Singularity Software
Milestone 3

October 9, 2011

By signing below, I approve the contents of the following document.

Alex Mullans

Ruben Rodriguez

Ethan Veatch

Kurtis Zimmerman

Contents

1 Executive Summary

This document is the third in a series of milestone documents that will accompany the planning of the Siftables Emulator. The Emulator project is an application that will allow developers of Sifteo applications to test the features of the cubes in a virtual programming environment. There is currently an emulator from Sifteo, Inc. that comes as part of the Software Development Kit (SDK) for the Cubes. However, Singularity intends to come up with a more natural interface than the one currently provided in that application.

This milestone enumerates all of the non-functional requirements of the the Siftables Emulator project and presents low-fidelity user interface mockups. Discussed first are the requirements for usability, performance, reliability, and supportability. Following that are considerations related to other software and hardware systems as well as documentation and legal requirements. Finally, mockups of the Emulator's design are presented for consideration and to solicit client feedback. Future milestones will present plans for change control, coding standardization, and testing. Finally, design and usability reports will make up the core of milestones near the end of the quarter as the software stabilizes.

2 Introduction

Developers of applications for the Sifteo Cubes currently must test programs they create for the platform within the emulator provided by Sifteo. While this emulator does cover all of the functionality of the Sifteo Cubes, it presents a user interface that Singularity Software believes could be more naturally implemented. As such, Singularity Software will provide, in the form of the Siftables Emulator, a new software-based emulator for the Sifteo Cubes that will allow developers to more naturally interact with the platform.

Milestone 3 elaborates the non-functional requirements of the system and presents low-fidelity mockups of the user interface design. It follows Milestone 2, which laid the foundation of the Siftables Emulator specification based on the high-level design created in Milestone 1. Milestone 4 will rely on these early milestones as it defines a change control plan and test cases, and Milestone 5 will elaborate the usability guidelines and interface design that implement the features and use cases described in Milestone 2 to the non-functional requirements described herein.

3 Project Background

The Siftables Emulator is being developed by Singularity Software as part of the Junior Project sequence of classes at Rose-Hulman Institute of Technology. When projects were solicited for the sequence, clients Tim Ekl and Eric Stokes (both Rose-Hulman alumni) submitted a request for an emulator for Sifteo Cubes, a new platform intended for "intelligent play." After Singularity was chosen for the project, we met with Mr. Ekl to determine the three primary features of the Emulator: a Workspace where 1-6 cubes could mimic the manipulations possible with physical Cubes, an Application Programming Interface (API) to program those virtual cubes, and a set of example games designed to show off the first two features. Singularity's Emulator will be the first program of its kind on the market for Sifteo Cubes.

4 Usability Requirements

The client would like the be accessible to new developers, but also efficient for experienced developers to use. The may have a familiarization time in order for users to be able to use it productively. It will include a help menu containing instructions for the various features. In addition, the client would like keyboard shortcuts and accelerators available for users who prefer keyboard-based controls. As the user must be able to understand how to interact with the cubes relatively quickly, the Cubes' control motions must be intuitive. The user must also be able to switch between motions quickly.

5 Performance Requirements

The performance requirements for the system are fairly basic. When the user interacts with the system there must be no visible graphics delay. Additionally, games simulated on the cubes must function with performance similar to that of the physical Cubes. Beyond these two key requirements, actions must be performed in a timely manner for all scenarios that may arise during use of the .

6 Reliability Requirements

As a system that emulates another system and runs user-created code, it is important that the Siftables Emulator does not contribute defects of its own to the application testing process. As such, Singularity Software will aim for a defect rate of 5 bugs/KLOC. As a predictor of this result, we will evaluate the cyclomatic complexity of our code at each development milestone, aiming for a score of 15 or less. [1] The final measure of the emulator's reliability will be the accuracy of the virtual screen renderings; it will attempt to produce renderings that are 98% accurate. In numbers, this means that up to 327 of each Cube's 16384 (128x128) pixels may be rendered inaccurately or not at all.

