
Requirements Specification

for

Rapid LDraw Parts Definition

Version 1.0

Prepared by Anthony Namba, Theron Anderson

PSU, Fall Capstone 2018

11-22-2018

Table of Contents

Introduction	4
Purpose	4
Intended Audience and Reading Suggestions	4
Project Scope	4
References	4
Overall Description	5
Product Perspective	5
Product Features	5
Minimum Viable Product	5
Usable Product	6
Stretch Goal	7
User Classes and Characteristics	7
Design and Implementation Constraints	7
User Documentation	7
Assumptions and Dependencies	7
Other Nonfunctional Requirements	8
Performance Requirements	8
Software Quality Attributes	8
Milestones	8
Design and Implementation Plan Completion	8
Alpha	9
Beta	9
Release	9
Disclaimers	9

Revision History

Date	Reason For Changes	Version
11-19-2018	Creation	0.1
11-21-2018	Revision	0.2
11-22-2018	Version 1	1.0

1. Introduction

1.1 Purpose

The purpose of this document is to describe the requirements specification and scope of the Rapid LDraw Parts Definition capstone project. These requirements have been derived from the LDraw project sponsor and the project team. The project described in this document will serve the project sponsor and other Lego enthusiast by allowing them to transform an STL file of a lego piece into an LDraw parts definition file.

1.2 Intended Audience and Reading Suggestions

This document is to be read and signed by the sponsor of the project, Gene. As well as referenced by the project team throughout development. To better understand this document, please refer to the references section of this document.

1.3 Project Scope

The primary deliverable for this project is a converter that transforms an STL file into an LDraw parts definition file. In addition to the converter, a GUI interface for input file selection as well as the output file metadata modification will be included as a deliverable. The GUI will also include buttons to start, pause, and cancel the conversion.

The expected artifact is: All source code made available to the open source community consistent with the course requirements. This will include any source code for custom tools. Further, as part of this artifact, detailed instructions on how to compile and use the product will be included.

The scope does not include the production of the 3D parts scan (STL file) or a graphical comparison of the output LDraw file to an existing and known good LDraw file of the same style. Although the initial scanning for retrieving the STL file for the expected input, and a graphical comparison are outside of the scope of the deliverables, they shall be included in the development of the project by the team.

1.4 References

LDraw specification information: <http://www.ldraw.org/>

LDraw Meta specification information: <http://www.ldraw.org/article/401.html>

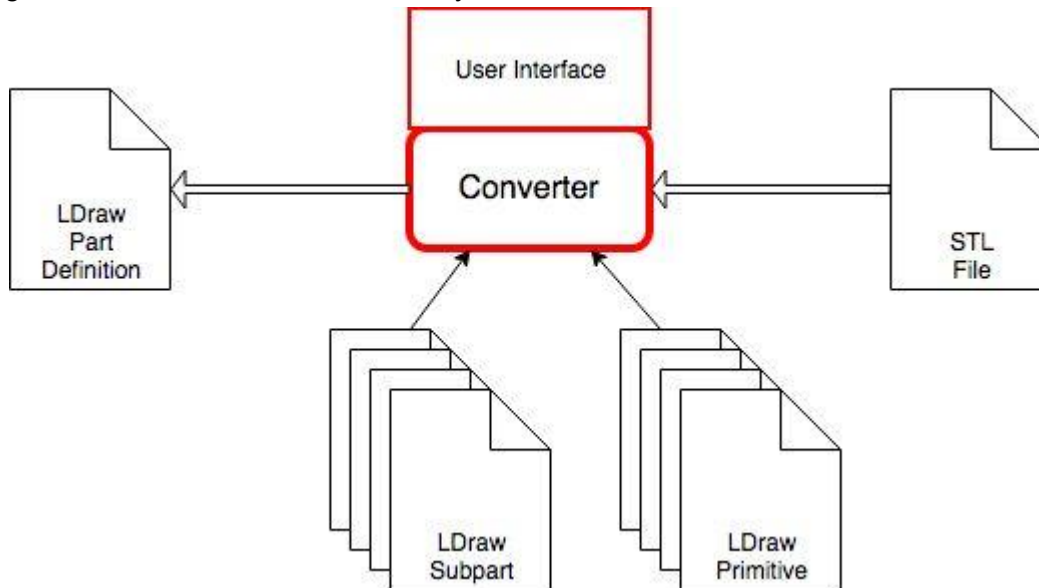
STL specification information: [https://en.wikipedia.org/wiki/STL_\(file_format\)](https://en.wikipedia.org/wiki/STL_(file_format))

