Sonic Pi Generative Music Unit Plan
Lesson # 7 - Random Durations of Notes and Samples

Lesson Objectives

Students will be able to use randomized output to affect the duration of notes and samples.

Suggested Duration

1 period (45 minutes)

NYS Computer Science and Digital Fluency Learning Standards

7-8.CT.7
Design or remix a program that uses a variable to maintain the current value of a key piece of information.

Vocabulary

Duration - The amount of time a note or sample will play for or the amount of time the sleep function will last before moving onto the next line of code

Assessments

- Assess ____. Check for the ability to:
    - Store randomized output into a variable to be used later in their code
    - Uses functions which affect the duration of notes and samples
    - Pass variables storing randomized values into different functions

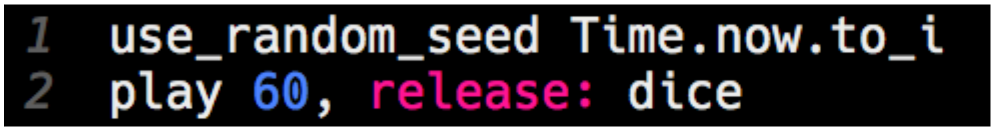
Do Now

Write a line of code that plays a note for longer than one beat.
The length of the note should be chosen randomly.
The length of the randomly chosen note should be different each time.

Reminder: Use attack or release functions to change length of notes.

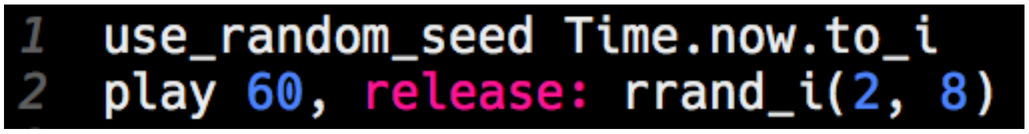
Possible Solutions

dice function:



```
use_random_seed Time.now.to_i
play 60, release: dice
```

rrand_i function



```
use_random_seed Time.now.to_i
play 60, release: rrand_i(2, 8)
```

Have students share out solutions. When one solution has been present, ask students to raise their hand if they have the same/similar solution. Look for different solutions, all of which are acceptable.

Lesson

Part 1 - Random Note Durations

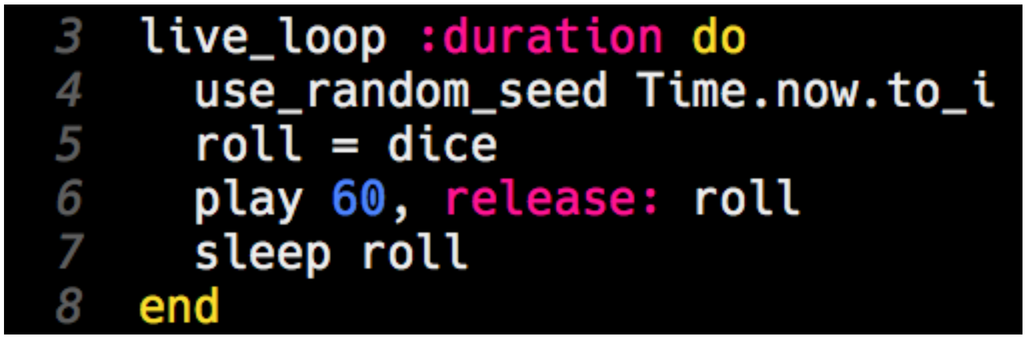
1. Have students put this code into a live loop, so that each time through the loop the note will play for a different length of time.

   In order to do this, we need to make the sleep value the same as the value passed to the attack or release function.

   Have students turn and talk to discuss how they can do this.

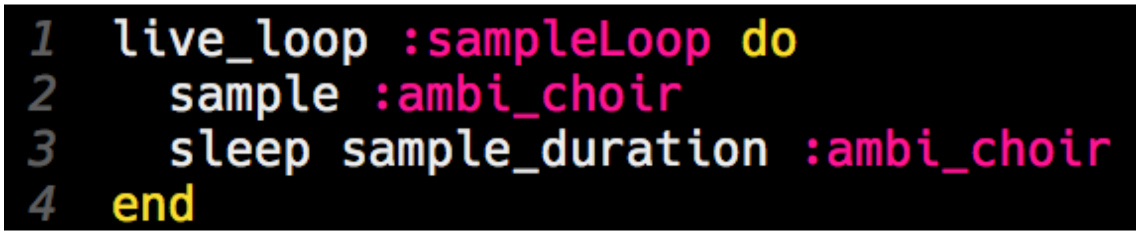
2. Have students share out possible solutions.
   Common Misconception: Using the same function (dice, rrand_i) as a sleep value.
   Remind students that we saw this when working with conditional statements. Calling those functions more than once is like flipping two coins or rolling two dice, not getting different results from the same coin or die.

