*Florida International University*

*School of Computing and Information Sciences*

Software Engineering Focus

Feature Document

User Story ID #692 Connection Design For Mouse

**Name: Pachev Joseph**

**Team Member(s): Bernardo Pla, Daniel Rivero, Daniel Khawand,**

**Project: WebVR 1.0**

**Product Owner(s)**:

**Mentor(s)**: Francisco Ortega

**Instructor**: Masoud Sadjadi

**User Story Name: Design Connection For Mouse**

* As a developer I want to connect a mouse to a computer and have universal way of representing its connection and receiving output.

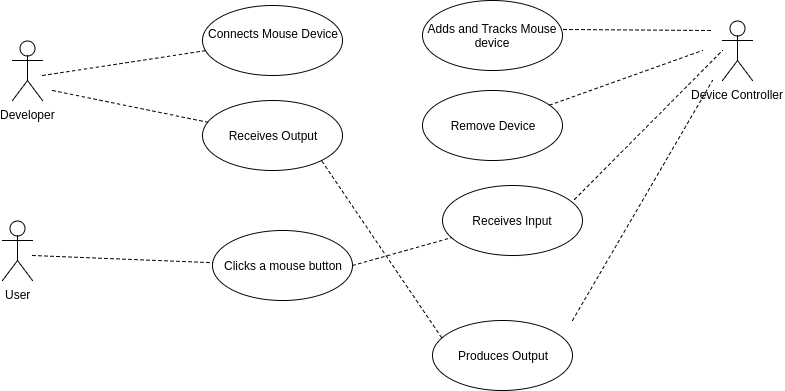
Acceptance Criteria

* Input can be read by device that it is connected to
* Connection is established and represented as a tuple
* Data output is sent in a universal JSON format

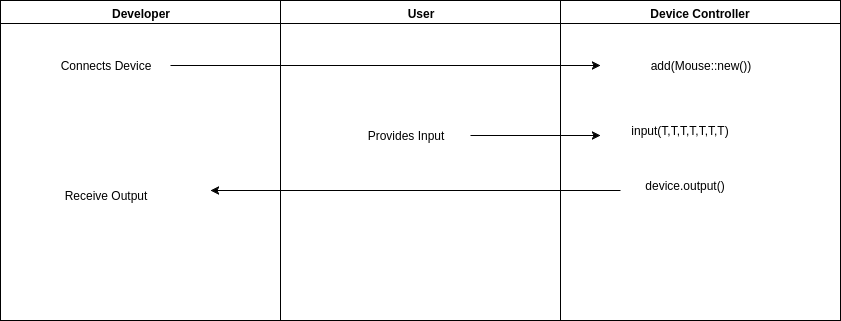
**Use Case**

* Name: Connection of Mouse
* Actor:Developer, User, Device Controller
* Preconditions:Webvr-input has started and is listening for devices
* Description :
  + Developer connects mouse
  + Device controller reads and adds mouse
  + User click(provides input)
  + Device controller receives input and provides appropriate output to developer.

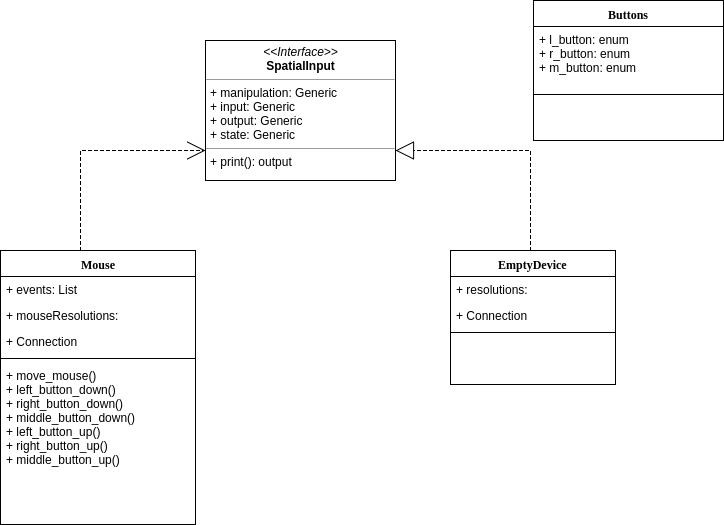
**Use Case Diagram <**you can use draw.io**>**



**Sequence Diagram**



**Class Diagram**



**Test case ID: WV-MO-IN001: Sunny**

* **Description/Summary of Test**: The developer will launch the demo class and record a left click, a right click, and a center click on different points of the screen.
* **Pre-condition**: The main class in the rust library is running. A mouse must be connected to the machine.
* **Expected Results**: When the mouse is connected, a left click will print “Left button down.” A right click will print “Right button down”. A center click will print “Middle button down”. The output of these will be recorded on a terminal.
* **Actual Result**: The terminal read “Left button down” on a left click event. It read “Right button down” on a right click event. It read “Middle button down” on a center click event. The output was recorded on a terminal thus proving the events were being handled as expected.
* **Status** (Fail/Pass): Pass

**Visual User Guide** <like one or two screenshots of the feature. For the hardware project, a photo of device is required>

