

# Evaluation et prototypage avancé de nouvelles technologies

---

Visualisation 3D dans votre navigateur web grâce au WebGL



STEINMETZ Arnaud



# ☐ SOMMAIRE

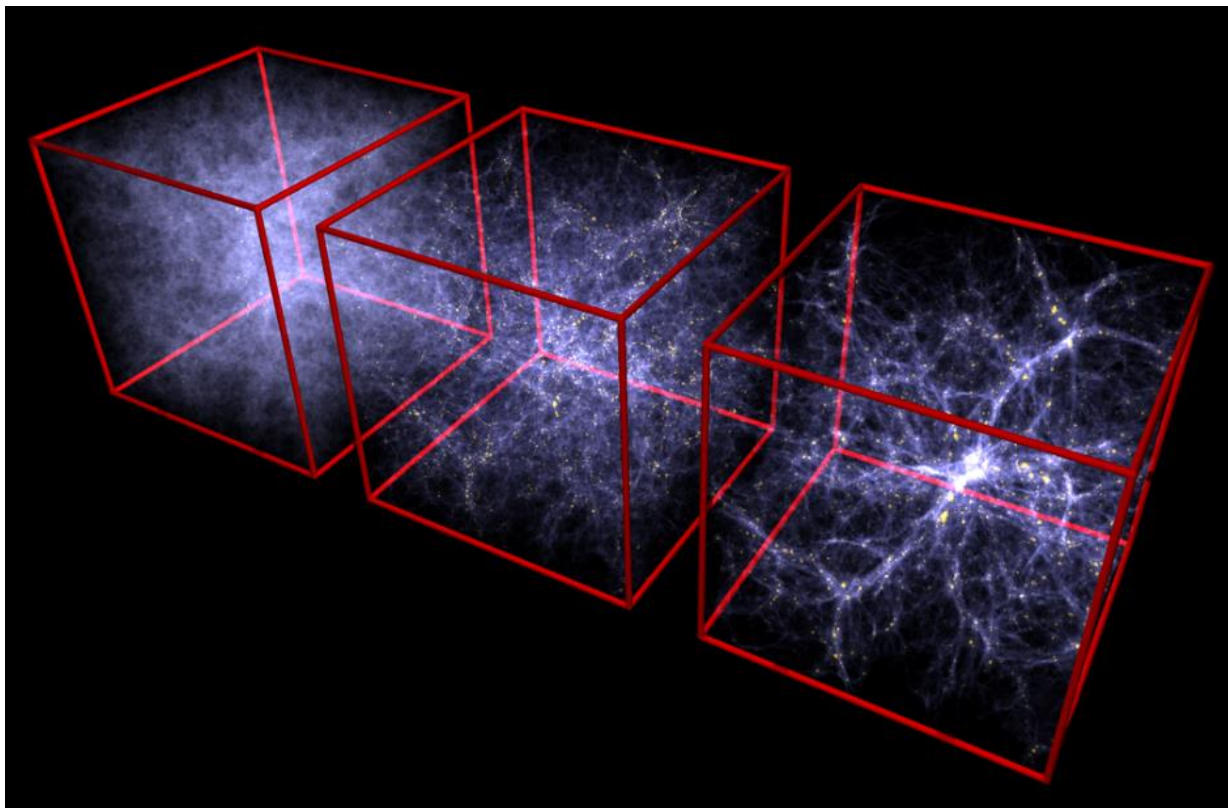
1. Contexte
2. Problématique
3. Outils et technologies
4. Phase d'évaluation
5. Phase de développement
6. Propositions d'évolution

# ☐ CONTEXTE

- **Observatoire astronomique est structuré en 3 équipes :**
  - L'équipe Galaxies
  - L'équipe Hautes Energies
  - L'équipe du CDS
  
- **Le CDS a pour mission de :**
  - Rassembler des informations concernant les objets astronomiques
  - Distribuer ces informations à la communauté astronomique internationale
  - Conduire des recherches utilisant ces données.

# □ PROBLEMATIQUE

« Création d'une application prototype permettant de visualiser des cubes de données 3D par l'utilisation de WebGL. »



# □ PROBLEMATIQUE



- Qu'est ce que signifie WebGL ?
  - Spécification d'affichage 3D pour le web
  - Basé sur le standard OpenGL
  - Utilisation de l'accélération matérielle (GPU)

# □ PROBLEMATIQUE

- Qu'est ce qui est utilisé actuellement ?

- Planétarium : OpenGL
- Skybot3D : OpenGL
- Stage Philippe Gaultier : OpenGL

- Pourquoi utilisé WebGL ?

- Portabilité
- Accessibilité
- Facilité de déploiement

- Avantage de WebGL par rapport aux concurrents :