3. Solution: We need to store the output from these function in a variable which we can use for both the attack/release argument and the sleep value.


```
live_loop :duration do
  use_random_seed Time.now.to_i
  roll = dice
  play 60, release: roll
  sleep roll
end
```

Part 2 - Random Sample Durations

1. In a new buffer, have students create a live_loop that plays a sample and sleeps for the duration of that sample

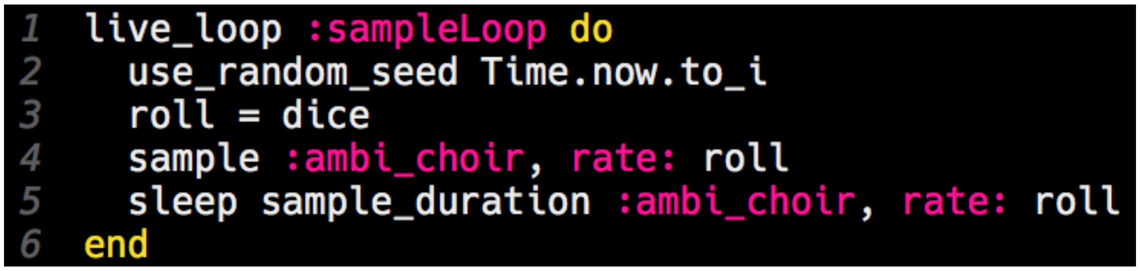

```
live_loop :sampleLoop do
  sample :ambi_choir
  sleep sample_duration :ambi_choir
end
```

2. Have students include the rate function to change the length of the sample. Tell them have the rate of the sample be chosen randomly each time through the loop
   Hint: Use same strategy as for note duration (create a variable to store random value)

   <u>**Common Misconception**</u>
   Students forget that the sample_duration also needs to include the rate: function
   Because rate changes the duration of the sample

3. Have students share out solutions.

   Possible solution:



```
live_loop :sampleLoop do
  use_random_seed Time.now.to_i
  roll = dice
  sample :ambi_choir, rate: roll
  sleep sample_duration :ambi_choir, rate: roll
end
```

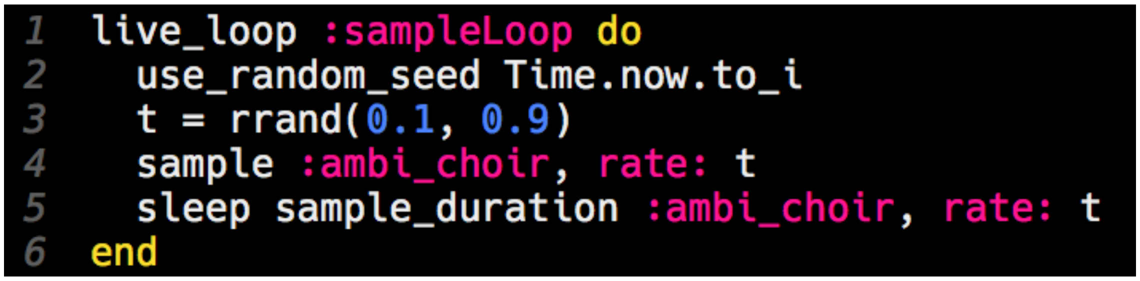
4. Ask students how does rate affect how the sample sounds
   Possible responses: It sounds higher in pitch, it plays faster

5. Point out that numbers passed to the rate: function that are above 1 will cause the duration to be faster (which also affects the pitch).

6. Introduce ***rrand***

   rrand will return a random value between specified range of numbers that includes a decimal point number. This can be used for any range of numbers, but is also useful for choosing random numbers that are less than 1.

   Example Code:

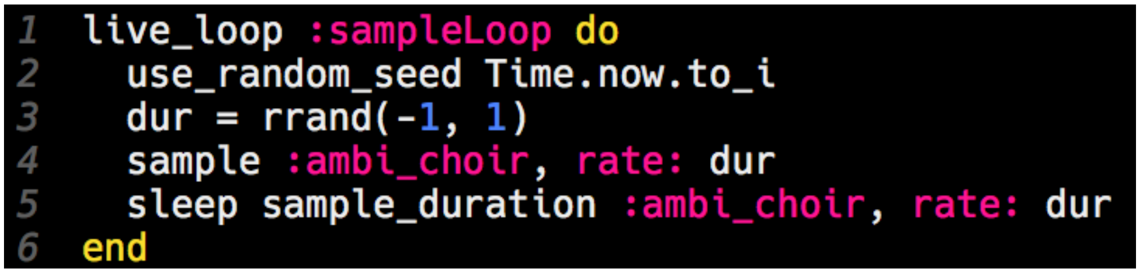


```
live_loop :sampleLoop do
  use_random_seed Time.now.to_i
  t = rrand(0.1, 0.9)
  sample :ambi_choir, rate: t
  sleep sample_duration :ambi_choir, rate: t
end
```

Remind students that when we use numbers that are less than 1 with the rate function, it will cause the duration of the sample to be slower.

