*Florida International University*

*School of Computing and Information Sciences*

CIS 4911 - Senior Capstone Project

Software Engineering Focus

Feature Document

### User Story #882 Add menu system eventing.

**Team Member:**

Alexander Karpis

**Product Owner(s)**:

Francisco Ortega

**Mentor(s)**:

Francisco Ortega

**Instructor**: Masoud Sadjadi

**User Story**

* As a developer I want to be able to add menus that contain how they should be draw and how they should behave inside of them in order to be able to separate out the responsibilities of each menu.

**Acceptance Criteria**

1. Menus now work with callbacks in response to events
2. Color picker menu still works
3. Shape picker menu still works

**Use Case** #882 **– Add menu system eventing.**

Actors: Developer**.**

Entry Condition:

This use case start when:

* When a developer adds a menu, it should be a stand alone object that contains its own visuals and behaviors.

Flow of events:

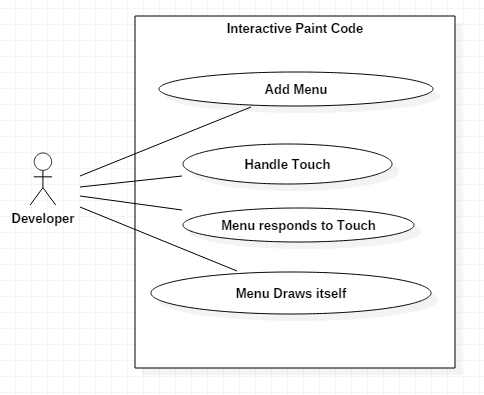
1. The developer adds a new menu to the UI

2. When the user taps on the correct location it should trigger the appropriate behavior

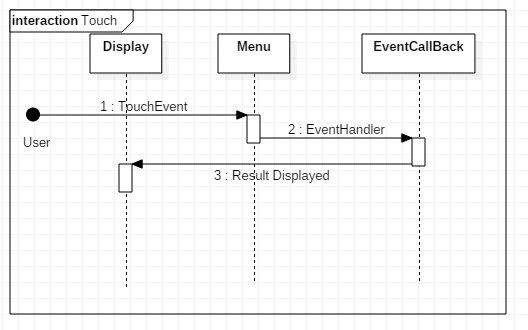
Exit Condition:

The callback is called in response to the touch event

**Use Case Diagram**



**Sequence Diagram**



**Class Diagram**

|  |
| --- |
|  |

**Unit Test**

Sunny Day Tests

Test Case: Menu responds to touch event

Test Purpose: Ensure menu can respond to touch event…

Test Setup:

⦁Run program.

Test Output:

Touch event does what it is supposed to do when touched.

Expected Output:

After touching the screen in the correct location the correct menu triggers the correct event.

**Integration Test**

Menus can interact with each other