

Singularity Software
Milestone 1

September 14, 2011

Contents

1	Executive Summary	3
2	Introduction	3
3	Client Background	3
4	Current System	3
5	User/Stakeholder Description	3
5.1	User/Stakeholder Profiles	3
5.1.1	Sifteo application developers	3
5.1.2	Clients	4
5.1.3	Singularity Software	4
5.1.4	Sriram Mohan (CSSE Department)	4
5.2	User Environments	4
5.2.1	Sifteo application developers	4
5.2.2	Clients	4
5.2.3	Singularity Software	4
5.2.4	Sriram Mohan (CSSE Department)	4
5.3	Key Needs	4
5.3.1	Emulate Sifteo Cubes in a desktop GUI application	4
5.3.2	Develop an API for creating applications in the emulators Cubes .	4
5.3.3	Provide several example games that showcase Cube/emulator func- tionalities	5
5.4	Alternatives & Competition	5
6	Product Overview	5
6.1	Product perspective	5
6.2	Elevator statement	5
6.3	Summary of capabilities	6
6.4	Assumptions and dependencies	6
6.5	Estimate of cost	6
7	Features	7
8	Constraints	7

1 Executive Summary

This is where the executive summary lives.

2 Introduction

This is where the introduction lives.

3 Client Background

Clients Tim Ekl and Eric Stokes are alumni of Rose-Hulman. Mr. Ekl is currently working on a M.S. degree in Engineering Management; Mr. Stokes is currently working in fix me. As (former) Rose-Hulman students, the clients are avid users of technology and follow new trends in the industry. This interest brought both Mr. Stokes and Mr. Ekl to discover Sifteo Cubes, a revolutionary product, still in the Early Access phase, that consists of a set (anywhere from 1 to 6) mini computers. While both clients have a set of 3 Sifteo Cubes, they realized that not everyone interested in the project could have the luxury of physical hardware to work with. As such, they asked Singularity Software via the Junior Project proposal process at Rose-Hulman to construct a software emulator for the cubes. The emulator is intended to make the development of Sifteo applications easier, specifically in the realm of testing.

4 Current System

Currently there is no system in place to simulate use of the Sifteo cube system. Tim Ekl did state that he had developed a basic emulator in the past, but asked that we not look at his code or acknowledge his system in any way. Additionally, Sifteo has announced that they will be releasing their API later this month, but currently the specifics of it are unavailable to the public.

5 User/Stakeholder Description

5.1 User/Stakeholder Profiles

5.1.1 Sifteo application developers

As the primary target audience of the system, developers targeting the Sifteo platform are assumed to have a reasonable amount of technical background; they are not novice computer users and are familiar with programming concepts like Object-Oriented Programming and Application Programming Interfaces. As developers may hail from platforms ranging from Windows to Mac to Linux, cross-platform support is an important consideration.

Possible problems for this user type include an unstable or crash-prone system; as developers are writing and testing their own code, it is essential that the emulator does not contribute to the failures that the user must debug. Developers will deem the emulator

project a success when they can successfully program and test software that uses any or all of the features of the Sifteo cubes on their development platform.

5.1.2 Clients

The clients (Tim Ekl and Eric Stokes) are assumed to be a subset of the Sifteo application developers user class. However, their programming knowledge and knowledge of the Sifteo Cubes is known to be more advanced than that of the average application developer. As such, their needs require that the emulator is capable of being pushed to the same limits as the actual platform.

5.1.3 Singularity Software

Singularity Software, as the team behind Siftables Emulator, is primarily interested in the creation of a finished product that can be delivered to the clients at the conclusion of the Rose-Hulman junior project cycle. As a team, we are less familiar with the Sifteo platform and are also relatively inexperienced with the scale of project the Emulator entails.

5.1.4 Sriram Mohan (CSSE Department)

As the advisor of the Junior Project, Dr. Mohan has a vested interest in the creation of a finished, deliverable product.

