

CRT - Development Journal

Hamza Haiken (hamza.haiken@heig-vd.ch)
Prof. Pier Donini (pier.donini@heig-vd.ch)

B.A. Thesis, HEIG-VD 2014-2015

Contents

1	09.03.2015 - 15.03.2015	3
2	16.03.2015 - 22.03.2015	3
3	23.03.2015 - 30.03.2015	3
4	01.04.2015 - 07.04.2015	3
5	08.04.2015 - 15.04.2015	3
6	16.04.2015 - 22.04.2015	4

1 09.03.2015 - 15.03.2015

- GitHub repository¹ created
 - `.gitignore` file tailored for project
- Specifications document
- Maven project created via NetBeans
 - Some dependencies
 - Default license header set (GPL3)
- Logo design
 - First simple attempt
 - Created CRT effect as a Photoshop action and patterns
 - Multiple sizes

2 16.03.2015 - 22.03.2015

- \LaTeX thesis template
- Python script for building thesis
- UML class diagrams for rendering process, ANTLR4 structure
- Some sequence diagrams
- Paragraphs on the rendering process and grammar parsing
- Reworked the grammar

3 23.03.2015 - 30.03.2015

- Established basic ray tracing skeleton
- Refactored code from old prototype
- Changed structure according to new decisions made with supervisor
- Added Box primitive

4 01.04.2015 - 07.04.2015

- Work on CSG operations:
 - Union OK
 - Intersection OK
 - Difference has buggy normals but works
- Change in `Hit` class model to indicate entry and exit points of `Ray`
- Corrections to Sphere primitive intersection and normals
- Rendering classes refactoring
- Rendering UML diagram split in two, added cardinalities and fixed a few aspects

5 08.04.2015 - 15.04.2015

- Wrote the grammar from scratch
- Wrote test cases

¹<https://github.com/Tenchi2xh/CRT>

- Started work on compiler:
 - General framework
 - Scope and variables
 - Scoped variable resolution
 - Variable reference pointers
 - Literal variable assignments
 - List variable assignments
 - List access
 - Function calls for color and vector types

6 16.04.2015 - 22.04.2015

- Corrections
- Typography improvements
- Section about ANTLR, the grammar, BNF listing and operators table
- Section about ray tracing history, workings, process
- Citing image sources
- Rewrote how the camera system works
 - Removed awkward 4×4 matrix system (more appropriate for rasterisation)
 - System with up and right vectors
 - Left-handed coordinate system
 - Corrected camera projections
- Rewrote how lights the work
 - Multiple light types
 - Each light type has to give its own direction vector and distance. This lead to correct a bug where a light source is inside a primitive, because the ray intersection point is exactly on the surface, it tries to go through anyway instead of stopping
- Much work on CSG
- Animation test
- Drew diagrams with TikZ for light explanations
- Experimented with global illumination
- Adapted depth-of-field calculations to new camera system