MORE

ACTIVITY PACK

WITH

MORE

10 LESSON PLANS

!

LEARN
PLAY

MORE

FOR USING

unruly
Splats™

CODE
JUMP
RUN

MORE

MORE

IN YOUR

CLASSROOM

!!

HAVE FUN

MORE

BE UNRULY ! ! ! ! ! ! ! ! ! ! !

unruly
STUDIOS

2

Vers. 3_24_20_SEL

DEAR EDUCATOR

Here are ten MORE lessons for using Splats in your classroom. They are designed to go further than our first Activity Pack, focusing on fun, creativity, and more advanced coding concepts. We had a blast creating these activities-- we hope you and your students enjoy them!

We frequently update our lessons, so make sure to come back and check out our resources from time to time. Our Twitter page, @unruly_studios is becoming a great resource for finding and sharing classroom Unruliness! Please share your creations and tag them with #UnrulyEdu.

We want to hear from you! Please email education@unruly-studios.com with any questions or feedback.

Thanks,
THE UNRULY TEAM

SOCIAL EMOTIONAL LEARNING WITH SPLATS

At Unruly, our core values are play and exploration. We are always thinking of ways to make learning meaningful and playtime memorable. At the end of every activity, we like to ask students:

1. "Did you have fun?"
2. "Did you learn something?"

If the answer to either is "No," then the next step is to figure out what got in the way. Some issues can be resolved over time with accommodations and practice. Often, however, when building and playing games in groups, you may find that many of the barriers to learning and fun can be traced back to foundational gaps in Social Emotional Learning (SEL).

Since organic play naturally involves many critical components of SEL, we believe Unruly Splats to be an excellent vehicle for exploring complex social concepts. If you're currently using Splats in the classroom, you'll notice students persevering through difficult problems, resolving conflicts, and working collaboratively to achieve goals. On the other hand, you might also notice students having difficulties self-monitoring, becoming frustrated, arguing with classmates, or fighting over materials.

We realize these challenges can get in the way of learning, but we also see them as great opportunities to grow classroom community. Though our activities are often STEM-focused and project-based, we believe it is so important to recognize SEL challenges. There is great power in pausing, reflecting, and persevering in order to start again with fresh perspectives. In other words, don't be afraid to stop an activity and try again another day! Some programs come together in a snap, and some are best enjoyed over time and with the right mindset.

Using the CASEL framework as a guide, we've created an activity pack that helpsw facilitate conversations around self-management, self-awareness, social awareness, relationship skills, and responsible decision-making. Though we recognize that certain SEL competencies may need intentional focus, we believe Splats can be a great way to get the conversation started in impactful and authentic ways.

To learn more about how we are addressing SEL needs in the classroom, please see our CASEL reference guide below and check out our **Social Emotional Learning with Splats** activity pack!

CASEL CORE COMPETENCIES	UNRULY PROCESS
SELF-AWARENESS: ☐ Identifying emotions ☐ Accurate self-perception ☐ Self-confidence ☐ Self-efficacy	When coding alone or with partners, students must tackle the code confidently. They need to be efficient with their time and keep track of their process to avoid making the same mistakes. This process can be very frustrating, so students need to identify their emotions in order to appropriately respond. **STUDENTS SHOULD ASK THEMSELVES:** • What do I know how to do? • How much can I handle on my own? • Do I want help, or do I need help? • If I need help, how should I ask for help? • Should I take a break if I am getting frustrated? • What is the appropriate way/amount of time to take a break?
SELF-MANAGEMENT: ☐ Impulse control ☐ Stress management ☐ Self-discipline ☐ Intrinsic motivation ☐ Goal-setting, organizational skills	Splats are made to be stomped on, poked, pushed, slapped—you name it, it can endure it! This doesn't mean that every student gets to play with the Splats at once. Students need to exercise not just their minds and bodies, but their self-discipline and impulse control. **STUDENTS SHOULD ASK THEMSELVES:** • What is my goal or job? • Do I know how to start? • Am I using my time wisely? • Am I "stuck" on a problem? If so, should I move on and come back to it later? • Am I using the materials as tools and not toys?

SOCIAL AWARENESS:

- ☐ Ability to see other's perspectives
- ☐ Empathy
- ☐ Appreciating diversity
- ☐ Respectfulness

When it comes to Splats, the more the merrier! Our activities often encourage partners or teams. This means students need to be able to work in diverse groups and understand that everyone has different strengths.

STUDENTS SHOULD ASK THEMSELVES:

- Am I sharing air time and materials?
- Am I speaking appropriately and respectfully?
- If someone is having a hard time, how can I help?
- Even if I'm not working with my friends, can I still have fun?

RELATIONSHIP SKILLS:

- ☐ Communication
- ☐ Social engagement
- ☐ Relationship-building
- ☐ Teamwork

Not everyone can win a game, but everyone can still have fun! It's important to show good sportsmanship so that everyone can learn and enjoy the experience.

STUDENTS SHOULD ASK THEMSELVES:

- Am I doing my part for my team?
- Am I listening? If not, what can I fix?
- Am I being heard? If not, what can I do?
- Am I showing good sportsmanship?

RESPONSIBLE DECISION-MAKING:

- ☐ Identifying problems
- ☐ Analyzing situations
- ☐ Solving problems
- ☐ Evaluating
- ☐ Reflecting

Conflicts are inevitable when working in a team. Frustration is also inevitable when coding! It's important to recognize when negative feelings are clouding our judgment. We encourage students to talk through social conflicts using "I" statements to avoid misplacing blame.

Likewise, we encourage students to persevere through difficult activities. Some programs, just like relationships, take time!

STUDENTS SHOULD ASK THEMSELVES:

- What's tricky for me?
- What tools are available for help?
- Which part of the problem can I work on right now?
- If I avoid a problem, does that mean it goes away?
- How have I seen other people handle this problem?

TABLE OF CONTENTS

ACTIVITIES

ACTIVITIES

SPLAT JACKS

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES 3-5
UNRULINESS	JUMPING IN PLACE
GROUP SETUP	2-4 STUDENTS SHARE 1 DEVICE. GAME PLAY USES 2 SPLATS.

GAME SUMMARY

Build a program where Bill, your trainer, will keep you in good form for jumping jacks (Splat jacks). If done properly, Bill will let you know with a quack! (Bill is a duck.)

GAME RULES: DO JUMPING JACKS WITH BILL

NOTES

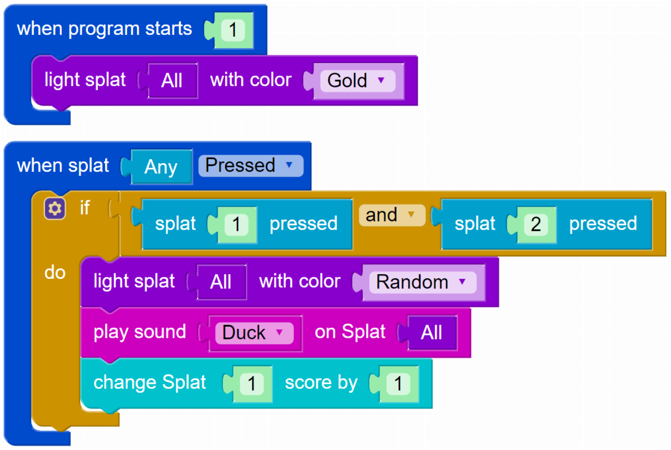
CODE KEY

HOW IT WORKS

This code counts jumping jacks with conditional statements. Using the **and** connector with **if/then** blocks; the program senses when both splats are jumped on at the same time.

If Splat 1 and Splat 2 are pressed at the same time, both Splats will light up a random color, make a duck sound, and add a point to the score!

CODE IMAGE



```
when program starts 1
    light splat All with color Gold ▾

when splat Any Pressed ▾
    if splat 1 pressed and ▾ splat 2 pressed
    do light splat All with color Random ▾
       play sound Duck ▾ on Splat All
       change Splat 1 score by 1
```

ACTIVITY OUTLINE

1. INTRODUCE EXERCISE

Introduce the activity. Explain that students will be writing a program to count jumping jacks. Review the essential blocks, focusing on the **if/do** and sensing blocks.

2. WORK TIME

Have students brainstorm different ways to count jumping jacks using Splats. Support groups as they build their programs. Encourage groups to add additional blocks, or Splats! Have students build multiple mini-programs and test them out. Lead a full-class discussion to compare and contrast different options that groups developed.

