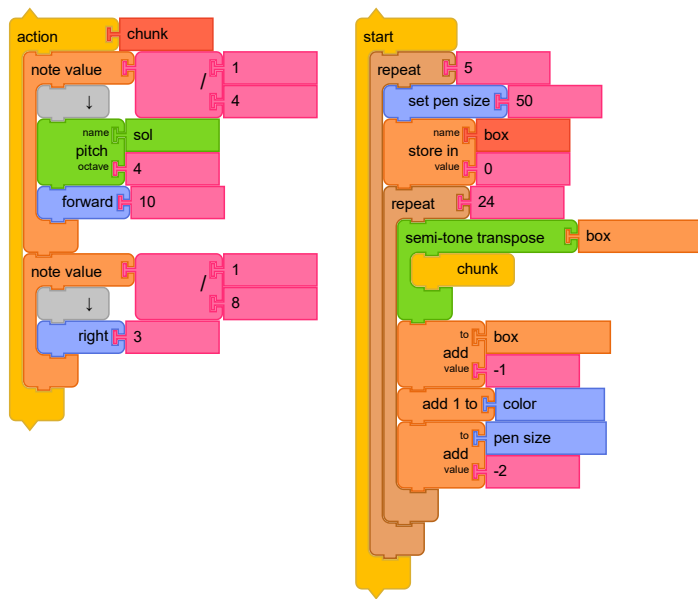
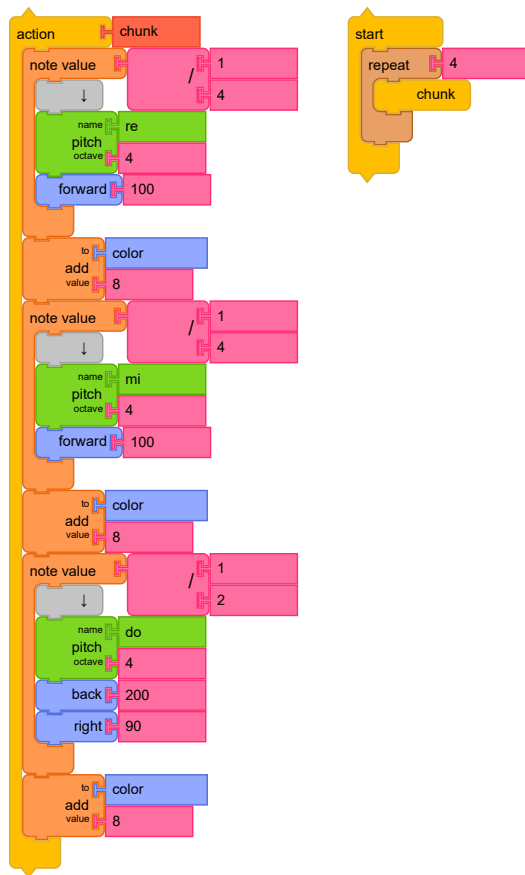


### 3.6 Adding graphics



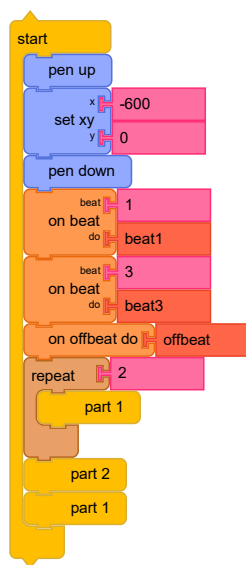
Turtle graphics can be combined with the music blocks. By placing graphics blocks, e.g., *Forward* and *Right*, inside of *Note value* blocks, the graphics stay in sync with the music. In this example, the turtle moves forward each time a quarter note is played. It turns right during the eighth note. The pitch is decreased by one half step, the pen size decreases, and the pen color increases at each step in the inner repeat loop.

[RUN LIVE](#)



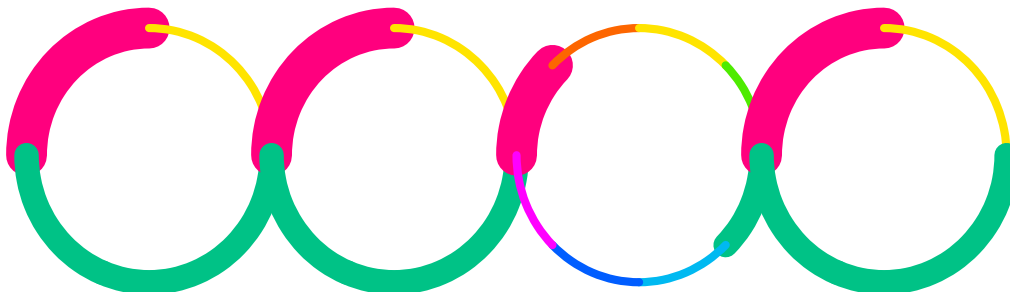
Another example of graphics synchronized to the music by placing the graphics commands inside of *Note value* blocks

