# **Fondamenti di Computer Graphics LM**

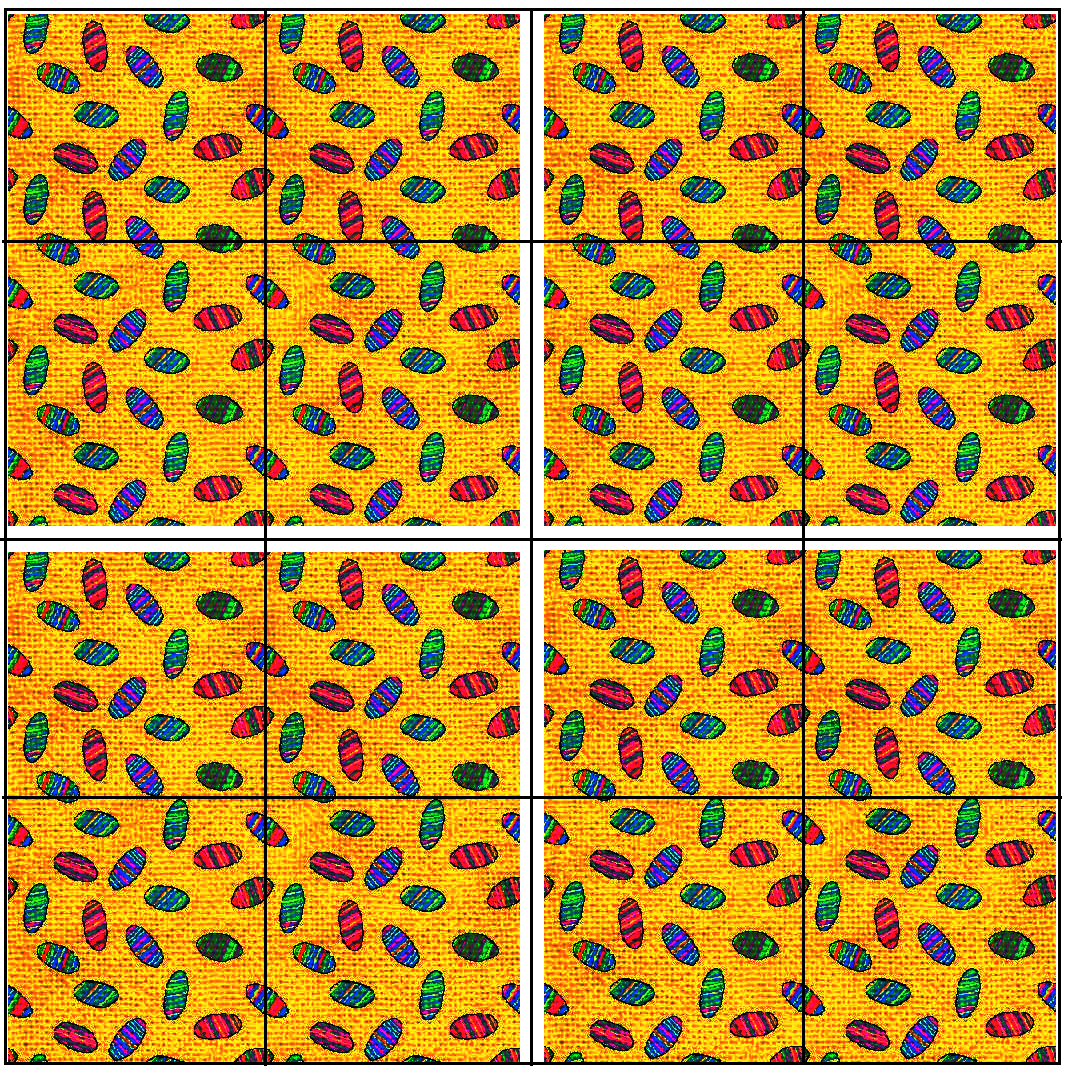
Lab 5 – Texture Mapping

Mattia Fucili

**Texture Mapping 2D**

Per applicare le texture è stato necessario fare dei ragionamenti sul mapping della texture stessa nel toro. I conti vengono fatti nel metodo putVert() e il ragionamento è spiegato di seguito.

