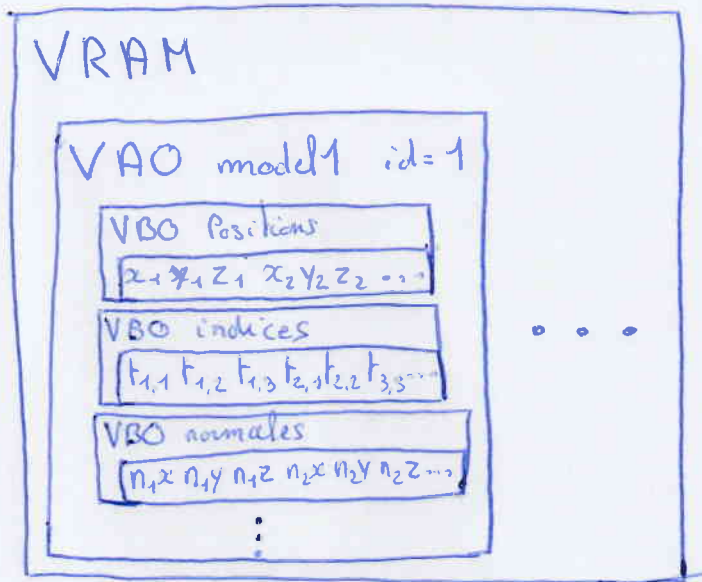


Initialisation (démarrage du moteur):

- Le CPU charge les modèles (depuis un fichier .obj ou une génération à la volée)
- Le CPU va créer les VAO* et les VBO* de chaque modèle et les stocker en VRAM, en conservant les ids pour pouvoir les récupérer.
- Le CPU charge les shaders* en VRAM



VAO: Vertex Array Object
encapsule une liste de VBO, et représente entièrement un modèle

VBO: Vertex Buffer Object
liste de variable représentant un attribut de modèle (position des vertex, indices des triangles, normales, coord des textures)

Shader: programme exécuté par le GPU, hautement parallélisable

Rendu:

- Le CPU enregistre les entités à afficher auprès du moteur de rendu
- Le moteur va associer un modèle et une liste d'entités utilisant ce modèle (pour ne charger qu'une fois le modèle)
- Pour chaque modèle, on récupère la liste des entités et on active le VAO associé à ce modèle.
 - Pour chaque entités on calcule la matrice de transformation et on charge les variables uniformes en VRAM.
 - on demande au GPU de rendre l'entité (`drawTriangles()`)