3. GAME PLAY

Give groups time to test and play their games.

4. STUDENT SHOWCASE!

Have groups show off their Splatjack counter. If time allows, lead a competition!

GOING FURTHER

EXTENSIONS

Students can add a stopwatch or countdown timer for a Splat Jacks competition. By copying this code and changing the pair of Splats numbered in the **and** block, Splat pairs 3 + 4 and 5 + 6 could be used for up to three players competing at the same time!

SUPPORT

Talk through what would happen if the code did not include the if/do block or the and connector block. These blocks are an easy way for us to add points only if someone does a correct Splatjack by stepping on both Splats at once.

CSTA STANDARDS: ALGORITHMS AND PROGRAMMING

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - 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-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-15 Development	Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. (P.6.1, 6.2)

SPLAT BRIDGE RELAY

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES 3-5
UNRULINESS	WALKING, JUMPING
GROUP SETUP	2-6 STUDENTS PER GROUP. GAME PLAY REQUIRES 2 SPLATS

GAME SUMMARY

The river is flooded and all you have to cross it with are your trusty Splats! With both feet on one Splat, move your other Splat further across the river. Repeat until you make it to the other side. The next player can begin once you make it across. The first team to cross the river safely wins! Game requires a start and finish point-- any distance apart.

GAME RULES: KEEPING BOTH FEET ON SPLATS, CROSS THE GAP WITHOUT FALLING OFF

NOTES

CODE KEY

HOW IT WORKS

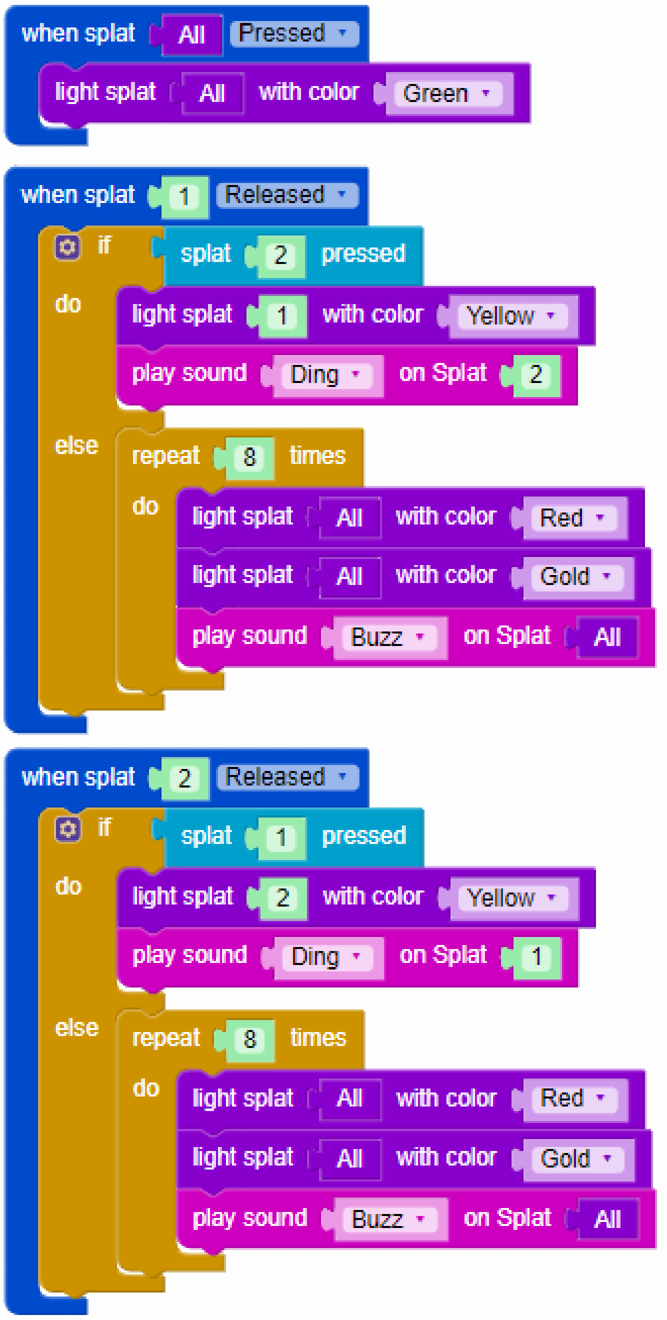
In order to make sure one Splat is always pressed, and that the player hasn't fallen off, this code uses **when splat released** blocks. If Splat one is released when Splat two is pressed (or vice versa), the player can continue their journey.

This code is a great example of using a **repeat/do** block to clearly show the player when the fail state of a game is triggered.

If ever a Splat is released while the other one isn't being pressed, the fail sequence will be triggered: they will both flash red and gold and buzz repeatedly.

The **when splat released** block lights either Splat as yellow when ready to be stepped on. For increased game feedback, the **when splat pressed** block lights the Splats green to show the game is still running, and the player has made a step successfully.

CODE IMAGE



```
when splat All Pressed
    light splat All with color Green

when splat 1 Released
    if  splat 2 pressed
    do  light splat 1 with color Yellow
        play sound Ding on Splat 2
    else
        repeat 8 times
        do  light splat All with color Red
            light splat All with color Gold
            play sound Buzz on Splat All

when splat 2 Released
    if  splat 1 pressed
    do  light splat 2 with color Yellow
        play sound Ding on Splat 1
    else
        repeat 8 times
        do  light splat All with color Red
            light splat All with color Gold
            play sound Buzz on Splat All
```

SPLAT BRIDGE RELAY

ACTIVITY

ACTIVITY OUTLINE

1. INTRODUCE EXERCISE

Introduce the activity. Explain the game rules and demonstrate how to play the game. Lead the class in identifying the objectives of their program, breaking down problems, and creating flowcharts / lists of tasks.

2. WORK TIME

Introduce the essential blocks and tie them directly to the game rules. Highlight the **when splat released** blocks and have students work together to build the rest of the code. Support groups in sharing note taking, testing, and coding roles.

3. GAME PLAY

Have students test out their programs as a group. Encourage students to brainstorm and test other innovative ways to cross the river. There will be a lot of fail noises!

4. STUDENT SHOWCASE!

It is relay time! Set up your “river” and line groups up for the relay.

GOING FURTHER

EXTENSIONS

Students can add code for more players, add a stopwatch or timer, and even change the code so the crossing can be done in one full group.

SUPPORT

Highlight the difference between when splat pressed and when splat released. Why do we use released in this program?

CSTA STANDARDS: ALGORITHMS AND PROGRAMMING

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - GRADES 3-5

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-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-16 Development	Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. (P2.2)
1B-AP-17 Development	Describe choices made during program development using code comments, presentations, and demonstrations. (P.7.2)

THUMB WAR

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES 3-8
UNRULINESS	SITTING
GROUP SETUP	GAME PLAY USES 1-6 SPLATS ON 1 DEVICE. ONE SPLAT PER TEAM. EACH TEAM CAN HAVE 1-4 STUDENTS.

GAME SUMMARY

Each student picks a corner of a Splat, and is assigned a color: Red, Green, Blue, or Yellow. One color at a time, in that order, students take turns pressing their corner's button as many times as they can. Each press is a point, and the Splat team with the highest score after all four colors, wins. It is important that only one button is being pressed at a time to track score properly!

GAME RULES: IN COLOR ORDER, PLAYERS TAKE TURNS PRESSING THEIR BUTTONS AS FAST AS THEY CAN

NOTES

CODE KEY

HOW IT WORKS

This code sets up its game rules, by using simple **when splat pressed** blocks to interrupt blocks already running in the **when program starts** block.

