```
enum ballPhase {
  HOME, betweenTrap, waitingLowerTrap, beforePusherPhase, readingPusherPhase,
    whiteBallFall, blackBallFall, inTheBox
}

public class Ball extends Actor implements
Runnable {
  private ballPhase phase;
  private int x0, y0;
  protected Terminal<Boolean> upperTrap=new Terminal<Boolean>(false);
  protected Terminal<Boolean> lowerTrap=new Terminal<Boolean>(false);
  protected Terminal<Boolean> pusher=new Terminal<Boolean>(false);
  protected Terminal<Integer> ballColor=new Terminal<Integer>(0);
  protected Terminal<Boolean> whiteBall=new Terminal<Boolean>(false);
  protected Terminal<Boolean> blackBall=new Terminal<Boolean>(false);
  private  int powerOnDelay=200;

   public Ball(JSONObject src) {
     super(src);
     x0=x;
     y0=y;
     home();
   }


```

```
void home() {
   x=x0;
   y=y0;
   index=(int)random(100)%2;
   phase=ballPhase.HOME;
   ballColor.write(index);
   blackBall.write((index==0));
   whiteBall.write(!(index==0));
 }

void run() {
   for (;; ) {
     if (powerOnDelay>0) {
       powerOnDelay--;
       visible.write(powerOnDelay==0);
     } else
       switch(phase) {
       case HOME:
         if (upperTrap.read())
           phase=ballPhase.betweenTrap;
         break;
       case betweenTrap:
         if (y<y0+64)
           y++;
```

```
          else
            phase=ballPhase.waitingLowerTrap;
          break;
        case waitingLowerTrap:
          if (lowerTrap.read())
            phase=ballPhase.beforePusherPhase;
          break;
        case beforePusherPhase:
          if (y<y0+176)
            y++;
          else
            phase=ballPhase.readingPusherPhase;
          break;
        case readingPusherPhase:
          if (pusher.read())
            phase=ballPhase.whiteBallFall;
          else {
            y++;
            if (y>325)
              phase=ballPhase.blackBallFall;
          }
          break;
        case whiteBallFall:
          x--;
```

```java
        y++;
        if (y>520)
          phase=ballPhase.inTheBox;
        break;
      case inTheBox:
        home();
        break;
      case blackBallFall:
        x++;
        y++;
        if (y>520)
          phase=ballPhase.inTheBox;
        break;
      }
    try {
      java.lang.Thread.sleep(samplingTime);
    }
    catch (Exception e) {
      println(e);
    }
  }
}


```

```
 public Terminal getTerminal(String terminalName) {
  if (terminalName.equals("upperTrap"))
   return upperTrap;
  else
  if (terminalName.equals("lowerTrap"))
   return lowerTrap;
  else
  if (terminalName.equals("pusher"))
   return pusher;
  else
  if (terminalName.equals("ballColor"))
   return ballColor;
  else
  if (terminalName.equals("whiteBall"))
   return whiteBall;
  else
  if (terminalName.equals("blackBall"))
   return blackBall;
  return super.getTerminal(terminalName);
 }
}
```