Debugging in Music Blocks

Learning is hard fun.—Marvin Minsky

Make the complicated comprehensible—Arthur Miller

Debugging is the learning opportunity of the 21st Century. — Cynthia Solomon

The important message that comes from ideas about debugging is that we learn from our mistakes; that the intricate process of making things work or learning new skills has to do with hypothesizing, testing, revising, etc.—Cynthia Solomon

Sometimes bugs are serendipitously adopted as features worth perpetuating, sometimes procedures must be constructed to deal with the phenomena caused by their appearance, and sometimes the bugs and their side effects need to be removed. But in this pursuit, children become creative researchers studying behavior, making up theories, trying out ideas, etc.—Cynthia Solomon

6 Stages of Debugging—Anonymous

1. That can’t happen.
2. That doesn’t happen on my machine.
3. That shouldn’t happen.
4. Why does that happen?
5. Oh, I see.
6. How did that ever work?

Programming is hard. Composing music is also hard. Both programming and composing involve some trial and error and serendipity. Inevitably you will make mistakes along the way. Music Blocks provides a number of mechanisms, reviewed below, to help you explore ideas and find mistakes.

1. Clicking on an individual stack of blocks

The *Play* button (in top left corner) will run all of the *Start* blocks simultaneously. (Every Music Blocks project has at least one *Start* block). But you can also run an individual stack of code by clicking on a stack. This lets you test and debug small sections of code, or, as in the example below, you can play a single voice by clicking on one of the *Start* blocks or single phase by clicking on one of the *Action* blocks.

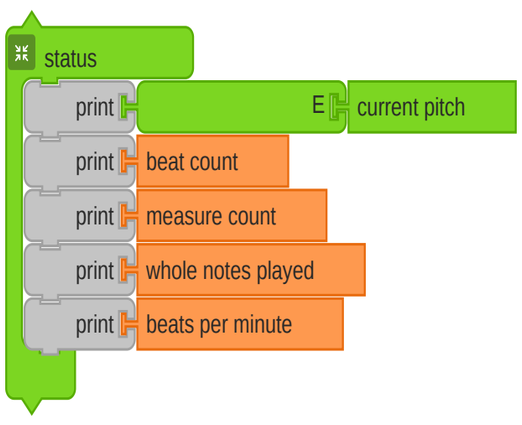
2. Print and Comment blocks

The *Print* block (found on the *Extras* palette) can be used to print a message while running a program. It is useful to determine if a section of code is being executed when expected or if a box or parameter contains an expected value.

The *Print* block is used to display the number of whole notes played, in this case, `1/4 + 1/4 + 1/2`, which adds up to `1`, which is displayed at the top of the browser window.

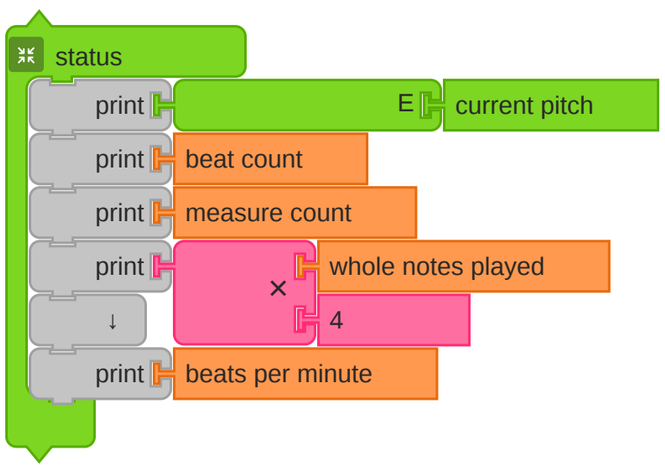
The *Comment* block (also found on the *Extras* palette) is similar to the *Print* block, except it only prints a message when the program is being run in *Playback Slow* mode (See below). Comments are also written to the browser console.

3. Status widget



The *Status widget* is a tool for inspecting the status of Music Blocks as it is running. By default, the key, BPM, and volume are displayed. Also, each note is displayed as it is played. There is one row per voice in the status table.

Additional *Print* blocks can be added to the *Status* widget to display additional music factors, e.g., duplicate, transposition, skip, staccato, slur, and graphics factors, e.g., x, y, heading, color, shade, grey, and pensize.



You can do additional programming within the status block. In the example above, `whole notes played` is multiplied by `4` (e.g. quarter notes) before being displayed.

4. Playback modes

Clicking on the Play button will play your program at full speed. (It will also hide the blocks while the program runs, which improves performance.) But there are two other playback modes.

On the Secondary Menu, there are two other Play buttons.

During Playback Slow mode the program will pause between the execution of each block and the block being executed will be highlighted. This is useful for following program flow, ensuring that the sequence of blocks being executed is what you expect. In addition, the value stored in any box or parameter is displayed on the block as the program runs, so you can “inspect” program elements as the program runs.

Run Step by Step advances one block per button press.

5. Show and Hide blocks

The *Show* and *Hide* blocks (found on the *Extras* palette) are useful for setting “breakpoints” in your program to debug a specific section of code. By putting a *Show* block at the start of a problematic section of code and a *Hide* block at the end of the section, your program can be run full speed until it gets to the *Show* block. Then the blocks are displayed and run in *Playback Slow* mode. When the *Hide* block is encountered, the blocks are hidden and the program resumes running at full speed.

6. Browser console

As Music Blocks runs, some debugging information is written to the browser console, such as the notes being played and comments (See the *Comment* block above). The console can be accessed by typing `Ctrl-Shift-J` on most web browsers.

Shown above is the console output from three notes: `sol mi sol`.
