

The Simple Multi-Touch Toolkit

Zachary Cook

vialab • Faculty of Science • University of Ontario Institute of Technology

zachary.cook@uoit.net

Erik Paluka

erik.paluka@uoit.ca

Christopher Collins

christopher.collins@uoit.ca

Mark Hancock

University of Waterloo

mark.hancock@uwaterloo.ca

Simple Multi-Touch (SMT)

With the Simple Multi-Touch Toolkit (SMT), we aim to make it easier to start prototyping and developing with multi-touch by:

- Extending the Processing Development Environment [1], designed for non-programmers, which simplifies installation, coding, and executing of code.
- Providing ready to use multi-touch UI components (Figure 2), along with low and high level Touch interaction primitives.
- Instead of formal classes, we only require implementing named methods, just as draw() and setup() are used in Processing.

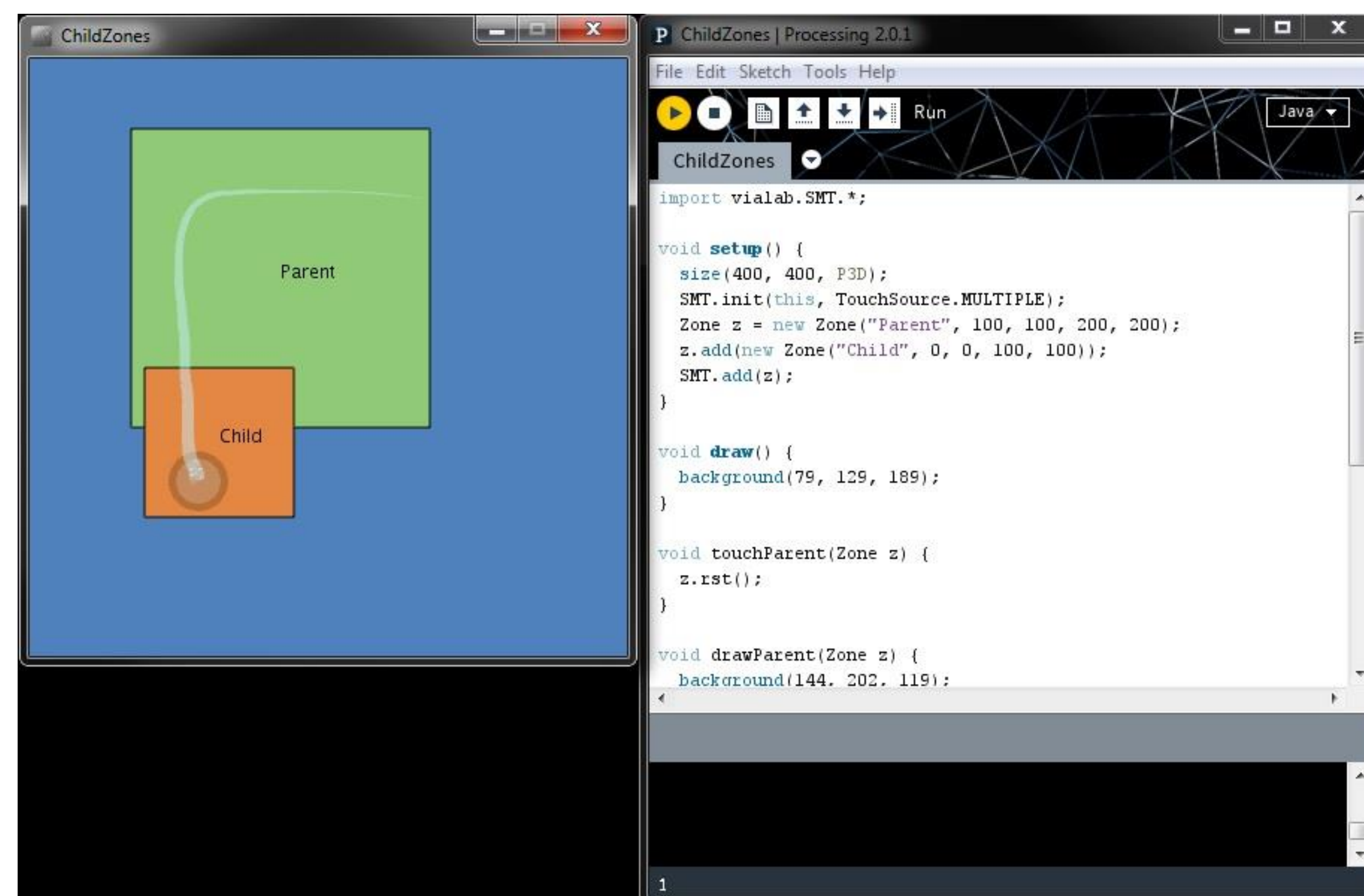


Figure 1. An example running alongside the Processing Development Environment.

