Rapport de TP – GMIN317 – Moteur de Jeu

Nom : El Aouaji
Prénom : Karim
M2 IMAGINA
Numéro étudiant : 20110471

Question 1 : Interactions clavier

- A :Rotation sur l'axe vertical vers le haut
- E : Rotation verticale vers le bas
- Q : Rotation horizontale vers la gauche
- D : Rotation horizontale vers la droite
- Z : Zoom
- S : Dézoom
- W : Modifie le type d'affichage de la carte
- X : Change la carte parmi les 3 disponibles

- etat : Mode d'affichage de la scène
- ss : Coefficient multipliant la matrice courante pour simuler le zoom/dézoom
- rotX, rotY : Angles de rotation de la caméra respectivement sur l'axe X et Y

Classe camera.h :

```
#ifndef CAMERA
#define CAMERA

class Camera
{
public :
    int getEtat(){
        return etat;
    }
    float getRotX(){
        return rotX;
    }
    float getRotY(){
        return rotY;
    }
    float getSS(){
        return ss;
    }
    void setEtat(int i){
        etat = i;
    }
    void setRotX(float f){
        rotX = f;
    }
    void setRotY(float f){
        rotY = f;
    }
    void setSS(float f){
        ss = f;
    }
private :
    int etat = 0;
    float rotX = -45.0;
    float rotY = -45.0;
```

```
    float ss = 1.0f;
};

#endif // CAMERA
```

Dans gamewindow.h :

```
public:
    void setCamera(Camera*);
private:
    Camera* cam;
```

Question 2 :

Dans gamewindow.h :

```
public:
    GameWindow(int);
    int getFrequency();
private:
    int frequency;
QTimer* timer;
```

Dans gamewindow.cpp :

```
GameWindow::GameWindow(int freq)
{
    frequency = freq;
    rotating = true;
    timer = new QTimer(this);
    timer->connect(timer, SIGNAL(timeout()), this, SLOT(renderNow()));
    timer->start(1000/getFrequency());
}

int GameWindow::getFrequency(){
    return frequency;
}
```

Question 4 :

Dans camera.h :

```
void sceneRotation(bool rotating){
        if(rotating){
            setRotY(getRotY() - 0.10f);
        }
    }
```

Dans gamewindow.h :

```
public:
    void updateId(int);
private:
    static bool rotating;
```

Dans gamewindow.cpp :

```
bool GameWindow::rotating;
GameWindow::GameWindow(int freq)
{
    rotating = true;
}
void GameWindow::updateId(int newId){
    id = newId;
}
void GameWindow::render()
```

```
{
    cam->sceneRotation(rotating);
    timer->start(1000/getFrequency());
}


void GameWindow::keyPressEvent(QKeyEvent *event)
{
    switch(event->key())
    {
    case 'C':
        if(id==0){
            if(rotating){
                rotating = false;
            }
            else{
                rotating = true;
            }
        }
        break;
}
```

On crée un attribut *rotating* pour gérer le fonctionnement de la rotation. Il faut qu'il ne soit modifiable que par une seule instance de *gamewindow*, sinon l'évènement permettant de la gérer va être déclenché autant de fois qu'il y a d'instances. On lui attribue donc le mot clé *static*.

Pour cela, on identifie également chaque instance avec l'attribut *id*. Finalement, il n'y a que l'instance 0 (la première), qui gère l'évènement d'activation/désactivation de la rotation.

Question 5 :
Dans gamewindow.cpp :

```
void GameWindow::keyPressEvent(QKeyEvent *event)
{
    switch(event->key())
    {
    case 'P':
        frequency *= 2;
        break;
    case 'M':
        frequency /= 2;
        break;
}
```