Singularity Software
Milestone 4

October 25, 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 fourth 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 Sifteo Cubes — miniature computers that interact and communicate when used in tandem — 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 defines the standards for code in the Siftables Emulator project as well as the manner in which change will be controlled. It also elaborates the test cases that sufficiently cover the system. Future milestones will present design and usability reports 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 covers all 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 4 relies on previous milestones as it defines a change control plan, coding standards, and test cases. It follows Milestones 2 and 3, which laid the foundation and elaborated the requirements of the Siftables Emulator specification based on the high-level design created in Milestone 1. Milestone 5 will elaborate on the upcoming usability study comparing the proposed Siftables Emulator design to the Sifteo emulator's design before presenting original and feedback-based interface designs for the Siftables Emulator.

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 is intended to build on the foundation of Sifteo, Inc.'s existing emulator by creating a more fluid and natural user interface.

4 Coding Standards

The developers of Singularity Software will adhere to a standardized coding "style guide" based on the language chosen for development. In the case of Python, we will reference the official Style Guide for Python Code [6]. Highlights of that document include 4-space indentation, a lack of extraneous space, and camel case for class names. In the case of C#, where no Microsoft-standard coding standards exist, we will use the standards laid out at [7], that emphasize the exclusive use of camel and Pascal case and ANSI-style bracing (where each brace gets its own line).

5 Change Control

Change requests may be submitted as issues on the project's GitHub repository[1]. At a minimum, requests for change should mention the screen(s) to be changed, the new workflow that is desired (preferably in context with how it differs from the current workflow), the rationale for the change, and the priority the submitter assigns to the change.

Changes submitted in the manner described will be considered by Singularity during our weekly project meeting. A majority vote (three of four team members) is required to accept a new feature into the system; however, a team member who feels very strongly may exercise a veto on the team's decision. Each team member will receive one veto for every four submitted changes.

Changes to product documents (i.e. milestones) will be managed using the Git version control framework within which the documents are currently stored. In the case of major changes, the team will consider the use of "delta" documents if such documents will be easier to read than heavily modified original documents.

[1]https://github.com/alexmullans/Siftables-Emulator/issues/new

6 Test Cases

6.1 Load program

Scenario #	Originating Flow	Alternate Flow	Next Alternate
1	Basic flow		
2	Basic flow	Alternate flow 1	
3	Basic flow	Alternate flow 2	
4	Basic flow	Alternate flow 3	
5	Basic flow	Alternate flow 4	
6	Basic flow	Alternate flow 4	Basic flow

Test Case ID	Scenario	Description	Expected Result
1	1	A program is successfully loaded in the emulator.	The Cubes run the selected program in the Workspace.
2	2	User selects incompatible file type.	"The selected file is not a .siftem emulator file and cannot be loaded." error is displayed.
3	3	User selects corrupt or unloadable file.	"The selected file is corrupt and cannot be loaded." error is displayed.
4	4	User cancels loading program.	The emulator returns to its state before the basic flow was entered.
5	5	User loads a program when a program is already running and chooses "Cancel" on the warning prompt.	Emulator returns to its state before the basic flow was entered after "Cancel" is chosen.
6	6	User loads a program when a program is already running and chooses "Continue" on the warning prompt.	Cubes run the selected program in the Workspace after "Continue" is chosen.

6.2 Reload program

Scenario #	Originating Flow	Alternate Flow
1	Basic flow	
2	Basic flow	Alternate flow 1

Test Case ID	Scenario	Description	Expected Result
1	1	The program is successfully reloaded in the emulator.	The Cubes run the selected program in the Workspace after the user presses “Continue” on the warning message.
2	2	The user selects “Cancel” on the warning dialog (which specifies that the emulator will be reset).	The Cubes run the program as originally specified; no change should be made to emulator’s state.

