**Touchless screen using Leap motion**

**Project Report**

Human-Computer Interaction

Dr. Montri Phothisonothai

59090011 | Napat Traikityanukul

**Introduction 2**

Project description

Purpose

**Hardware 2**

**Software 3**

**Source Code 4**

**Project description**

The touch screen display provides a greater flexibility to user but after some years touch screen display becomes less sensitive which causes failure of touch on touch screen display. If we use a screen protector still dirty marks present on the display to avoid this problem a simple user interface for touch less control of electrically operated equipment is being developed.

**Propose**

By doing this project to study how Leap Motion works and apply to use it instead of touch screen

**Hardware**

* 1x Screen (laptop)
* 1x [Leap](https://www.adafruit.com/products/65) motion

**Software**

**Java**

**import** com.leapmotion.leap.\*;

**import** com.leapmotion.leap.Gesture.Type;

For connect java with leap motion.

**import** java.awt.Dimension;

To get screen dimension.

**import** java.awt.Robot;

To make leap motion be a robot to control a screen.

**import** java.awt.event.InputEvent;

To create command for a robot(leap motion).

**Source Code**

**import** com.leapmotion.leap.\*;

**import** com.leapmotion.leap.Gesture.Type;

**import** java.awt.Dimension;

**import** java.awt.Robot;

**import** java.awt.event.InputEvent;

**class** CustomListener **extends** Listener{

**public** Robot robot;

**public** **void** onConnect(Controller c) {

c.setPolicy(Controller.PolicyFlag.***POLICY\_BACKGROUND\_FRAMES***);

c.setPolicy(Controller.PolicyFlag.***POLICY\_IMAGES***);

c.setPolicy(Controller.PolicyFlag.***POLICY\_OPTIMIZE\_HMD***);

c.enableGesture(Gesture.Type.***TYPE\_CIRCLE***);

c.enableGesture(Gesture.Type.***TYPE\_SCREEN\_TAP***);

c.enableGesture(Gesture.Type.***TYPE\_SWIPE***);

}

**public** **void** onFrame(Controller c) {

**try** { robot = **new** Robot(); } **catch** (Exception e) {}

Frame frame = c.frame();

InteractionBox box = frame.interactionBox();

**for**(Finger f: frame.fingers()) {

**if** (f.type() == Finger.Type.***TYPE\_INDEX***) {

Vector fingerPos = f.stabilizedTipPosition();

Vector boxFingerPos = box.normalizePoint(fingerPos);

Dimension screen = java.awt.Toolkit.*getDefaultToolkit*().getScreenSize();

robot.mouseMove((**int**) (screen.width \* boxFingerPos.getX()), (**int**) (screen.height - boxFingerPos.getY() \* screen.height));

}

}

**for** (Gesture g: frame.gestures()) {

**if**(g.type() == Type.***TYPE\_SCREEN\_TAP***) {

robot.mousePress(InputEvent.***BUTTON1\_DOWN\_MASK***);

robot.mouseRelease(InputEvent.***BUTTON1\_DOWN\_MASK***);

}

**else** **if**(g.type() == Type.***TYPE\_CIRCLE***) {

CircleGesture circle = **new** CircleGesture(g);

**if**(circle.pointable().direction().angleTo(circle.normal()) <= Math.***PI*** /4) {

robot.mouseWheel(1);

**try** {Thread.*sleep*(30);}**catch**(Exception e) {}

}**else** {

robot.mouseWheel(-1);

**try** {Thread.*sleep*(30);}**catch**(Exception e) {}

}

}

}

}

}

**public** **class** LeapMouse {

**public** **static** **void** main(String[] args) {

CustomListener l = **new** CustomListener();

Controller c = **new** Controller();

c.addListener(l);

**try** {

System.***in***.read();

}**catch** (Exception e) {}

c.removeListener(l);

}

}