2. Overall Description

2.1 Product Perspective

This project was requested as a capstone project by Gene Welborn. The Rapid LDraw Parts Definition project is a standalone piece of software that will be used to convert 3D STL files of Lego parts to a known, valid LDraw parts definition file.

The diagram below illustrates the overall system.



In addition to the above diagram, the GUI wireframe diagram will be included in the design document.

The conversion will take full advantage of the LDraw primitives and subpart definitions and use those where applicable to replace verbose line, triangle, rectangles. For example, a primitive exists to model the standard Lego™ stud, this will be used instead of a series of rectangles and triangles.

2.2 Product Features

2.2.1 Minimum Viable Product

1. Product will execute on a standard commercial computer (basically, it will not require an exotic computing platform).

2. Conversion of STL scan into compliant¹ LDraw part definition that will use LDraw primitives and subparts definitions where appropriate.
3. The converted LDraw part definition can be seamlessly used by most (if not all) existing CAD programs that use the LDraw parts library. The initial selection of CAD programs to be used for testing will include Bricksmith and LeoCAD.
4. A minimal user interface with the following features
 - a. GUI (not command line) that uses standard GUI paradigms (mouse control, menus, buttons, etc.).
 - b. Allows file selection of an STL file to be converted. File selection process should highlight allowed file extensions.
 - c. Allows for entry of new LDraw part metadata, as well as the ability to edit existing LDraw part metadata. Where possible, fields should be pre-populated. Allows user to edit pre-populated fields. At a minimum, the metadata required by the LDraw parts organization will be added to resulting output LDraw file. A link to this part of the specification is posted in the references section of this document.
 - d. Where possible, automatically creates the converted LDraw part file name consistent with the part under conversion. Allows user edit of file name.
 - e. “Go”, “Pause” and “Cancel” buttons
 - f. Aesthetics (like animated transitions or fancy logos), while they enhance the professional appearance, are not necessary. However, the UI should be perceived as “usable” (among other things, this implies a responsive and performant interface).
 - g. Localization is not required. The user interface will be in en-US.
5. A mechanism for comparing official LDraw part definition with the converted LDraw part definition. *The exact measure of the “comparison” is to be determined.*

2.2.2 Usable Product

In addition to the MVP Goals, the Usable Product will primarily expand the functionality of the User Interface.

1. Ability to visually inspect the converted LDraw part in wireframe. This will include scaling and rotation².
2. Ability to visually inspect the initial 3D scan in wireframe this will include scaling and rotation³.
3. Ability to visually inspect the initial 3D scan and converted LDraw part side-by-side with synchronized scaling and rotation⁴.

Although not required⁵, it may be useful to have a mechanism for batch processing of the conversion process⁵.

¹ In this context, compliant means fully adheres to the LDraw Parts Specification.

² Caution, if a third party library is used to render the part, insure its license allows for usage in this product.

³ Same as 3.

⁴ Seemingly, this would imply the converted LDraw part file will have some reference to the original 3D scan. This should be achievable with a metadata comment in the LDraw file.

⁵ If attempted, this could be a command line or scripted feature.

2.2.3 Stretch Goal

The stretch goal would be to have a fully packaged software product that could be presented to the Lego™ fan community.

2.3 User Classes and Characteristics

This will be used by advanced Lego enthusiasts who have at least a basic familiarity with LDraw parts definition files.

2.4 Design and Implementation Constraints

There will be many challenges in the development of this product. The first of which will be time. We must have a minimum viable product to present by the end of the Winter 2019 term. This is a hard deadline. Any improvements beyond the minimum viable product may be implemented with the agreement between the sponsor and project team members. We will be developing using Python (3.6) as a language. This means that the software must conform to the licensing requirements of any 3rd party libraries that are used. All libraries and 3rd party dependencies must also have the ability to be licensed under an open source license.