- Fonctionne sans plugin
- Basé sur HTML5
- Intègre un moteur 3D très complet



# □ OUTILS ET TECHNOLOGIES

- Choix de la bibliothèque JavaScript

- BabylonJS
- ThreeJS

babylon



Logo Mozilla Firefox

Logo Google Chrome

- Logiciel de développement

- SublimText2
- Plugin Emmet

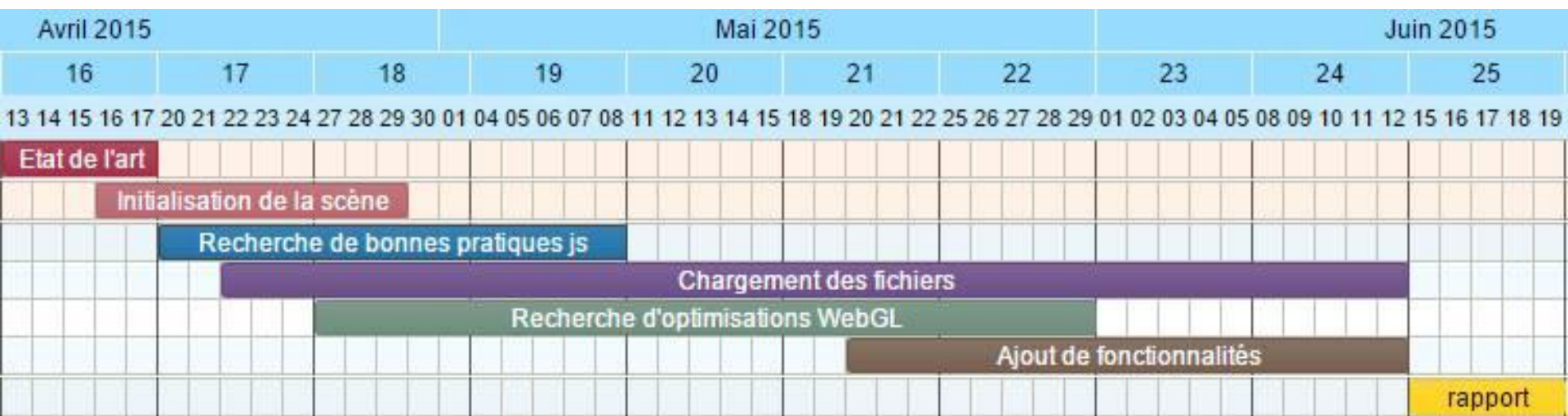


ee.js

- Outils d'analyse

- Sous chrome : outils de développement (Console, Timeline, Profiles)
- Sous Mozilla Firefox : FireBug

# □ Déroulement du stage

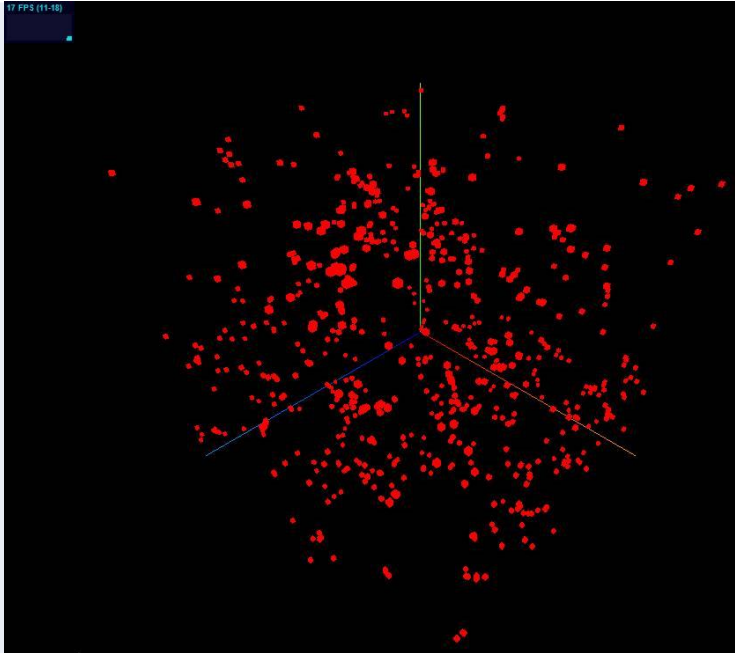
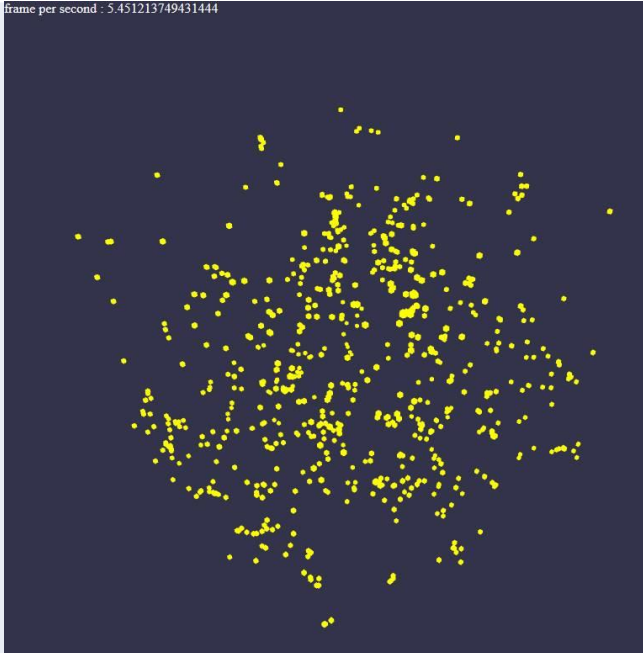




# □ Phase d'évaluation

	ThreeJS	BabylonJS
Date de sortie	Avril 2009	Eté 2013
Licence	Open Source	Open Source
Documentation/Tutoriels	Nombreux	Rare
Github	~10 300 commits 443 contributeurs	1500 commits 48 contributeurs
Stackoverflow	~6400 tags	46 tags

# □ Phase d'évaluation

ThreeJS	BabylonJS
Chargement de 3282 éléments en 52 ms avec 17 fps	Chargement de 3282 éléments en 1177 ms avec 5,5 fps
	

