Module 5: CS Unplugged Simulation

Version of 14 June 2015

In this exercise we will simulate the kodu in the Apple3 world and watch as it pursues and consumes apples and changes its state.

Materials: To simulate apples you will need two blue things and two red things. Plastic balls are a good choice. You can also use party balloons; attach a binder clip or some other type of weight to each balloon to prevent it being blown around when people walk by.

You will also need to make a large sign with writing on both sides that a person can hold up. One side of the sign should be colored blue (use a blue magic marker, or add some blue stickers or blue construction paper). The blue side should say:

> **PAGE 1:**
>   **[1] WHEN see blue apple DO move towards**
>   **[2] WHEN bump blue apple DO grab it**
>   ↳ **[3] WHEN DO switch to page 2**

The other side of the sign should be colored pink (use a pink or red magic marker or add some pink stickers or pink construction paper). The pink side should say:

> **PAGE 2:**
>   **[1] WHEN see red apple DO move towards**
>   **[2] WHEN bump red apple DO eat it**
>   ↳ **[3] WHEN DO switch to page 1**

Setup: Place the balloons on tabletops or shelves around the room. The two blue balls should be closer to each other than to either red ball. The person playing the role of the kodu starts out near a red ball, and should be holding the sign up to their chest with the blue side facing the audience.

Instructions: One person plays the kodu, while the rest of the class interprets the rules on the current page and tells the kodu what to do. The crucial thing is *how* the rules are interpreted and what the students shout out to the kodu.

- On each rule cycle, go through the rules in order.
- Call out the rule number, the WHEN part of the rule, and the word "true" or "false".
- If the WHEN part is true, call out the DO part of the rule.
- For "move toward", have the student take just one step toward the apple. (Ask students which apple to move toward. Answer: the closest one of the correct color.)
- If rule 2's condition is false, tell them to say rule 3 "can't run".
- The class gets to decide when "bump" is true.
- If rule 2's condition is true, simulate eating or grabbing the apple by dropping it on the floor.
- Simulate the "switch to page" action by having the kodu flip the sign around.
- Since each "move toward" takes just one step, there will be lots of repetition.

Model the dialog for the students. It should sound exactly like this:

One, WHEN see blue apple, TRUE, DO move toward *(takes a step)*
Two, WHEN bump blue apple, FALSE
Three, can't run.

One, WHEN see blue apple, TRUE, DO move toward *(takes a step)*
Two, WHEN bump blue apple, FALSE
Three, can't run.

One, WHEN see blue apple, TRUE, DO move toward *(takes a step)*
Two, WHEN bump blue apple, FALSE
Three, can't run.

One, WHEN see blue apple, TRUE, DO move toward *(takes a step)*
Two, WHEN bump blue apple, TRUE, DO grab it *(drops the apple on the floor)*
Three, WHEN DO switch to page 2 *(reverses the sign)*

Once students "get it" you can speed up the simulation by having the kodu take bigger steps. Continue until all the apples are gone.

Questions to throw out during the activity:

1. Which side of the sign should the kodu display initially, and why? (The blue side because the kodu always starts on page 1.)
2. Which blue apple will the kodu go to initially? (The closest one.)
3. Why can't rule 3 run initially. (Because it's indented and its parent's condition is false.)
4. What happens when all the apples are gone? (The program is still running but none of the rules has a true condition, so the kodu just sits there.)
5. What state will the kodu be in at the end? (After eating the last red apple it will switch to state 1 and stay there.)

Advanced Activity

For advanced students you can do a simulation of a faulty solution to the State1 world, where one student plays the castle (have them hold up a picture of a castle), another plays the red tree (give them a picture of a tree), and a third plays the kodu with the following rules:

[1] WHEN bump red tree DO
    ↳ [2] WHEN see castle DO move toward
[3] WHEN see red tree DO move toward

The simulation will show that when the kodu bumps the red tree (rule 1) and takes a step toward the castle (rule 2), rule 3 is blocked due to a conflicting action. But now "bump" is no longer true, so on the next cycle rule 1's condition is false, rule 2 can't run, and rule 3 applies again. The kodu turns back and forth between tree and castle. Run this on the computer and zoom in: you'll actually see the kodu wavering.