2.5 User Documentation

A basic user manual will be provided to the user as well as an extensive help page accessible from the GUI via a “help button”. There will be an associated readme with each major group of code.

2.6 Assumptions and Dependencies

The main assumption for this project is that the STL files for the input of our program will be available and of good enough quality for our algorithms to properly convert to the output LDraw files. For the development of the project, we will be starting with an idealized image and converting that STL file into an accurate LDraw file. The next development step will see us adding noise to the idealized image STL file and converting that into an accurate LDraw file. Finally, we will be taking a 3D scanned image of a Lego brick, as an STL file, and convert that into an accurate LDraw file.

3. Other Nonfunctional Requirements

Performance Requirements

Conversion completes rapidly. Definition of rapidly will depend on available processing power, the complexity of algorithms and the part being converted. However, the expectation is that the elapsed time will be no more than 2-3 hours (excluding the parts scan). *The time will be negotiable once enough of the product is implemented to gain a full understanding of the complexities involved.*

Software Quality Attributes

All software will adhere to good software practices, including but not limited to

- Verbosely commented
- Adheres to a reasonable and consistent coding style.
- Well structured with “isolation of concerns”.
- Correct spelling.
- Testable. Unit and integration tests where these tests add value to either the programmer or maintainer.
- Maintainable. To add clarity here, what this means is to make readable and understandable rather than clever code. There are exceptions when complexity or clever coding is required to address performance concerns. However, in those cases, explicit, accurate and verbose comments are expected.

Project source code organization will have a reasonable partitioning of source files (UI files separated from conversion algorithms, etc.). Each “group” will have an associated readme file detailing the purpose of the files in that group.

Extreme care will be taken to ensure adherence to any licensing requirements. Obscure or unmaintained libraries will be avoided.

All tools (IDEs, compilers, etc.) necessary to the project, will be free and generally available.

4. Milestones

Design and Implementation Plan Completion

1. A document detailing the design of the product.
 - a. Description of algorithms used to perform the conversion from a STL file to an LDraw parts definition.
 - b. Overview of the data flow within the product.
 - c. Any expected errors and how they are handled (invalid file format, etc.)
 - d. Logging (if available).
 - e. Links to all relevant specifications.
2. A document detailing the implementation plan of the product

- a. Programming language(s) used
 - b. Third Party libraries used (with references to any licenses)
 - c. Standard tools used (compilers, IDEs.)
 - d. Any custom tools
 - e. Source control information (paths, etc.)
 - f. Testing plan, both unit/integration tests and final acceptance tests.
3. A rough schedule with specific delivery milestones. This schedule will also include any germane course deadlines.

Note: The design and implementation documents are expected to be living documents that will change during the course of the project. Reasonable efforts will be made to keep them current, and they will be accurate at project completion.

Alpha

Product is functional enough to demonstrate its features. It will allow completion of the requirements (conversion time, comparison measure, etc.)

Beta

Product is fully functional. It could be put in the hands of a user (Gene).

Release

Product is both fully functional and essentially bug free.

Disclaimers

This project is not endorsed, sponsored or associated with The LEGO™ Group. LEGO® is a trademark of the LEGO Group of companies.

This project is not endorsed or sponsored by the LDraw organization.

Appendix A: Glossary

CAD (Computer-aided design)	the use of computer systems (or workstations) to aid in the creation, modification, analysis, or optimization of a design
Part Number	a special number LEGO assigns to a specific mold.
LEGO	a line of plastic construction toys that are manufactured by The Lego Group
LEGO Primitive	a highly reusable component of a LEGO part modelled for LDraw.
LEGO Sub-part	
LDraw	an open standard for LEGO CAD programs that allow the user to create virtual LEGO models and scenes.
STL	a specific file format known as stereolithography. Used by CAD programs to view digital 3D objects.

Appendix B: Issues List

Questions::

1. Should user selection data be included in the GUI, such as number of studs, etc.?
2. Is Pygame Python library (among others) going to work with our licensing?