Fondamenti di Computer Graphics LM

Lab 6 - SHADER part I: Lighting, Shading & Texture Mapping

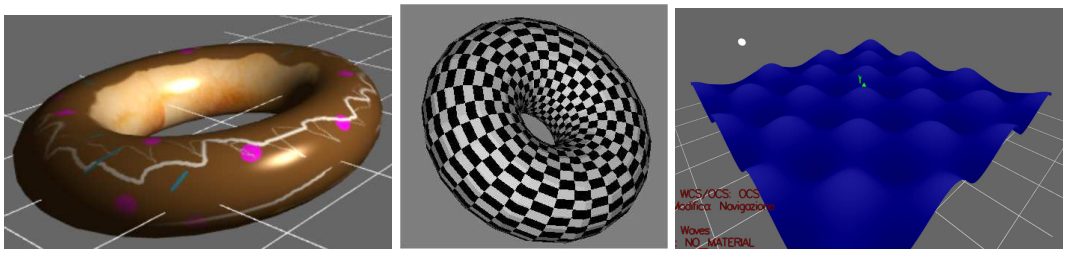


Figure 1: (sinistra) Applicazione di una texture image sulla mesh output della tassellazione del toro; (centro) l'applicazione di procedural texture alla mesh toro; wave motion di una mesh piano (destra).

Scaricare i file necessari dalla pagina WEB del docente. L'archivio contiene un semplice programma per la gestione di lighting, shading e texture 2D con OpenGL e GLSL, compilare ed eseguire il programma fornito. Il programma è basato sul codice dell'esercitazione 3 ma la resa è basata su shaders. Se lo si ritiene opportuno e utile si possono ereditare dalla esercitazione 3, se svolta, i tool di navigazione in scena e gestione delle trasformazioni, questo upgrade sarà valutato come opzionale.

I tasti frecce dx e sx permettono di selezionare uno tra i seguenti modelli:

- mesh plane con 2D texture image creata in modo procedurale (checkboard) ai vertici;
- mesh cubo con 2D texture image caricata da file e associata alle facce. Legge il formato file immagine *nomefile.jpg*; la function utilizzata per leggere le immagini puó gestire molti altri tipi di file immagine sia in 24bit RGB che in 32 bit RGBA.
- mesh toro creato attraverso la tassellazione di una superficie parametrica su un dominio parametrico. Con i tasti W/w e N/n si incrementa/decrementa il numero di avvolgimenti W/w e il numero di vertici per ogni avvolgimento N/n del raggio maggiore e del raggio minore.
- mesh sfera flat, caricata da file .obj e creata con normali alle facce e parametrizzazione associata ai vertici;
- mesh sfera smooth, caricata da file .obj e creata con normali ai vertici e parametrizzazione associata ai vertici.

Estendere il programma inserendo la gestione, mediante opportuni vertex e fragment shader, delle seguenti funzionalità:

1. **Lighting**: Permettere lo spostamento (traslazione) interattivo della luce posizionale/direzionale in scena.
2. **Shading**: Al toro è associato un materiale ed é illuminato in modalità shading Gouraud (mediante l'attivazione degli shaders *v_gouraud.glsl*, *f_gouraud.glsl*). Sperimentare la modalità shading Blinn, che approssima Phong shading con l'half vector H, rendendo attivi gli shaders *v_blinn.glsl*, *f_blinn.glsl*. Realizzare la modalità shading Phong realizzando gli shaders *v_phong.glsl* e *f_phong.glsl*;

3. **Texture mapping 2D** del toro con immagine letta da file di formato *nomefile.jpg* mediante gli shaders *v_texture.glsl* e *f_texture.glsl*

4. **Texture mapping 2D + Shading** Realizzare gli shaders *v_texture_phong.glsl* e *f_texture_phong.glsl* per combinare l'effetto shading Phong con la texture image sulla mesh toro; effetto illustrato in Fig. 1 sinistra.

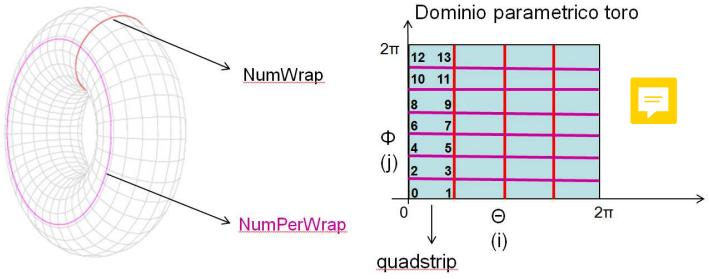
5. **Procedural mapping** basato su un procedimento algoritmico a piacere sul toro.

6. **Wave motion**: Si crei l'animazione di un height field mesh (oggetto mesh *GridPlane.obj* contenuto nella directory Mesh) modificando la posizione dei vertici in un vertex shader *v_wave.glsl* (effetto da ottenere mostrato in Fig. 1 destra). Utilizzare la variabile elapsed time t, passata da applicazione al vertex shader, per riprodurre il moto ondoso ottenuto con la sola modifica della coordinata y mediante la formula:
$$v_y = a \sin(\omega t + 10 v_x) \sin(\omega t + 10 v_z)\,,$$
dove l'ampiezza dell'oscillazione a e la frequenza ω siano scelte a piacere, es. $a = 0.1$ e $\omega = 0.001$.

7. OPZIONALE: **Toon shading** Realizzare gli shaders *v_toon.glsl* e *f_toon.glsl* per la resa non-fotorealistica nota comunemente come "Toon shading". Un esempio è illustrato in Fig. 2.

Figure 2: Rendering Non-fotorealistico

Osservazione: Tassellazione e parametrizzazione della superficie toro Si ricorda che il toro ha la seguente rappresentazione parametrica $\mathbf{S}(\theta, \phi)$ con $\theta, \phi \in [0; 2\pi]$:



$$\begin{aligned} x(\theta, \phi) &= \sin(\theta)(R + r\cos(\phi)) \\ y(\theta, \phi) &= \sin(\phi) r \\ z(\theta, \phi) &= \cos(\theta)(R + r\cos(\phi)) \end{aligned}$$

Il vettore normale è definito come

$$\mathbf{n}(\theta,\phi) \quad = \quad \mathbf{S}_\theta(\theta,\phi) \times \mathbf{S}_\phi(\theta,\phi)$$

con derivate parziali

$$\mathbf{S}_\theta(\theta,\phi) = \begin{bmatrix} dx/d\theta \\ dy/d\theta \\ dz/d\theta \end{bmatrix} \quad = \quad \begin{bmatrix} \cos(\theta)(R + r\cos(\phi)) \\ 0 \\ -\sin(\theta)(R + r\cos(\phi)) \end{bmatrix}$$

$$\mathbf{S}_\phi(\theta,\phi) = \begin{bmatrix} dx/d\phi \\ dy/d\phi \\ dz/d\phi \end{bmatrix} \quad = \quad \begin{bmatrix} -r\sin(\phi)\sin(\theta) \\ r\cos(\phi) \\ -r\sin(\phi)\cos(\theta) \end{bmatrix}$$