# □ Phase de développement

→ Éléments fondamentaux

- Éléments fondamentaux

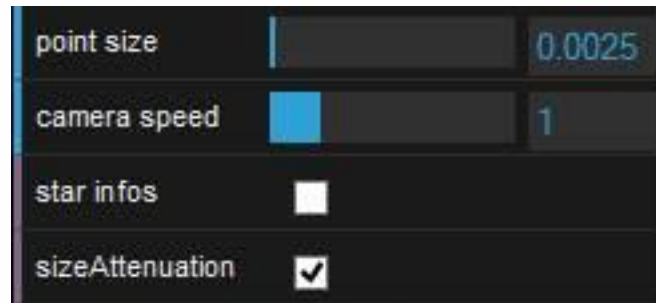
- Une scène : contient tous les éléments à afficher.
- Une caméra : sert à la prise de vue
- Un renderer : s'occupe d'afficher les éléments de la scène à l'écran

- Boucle de rendu

```
function render() {  
  requestAnimationFrame(function () {  
    render();  
  });  
  renderer.render( scene, camera );  
  controls.update(clock.getDelta());  
}
```

- Interaction

- Utilisation de dat-gui



# □ Phase de développement

→ Chargement des données

- Possibilités

- XMLHttpRequest
- File API

- Formats de données rencontrés

- String (CODA)

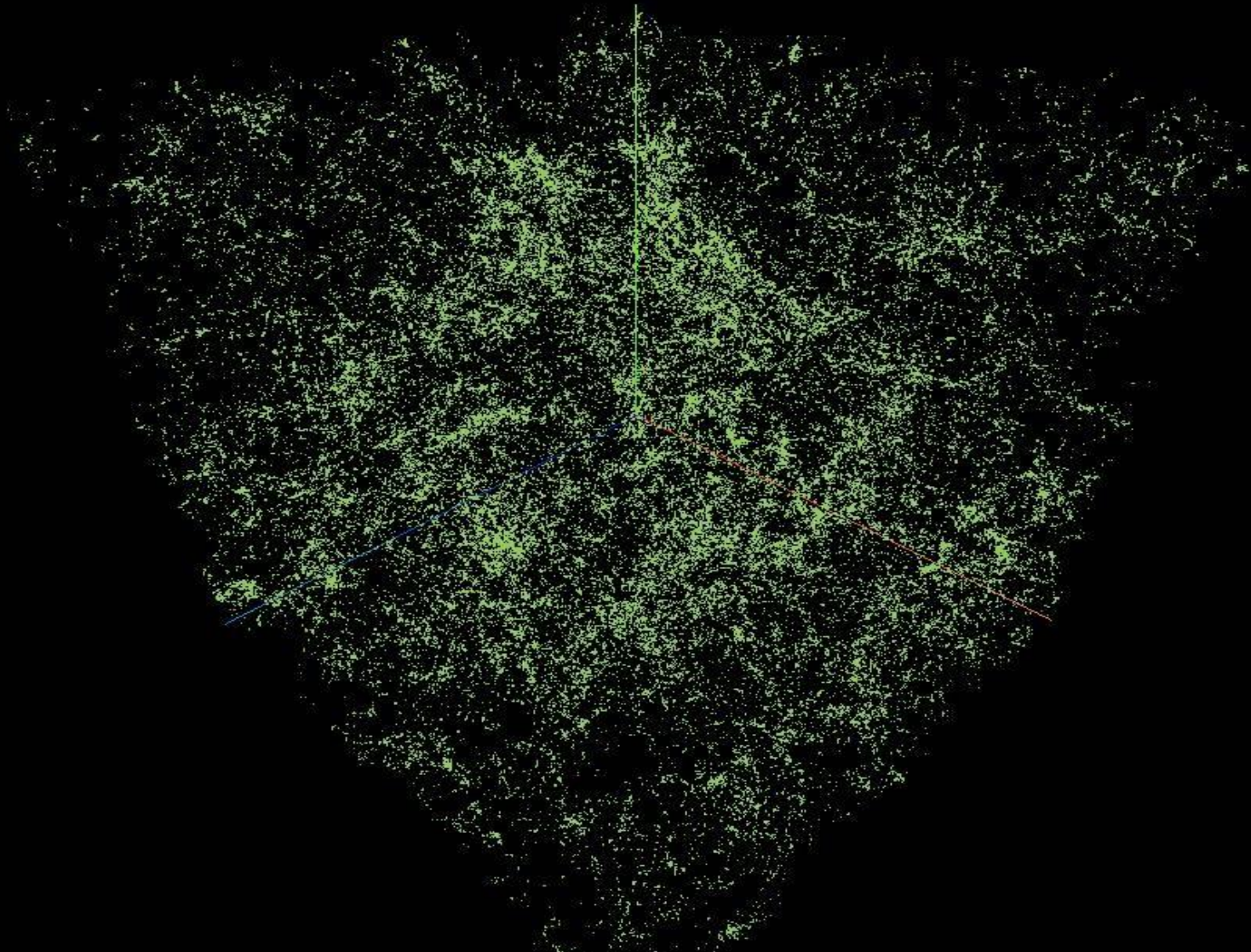
```
1 Age of the universe in Myr 247.229670543717
2 Id Mass(Msol) x(Lbox) y(Lbox) z(Lbox) Age(Myr)
3 10040 3.19393E+03 4.56792E-02 5.77574E-02 1.16433E-02 2.01123E+01 0.00000E+00
4 19692 3.19393E+03 3.79545E-02 4.76482E-02 9.89502E-03 7.18425E+00 0.00000E+00
```

- Binaire (EMMA)

