



Music Blocks Lesson Plan

Crazy Keyboard

Age:

7-12 years

Lesson duration:

60 minutes

- Introduction: Other approaches to keyboards (15m)
- Part 1: The Keyboard block (20m)
- Break (5m)
- Part 2: Enhancements (15m)
- Performance/Critique (10m)

Number of students:

Up to 10.

Rationale:

Students will learn about different approaches to designating change in pitch.

Objectives:

Students will understand how to use the Keyboard block from the Sensor palette. They explore alternative ways to design a “piano keyboard”.

LESSON

Introduction: Keyboards.

Lead the students in a discussion about the organization of the keys on a piano. The keys are arranged in a row. Each key is dedicated to a single pitch. As you move right, the pitch increases by a half-step. As you move left, the pitch decreases by a half-step.

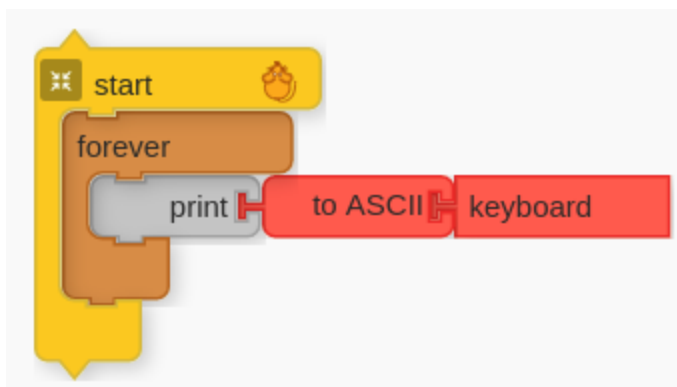
What if the keys were not dedicated to a pitch? What if they were dedicated to a “change in pitch”?

Discuss how that might work? How would you go from one pitch to another? How would you implement something like that in Music Blocks?

Introduce the Keyboard block from the Sensor palette. We can dedicate a row of keys on the computer keyboard to our “crazy piano”.

Part 1: The Keyboard block.

- A. Grab Keyboard block from the Sensor palette. Notice that it comes with a “to ASCII” block attached. This is because the Keyboard block returns a number and that number needs to be converted into a letter.¹ Using a Print block inside of a Forever block, watch what happens when you type.

