TAIPEI TECH

進階C語言實務

(20230523)

Homework 4 - BVH

Reference

- **Fast-BVH**:
  - Author:
    - Acknowledge: **Brandon Pelfrey**
  - Purpose:
    - BVH structure construction
    - Object-Orientation Class design
    - Simple ray-tracing test
  - Link:
    - https://github.com/brandonpelfrey/Fast-BVH
- **Modified Fast-BVH**:
  - Download from **iStudy at NTUT**.

Extension of Fast-BVH

- Download the package source code from **iStudy**
- Homework items:
  - Giving a rendered sample image from the sample code (debug is necessary)
  - Add some additional object (e.g. Cube, Start, Pyramid, Pikachu…)
  - Comparisons for different types:
    - Scene sizes
    - Number of objects
  - Documents:
    - Address the comments for each/important code section
    - Address the critical section for computing loading

Same code

- Main program:

```
int main(int argc, char **argv) {

    // Create a million spheres packed in the space of a cube
    const unsigned int N = 1000;
    vector<Object*> objects;
    printf("Constructing %d spheres...\n", N);
    for(size_t i=0; i<N; ++i) {
        objects.push_back(new Sphere(randVector3(), 0.05f));
    }

    // Compute a BVH for this object set
    BVH bvh(&objects);

    // Allocate space for some image pixels
    const unsigned int width=1024, height=1024;
    float* pixels = new float[width*height*3];

    // Create a camera from position and focus point
    Vector3 camera_position(1.6, 1.3, 1.6);
    Vector3 camera_focus(0,0,0);
    Vector3 camera_up(0,1,0);

    // Camera tangent space
    Vector3 camera_dir = normalize(camera_focus - camera_position);
    Vector3 camera_u = normalize(camera_dir ^ camera_up);
    Vector3 camera_v = normalize(camera_u ^ camera_dir);
```

- The submitted folder needs include the C/C++ program file (only *.h & *.c/*.cpp).
- To present the results in word or pdf format, it includes snapshot of console display results and program descriptions, etc.