

LABORATORIO 3

Nel laboratorio 3, ho iniziato creando un nuovo materiale cercando di rendere il legno il più realistico possibile. Ecco i parametri utilizzati per il materiale:

```
glm::vec3 wood_ambient = { 0.2, 0.1, 0.0 }, wood_diffuse = { 0.6, 0.3, 0.0 }, wood_specular = { 0.3, 0.15, 0.0 }; GLfloat wood shininess = 10.0f;
```

Successivamente ho implementato l'animazione di un height field mesh creando v_wave.glsl nel codice ho calcolato l'altezza del vertice nel seguente modo e l'ho rimpiazzata alla coordinata y del punto position:

```
height = a * sin(omega * t + 10 * position.x) * sin(omega * t + 10 * position.z) ;  
position.y = height;  
gl_Position = P * V * M * position;
```

Per quanto riguarda il Toon shading, ho scritto lo "shader vertex" v_toon.glsl. e f_toon.glsl. Il toon shading è una tecnica di rendering grafico che simula un aspetto cartonesco o fumettistico, suddividendo i valori di illuminazione in un numero limitato di gradazioni piatte, senza sfumature continue come nel rendering tradizionale.

Ho assegnato un colore in base all'intensità dell'illuminazione

```
if (intensity > 0.95)  
    color = a;  
else if (intensity > 0.5)  
    color = b;  
else if (intensity > 0.25)  
    color = c;  
else if (intensity > 0.15)  
    color = d;  
else  
    color = e;
```

Per implementare il pan orizzontale ho scritto le funzioni moveCameraLeft e moveCameraRight

Le due funzioni sostanzialmente calcolano il vettore di direzione da ViewSetup.position a ViewSetup.target e calcolano il vettore di spostamento (slide vector) prendendo il prodotto cross tra il vettore di direzione e il ViewSetup.upVector, e quindi lo moltiplicano per CAMERA TRASLATION_SPEED. Questo vettore di spostamento determina la quantità e la direzione con cui la telecamera si muoverà quando va a sinistra.

Aggiorna la posizione della telecamera e il punto di vista (target) sottraendo o aggiungendo il vettore di spostamento da entrambi, spostando quindi la telecamera a sinistra o a destra.

Si sono infine gestite le trasformazioni di traslazione, rotazione e scalatura dei singoli oggetti in scena rispetto al sistema di riferimento WCS o OCS (World e Object Coordinate Systems).

Se la condizione `TransformMode == WCS` è vera, significa che la modalità di trasformazione è impostata su "WCS" (World Coordinate System). In tal caso, viene creato un `rotation_matrix` utilizzando la funzione `glm::rotate`. Questa matrice rappresenta la rotazione attorno all'asse specificato da `rotation_vector` con l'angolo `angle` specificato. La matrice di trasformazione del modello (M) dell'oggetto selezionato viene quindi moltiplicata per `rotation_matrix`.

Se la condizione `TransformMode == WCS` è falsa, significa che la modalità di trasformazione è impostata su "OCS ". In questo caso, viene utilizzata la funzione `glm::rotate` per applicare una rotazione alla matrice di trasformazione del modello dell'oggetto selezionato (`objects[selected_obj].M`). La rotazione viene eseguita attorno all'asse specificato da `rotation_vector` con l'angolo `angle`.