6.3 Zoom screen

Scenario #	Originating Flow
1	Basic flow

Test Case ID	Scenario	Description	Expected Result
1	1	User zooms in fully.	The Workspace view is centered on the first cube with the edges of the adjacent Cubes visible.
2	1	User zooms out fully.	The entire Workspace is visible.
3	1	A zoom level in the middle is selected.	The slider moves to the closest predefined increment and the correct zoom level is shown.

6.4 Add/remove Cubes

Scenario #	Originating Flow
1	Basic flow

Test Case ID	Scenario	Description	Expected Result
1	1	The slider is moved to one of the predefined increments or the spinbox is changed to an integer in the range [1, 6].	The specified number of Cubes is shown in the Workspace.

6.5 Snap Cubes to grid

Scenario #	Originating Flow
1	Basic flow

Test Case ID	Scenario	Description	Expected Result
1	1	User presses the “Snap to Grid” button.	The emulator aligns the Cubes in a grid based on their current positions.

6.6 Manipulate Cube

Scenario #	**Originating Flow**
1	Basic flow
2	Alternate flow 1
3	Alternate flow 2
4	Alternate flow 3

Test Case ID	**Scenario**	**Description**	**Expected Result**
1	1	The user double clicks on a Cube.	The Cube responds as if a screen click occured.
2	2	The user clicks on one of the buttons on the Cube.	The Cube executes and responds to the associated action (flip, rotate, or tilt).
3	3	The user drags a Cube next to another Cube.	The Cube neighbors with the adjacent Cube, and the two recognize each other as neighbors.
4	4	The user drags a Cube back and forth in a shaking manner.	The Cube responds as if shaken.

A Features

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

B Use Cases

B.1 Load program

Name: Load program

Description: The User selects the program file to be loaded and run by the emulator.

Actors: User

Basic flow:

1. The User presses the “Load a program” button.
2. The User selects a *.siftem file in the file dialog.
3. The User presses “Open” button.
4. The emulator loads the selected program on the Cubes in the emulator.

Alternate flows:

When the User opens an incompatible file (i.e. any file without the .siftem extension),

1. An error dialog is presented to the User with the message: “The selected file is not a .siftem emulator file and cannot be loaded.”
2. The use case terminates and no program is loaded.

When the User opens a corrupt or otherwise unloadable file,

1. An error dialog is presented to the User with the message: “The selected file is corrupt and cannot be loaded.”
2. The use case terminates and no program is loaded.

When the User presses the “Cancel” button:

1. The use case terminates and no program is loaded.

When the User is already running a program,

1. A warning dialog is presented to the User with the message: “Loading this program will clear all data from the previous program run. Proceed?”
2. If the User presses “Yes” on the warning dialog, flow returns to Step 2 of the basic flow.

When the User presses “No” on the warning dialog:

1. The use case terminates and the program is not reloaded.

Pre-conditions:

1. The emulator is running.

Post-conditions:

1. The program is loaded or the User cancelled loading.

Special requirements

1. The emulator should indicate that loading the program is in progress.
2. The emulator should indicate when the program is finished loading.

B.2 Reload program

Name: Reload program

Description: The User reloads the program currently running in the emulator.

Actors: User

Basic flow:

1. A warning dialog is presented to the User with the message: ”Reloading this program will clear all data from the previous program run. Proceed?”
2. If the User presses “Yes” on the warning dialog, the Emulator loads the program onto the Cubes in the emulator.

Alternate flows:
When the User presses “No” on warning dialog:

1. The use case terminates and the program is not reloaded.

Pre-conditions:

1. A program is loaded in the emulator.

Post-conditions:

1. The program is reloaded or the current program state remains on the Cubes.

Special requirements:

1. The emulator should indicate that loading the program is in progress.
2. The emulator should indicate when the program is finished loading.

B.3 Zoom screen

Name: Zoom screen

Description: The User zooms the Workspace to the desired level.

Actors: User

Basic flow:

1. The User adjusts the zoom slider.
2. The Emulator magnifies the Cubes in the emulator according to the zoom level.

Alternate flows:
None

Pre-conditions:

1. The emulator is running.

Post-conditions:

1. The program running at the beginning of this use case, if any, is still running.

Special requirements:

1. The zoom slider moves in discrete increments. The lowest (and default) level shows the whole workspace and the highest level shows one Cube with the edges of the surrounding Cubes visible for context.

