

# WEB – Projet Javascript

---

Présentation du 26 janvier 2022



# Contenu de la présentation

## 01 La 3D

---

De la 2D à la 3D

## 02 Nouvelles entités

---

Des nouveaux  
éléments de jeu

## 03 Three.js - Introduction

---

C'est quoi ? Comment ça  
marche ?

## 04 Three.js - La caméra

---

Les yeux de  
l'utilisateur

## 05 Three.js - Les modèles

---

A quoi ressemblera  
mon vaisseau ?

## 06 Démonstration et conclusion

---

Que la bataille  
commence !



## De la 2D ...

id: 4  
t: 110530  
x: 200  
y: 53  
speed: 100  
angle: 2

id: 1  
t: 110530  
x: 302  
y: 127  
speed: 250  
angle: 275

id: 1  
t: 110530  
x: 637  
y: 72  
speed: -125  
angle: 141

id: 1  
t: 110530  
x: 510  
y: 150  
speed: 250  
angle: 0

id: 4  
t: 110530  
x: 216  
y: 416  
speed: 100  
angle: 1

id: 4  
t: 110530  
x: 453  
y: 400  
speed: 100  
angle: 0

... À la 3D

---

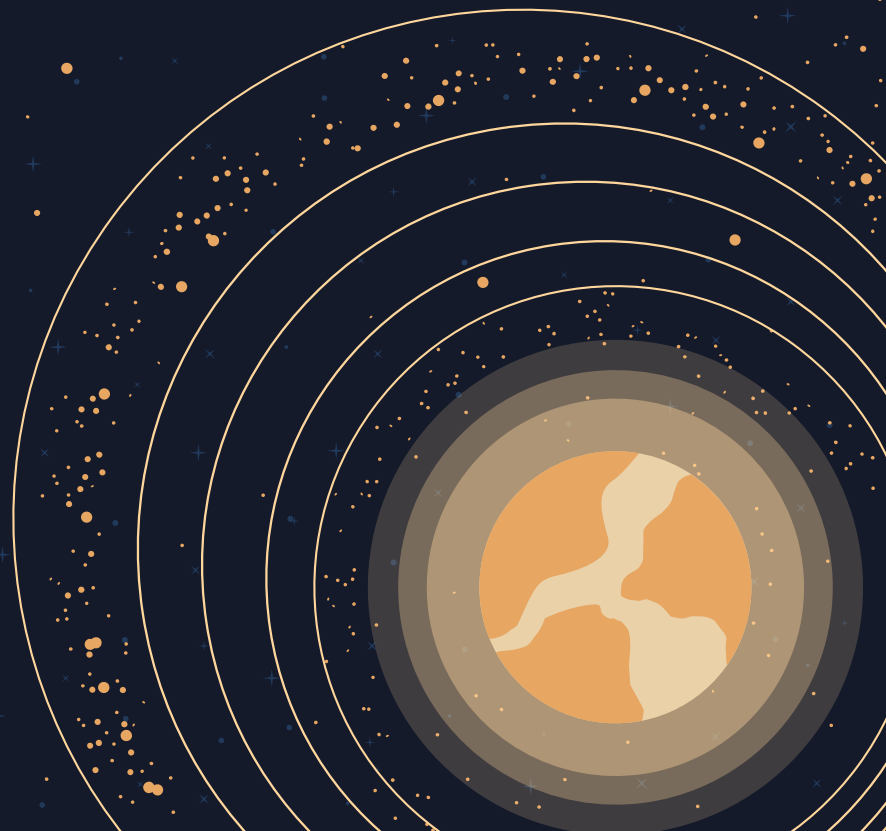


# La 3D ? Et alors ?

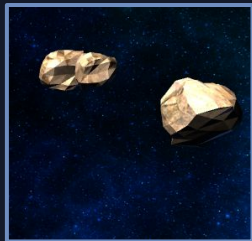
Le passage à la 3D implique de revoir la structure de base et repenser à la façon de faire le jeu

- Quel code doit être gardé / remplacé ?
- Quelles technologies permettent de faire de la 3D ?
- À quoi doit ressembler le rendu final ?

Beaucoup de questions se posent dès qu'il faut gérer une dimension de plus !



