VISION & SCOPE

SKETCHUP BLOCKS - ANDREW SMITH - CSIR

Version 1.4

13/10/2013

11081092 - ET, van Zyl

11211548 - WH, Oldewage

11183153 - JL, Coetzee

CONTENTS

CHANGE LOG

Version 1.0	10/05/2013	Document created.
Version 1.01	13/06/2013	Change log added.
Version 1.2	24/07/2013	Revised requirements for Release 1.
Version 1.3.1	25/07/2013	Added traceability matrix.
Version 1.3.2	26/07/2013	Added test cases to traceability matrices.
Version 1.4	13/10/2013	Revised cover page

1. CLIENT PERSPECTIVE

1.1 BACKGROUND

3D models of buildings and objects can be assembled more easily and intuitively with physical components than with modelling software. Physical models require no technical experience to construct; they allow interaction by multiple people simultaneously and are simple to manipulate and view.

Unfortunately, 3D models are usually required to be in digital format for use in industry.

This project will develop a system that converts physical 3D models into digital form.

The project subscribes to the concept of the Internet of Things. The Internet of Things is centred around the idea of 'smart objects' which are physical objects and devices that are given a virtual representation and linked to information networks. These smart objects will ideally be able to capture, react to and communicate about data captured from their physical environment autonomously.

The Internet of Things Engineering Group requested the implementation of such a system for research purposes.

1.2 BUSINESS OPPORTUNITY

Sketchup Blocks is not intended to have any immediate business application. It is a research project that will investigate the concept of a smart 3D modelling system. The difficulties encountered, possible solutions to them and overall feasibility of the system will be used in a research paper by IoTEG.

Sketchup Blocks may be used by any organization that requires the construction of 3D models and wishes to minimize the learning curve of 3D-modelling programs. It will also be particularly useful for users that are not computer literate or are technophobic. However, as mentioned previously, it is not intended to be a marketable product.

Eventually, the concept of converting physical objects into virtual representations may be used in many different products. For example, the configuration in which the smart blocks are built may also be used to represent more abstract data like the configuration parameters of an electricity system or a DVD player.

1.3 RELATED DOCUMENTS

Architectural Requirements (Sketchup Blocks project, by CICO).

Design Specification (Sketchup Blocks project, by CICO).

Technical Specification (Sketchup Blocks project, by CICO).

Project Management (Sketchup Blocks project, by CICO).

Test Document (Sketchup Blocks project, by CICO).

Sketchup Blocks Developer Blog (http://sketchupblocks.wordpress.com/).

2. VISION

The vision for this project is to provide the user with the best level of performance, freedom and flexibility from the system that can be provided within reasonable bounds.

The 3D rendering performance of the system should be real-time, the user will be presented with this model to digitally resemble the users progress. The user should be given the freedom to setup the system wherever the user chooses. A mobile version of this system should be made to provide the user with this level of freedom. The user should be provided the flexibility to be able to setup the system in sub optimally lit rooms without sacrificing any system functionality. Finally the stand-alone system should be extended to make provision for the collaboration between two or more people to work on large projects together and ultimately promote productivity.

3. SCOPE AND LIMITATIONS

3.1 RELEASES

Release 1:

The initial system will be able to recognize a small set of geometrically simple objects without any time constraints. After the blocks have been transformed into a digital model, the vertices of the model will be output to file for import in Sketchup. The initial release should contain a 3D debugging window that shows the digital model as it is constructed by the system.

Release 2:

In the second release we aim to limit the time taken to recognize a fiducial and recreate the model it represents. The amount of constructed and recognizable blocks will also be increased. The setup of the system will be simplified to improve mobility and allow for fast setup. The removal of blocks will be detected without extra input from the user.

Release 3

We aim to extend the 3D debugging window to a user-friendly viewing window that will display the model as it being created thereby providing visualization before exporting to a model. The detection of the fiducials should be possible in suboptimal light.

Optional release

Custom blocks may be imported to the smart block database. The construction can be viewable over a network to allow social construction.

3.2 LIMITATIONS

The image recognitioning will only be for fiducials not any object. The fiducials must be registered in the database before use. There will be a minimum amount of cameras required for model construction.

4. HIGH-LEVEL NON-FUNCTIONAL REQUIREMENTS

Speed/Throughput (NFR1)

The system must generate a digital 3D representation for the physical model as quickly as possible to promote maximum productivity from its users. Generation of the object file should happen within a few minutes.

Portability (NFR2)

The system is investigating the concept of providing smart objects to the everyday user. This is only feasible if the system is portable enough to allow for distribution.