7 Supportability Requirements

Because all support for Siftables Emulator will be done by the client after the project has ended, it is imperative that the emulator have well-documented code that is easy for developers unfamiliar with the project to follow. It will also need to follow a recognized coding standard for the language chosen. As a result, it must be in a language like Python or C# that is fairly well-known and frequently used.

8 Hardware and Software Interfaces

The Siftables Emulator will work as a standalone software package and will not interface with any other software or hardware. Singularity Software considered the integration of the emulator in a Integrated Development Environment (IDE) but decided that the emulator was best served as an independent tool. Similarly, the emulator is to serve

[1] Expected cyclomatic complexity scores are based on the blog post at: http://gdub.wordpress.com/2006/07/09/cyclomatic-complexity-for-python-code/.

as a method for developing applications for the cubes without the need for the physical hardware, so it will not interface with any hardware either.

9 Documentation, Installation, Legal and Licensing Requirements

The clients have stated that this is to be an open source, version-controlled project; therefore, many of the standard legal and licensing issues are eliminated. Mr. Ekl has requested that Singularity Software keep documentation of development actions according to project milestone standards as well as using an error tracking system during the design process.

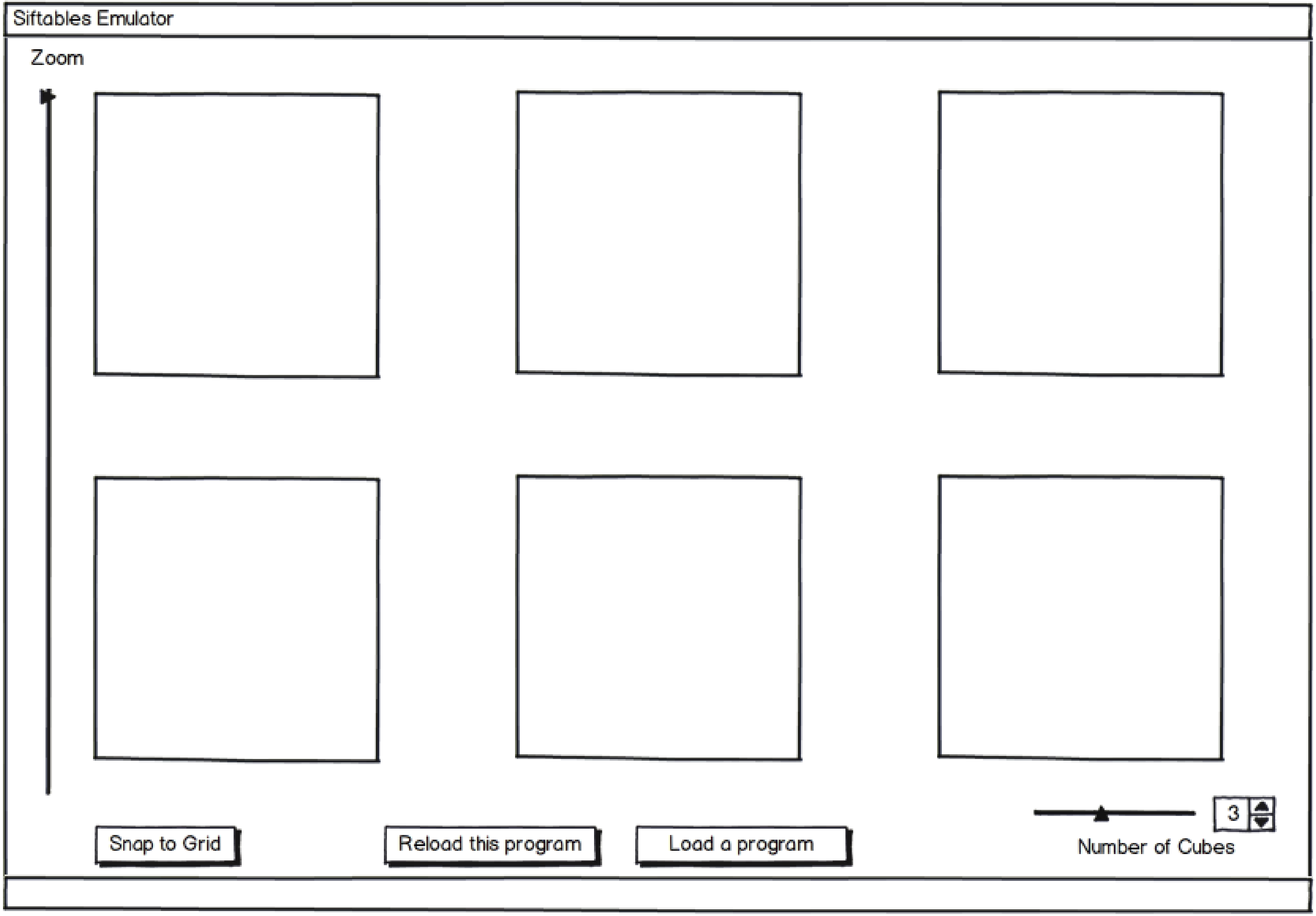
The will be installable via an executable (on Windows) or disk image (on Mac) that will place the program's files in a user-specified directory and create shortcuts as appropriate.

