Assignment 4

Full Solar System

Affiliation: Mathematics, CS

student ID: 2017313260

Name: Lee Jae Min

It started with texphong example and last assignment. Same algorithm with last assignment, create_sphere_vertices, update_index_buffer. It has same variable.

1. Data Structure (Additions, modified)

1.1 planet.h (Addition file)

a) struct planet

It has radius, rotation speed, revolution speed, au(distance from sun. If it is dwarf, distance from parent planet.), texture path, normal texture path, model matrix, dwaf_au(if it is dwarf, parents planet's au. If it is main planet, set 0), GLuint value of texture and normal texture.

b) struct ring

It has au(distance between parent planet with sun), scale(ring size), revolution speed, model matrix, texture path, alpha texture path, Gluing value of texture and alpha texture.

Both structure has setVal(), update_model() function, setVal function is initialize of elements. update_model function is update model matrix with time values.

Declaration of each planet(struct planet). and setVal() in setPlanets(). Then, use create_solar(), make container of planets (vector). Of course, ring has similar algorithm. (create_ring())

1.2 main.cpp (modified)

"vector<planet> solar", "vector<ring> rings" are container of planets and rings.

longitude, latitude, unit_theta, unit_phi, b_rotate, t_gap, t_passed same with last assignment.

ring_tess is number of tess for make ring vertices.

create_ring_vertices() is make vertices of ring, ring_index_buffer() is make indices of ring.

update() function is update mode, light data(shading).

In render() function, rendering planets, rings and texturing, dump mapping, alpha blending.

In user_init() function, create sphere, ring vertices and indices. And create texture each texture path, then save in GLuint ~_texture (include normal, alpha).

mouse(), motion() same with last assignment. keyboard() same with texphong example.

2.2 texphong.frag (Addition from texphong example)

It has uniform value planet_cnt, check_norm, is_alpha. planet_cnt is value for check that planet is sun to except shading sun. check_norm is value for check that it has normal texture(it can be bump mapping). is_alphs is value for check that it has alpha texture(it can be alpha blending).

And it has vectors for bump mapping.

2. Algorithm

2.1 render()

use solar, rings and iteration method, each planets texturing and update model matrix(update_model()). If it can be bump mapping, load normal texture too, calculate fragColor with new normal vector in texphong.frag.

If it can be alpha blending, load alpha texture too, input alpha value(texture's rgb) in fragColor.a

planet_cnt count start from 0, 1, 2... And rerun render(), initialize to 0. Then, Sun has planet_cnt=0.

2.2 texphong.frag

If planet_cnt is 0, This is Sun, don't calculate phong().

For bump mapping, calculate TBN matrix. It's in assignment pdf. Load rgb from normal texture. value of texture(rgb) = vec/2 + 0.5, So we need vector, normal vector is 2*rgb-1. Then, multiplicate with TBN matrix, and normalize. It will be new normal vector.

If it can be bump mapping, check_norm will be 1. Then, calculate fragColor with new normal vector. (phong())

If it can be alpha blending, load alpha texture, and get Kd(iKd_alpha). Then put iKd_alpha.r in fragColor.a. Because alpha texture is grayscale image, so r=g=b, 0~1. Then It brighter, It more transparent.

Follow the mode, fragColor will be changed.

3. Discussions

It is important to understand about normal vector and tex_coord(mapping on image).