

## ***EE/CprE/SE 491 WEEKLY REPORT 1***

***January 25 – February 6***

***Group number: 18***

***Project title: GPGPU Parallelization of Memworld***

***Client &/Advisor: Dr. Wymore***

***Team Members/Role:***

- ***William Blanchard, Parralelization Lead***
- ***Mason DeClercq, Team Lead***
- ***Jay Edwards, Documentation Lead***
- ***Cristofer Medina Lopez, Integration Lead***
- ***Dalton Rederick, Communications Lead***
- ***Collin Reeves, Game Development Lead***

### ○ **Weekly Summary**

This week, the group set up a meeting and met with Dr. Wymore to discuss the project we were working on and the expectations he had for us. After the meeting, we talked about what we wanted to work on for the week. We decided that researching projects that could be using the same technology and researching the libraries that we would be using would be best. We met with each other at the end of the week in order to discuss what we found during research. We decided on using OpenCL for the GPGPU programming because it was the most portable. The Git repository was also set up using the code provided by Dr. Wymore.

○ **Past week accomplishments**

• Wil Blanchard:

- Compared OpenCL and CUDA to each other to make sure we will move forward with the proper tools. Took a quick glance at ray tracing source engine to get a base understanding of what we are parallelizing

• Mason DeClercq :

- Worked on porting over the Git repository that Dr. Wymore provided to the Git repository for this course.
- Research
  - Found different types of GPGPU frameworks that could be possibly used.
  - Found different industry projects that implement voxel technologies.
  - Found that OpenCL would be the best choice for us to have a portable application.
  - The industry projects will help in understanding why this project could be important and what types of applications that it could be used for in the future.
- Set up a development environment on my windows machine. This was not a simple task because most of the technologies that are being used are Linux based.
- Started looking through the Memworld code in order to determine where the GPGPU optimization would be located.

• Jay Edwards:

- Went through past works done with ray tracing and OpenGL to remind myself how it works. Looked up resources on those topics to relearn what I forgot and sent links to the group so they can utilize it as well.

• Cristofer Medina Lopez:

- Setup environment on computer to run the clients memworld application. Looked into OpenGL to understand how it works and gain a better understanding of the client's application. Did research on OpenCL to understand the toolset better and how to utilize it in the future.

• Dalton Rederick:

- Researched into projects similar to our own. Found a thesis document which seems to be similar to what we are working on (voxel based rendering for high definition objects). Currently reading through the document (200 pages) and, should it seem worthwhile, will contact the writer to have them as a point of contact.
- Set up a shared drive for storing files for the project. Wrote up notes for meetings

• Collin Reeves:

- Set up a development environment that Mason found, found an updated guide to utilize some additional applications.

○ **Pending issues**

- There are no pending issues at this time

○ **Individual contributions**

<b><u>NAME</u></b>	<b><u>Individual Contributions</u></b> <i>(Quick list of contributions. This should be short.)</i>	<b><u>Hours this week</u></b>	<b><u>HOURS cumulative</u></b>
Wil Blanchard	Comparison of CUDA/OpenCL	3	3
Mason DeClercq	Set up Git repository, preliminary research, set up development environment, investigated memworld code	8	8
Jay Edwards	Researched OpenGL and Ray tracing	3	3
Cristofer Medina Lopez	Researching tools expected to be used for project.	4	4
Dalton Rederick	Looked into other projects similar to our own, wrote up meeting notes, organized folders, installed dev environment	4.5	4.5
Collin Reeves	Set up the dev environment that Mason found.	2	2

○ **Plans for the upcoming week**

- Wil Blanchard: Searching through OpenCL API to understand how to use it. Research into the current Memworld code. Stretch goal to develop some example applications of OpenCL to demonstrate that knowledge.
- Mason DeClercq : I plan on working on determining where the GPGPU parallelization should occur within Memworld. Try to learn the OpenGL and OpenCL languages.
- Jay Edwards: Look into the differences between OpenCL and OpenGL. Go through Memworld to see it's current implementation.
- Cristofer Medina Lopez: Continue researching OpenGL and OpenCL to understand how to work with the tools in the future.
- Collin Reeves: Researching OpelCL and OpenGL, making sure the dev environment is correctly set up.
- Dalton Rederick: Research OpenCL/GL. Test out dev environment. Read through the thesis and look into other projects.

- **Summary of weekly advisor meeting**

This week's advisory meeting was an introduction to the problem at hand and a discussion of the relevant concepts. After exchanging introductions, we let Professor Wymore introduce us to the software we are working with in its current state and elaborate on the problem description. Right now, the software is in a state where a base raytracer is being used to render a test room with multiple colors, but it can only do so at a low resolution or else the engine slows down by a considerable amount. We were told that the primary goal of the project as of now is to parallelize the process using GPGPU libraries for a quicker ray tracing process. This led to a short discussion on what GPGPU library would be best to look into for this project, the result of which was the decision that OpenCL would be our focus for now. After this, we talked about where we would go after we speed up the engine. In the end, this included expanding on the process to make a visually better output. The end of the meeting included an introduction to rendering concepts, such as the difference between ray tracing and ray casting and the base knowledge of how each pixel on a screen can be given a value.