[RUN LIVE](#)

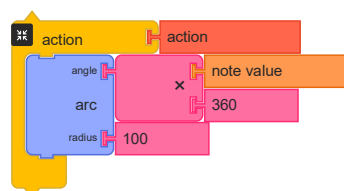
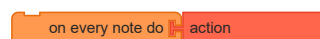




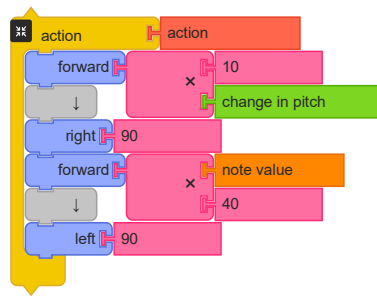
Another approach to graphics is to use modulate them based on the **beat**. In the example above, we call the same graphics action for each note, but the parameters associated with the action, such as pen width, are dependent upon which beat we are on. On Beat 1, the pen size is set to 50 and the volume to 75. On Beat 3, the pen size is set to 25 and the volume to 50. On off beats, the pen size is set to 5 and the volume to 5. The resultant graphic is shown below.



The *On-Every-Note-Do* block lets you specify an action to take whenever a note is played. In the example above, the note value is used to determine the portion of an arc to draw, i.e., a 1/4 note draws a 1/4 circle, a 1/2 note draw 1/2 circle, and a whole note draws a full circle.



The *On-Every-Note-Do* block is found in the Crab Canon project on the Planet to “plot the music”. The mouse moves up and down based on the change in pitch between notes and to the right in proportion to the note value.



## RUN LIVE

Music Blocks has an internal “conductor” maintaining the beat. When the Run button is clicked, the program begins and an internal master (or “conductor”) clock starts up. All of the music tries to stay synced to that clock.



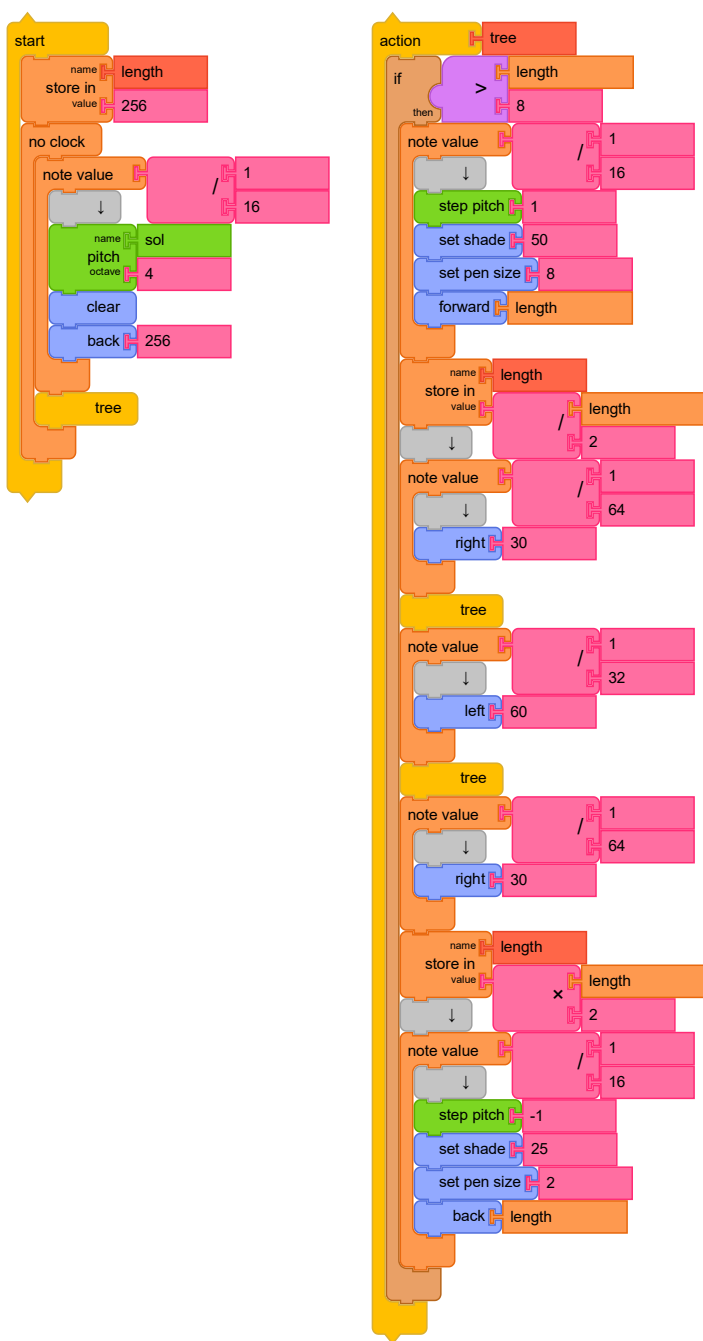
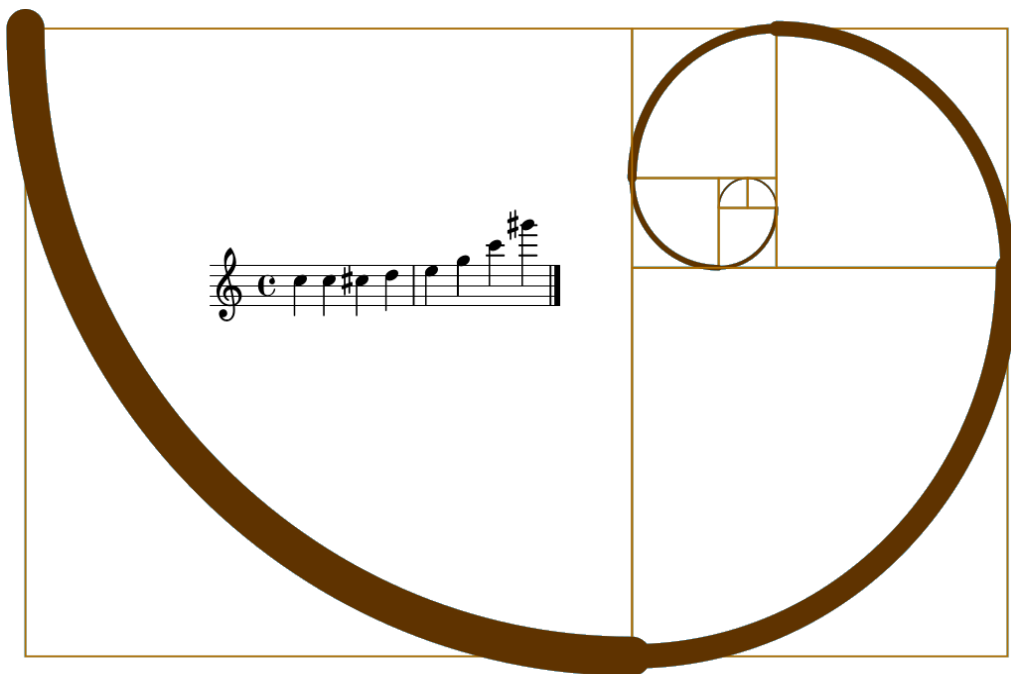
For example, if you have multiple voices (mice), they all share the same conductor in order to keep on the same beat. If a voice (mouse) is falling behind, Music Blocks tries to catch up on the next note by truncating it. If it is an 1/8 note behind and the next note is a 1/2 note, then only an 3/8 note would be played, so as to catch up. That is a somewhat extreme example—usually the timing errors are only very very small differences.