Hardware Support

SMT supports a wide variety of operating systems and multi-touch hardware with little to no setup, these include:

- Windows touch devices (Touch Screens)
- Leap Motion
- SMART tables
- Mouse to emulate touch (Windows, Mac OS X, Linux)
- Any hardware with a TUIO [2] tracker (Microsoft Surface, etc)
- Planned Android support

Multi-Touch UI Components

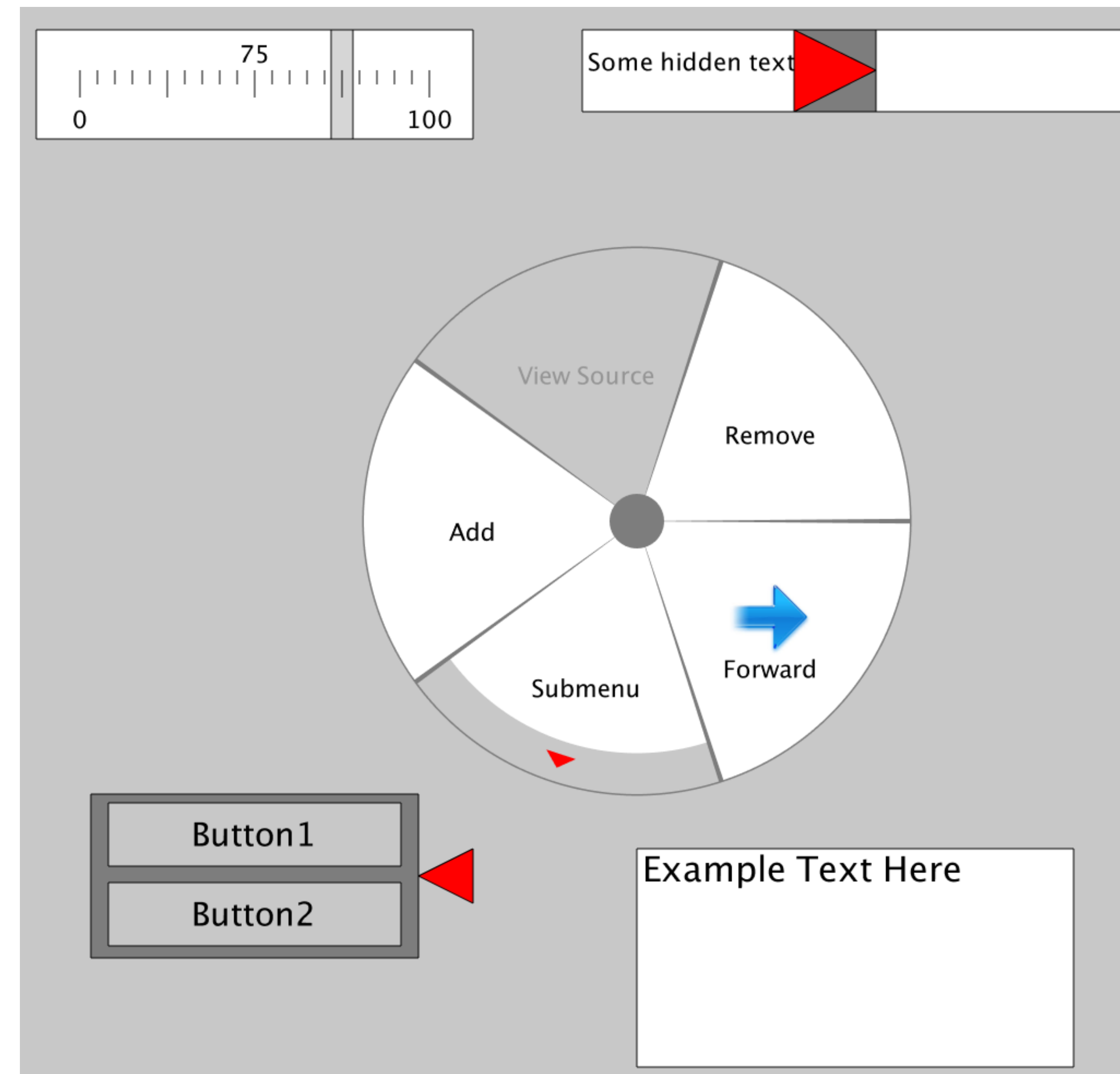


Figure 2. Built-in UI components which support multi-touch are created using Zones and can be customized.

Zones

SMT's central object is the Zone. It provides features such as:

- Custom drawing using Processing rendering commands [3].
- Built-in methods for specifying touch manipulations such as drag, rotate, scale and their combinations.
- Nesting of sub-zones to create components with spatial relations independent of rotation and scaling.
- Built-in 2D physics engine for supporting tossing of virtual objects across a multi-touch surface.