B.4 Add/remove Cubes

Name: Add/remove Cubes

Description: The User adjusts the number of Cubes present in the emulator.

Actors: User

Basic flow:

1. The User drags the "Number of Cubes" slider or uses the up/down arrows on the spinbox to increment or decrement the number of available Cubes by one.
2. The emulator adds/removes Cubes in emulator and resets its Workspace.
3. If Cubes are to be removed, the emulator starts with the bottom right-most of the Cubes (at their current positions) and works left. If more Cubes are to be removed after the second row is depleted, the emulator again starts at the bottom right-most of the remaining Cubes.

Alternate flows:
None

Pre-conditions:

1. The emulator is running.

Post-conditions:

1. The number of Cubes has been adjusted to the number specified.
2. The running program, if any, is terminated.

Special requirements:

1. The "Number of Cubes" slider moves in discrete increments. The leftmost level shows one Cube and the rightmost level shows six Cubes.

B.5 Snap Cubes to grid

Name: Snap to grid

Description: The Users pulls the Cubes into a grid orientation.

Actors: User

Basic flow:

1. The User presses the “Snap to Grid” button.
2. The Emulator moves the Cubes to a grid orientation based on their current positions. It will maintain the Cubes’ positions relative to other Cubes while doing so.

Alternate flows:
None

Pre-conditions:

1. The emulator is running.

Post-conditions:

1. The Cubes are arranged in a grid.

B.6 Manipulate Cube

Name: Manipulate Cube

Description: The User manipulates a Cube by clicking the buttons or the Cube itself.

Actors: User

Basic flow:

1. The User double clicks on a Cube.
2. The Cube responds as if a screen click occured.

Alternate flows:

1. The User clicks on one of the buttons superimposed on the Cubes’ edges.
2. The Cube executes the appropriate action (i.e. flips, rotates, or tilts).

1. The User drags a Cube next to another Cube.
2. Cube communicates (“neighbors”) with the Cube(s) it is adjacent to.

The User “shakes” a Cube back and forth with the mouse (i.e. he laterally moves the Cube back and forth several times).

The Cube responds as if shaken.

Pre-conditions:

1. The emulator is running.

Post-conditions:

1. If a program is running, the emulator has updated its state based on the Cube's change.

B.7 Other functional requirements

B.7.1 API

The emulator will include an API in order to define a set of rules and specifications via which Cube programs can be created.

B.7.2 Example games

The emulator will include games as examples that demonstrate to the User how the Cubes interact with each other.

C Feature Mapping

ID	Name	Features
U1	Load program	F5
U2	Reload program	F5
U3	Zoom screen	F7
U4	Add/remove Cubes	F3, F4
U5	Snap Cubes to grid	F6
U6	Manipulate Cube	F1, F2, F3
OR1	API	F2
OR2	Example games	F3

Glossary

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

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

usability study is a manner of evaluating the design and user experience of a product by testing it on users. 3

References

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

2. Tim Ekl. Client Meeting. 4 10 October 2011 4:15 p.m.

3. Milestone 1. Singularity Software. Online: https://github.com/alexmullans/Siftables-Emulator/blob/master/docs/pdfs/m1.pdf

4. Milestone 2. Singularity Software. Online: https://github.com/alexmullans/Siftables-Emulator/blob/master/docs/pdfs/m2.pdf

5. Milestone 3. Singularity Software. Online: https://github.com/alexmullans/Siftables-Emulator/blob/master/docs/pdfs/m3.pdf

6. Guido van Rossum. “PEP 8 Style Guide for Python Code.”
   Online: http://www.python.org/dev/peps/pep-0008/

7. “C# Coding Standards Document.”
   Online: http://weblogs.asp.net/lhunt/pages/CSharp-Coding-Standards-document.aspx

Index