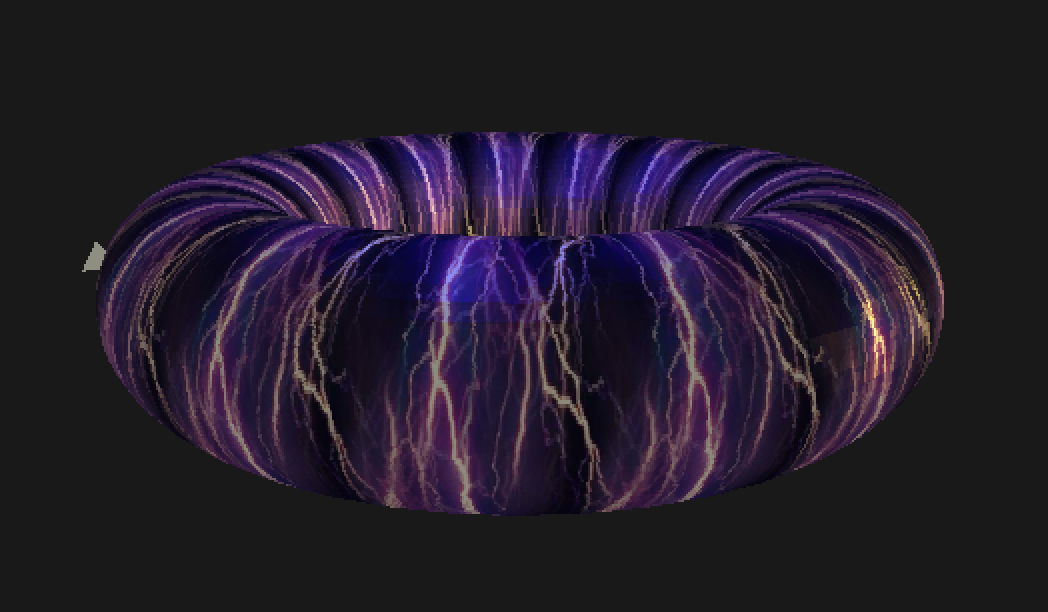
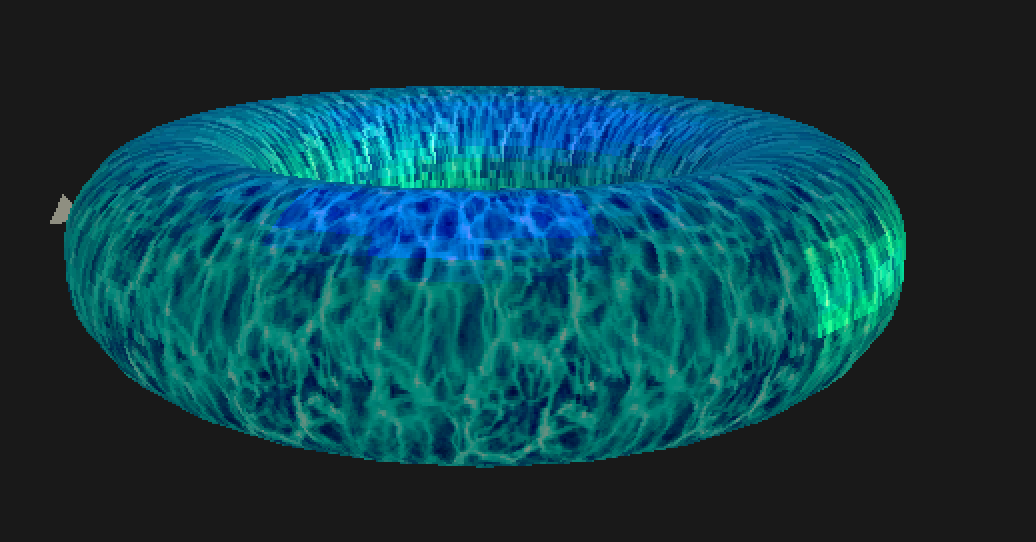
Facciamo l’esempio di voler ripetere la texture 2 volte in orizzontale e 2 volte in verticale, si avrà allora una situazione come sotto:



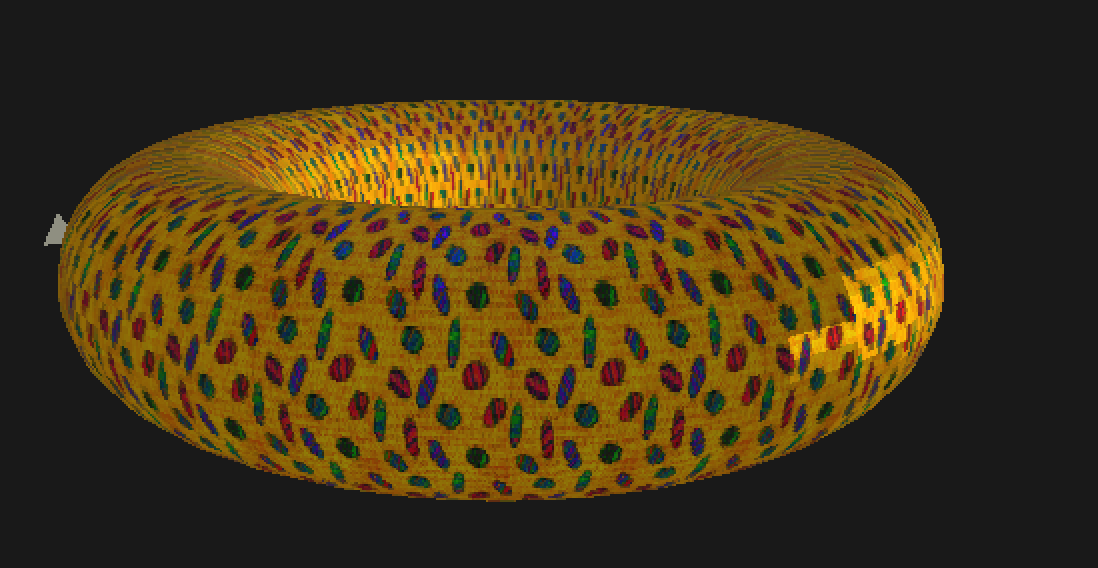
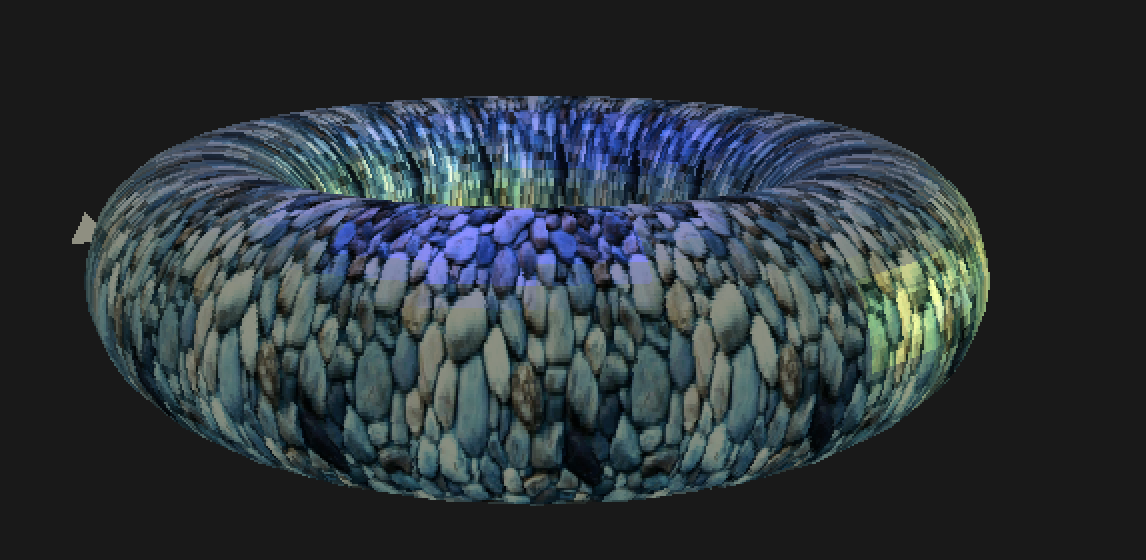
Il mapping consiste nel dividere ogni vertice che viene passato al metodo putVert() per il numero totale di NumWrap, nel caso la coordinata sia la , e per il numero totale di NumPerWraps, nel caso in cui la coordinata sia la . Dopodiché occorre moltiplicare le componenti per il numero di ripetizioni della texture, in questo caso 2.

