RAINBOW KEYBOARD

GAME SETUP

2–4
PLAYERS
■■□□

6
SPLATS
■■■■■■

ROCK ON!

GAME SUMMARY

UNRULINESS: Standing, Walking

GAME RULES: Be Unruly. Play MIDI keyboard in style!

Build a rainbow piano keyboard using MIDI. The Splats will keep cycling rainbow colors as students step on Splats to make some Unruly music! Start with one note per splat, or add multiple notes for some more intense Unruly compositions.

HOW IT WORKS

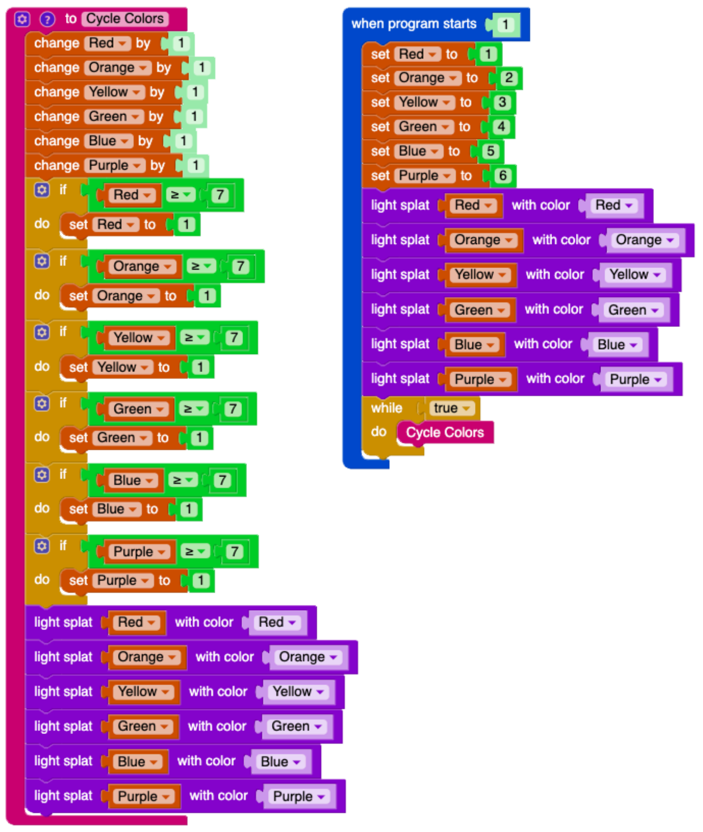
This program uses a function, **CYCLE COLORS**, to create a moving rainbow pattern. This code also uses a **'WHILE/TRUE'** loop to keep that color pattern moving as long as the program is running.

As the program runs, the different colors are given an order, one through six. The **CYCLE COLORS** function, when called, moves each color up to the next Splat in line by adding one to its value.

Each color then lights its numbered Splat the correct color, and if that color was the last in line, it loops back to Splat number one to keep the cycle going.

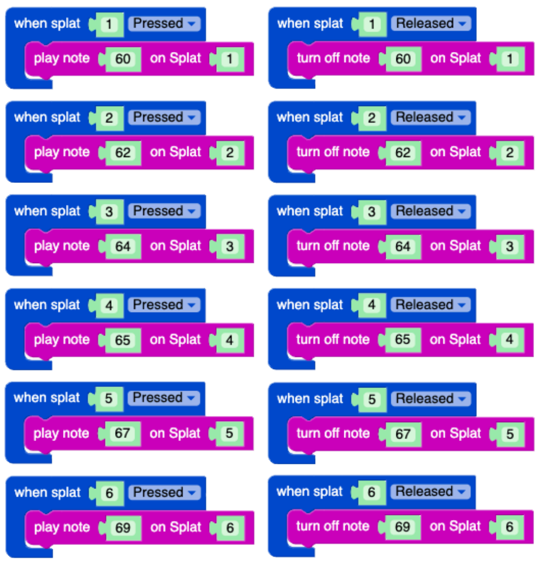
The rest of the code blocks are to have the Splats play MIDI notes when pressed, to make a fun six key keyboard.

