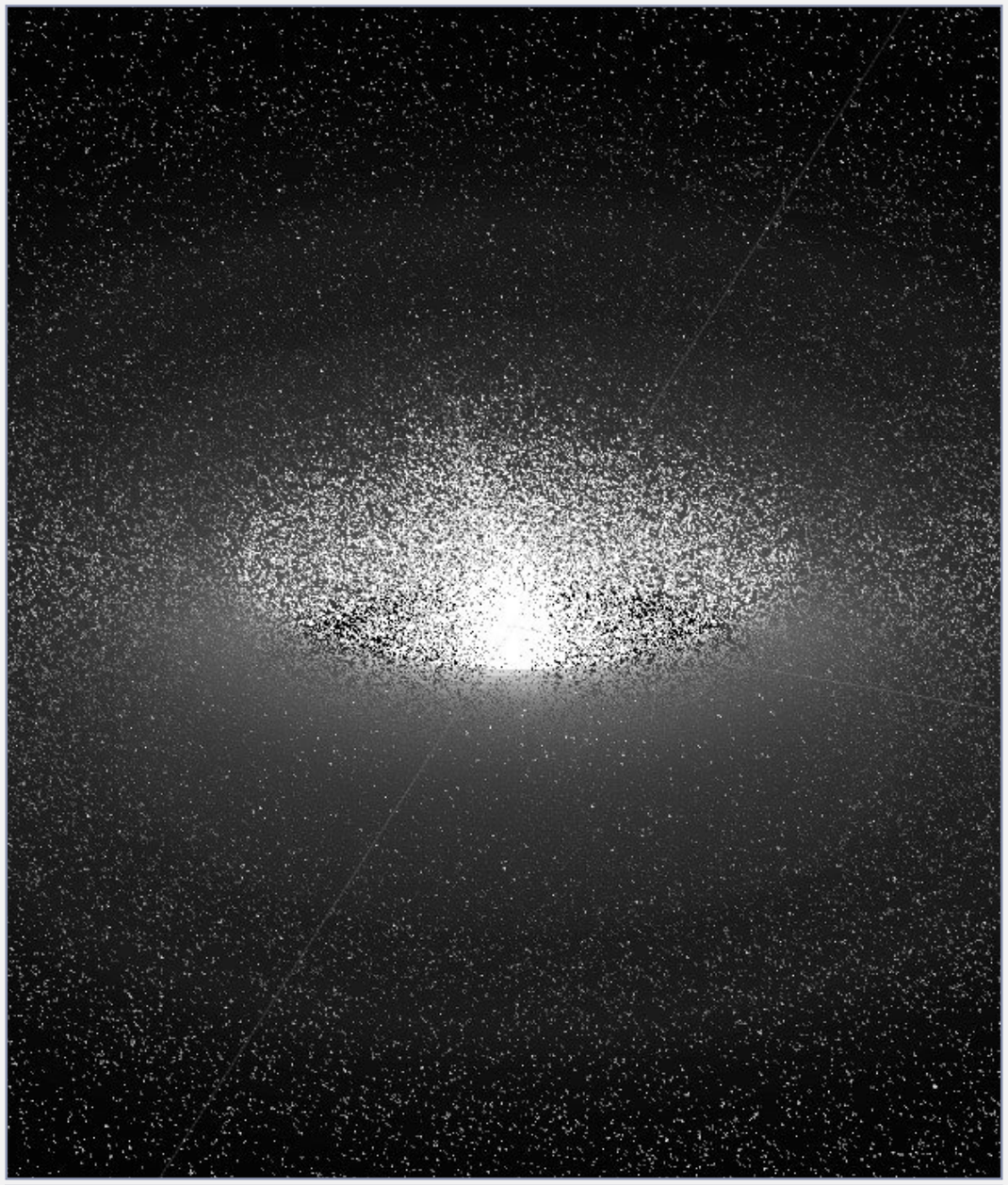
Stelle

Le stelle sono distribuite nei pressi degli stessi punti guida.
Ognuno dei punti è considerato il centro di una sfera nel volume della quale, in maniera casuale, è posizionata una stella.



Polveri

Le polveri presenti nei dischi sono caratterizzate da una densità esponenziale massima a circa 1.5 kpc dal centro e decrescente verso l'esterno fino ad essere nulla a circa 10 kpc.
Il contributo che queste forniscono all'estinzione è reso come livello di trasparenza assegnato a un insieme di mesh disco, di dimensioni decrescenti verso il centro della Galassia, unite con lo scopo di simulare un volume.



```
// distanza dal vertice della mesh
// al centro della galassia
float dist = Distance(
    currentVertexPos, float4(0, 0, 0, 0));

// se fuori dal range delle polveri, la mesh
// assume un colore totalmente trasparente
if (dist < minRadius || dist > maxRadius)
    return float4(0);

// la trasparenza decresce esponenzialmente
return float4(0, 0, 0, exp(-dist / minRadius));
```

Ottimizzazione

Per visualizzare ogni GameObject, il game engine ha necessità di inviare alle API grafiche una **richiesta di aggiornamento dei contenuti visualizzati** (*redraw call*).
Ognuna di queste richiesta è dispendiosa da un punto di vista computazionale e, al crescere del numero di redraw calls, viene a crearsi un **collo di bottiglia nella comunicazione tra CPU e GPU**.