# Nouvelles entités



## 01 Astéroïdes

---

Les joueurs  
peuvent se cacher

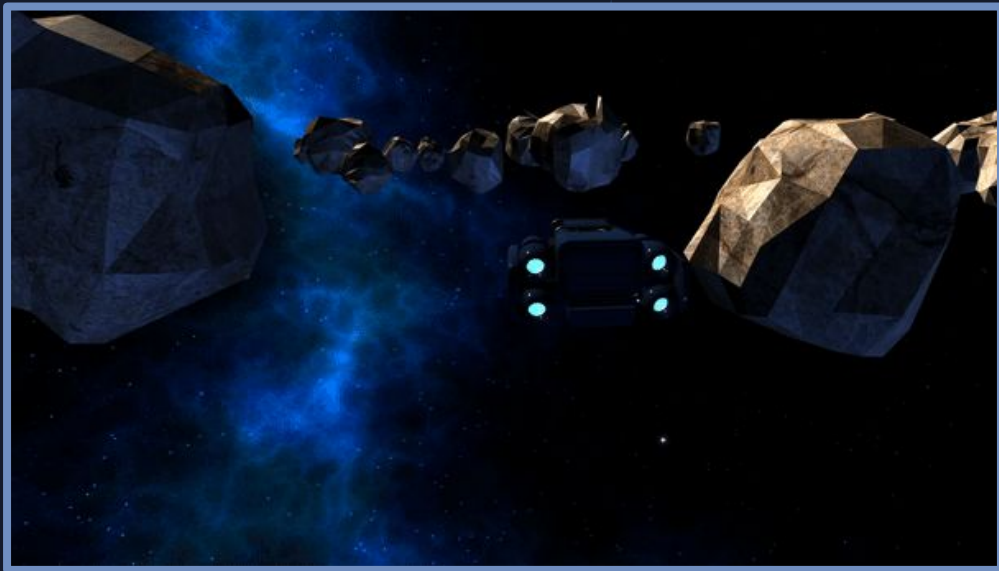


## 02 Power Up

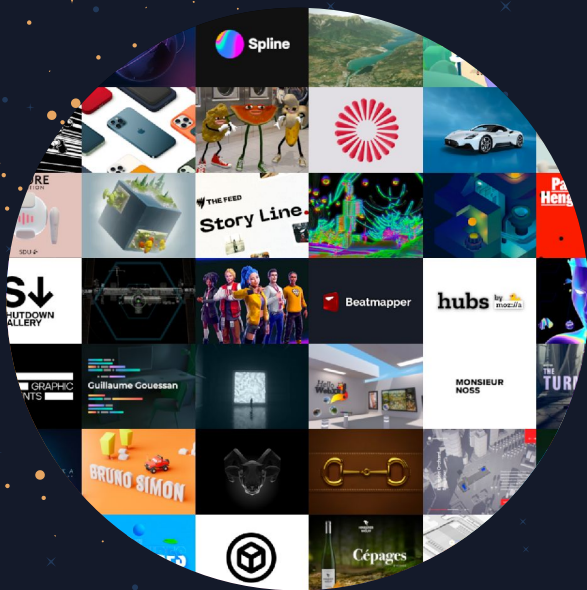
---

Récupération de  
points de vie

# Des résultats parfois inattendus



# THREE.JS



- Librairie Javascript
- Affichage 3D
- Utilise WebGL
- Cross-Browser

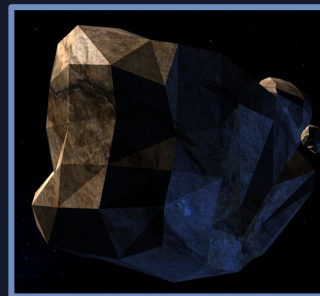
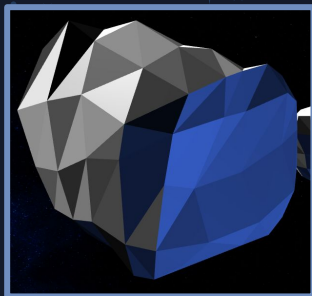


# THREE.JS - La caméra



- Doit suivre le joueur
- Respecte une distance avec le joueur
- Sa position est basée sur celle du joueur
- Calculée du côté client

# THREE.JS - Les modèles



## 01 Mesh + Material

---

Des faces, des  
vertices, des  
arrêtes.

Bref, des triangles

## 02 Texture

---

On place une image  
sur tout le reste

# C'est l'heure de la DEMO

---



# Merci!

Des questions ?