Experiment by changing the colors and the MIDI note values to alter the tones being played.



```
to Cycle Colors
  change Red by 1
  change Orange by 1
  change Yellow by 1
  change Green by 1
  change Blue by 1
  change Purple by 1
  if Red ≥ 7
  do set Red to 1
  if Orange ≥ 7
  do set Orange to 1
  if Yellow ≥ 7
  do set Yellow to 1
  if Green ≥ 7
  do set Green to 1
  if Blue ≥ 7
  do set Blue to 1
  if Purple ≥ 7
  do set Purple to 1
  light splat Red with color Red
  light splat Orange with color Orange
  light splat Yellow with color Yellow
  light splat Green with color Green
  light splat Blue with color Blue
  light splat Purple with color Purple

when program starts 1
  set Red to 1
  set Orange to 2
  set Yellow to 3
  set Green to 4
  set Blue to 5
  set Purple to 6
  light splat Red with color Red
  light splat Orange with color Orange
  light splat Yellow with color Yellow
  light splat Green with color Green
  light splat Blue with color Blue
  light splat Purple with color Purple
  while true
  do Cycle Colors
```



```
when splat 1 Pressed
  play note 60 on Splat 1
when splat 1 Released
  turn off note 60 on Splat 1

when splat 2 Pressed
  play note 62 on Splat 2
when splat 2 Released
  turn off note 62 on Splat 2

when splat 3 Pressed
  play note 64 on Splat 3
when splat 3 Released
  turn off note 64 on Splat 3

when splat 4 Pressed
  play note 65 on Splat 4
when splat 4 Released
  turn off note 65 on Splat 4

when splat 5 Pressed
  play note 67 on Splat 5
when splat 5 Released
  turn off note 67 on Splat 5

when splat 6 Pressed
  play note 69 on Splat 6
when splat 6 Released
  turn off note 69 on Splat 6
```

MIDI NOTE MUSICAL CONVERSON GUIDE

MIDI values are numbers that represent notes on a regular piano keyboard. For Splats, these notes range from the very low ‘C1 (24)’ all the way up to the very high ‘C8 (108)’.

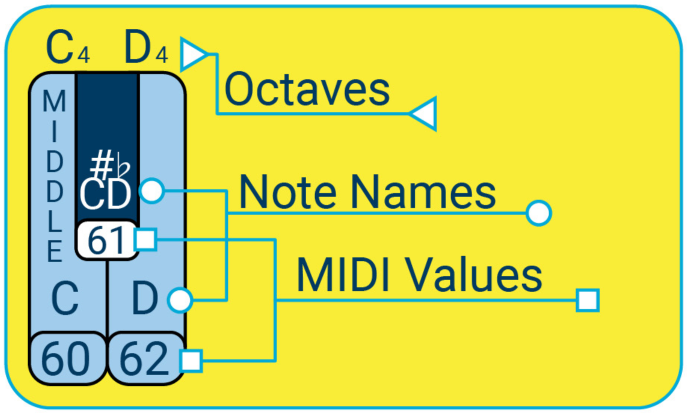
Because there are 12 notes from C to B in each octave, you can go up or down an octave by adding or subtracting 12 from that note’s MIDI value.

In the Little Lamb example, the same notes could be played, just an octave higher in pitch, by adding 12 to each of the notes: 64, 62, 60, 62, 64, turns into 76, 74, 72, 74, 76.

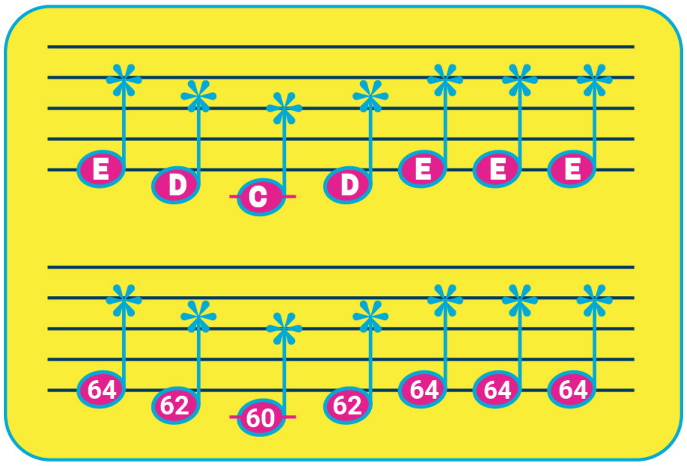
These MIDI conversions don’t have to be limited to single notes either! You can build and experiment with basic three or five note chords played at the same time.

