

Initial Plan – Raycasting Game Engine for Perceptual Awareness

Author: Simon Titcomb, 1221969

Supervisor: Dr F.C. Langbein

Module: CM3203 One Semester Individual Project, 40 Credits

Project Description:

Traditionally rendering for real-time applications such as games have used rasterisation as the rendering method. This gives a good approximation of physically correct environments however due to the constraint of real-time, some rendering effects such as lighting leave a bit to be desired.

Raycasting is a technique that was traditionally used to render highly physically accurate detailed scenes offline. As hardware performance has increased dramatically over the last few years and indeed lower level APIs have been introduced to access this more powerful hardware, it is now possible to render scenes using raycasting in more than just an offline capacity. It may even be possible to achieve raycasting in a real-time manner such that it can potentially replace rasterisation as the 'standard' rendering technique.

To this end I am aiming to replace the rendering engine of an open source game engine with a GPU based raycasting engine. The aim is to investigate whether the rendering engine of an existing game engine can be replaced with an experimental rendering engine to still achieve real-time rendering (on a powerful GPU).

Raycasting is the basic algorithm type we need for ongoing research into human perception. The overall aim is to replace the rendering engine with a rendering engine for human perceptual awareness and evaluate various different non-optically correct rendering modes and judge their effect on human perception. The basic version of this project will simply be about getting the raycasting engine to work and document how to change it further and study the impact on performance. A more advanced and challenging version of the project is to include some of the perceptual rendering effects in the rendering engine.

I will be looking at implementing this in either nVidia CUDA framework or just pure OpenGL. In order to decide which method is better I will be researching into existing techniques and engines to see the pros and cons of either method.

Project Aims and Objectives:

My aims for this project are:

- **Create Rendering Engine:** I shall need to create a rendering engine that uses the technique of raycasting instead of traditional rasterisation
- **Increase Performance of Engine:** As is the core goal of this project, I am aiming to achieve real-time or close to real-time performance in the rendering engine to make it suitable for applications such as games

- **Performance Analysis:** Look into the performance increases between different iterations of the engine and see where improvements could be made in an effort to increase the real-time performance
- **Advanced Techniques:** Add into the rendering engine advanced techniques such as ray tracing and additional shaders for techniques such as fog

Work Plan:

I have arranged weekly meetings with my supervisor because, as you will see below, I plan to have weekly or fortnightly objectives. Meaning I should have significant progress to report on every week.

Additionally at the end of every iteration of the engine I will be recording performance in an effort to see what techniques yield the best results and where more optimisations can be made.

Week	Objectives
01/02/2016 – 08/02/2016	Research into the different methods of implementing raycasting and some examples of engines that use these methods (ie. CUDA vs OpenGL)
08/02/2016 – 15/02/2016	Implement a basic raycasting program based on the research done from the previous week. Basic means here that we are not expecting real-time performance or any optimisations and we are only rendering a simple hard-coded scene.
15/02/2016 – 29/02/2016	Perform optimisations and use other techniques to achieve real-time rendering on a basic hard-coded scene.
29/02/2016 – 14/03/2016	Modify the existing basic raycasting program from earlier to load more complex scenes.
14/03/2016 – 21/03/2016	Modify the existing real-time raycasting program to load more complex scenes based on information learned the week before.
21/03/2016 – 04/04/2016	Implement more advanced rendering techniques as described above. (eg. Shader techniques for fog or add ray tracing to the engine)
04/04/2016 – 18/04/2016	Write up final year report.
18/04/2016 – 06/05/2016	Review final year report and hand in for deadline on the 06/05/2016.