10 Design Constraints

The emulator must be able to be function on , OS, and operating systems with reasonable hardware and software system requirements.

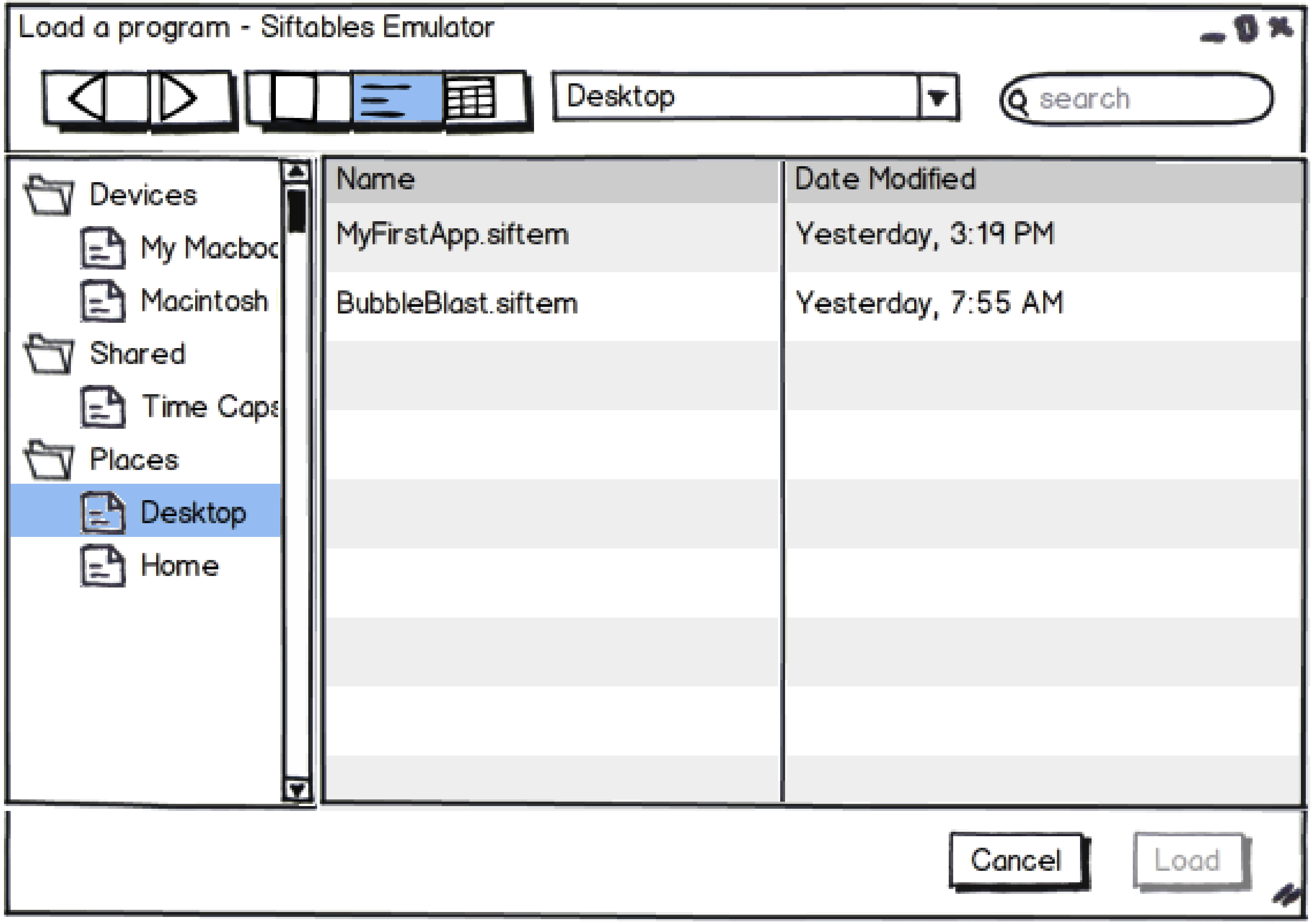
11 User Interface Mockups

Preliminary mockups of the Siftables Emulator user interfaces are presented below. The mockups are rough sketches not intended to convey the product's final look. Rather, they present a simplified view of what Singularity thinks the emulator will look like when created in code.