5.2 User Environments

5.2.1 Sifteo application developers

5.2.2 Clients

5.2.3 Singularity Software

5.2.4 Sriram Mohan (CSSE Department)

5.3 Key Needs

5.3.1 Emulate Sifteo Cubes in a desktop GUI application

No emulator currently exists for the Sifteo platform; the need is currently filled by home-brew efforts like Mr. Ekl's Java-based emulator, or developers simply use the Cubes themselves as a test environment. The clients envision a solution where all of the interactions possible with a set of Early Access Sifteo Cubes can be replicated in a software emulator.

5.3.2 Develop an API for creating applications in the emulators Cubes

An Application Programming Interface (API) is necessary to facilitate interaction with the virtual Sifteo Cubes. Currently, no API is made available by Sifteo for the physical cubes, and no emulator API exists because no emulator exists. The clients would like an API (that may or may not resemble the to-be-released Sifteo API) with which the virtual Cubes can be programmed.

5.3.3 Provide several example games that showcase Cube/emulator functionalities

Sifteo currently provides example games that run on the Cubes as a showcase of what can be achieved with the Cubes and what the Cubes unique futures offer to the programmer and to the user. The clients would like to have a similar showcase available for the emulator; such examples would aid in understanding both the emulator platform and the larger Sifteo Cubes programming platform.

5.4 Alternatives & Competition

At the moment, no known emulator exists for the Sifteo Cubes. As such, Singularity Softwares Siftables Emulator will be the first software of its kind for the Cubes. The primary competition, then, is the Sifteo Cubes themselves. The Cubes have the advantage of physicality as tactile objects, they will always be superior in terms of user experience when compared to an emulator. However, they are expensive and only manufactured in limited quantities at the moment; the Siftables Emulator is, by contrast, infinitely available.

6 Product Overview

6.1 Product perspective

The product is a free independent system used to emulate the way Sifteo Cubes (Siftables) handle motions and interactions. Siftables can be used but are not necessary for this product.

6.2 Elevator statement

Due to the quantity of Siftables being limited, developers have a difficult time to create programs for them. At Singularity Software, our goal is to develop a computing emulator for these cubes. Our emulator, which consists of a computing platform, an API, and example games, will be able to emulate an arbitrary number of Siftables and the way they handle motions and interactions in the physical sense.

6.3 Summary of capabilities

The main features of our emulator allows for developers to get started quickly by making development for the cubes more accessible.

Feature	Benefit
Workspace where multiple cubes can be emulated	Allows for a more user-friendly way to develop for multiple cubes
Buttons to control the cubes	Allows for an easier way to control the basic movements of the cubes
The ability to load programs into the cubes	This allows developers to demonstrate that their programs work on the actual cubes.
Example games (requirement)	Examples to get started
Open source (requirement)	The emulator is modifiable.
API (requirement)	Gives developers a programming interface

6.4 Assumptions and dependencies

Sifteo has plans to release an API of their own for the Siftables so we assume our API will coincide with their API.

6.5 Estimate of cost

7 Features

The two attributes assigned for these features include the status to keep track of the progress and the priority to insure which features should be implemented first

Feature	Description	Status	Priority
Individual Cube as Foundation	The emulator will start off with just one cube as a base for a project	Approved	Critical
Workspace where multiple cubes can be emulated	This allows to control multiple cubes	Approved	Critical
Buttons to control the cubes	Having specified buttons that correspond to each motion of a cube	Approved	Critical
Cubes interact with each other	Being able to emulate the cubes interacting with each other.	Approved	Critical
The ability to load programs into the cubes	Programs can be loaded to actual cubes	Approved	Critical
No Save States	Not able to save the state of the cubes	Approved	Important
Auto-align	Automatically aligns multiple cubes when necessary.	Proposed	Useful
Zoom ability	Be able to closely inspect the cubes during development	Proposed	Useful

8 Constraints

While much of this project will be open-ended, there are a few basic constraints. Primarily all code should be open source and version controlled. Tim requested that the emulator run easily on Mac OS as well as Windows, but allowed that Linux compatibility may replace the Mac usage if conflicts arise in testing. Also, an issue tracking system must be in place and used throughout the development process. The only other constraint set forth at this point in the project is that the emulator must be completed by the end of the year.