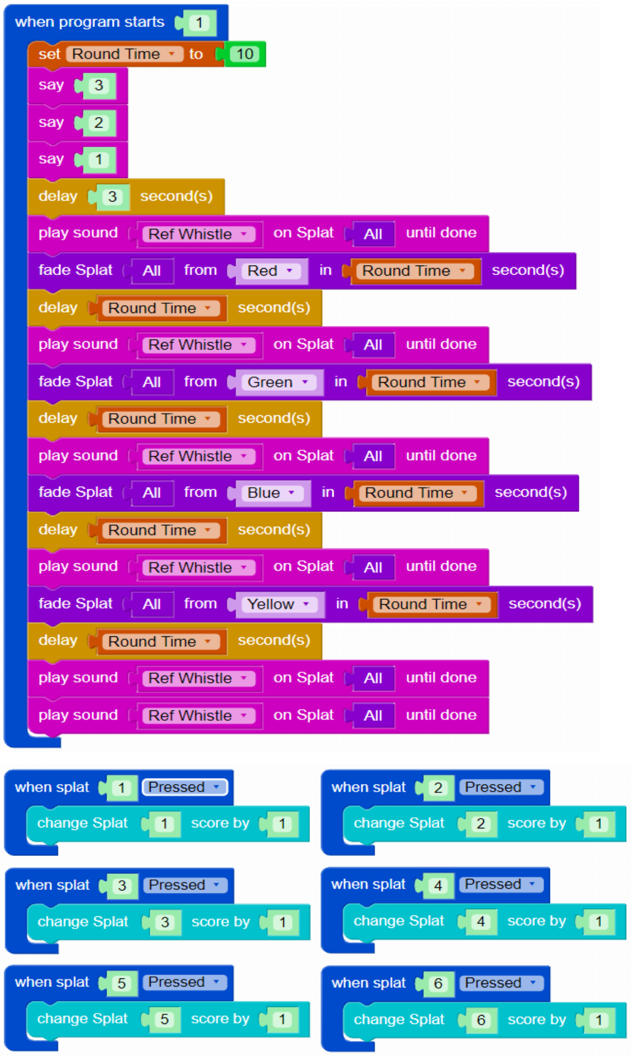
The **Round Time** variable is used to allow easy control over the round length, by changing the delay between each color change cycle.

When the program starts the **say** blocks read out a countdown, and the game delays by the set **Round Time** variable. After the delay, all Splats sound the start whistle, and fade each color over that same **Round Time** duration.

This process is repeated for all four team colors, until the last cycle, when the whistle blows twice and the program ends.

Each **when splat pressed** block increases that Splat's score by one. This mechanic relies on the fact that only one Splat foot needs to be pushed to register a press, and add a score to that team!

CODE IMAGE



```
when program starts 1
  set Round Time to 10
  say 3
  say 2
  say 1
  delay 3 second(s)
  play sound Ref Whistle on Splat All until done
  fade Splat All from Red in Round Time second(s)
  delay Round Time second(s)
  play sound Ref Whistle on Splat All until done
  fade Splat All from Green in Round Time second(s)
  delay Round Time second(s)
  play sound Ref Whistle on Splat All until done
  fade Splat All from Blue in Round Time second(s)
  delay Round Time second(s)
  play sound Ref Whistle on Splat All until done
  fade Splat All from Yellow in Round Time second(s)
  delay Round Time second(s)
  play sound Ref Whistle on Splat All until done
  play sound Ref Whistle on Splat All until done

when splat 1 Pressed
  change Splat 1 score by 1

when splat 2 Pressed
  change Splat 2 score by 1

when splat 3 Pressed
  change Splat 3 score by 1

when splat 4 Pressed
  change Splat 4 score by 1

when splat 5 Pressed
  change Splat 5 score by 1

when splat 6 Pressed
  change Splat 6 score by 1
```

ACTIVITY OUTLINE

1. INTRODUCE EXERCISE

Introduce the activity and assign players to teams: red, green, blue, and yellow. Explain the game rules and demonstrate how to play the game. Explain how only one foot being pressed registers as 'Splat being pressed' and that only one corner can be pressed at a time. This is why the game must be played in rounds.

2. WORK TIME

Introduce the essential blocks and tie them directly to the game rules. Highlight how variables are used to keep track of and change the round time. Build the code together as a class and document what each portion of the program does. Discuss code attribution and how programmers frequently build off of others work, but give credit. Debug and test code in groups.

3. GAME PLAY

Distribute Splats, up to four students per Splat. Lead a few practice rounds!

4. STUDENT SHOWCASE!

Let the games begin! Referee the tournament and track each team's score from round to round.

GOING FURTHER

EXTENSIONS

Students can change the round timing (longer or shorter rounds change game play significantly), team structure, and the ways points work. Rather than players being given colors, they could each be assigned a sound to listen for, and change the code accordingly!

SUPPORT

This program uses a variable for the round time, but it can easily be removed. Just replace the variable with the number of seconds you'd like each round to be.

CSTA STANDARDS: ALGORITHMS AND PROGRAMMING

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - GRADES 3-5

1B-AP-9 Variables	Create programs that use variables to store and modify data. (P5.2)
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-12 Modularity	Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.(P5.3)
1B-AP-14 Development	Observe intellectual property rights and give appropriate attribution when creating or remixing programs.(P5.2, 7.3)
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)

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - 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-19 Development	Document programs in order to make them easier to follow, test, and debug. (P7.2)

ADVENTURE SPLATS

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES 3-8
UNRULINESS	WALKING
GROUP SETUP	2-4 STUDENTS SHARE 1 DEVICE. GAME PLAY USES 5-6 SPLATS.

GAME SUMMARY

Two to four students, each holding a button of Splat one, have to work as a team to pick up and rescue the Splats scattered around the room. Players to pick up each 'Lost Spat' one at a time, holding each very carefully, until they make it to the finish Splat!

GAME RULES: EACH PLAYER, GRAB AND HOLD A BUTTON OF SPLAT ONE TO BEGIN THE GAME. TOGETHER, PICK UP SPLATS 2-4 BY PRESSING AND HOLDING THEIR BUTTONS. WITHOUT DROPPING ANY BUTTONS, REACH THE FINISH SPLAT

NOTES

CODE KEY - PART ONE

HOW IT WORKS

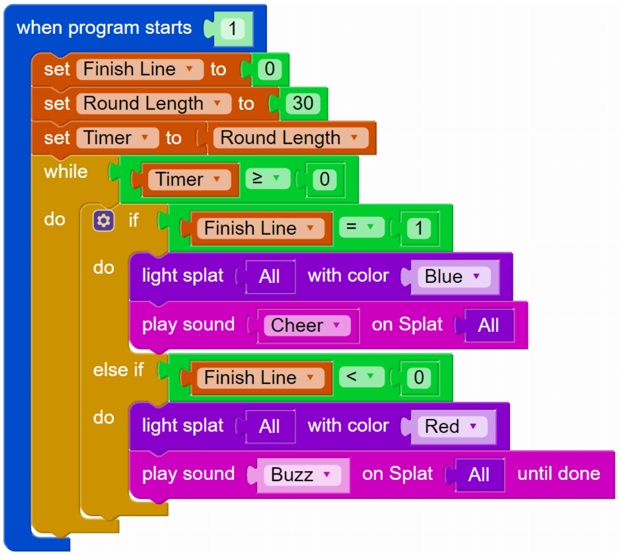
This program features a **while/do** conditional block, as well as an audible timer using the **Timer** variable, used later inside of a **say** block. When the program starts, the game ending variable **Finish Line** is set to zero to start the game, the **Round Length** is set to some number of seconds, and the **Timer** variable is set to match it.

Once the variables are set up, the **while/do** block is used to monitor that the **Timer** variable hasn't reached zero, ending the game if it does.

While the game is running, the **Finish Line** variable is tracked. If the finish Splat is pressed, and the **Finish Line** variable is set to one, the game will end with a cheer! If any of the Splat buttons are released, and **Finish Line** becomes a negative number, the game ends with a buzz.

Continued on next page.

