SYSTEMS AND SOFTWARE DESIGN DESCRIPTION (SSDD)

FOR

RuleBear
(a Google SketchUp Plugin)

Version 1.0
April 2, 2013

Prepared by:
Jason Fletcher
Maxine Major

University of Idaho
Moscow, ID 83844-1010

RECORD OF CHANGES

Change No.	Date	Location of Change (e.g., page or figure #)	A M D	Brief Description of Change	Changed By (Initials)
1	4/2/2013	Document	A	Document Created	MM
2	4/3/2013	4.1.2 Lego Class	A	Class diagram added	MM

***A** - ADDED **M** – MODIFIED **D** - DELETED

RuleBear
TABLE OF CONTENTS

1. INTRODUCTION

1.1. IDENTIFICATION

The plugin under development is named RuleBear. Further Revisions of this document are incremental starting with version 1.0.

1.2. DOCUMENT PURPOSE, SCOPE, AND INTENDED AUDIENCE

1.2.1. DOCUMENT PURPOSE

The RuleBear plugin is being developed according to a set of requirements out lined in the RuleBear System and Software Requirements Specification. This document will provide detailed information regarding the desired effect of the RuleBear design.

1.2.2. DOCUMENT SCOPE

This document contains information regarding the design and components of the RuleBear plugin. The structure of the plugin, and design decisions for this plugin are included here.

1.2.3. INTENDED AUDIENCE FOR DOCUMENT

This document is intended to be referenced by RuleBear stakeholders and further developers on the RuleBear project.

1.3. SYSTEM AND SOFTWARE PURPOSE, SCOPE, AND INTENDED USERS

1.3.1. System and Software Purpose

The RuleBear plugin is intended to create a rules-based environment for Lego object placement in Google SketchUp.

1.3.2. System and Software Scope and/or Context

This plugin will permit users to both define and use rules which apply to the SketchUp modeling environment.

1.3.3. Intended Users for the System and Software

This plugin is intended to be used by the Mechanical Engineering Department at University of Idaho, primarily for educational purposes. However, this plugin may be publicly distributed.

1.4. DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

Term or Acronym	Definition
Alpha Test	Limited release(s) to selected, outside testers
Beta Test	Limited release(s) to cooperating customers wanting early access

	to developing systems
Final Test	Release of fully functional product to customer for approval, aka, Acceptance Test.
SSDD	System Specifications and Design Document
SSRS	System and Software Requirements Specification

1.5. REFERENCES

There are no references to be cited for the RuleBear plugin at this time.

1.6. DOCUMENT OVERVIEW

Section 2 of this document describes the system and software constraints imposed by the operational environment, system requirements and user characteristics, and then identifies the system stakeholders and describes their concerns and mitigations to those concerns.

Section 3 of this document describes the system and software architecture from several viewpoints, including, but not limited to, the developer's view and the user's view.

Section 4 provides detailed design descriptions for every component defined in the architectural view(s).

Sections 5 provides traceability information connecting the original specifications (referenced above) to the architectural components and design entities identified in this document.

Section 6 and beyond are appendices including original information and communications used to create this document.

1.7. DOCUMENT RESTRICTIONS

This document is intended to only be distributed in conjunction with the RuleBear plugin, and only as deemed necessary for research, development, or educational purposes.

2. CONSTRAINTS AND STAKEHOLDER CONCERNS

2.1. CONSTRAINTS

2.1.1. Environmental Constraints

The RuleBear project does not have any effect beyond the software plugin, and so environmental constraints do not apply to this project.

2.1.2. System Requirement Constraints

Because the RuleBear plugin is designed solely for use within Google SketchUp, the system upon which RuleBear will be functioning will need to be capable of running SketchUp effectively.

As of the date of this SSDD revision, supported systems include:

- Windows XP/Vista/7
- Mac OS X 10.5+, 10.6+, and 10.7+

Up-to-date system requirements can be found on the SketchUp website at http://support.google.com/sketchup/bin/answer.py?hl=en&answer=36208.

2.1.3. User Characteristic Constraints

Because the RuleBear plugin is developed for use by the Mechanical Engineering Department at University of Idaho, it is expected that users of this plugin will primarily consist of Mechanical Engineering undergraduate and graduate students, faculty, and potential applicants to the program. These applicants are expected, at a minimum, to be familiar with either Windows or Mac operating systems, and will be able to intuitively navigate basic programs in these operating systems.

The RuleBear plugin will be administered under the oversight of those who have already been introduced to the plugin, so it is not expected of basic users to understand how to install and launch the plugin, and consequently, RuleBear will not be including an installation “wizard” or in-program tutorial as of this version. However, understanding that users of the plugin may not be familiar with the SketchUp environment, the plugin will be intuitive enough to be used as a stand-alone product.

2.2. STAKEHOLDER CONCERNS

The primary stakeholder for the development of this plugin is the Mechanical Engineering Department at University of Idaho. This plugin is intended to be an ongoing project, and as such, considerations must be made to ensure that the design may be further developed and modified as the needs of the department are incorporated into future editions of this plugin. The stakeholders must be consulted - and must approve - of each feature incorporated into the program.