But in some situations, the timing errors can be very large. This is when the *No-clock* block is used.

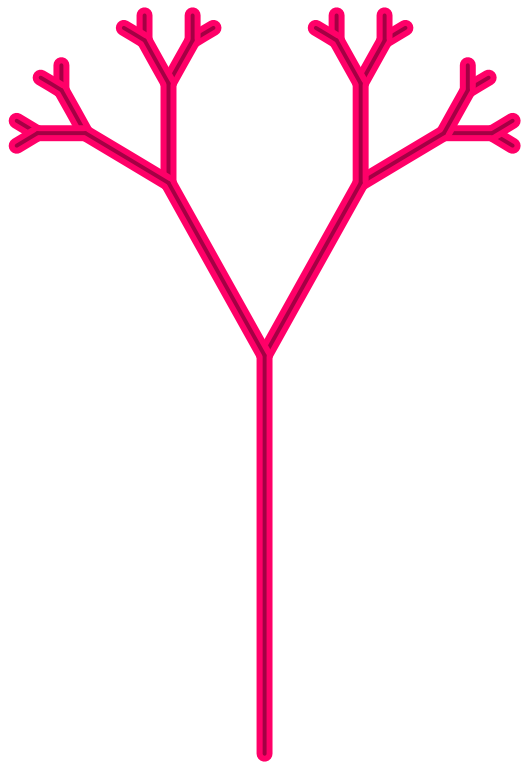
A typical problem is when the music is not played continuously. Imagine an interactive game where a hero is battling a monster. Our hero plays theme music whenever the monster is defeated. But that might occur at any time, hence it is not going to be in sync with the conductor. The offset could be tens of seconds. This would mean that all of the notes in the theme music might be consumed by trying to catch up with the conductor. The *No-clock* block essentially says, do your own thing and don’t worry about the conductor.



In this example, because the computation and graphics are more complex, a *No-clock* block is used to decouple the graphics from the master clock. The “No-clock” block prioritizes the sequence of actions over the specified rhythm.



Another example of embedding graphics into notes: in case, a recursive tree drawing, where the pitch goes up as the branches ascend.



RUN LIVE