Il laboratorio 3 si è concentrato sull’utilizzo alcune mesh poligonali di oggetti tridimensionali già fornite e la creazione/modifica di fragment e vertex shaders per permettere lighting e shading della scena e aggiungere anche il movimento. Nella prima parte si è presa visione delle funzionalità già esistenti, con alcuni test pratici per prendere familiarità con l’ambiente. Nelle parti successive, invece, le richieste di modifica/implementazione sono state:

* Parte 1:
  + Calcolo delle normali ai vertici per il Phong e il Gouraud Shading (a differenza di quello Flat già implementato);
  + Creazione di un materiale diverso da quelli forniti;
  + Aggiungere un’animazione di tipo “onda” all’height field mesh GridPlane, inizialmente piatto;
  + Implementare lo shading aggiuntivo Toon Shading per la resa non fotorealistica;
* Parte 2: aggiunta dello spostamento orizzontale, sia di camera sia di punto di riferimento, alle funzionalità della trackball;
* Parte 3: aggiunta delle trasformazioni di traslazione, rotazione e scala per ogni oggetto (relativamente sia al WCS sia all’OCS).

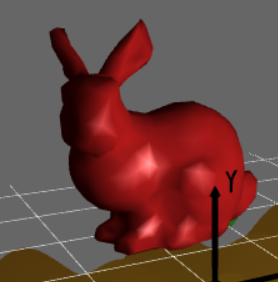
1. Shading

1.1. Calcolo delle normali per il Phong Shading

In generale, le normali ai vertici o alle facce vengono calcolate, se non presenti, al caricamento delle informazioni dal file .obj. Nel caso di normali alle facce, quindi nel caso di Flat Shading, ci si limita a calcolare il valore di una normale (di solito ottenuta dai tre vertici del triangolo) e ad utilizzarla per tutti i pixel della faccia stessa. Il risultato sarà quello di avere la stessa luce (e quindi lo stesso colore per tutti i pixel) su tutta la singola faccia, ottenendo un effetto “faccettato” anche un oggetto in realtà liscio.

Di conseguenza, è stato modificato l’algoritmo e sono state calcolate le normali di ogni vertice di ogni faccia. In particolare, per ogni vertice si effettua la somma delle normali di tutti i triangoli che incidono su quel vertice, per poi normalizzare i valori.

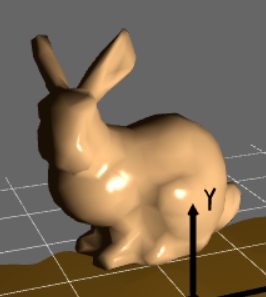
Immagine che contiene oggetto da esterni

Descrizione generata automaticamenteImmagine che contiene rosso

Descrizione generata automaticamente

(Nell’immagine, in ordine: Flat Shading, Gouraud e Phong Shading)

1.2. Creazione di un materiale diverso

In questo caso, è stato necessario creare un materiale diverso da quelli forniti, specificando la componente ambientale (rose\_gold\_ambient), ovvero la componente che simula la luce ambiente indiretta, la componente diffusiva (rose\_gold\_diffuse), che fornisce la frazione di luce che viene riflessa in tutte le direzioni e quindi il colore dell’oggetto, la componente speculare (rose\_gold\_specular), che indica quando il materiale sia “lucido” e, infine, il l’esponente di riflessione speculare (rose\_gold\_shiness - visivamente, modifica l’ampiezza dello spot di riflessione della luce).

1.3. Animazione wave

Per introdurre un’animazione ad “onda” al piano presente nella scena, bisogna agire, come suggerito dalla consegna, modificando la posizione dei punti del piano con una funzione che dipende da un tempo *t* passata da applicazione a shader, in particolare al vertex shader, che si occupa di processare i punti e non i fragment. Per ottenere l’effetto voluto, la funzione è una sinusoidale del tipo:

La nuova posizione è calcolata solo per la coordinata y, poiché il movimento ondulatorio è voluto in quella direzione. Inoltre, nello stesso shader, sono calcolate anche le trasformazioni sull’oggetto (con le matrici di proiezione, vista e modello) e il colore punto.

