GMINT317 - Moteurs de jeux – TP2
Timers

Rémi Ronfard  *remi.ronfard@inria.fr*  *http://team-inria.imagine*

Objectifs

- Etudier et modifer la boucle (game loop) de rendu du TP1.
- Afficher la scène dans 4 fenêtres différentes, avec 4 timers
- Animer la scène à vitesse constante.
- Bonus : étudier et modifier l'organisation logicielle des différentes classes.

Question1 : Ouvrir, compiler et exécuter le projet qt TP2.pro. Tester les interactions utilisateurs (touches A,Z,E,Q,S,W,X). Décrire ce que fait chacune.

A quoi servent les paramètres etat, ss, rotX et rotY ? Créer une classe Camera contenant tous ces paramètres, et modifier la classe GameWindow pour qu'elle possède un pointeur vers sa caméra.

Question 2 : Modifier la classe GameWindow pour qu'elle effectue le rendu de la scène à intervalles fixes (et non pas à chaque modification). On utilisera pour cela la classe QTimer qui permet d'associer l'exécution d'une méthode de la classe GameWindow (renderNow) à un signal (timeout) donné par le timer.

```
timer->connect(timer, SIGNAL(timeout()),this, SLOT(renderNow()));
```

Modifier le constructeur de la classe GameWindow pour qu'il prenne en paramètre la fréquence de mise à jour (frames per second).

Question 3 : Créer quatre instances de la classe GameWindow, avec des fréquences de mise à jour différentes de 120 fps, 60 fps, 30 fps et 1 fps.

Par souci de simplicité, les quatre instances partageront la meme caméra et recevront les memes événements utilisateurs (touches clavier).

Question 4 : Ajouter une fonction d'animation qui fait tourner la scène à vitesse constante autour de l'axe Y, sans intervention de l'utilisateur. Utiliser pour cela une touche clavier en mode « on/off » : lorsque l'utilisateur tape la touche « C » une première fois la scène se met en mouvement ; lorsque l'utilisateur tape la même touche « C » une seconde fois, la scène s'immobilise.

Question 5 : Utiliser les touches « P » et « M » pour modifier la fréquence d'échantillonnage de chaque instance de votre moteur de jeu. Par exemple, on pourra multiplier la fréquence par 2 à l'aide de la touche « P » et la diviser par 2 à l'aide de la touche « M ». Que se passe-t-il lorsque la fréquence

Bonus : pourquoi la classe GameWindow hérite-t-elle de OpenGLWindow ? Quels sont les avantages et inconvénients de cet héritage ? Comment serait-il possible d'éviter cet héritage ?