

Introduction au modèle de  
conception :  
Entités / Composants / Systèmes  
[mathieu.muratet@lip6.fr](mailto:mathieu.muratet@lip6.fr)

Gamagora 02/10/2020

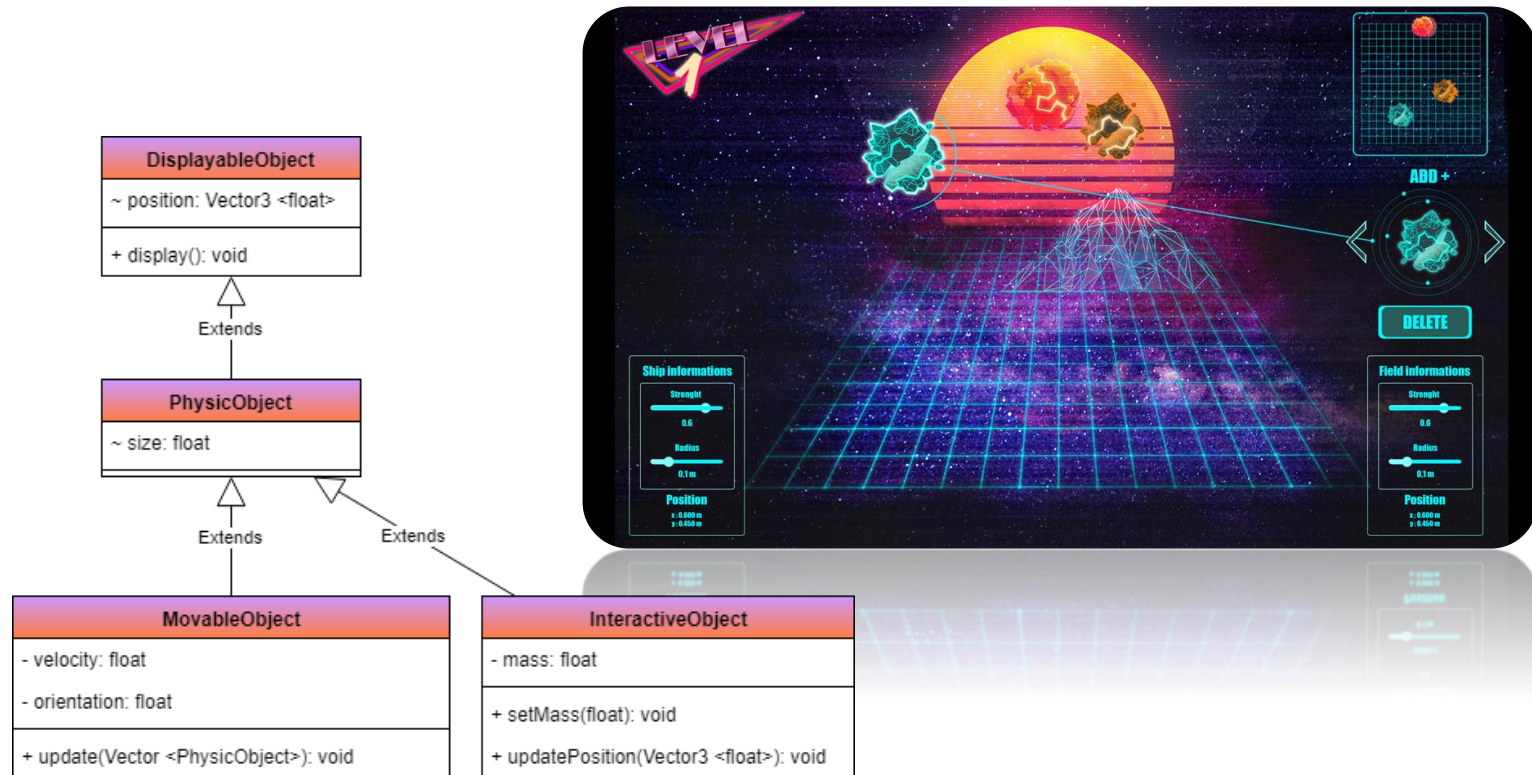
# Origine de l'ECS

- Issu du jeu vidéo
  - Thief: The Dark Project (1998)
  - Dungeon Siege (2002)
    - Bilas, S. (2002). A data-driven game object system. In Game Developers Conference Proceedings.
  - Operation Flashpoint: Dragon Rising (2007)
- Pourquoi cette architecture ?
  - Le processus de création d'un jeu vidéo est hautement itératif
    - Besoin de tester de nouvelles mécanique de jeu
    - Besoin de souplesse

# Notre intérêt pour les ECS

- Génome (2013)
  - Thèse de Bruno Capdevilla (2009-2013)
  - Bourse CIFRE avec KTM Advance
  - Framework AS3 (Flash)
- FYFY (Family For unitY) (2016)
  - Plugin Unity
    - <https://gitlab.lip6.fr/mocah-public/FYFY.git>
    - <https://assetstore.unity.com/packages/tools/ai/fyfy-157154>