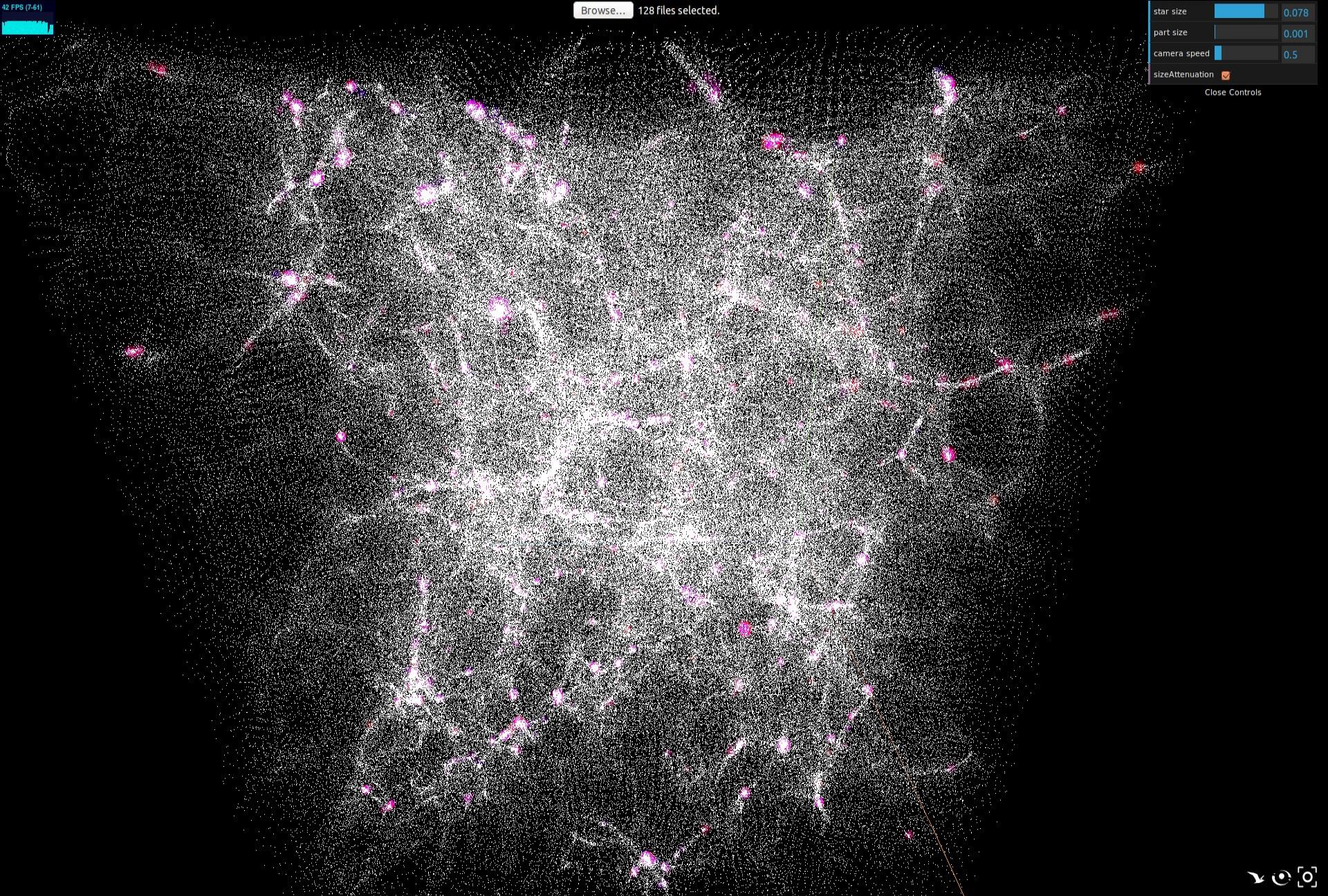
```
1 d23f 0000 cb3d 0d3c 4942 743a 349f 163b
2 03e6 d038 7acb 10ba e08b 9cb9 c64d 39ba
3 0000 0000 c87d d834 0000 0000 0000 0000
4 1636 113c 3e46 183b a61b 9438 0604 0bba
```

- JSON(Skybot)

```
{"flag":1,"ticket":1433200121015,"nbast":681379,"refdate":2457170,"asteroids":[[[536426,-11.09589863,-3.344656832,-3.549146778,-0.000720862,-0.005600321,-0.001076604,23.016301089,11.97,0.15,"Centaur","2010 BL4"],[577692,-7.782323979,6.424105122,1.609726424,-0.006483396,0.00092032,0.000197874,27.347271872,16.75,0.15,"Centaur","2011 SP25"],[676173,-4.007920896,2.591785803,0.725645607,-0.007018466,-0.006547989,-0.002977982,22.711850885,15.06,0.15,"Centaur","2015 BX514"],[631506,-2.229077566,-1.357117298,3.909654162,-
```



star size 0.078  
part size 0.001  
camera speed 0.5  
sizeAttenuation   
Close Controls