CODE IMAGE - PART ONE



```
when program starts 1
  set Finish Line to 0
  set Round Length to 30
  set Timer to Round Length
  while Timer ≥ 0
  do  if Finish Line = 1
      do  light splat All with color Blue
          play sound Cheer on Splat All
      else if Finish Line < 0
      do  light splat All with color Red
          play sound Buzz on Splat All until done
```

CODE KEY - PART TWO

HOW IT WORKS

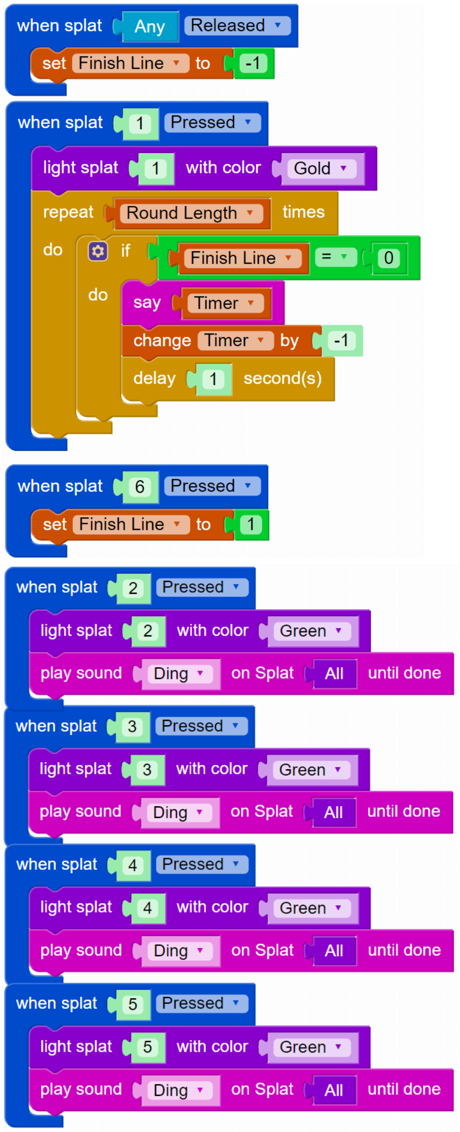
Connecting the code to the game rules: the **when splat released** block is triggered if any Splat is dropped, setting **Finish Line** to a negative number and triggering the buzz sound from part one while stopping the timer.

The round timer begins counting when Splat 1 is pressed. While the game hasn't finished, the **repeat** block announces the time remaining by using the **say** block. Each repeat takes a second off of the **Timer** variable, and continues counting down for however long the **Round Length** variable was set to in the **when program starts** block.

Splats 2 through 5 only need to show the players that they have been picked up, by lighting up and making a sound.

Splat 6 acts as the finish line. When pressed, it sets the **Finish Line** variable to one, ending the round with a win!

CODE IMAGE - PART TWO



```
when splat Any Released
    set Finish Line to -1

when splat 1 Pressed
    light splat 1 with color Gold
    repeat Round Length times
    do  if Finish Line = 0
        do  say Timer
            change Timer by -1
            delay 1 second(s)

when splat 6 Pressed
    set Finish Line to 1

when splat 2 Pressed
    light splat 2 with color Green
    play sound Ding on Splat All until done

when splat 3 Pressed
    light splat 3 with color Green
    play sound Ding on Splat All until done

when splat 4 Pressed
    light splat 4 with color Green
    play sound Ding on Splat All until done

when splat 5 Pressed
    light splat 5 with color Green
    play sound Ding on Splat All until done
```

ADVENTURE SPLATS

ACTIVITY

ACTIVITY OUTLINE

1. INTRODUCE EXERCISE

Explain the game rules and demonstrate how to play the game. Lead groups in identifying the objectives of their program and planning first steps. Have students share their sub-tasks and objective lists with the class

2. GUIDED WORK TIME

Introduce the essential blocks and tie them directly to the game rules. Give groups time to brainstorm different ways to track game elements like round length, when a Splat is picked up or dropped, and when the game is over.

3. GROUP WORK TIME

Support groups as they build their programs. Ensure group members are sharing note taking, testing, and coding roles.

4. STUDENT SHOWCASE!

Let the games begin! Have each group connect to Splats one at a time and complete the adventure!

GOING FURTHER

EXTENSIONS

Students can add additional challenges to this adventure, such as time restrictions and obstacles between each Splat.

SUPPORT

Have students draw a diagram of how the game will be played, identifying the role each Splat will play in the game.

CSTA STANDARDS: ALGORITHMS AND PROGRAMMING

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - 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-09 Variables	Create programs that use variables to store and modify data. (P5.2)
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-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-16 Development	Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. (P2.2)
1B-AP-17 Development	Describe choices made during program development using code comments, presentations, and demonstrations. (P.7.2)

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - 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-12 Control	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. (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-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. (P7.2)

TEAM SPLAT TAG

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES K-5
UNRULINESS	RUNNING, WALKING
GROUP SETUP	2 GROUPS OF ANY SIZE. GAME PLAY USES 3-6 SPLATS, 1 DEVICE

GAME SUMMARY

The starting game rules are an old fashioned team tag; one team tagging out the other, with Splatified safe bases. Add YOUR favorite tag variant to take this game to the next level!

GAME RULES: CLASSIC TEAM TAG WITH RANDOMLY CHANGING TEAM BASES. BASES ARE ONLY SAFE WHILE LIT WITH YOUR TEAM'S COLOR, NOT ALL BASES ARE LIT

NOTES

CODE KEY

HOW IT WORKS

This game runs entirely in one ‘**while/true**’ loop, with a delay timer added. A **while/do** block will continue to run, as long as its condition is met. When that condition is the **true** block, the condition is always met, and the loop will repeat as long as the program is running.

As the loop begins, it lights some number of Splats green and some number purple.

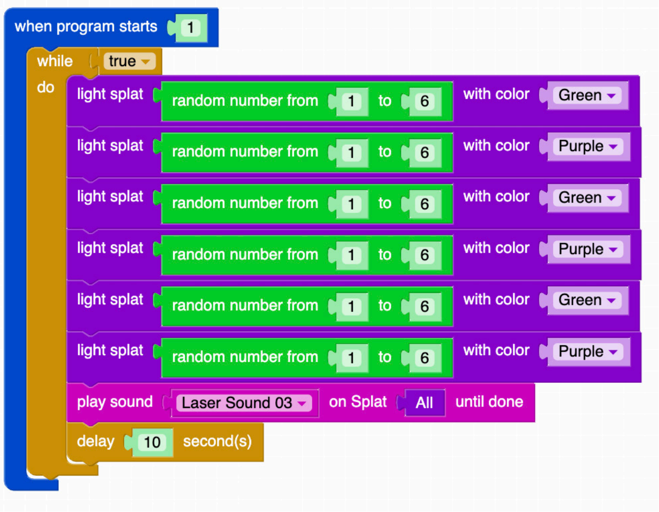
The colors green and purple alternate lighting up, because it more fairly distributes the colors across the Splats.

With **random number** blocks, not all of the Splats will be lit up at the same time, adding to the complexity of the game play.

The **random number** Splat from one to six, still functions well even with less than six Splats connected! If the program lights a non-existent Splat number, the game continues normally.

Try adjusting the round delay, or adding a third team color in the mix.

CODE IMAGE



```
when program starts 1
  while true
  do light splat random number from 1 to 6 with color Green
     light splat random number from 1 to 6 with color Purple
     light splat random number from 1 to 6 with color Green
     light splat random number from 1 to 6 with color Purple
     light splat random number from 1 to 6 with color Green
     light splat random number from 1 to 6 with color Purple
     play sound Laser Sound 03 on Splat All until done
     delay 10 second(s)
```

TEAM SPLAT TAG

ACTIVITY

ACTIVITY OUTLINE

1. INTRODUCE EXERCISE

Introduce the activity and assign teams. Explain the game rules and demonstrate how to play the game. Lead a class discussion about program objectives and potential ways to code this program.

2. WORK TIME

Highlight how the random number blocks assign safe zones. Support groups as they build their own code, encouraging groups to change the colors, add sounds, and change up the rules -- while documenting and explaining each chosen modification. Give groups time to test and debug their code.

3. GAME PLAY

Lead the full class through multiple games of tag-- playing as many teams' programs as possible.

4. STUDENT SHOWCASE!

Have groups present their modified games and talk through the connection to new game rules.

GOING FURTHER

EXTENSIONS

Encourage students to find different uses for this code by creating completely new game rules. If you would like to play with even more bases, build this code on two devices to connect 12 Splats total!

SUPPORT

This game turns six splats into randomly assigned safe zones. Splats can be placed on the floor, on chairs, or on tabletops.

CSTA STANDARDS: ALGORITHMS AND PROGRAMMING

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - GRADES K-2

1A-AP-10 Control	Develop programs with sequences and simple loops, to express ideas or address a problem. (P5.2)
1A-AP-11 Modularity	Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. (P3.2)
1A-AP-12 Development	Develop plans that describe a program's sequence of events, goals, and expected outcomes. (P5.1, 7.2)
1A-AP-14 Development	Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops. (P6.2)

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - 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-15 Development	Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. (P.6.1, 6.2)

THREE SPLATSKETEERS

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES 3-8
UNRULINESS	STANDING, STOMPING
GROUP SETUP	3 STUDENTS SHARE 1 DEVICE. GAME PLAY USES 3 SPLATS

GAME SUMMARY

Defend yourselves! Three players stand back-to-back, each with a Splat out in front of them. Players take turns jumping onto Red Splats and back to formation. Each successful press is one point, and each error press is negative five points. Default round goes to twenty points!

