



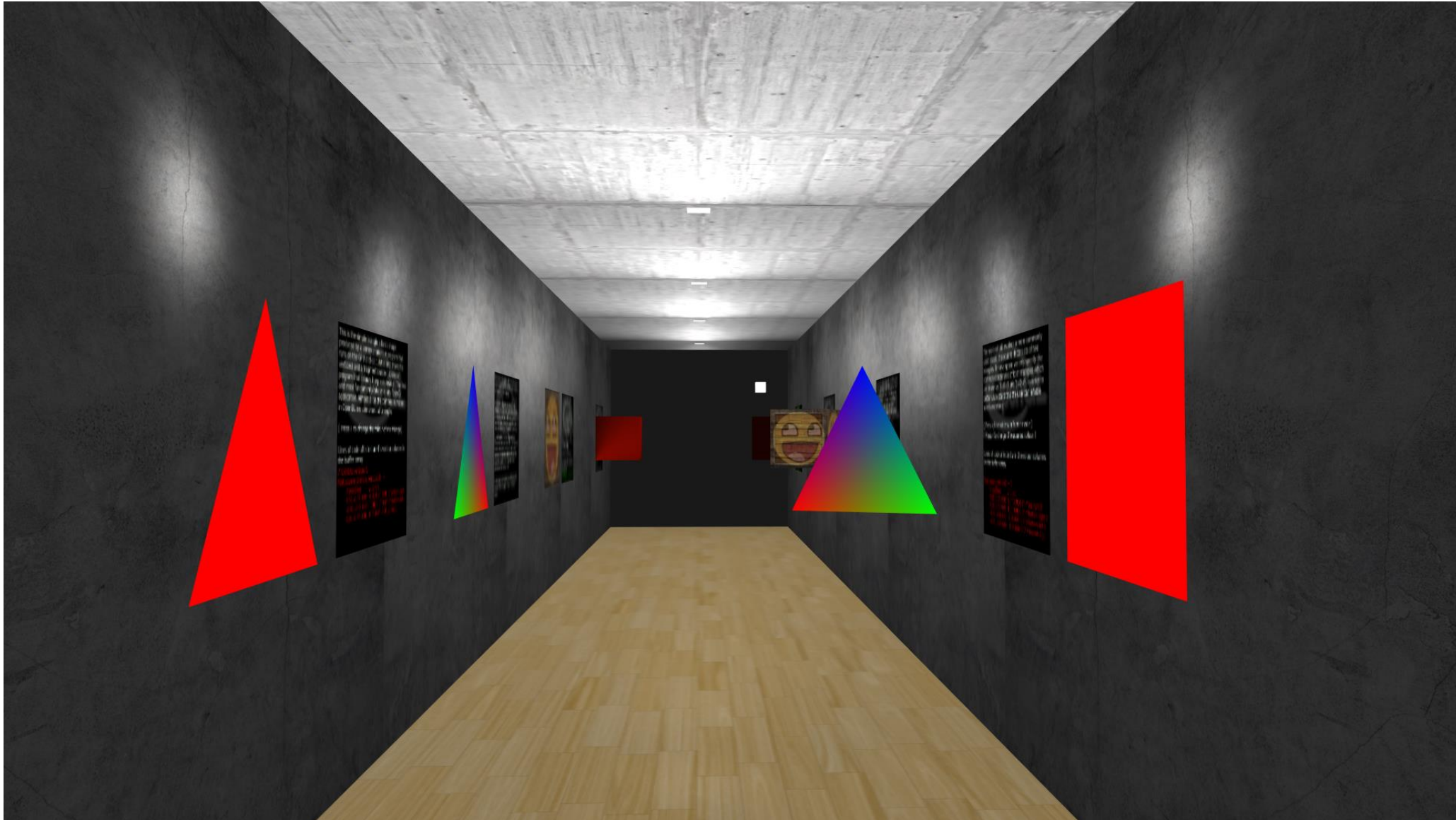
Ray Tracing on FPGA

NIKOLAS KOXENOGLOU – DIPLOMA THESIS PROPOSAL

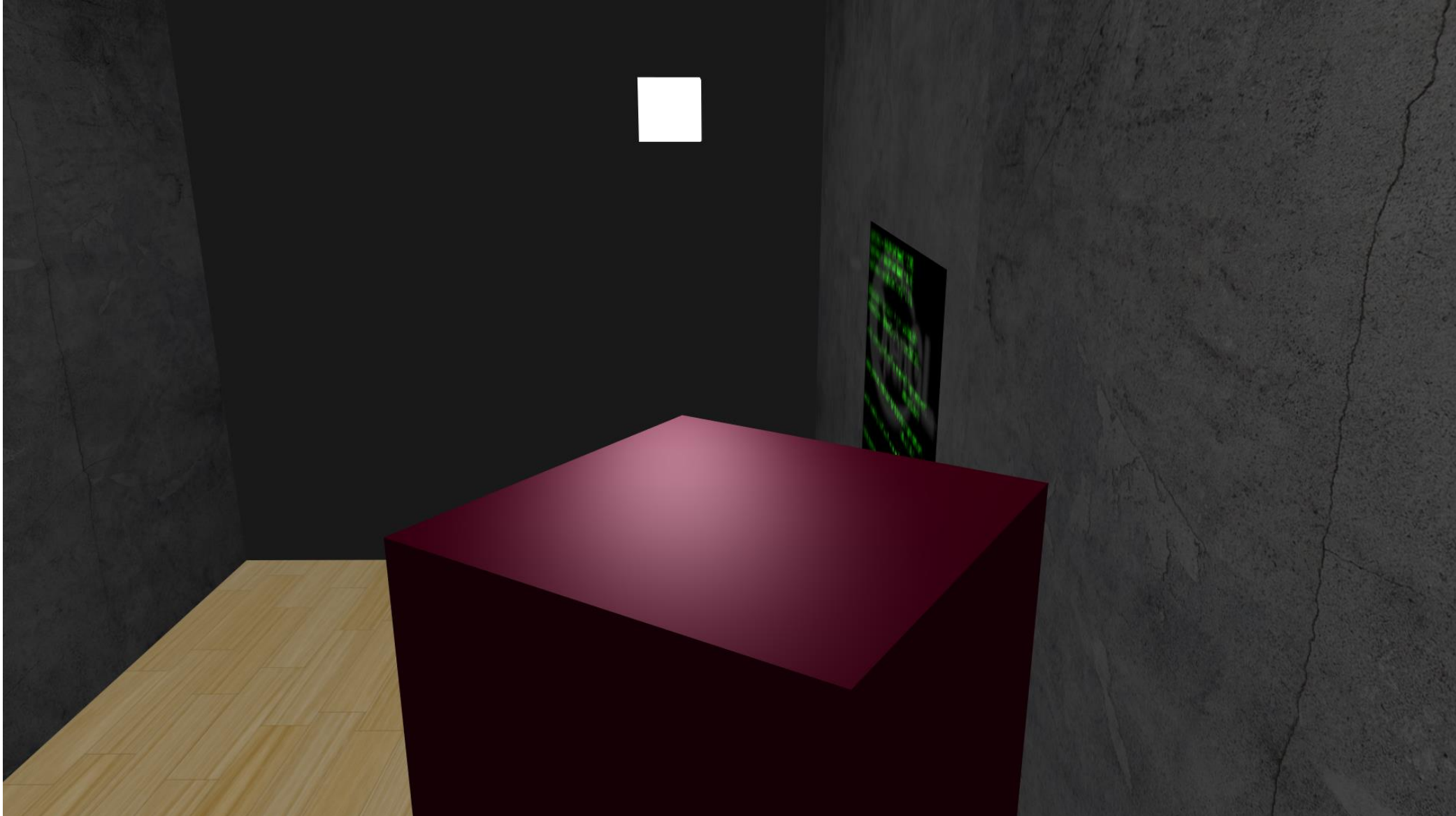
My OpenGL Project

- ▶ An idea to avoid a final exam ..
- ▶ Turned into an idea for my diploma thesis
- ▶ It involves me using OpenGL to understand the maths that use to produce graphics
- ▶ A museum styled corridor that displays “exhibits” which display all the things we learned in the OpenGL course

A small peak



A small peak



My OpenGL Project

- ▶ As you can observe this scene although beautiful still lacks fancy graphics and shadows etc.
- ▶ This a very simple OpenGL implementation
- ▶ In creating this project I became interested in working with graphics
- ▶ It is an interesting field with many problems that need solutions

Graphics pipeline

- ▶ OpenGL is colouring pixels with a processes called rasterization
- ▶ It projects triangles on the view plane using perspective projection and then fills them
- ▶ GPUs are build to rasterize
- ▶ Rasterization can only get you so far
 - ▶ Greater graphical fidelity at the cost of greater complexity (production time and cost increase)
 - ▶ Last major graphics upgrade: Crisis (2007)
 - ▶ But can it run Crisis ??

History

- ▶ The first Ray Tracing scene took 2 weeks to render at Bell Labs 1978
 - ▶ “The Compleat Angler” – Turner Whitted
 - ▶ First true Ray Tracing
- ▶ A good looking but still lacking scene
- ▶ Today it is still a very demanding process
- ▶ Used in game-engines and more

What is Ray Tracing

- ▶ A rendering method based on Ray Casting (Wolfenstein 3D - 1992)
- ▶ Ray Tracing adds recursion to a casted ray
 - ▶ Additional rays generate from the point of intersection for various effects