In order, the mockups display the main screen of the application, the "Load a program" dialog, and the various warning and alert dialogs that the program can display.



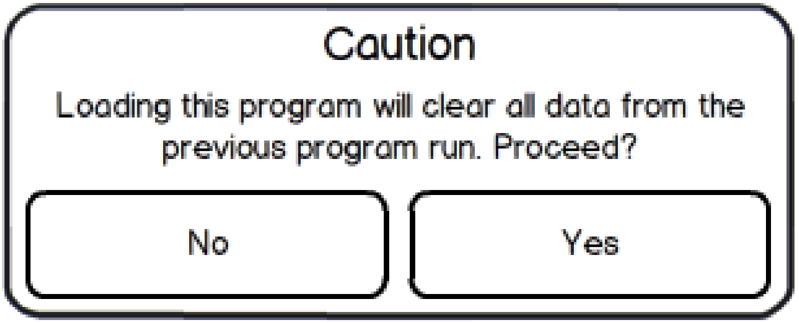
Number of Cubes 3

created with Balsamiq Mockups - www.balsamiq.com



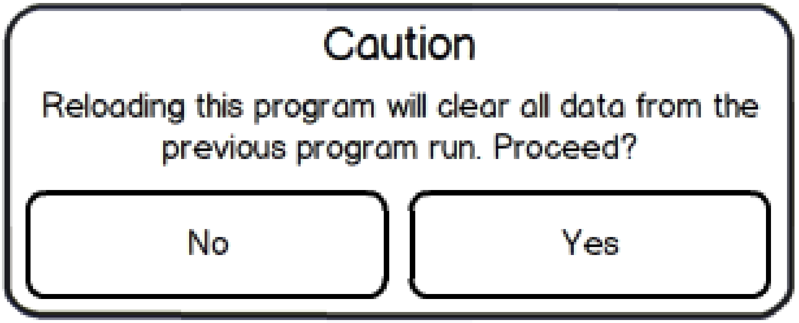
Name	Date Modified
MyFirstApp.siftem	Yesterday, 3:19 PM
BubbleBlast.siftem	Yesterday, 7:55 AM

