

## ผนวก ก.

### ภาษาคอมพิวเตอร์ที่ใช้ในระบบต้นแบบ

/\*

Rom Khampanya & Rapee Photichai (Edited)

TUIO processing demo - part of the reacTIVision project

<http://reactivision.sourceforge.net/>

Copyright (c) 2005-2008 Martin Kaltenbrunner <mkalten@iua.upf.edu>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software

Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

```
// we need to import the TUIO library
```

```
// and declare a TuioClient variable
```

```
import tuio.*;
```

```
TuioClient tuioClient;
```

```
// these are some helper variables which are used
```

```
// to create scalable graphical feedback
```

```
float cursor_size = 15;
```

```
float object_size = 60;
```

```
float table_size = 760;
```

```
float scale_factor = 1;
```

```
PFont font;
```

```
PImage a;
```

```
PImage b;
```

```
PImage c;
```

```
int fiducial_id;
```

```
// load text variable
```

```
String[] lines;
```

```
int index = 0;
```

```

String[] name = new String[4];
int[] cost = new int[4];
int[] area = new int[4];
//count v
int rdnum[] = new int[8];
int rdsum = 0;
int hnum[] = new int[8];
int hsum= 0;
int gnum;
int gsum= 0;
int cnum;
int csum= 0;
int allarea = 4000;
//end count v
//value
int costsum = 0;
int areasum = 0;
int waste = 0;
int percent = 0;
int gren=0;
int pergren=0;
//end value calculte
void setup()
{
  size(920,750);
  noStroke();
  fill(0);
  loop();
  hint(ENABLE_NATIVE_FONTS);
  font = createFont("Arial", 14);
  scale_factor = height/table_size;
  // an instance of the TuioClient
  // since we ad "this" class as an argument the TuioClient expects
  // an implementation of the TUIO callback methods (see below)
  tuioClient = new TuioClient(this);
  a = loadImage("6.jpg");
  b = loadImage("tree.jpg");
  c = loadImage("club.jpg");
  lines = loadStrings("data.txt");
}

```

```

}

// within the draw method we retrieve an array of TuioObject and TuioCursor
// from the TuioClient and then loop over both lists to draw the graphical feedback.

void draw()
{
  background(255);
  textFont(font, 18*scale_factor);
  float obj_size = object_size*scale_factor;
  float cur_size = cursor_size*scale_factor;
  fill(#00B464);
  rect(0,0,920,50);
  fill(0);
  text("TANGIBLE USER INTERFACE FOR URBAN PLANNING PROTOTYPE",120,30);
  fill(#2CFF9A);
  rect(0,0,100,750);
  fill(0);
  text("Project ",0,20);
  text("Information",0,40);
  fill(#A7A7A7);
  rect(100,50,90,750);
  fill(0);
  fill(#A7A7A7);
  rect(100,50,920,100);
  fill(0);
  //load data ID
  if (index < lines.length) {
    String[] pieces = split(lines[index], '\t');
    if (pieces.length == 3) {
      name[index] = pieces[0];
      cost[index] = int(pieces[1]);
      area[index] = int(pieces[2]);
    }
    index = index + 1;
  }
  //end load data
  TuioObject[] tuioObjectList = tuioClient.getTuioObjects();
  for (int i=0;i<tuioObjectList.length;i++) {
    TuioObject tobj = tuioObjectList[i];
    int fiducial_id= +tobj.getFiducialID());