GAME RULES: IF YOUR SPLAT LIGHTS UP, **AND ONLY IF YOUR SPLAT LIGHTS UP**, JUMP ON IT AND QUICKLY RETURN TO FORMATION

NOTES

CODE KEY - PART ONE

HOW IT WORKS

This program features **if/do/else** blocks, with each Splat lighting the next target, and scoring. On the next page, the **when program starts** block gets the game going by lighting up one random Splat red.

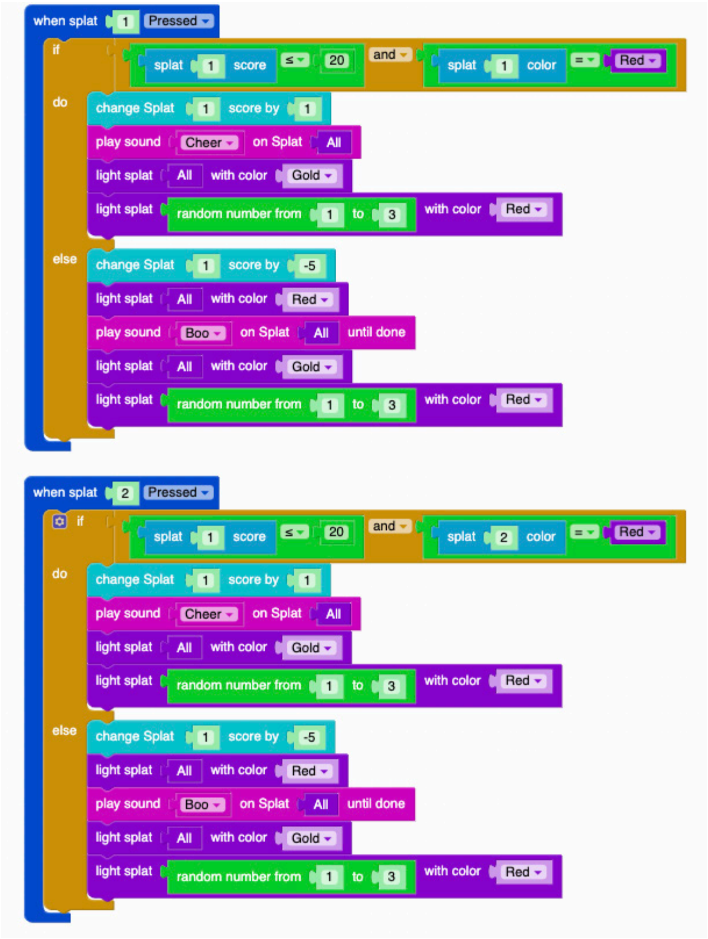
When pressed, each Splat will look at its first condition: that it hasn't scored enough points to win, AND that it is lit up red and ready to be pressed.

When pressed successfully, each Splat triggers a random Splat to light up red again continue the game.

If it isn't the correct time for a Splat to be pressed, the ELSE condition triggers, making it light up yellow and subtracting the score.

Continued on next page.

CODE IMAGE - PART ONE



```
when splat 1 Pressed
  if splat 1 score ≤ 20 and splat 1 color = Red
  do
    change Splat 1 score by 1
    play sound Cheer on Splat All
    light splat All with color Gold
    light splat random number from 1 to 3 with color Red
  else
    change Splat 1 score by -5
    light splat All with color Red
    play sound Boo on Splat All until done
    light splat All with color Gold
    light splat random number from 1 to 3 with color Red

when splat 2 Pressed
  if splat 1 score ≤ 20 and splat 2 color = Red
  do
    change Splat 1 score by 1
    play sound Cheer on Splat All
    light splat All with color Gold
    light splat random number from 1 to 3 with color Red
  else
    change Splat 1 score by -5
    light splat All with color Red
    play sound Boo on Splat All until done
    light splat All with color Gold
    light splat random number from 1 to 3 with color Red
```

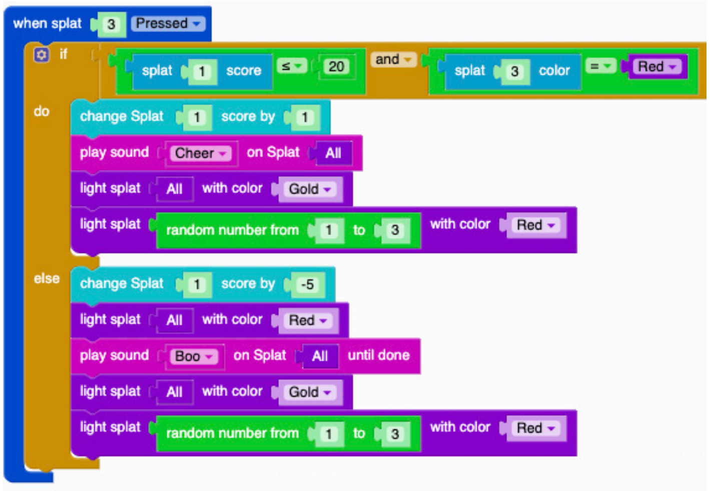
CODE KEY - PART TWO

HOW IT WORKS

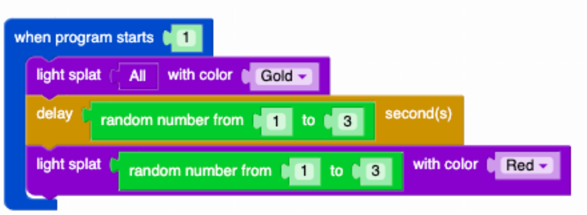
With the same code running for all three Splats, the game will continue no matter which of the three is pressed next.

In this code, the **when splat released** block controls the game winning score level, and will trigger when the team gets enough points.

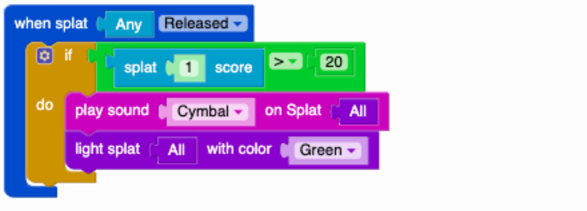
CODE IMAGE - PART TWO



```
when splat 3 Pressed
  if splat 1 score ≤ 20 and splat 3 color = Red
  do change Splat 1 score by 1
     play sound Cheer on Splat All
     light splat All with color Gold
     light splat random number from 1 to 3 with color Red
  else change Splat 1 score by -5
     light splat All with color Red
     play sound Boo on Splat All until done
     light splat All with color Gold
     light splat random number from 1 to 3 with color Red
```



```
when program starts 1
  light splat All with color Gold
  delay random number from 1 to 3 second(s)
  light splat random number from 1 to 3 with color Red
```



```
when splat Any Released
  if splat 1 score > 20
  do play sound Cymbal on Splat All
     light splat All with color Green
```

ACTIVITY OUTLINE

1. INTRODUCE EXERCISE

Introduce the activity. Explain the game rules and demonstrate how to play the game. Compare and contrast the game rules to Whack-a-Mole, a favorite Unruly Example. How do players gain points? Lose them? How do you win?

2. GUIDED WORK TIME

Introduce the essential blocks and tie them directly to the game rules. Show students how to duplicate blocks of code by pressing and holding the outer most block. Talk through the difference between the **when splat pressed** and **when splat released** code. Give students the option to start their program using the 'whack-a-mole' code, highlighting main differences between programs.

3. GROUP WORK TIME

Support groups as they work to build their programs. Ensure students are rotating through a variety of roles, documenting, testing, and coding.

4. STUDENT SHOWCASE!

Time to play Three Splatsketeers! Have groups share their programs and explain why they made the changes the choices they did.

GOING FURTHER

EXTENSIONS

Have students change the timing, scoring, and sounds.

SUPPORT

Go through the conditional statements as a full class. Have students outline the needed code before programming.

CSTA STANDARDS: ALGORITHMS AND PROGRAMMING

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - 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-12 Modularity	Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.(P5.3)
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-16 Development	Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. (P2.2)
1B-AP-17 Development	Describe choices made during program development using code comments, presentations, and demonstrations. (P.7.2)

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - GRADES 6-8

2-AP-10 Algorithms	Use flowcharts and/or pseudocode to address complex problems as algorithms. (P4.4, 4.1)
2-AP-13 Modularity	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. (P3.2)
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. (P7.2)

SPLAT MINING

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES 6-8
UNRULINESS	SITTING OR STANDING
GROUP SETUP	2-4 STUDENTS SHARE 1 DEVICE. GAME PLAY USES 1 SPLAT.

