

# Multi touch support implementation in Pharo

François LEPAN  
Benjamin VAN RYSEGHEM

1<sup>er</sup> mars 2013

## Introduction

During our project our main goal was to introduce a new way to handle multi touch events and gesture in Pharo. Starting from what we previously did in last semester during the PJE lecture, we used the same approach to introduce a handling of gestures based on TUIO with an architecture allowing to switch to virtual machine events.

In order to analyse these gestures we needed a state machine that would fill the gap between the human gesture and a perfect gesture.

Another goal was to clean and reunify the whole hierarchy of system events and to provide a clean abstraction of low level data structure.

## Table des matières

<b>1</b>	<b>TUOI and blobs analysis</b>	<b>3</b>
<b>2</b>	<b>System events hierarchy</b>	<b>3</b>
<b>3</b>	<b>Switching to VM events</b>	<b>3</b>
<b>4</b>	<b>State machine limits</b>	<b>3</b>
<b>5</b>	<b>Gesture implementation</b>	<b>3</b>
<b>6</b>	<b>Conclusion</b>	<b>3</b>
<b>7</b>	<b>Appendices</b>	<b>3</b>

## 1 TUOI and blobs analysis

Our first goal is to be able to retrieve gesture from the user. In order to do so we used TUIO which is a protocol for multitouch events. It retrieve the multi-touch gesture from the user on a surface and then generate events for each finger on the surface (*cf* Fig. 1). For our tests we used a small software called Tongseng that generate TUIO events from the trackpad and a small library created by our tutor that parse the TUIO events.

## 2 System events hierarchy

## 3 State machine limits

## 4 Gesture implementation

## 5 Switching to VM events

## 6 Conclusion

## 7 Appendices

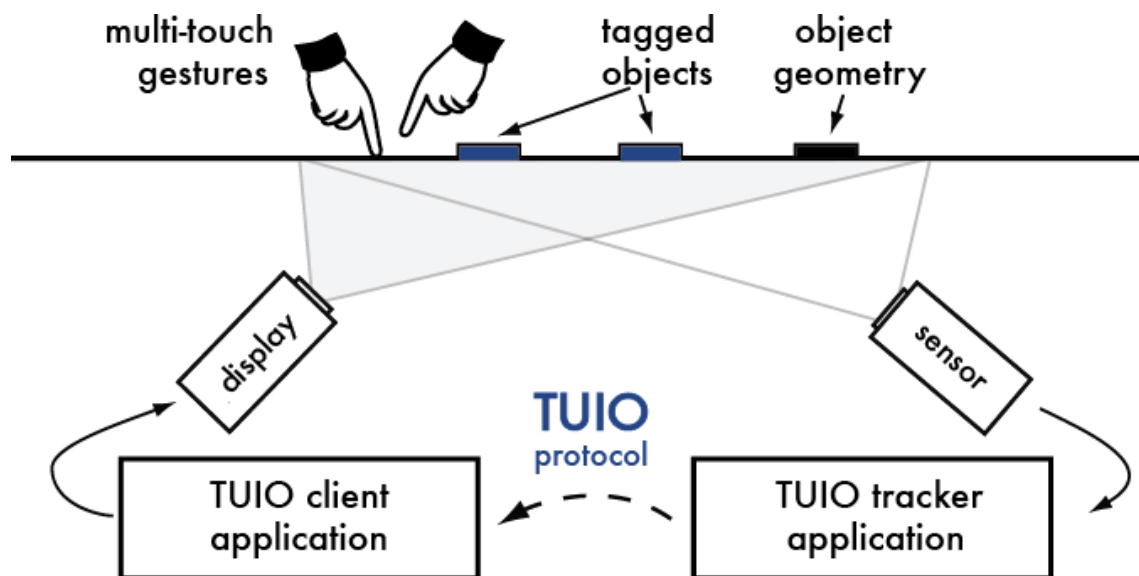


FIGURE 1 – Diagram of the TUIO protocol