```
/////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////        CAMPO PISANI        ////////////////////////////////
/////////////////////////////////////////////////////////////////////////////////////////////

import ddf.minim.*;
CYAClient cya;

PImage Background;

int  tol;// blobs movement allowance (tolleranza)
ArrayList user;
float atickTime;/// main timer
float atickPeriod;\

float  GtickTime;///timer della grafica
float GtickPeriod;


///-----------Audio code
int value = 1;

int numChords;
int numRhythms;
int numSolist;
int numTone;

Minim minim;

AudioSnippet[][] Chords;

AudioSnippet[][][][] Rhythm;
///-----------


void setup() {
   user  =new ArrayList();  // Create an empty ArrayList
  //user.add(new Persona(this,));  // Start by adding one element
  tol=10;
   cya  =new CYAClient(this,12000); // connect to the CYA PeopleVision application using port 12000

    ////--------------------

   Background  =loadImage ("background1.jpg");
  size(1400,1050);
  smooth();

  atickPeriod  =  2000;/// period that mast be counted by the main timer
  GtickPeriod  =  4000;/// period that mast be counted by the graphic timer

  ///-----------Audio code
   minim  =new Minim(this);

   numChords=7;/// Chords number
     numTone=3;/// Chords tone: minor, major, seventh
   numSolist=2;/// number of number of kind of rhythms variations that we have studied
   numRhythms=5;/// number of rhythms variations for each chords end tone that we have studied

  ///that is an array that load the soloist tracks
   Chords  =new AudioSnippet[numChords][numTone];

  ///that is an array that load the tracks of the rhythms variations
   Rhythm  =new AudioSnippet[numChords][numSolist][numRhythms][numTone];

 ///in this cicle the software load all the tracks of the base
  for (int i=0; i < numChords; i++) {
    Chords[i][0] = minim.loadSnippet((i+1)+".mp3");
     Chords[i][1]  = minim.loadSnippet((i+1) +"min.mp3");
     Chords[i][2]= minim.loadSnippet((i+1) +"set.mp3");

 ///in this cicle the software load all the tracks of the soloists
    for(int s=0; s< numSolist; s++){
      for (int r=0; r < numRhythms; r++) {
        Rhythm[i][s][r][0] = minim.loadSnippet((i+1)+"s"+(s+1)+"r"+(r+1)+".mp3");
        Rhythm[i][s][r][1] = minim.loadSnippet((i+1)+"s"+(s+1)+"r"+(r+1)+"min.mp3");
        Rhythm[i][s][r][2] = minim.loadSnippet((i+1)+"s"+(s+1)+"r"+(r+1)+"set.mp3");

      }
    }
  }
}

void draw() {
/// refresh the background with a campo image
  image (Background, 0, 0);

/// main timer count 2 seconds periods
  if( millis() > atickTime){
     atickTime =millis() + atickPeriod;
    //// every two seconds  start the most important cycle: TheTimer
    TheTimer();
  }

///grafic timer count 4 seconds periods
  if( millis() > GtickTime){
     GtickTime =millis() + GtickPeriod;
  }
  float Gtime= GtickTime -millis(); /// Gtime count milliseconds from 0 to 4000

//// this is some variable thet
  float Gfactor=sin(Gtime/(GtickPeriod)*PI);
  float ritmo1=sin(8*(Gtime/GtickPeriod)*PI);
  float ritmo2=sin(2*(Gtime/GtickPeriod)*PI);
  float ritmo3=sin((Gtime/GtickPeriod)*PI);

  for(int i =0; i<user.size();i++){
    Persona aman=(Persona)user.get(i);
    aman.drawPersona(Gfactor,ritmo1,ritmo2,ritmo3);
  }

}

void TheTimer() {
  // get the array of people
  int[][] cameraPeople = cya.updatePeople();//getMouse();

  // check if there are people in the active zone
  if (user.size() != 0){
    for(int i = user.size(); i>0;i--){
      Persona aman = (Persona) user.get(i-1);
      boolean removeMe=true;
      for(int j=0; j<cameraPeople.length;j++){
        if (cameraPeople[j][0] == (aman.getId())){
          removeMe=false;
        }
      }
      // remove man
      if(removeMe==true){
        aman.stop();
        user.remove(i-1);
      }
    }
  }
  // are ther new people?
  for(int j=0; j<cameraPeople.length;j++){

    boolean addMe=true;
    for(int i =0; i<user.size();i++){
      Persona aman=(Persona)user.get(i);
      if (cameraPeople[j][0] == (aman.getId())){
        addMe=false;
        updateInfo(aman, cameraPeople[j][3], cameraPeople[j][4]);//update each person x & Y
      }

    }

    if(addMe==true){// we are going to make a new person!!!!!
      // should wait until they are still and have a previous direction
      int id = cameraPeople[j][0];
       Persona aman =new Persona(this,id);
      user.add(aman);// add to list

      aman.setOrigin(cameraPeople[j][3], cameraPeople[j][4]);//set the new mans position
    }
  }

  int tempBaseTone=0;
  int tempBaseChord=0;
  for(int i =0; i< user.size();i++){// loop forward and set the base info in to each person/
    Persona aman = (Persona) user.get(i);


    if (i == 0){
      tempBaseTone = aman.baseTone;// this is 0 to 2
      tempBaseChord = aman.BaseObject.chordType;
      aman.setMusicType('B');
    }

    for(int c =1; c< user.size();c=c+2){
      if(i==c){
        aman.setBaseInfo(tempBaseChord,tempBaseTone);
        aman.setMusicType('S');
      }
    }
    for(int c =2; c< user.size();c=c+2){
      if(i==c){
        aman.setBaseInfo(tempBaseChord,tempBaseTone);
        aman.setMusicType('C');
      }
    }
  }

  //////update information and get rid of bad people
  for(int i = user.size(); i>0;i--){
    Persona aman = (Persona) user.get(i-1);
    if  (aman.checkPosition()  ==false){/// might be  a good place to se the base
      aman.stop();
      user.remove(i-1);
    }
   }
}

void updateInfo(Persona  aman,float  x,float y){
  aman.setPosition(x,y);
}
```