Milestone 5
Singularity Software
May 11, 2012

Integration Testing

Without true integration tests, we are using example applications to drive integration testing of our emulator. No one application includes each available portion of our emulator, but complete coverage is attained by testing each of the three example applications.

CubeTest: CubeTestApp is a simple (read "not fun") game which responds to cube manipulations by changing the background color of the manipulated cube. This application exploits the following functionalities:

- Filling background color
- Cube tilting (all four directions)
- Cube screen press
- Cube flip
- Cube shake
- Pause application event handling
- Resume application event handling

FractionOrdering: FractionOrderingApp is a fun (and educational!) game that serves as an entertaining way to learn fractions. Each cube displays a unique, nonzero fraction. The player must arrange the cubes from left to right such that the fractions are in an increasing order. When they are successful at arranging two cubes relatively, they are met with a confirmation message, and when all of the cubes are ordered correctly, a reassuring vote of confidence is played. However, when cubes are placed in an incorrect order, the user is confronted with a failure message. This application exploits the following functionalities:
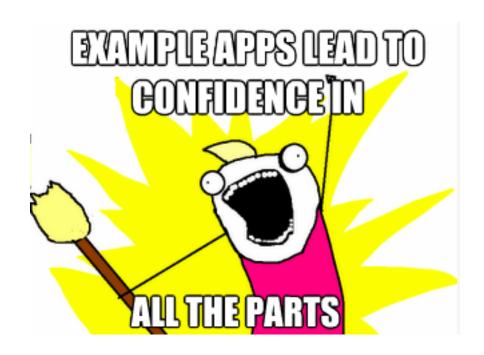
- Filling background color on a cube
- Displaying images on cubes
- Playing sounds in the emulator
- Checking cube neighbors
- Drawing rectangles on the cube

WordMaker: WordMakerApp is a fun (and educational!) game that serves as an entertaining way to learn new words or practice spelling. Each cube displays a letter. The player must arrange the cubes from left to right such that the letters spell out an English word. Successes and failures are met with responses like in the FractionOrderingApp. This application exploits the following functionalities:

- Filling background color on a cube
- Displaying images on cubes

- Playing sounds in the emulator
- Checking cube neighbors

These integration tests cover the major functionality of the Sifteo cubes as integrated into our emulator. Other features, like pausing/resuming, exception handling, and zooming, are tested on the emulator only and don't require integration testing with the applicaations. In essence,

Code Maintenance Plan

Likely changes

The emulator must support changes to the Sifteo API which may require changes in the Sifteo package or the Siftables ViewModel implementations. However, if the change is limited to the Sifteo.Util and/or Sifteo.MathExt namespaces, the Sifteo.dll file can simply be decompiled with the necessary pieces recompiled into SifteoExtensions.dll (the Siftables team was previously given permission to do so by Sifteo, Inc.). The sample applications provided with the project should not require any changes unless an API-breaking change occurs.

Variation points

Since the Emulator is dependent on the Sifteo API, there are no variation points in the emulator project. However, it is encouraged that the client alter and observe the code for the sample applications in order to better understand how to approach development on the Sifteo cubes.

Future code maintenance

Part of the clients specification was that the code be licensed with the BSD 2-clause license. Hence, the source code will remain available at https://github.com/alexmullans/Siftables-Emulator. Other parties, including the client, can clone this GitHub repository and modify the project as they see fit. The final Singularity Software release of the Siftables Emulator project will be tagged in the repository for reference purposes.

Sprints 4 Backlogs

The following pages show the backlog for the previous sprint (sprint 4).

Backlog	(None)	In Progress	Completed	Summary
S–01024 Example game: Reflex game Ethan 6.00		Implement Reflex game Ethan 3.00		Test Results: To Do: 3.00
S–01027 Documentation: Code Maintenance Plan `Accepted` Richard 3.00			Write Code Maintenance Plan Richard 0.00	Test Results: To Do: 0.00
S–01028 Documentation: All Deliverables –> ANGEL `Accepted` Alex 0.50			Submit Documentation Alex, Kurtis, Ethan, Richard 0.00	Test Results: To Do: 0.00
S–01030 Emulation: Pause/Resume Application `Accepted` Kurtis 5.00			Add UI Pause/Resume Button Kurtis 0.00 Implement Pause/Resume Model Events/Functionality Kurtis 0.00	Test Results: To Do: 0.00
S–01031 Development: Continuous Integration Server `Done` Alex 5.00			Install CI Server Alex 0.00 Configure CI Builds Alex 0.00 Configure CI Test Runs Alex 0.00	Test Results: To Do: 0.00
S–01032	Test Application Richard		Build Application Richard	

~~Example Game: Word Ordering~~ [Accepted] Richard 5.00	0.00		0.00	Test Results: To Do: 0.00
S-01033 ~~Emulation: Exception Handling~~ [Accepted] Alex, Kurtis 3.00			~~Add UI Exception Reporting Element~~ Alex 0.00 ~~Catch Exceptions in ViewModel~~ Kurtis 0.00	Test Results: To Do: 0.00
S-01034 ~~Documentation: MS5~~ [Accepted] Alex, Kurtis, Ethan, Richard 3.00			~~Write Milestone~~ Alex, Kurtis, Ethan, Richard 0.00	Test Results: To Do: 0.00
S-01035 ~~UI: Refine Zoom~~ [Done] Alex 3.00			~~Refine zoom~~ Alex 0.00	Test Results: To Do: 0.00