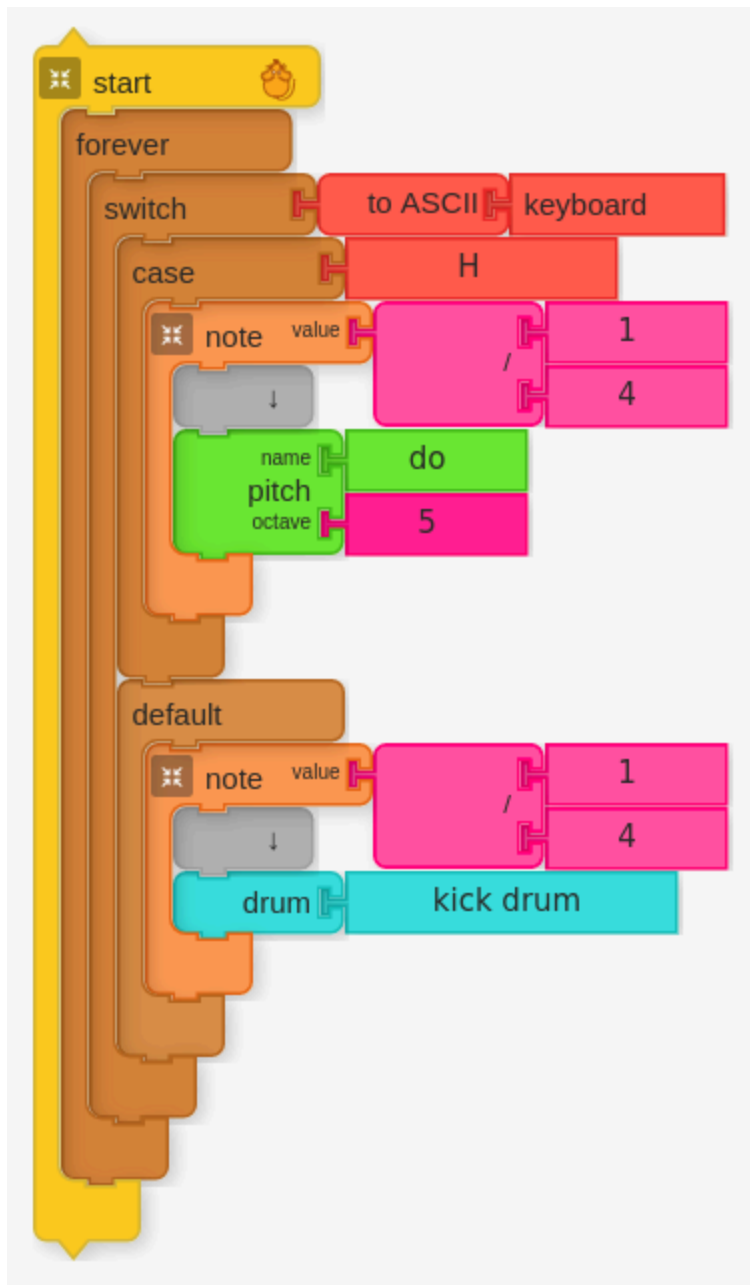


You will mostly see 0s. This is because the Keyboard block is set to 0 when there is no key pressed. Try holding down a key. You’ll see an occasional flash of the key you are pressing.²

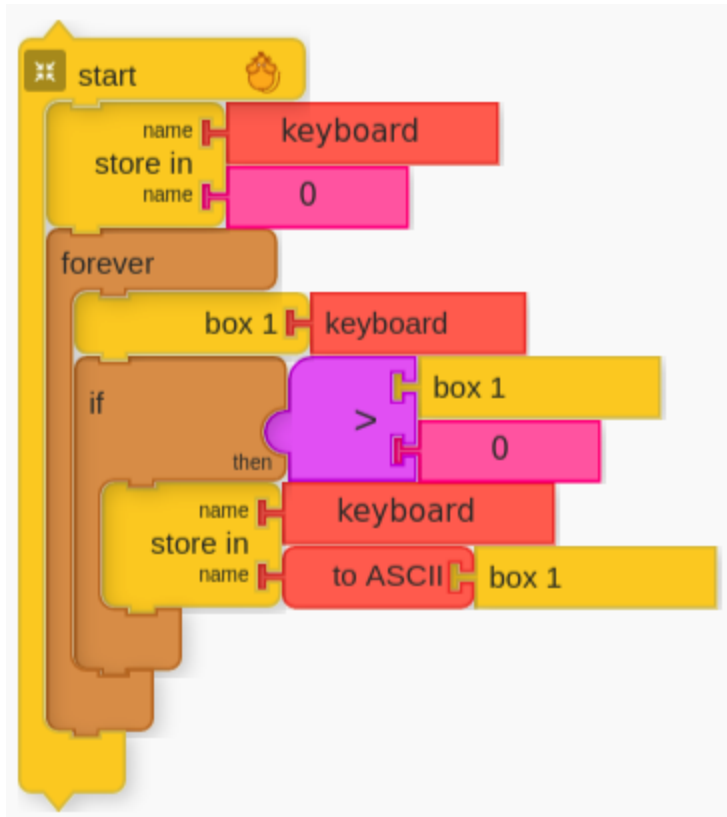
- B. Now we’ll try mapping keys to music. We can use a Switch block to determine what action to take when a key is pressed. Let’s try mapping H, which is in the middle of the computer keyboard, to Middle C, which is in the middle of the piano keyboard. (Use a Text block from the Media palette for H.) The default action, when no key is pressed is a drum beat.