GAME SUMMARY

Very carefully clean the dirt off of your precious Splatjewel.

The program starts with the center of the Splat lit cyan and the outer lit gold. Press carefully to remove pieces until you only have your clean, blue jewel. The largest jewel, the most cyan LEDs in one clump, wins!

GAME RULES: TAP TO CAREFULLY REMOVE DIRT FROM YOUR SPLAT JEWEL

NOTES

CODE KEY - PART ONE

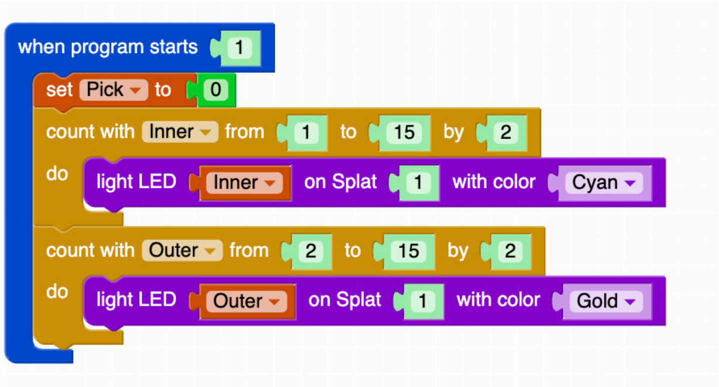
HOW IT WORKS

This program uses the ‘inside, outside’ numbering approach to light the inner lights one color, and the outer lights another. For each Splat, the inner lights are ODD numbers, and the outer ones are EVEN. By using the **count with / by** block and two variables, you can count the even and odd numbers separately.

In the **when program starts** block; the variables **Inner**, and **Outer**, represent the odd and even sets of numbers. Starting at 1, and counting by 2, you will go through all of the ODD numbered lights (1, 3, 5, 7, ... ect) while starting from 2, will get you the even ones (2, 4, 6, 8, ... ect). The **Pick** variable will choose the numbered LED that gets turned off or ‘picked’ away.

Continued on next page.

CODE IMAGE - PART ONE



```
when program starts 1
    set Pick to 0
    count with Inner from 1 to 15 by 2
    do  light LED Inner on Splat 1 with color Cyan
    count with Outer from 2 to 15 by 2
    do  light LED Outer on Splat 1 with color Gold
```

CODE KEY - PART TWO

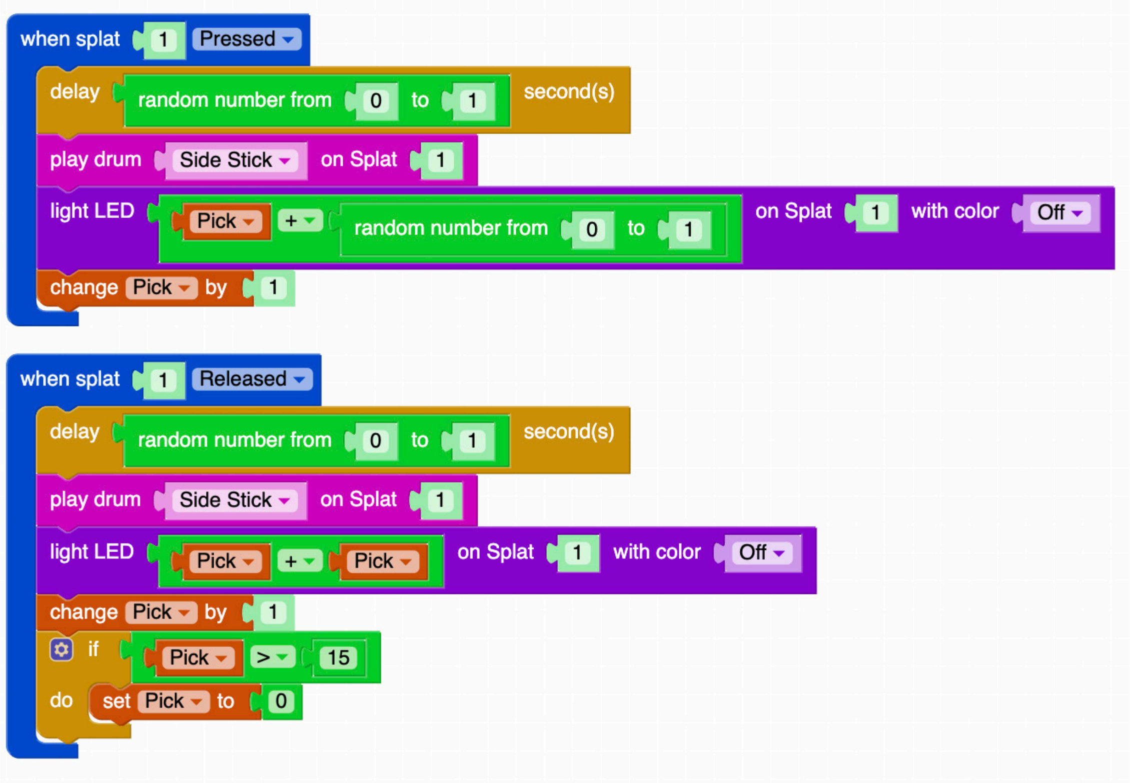
HOW IT WORKS

First, the **when splat pressed** starts a short **random number delay**, after which a sound plays and a random LED is turned off. The **Pick** variable is then increased by one, to continue the game by moving around the circle of LED lights.

The **when released** block turns off a second LED for each Splat release. This second LED is chosen by the number **Pick** + **Pick** to guarantee that this is one of the outer (even numbered) LEDs. This is done to make sure the game is more likely to end with a center 'Gem' of lights remaining.

The final **if/do** block keeps the **Pick** variable inside the range of LED light numbers, 1-14.

CODE IMAGE - PART TWO



```
when splat 1 Pressed
    delay random number from 0 to 1 second(s)
    play drum Side Stick on Splat 1
    light LED Pick + random number from 0 to 1 on Splat 1 with color Off
    change Pick by 1

when splat 1 Released
    delay random number from 0 to 1 second(s)
    play drum Side Stick on Splat 1
    light LED Pick + Pick on Splat 1 with color Off
    change Pick by 1
    if Pick > 15
    do  set Pick to 0
```

ACTIVITY OUTLINE

1. INTRODUCE EXERCISE

Introduce the activity. Explain the game rules and demonstrate how the activity works. Work together as a class to list program objectives, and draft a simple flowchart of key actions.

2. GUIDED WORK TIME

Introduce the essential blocks and tie them directly to the game rules. Highlight the **count by** blocks and explain how they work to remove cyan and gold LEDs. Have students compare the use of the **when splat pressed** and **when splat released** blocks.

3. GROUP WORK TIME

Support groups as they work to build their programs, and through the testing and debugging process. Encourage groups to collaborate across the class to work through challenges. Ensure groups are documenting their development process and adding comments as necessary.

4. STUDENT SHOWCASE!

Give groups time to play their games. Have groups present their largest jewel finds!

GOING FURTHER

EXTENSIONS

This program is written for one Splat, but encourage students to find ways to incorporate additional Splats into the game play.

SUPPORT

Try building the code within **when program starts** block as a full class. Have students outline the needed code on paper before they begin programming.

CSTA STANDARDS: ALGORITHMS AND PROGRAMMING

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - GRADES 6-8

2-AP-10 Algorithms	Use flowcharts and/or pseudocode to address complex problems as algorithms. (P4.4, 4.1)
2-AP-12 Control	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. (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-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. (P7.2)

SPLATICHUTE

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES 6-8
UNRULINESS	RUNNING, KNEELING
GROUP SETUP	GAME PLAY WITH 6+ STUDENTS AND 6 SPLATS ON ONE DEVICE

GAME SUMMARY

The game begins 1-3 students per outer Splat, just like everyone circling an imaginary parachute. All players quickly press their Splats to ‘inflate’ the parachute (center Splat). When the center Splat sounds, ALL players run inside the circle and hold their Splats down behind them as long as they can, while some Students run to press the center Splat to deflate the parachute. When the Splats completely fade, the crowd will cheer, and everyone can leave the circle and repeat!