# Exemple avec StrangerFields



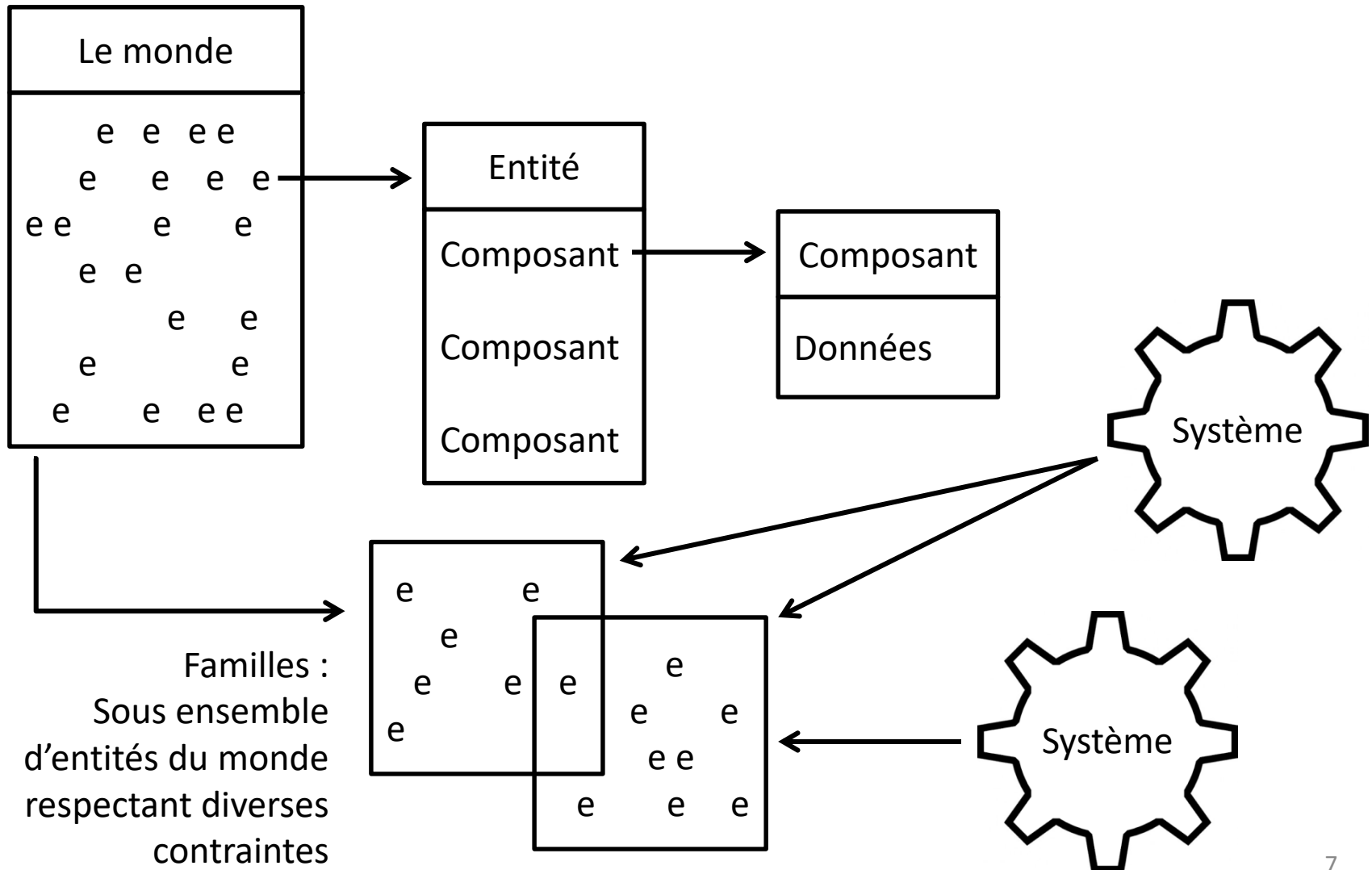
# Un peu de vocabulaire

- **Entité :**
  - Représente un objet dans le jeu, c'est-à-dire n'importe quel élément d'un jeu
  - **Elle ne possède ni données propres, ni méthodes propres**
  - Une entité est une sorte d'identifiant d'un objet du jeu
- **Composant :**
  - représente un aspect d'un objet ou d'un ensemble d'objets: la vitesse, la position, etc.
  - **Il n'y a jamais de code dans les composants**
  - Une entité est caractérisée par un ensemble de composants
  - Plusieurs entités peuvent contenir les mêmes types de composants

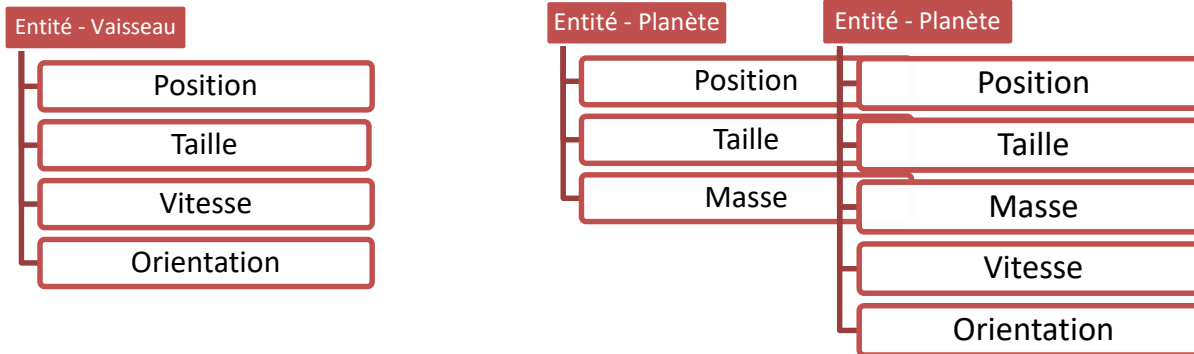
# Un peu de vocabulaire

- **Système :**
  - Implémente la fonction « onProcess » qui sera appelé à chaque pas de simulation :
    - Récupération des entités en relation avec le système
    - Traitement des entités récupérées (lire/modifier des composants, ajouter/supprimer des composants, créer/supprimer des entités, ...)
- **Famille :**
  - Une famille est définie comme l'ensemble des entités satisfaisant certains critères (contenir ou pas tel et tel composant)
  - La famille est le mécanisme permettant aux systèmes d'accéder aux entités
  - Réalisation de traitements particuliers lors de l'entrée et de la sortie d'une entité dans la famille

# Vue d'ensemble du formalisme



# Un modèle centré sur les données



Système **AppliquerForces** : familles de travail

[Position, Vitesse, Orientation]