¹ Each key on a computer keyboard is mapped to a number. Those numbers need to be translated into letters. A common mapping is called ASCII (American Standard Code for Information Interchange), which is the one we use in Music Blocks. In ASCII, 65 is mapped to A, 66 is mapped to B, etc.

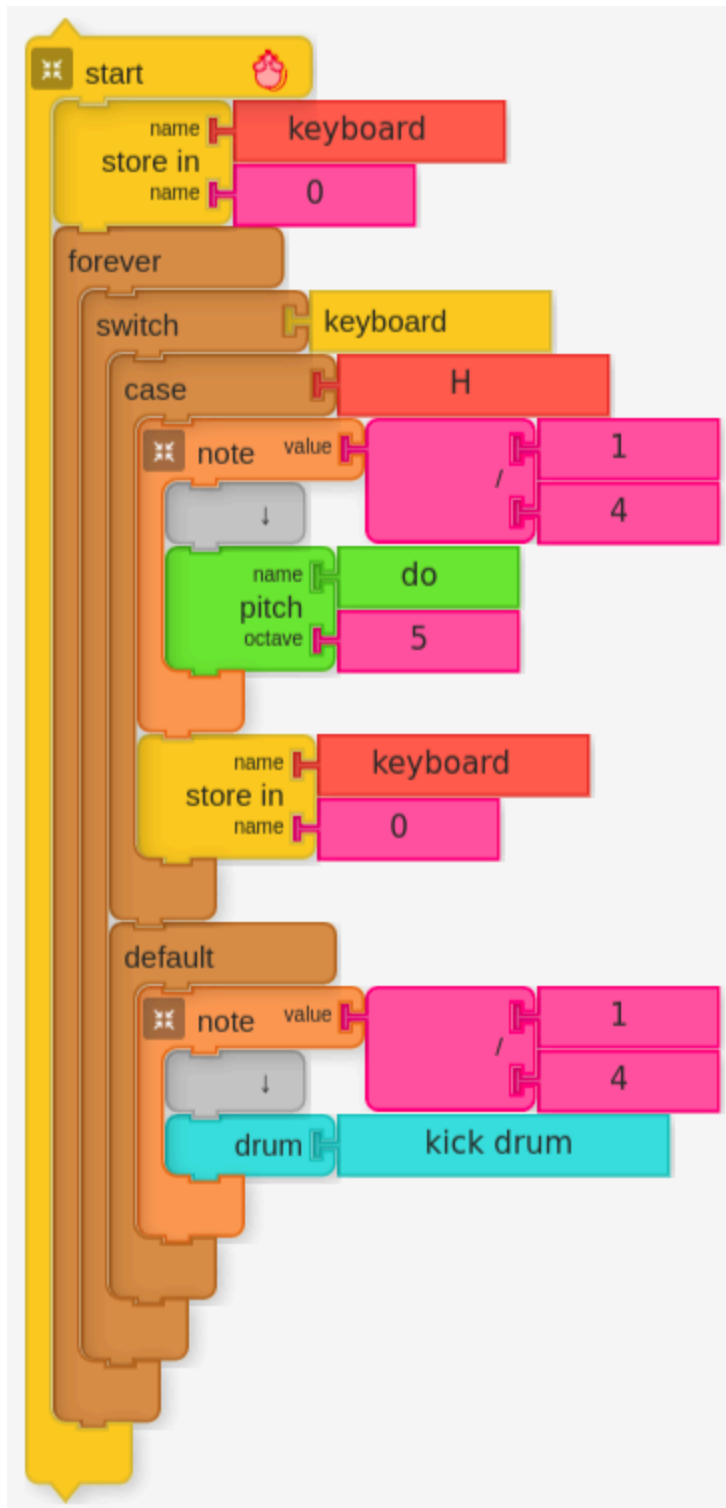
² Each time you press a key, there is a “keyboard” event. When you hold down a key, multiple keyboard events are triggered.



Even when we hold the H key down, we still occasionally miss a beat. We can just accept that imperfection or use a separate process (another mouse) to manage the keyboard input for use, as per below:

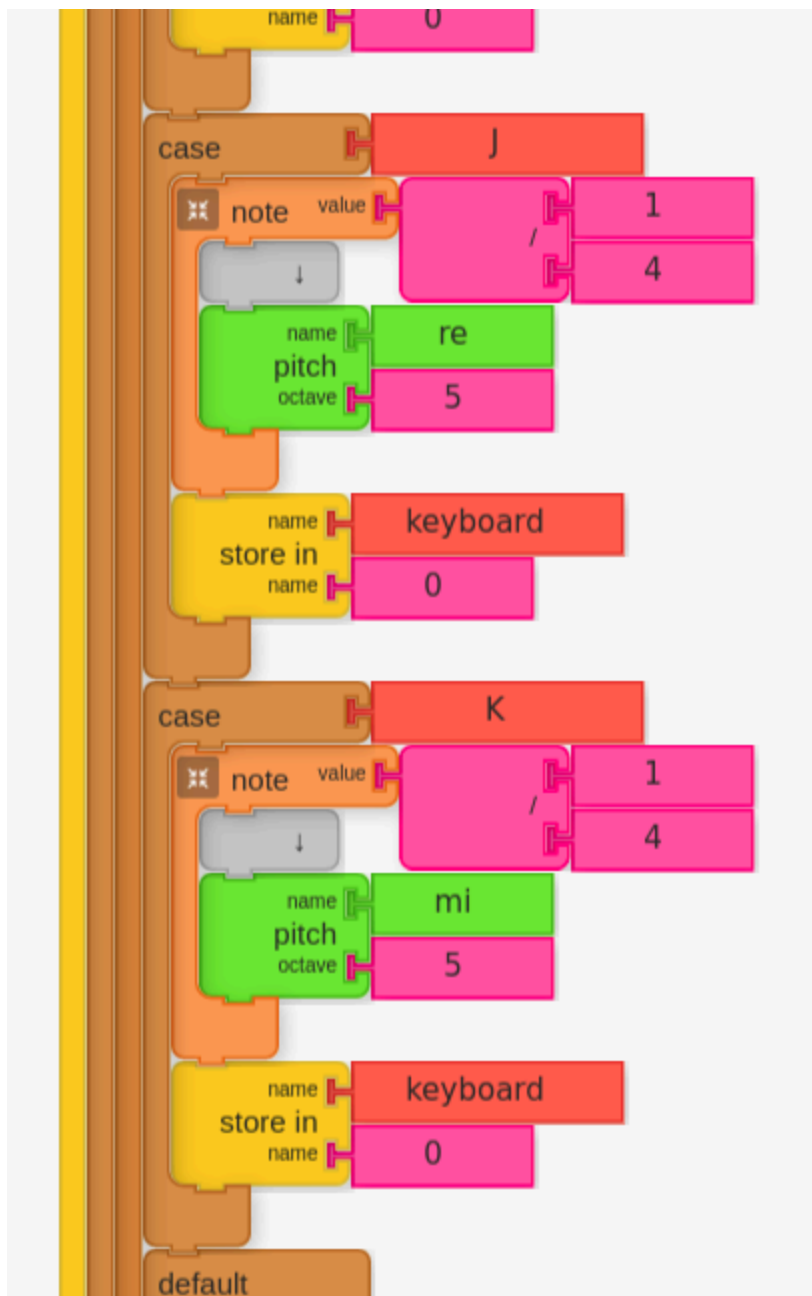


We'll use a box to store the value of the keyboard. To start, we'll put a 0 in the box, since no key has been pressed. We use another box (Box 1) to "read" from the Keyboard block. If a key is pressed, then the value in Box 1 will be > 0 , so we copy Box 1 into the keyboard box. This way, the keyboard box remembers the most recent key pressed. This process is sometimes referred to as "debouncing".



We can use the value stored in the keyboard box instead of the Keyboard block. Note that we put a 0 into the keyboard box after we “use” it so that we don’t get repeats.

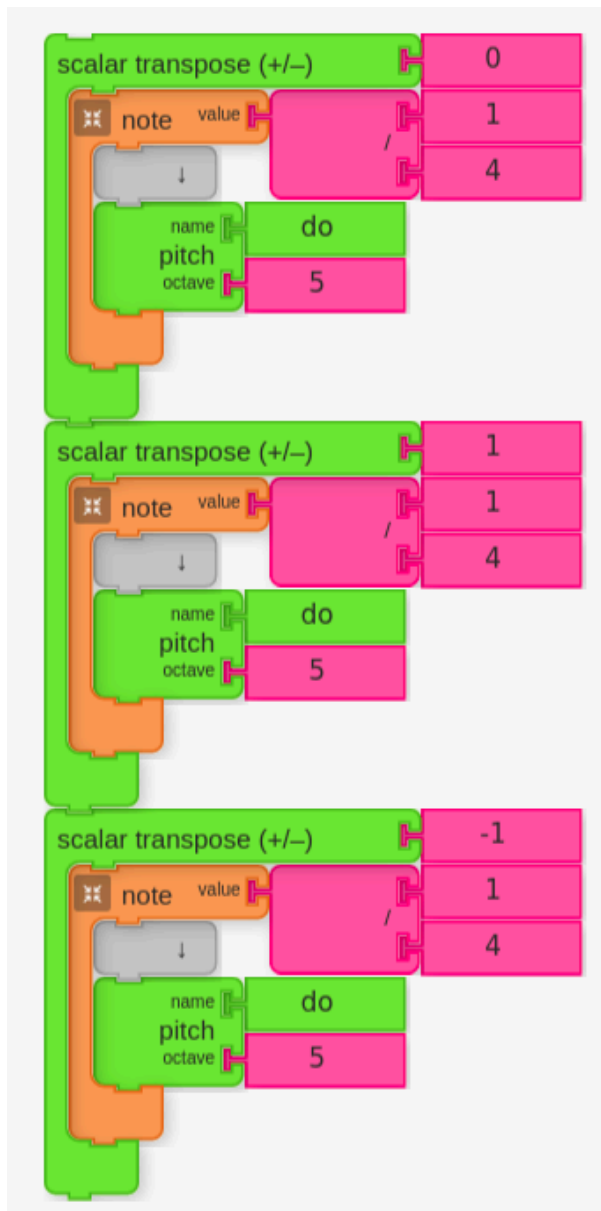
C. We can make a traditional keyboard by adding more Case blocks, one for each key.



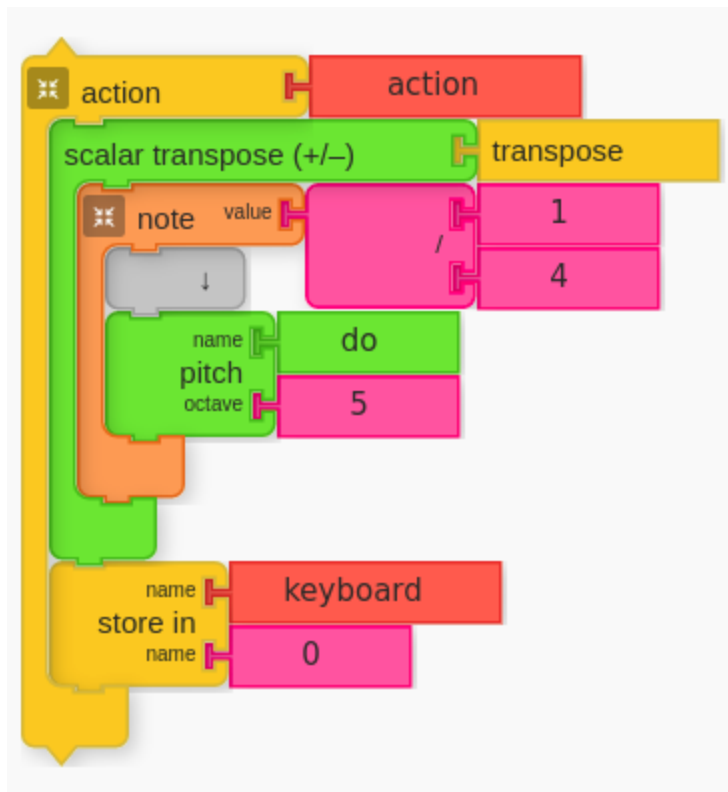
Break

Part 2: Crazy keyboards

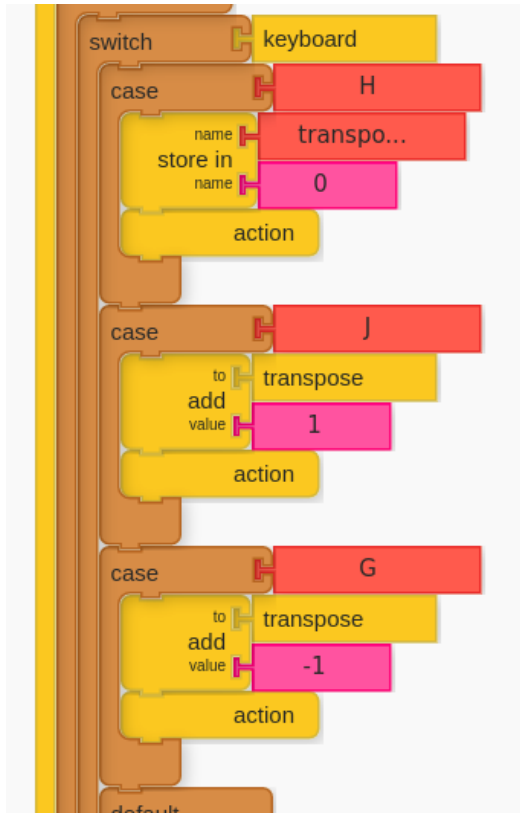
- A. We can change the behavior of our keyboard by using the Scalar Transpose block. This raises or lowers the pitch by some number of steps. In the example below, a transpose of 0 has no affect, so Do is played. A transpose of 1 raises the pitch, so Re is played. A transpose of -1 lowers the pitch, so Ti is played.



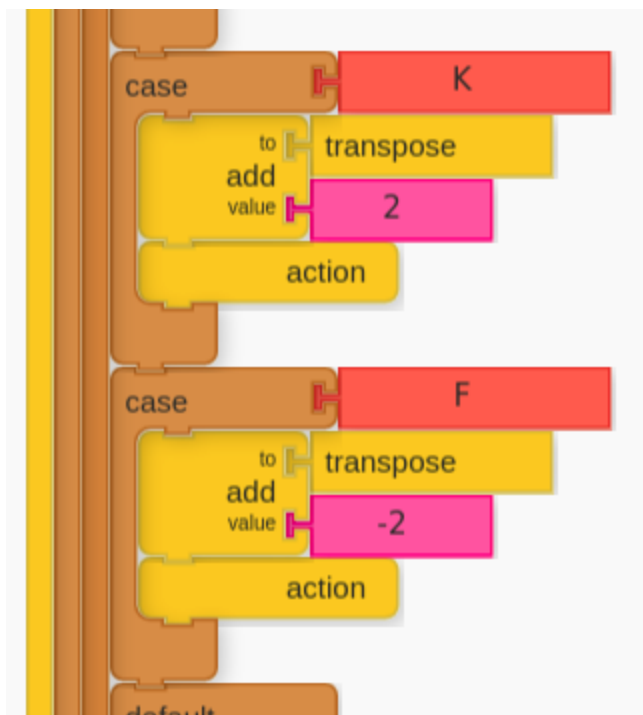
B. We can map our keyboard to transpositions instead of notes.



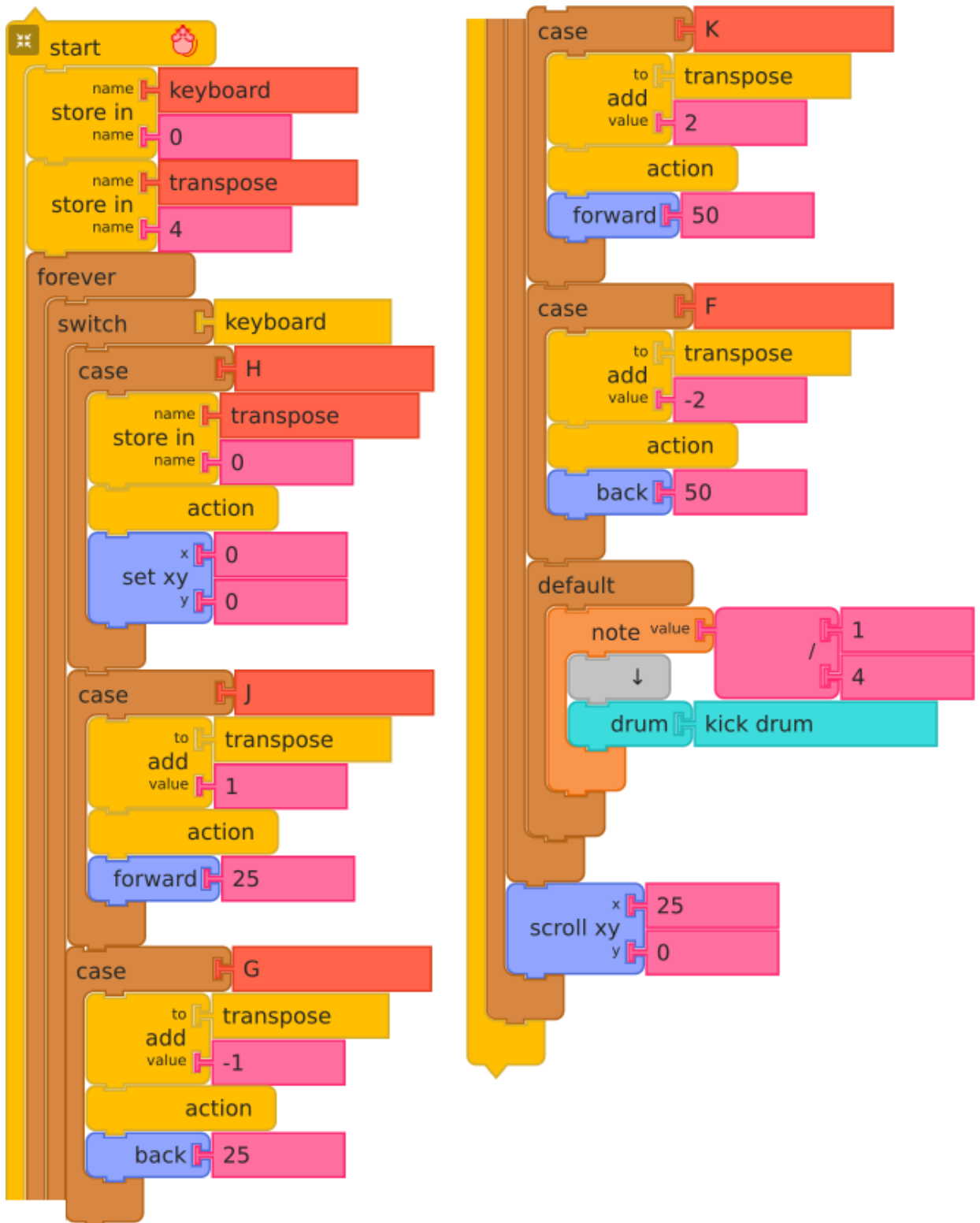
And use this action in our switch.



C. Add more keys that do different transpositions.



D. Add some graphics.



Performance/Critique:

1. Have each student talk about keyboard designs.
2. Engage in a discussion about their different approaches.

Key events:

- Introduction of key concepts: scalar transposition, keyboard input, debouncing.

Materials:

- Music Blocks software

Assessment:

- Observe participation.
- Does the program perform as expected?



© Copyright Walter Bender, 2020