

Deadline

Vendredi 04/06/2021 – 00h
(Montréal)

Nombre d'étudiants par groupe : 3

GIT:

git clone

ssh://git@git.isartintra.com:2424/2019/GP_2022MTL_Moteur/{nom_du_groupe}.git

Rendu :

- Un Tag GOLD.
- Il est attendu un binaire pour le push final.
- La documentation à jour.

« Vieux moteur que j'aimais »

Ce sujet fait suite à celui du Moteur Partie 2.

Vous **devez** avoir terminé la phase 2 avant de vous attaquer aux features de la phase 3 (sauf feu vert du professeur).

Tâches

- À la suite de la Phase 2, vous devriez avoir tous les outils de base pour commencer à créer des jeux entiers. La phase 3 est essentiellement dédiée à l'intégration de features additionnelles et la consolidation de l'existant.

Etape 1 : Son

Intégrer l'API audio de votre choix à votre moteur de façon à gérer les points suivants:

- Prendre en charge le format ogg
- Lire un fichier audio (*play/pause/stop, en lecture simple ou en boucle*) à l'exécution du jeu (*via un script*) et également en mode éditeur
- Gérer la spatialisation du son 2D et 3D
- Concevoir une interface de mixage des volumes sonores (*master et sfx*) dans l'éditeur
- Permettre le ré-échantillonnage d'un son (*192 kHz, 64 kHz, etc.*)
- Choisir son mode d'import (*decompress on load, decompress on the fly*)

Les fichiers audio doivent bien sûr être pris en charge par votre gestionnaire de ressources.

Etape 2 : Animation & Editeur

Intégrer le SDK FBX à votre moteur afin de gérer les points suivants :

- Charger/décharger un .fbx

- Extraire les informations du fichier (*bones, animations, materials*) et les stocker dans des structures de données adaptées
- Utiliser les différentes animations d'un modèle au sein du jeu
- Visualiser dans l'éditeur un fbx et la liste de ses animations existantes.
- Jouer une animation sélectionnée (*play/pause/stop*) à l'exécution du jeu (*via un script*) et également en mode éditeur
- Permettre la navigation dans les keyFrames de l'animation
- Choisir une vitesse d'animation (*15 fps, 30 fps, 60 fps, etc.*)

- **Vous devez coder le plus proprement possible, avoir des fichiers de configurations et de logs.**
- **Votre git doit être propre et refléter le travail de chacun.**
- **La vitesse de votre jeu ne doit pas dépendre de la puissance de la machine.**

La soutenance évaluera :

- Si la liste des tâches est correctement effectué.
- Sur la documentation rendue.
- Si le code est propre, bien architecturé et aussi performant que possible.
- **Bonus:** L'ergonomie du moteur et le polish du jeux exemple.

Liens utiles

- [Game design Pattern - Vidéo](#)
- [Design-Patterns and Anti-Patterns](#)
- [Dependency Injection and Abstractions](#)