created with Balsamiq Mockups - www.balsamiq.com



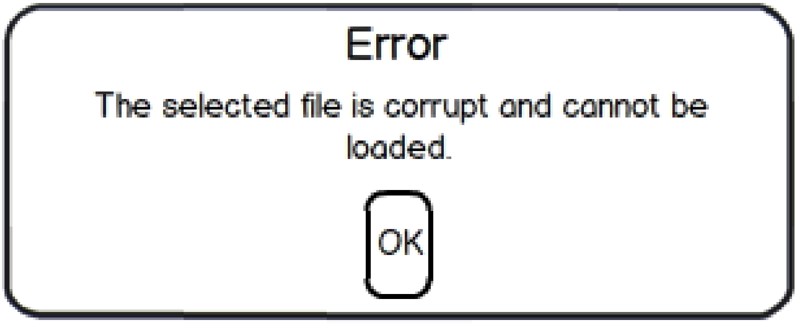
Caution

Loading this program will clear all data from the previous program run. Proceed?



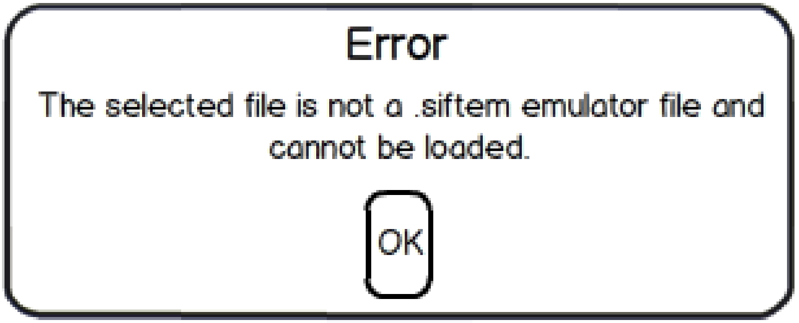
Caution

Reloading this program will clear all data from the previous program run. Proceed?



Error

The selected file is corrupt and cannot be loaded.



Error

The selected file is not a .siftem emulator file and cannot be loaded.

A Features

ID	Feature	Description	Reason
F1	Individual, virtual Sifteo Cube	A virtual representation of a single Sifteo Cube	Replicates physical Sifteo Cube
F2	Buttons to manipulate each virtual Cube	Buttons on the virtual Cube will allow the User to flip and tilt it	Replaces physical actions where said actions would be impractical with a mouse
F3	Workspace where multiple Cubes can be emulated	Multiple Cubes will be displayed on a workspace that replicates the free-form nature of physical Sifteo Cubes	Replicates multiple Sifteo Cubes in a natural, free-form environment
F4	Interactions between Cubes	The Cubes present on the workspace will communicate when they are neighbored	Cubes can simulate the interactions possible with physical Cubes
F5	Load programs into the Cubes	The User will load his own and example programs into the emulators Cubes	The ability to program programs for the emulator is dependent on a common interface
F6	Snap Cubes to invisible grid	The Cubes will snap into an invisible grid when a button is clicked	Increases productivity by allowing a quick reset if the Cubes are in disarray
F7	Zoom Workspace	The Workspace will zoom to the level of an individual Cube or the whole space	Inspecting individual Cubes allows for precise checks of program Graphical User Interfaces (GUIs)

Glossary

Graphical User Interface is a visual way of allowing the user to interace with a computer program. 9

Mac is a series of lines of personal computers developed by Apple. 5

Sifteo Cubes are small machines capable of loading programs and interacting with one another as well as responding to predefined movements. 3

Software Development Kit is a collection of tools designed to help build software for a particular platform. It may include an and an emulator of the target platform among other things.. 3

Windows is a series of operating systems developed by Microsoft. 5

References

1. Sifteo Inc. Online: http://www.sifteo.com

2. Tim Ekl. Client Meeting. 12 September 2011 12:45 p.m.

Index