7. ***rrand*** can also include negative numbers (remind students that using negative numbers for the rate function will cause the sample to play backwards)

   Example code:



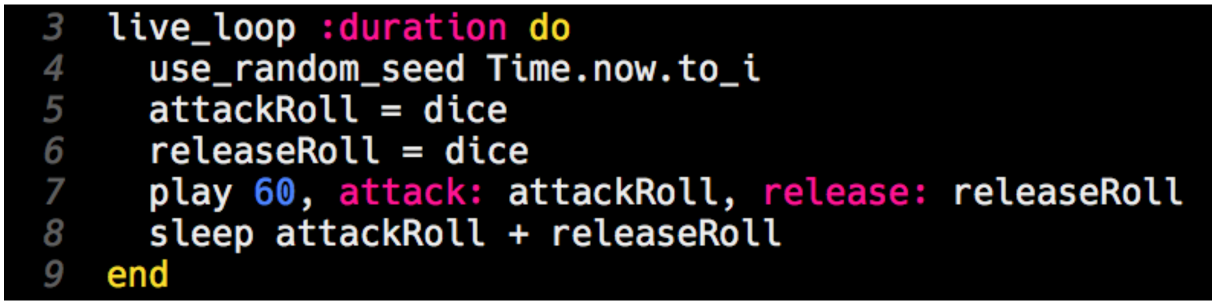
```
live_loop :sampleLoop do
  use_random_seed Time.now.to_i
  dur = rrand(-1, 1)
  sample :ambi_choir, rate: dur
  sleep sample_duration :ambi_choir, rate: dur
end
```

Wrap Up/Assessment

Students may complete one of the two following programs.

1. Use both attack and release with the play function but have different attack and release values chosen randomly and sleep for the correct amount

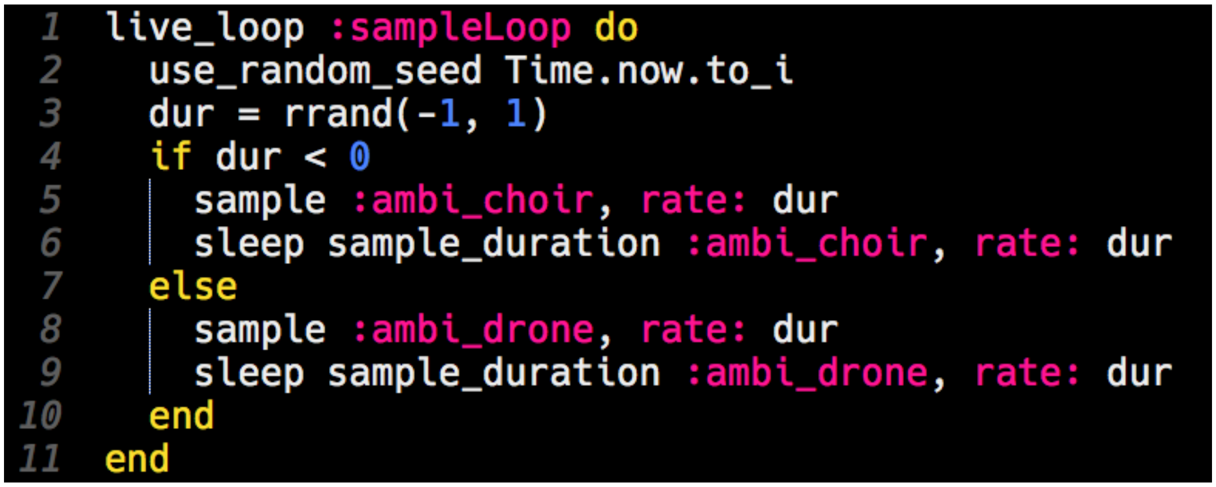
   <u>Example of Finished Code:</u>



```
live_loop :duration do
  use_random_seed Time.now.to_i
  attackRoll = dice
  releaseRoll = dice
  play 60, attack: attackRoll, release: releaseRoll
  sleep attackRoll + releaseRoll
end
```

2. Use a conditional statement to randomly choose between two different samples that play at a randomly chosen rate. The value used for the rate should also be used as the value in the conditional statement

   <u>Example of Finished Code:</u>



```
live_loop :sampleLoop do
  use_random_seed Time.now.to_i
  dur = rrand(-1, 1)
  if dur < 0
    sample :ambi_choir, rate: dur
    sleep sample_duration :ambi_choir, rate: dur
  else
    sample :ambi_drone, rate: dur
    sleep sample_duration :ambi_drone, rate: dur
  end
end
```

Rubric: Lesson 7 - Random Durations Assignment Checklist Assessment Tool

<u>Random Durations Assignment Checklist</u>

Each part is worth a total of 2 points.
2 - Accurately completes requirement
1 - Requirement is attempted but not completed accurately
0 - Does not include requirement

Task completed (Circle one) - Note Duration | Sample Rate

_______Uses true randomness

_______Uses method to choose random value(s)

_______Stores random value(s) in variable(s)

_______Correctly uses variable(s) as argument for function which affects duration

_______Sleep function value is correct length based on random value used for duration

_______ = Total Score out of 10