

## Day 4 Outline

### Objectives:

- Demonstrate the use of axes to determine the difference between *turn* and *roll*.
- Students will create their own storylines through the use of storyboards.

### Talking points:

1. Talk a bit about how some new programmers fall into the bad habit of coding by “trial and error.” If something isn’t working like they think it should, they try something else, and then something else, and so on, in the hopes of coming across the sequence of instructions that will do what they want. It is better to understand how things work, and take out the guess work out of programming.
2. Demonstrate how to use the *axes* to determine which to use of *turn* and *roll*.
  - a. Create a world with a *Rockette*. Add a set of *axes* from the Shapes Gallery.
  - b. Standing in front of the class, (I had them move their chairs away from the computers in order to get their undivided attention.) imitate the *axes* in the Shapes Gallery, hold your right arm out as the right arrow, hold your left arm in front of you as your forward arrow, and your head becomes the up arrow (raise up and down on your toes). Demonstrate each of the method calls in the grid below (which is written on the board), asking the students which arrow doesn’t change direction (answers in red – fill in on the board as you go)

method call	turn left or right	turn forward or backward	roll left or right
axis that does NOT move	up	right	forward

- c. Demonstrate the various method calls individually on an *axes* object in an Alice world, confirming what we have in the grid.
3. Create a *kick* method for the *Rockette*, demonstrating how the axes can be used to determine which method to call and which direction to use. (For our complete kick motion, we had her bend her right knee back, then lift her right thigh, then straighten out her right leg, then lower her right leg. For the first two, we did the following to determine which method to call (done by clicking and dragging the axes object from the Object Tree over the image and then selecting the methods)
    - a. First call `axes.moveTo` to the *Rockette*’s right calf (right thigh).
    - b. Next call `axes.orientTo` to the *Rockette*’s right calf (right thigh).
    - c. Finally call `axes.setVehicleTo` to the *Rockette*’s right calf (right thigh).

We did this to determine the first two moves of the kick action. When we had completed the kick, we put the first two moves in a *doTogether* to give it a more realistic look. I mentioned that I could also make her point her toe if I wanted to.)

4. **After creating the kick method**, make copies of the *Rockette*.

## Day 4 Outline

- a. Demonstrate that more *Rockettes* added from the gallery will not have the *kick* method that we wrote. Only copying our original *Rockette* will give us objects that have the new methods we created. (We made copies of the *Rockette* and had them kick together. We then put that in a loop and had them do several kicks.)
5. Reinforce that it is better to understand how things are working, rather than coding by “trial and error.”
6. Another bad habit that new programmers can fall into is to begin to code without thinking first about what they want to do. “Think before you code.” Talk about the “Making of Shrek” type special features on DVDs where they show how animators sketch out their stories before they begin to animate.
7. Show students my original storyboard, the animation from the storyboard, and then my final animation (where I added other things that I had time to add). Emphasize that the quality of the artwork is unimportant.
8. Give students storyboard sheets for them to sketch out their stories.
9. Once they complete their storyboards, students will spend the rest of the time creating their own stories.
10. At the end of day 4, the parents were invited to visit the lab and students demonstrated to their parents what they had created throughout the week.