GAME RULES: INFLATE THE SPLATICHUTE BY PRESSING THE SPLATS IN THE OUTER CIRCLE. RUN INSIDE WHEN FULL AND WAIT / PRESS CENTER SPLAT TO DEFLATE!

NOTES

CODE KEY - PART ONE

HOW IT WORKS

This program features a ‘**while/true**’ loop. Any code in the do section of a **while/do** block will be repeated as long as its condition is met. With ‘**while/true**’, these blocks will run repeatedly, because the condition is always **true**. This allows for constant effects or conditions, such as tracking the **Center** variable featured here. When the program starts, it announces 3...2...1... and sets up variable tracking. The **while/true** block starts repeating, and the **if/else if/else if/do** will be triggered if any of its conditions are met during the program.

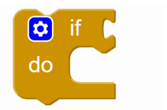
The first **if/do** condition lights the center Splat slowly, while the **Center** score is below the high number required. The next conditional **else/if**, looks for the **Center** score being higher than the goal, and buzzes to tell the players to move into the circle. The final **else/if** makes sure that while deflating the center Splat, the score doesn’t go below zero, and gives everyone a cheer when they complete the cycle.

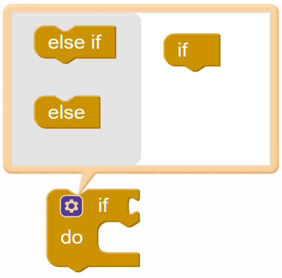
The **when splat pressed** block calls the function **Light Outer Splat**, which does two things: lights the outer Splats (Splats 1-5) random colors, and tallies up the score for everyone to see how many times they pressed the buttons this round. After calling the function, it also adds one to the **Center** variable, progressing towards the ‘inflated’ state.

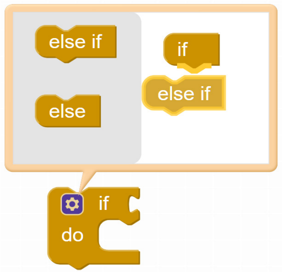
When Splat 6 is pressed, during the deflating phase, it decreases the **Center** variable and provides lights as feedback that it is doing its job.

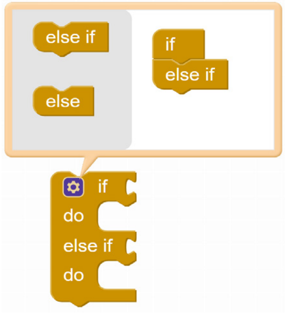
HOW TO MODIFY IF/DO BLOCKS:

Tapping the gear icon opens a small window to modify **if/do** blocks, by adding **else**, and **else/if** blocks. Drag as many **else**, and **else/if** blocks as you need from the left to the right side, and tap the gear icon again close the tray when you are done!



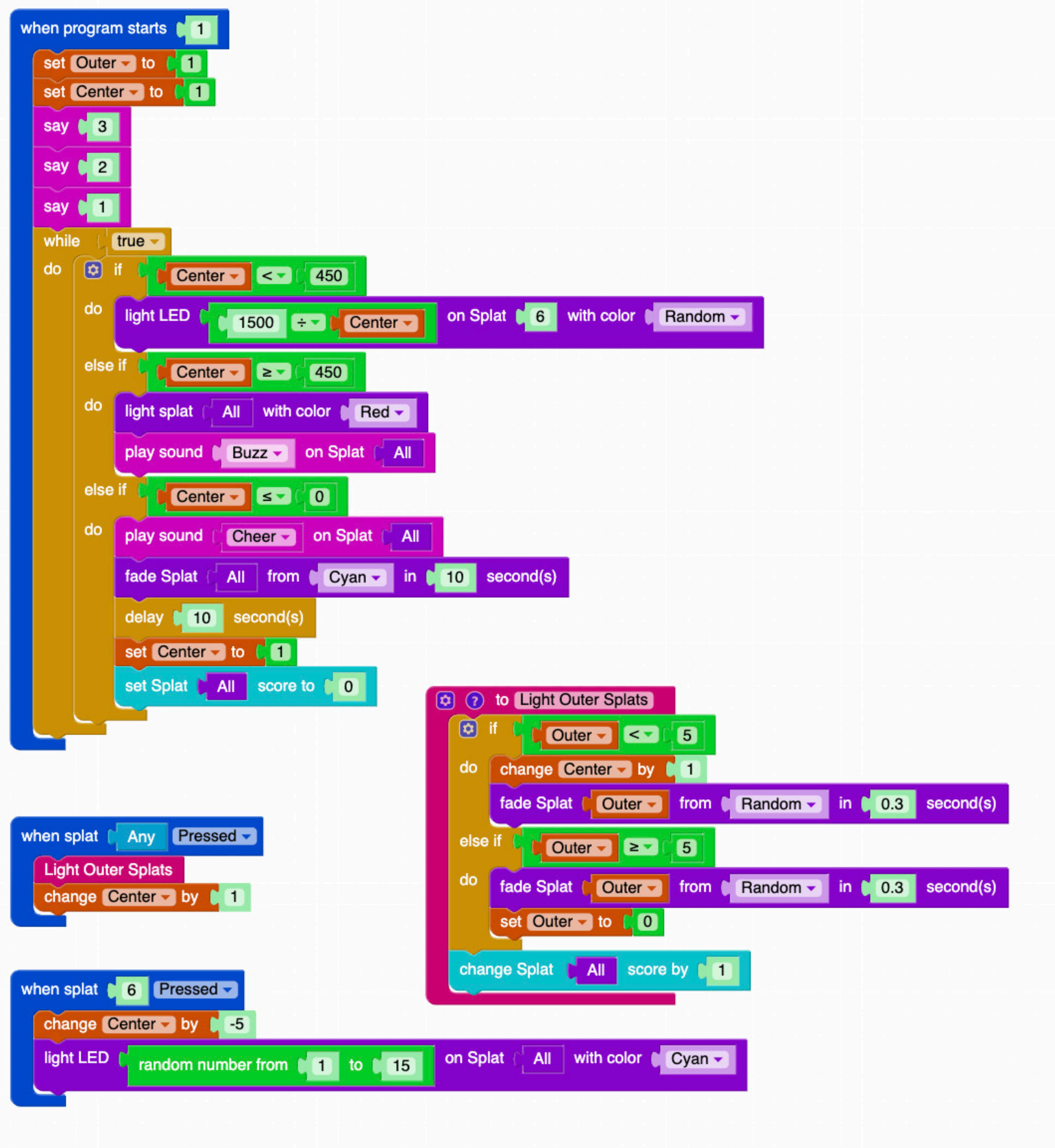






CODE KEY - PART TWO

CODE IMAGE



```
when program starts 1
    set Outer to 1
    set Center to 1
    say 3
    say 2
    say 1
    while true
    do
        if Center < 450
        do
            light LED 1500 ÷ Center on Splat 6 with color Random
        else if Center ≥ 450
        do
            light splat All with color Red
            play sound Buzz on Splat All
        else if Center ≤ 0
        do
            play sound Cheer on Splat All
            fade Splat All from Cyan in 10 second(s)
            delay 10 second(s)
            set Center to 1
            set Splat All score to 0

when splat Any Pressed
    Light Outer Splats
    change Center by 1

when splat 6 Pressed
    change Center by -5
    light LED random number from 1 to 15 on Splat All with color Cyan

to Light Outer Splats
    if Outer < 5
    do
        change Center by 1
        fade Splat Outer from Random in 0.3 second(s)
    else if Outer ≥ 5
    do
        fade Splat Outer from Random in 0.3 second(s)
        set Outer to 0
    change Splat All score by 1
```

ACTIVITY OUTLINE

1. INTRODUCE EXERCISE

Introduce the activity. Explain the game rules and demonstrate how to play the game. Make a flow chart outlining the program and list key objectives

2. GUIDED WORK TIME

Introduce the essential blocks and tie them directly to the game rules. Highlight the variables used in this program and draw a diagram showing the role of each Splat. Have students talk about the ways variables are modified in this program. Discuss how a function could be used to make the program simpler.

3. GROUP WORK TIME

Support groups as they work to build their programs, and in breaking the program down into chunks if necessary.

4. STUDENT SHOWCASE!

Gather the class and give groups an opportunity to present and play their games!

