# PROJECT MANAGEMENT

## SKETCHUP BLOCKS - ANDREW SMITH - CSIR


Version 2.3

13/10/2013

11081092 - ET, van Zyl

11211548 - WH, Oldewage

11183153 - JL, Coetzee

## CONTENTS

## CHANGE LOG

| | | |
|---|---|---|
| Version 1.0 | 25/08/2013 | Document created |
| Version 1.1 | 25/08/2013 | Added member responsibilities |
| Version 2.0 | 15/09/2013 | Updated progress |
| Version 2.1 | 15/09/2013 | Modified cover page and added introduction |
| Version 2.2 | 13/10/2013 | Revised cover page |
| Version 2.3 | 13/10/2013 | Updated project progress |

# 1. INTRODUCTION

## 1.1 PURPOSE

The purpose of this document is to give a high-level abstraction of the architectural strategies/tactics that will be used in the Sketchup Blocks project. This document aims to provide clear motivation for the chosen architectural design as well as an explanation of how it will serve to implement the specifications required by the Vision and Scope document for the Sketchup Blocks project.

## 1.2 PROJECT SCOPE

Sketchup Blocks aims to provide the client with a system that will convert physical 3D models into digital form. The system will include software and hardware components.

**Hardware components:**

- The physical blocks with fiducial markers attached (called 'smart blocks').
- The construction floor on which the smart block constructions are built.
- Approximately five web cameras.

The scope of the software components will broaden from release to release. The client only requires the first release to consider the project successful. The releases following the first release will address additional concerns and implement extra features mentioned by the client.

**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 an object file for import into Google Sketchup.

**Release 2:**

This release aims 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**

This release aims to add a 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 lighting conditions.

**Optional release**

Custom blocks may be imported to the smart block database. Users can view the construction of other users over the network.

## 1.3 RELATED DOCUMENTS

Project Vision and Scope (Sketchup Blocks project, by CICO).

Design Specification (Sketchup Blocks project, by CICO).

Architectural Requirements (Sketchup Blocks project, by CICO).

Technical Specification (Sketchup Blocks project, by CICO).

Test Document (Sketchup Blocks project, by CICO).

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

# 2. SOFTWARE DEVELOPMENT PROCESS

The project given to us by our client is currently done for research purposes. The project requirements will not change during the course of the project, because that would consequently change the parameters of the experiment.

Given that the project has fixed requirements, the team decided to take the Waterfall approach to do the documentation. Mainly, the Vision & Scope and Architectural requirements of the project will remain the same throughout the course of the project.

The team adopted most of their development methodologies from XP.

## 2.1 XP

The team knows the importance of communication, thus face-to-face communication is practiced whenever possible.

Code is a step-by-step explanation of what a system does and that is why the team believes that code serves as documentation. Although it serves as very good documentation, it is not thorough enough for the purposes of the project. Additional documentation is provided to explain the complex mathematical processes used in the system.

The team adheres to the following XP practices:

**Planning**

The team defined user stories to lay out the different ways in which the user would interact with the system and also brainstormed the difficult concepts to get solutions faster and get better solutions to problems.

**Simple Design**

The team believes in a modular environment. This makes it easy to break up complex components into smaller and simpler components which are easier to maintain, test and debug.

**Small Releases**

The team tries to keep the feedback cycles as short as possible. Any quantifiable improvement to the system is merged with the main release of the project.

**Metaphor**

The problems the team are solving are very complex, thus it is very important for the team to find conceptual equivalents for these difficult concepts in order to ease the process of finding a solution.

### Refactoring

Continuous refactoring is practiced in order to prevent that the code's complexity gets out of hand. Fresh code needs to be refactored, because old code needs to be read and understood wasting valuable time.

### Pair Programming

The problems being solved by the team are very complex. The team uses pair programming as a tool to boost productivity and code quality. Pair programming is not enforced when coding simple functionality.

### Collective ownership

No team member owns code, any member can refactor or improve any code given that the functionality of the code is preserved.

### 40 hour weeks (or even less)

Solving complex problems are very taxing on the members of the team, so it is not expected of them to sit for long hours and remain productive.

### Coding Standards

Clean and readable code is very important. The team enforces coding standards throughout the project to ensure readability and maintainability of the code.

## 2.2 SCRUM

The team used the research concepts from scrum to explore solutions to complex problems and how it would be integrated with the system.

### Spike

Time invested to research for solutions to the complex problems.

### Tracer Bullet

This is usually the implementation of the findings from the Spikes to see if the solutions work as expected.

# 3. TEAM MEMBER ROLES AND RESPONSIBILITIES

## ELRE VAN ZYL

### ROLES

- Programmer
- Researcher

### RESPONSIBILITIES

- Numerical methods (LU decomposition, SVD decomposition, etc)
- Block position calculation
- Testing and documentation delegation

## HEIN OLDEWAGE

### ROLES

- Programmer
- Researcher

### RESPONSIBILITIES

- Particle Swarm Optimization implementation
- Block position calculation
- Camera calibration

## JACQUES COETZEE

### ROLES

- Team Leader
- Programmer
- Researcher

### RESPONSIBILITIES

- Leading the team.
- Ensuring all team members remain motivated.
- Ensuring the client remains updated and happy.
- Responsible for all hardware and graphic design.
- Mainly in charge of GUI and 3D Graphics.

## 4. ISSUE MANAGEMENT

**Requirement Issue**

The team contacts their client and clears up what the clients wants to be done regarding the uncertainty about the requirements.

**Mathematical Issue**

Mathematicians could be contacted to make use of their expertise or research can be done by the team to resolve the problem.

**Design Issue**

Thorough design was done before implementation started on the project, so there should not exist design issues. If a design issue does come to light, then the architectural design of the project should be redesigned to fix the problem.

**Technology Issue**

The team should do research and try to find a solution. If no solution is found for the problem, then the client should be contacted and informed. The client may provide some alternative technologies to work with as a substitute.

## 5. PROJECT PROGRESS

The system can detect building blocks and correctly determine their positions in 3D space and orientation can be determined with 3 or more fiducials visible. This can be viewed in a real-time 3D view and also as a spectator over a network connection. User commands for exporting, rotating the 3D view and spectating can be detected and executed. The system supports multiple camera input. The system can correctly determine each camera's position in 3D space. The final 3D model can be exported as Collada file which Google Sketchup can import. All export and debug functionality are also provided as a spectator.

Removing a block from the model must still be refined. Developing an algorithm for determining the block position with only 2 fiducials visible must still be implemented, but is likely not to see demo day. Extensive unit tests must also be implemented.

## 6. OUTSTANDING RISKS AND CHALLENGES

**ReacTIVision**

ReacTIVision does not allow running multiple cameras on a single computer, because the software cannot grab more than one camera at a given time. This means that a computer would need to be dedicated for every camera used by the system and consequently means that the system would no longer be portable or affordable. The usage of virtual machines have been considered but this solution only works on some computers and requires commercial version of virtual machine software.

**Block Orientation**

In the case that the camera's of the system only detect two fiducial markers of a given building block, then the system is going to have to assume that the block is flat. This is due to the limited information that only two fiducial markers provide the system. This can limit the user in terms of building freedom or even result in some incorrect orientations. Thus a minimum of three fiducial markers need to be detected by the camera's in order to deliver accurate block orientations. Further investigation is pending to make the system more robust, calculate more accurate orientations and provide the user with more building freedom.

**More Complex Shapes**

The system has only been tested with basic cubes as building blocks. The team has to introduce more complex building blocks into the system. This tests the system's ability to dynamically handle information about new building blocks and determine their orientation and position in 3D space.