

Compte rendu de moteur de jeu – TP3

Harbulot Xavier

Synchronisation des fenêtres :

Pour réaliser la synchronisation des fenêtres, j'ai réalisé une nouvelle classe calendrier.

Voici le header du calendrier, qui sera implémenté dans le main.cpp :

```
class Calendrier : public QObject
{
    Q_OBJECT
public:
    explicit Calendrier(QObject *parent = nullptr);
    void timer();

signals:
    void signalSaison();

public slots:
    void calendrierChangeSaison();
};
```

Et son code source :

```
void Calendrier::calendrierChangeSaison() {
    std::cout << "Changement de saison" << std::endl;
    emit signalSaison();
}

void Calendrier::timer() {
    QTimer *timer = new QTimer(this);
    connect(timer, SIGNAL(timeout()), this,
    SLOT(calendrierChangeSaison()));
    timer->start(5000);
}
```

Ensuite, les quatre widgets que nous avons créés pour le TP 2 sont connectés au calendrier.

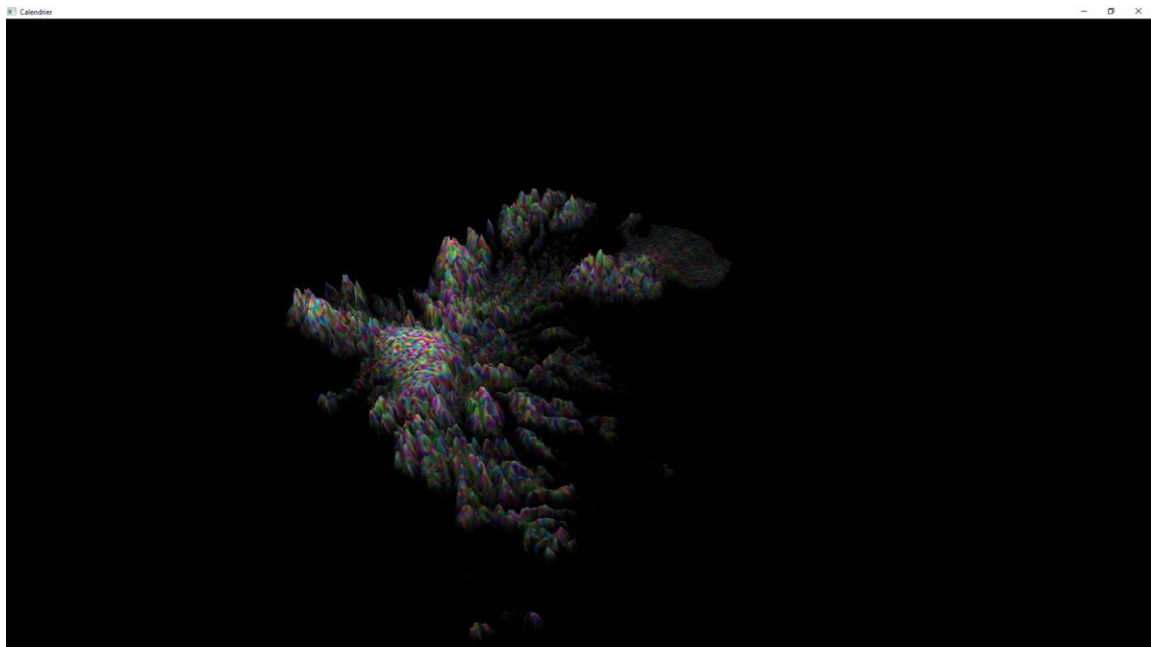
```
QObject::connect(&calendrier, SIGNAL(signalSaison()), &widget1,
    SLOT(changeSaison()));
```

Chaque widget possède désormais un slot changeSaison() et un int saison. La fonction changeSaison incrémente simplement cet entier.

Simuler le changement de saison :

J'ai rajouté un QVector3D à la structure VertexData pour tenter de rajouter une couleur au terrain. J'ai réussi à l'initialiser correctement, cependant je n'arrivais jamais à modifier les valeurs des couleurs.

Voici le résultat que j'ai obtenu en mettant une couleur aléatoire en chaque point :

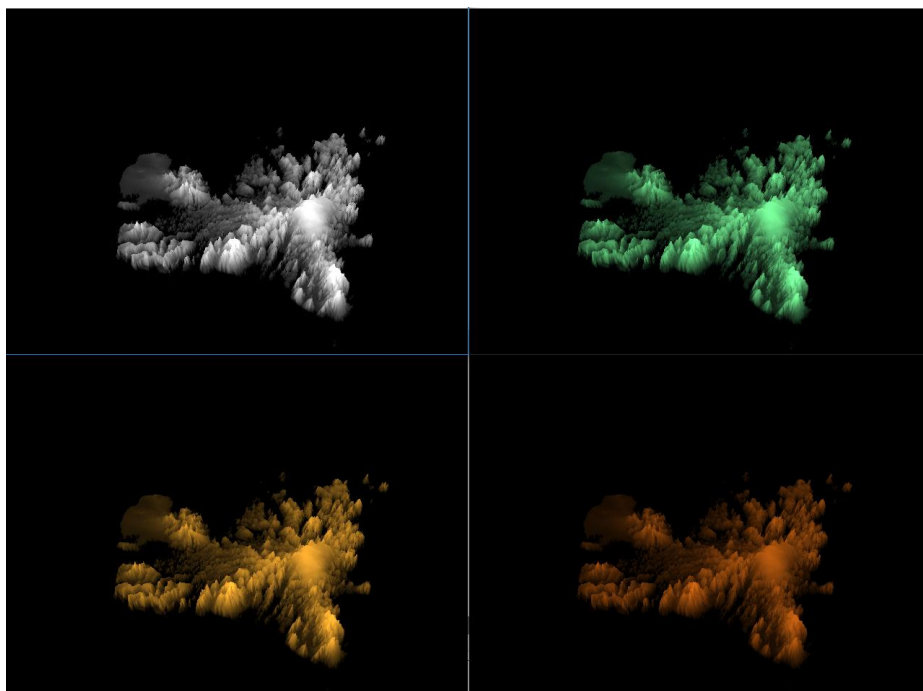


J'ai donc opté pour une autre solution : j'ai créé un uniform dans le shader et avec une formule différente, je change l'entièreté de la couleur du terrain. Le code en question est le suivant :

```
uniform vec3 v_color;
void main() {
    gl_FragColor = texture2D(texture, v_texcoord) * vec4(v_color, 1.0) *
0.005;
}
```

Ensuite, j'applique une couleur différente pour chaque saison avec un simple switch, dans la fonction paintGL ;

Voici le résultat final :



De haut en bas et de gauche à droite : Hiver, printemps, été, automne.

Pour les particules, je n'ai pas pu les gérer.

J'ai tenté de créer une classe pour les particules et une classe pour l'émetteur de particules. J'ai voulu donner une texture à mes particules mais je n'arrive qu'à faire crasher l'application lorsque je tente leurs créations. Je pensais faire en sorte qu'update réactualise la position de chaque particule dans l'émetteur, en parcourant une liste.