Se ad esempio abbiamo 4 NumWrap e 4 NumPerWraps si ottiene:

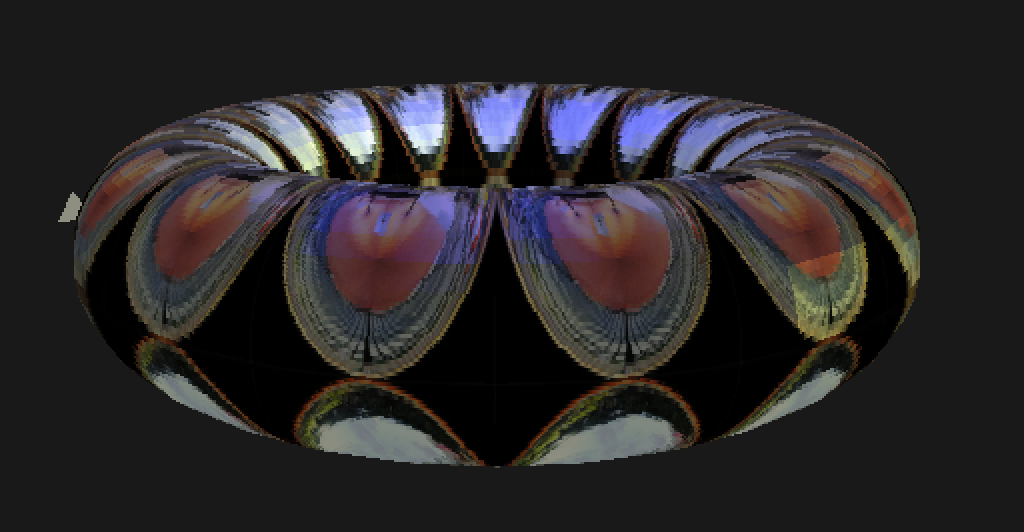
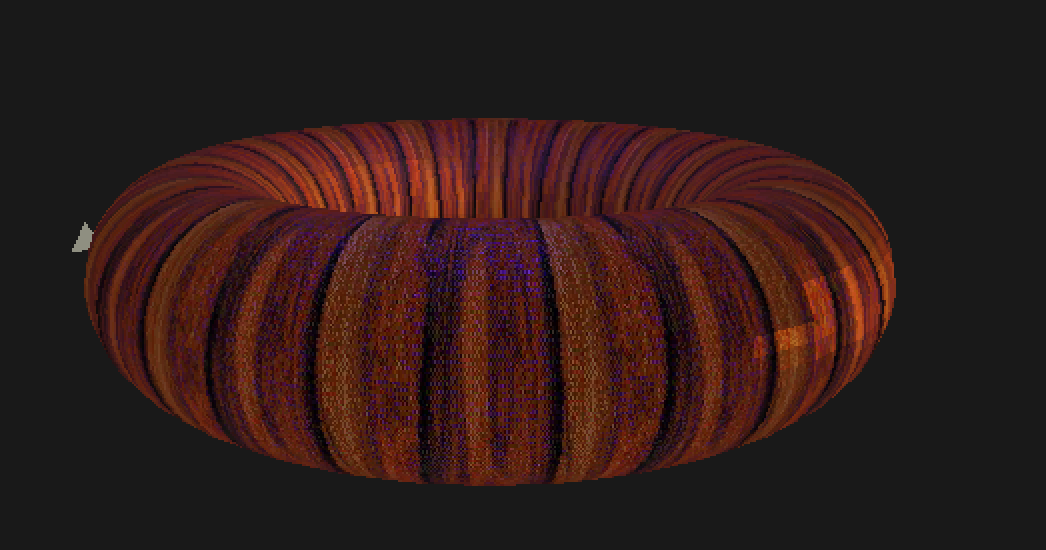
Il risultato del mapping è il seguente:

fulmini acqua

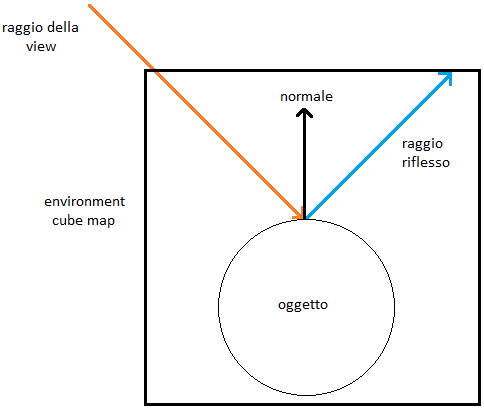
caramelle sassi

** **

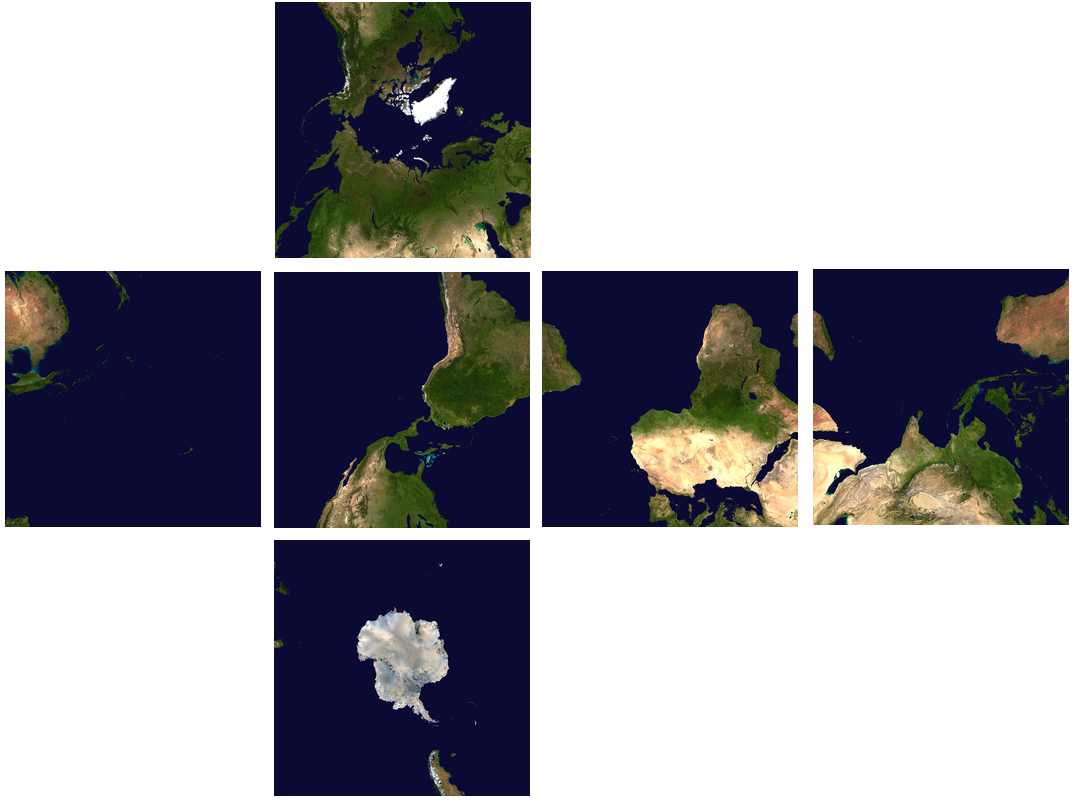
sfere legno

**Environment Mapping**

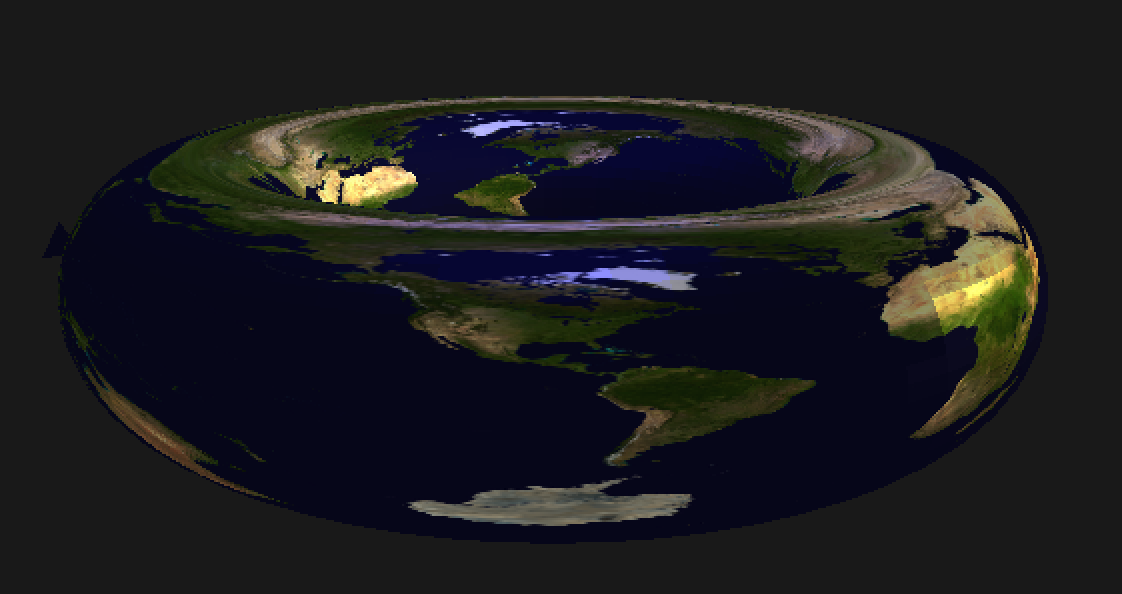
Per quanto riguarda l’environment mapping è stato sia implementato il cube mapping sia lo sphere mapping. L’environment mapping funziona assegnando ad ogni punto dell’oggetto il colore che viene catturato dal raggio riflesso, come mostrato in figura:



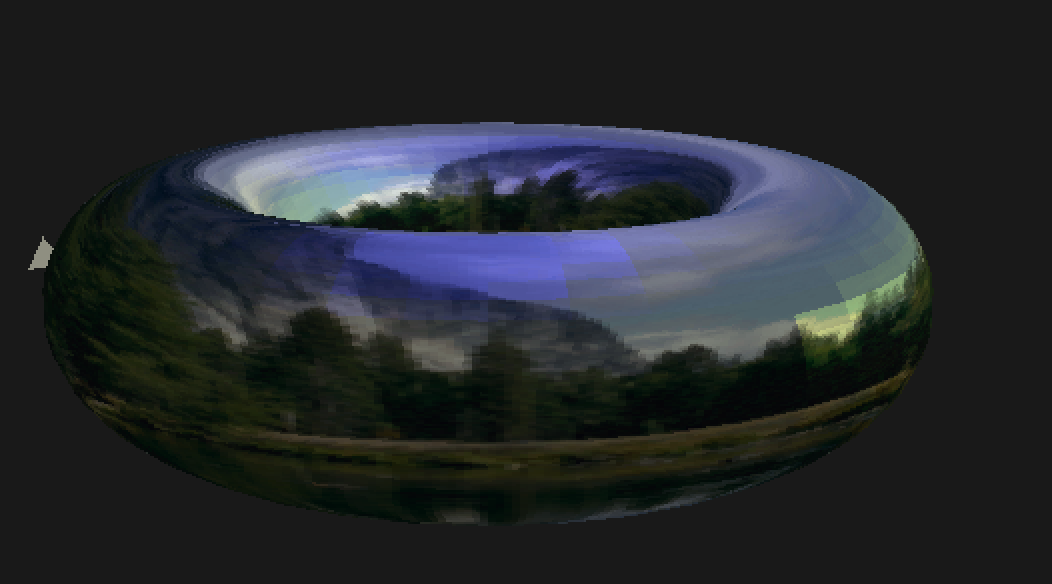
Per ottenere il cube mapping è stato necessario fornire un immagine per ogni faccia del cubo. Un esempio è il seguente:



Ottenendo quindi:



Per fare lo sphere mapping è stato più semplice in quanto è stato necessario caricare una sola immagine a 360°, con risultato:



**Procedural Mapping**

Il mapping procedurale viene fatto in modo tale da ottenere una decorazione a tema “Italia”. Il codice può essere visto nel metodo initProceduralTexture(). Il risultato è il seguente:

