Kodu Lesson 2: Color Filters with Pursue and Consume

David S. Touretzky
Version of June 7, 2016

Learning Goals

- Review of the "Pursue and Consume" idiom.
- Using a color filter with "see" and "bumped".
- The "grab" action picks up an object.
- The "color it" action changes the color of an object.
- Use of "once" in Apple2X for button-triggered actions.

Worlds

- Apple2 – with Debug LOS turned on
- Apple2X – with Debug LOS turned on for the kodu
- Ball1 – with Debug LOS turned on
- Paint1

Handouts

- Apple2X World
- Ball1 World
- Paint1 World

Tile Manipulatives

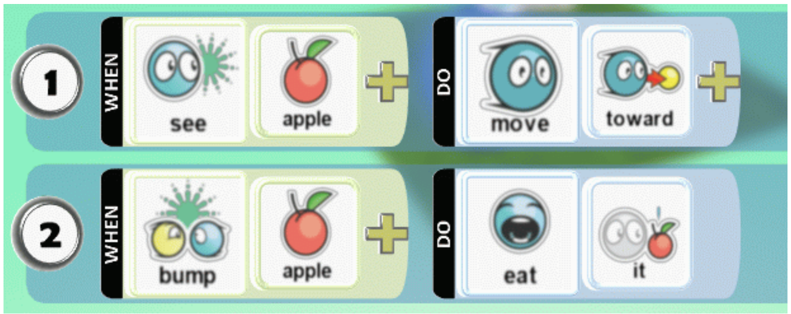
- WHEN-see, WHEN-bumped
- DO-move (2 tiles), DO-eat, DO-grab
- apple (4 tiles), ball (4 tiles)
- red (2 tiles), blue (2 tiles)
- toward (2 tiles), it (2 tiles)

Part 1: The “Pursue and Consume” Flash Card in Depth

1. Have students re-examine the “Pursue and Consume” flash card.
2. Start with the picture on the front. What is going on?
3. Explain that “pursue” means to chase, follow, or seek after. Pursue rules always involve motion.
4. Explain that “consume” means to eat or use up. Examples include a person consuming food, a car consuming gasoline, or a fire consuming wood. Consume rules cause the consumed thing to disappear or get used up.
5. Switch to the back of the flash card and review the tiles.
6. Ask the students and have them explain:
    - Which rule is the “pursue” rule, and why? (rule 1: it makes the kodu *move toward* the apple)
    - Which rule is the “consume” rule, and why? (rule 2: it makes the apple disappear)

Part 2: The Apple2 World, Where Apples Are Either Red or Blue

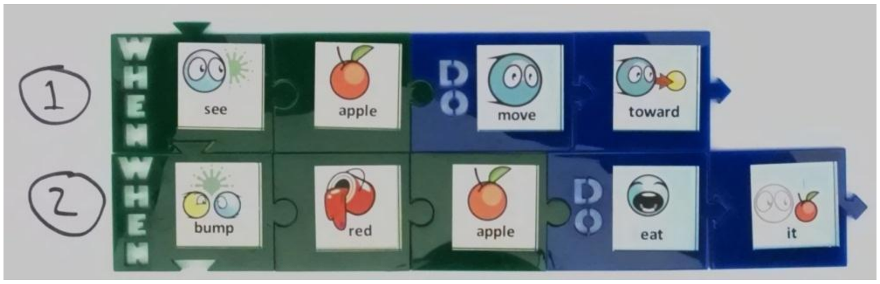
1. Load and run the Apple2 world. Let the students explore it. (Remind them of the motion controls.)
2. Ask the students what colors the apples come in. (red and blue)
3. Ask them what they think will happen if the kodu eats a blue apple.
4. Have students enter the two rules from the Pursue and Consume flash card.



5. Let students run the program and observe what happens. (The game ends when the kodu eats a blue apple and dies.)
6. Explain to the students that the blue apples are poisonous and the kodu mustn’t eat them. We want the kodu to eat only red apples. How can we tell it that?

Part 3: Adding A Color Filter

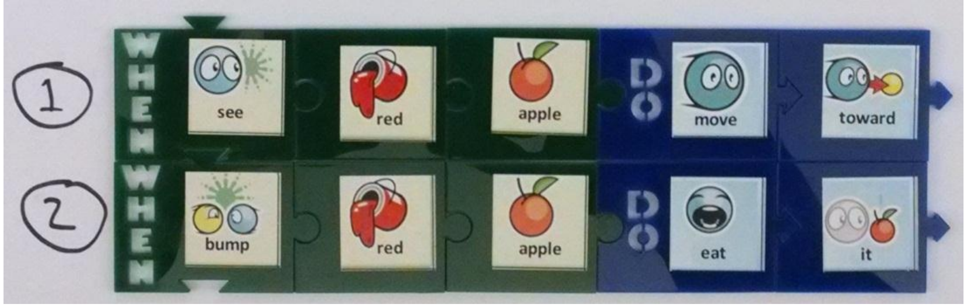
1. Have students use the tile manipulatives to make the pursue and consume rules.
2. Using the tile manipulatives, have students add a “red” color filter to rule 2.
3. Students may wish to add a color filter to rule 1 as well, as shown on the flash card. Tell them not to do this yet.



4. Have the students re-enter the rule editor and add the color filter to rule 2, but not rule 1:

   [2] WHEN bumped *(colors)* red *(objects)* apple DO eat it

5. Ask the students what they think will happen when they run the program. Write down their guesses.
6. Run the program and observe the result:
   - The Kodu goes to and eats one or two red apples.
   - Then it goes to a blue apple and just sits there. Why?
7. Ask the students:
   - Which rules apply in this situation with the stuck kodu? (only rule 1)
   - Why doesn’t rule 2 apply? (the bumped apple isn’t red)
   - How can we fix this problem? (make the Kodu only go to red apples)
8. Using the tile manipulatives, have students add the color filter to rule 1.



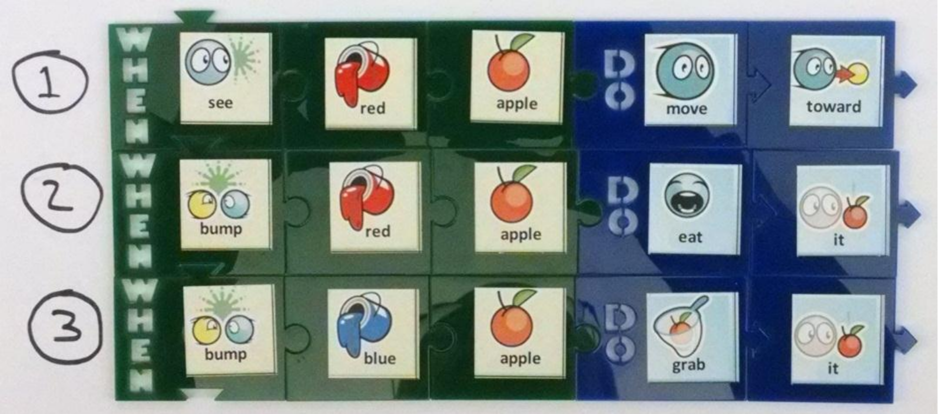
9. Have the students make this edit to the program:

   [1] WHEN see *(colors)* red *(objects)* apple DO move toward

10. Test the modified program: now the Kodu will ignore blue apples entirely.
11. Ask the students to point out where this color filter trick is shown on the flash card.

Part 4: Trying to Consume the Blue Apples

1. Show the students the “grab” tile. Tell them that if the Kodu grabs a blue apple, it will vanish.
2. Ask them: is this a “pursue” action, or a “consume” action? (consume)
3. Give students the “grab” tile and ask them to make a complete consume rule using the tile manipulatives to get rid of the blue apples. The result should be:



4. Have students add this rule in the rule editor.

   [3] WHEN bumped blue apple DO (holding) grab it

5. Ask the students: is this rule enough to get rid of the blue apples? Answer: no – we need a pursue rule as well. But they might not see that yet; it’s okay if they don’t.
6. Run the program and demonstrate that the blue apples are ignored; the new rule never has a chance to fire.

Part 5: Pursuing Blue Apples

1. Ask the students how we can make the kodu pursue the blue apples. There are two possible answers, which produce different behaviors.
2. Favored answer: remove the "red" filter from rule 1 so the kodu pursues apples of any color.

   [1] WHEN see apple DO move toward

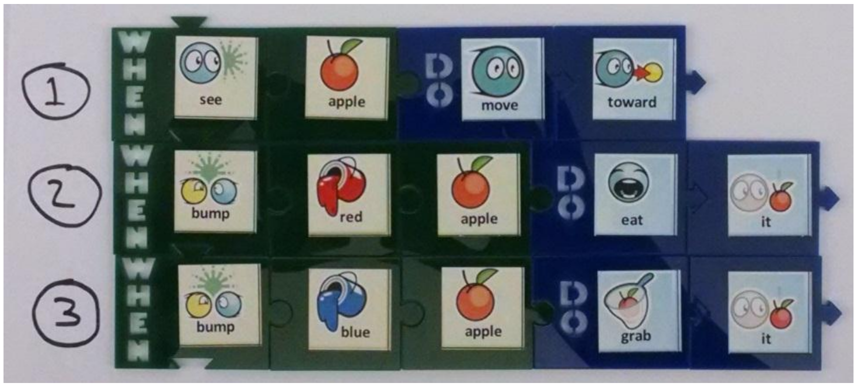
3. Alternative answer (not favored): add a second pursue rule for blue apples:

   [1] WHEN see red apple DO move toward

   ….