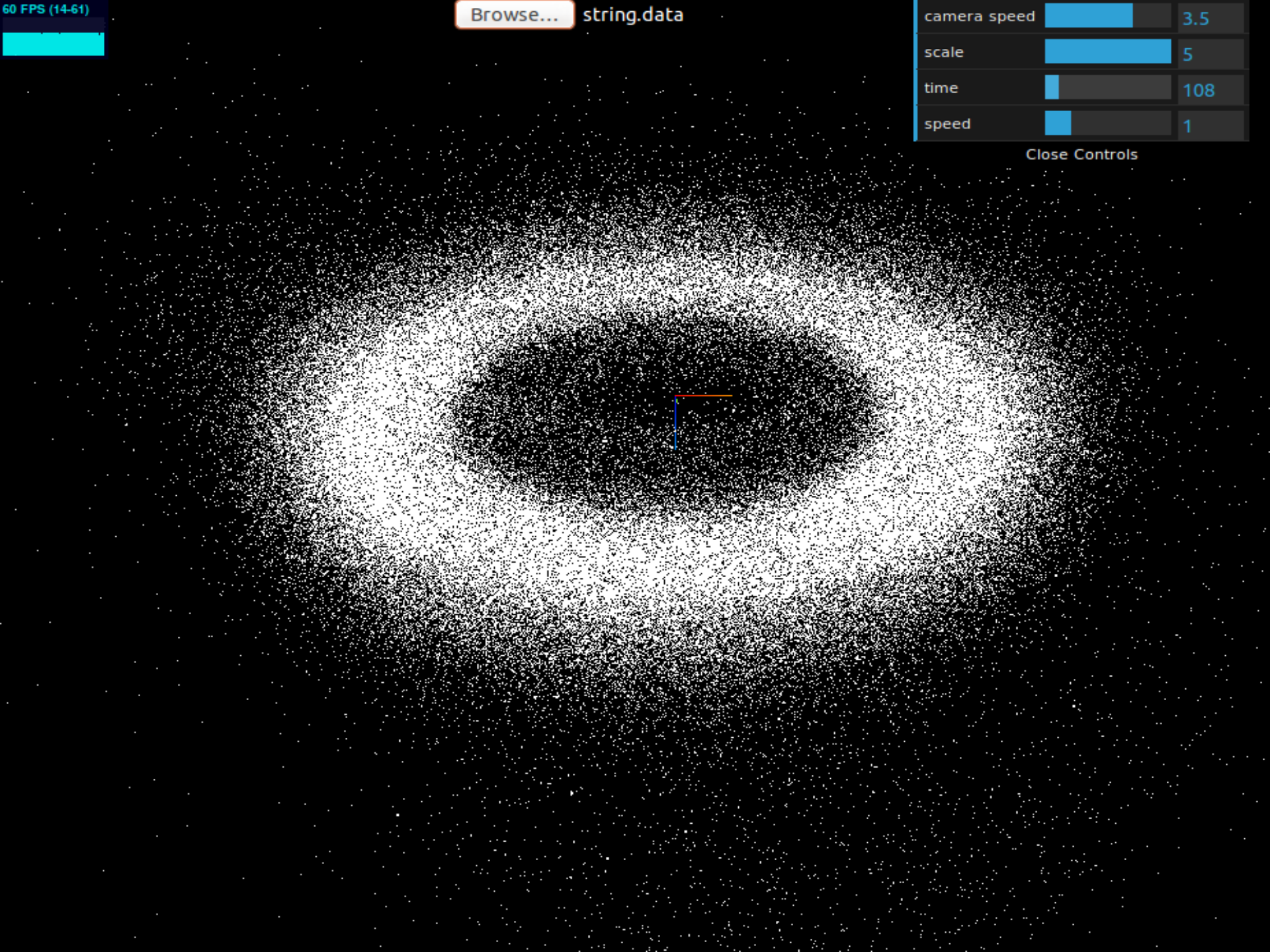


60 FPS (14-61)

Browse... string.data

camera speed	<input type="range"/>	3.5
scale	<input type="range"/>	5
time	<input type="range"/>	108
speed	<input type="range"/>	1

Close Controls



# □ Phase de développement

→ Représentation et structure dans l'espace

- **Type d'objet**

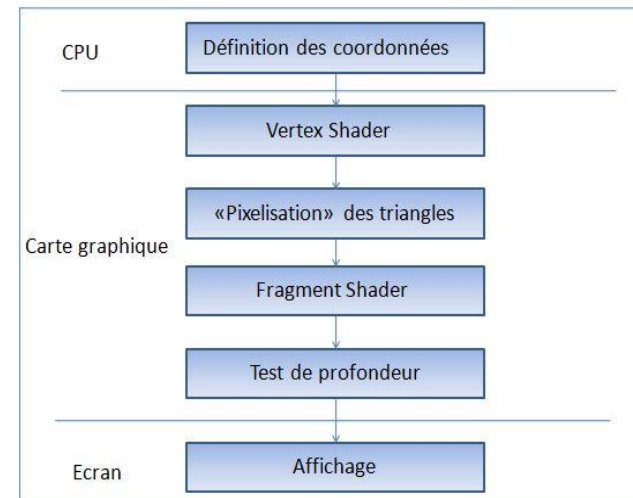
- Mesh (Sphères/Cubes) : 6 faces -> 12 triangles -> 36 points
- PointCloud (GL\_POINTS) : 1 point

- **Modification de l'apparence des points**

- Couleur dépendant de l'âge des étoiles
- Texture dépendant du type de fichier
- Autres possibilités (taille pour la masse, opacité pour de la densité)

- **Shader**

- Vertex shader : permet d'effectuer un traitement pour chaque sommet.
- Fragment shader : permet d'effectuer un traitement pour chaque pixel

