Trevor Krenz and Ethan Veatch

Project Component 2

Online Repository

The existing source code for our project has been uploaded to https://github.com/veatchje/Siftables-477.git, a public GitHub repository.

Modifiability

There are no obvious improvements which can be made to the project with regards to modifiability. The MVVM architecture was closely followed during development, which led to a system with loose coupling and high module level cohesion. The dynamic application loading provides more than adequate deferred binding and the high level of cohesion ensures relatively small module size.

Usability

Support User Initiative:

- Cancel: The user can cancel the application loading dialog, but no other cancel is possible or applicable.
- Undo: The “Snap to Grid” functionality acts as an undo for repositioning all the cubes, but is not specifically an undo function as it goes to the initial position rather than stepping back one movement at a time.
- Pause/Resume: Applications can be paused and resumed during their run process.
- Aggregate: One application is loaded onto all cubes at one time, with the number of cubes being dynamic.

Support System Initiative:

- Maintain Task Model: Rotation can only occur in 90 degree steps.
- Maintain User Model: The user can control the number of cubes displayed on screen.
- Maintain System Model: Cubes can tell which other cubes are close to them.

Improvements:

1. Allow user to change background color.
2. Reduce tendency for cube rotation to render cube graphically off axis.

Project Component 3

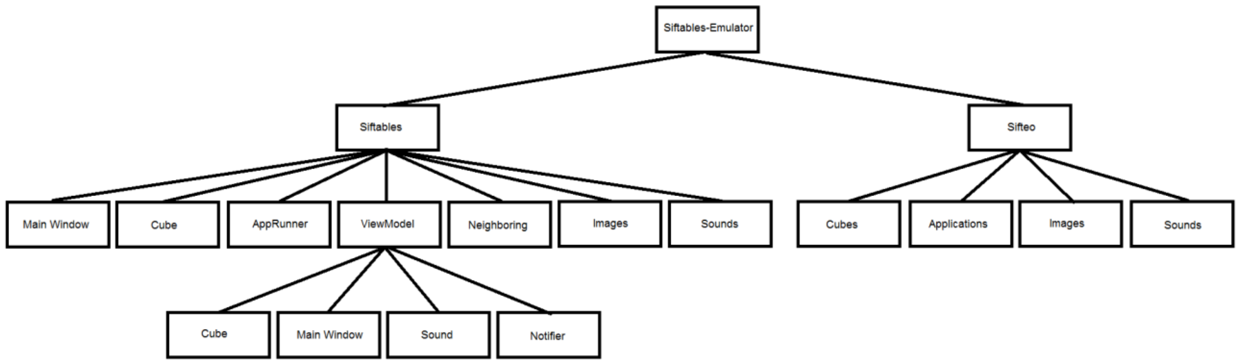
Testability

Tactics

1. Specialized interfaces – All important variables can have getter/setter and cubes can return their state through various levels of feedback. Data can be reset by unloading the program in the cubes.
2. Record/playback – It is difficult to use record/playback because much of the testing involves states of a cube, which is reliant upon the other cubes within the set. Simply stated, there are too many entities in control to record the state at each control switch with any reliability.
3. Localize state storage – The system is always started with no program loaded into the cubes, allowing easy testing of the emulator space independent of a program. Additionally, each application starts from a given state as defined by startup procedure.
4. Abstract data sources – The BaseApp class is an abstract class implemented by all of the applications written for the cubes in the emulator. Testing can be done using BaseApp rather than testing each application individually.
5. Sandbox – The purpose of the project was to create an emulator, which is in a sense a sandbox for the cubes. The program itself allows the cubes to run in a sandbox environment, but developing a way to sandbox our emulator seems virtually impossible. As stated before, when the emulator starts it operates totally independent of any outside influence, this is as close to a sandbox as we could come.
6. Executable assertions – Executable assertions occur throughout the program in the form of value checks on many variables during method execution. If the assertion fails an error message is displayed or other appropriate actions are taken.
7. Limit structural complexity – Limiting the structural complexity of this program is difficult because of the challenges it faces. In general the project has high cohesion and loose coupling, but in some situations this was not enough to ensure limited complexity. Issues such as dynamic loading necessitate a certain amount of complexity.
8. Limit non-determinism – With the exception of dynamic class loading, there is no non-deterministic behavior in the program. The information flows in a well-defined path and is, in most cases, event driven. Applications that are loaded into the cubes are started at a specific point in program flow and then occur whole within their own thread.

Module Decomposition Diagram

There were no changes needed to the module decomposition diagram.


Siftables-Emulator
Siftables
Sifteo
Main Window
Cube
AppRunner
ViewModel
Neighboring
Images
Sounds
Cubes
Applications
Images
Sounds
Cube
Main Window
Sound
Notifier