```

```
text(name[0],0,250);
text(name[1],0,230);
text(name[2],0,270);
text(name[3],0,290);
if (fiducial_id == 30)
{
  pushMatrix();
  translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
  //rotate(tobj.getAngle());
  fill(0);
  rect(-obj_size+50,-obj_size/2,20,730);
  popMatrix();
  rdnum[0] = 1;
}
  if (fiducial_id == 31)
{
  pushMatrix();
  translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
  //rotate(tobj.getAngle());
  fill(0);
  rect(-obj_size+50,-obj_size/2,30,750);
  popMatrix();
  rdnum[1] = 1;
}
  if (fiducial_id == 32)
{
  pushMatrix();
  translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
  //rotate(tobj.getAngle());
  fill(0);
  rect(-obj_size+50,-obj_size/2,30,750);
  popMatrix();
  rdnum[2] = 1;
}
  if (fiducial_id == 33)
{
  pushMatrix();
  translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
  //rotate(tobj.getAngle());
```

```
fill(0);
rect(-obj_size+50,-obj_size/2,30,750);
popMatrix();
rdnum[3] = 1;
}
// ROAD Vertical
if (fiducial_id ==24)
{
pushMatrix();
translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
//rotate(tobj.getAngle());
fill(0);
rect(-obj_size+50,-obj_size/2,750,30);
popMatrix();
rdnum[4] = 1;
}
if (fiducial_id ==25)
{
pushMatrix();
translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
//rotate(tobj.getAngle());
fill(0);
rect(-obj_size+50,-obj_size/2,750,30);
popMatrix();
rdnum[5] = 1;
}
if (fiducial_id ==26)
{
pushMatrix();
translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
//rotate(tobj.getAngle());
fill(0);
rect(-obj_size+50,-obj_size/2,750,30);
popMatrix();
rdnum[6] = 1;
}
if (fiducial_id ==27)
{
```

```

pushMatrix();
translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
//rotate(tobj.getAngle());
fill(0);
rect(-obj_size+50,-obj_size/2,750,30);
popMatrix();
rdnum[7] = 1;
}
rdsum = rdnum[0] + rdnum[1] + rdnum[2] + rdnum[3] + rdnum[4]+ rdnum[5]+ rdnum[6]+ rdnum[7] ;
text(rdsum,90,250);
// house
if (fiducial_id == 3)
{
stroke(0);
fill(0);
pushMatrix();
translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
rotate(tobj.getAngle());
//rect(-obj_size/2,-obj_size/2,obj_size,obj_size);
//line(-100,0,100,0);
//line(0,-100,0,100);
image (a, a.width/2-121,a.height/2-110);
popMatrix();
fill(0);
hnum[0] = 1;
}
if (fiducial_id == 5)
{
stroke(0);
fill(0);
pushMatrix();
translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
rotate(tobj.getAngle());
//rect(-obj_size/2,-obj_size/2,obj_size,obj_size);
//line(-100,0,100,0);
//line(0,-100,0,100);
image (a, a.width/2-121,a.height/2-110);
popMatrix();
fill(0);

```

```

    hnum[1] = 1;
}
    if (fiducial_id == 6)
{
stroke(0);
fill(0);
pushMatrix();
translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
rotate(tobj.getAngle());
//rect(-obj_size/2,-obj_size/2,obj_size,obj_size);
//line(-100,0,100,0);
//line(0,-100,0,100);
image (a, a.width/2-121,a.height/2-110);
popMatrix();
fill(0);
hnum[2] = 1;
}
    if (fiducial_id == 7)
{
stroke(0);
fill(0);
pushMatrix();
translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
rotate(tobj.getAngle());
//rect(-obj_size/2,-obj_size/2,obj_size,obj_size);
//line(-100,0,100,0);
//line(0,-100,0,100);
image (a, a.width/2-121,a.height/2-110);
popMatrix();
fill(0);
hnum[3] = 1;
}
    if (fiducial_id == 8)
{
stroke(0);
fill(0);
pushMatrix();
translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
rotate(tobj.getAngle());

```

```

//rect(-obj_size/2,-obj_size/2,obj_size,obj_size);
//line(-100,0,100,0);
//line(0,-100,0,100);
image (a, a.width/2-121,a.height/2-110);
popMatrix();
fill(0);
hnum[4] =1;
}
if (fiducial_id == 9)
{
stroke(0);
fill(0);
pushMatrix();
translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
rotate(tobj.getAngle());
//rect(-obj_size/2,-obj_size/2,obj_size,obj_size);
//line(-100,0,100,0);
//line(0,-100,0,100);
image (a, a.width/2-121,a.height/2-110);
popMatrix();
fill(0);
hnum[5] =1;
}
if (fiducial_id == 10)
{
stroke(0);
fill(0);
pushMatrix();
translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
rotate(tobj.getAngle());
//rect(-obj_size/2,-obj_size/2,obj_size,obj_size);
//line(-100,0,100,0);
//line(0,-100,0,100);
image (a, a.width/2-121,a.height/2-110);
popMatrix();
fill(0);
hnum[6] =1;
}

```

```

    if (fiducial_id == 11)
{
stroke(0);
fill(0);
pushMatrix();
translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
rotate(tobj.getAngle());
//rect(-obj_size/2,-obj_size/2,obj_size,obj_size);
//line(-100,0,100,0);
//line(0,-100,0,100);
image (a, a.width/2-121,a.height/2-110);
popMatrix();
fill(0);
hnum[7] =1;
}
hsum = hnum[0] + hnum[1] + hnum[2] + hnum[3] + hnum[4]+ hnum[5]+ hnum[6]+ hnum[7] ;
text(hsum,90,230);
// garden
if (fiducial_id == 14)
{
stroke(0);
fill(0);
pushMatrix();
translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
rotate(tobj.getAngle());
//rect(-obj_size/2,-obj_size/2,obj_size,obj_size);
//line(-100,0,100,0);
//line(0,-100,0,100);
image (b, a.width/2-150,a.height/2-130);
popMatrix();
fill(0);
gnum = 1;
}
gsum = gnum;
text(gsum,90,270);
// club
if (fiducial_id == 12)
{
stroke(0);