- ▶ Adds depth and complexity to a scene by tracing the path of light or shadow sources in order to evaluate the colour of a pixel
- ▶ Each pixel can be calculated independently from other pixels

Why Ray Tracing

- ▶ It is an interesting method that produces stunning images and scenes
- ▶ Implementing it has great computational cost
- ▶ It has good parallelizable potential
- ▶ A more sensible method for creating

What Problems Can be Solved

- ▶ Nvidia is using a combination of Ray Tracing and rasterization
 - ▶ Small number of rays
 - ▶ Using de-noising hardware to fix the image
- ▶ The main problem Ray Tracing solves are graphical FX rendering problems
- ▶ There is a case to be made by utilising a form of the algorithm in other problems as well
- ▶ A real or near real-time ray tracing implementation will save a lot of time in various work environments that utilize it

What I want to accomplish

- ▶ I want to further familiarise myself with Digital Design using FPGAs
- ▶ Work on an idea that excites me
- ▶ Develop, implement and optimize an FPGA project
- ▶ This will represent what I can accomplish when properly working on a project

What is Planned

- ▶ The plan is to define the problem to be solved with more detail
- ▶ Build a testing environment along with golden samples to verify the progress and correctness of the project
- ▶ Develop the accelerator main architecture in a modular fashion
- ▶ Build a base implementation with no optimizations

Sources

- ▶ <https://www.cs.unc.edu/~rademach/xroads-RT/RTarticle.html>
- ▶ [https://en.wikipedia.org/wiki/Ray_tracing_\(graphics\)](https://en.wikipedia.org/wiki/Ray_tracing_(graphics))