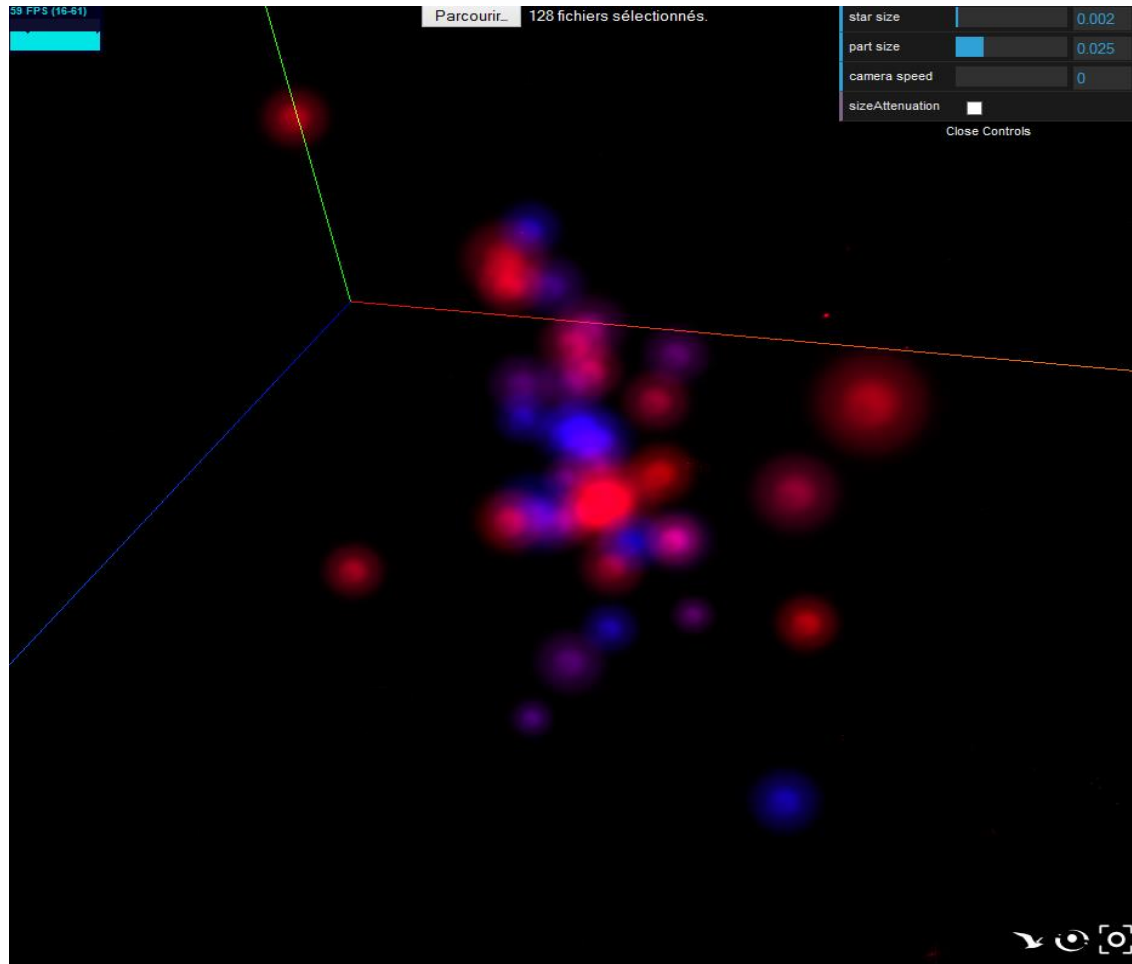




# □ Phase de développement

→ Représentation et structure dans l'espace

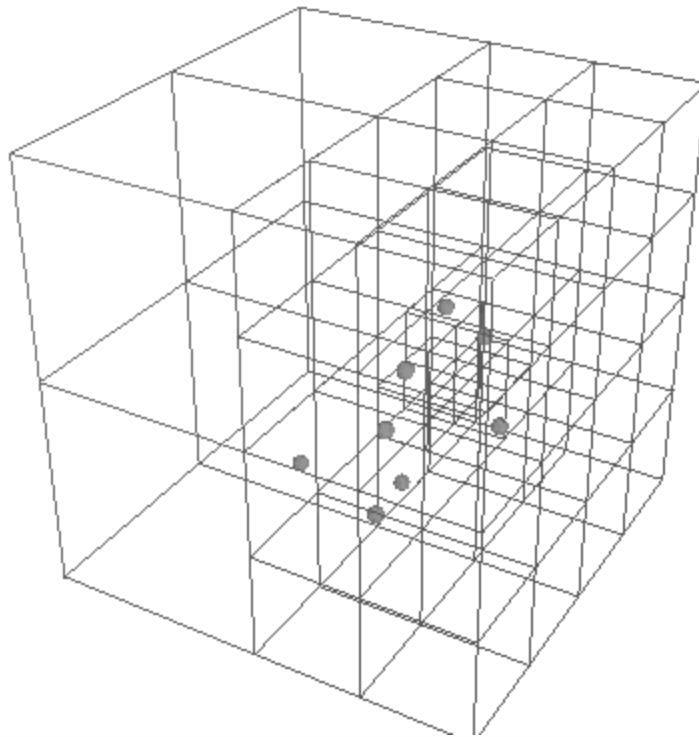
## Visualisation d'étoiles avec texture et couleurs



# □ Phase de développement

→ Représentation et structure dans l'espace

Schéma représentant l'implémentation des octrees



# □ Phase de développement

→ Fonctionnalités

- Différents déplacements

- OrbitControls : Rotation autour d'un point invisible, zoom avec molette
- FirstPersonControls : Rotation de l'angle de la caméra, touches directionnelles
- EarthControls : Sélection d'un point du PointCloud et déplacement similaire à OrbitControls

