# Agile

# Collaboration entre des équipes auto-organisées, une amélioration continue, et elles encouragent des réponses flexibles au changement.

Les méthodologies agiles permettent une évolution continue du projet.

## Scrum

Elle se base sur une approche collaborative, où le projet est divisé en sprints, généralement de deux à quatre semaines.

## Kanban

Kanban est une méthode visuelle basée sur des tableaux de tâches.

# WebGL

Web Graphics Library (bibliothèque de graphismes pour le Web) sert à effectuer du rendu 3D dans un navigateur web.

Permettant le rendu de graphismes en 2D ou 3D avec de hautes performances en utilisant le GPU, sans avoir à utiliser de plugins additionnels.

# Diagramme de classe

**Association bidirectionnelle** : Une image contenant outil

Description générée automatiquement

Cette association est utilisée lorsque deux classificateurs sont étroitement liés et peuvent se stocker mutuellement dans leurs champs.

**Généralisation/Héritage** : 

Relations entre la classe parent et la classe enfant.

**Dépendance** : 

La relation de dépendance existera et le changement d'une classe entraînera le changement de l'autre entité.

# Poo

Programmation Orientée Objet, permet d'organiser et de structurer le code en utilisant des objets, qui sont des instances de classes. La POO se base sur les concepts suivants.

# Corse

Cross-Origin Resource Sharing = mécanisme de sécurité utilisé par les navigateurs web pour contrôler les requêtes d'accès aux ressources entre différents domaines (origines). Il permet à un site web d'accéder à des ressources provenant d'un autre domaine, ce qui serait normalement restreint pour des raisons de sécurité.

# Dom

Document Object Model, Document Object Model : Le DOM (Document Object Model) est une représentation de la structure d'un document HTML, XML ou d'autres types de documents. Il fournit une interface de programmation qui permet aux langages de programmation, tels que JavaScript, d'accéder dynamiquement aux éléments, aux attributs et au contenu d'un document, et de les modifier. Le DOM est souvent utilisé dans le développement web pour la manipulation et la modification de pages web en temps réel.

# Vite

**Build tools** = Utilisé pour automatiser le processus de construction, de compilation et de déploiement d'une application JavaScript.

Vite est conçu pour accélérer le processus de développement

Est un outil utilisé dans le développement web pour regrouper et organiser efficacement les modules JavaScript d'une application en un seul fichier ou ensemble de fichiers, prêts à être déployés sur un navigateur web. Semblable Webpack

Hot module realoding : mettre à jour dynamiquement les modules d'une application en cours d'exécution, sans avoir besoin de recharger complètement la page.

# Three.js

**Scène** : La scène est l'endroit où tous les objets 3D seront placés. C'est l'environnement principal dans lequel vous travaillerez.

**Caméra** (Camera) : La caméra définit le point de vue depuis lequel vous verrez la scène. Elle peut être positionnée et orientée pour obtenir différents angles de vue.

* **Fov (field of view)**  
  Le champ de vision. Cela dicte la taille de l'espace vertical que la vue de votre caméra peut atteindre. (Angle de vue)
* **Aspect**  
  Il s'agit du format d'image que vous utilisez pour créer le champ de vision horizontal basé sur la verticale.
* **Near**  
  Il s'agit du plan de vue le plus proche (là où commence la vue de la caméra).
* **Far**  
  C'est le plan de vue éloigné (où la vue de la caméra se termine)

**Rendu** (Renderer) : Le rendu est le processus de conversion des objets 3D de la scène en images 2D à afficher sur l'écran. Three.js utilise WebGL pour effectuer le rendu en exploitant la puissance du GPU de l'ordinateur.

Dessine

**Maillage** (Mesh) : Un maillage représente un objet 3D dans Three.js. Il est composé de sommets (vertices) reliés par des arêtes (edges) pour former des triangles. Les maillages peuvent être géométriques (primitifs tels que des cubes ou des sphères) ou importés à partir de fichiers 3D.

**Matériau** (Material) : Le matériau définit l'apparence visuelle d'un maillage, comme sa couleur, sa texture, son éclairage, etc. Three.js propose une variété de matériaux prédéfinis que vous pouvez utiliser, ou vous pouvez créer vos propres matériaux personnalisés.

**Lumière** (Light) : Les lumières sont utilisées pour illuminer la scène et donner aux objets des reflets et des ombres réalistes. Three.js prend en charge différents types de lumières, tels que les lumières directionnelles, les lumières ponctuelles et les lumières de tache.

**Contrôles** (Controls) : Les contrôles permettent à l'utilisateur d'interagir avec la scène 3D. Par exemple, vous pouvez utiliser les contrôles de la souris pour effectuer une rotation, un zoom ou un déplacement de la caméra.

**Animation** : Three.js propose des outils d'animation pour animer les objets 3D dans la scène. Vous pouvez créer des animations basées sur des keyframes, des courbes de mouvement ou des calculs personnalisés.

# WebGL

**WebGL** : WebGL (Web Graphics Library) est une API JavaScript qui permet d'accéder à l'accélération matérielle du GPU (unité de traitement graphique) pour effectuer du rendu 3D dans un navigateur web. Il utilise le langage de shading GLSL (OpenGL Shading Language) pour décrire les calculs graphiques.

**Contexte WebGL** : Le contexte WebGL est un objet JavaScript qui représente une zone de rendu WebGL dans une page web. Il est obtenu en récupérant le contexte WebGL à partir d'un élément canvas HTML.

**Shader** : Les shaders sont des programmes écrits en GLSL qui sont exécutés sur le GPU pour effectuer différentes opérations de traitement graphique. Il y a deux types de shaders en WebGL : les shaders de sommet (vertex shaders) qui manipulent les sommets des maillages, et les shaders de fragment (fragment shaders) qui calculent la couleur de chaque pixel visible.

**Buffer** : Les buffers sont utilisés pour stocker les données géométriques des maillages, telles que les positions des sommets, les normales, les coordonnées de texture, etc. Les buffers sont créés et remplis avec les données nécessaires, puis liés aux attributs des shaders pour le rendu.

**Textures** : Les textures sont des images qui peuvent être appliquées sur les surfaces des maillages pour ajouter des détails visuels tels que des couleurs, des motifs, des reflets, etc. Les textures sont généralement chargées à partir d'images et peuvent être utilisées pour modifier l'apparence des matériaux dans Three.js.

**Pipeline de rendu** : Le pipeline de rendu décrit les différentes étapes de traitement nécessaires pour convertir les données 3D en images 2D affichées à l'écran. Cela inclut les transformations des sommets, l'application des shaders, le culling des faces, l'application des lumières, etc.

**VAO et VBO** : Les VAO (Vertex Array Objects) et VBO (Vertex Buffer Objects) sont des objets WebGL utilisés pour stocker et organiser les données géométriques des maillages. Les VAO enregistrent les états des VBO et des attributs de shader associés pour une utilisation ultérieure.

Deltatime : **deltaTime** est généralement utilisé pour rendre les mouvements et les animations plus fluides et indépendants de la fréquence d'images (frame rate) du système. Au lieu de déplacer ou d'animer les objets d'une quantité fixe à chaque frame, on les déplace proportionnellement au temps écoulé depuis le dernier frame. Cela garantit que le mouvement ou l'animation se déroule à peu près à la même vitesse, quel que soit le taux de rafraîchissement du rendu.