This plugin must be designed so that additional rules and objects may be incorporated without excessive complication in further design and coding. This version of the RuleBear plugin is not the final version, and if it should be too difficult to modify for the Mechanical

Engineering Department’s needs, it would be not serve the stakeholders’ ultimate goal for this plugin.

Since the RuleBear plugin project is developed for University of Idaho use, further development of this project will halt if the stakeholders who are overseeing this project decide that this project should to no longer continue.

3. SYSTEM AND SOFTWARE ARCHITECTURE

3.1. DEVELOPER'S ARCHITECTURAL VIEW

3.1.1. Developer's View Identification

This is the architecture of the program from the viewpoint of the developer. The purpose is to give an overview of the details of the major components of the architecture.

3.1.1.1. In order to have the program be able to display available rules, __________ is developed. Explain.

3.1.1.2. In order to have the program be able to display available shapes, _________ is developed. Explain

3.1.1.3. In order to have the program be able to place objects, _______ is developed. Explain

3.1.1.4. In order to have the program allow or disallow object placement per the rules, __________ is developed. Explain

3.1.1.5. In order for the user to define rules, ____________ is developed, which interfaces with _____________, to send the configuration to the SketchUp environment. Explain

3.1.2. Developer's View Representation and Description

Explain how each module above interacts with the others, so that the complete program is designed. This is essentially a paragraph which will explain our design from a non-detailed perspective.

3.1.3. Developer's Architectural Rationale

Explain why we decided to have the above components.

For further detail on alternate designs considered in the development of this project, please refer to Appendix A.

3.2. USER'S ARCHITECTURAL VIEW

3.2.1. User's View Identification

This is the user's viewpoint of the plugin. From the user's viewpoint, there are three main components to this program: the rules.txt file, the rule selection menu, and the object selection menu.

3.2.2. User's View Representation and Description

The rules.txt file is the location in which the user may specify specific rules or limitations to the objects and their placement within SketchUp.

The rules menu is the location where a user may select or de-select the rules for the SketchUp environment. Only zero or one rule may be selected at any time (for this version of RuleBear). The rule selected will affect which of the RuleBear objects may be placed into the SketchUp environment.

The object placement menu is a list of available objects which the user may place into the SketchUp environment. The success of object placement will depend on whether or not the placement is permitted by the rule selected at that time.

SketchUp functionality is not affected by any of the above, except where otherwise specified in this document.

3.3. CONSISTENCY OF ARCHITECTURAL VIEWS

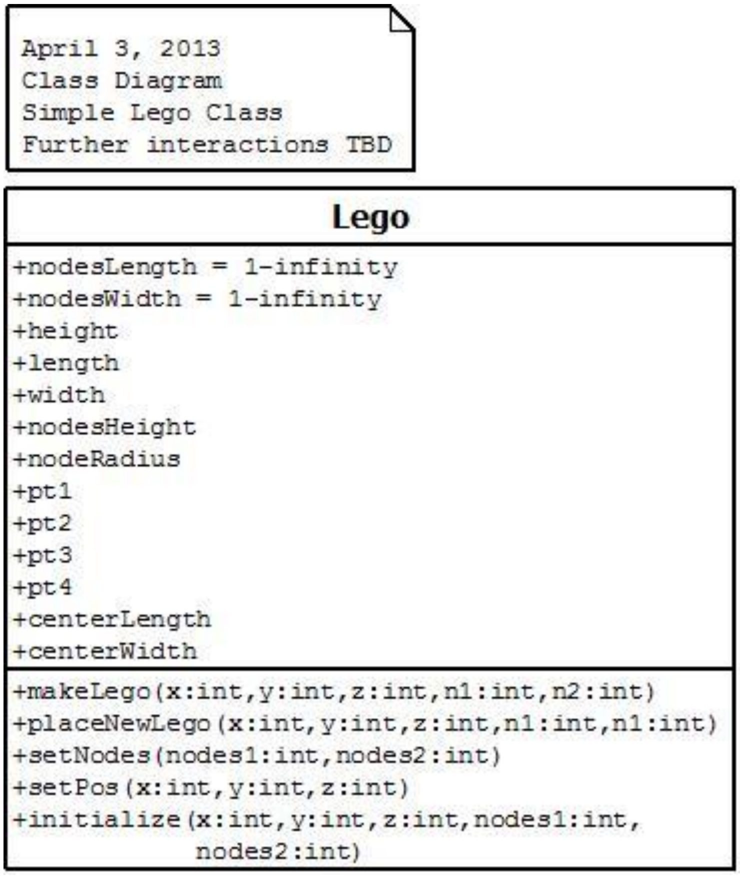
The architectural views are consistent with the exception of the rules.txt file. The rules.txt file does not directly translate into the RuleBear plugin, and instead must be interpreted by an intermediate program, with which the end user is not concerned.