Sample Login Interface Code

```
1. import vialab.SMT.*;
2. Zone text, kBoard;

3. void setup(){
4.   size(800, 600, P3D);
5.   SMT.init(this, TouchSource.MULTIPLE);
6.
7.   kBoard = new KeyboardZone(25, 300);
8.   text = new TextZone("text", 125, -150, 150, 50, 32);
9.   kBoard.add(text);
10.  SMT.add(kBoard, new ButtonZone("LoginButton",
11.  550, 150, 150, 50, "Login", 32));
12.}

13.void pressLoginButton(Zone self){
14.  SMT.remove(self);
15.  SMT.remove(kBoard);
16.}

17.void draw(){
18.  background(150);
19.}
```

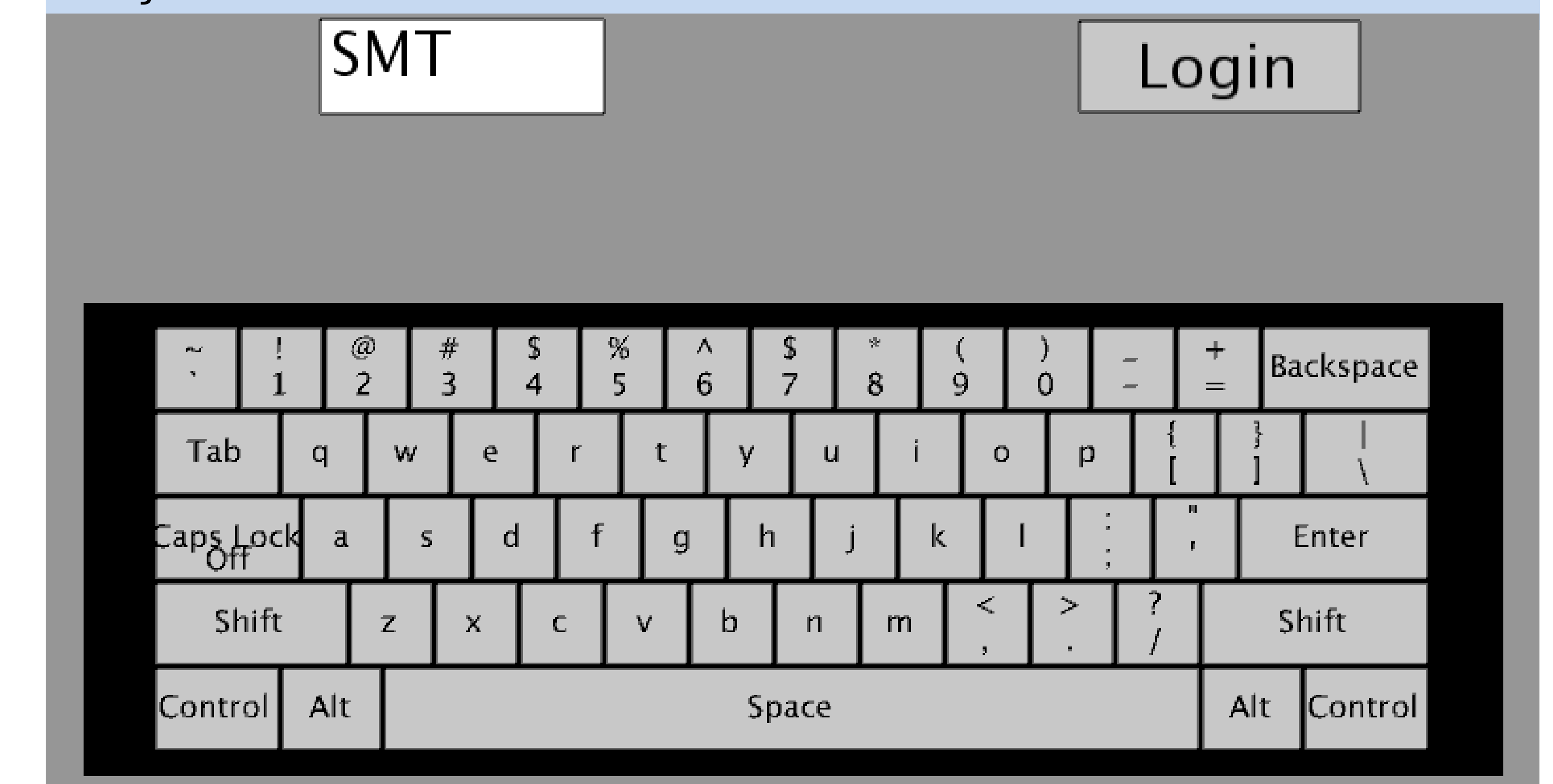


Figure 3. The interface generated by the above code.

Future Work

Our toolkit will be used and formally evaluated in HCI classes at UOIT and at the University of Waterloo in the 2013 fall term. SMT is open source and available online at <http://vialab.science.uoit.ca/smt/>

In the future, we would like to add:

- Named gesture event methods
- Support for Processing's Android mode

References

- [1] Processing website. August 2013. <http://www.processing.org>
- [2] TUIO software website. August 2011. <http://www.tuio.org/?software>
- [3] Processing API reference. August 2013. <http://www.processing.org/reference>