Compte-rendu TP3

Simulation du passage des saisons

HMIN317 - Moteur de jeux

BOYER Benoît

M2 IMAGINA - Septembre 2017

Table des matières

0.1 Question 1

0.1.1 Synchroniser les fenêtres entre elles

La synchronisation des fenêtres se fera via les signaux de Qt, et en mettant le timer des fenêtres à la même cadence.

0.2 Question 2

0.2.1 Création des saisons

On va reprendre les éléments du TP précédent. Pour le printemps et l'été, nous aurons juste à gérer la lumière, et pour l'automne et l'hiver, nous allons devoir rajouter des particules.

Création du printemps et de l'été

La première étape est de refaire le terrain, on va repartir sur un terrain plat et simple, et modifier le matériau pour changer sa couleur. Pour cela nous allons modifier le shader pour qu'il prenne en paramètre un Vecteur4D qui va contenir la couleur à appliquer au shader lors de la saison.

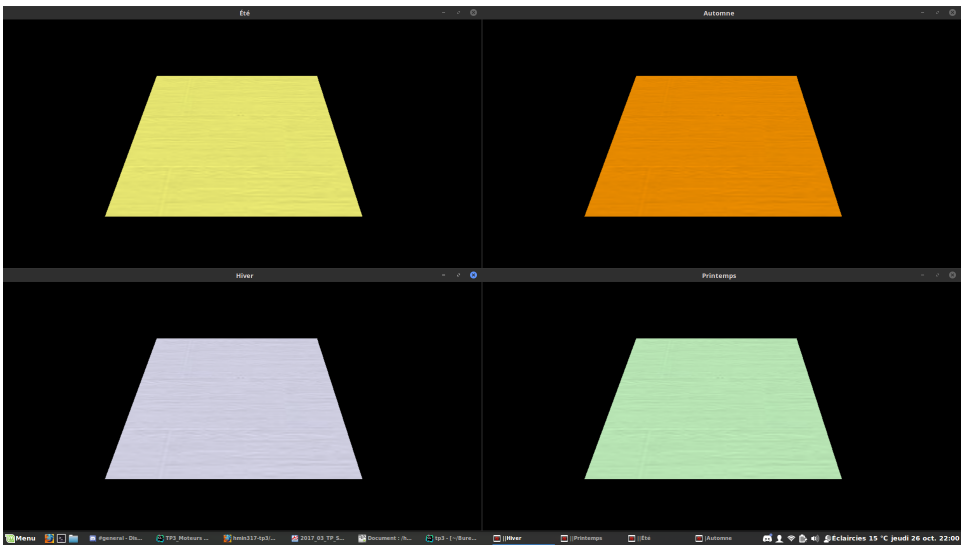

FIGURE 1 – Les saisons de base

0.2.2 Changement de la saison

Pour cela, nous allons utiliser un timer qu'on va appliquer aux 4 fenêtres :

```
71    MainWidget* widgetP = new MainWidget(nullptr, 60, PRINTEMPS)
      ;
```

```
72      MainWidget* widgetE = new MainWidget(nullptr, 60, ETE);
73      MainWidget* widgetA = new MainWidget(nullptr, 60, AUTOMNE);
74      MainWidget* widgetH = new MainWidget(nullptr, 60, HIVER);
75
76      widgetP->show();
77      widgetE->show();
78      widgetA->show();
79      widgetH->show();
80
81      QTimer* seasonTimer = new QTimer;
82
83      QObject::connect(seasonTimer, SIGNAL(timeout()), widgetP,
        SLOT(nextSeason()));
84      QObject::connect(seasonTimer, SIGNAL(timeout()), widgetE,
        SLOT(nextSeason()));
85      QObject::connect(seasonTimer, SIGNAL(timeout()), widgetA,
        SLOT(nextSeason()));
86      QObject::connect(seasonTimer, SIGNAL(timeout()), widgetH,
        SLOT(nextSeason()));
87
88      seasonTimer->start(5000);
```

Listing 1 – Déclaration des fenêtres et raccordement au timer

Ensuite, il suffit de se référencer aux fonctions types `public slots` (ici nextSeason()). qui vont être appelées à chaque fois que le timer est écoulé. On effectuera le changement de paramètres dans cette fonction.

0.3 Questions bonus

0.3.1 Accumulation de particules

On pourrait utiliser un plan parallèle et détecter le point d'impact de la particule pour faire monter le sommet, et faire une accumulation de ce plan parallèle avec la même texture (ou couleur) pour simuler cette accumulation.

0.3.2 Localisation des effets climatiques

Le rendu différent peut être effectué en jouant sur quelques éléments, comme la lumière, ou l'affichage via les shaders, et plus précisément le contraste ou la luminosité, on peut faire un rendu plus terne pour rendre plus triste et donc évoquer encore plus l'automne par exemple.

0.3.3 Utilisation d'OpenMP

Pour OpenMP, on peut s'en servir facilement pour paralléliser les événements (ce qui permet d'améliorer la vitesse de rendu), comme par exemple à attribuer une fenêtre à un thread différent, ou lors du parcours de `for` en parallélisant les calculs.