4. SOFTWARE DETAILED DESIGN

4.1. DEVELOPER’S VIEWPOINT DETAILED SOFTWARE DESIGN

4.1.1. Class Overview

4.1.2. Lego class



```
April 3, 2013
Class Diagram
Simple Lego Class
Further interactions TBD
```

```
Lego
+nodesLength = 1-infinity
+nodesWidth = 1-infinity
+height
+length
+width
+nodesHeight
+nodeRadius
+pt1
+pt2
+pt3
+pt4
+centerLength
+centerWidth
+makeLego(x:int,y:int,z:int,n1:int,n2:int)
+placeNewLego(x:int,y:int,z:int,n1:int,n1:int)
+setNodes(nodes1:int,nodes2:int)
+setPos(x:int,y:int,z:int)
+initialize(x:int,y:int,z:int,nodes1:int,
            nodes2:int)
```

4.2. COMPONENT ENTITY DICTIONARY

4.3. FEATURE DETAILED DESIGN

4.3.1. Detailed Design for Feature: (verb)

4.3.2. Detailed Design for Feature (verb)

5. REQUIREMENTS TRACEABILITY

This section will detail all the requirements outlined in the Requirements Specifications document, and identify which sections of this Design document meet each of the requirements.

This section will be completed as the design is finalized.

APPENDIX A

Design Alternatives

Design Proposal #1

This proposal focuses more on the order of development of the plugin, rather than the final effect.

- The rules file is created first
  - Rules file includes objects, the way of specifying the objects, and the rules that define object interactions
- Conversion program takes rule file and generates a ruby file, which would then become the plugin for SketchUp

PROS:

- Much more flexible, since from this aspect, the rule-creator could define their own objects.

CONS:

- Much more complex for the rule creator, and highly complicated for programmers. Higher complexity results in a higher probability of programming errors.
- More suitable for an extended project. Not feasible to complete in a single semester.

EFFECT ON FUTURE DEVELOPMENT

Design Proposal #2

this strategy focuses on implementing the Lego environment first, and ensuring that it works with sketchup, and the rules file is to be developed last, as a way to modify the hard-coded plugin.

- Create the Lego objects
  - Create object in SketchUp. Establish that the object may be copied and repeatedly placed.
  - Define additional properties for object
    - rules will apply based on what is defined within an object. Additional properties may be added to a object based on what rules may be needed, but the objects themselves do not care what the rules are; only what properties they contain.
  - Ensure that the object may be replicated into the same environment
    - **At what point do we care that this happens? GUI related.**
- ensure that the Lego objects connect to each other via studs → sockets
  - The
- Lego objects can only be placed via rules
  -
- Sketchup interaction: turning on and off environment
- Create the rules file to affect the environment.

APPENDIX B

Features Considered for Future Revisions

Many of the features listed below have been identified as crucial to the full functionality of this plugin as envisioned by the stakeholders. However, given the limited time and manpower in developing the pilot version of RuleBear, these features are not practical for addition at this time.

Every consideration will be made in the development of RuleBear to ensure that these future additions may be added without much reconfiguration of the existing design.

Scaling

Inevitably, in a design environment, objects may need to be resized to meet specifications or to perform a particular function, etc., and so the omission of scaling from RuleBear would be a fairly major oversight.

Objects should be able to do the following:

1. Be resized according to an exact scale of the original object's dimensions.
2. If two or more objects are resized to the same scale, the corresponding sockets, studs (and other future connectors developed) will also "fit" in the same manner as the original size objects.
3. Objects may be scaled to create new types of "fit" based on the connectors' dimensions, rather than pre-designated connectors. This may be a design included late in RuleBear development.

Multiple Base Objects

The current version of RuleBear strictly limits the environment to a single base object. The reasoning for this is as follows:

In this pilot version of RuleBear, because the initial placement of a base object may be located at any set of coordinates, this does not ensure that the dimensions are exact to the distance between sockets and studs in a Lego object placed elsewhere in the same environment. As a consequence, two objects placed in the base environment may not align properly to allow an object placed on top to correctly align its sockets with multiple base objects' studs.

In subsequent versions of RuleBear, the development environment may not be limited to Lego objects. and an object/object structure may connect to a different object/object structure not directly connected within its own structure, located at any set of coordinates.

In order to determine whether or not an object may connect to another object in the environment, additional algorithms will need to be created to calculate the exact coordinates of connectors and verify that they are properly aligned. This may be a complicated addition to the existing design, and rather than pursue advanced connective functionality at this time, the development team has decided this is best developed in a later version of RuleBear.

Object Rotation

In the current version of RuleBear, all Lego objects may be placed at 90° angles. For the same reasons listed above in the Multiple Base Objects section, the design will not consider what would happen to a Lego object placed at some arbitrary angle (e.g., 60°) in the SketchUp environment.

Future revisions of this plugin would require objects to be placed at many angles in the SketchUp environment, and the RuleBear rules would permit object placement as long as the connectors lined up in the 3-dimensional space.