

2016.1 Human Media Multicore Computing
Project 3: CUDA/OpenMP Ray-Tracing (Team Project)
(Deadline : June 13th 11:59pm)

Submission method : eClass

- All students need to submit a zip file containing (i) final report, (ii) source code, (iii) executable code, (iv) README.txt
- The leader of each team needs to submit DEMO video file in addition to above (i)~(iv) items.

Step 1. make a team

- team size : 1~4 members

Step 2. Implementing two versions of Ray-Tracing that utilizes CUDA and OpenMP

- Look at the partial CUDA code of ray-tracing of random spheres available on our class webpage. and modify the code for your purpose.
- You may assume the simplest form of Ray tracing that renders a scene with only spheres.
- Program input :
 - (i) [option]: 0 means using CUDA, 1~16 means using OpenMP with 1~16 threads
 - (ii) [output filename]
- Program output :
 - (i) print ray-tracing processing time of your program using OpenMP or CUDA
 - (ii) generate image file (format: .ppm or .bmp, image size: 2048X2048) that shows the rendering result

Execution example 1) > a.out 0 result.ppm
CUDA ray tracing: 0.15 sec
[result.ppm] was generated.

Execution example 2) > a.out 8 result.ppm
OpenMP (8 threads) ray tracing: 0.41 sec
[result.ppm] was generated.

Step 3. write a final report (pdf) that includes

- project title, member list (name and student id)
- execution : describe (i) execution environment (OS type, CPU type, graphics card/GPU type, memory size) (ii) how to compile, (iii) how to execute
- your group's contribution (describe exactly what your group actually did for this project)
- entire source code and detailed explanation on the OpenMP code and CUDA code
- other implementation issues (describe how you implemented)
- **program output results including screen capture pictures.**
- experimental results : measuring the performance (execution time) of your OpenMP/CUDA implementation and your single threaded CPU implementation. show the performance results and screen capture of output results.
- conclusion : summarize your project result

Step 4. submission (to eClass)

- final report (should include a list of team members with student ID#)
- source code files
- executable file
- README.txt file (describe (i) execution environment (OS type, CPU type, GPU type, memory size) (ii) how to compile, (iii) how to execute)
- DEMO Video file in .mp4 or .avi format (Only team leader needs to submit this) : video-record processes of (a) program compilation, (b) execution and (iii) results including execution times and rendering images