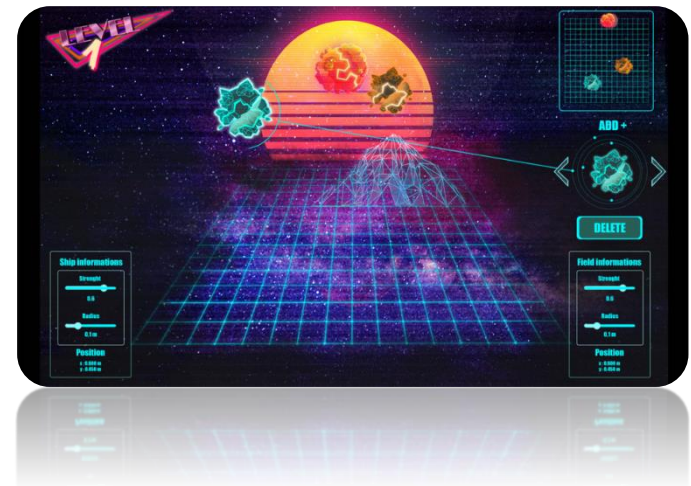
[Position, Masse]

- Applique les forces gravitationnelles sur le(s) vaisseau(x)

Système **Déplacement** : famille de travail

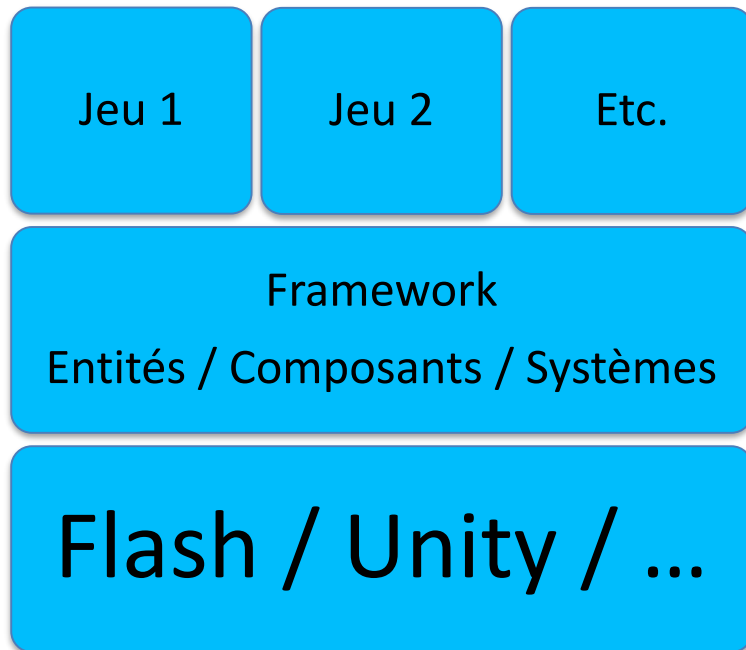
[Position, Vitesse, Orientation]

- Calcule la nouvelle position en fonction de la vitesse et de l'orientation courante





# Architecture générale



Entités et systèmes spécifiques à chaque jeu

Boucle de gameplay temps réel  
Gestion des systèmes & entités

Moteur

# ECS et Unity

- Architecture orienté objet classique
- Synchronisation entre GameObjects



- Accéder à un ensemble de GameObjects
- Donne un cadre méthodologique de conception

# Retour sur nos motivations

- D'un point de vue recherche
  - Simplifier l'intégration de nos outils de recherche dans le processus de développement d'un jeu sérieux :
    - Module de suivi
    - Génération de feedbacks adaptatifs
    - Recommandation de ressources
    - Accessibilité
- D'un point de vue enseignement
  - Découvrir une nouvelle méthode de conception

# Conclusion

- Avantages
  - Grande évolutivité pour tester de nouvelles mécanique de jeu
  - Travail en équipe
  - Pause, Replay, Enregistrement, Chargement
- Inconvénients
  - Changer son schéma de penser
  - Verbeux