AR 3Dmodel　静止画プログラム例

```
import jp.nyatla.nyar4psg.*;

PShape model;
MultiMarker ar;
int id1;
PImage img;

void setup() {
  size(1024, 768, P3D);
  //img = loadImage("ar_photo0.png");
  //img = loadImage("ar_photo1.png");
  img = loadImage("ar_photo2.png");
  ar = new MultiMarker(this, width, height, "camera_para.dat",
NyAR4PsgConfig.CONFIG_PSG);
  id1 = ar.addNyIdMarker(1, 60);
  model = loadShape("hamster.obj");
  model.scale(15);
}
```

```
void draw() {
  background(img);
  lights();
  ar.drawBackground(img);
  ar.detect(img);
  if (ar.isExist(id1)) {
    ar.beginTransform(id1);
    pushMatrix();
    rotateX(PI/2);
    rotateY(0);
    shape(model);
    popMatrix();
    ar.endTransform();
  }
}
```

AR 3Dmodel　動画プログラム例

```
import jp.nyatla.nyar4psg.*;
import processing.video.*;

PShape model;
Movie movie;
MultiMarker ar;
int id1, id2, id3, id4, id5;

void setup() {
  size(640, 360, P3D);
  movie = new Movie(this, "idmovie_nosound_360p.mov");
  movie.loop();
  ar = new MultiMarker(this, width, height, "camera_para.dat",
NyAR4PsgConfig.CONFIG_PSG);
  id1 = ar.addNyIdMarker(1, 60);
  id2 = ar.addNyIdMarker(2, 60);
  id3 = ar.addNyIdMarker(3, 60);
  id4 = ar.addNyIdMarker(4, 60);
  id5 = ar.addNyIdMarker(5, 60);
  model = loadShape("hamster.obj");
  model.scale(10);
}
```

```
void draw() {
  if (movie.available() == false) return;
  movie.read();
  background(movie);
  ar.drawBackground(movie);
  ar.detect(movie);

  if (ar.isExist(id1)) {
    ar.beginTransform(id1);
    pushMatrix();
    rotateX(PI/2);
    rotateY(0);
    shape(model);
    popMatrix();
    ar.endTransform();
  }
```

AR 3Dmodel　動画プログラム例　続き

```
if (ar.isExist(id2)) {
  ar.beginTransform(id2);
  fill(240, 160, 120, 200);
  translate(0, 0, 30);
  box(60);
  fill(0, 0, 0, 255);
  translate(0, 0, 30+15);
  ellipse(-10, 10, 10, 10);
  ellipse( 10, 10, 10, 10);
  fill(255, 0, 0, 255);
  arc(00, 00, 20, 20, -PI, 0);
  ar.endTransform();
}

if (ar.isExist(id3)) {
  ar.beginTransform(id3);
  fill(120, 160, 240, 200);
  translate(0, 0, 21);
  box(42);
  fill(0, 0, 0, 255);
  translate(0, 0, 21+20);
  pushMatrix();
  translate(-10, 10, 0);
  sphere(5);
  popMatrix();

  pushMatrix();
  translate(10, 10, 0);
  sphere(5);
  popMatrix();

  fill(255, 50, 0, 255);
  translate(0, 0, -10);
  arc(00, 00, 20, 20, 0, PI);
  ar.endTransform();
}

if (ar.isExist(id4)) {
  ar.beginTransform(id4);
  fill(160, 240, 240, 200);
  translate(0, 0, 30);
  box(60);
  fill(0, 0, 0, 255);
  translate(0, 0, 30+15);
  ellipse(-10, 10, 10, 10);
  ellipse( 10, 10, 10, 10);
  fill(255, 0, 0, 255);
  arc(00, 00, 20, 20, -PI, 0);
  ar.endTransform();
}

if (ar.isExist(id5)) {
  ar.beginTransform(id5);
  fill(120, 160, 240, 200);
  translate(0, 0, 40);
  box(80);
  fill(0, 0, 0, 255);
  translate(0, 0, 40+20);
  ellipse(-5, 10, 10, 10);
  ellipse( 5, 10, 10, 10);
  fill(255, 0, 0, 255);
  arc(00, 00, 20, 20, -PI, 0);
  ar.endTransform();
}
}
```

AR 3Dmodel　カメラ映像プログラム例

```
import jp.nyatla.nyar4psg.*;
import processing.video.*;

PShape model;
Capture cam;
MultiMarker ar;
int id1;

void setup() {
  size(640, 480, P3D);
  String[] cameras = Capture.list();
  println("Available cameras:");
  for (int i = 0; i < cameras.length; i++) {
    println(i, cameras[i]);
  }
  cam = new Capture(this, width, height, cameras[0]);
  //cam = new Capture(this, width, height, cameras[3]);  // for Surface PC
  cam.start();
  ar = new MultiMarker(this, width, height, "camera_para.dat",
NyAR4PsgConfig.CONFIG_PSG);
  id1 = ar.addNyIdMarker(1, 60);
  model = loadShape("hamster.obj");
  model.scale(15);
}
```

```
void draw() {
  if (cam.available() == false) return;
  cam.read();
  background(cam);
  lights();
  ar.drawBackground(cam);
  ar.detect(cam);
  if (ar.isExist(id1)) {
    ar.beginTransform(id1);
    pushMatrix();
    rotateX(PI/2);
    rotateY(0);
    shape(model);
    popMatrix();
    ar.endTransform();
  }
}
```