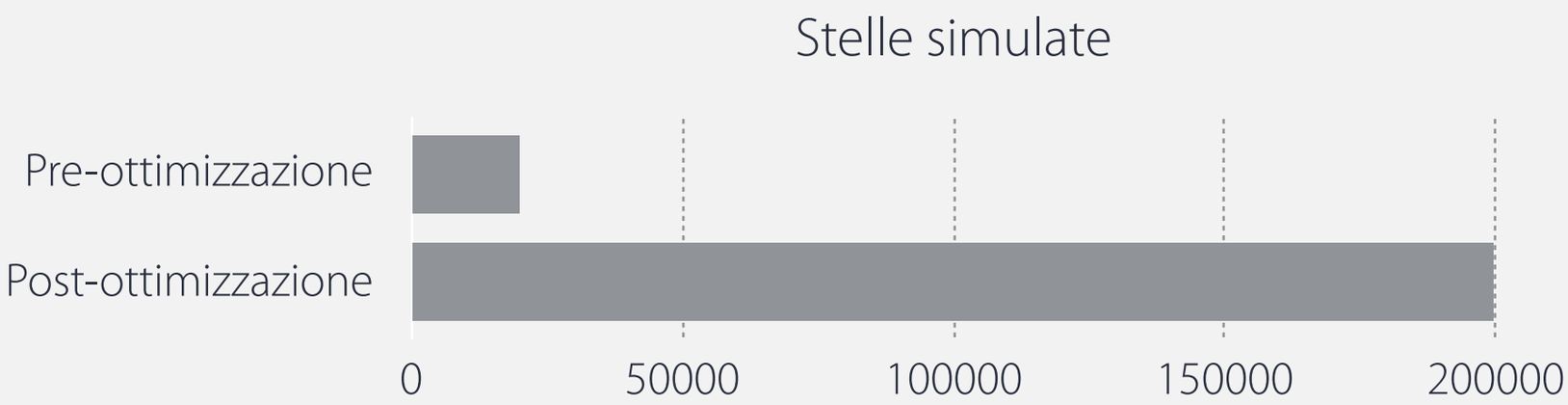
In un contesto nel quale è importante riuscire a gestire un elevato numero di mesh (e conseguentemente effettuare un gran numero di redraw calls), come la simulazione, **è indispensabile ottimizzare il processo di rendering**.

Ottimizzazione

Per ottenere un risultato soddisfacente sono state attuate due tecniche di ottimizzazione:

A. **Costruzione procedurale delle mesh** che consente di poter regolare il numero di vertici e triangoli da adoperare.

B. **Riduzione del numero di richieste di aggiornamento dei contenuti (redraw calls)**, ottenuta grazie alla fusione di più mesh, unendone vertici e triangoli in *“mesh gruppo”*.



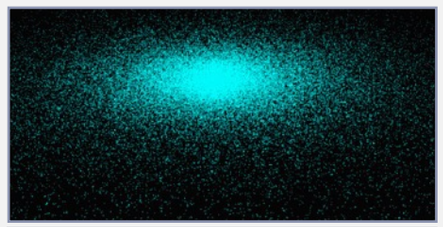
Possibili sviluppi futuri

- **Maggiore realismo visivo delle componenti** della Galassia utilizzando materiali e shader per riprodurne le caratteristiche fisiche.

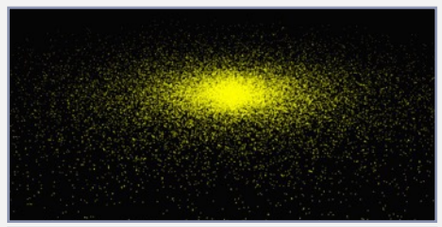
- **Aggiunta di ulteriori dettagli al modello della Galassia** attualmente in uso.

- **Upgrade dalla versione free alla versione pro di Unity** per poter sfruttare le funzionalità avanzate messe a disposizione (nebbia, occlusion culling).

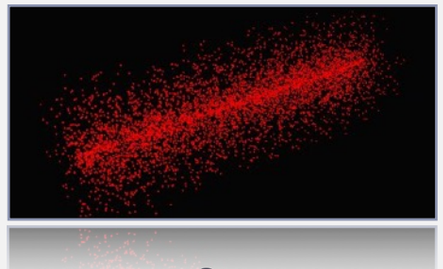
La Simulazione



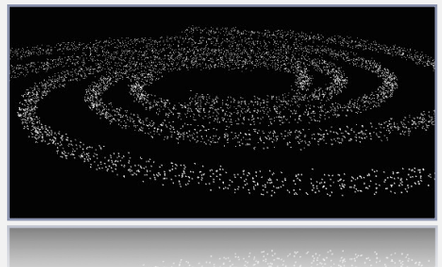
ThickDisk



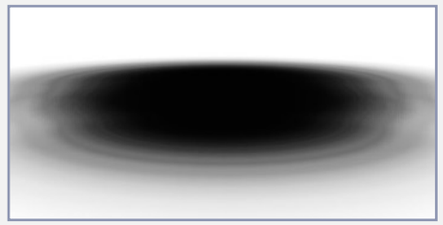
ThinDisk



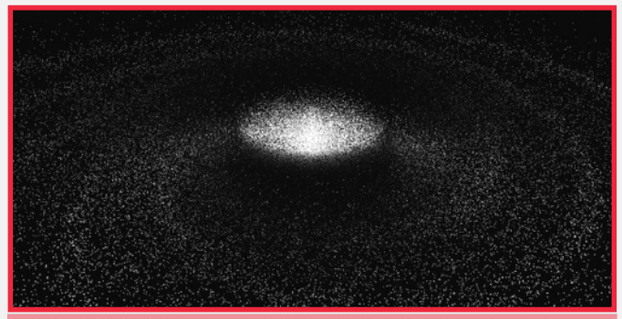
Bar



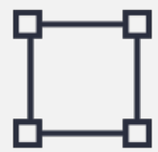
SpiralArms



Dust



MilkyWay



ProceduralMeshes

Mesh

GameObject
osservatore + Camera

Dati da file (CSVParser)

Modello

Parametri del modello
e della simulazione

UI

Utente

Demo