*Florida International University*

*School of Computing and Information Sciences*

CIS 4911 - Senior Capstone Project

Software Engineering Focus

Feature Document

User Story #731

**Team Member:**

Andrew Mitchell

**Product Owner(s)**:

Francisco R. Ortega

**Mentor(s)**:

Francisco R. Ortega

...

**Instructor**: Masoud Sadjadi

**User Story –** Implement Drawui Improvements.

* As a User I would like a smooth and lag-less experience even opening many UI elements so I can have a nice smooth experience.

Acceptance Criteria

* Move all the buttons and UI Draw calls to individual frame buffer objects.
* Only draw the framebuffers in ‘DrawUI’ instead of every individual icon and line.

**Use Case - View User Interface**

Use Case

User launches application.

Details:

Actor: User

Pre-conditions:

Program must be running.

Description:

Use case begins when the user launches the application. After the application is launched it should perform a lot less draw calls (Pre draw all the buttons onto frame buffers).

Post-conditions:

The Program should run at a much higher framerate with all the menus open.

Decision Support:

Frequency: Very Often. This will improve performance any time a UI element is drawn.

Criticality: Very High. Certain devices already strain the system. Improving performance is critical anywhere we can muster.

Risk: Medium. Setting up all the Frame Buffers and making sure they draw properly may take a while.

Reliability: Highly.

Mean time to Failure – N/a. Icons should be available when they are called.

Availability – Always running in the background.

Performance:

Should have much lower performance impact when opening all the menus.

Supportability:

Entire Program and all devices.

Supported by LibUSB – version 1.0

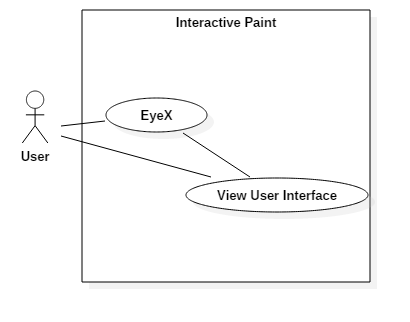
Modification History:

Owner: Andrew Mitchell

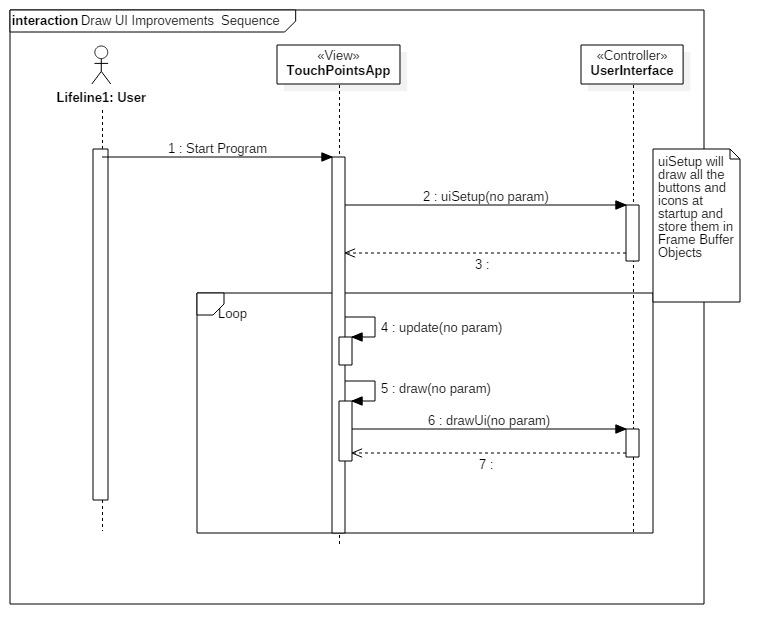
Initiation date: 4/18/2016

Date last modified: 4/29/2016

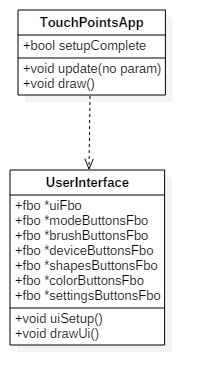
**Use Case Diagram**



**Sequence Diagram**



**Class Diagram**



**Unit Test**

Sunny Day:

Test Case: Open UI Menus

Test Purpose:

Ensure the UI elements can still open up and display all the correct buttons.

Test Setup

1. Press Colors button
2. Press Shapes Button
3. Press Device modes button
4. Press Settings Button
5. Repeat steps 1-4

Test Output:

The first repetition of step 1-4 resulted in opening all appropriate menus.

Repeating the steps closed them all.

Expected Output:

The menus should open up just as they did before this user story. The colors buttons, shapes button, device modes button, and settings button should all toggle open or closed with a button pressed.

Rainy Day Tests:

Test Case: All Menus Open Performance Test

Test Purpose:

Ensure that the UI changes have improved performance, even when trying to open all of our menus!

Test Setup:

1. Press settings button
2. Press ‘Frames Per Second’
3. Press settings button
4. Record FPS
5. Press Settings button
6. Press Colors Button
7. Press Shapes Button
8. Press Device Modes Button
9. Record FPS

Test Output:

Step 4 : FPS 49~

Step 9 : FPS 45~

Expected Output:

FPS Drop should be substantially lower (Pre-User story implementation frame drop was around 30~ FPS). About a 4-5 FPS drop should be normal.

**Integration Test**

Currently improved the performance of our entire program.

Actively increased the frame rate of the idle program with all menus open from 22 to 46 FPS.

Buttons are currently integrated with the new Brush Menu as of 4/25/2016.

**User Guide**

Here we can see the 4 menus opened up. The framerate is at 22 FPS with an idle program, with all the menus open.



Currently after implementation we have a much better framerate of 45 FPS, including one additional menu that was recently added. While the Idle program (no menus) runs at around 49 FPS on the testing computer.



**Glossary**

N/a