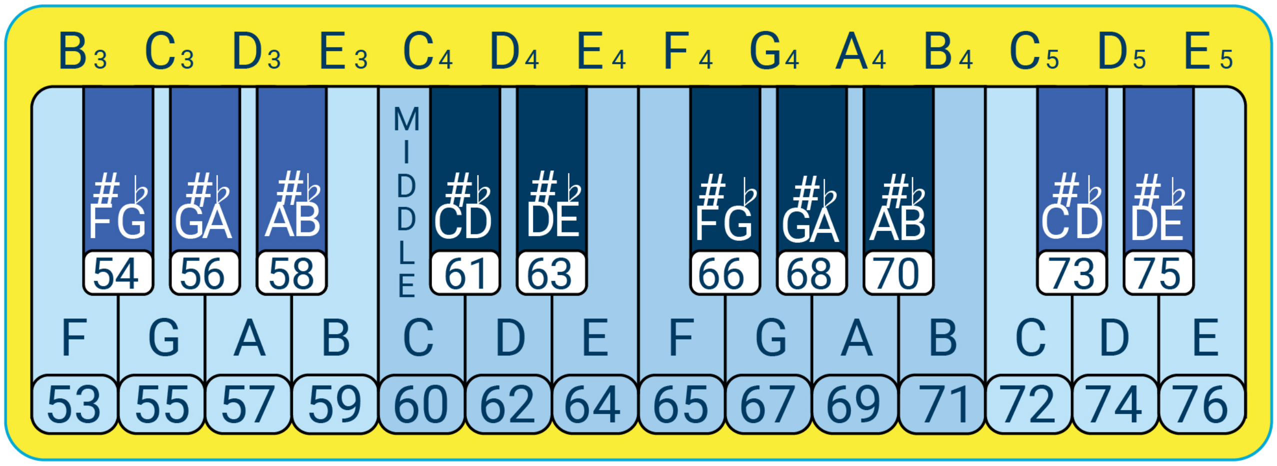
Some examples are C (60, 64, 67), E (64, 68, 71), and G# (68, 72, 75) or even a C major pentatonic (60, 62, 64, 67, 69).

KEY



EXAMPLE: MARY HAD A LITTLE LAMB





SUGGESTED OUTLINE

1 **INTRODUCE EXERCISE**
Introduce the activity and show how the keyboard works. Work through a diagram of the Splats keyboard and outline the key objectives for the program.

2 **GUIDED WORK TIME**
Review the MIDI conversion guide on the previous page with your class. Go through the rainbow variables and talk through the Cycle Colors variable. Compare how **WHEN SPLAT PRESSED** and **WHEN SPLAT RELEASED** are used in this program. Talk about the role variables play in cycling the colors and the benefits of using a function.

3 **GROUP WORK TIME**
Support students as they build their keyboards. Ensure students are playing a variety of roles in their groups, including testing, documenting, and coding. Encourage different groups to work together on challenges or even share different parts of their code!

4 **STUDENT SHOWCASE!**
Give groups time to share their Splat keyboards with the class.

GOING FURTHER

EXTENSION
Have students compose a song and share it with the class. Students can also build a different scale.

SUPPORT
Have students diagram their piano before building. Provide the MIDI numbers and build the **WHEN PROGRAM STARTS** code as a class.

CSTA STANDARDS

ALGORITHMS & PROGRAMMING

GRADES 6—8

2-AP-10 ALGORITHMS	Use flowcharts and/or pseudocode to address complex problems as algorithms. **(P4.4, 4.1)**
2-AP-11 VARIABLES	Create clearly named variables that represent different data types and perform operations on their values. **(P5.1, 5.2)**
2-AP-13 MODULARITY	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. **(P3.2)**
2-AP-14 MODULARITY	Create procedures with parameters to organize code and make it easier to reuse. **(P4.1, 4.3)**
2-AP-17 DEVELOPMENT	Systematically test and refine programs using a range of test cases. **(P6.1)**
2-AP-19 DEVELOPMENT	Document programs in order to make them easier to follow, test, and debug. **(P.7.2)**

CSTA STANDARDS

ALGORITHMS & PROGRAMMING

GRADES 3—5

1B-AP-08 ALGORITHMS	Compare and refine multiple algorithms for the same task and determine which is the most appropriate. **(P6.3, 3.3)**
1B-AP-10 CONTROL	Create programs that include sequences, events, loops, and conditionals. **(P5.2)**
1B-AP-11 MODULARITY	Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. **(P3.2)**
1B-AP-13 DEVELOPMENT	Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences. **(P.1.1, 5.1)**
1B-AP-14 DEVELOPMENT	Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. **(P.6.1,6.2)**
1B-AP-15 DEVELOPMENT	Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. **(P.6.1, 6.2)**
1B-AP-17 DEVELOPMENT	Describe choices made during program development using code comments, presentations, and demonstrations. **(P.7.2)**