ENSSAT
LANNION

Algorithmique distribuée
--
contenu multimédia

What the duck !

Situation actuelle

L'archive disponible contient le code de démonstration permettant de débuter le projet. Le projet actuel permet le déplacement libre à la souris et au clavier avec pour but la recherche des canards jouant des sons. Lorsque la caméra est suffisament proche d'un canard, le canard s'affiche et arrête la lecture du son. Le but est de trouver les deux canards présents dans la scène.

Le but du projet est d'améliorer l'expérience utilisateur et développer un mode multi-utilisateurs. La réalisation de ce projet est individuelle.

Études à mener

1. Afin de vous approprier le programme, vous devez dans un premier temps adapter le cône directif et l'atténuation du son des canards afin de simplifier la recherche des canards. (cf la doc d'openAL : http://openal.org/documentation/openal-1.1-specification.pdf)

2. Vous devez également prévoir le chargement d'un fichier de configuration permettant de décrire le nombre de canard, le son joué par les canards, leurs positions et orientations à la place de l'actuelle description de scène codée dans la classe scène. Pour cela il faut prendre la liberté de modifier la classe scène, soit par l'ajout du fichier de configuration au constructeur, soit par l'ajout de méthodes permettant de modifier les membres de la scène).

   Exemple : pour la ligne **d:data/sound.wav:-10:0:-5:0:45:12** le canard (**d** pour duck) jouera le son **data/sound.wav** à partir de la position 3d (**-10,0,-5**) et avec une orientation autour de l'axe X de **0**°, de l'axe Y de **45**° et de l'axe Z de **12**°

3. Pour débuter le processus multijoueur, vous devez créer deux applications, un logiciel serveur (appelé serveur dans la suite) et un logiciel client (appelé client dans la suite). La communication client/serveur est bidirectionnelle :

   - le serveur se charge de lire la configuration et de répondre aux requêtes des clients

   - le client initie la communication par une requête de connexion à laquelle le serveur répond par un ou plusieurs message(s) d'initialisation de canard

   - le client informe le serveur lorsqu'il a trouvé un canard par l'envoi d'un message spécifique

   - lorsque tous les canards ont été trouvés par le joueur, le serveur informe le client par l'envoi d'un message de fin de jeu

4. Passage au multijoueur :

    - le client informe le serveur de ses déplacements et de la découverte d'un canard

    - le serveur informe tous les clients de la découverte des canards par les autres joueurs

    - le message de fin de jeu est renvoyé par le serveur à tous les joueurs

5. Adaptions possible quand tout fonctionne :

    - ajouter une indication de distance vers le plus proche canard pour aider le joueur à trouver les canard

    - utiliser la bibliothèque libjsoncpp pour charger le fichier de configuration

    - ajouter une boussole permettant au joueur de mieux se repérer

    - proposer une ou plusieurs modification(s) du protocole de communication afin de limiter les possibilités de triche entre clients

    - proposer la possibilité d'avoir plusieurs objets (autre que des canards) ayant des propriétés acoustiques différentes (directivité, atténuation, effet, etc.)

Livrables

Vous devez fournir le code de votre projet (le code complet et le Makefile permettant de compiler votre projet : si ça ne compile pas, ça ne note pas…).

Vous devez également fournir un rapport en pdf de 5 pages :

- 1 page de garde
- 1 page contenant diagramme de communication de votre projet
- 2 pages description du travail réalisé (ce qui est fait et les éléments spécifiques pour la compréhension, mais également ce qui n'est pas terminé ou ce qui reste à faire)
- 1 page reprenant les tests effectués (quel tests, comment reproduire le test, quel résultat attendu, quel résultat obtenu).

L'ensemble doit être déposé dans une archive .zip sur moodle.