Resilience/Flexibility (NFR3)

The system must be flexible enough for use in many different environments and should not require very good lighting conditions (which ties in with portability).

Low production cost (NFR4)

The materials used in system construction must be affordable.

5. PROJECT SUCCESS FACTORS

5.1 DRIVER:

Cubic shapes (FR1)
Smart blocks that are basic cubic shapes must be supported i.e. The model constructed by the user must consist of cubic blocks.

3D Model (FR2)
The system must detect the configuration of the model blocks then convert it into a 3D model by producing a corresponding model file that is native to the Sketchup Blocks System.

Sketchup Integration (FR3)
The object file produced must be in a format that is supported by the Google Sketchup 3D modelling application.

5.2 CONSTRAINTS

Auto-detected block addition (FR4)
The position and orientation of blocks must be detected and interpreted automatically. There are other, similar 3D modelling systems that convert physical models into 3D, but they require the model corners to be marked during construction. Sketchup Blocks must do model detection autonomously.

5.3 IMPORTANT

Auto-detected block removal (FR5)
Removal of blocks from the model should be detected automatically (as opposed to having the user register each block deletion manually).

More complex shapes (FR6)
Smart blocks of many different shapes (cones, cylinders, spheres, etc.) should be supported so that the user can build more complex models.

Throughput (NFR1)
The system must be able to generate the corresponding 3D representation of the model in under half an hour, preferably within a few minutes.

Mobile setup (NFR2)
The user must be able to change the location at which the system is set up. Ideally, setting up the system should require minimum technical ability on the part of the user and not take more than a few minutes.

5.4 NICE-TO-HAVES

Real-Time Display (FR7)

The system must provide a 3D representation of the physical blocks that is rendered in real-time and will reflect the digital version of the work done so far.

Network Interaction (FR8)

Collaboration over a network between two or more users must be supported so that people all around the world can contribute a joint effort in the making of a large project.

Custom Shapes (FR9)

The system must have the ability to use image recognition to analyse the dimensions of newly introduced shapes so that these new shapes can be used in the making of 3D models. This is not within project scope.

6. REQUIREMENT MATRICES

6.1 REQUIREMENT IDENTIFIERS

FUNCTIONAL REQUIREMENTS

REQUIREMENT	DESCRIPTION	SOURCE	PRIORITY
FR1	Cubic/Simple shapes	Andrew Smith	High
FR2	3D Model	Andrew Smith	High
FR3	Sketchup Integration	Andrew Smith	High
FR4	Auto-detected block addition	Andrew Smith	High
FR5	Auto-detected block removal	Andrew Smith	High
FR6	More complex shapes	Andrew Smith	Medium
FR7	Real-Time Display	Andrew Smith	Medium
FR8	Network Interaction	Andrew Smith	Low
FR9	Custom Shapes	Andrew Smith	Low

NONFUNCTIONAL REQUIREMENTS

REQUIREMENT	DESCRIPTION	SOURCE	PRIORITY
NFR1	Throughput	Andrew Smith	High
NFR2	Portability	Andrew Smith	Medium
NFR3	Resilience	Andrew Smith	Medium
NFR4	Low production cost	Andrew Smith	Low

6.2 REQUIREMENTS TRACEABILITY MATRIX

TRACEABILITY MATRIX – FUNCTIONAL REQUIREMENTS

	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	FR9
Vision & Scope	X	X	X	X	X	X	X	X	X
BlockDatabase Test	X	X				X			X
Calibration Test		X					X		
Evaluator Test		X							
Exporter Test			X						
Interpreter Test				X	X				
LinearSolver Test		X							
Local Lobby Test		X					X	X	
Matrix Test		X							
Menu Test	X						X	X	
Model Viewer Test							X		
Networked Lobby Test								X	
Particle Creator Test		X							
Server Test								X	
Vec3 Test		X					X		
Vec4 Test		X					X		
Done				X	X				
Acceptance Test 1									
Accepted									

TRACEABILITY MATRIX – NONFUNCTIONAL REQUIREMENTS	NFR1	NFR2	NFR3	NFR4
Vision & Scope	X	X	X	X
BlockDatabase Test				
Calibration Test	X	X	X	
Evaluator Test	X		X	
Exporter Test				
Interpreter Test			X	
LinearSolver Test	X	X	X	
Local Lobby Test				
Matrix Test				
Menu Test				
Model Viewer Test				
Networked Lobby Test				
Particle Creator Test	X	X	X	
Server Test				
Vec3 Test				
Vec4 Test				
Done	X			X
Acceptance Test 1				
Accepted				