Sonic Pi Generative Music Unit Plan
Lesson # 2 - Pseudo vs True randomness

Lesson Objectives

Students will be able to
- To understand the difference between true random and pseudorandom
- Include a method to generate true random output in their code

Suggested Duration

1 period (45 minutes)

NYS Computer Science and Digital Fluency Learning Standards

7-8.CT.1 Compare the results of alternative models or simulations to determine and evaluate how the input data and assumptions change the results.

Vocabulary

Pseudorandom - A sequence that appears to be statistically random, despite having been produced by a completely deterministic and repeatable process

Assessments

- Assess ____. Check for the ability to:
  - Write a block of code to produce a random sequence of numbers
  - Use the console to obtain information about their program
  - Determine the difference between true random and pseudorandom
  - Use a method to generate True random sequences

Do Now

Have students write out a series of 8 numbers between 40 and 100
Have students compare their series of numbers with students around them.

Ask how many students had the same sequence of numbers.
Establish that it would be very rare for multiple people to come up with the same sequence of randomly chosen numbers.

Lesson

Part 1 - Generating random numbers

1. Introduce students to ***rrand_i( ).***
   This function will choose a random integer between a specified range of numbers. The function takes two parameters, the minimum and maximum value of the range
   Example: rrand_i(10, 20) will return a number between 10 and 20.

2. Have students write a short program in Sonic Pi that will recreate the Do Now which uses the ***rrand_i*** function. However, tell them this program should play each value and then sleep for a certain amount of time.

3. Allow students 3-5 minutes to complete.
   Remind students who are stuck that you need to choose a random number 8 times, so they should include some type of loop.

Example of completed code

```
2  8.times do
3    play rrand_i(40, 100)
4    sleep 1
5  end
```

Part 2 - Pseudorandom

1. Have students run their program and write down the notes that appear in the console when the program has finished.

   Students should then compare their results with the students around them. Students should observe that everyone got the same results.

   Example of console output

```
{run: 5, time: 0.0}
 └─ synth :beep, {note: 85.0}

{run: 5, time: 1.0}
 └─ synth :beep, {note: 84.0}

{run: 5, time: 2.0}
 └─ synth :beep, {note: 68.0}

{run: 5, time: 3.0}
 └─ synth :beep, {note: 54.0}

{run: 5, time: 4.0}
 └─ synth :beep, {note: 46.0}

{run: 5, time: 5.0}
 └─ synth :beep, {note: 72.0}

{run: 5, time: 6.0}
 └─ synth :beep, {note: 90.0}

{run: 5, time: 7.0}
 └─ synth :beep, {note: 73.0}
```

Note: Students who do not get the same results may have included *use_random_seed* which will be discussed next

2. Explain to students that Sonic Pi, along with many other random number generators, does not actually generate random sequences of numbers. (The purpose of this is so that if you create a random number sequence that sounds good, you will be able to repeat it)

3. Ask students how we could fix this program to get different sequences of numbers. Students should recall ***use_random_seed*** from a previous lesson.

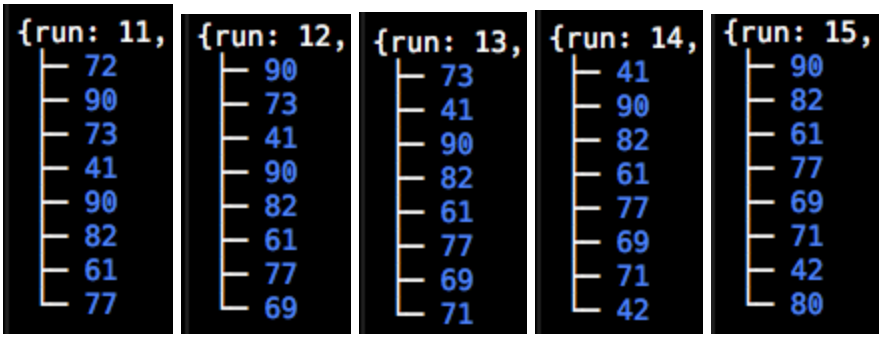
4. Have students add ***use_random_seed*** to the top of their code, but assign each student a specific number to use as an argument for that function. (1, 2, 3, 4, 5, 6, 7, 8 etc)

```
1  use_random_seed 5
2  8.times do
3    play rrand_i(40, 100)
4    sleep 1
5  end
```

Students should write down the sequence they get. Have them compare with each other. Now they should see that they are getting different sequences.

5. Ask for the results of 5-6 students who had sequential seed numbers (4-10, 11-16 etc) Write the results on the board

   Output examples (use_random_seed 5 - 9)



   Ask students if they see a pattern. They should notice that each sequence contains 7 of the same numbers as the previous sequence.

6. Explain that when Sonic Pi generates a number sequence using use_random_seed, it will always make the same number sequence (which is millions of numbers long) but it will displace the first number for each increment of the number passed as an argument in use_random_seed. This is what we call Pseudorandom. It appears random but it actually will produce the same results.

Part 3 - True Randomness

1. All random functions in Sonic Pi generate pseudorandom patterns, so even if we tried to do something like this:
   use_random_seed rrand_i(1, 100000)
   It will still always create the same pattern.

2. Introduce **Time.now**
   This function will output the current date and time
   Demo the following code for students

   puts Time.now
   Point out the output on in the console.
   2021-11-29 22:04:49.908957 -0500

3. We can convert this output into an integer by adding ***.to_i*** to the end of this function.

   Demo the following code for students

   puts Time.now.to_i
   Point out the output on in the console.
   1638241639

Run the code again and point out that the number output in the console is different

4. We can now use this function output as the argument to **use_random_seed** and will always get a different number which will change depending on the day and time we run our program. This is as close as we can get to true randomness in our code.

5. Have students use Time.now.to_i for use_random_seed and run their code. Have some students wait a few seconds before running their code to ensure they get different results.

   Ask students why they should wait instead of running their code at the same time
   (If they run their code at the same time, they will get the same output since the number is based on the date and time.)

   Example of code

```
use_random_seed Time.now.to_i
8.times do
  play rrand_i(40, 100)
  sleep 1
end
```

6. Explain that having random number sequences generated by using some outside influence to determine the seed is called True Random.

Wrap Up/Assessment

Have students read the following webpage which discusses the differences between PseudoRandom and True Random number generators.
When finished, students should complete the Lesson 2 - PseudoRandom Exit Slip

PseudoRandom Exit Slip

Answer the Following Questions

1. True / False (circle one) - The following code will always produce the same sequence of notes

```
use_random_seed rrand_i(1, 10000)
16.times do
  play rrand_i(40, 80)
  sleep 1
end
```

Explain why you chose your answer.

2. What is the difference between Pseudo Random and True Random?

3. Why does ***Time.now.to_i*** always provide us with a different number that we can use in the ***use_random_seed*** function?