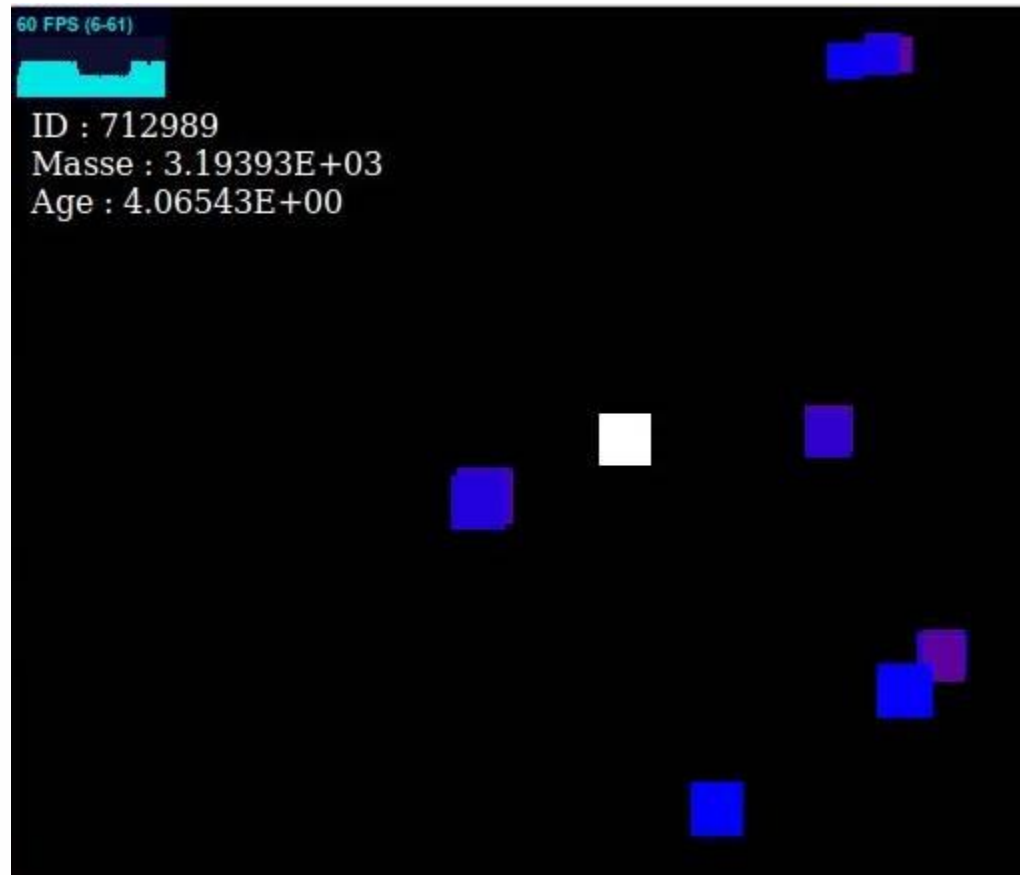
- Mode Multivues

- Pour comparer un même jeu de données à deux instants différents
- Tentative d'utilisation de la même caméra pour deux renderer
- Problème persiste lors du chargement des fichiers

# □ Phase de développement

→ Fonctionnalités

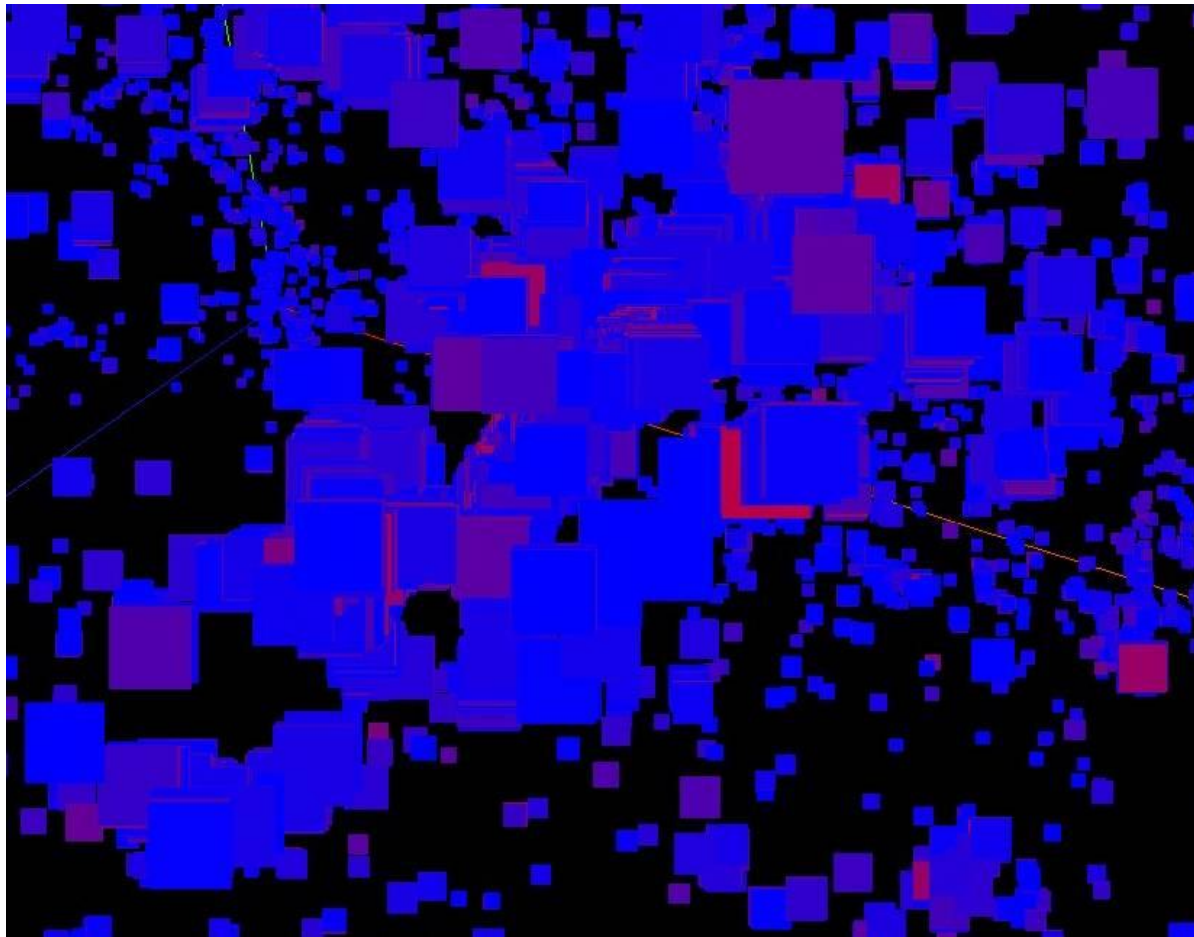
- Sélection d'un point et affichage de données le concernant