   [4] WHEN see blue apple DO move toward

4. If the students suggest the second approach, tell them that it's correct, but there is a simpler solution, and guide them toward the favored one. (The alternative answer is not favored because the kodu will visit all the red apples first, and only visit the blue apples when there are no red ones left. Also, it requires four rules instead of three.)
5. Have students make the change to the tile manipulatives: remove the color filter from rule 1.



6. Have students make the same change in the rule editor. Note: use the blue X button to delete a tile.
7. Test the program and verify that all apples are now consumed in the correct way (red ones are eaten, blue ones are grabbed and vanish), and the Kodu always goes to the nearest apple of any color.

Part 6: Competitive Apple Eating with Apple2X (Optional)

1. Distribute the Apple2X handout and have students follow the instructions.
2. First they will implement the "eat red apples and grab blue ones" for the kodu character.
3. Next they will program the flying fish so they can control it: Note: the "once" tiles are important.

   [1] WHEN gamepad L-stick DO move

   [2] WHEN gamepad A-button DO grab once

   [3] WHEN gamepad B-button DO eat once

4. The student-controlled flying fish can compete against the kodu to see who can score the most points.

Part 7: Chasing Soccer Balls

1. Distribute the Ball1 World handout and have students load and play the Ball1 world.
2. Explain that the cycle character shoots soccer balls that are either orange or purple. It only releases one ball at a time. The kodu needs to chase down and eat the orange balls, and grab the purple ones.
3. Ask students how they would program the kodu to do this. Have them use the tiles to model their answer. They can use a “red” tile to substitute for orange and a “blue” tile to substitute for purple. If using a standard tile set, they will have to borrow a third “ball” tile from another set, or else make one using a Post-It note.
4. Ask the students to go into the rule editor and program the Kodu with their solution. Note: to get to “ball” you must select: “(objects) (more) (more) ball”.
5. The result should look like this:

```
[1] WHEN see ball DO move toward
[2] WHEN bumped orange ball DO eat it
[3] WHEN bumped purple ball DO grab it
```

6. Have them run the program to verify that their solution works correctly.
7. **Discussion question**: How does the Second Law of Kodu apply to this program?
8. Answer: depending on what color ball the kodu bumps, either rule 2 or rule 3 will run. It doesn’t matter what the other rule is doing; if one of the rules can run, it will.

Part 8: Painting the Huts (Advanced)

1. Distribute the Paint1 World handout and have students load that world.

2. This exercise is a little different than the apple world, although we again have objects that are initially of two colors: the huts are either black or white. Let students work out the solution themselves, but here are some key insights:
   a. There are really three colors to consider, not two, because the huts are being painted blue.
   b. Although in English we would say "go to a back or white hut", we can't say that in Kodu. Instead we must write separate rules, one for black huts and one for white ones. But unlike with the red and blue apples, the action here is the same in both cases. The reason for including the color filter is that we do not want to go to blue huts.
   c. The consume rule (for painting the huts) does not need a color filter.
   d. The world requires the kodu to visit each hut before painting it. This means a "bump", so the rule that colors the hut should use "bumped" in the WHEN part:

      [1] WHEN see black hut DO move towards

      [2] WHEN see white hut DO move towards

      [3] WHEN bumped hut DO color it blue

   e. The code will *not* work without a color filter because the kodu will get stuck at the first hut instead of moving on after painting it::

      [1] WHEN see hut DO move towards

      Students will probably make this mistake; let them figure it out for themselves.

   f. It's possible to color a hut from any distance, so students might want to solve the problem with the following rules:

      [1] WHEN see white hut DO color it blue

      [2] WHEN see black hut DO color it blue

      This will in fact color all the huts blue, but the world is programmed to detect this "cheating" and complain that the hut wasn't bumped before painting it. We did this to force the exercise to fit the standard Pursue and Consume form.

   g. If a student writes "color blue" instead of "color it blue", the kodu will color itself blue.

3. Class discussion: why is the rule with the "color it" action a consume rule? Answer: a consume rule either *removes* or *uses up* the object so it is no longer pursued. In this case, where we're pursuing black or white huts, the consume rule changes the hut's color to blue, so the pursue rules no longer see it. It's as if the hut has disappeared.

Part 9: Free-Style Color Filter

1. Have students load the FreeWorld2 world.
2. Let them populate it with objects of their choosing.
3. Show them how to change the color of an object by putting the cursor over it and then moving the D-Pad left or right to cycle through the color menu at the top of the screen.
4. Ask them to implement a version of Pursue and Consume that uses a color filter to only eat objects of a particular color.