```

```

fill(0);
pushMatrix();
translate(tobj.getScreenX(width)-20,tobj.getScreenY(height)+20);
rotate(tobj.getAngle());
//rect(-obj_size/2,-obj_size/2,obj_size,obj_size);
//line(-100,0,100,0);
//line(0,-100,0,100);
image(c, a.width/2-150,a.height/2-100);
popMatrix();
fill(0);
cnum = 1;
}
csum = cnum;
text(csum,90,290);
//cost value
text("Project Cost",0,540);
costsum = cost[0]*rdsum + cost[1]*hsum + cost[2]*gsum + cost[3]*csum ;
fill(255,0,0);
text(costsum,20,560);
fill(0);
text("Project Area",0,480);
text("sqm.",60,500);
areasum = area[0]*rdsum + area[1]*hsum + area[2]*gsum + area[3]*csum ;
fill(255,0,0);
text(areasum,20,500);
waste = allarea-areasum;
percent = waste/40;
fill(0);
text("Waste Space",0,420);
fill(255,0,0);
text(percent,20,440);
text("%",40,440);
fill(0);
gren = area[2]*gsum;
pergren = gren/40;
text("Green Area",0,360);
fill(0,82,9);
text(pergren,20,380);
fill(0);

```

```

    text("%",40,380);
// called when an object is moved
}
if(keyPressed) {
    if(key == 's') {
        saveFrame("schem-####.jpg");
    }
}
}
// these callback methods are called whenever a TUIO event occurs
// called when an object is added to the scene
void addTuioObject(TuioObject tobj) {
    println("add object "+tobj.getFiducialID()+" ("+tobj.getSessionID()+") "+tobj.getX()+" "+tobj.getY()+"
"+tobj.getAngle());
}
// called when an object is removed from the scene
void removeTuioObject(TuioObject tobj) {
    println("remove object "+tobj.getFiducialID()+" ("+tobj.getSessionID()+)");
    int fiducial_id= +tobj.getFiducialID();
    if (fiducial_id == 30) {
        rdnum[0] = 0;
    }
    if (fiducial_id == 31) {
        rdnum[1] = 0;
    }
    if (fiducial_id == 32) {
        rdnum[2] = 0;
    }
    if (fiducial_id == 33) {
        rdnum[3] = 0;
    }
    if (fiducial_id == 24) {
        rdnum[4] = 0;
    }
    if (fiducial_id == 25) {
        rdnum[5] = 0;
    }
    if (fiducial_id == 26) {
        rdnum[6] = 0;
    }
}

```

```
}  
if (fiducial_id == 27) {  
    rdnum[7] = 0;  
}  
if (fiducial_id == 3) {  
    hnum[0] = 0;  
}  
if (fiducial_id == 5) {  
    hnum[1] = 0;  
}  
if (fiducial_id == 6) {  
    hnum[2] = 0;  
}  
if (fiducial_id == 7) {  
    hnum[3] = 0;  
}  
if (fiducial_id == 8) {  
    hnum[4] = 0;  
}  
if (fiducial_id == 9) {  
    hnum[5] = 0;  
}  
if (fiducial_id == 10) {  
    hnum[6] = 0;  
}  
if (fiducial_id == 11) {  
    hnum[7] = 0;  
}  
if (fiducial_id == 12) {  
    cnum = 0;  
}  
if (fiducial_id == 14) {  
    gnum = 0;  
}  
}  
void updateTuioObject (TuioObject tobj) {  
    println("update object "+toObj.getFiducialID()+" ("+toObj.getSessionID()+") "+toObj.getX()+" "+toObj.getY()+"  
"+toObj.getAngle()  
    +" "+toObj.getMotionSpeed()+" "+toObj.getRotationSpeed()+" "+toObj.getMotionAccel()+" "+toObj.getRotationAccel());
```

```
}  
  
// called when a cursor is added to the scene  
void addTuioCursor(TuioCursor tcur) {  
    println("add cursor "+tcur.getFingerID()+" (" +tcur.getSessionID()+ ") " +tcur.getX()+" "+tcur.getY());  
}  
  
// called when a cursor is moved  
void updateTuioCursor (TuioCursor tcur) {  
    println("update cursor "+tcur.getFingerID()+" (" +tcur.getSessionID()+ ") " +tcur.getX()+" "+tcur.getY()  
        +" "+tcur.getMotionSpeed()+" "+tcur.getMotionAccel());  
}  
  
// called when a cursor is removed from the scene  
void removeTuioCursor(TuioCursor tcur) {  
    println("remove cursor "+tcur.getFingerID()+" (" +tcur.getSessionID()+")");  
}  
  
// called after each message bundle  
// representing the end of an image frame  
void refresh(long timestamp) {  
    //redraw();  
}
```