# □ Phase de développement

→ Fonctionnalités

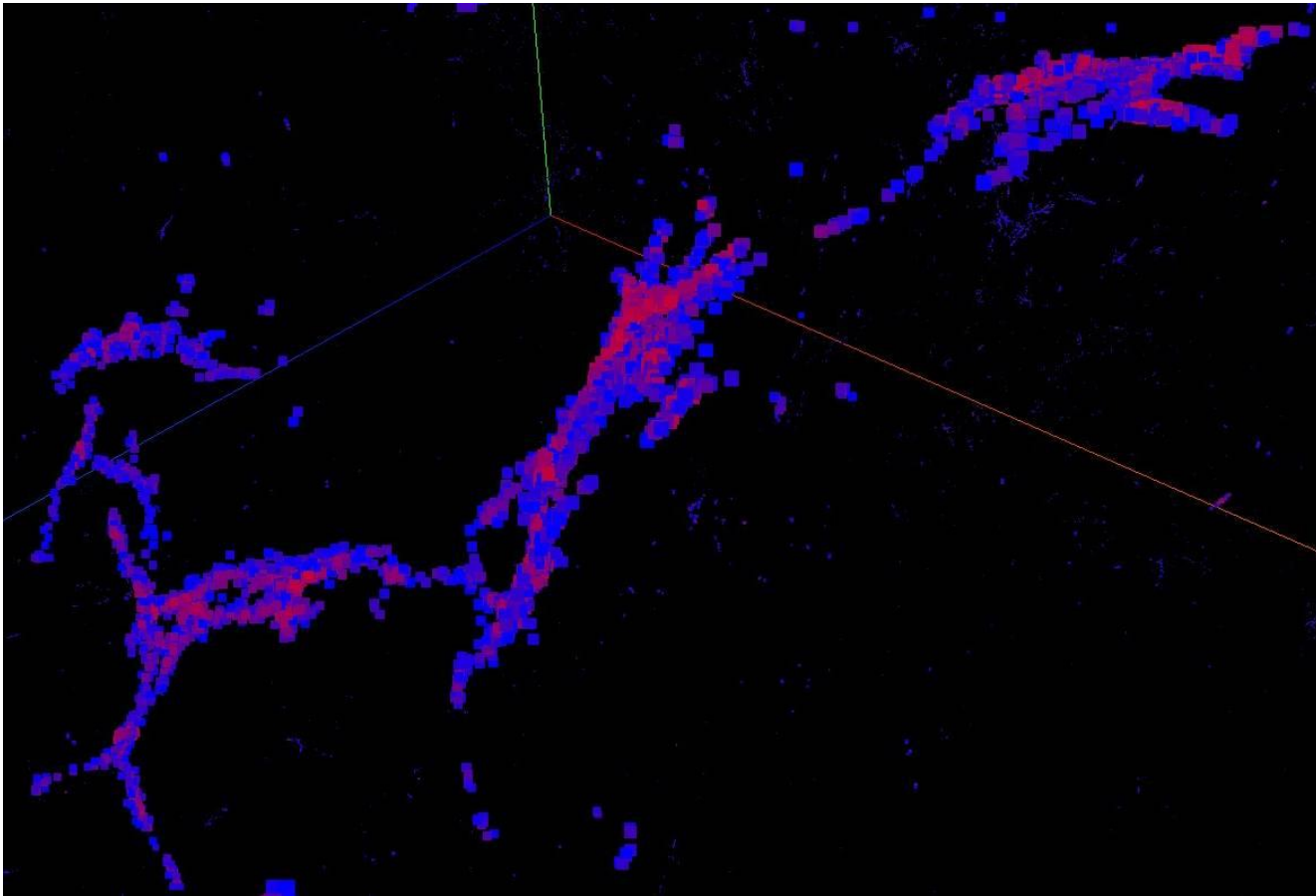
- Zoom et modification d'échelle (précision)



# □ Phase de développement

→ Fonctionnalités

- Zoom et modification d'échelle (précision)



# □ Phase de développement

→ Fonctionnalités

- Notion de temps (animation)

Démonstration vidéo

# □ Propositions d'évolution

- **Formats de données**
  - Utilisation d'un format de données standardisé
- **Octrees et WebWorkers**
  - Chargement des sous-cubes parallélisé grâce aux workers.
  - Chargement de textures de galaxies puis des textures d'étoiles lorsque l'on s'en approche
- **Ouverture vers les standards de l'OV**
  - Interaction avec les services de CDS
- **Fonctionnalités en cours de développement**
  - Multi-processing
  - Mozilla Virtual Reality



# □ Des questions ?



