ServoMaster: Transition Controllers, Coordinate Transformers & Their Combinations

1. Coordinate Transofmers

The coordinate transformers (see the API documentation) are stacked on top of the servo abstraction (actually, the coordinate transformer *is a* servo, in an object oriented sense).

2. Transition Controllers

Since the controlled transition may be supported by the controller itself, it makes sense to make the transition controller abstraction a part of servo abstraction, unlike coordinate transformer.

3. Working Together

There are interesting side effects. Consider this example:

```
ServoController controller = new FooController();
Servo servo = controller.getServo("s");
Servo linearServo = new LinearTransformer(s);

linearServo.setPosition(0);
servo.attach(new CrawlTransitionController());
linearServo.setPosition(1);
```

Pay attention to the fact that the transition controller is attached to the servo itself.

This will require the servo to move as fast as it can to the linear 0 position, then crawl to the linear 1.0. However, in this example the angular movement of the servo will be smooth, not linear.

The more is a difference between the angular position required from the servo and the position requested from the topmost transformer, the more noticeable this discrepancy will become.

However, there are cases when this is perfectly acceptable - for example, for the high latency systems where the servo controls the valves, and the transition time is negligible, but the linearity of the control is important.

Compare:

```
    ServoController controller = new FooController();
    Servo servo = controller.getServo("s");
    Servo linearServo = new LinearTransformer(s);

    linearServo.setPosition(0);
    linearServo.attach(new CrawlTransitionController());
    linearServo.setPosition(1);
```

In this example, the transition controller is attached to the linear coordinate transformer, and the *linear* movement of the servo controlled object will be smooth, but not the *angular* movement of the servo.