

HMINT317 - Moteurs de jeux – TP3 et TP4

Niveaux de détails

Objectifs

Le but de ce TP est de mettre en place des communications entre plusieurs fenêtres gérées par le moteur de jeu et de créer des effets graphiques permettant de simuler le passage des saisons.

Nous allons ensuite nous intéresser à étudier la gestion efficace de scène 3D. Pour cela, nous allons modifier notre application précédente pour intégrer :

- Une structure de quadtree pour le terrain
- Un gestionnaire de niveau de détails des autres éléments de la scène
- Un gestionnaire de scène

Bonus :

- Afficher une scène infinie
- Garder un rendu temps réel

Synchroniser les fenêtres entre elles

La semaine dernière, vous avez mis en place une décomposition de votre application en plusieurs sous-fenêtres. Nous allons réutiliser cette approche pour étudier la communication entre ces fenêtres.

Vous utiliserez pour cela un timer dédié (calendrier) qui simule le décompte des jours et envoie des signaux aux quatre fenêtres à chaque changement de saison.

Le but est de transférer un message à chaque « application » cliente, en fonction du temps écoulé (quelques secondes). Vous simulerez pour chaque fenêtre une période différente de l'année (les 4 saisons). Le message devra ainsi contenir la saison courante de la fenêtre. Ensuite, faites circuler les différentes saisons entre les fenêtres.

Conseil : utiliser les fonctions de passage de signaux de QT (signals, slots) pour synchroniser les communications entre les fenêtres et le « calendrier ».

Simuler les changements de saisons

Nos saisons seront caractérisées par différents comportements.

L'été – une lumière vive, une couleur vive au sol

L'automne – une couleur rouge/orange au sol

L'hiver - une couleur blanche au sol

Printemps - des couleurs vives (incluant du vert) au sol

Pour vos fenêtres, vous allez devoir ajouter pour chaque sommet de notre terrain et un attribut de couleur.

Intégrer un Quadtree

Nous allons appliquer un quadtree sur notre terrain. Une fois ce travail réalisé, nous allons afficher ce terrain dans chaque fenêtre.

A l'aide du clavier, nous allons limiter la profondeur de lecture de notre arbre pour chaque fenêtre. Ainsi nous avons réalisé notre première solution de simplification (LOD) !

Mettre en place une solution de multi-résolution progressive en fonction de la distance du sommet à la camera (le réaliser en temps réel).

Attention, à correctement trianguler votre surface.

Un gestionnaire de niveau de détails

Maintenant, nous allons nous intéresser un peu plus au niveau de détails. Charger un objet 3D à différentes résolutions.

Afficher l'objet sur la scène, et permettre le déplacement de celui-ci. En fonction de la distance avec la camera, afficher un modèle plus ou moins simplifié.

Un gestionnaire de scène

Mettre en place un graphe de scène pour structurer les objets placés sur votre terrain. Prévoir pour chaque feuille un ensemble de paramètres spécifique pour chaque fenêtre. Ecrire les deux méthodes de parcours de votre graphe de scène

- mettre à jour les éléments de la scène (changements de saisons, etc.)
- afficher les éléments de la scène (avec gestion des niveaux de détails)

Compte rendu

Présenter vos fonctionnalités

Expliquer votre démarche de développement.

Expliquer comment vous vous y prendriez pour les parties bonus.

Bonus

- Saisons

Ajouter de la pluie en automne et de la neige en hiver (animation de particules). Proposer une approche pour accumuler les particules, et les transformer en surface (rivière, tas de neige).

- Niveaux de détails

Proposer une méthode de simplification d'objets 3D ainsi qu'une structure de donnée adaptée. Appliquer cette méthode à notre chargement d'objet 3D.

- Afficher une scène infinie

Elaborer la meilleure structure de données, ou la meilleure composition de structure de données pour réaliser une scène infinie. Appliquer des méthodes de multi-résolution, imposteurs, etc. en fonction de la position de la caméra.

- Garder un rendu temps réel

Proposer des optimisations pour mettre à jour et afficher votre scène à 30 fps, 60 fps, 120 fps.