GOING FURTHER

EXTENSIONS

Have students modify the timing, game cues (the **say** blocks), sounds, and colors. Encourage students to come up with new rules for this code, creating a completely different game!

SUPPORT

Have students outline this program on paper before coding, specifically variable modification pieces. Additionally, this program can be built as a whole class!

CSTA STANDARDS: ALGORITHMS AND PROGRAMMING

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - 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-12 Control	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. (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. (P7.2)

RAINBOW KEYBOARD

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES 6-8
UNRULINESS	STANDING, WALKING
GROUP SETUP	2-4 STUDENTS PER GROUP. GAME PLAY USES 6 SPLATS.

GAME SUMMARY

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.

GAME RULES: BE UNRULY. PLAY MIDI KEYBOARD IN STYLE.

NOTES

CODE KEY

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.

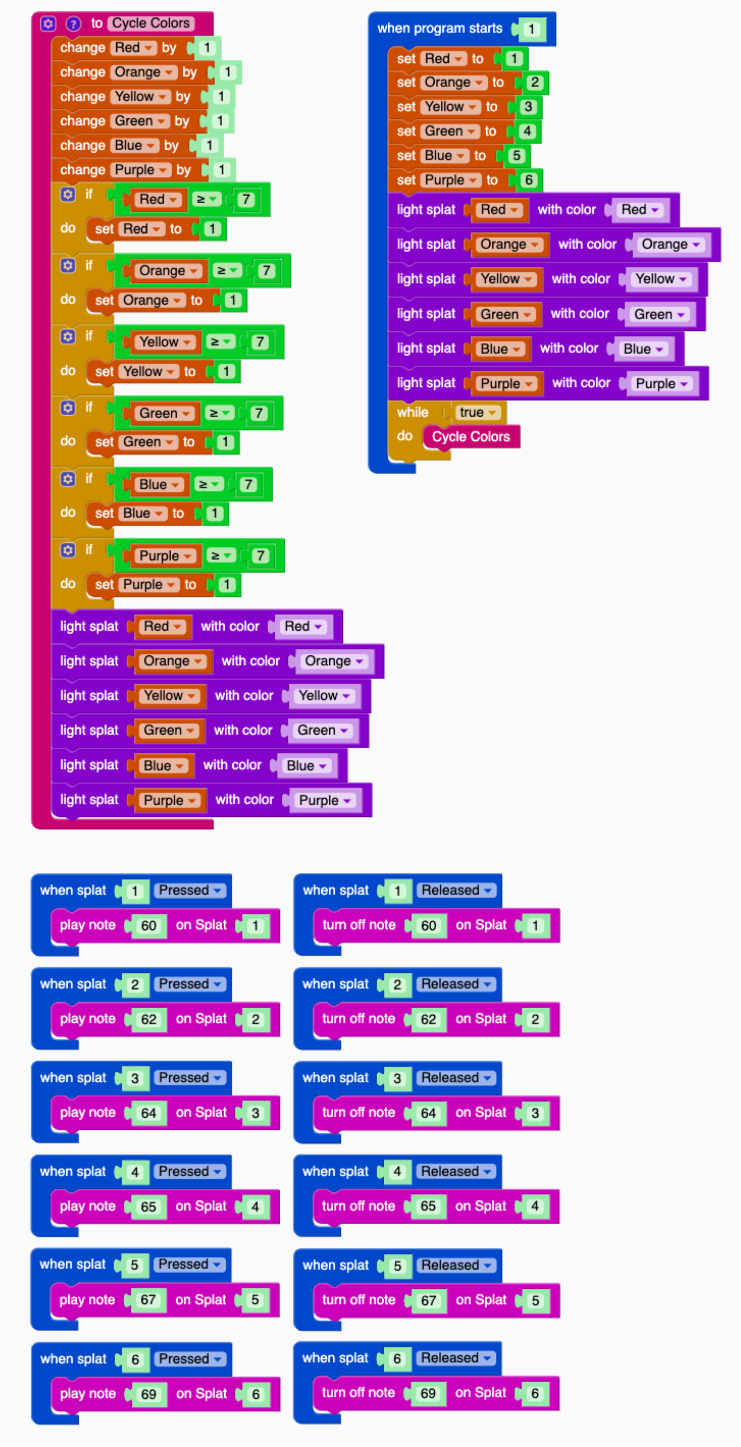
As the program runs, the different colors are given an order, one through six. The **Cycle Colors** function, when called, moves each color up to the next Splat in line by adding one to its value.

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.

CODE IMAGE



```
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 CONVERSION 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)’.

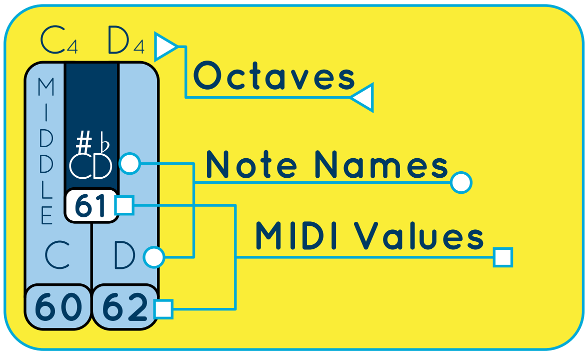
Because there are 12 notes from C to B in each octave, you can go up or down an octave by adding or subtracting 12 from that note’s MIDI value.

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.

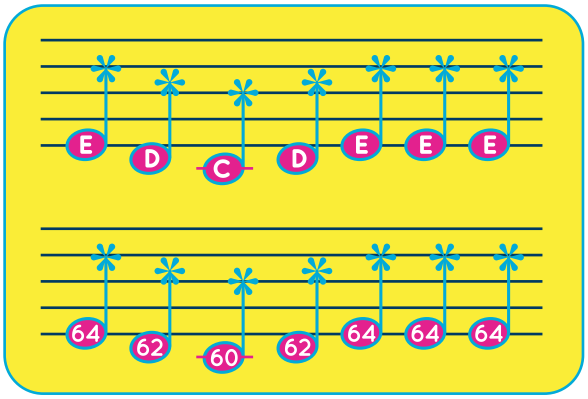
These MIDI conversions don’t have to be limited to single notes either! You can build and experiment with basic three or five note chords played at the same time.

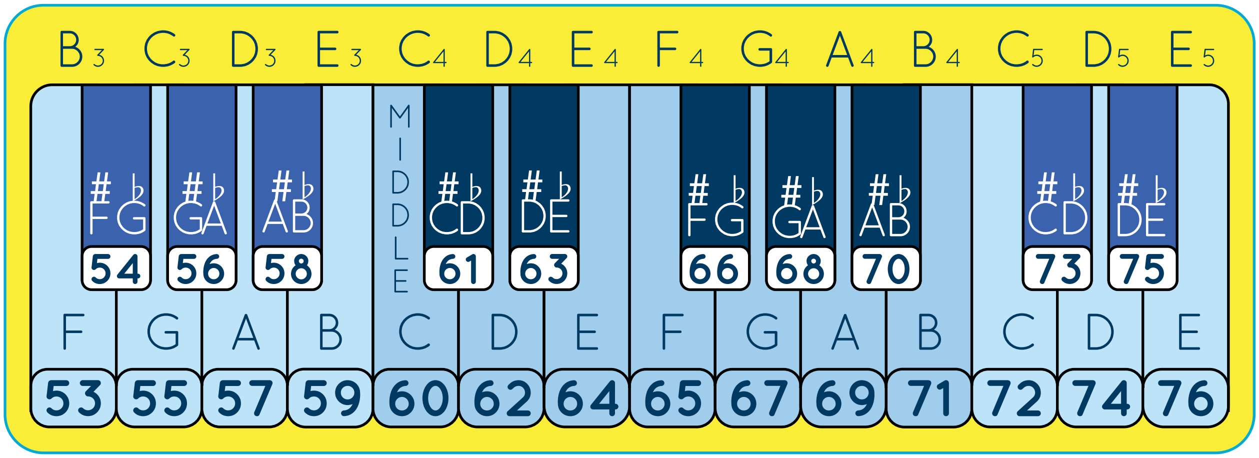
Some examples are C (60, 64, 67), E (64, 68, 71), and G# (68, 72, 75) or even a C major pentatonic (60, 62, 64, 67, 69).

KEY



EXAMPLE: MARY HAD A LITTLE LAMB





RAINBOW KEYBOARD

ACTIVITY 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

EXTENSIONS

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 AND PROGRAMMING

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - 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. (P7.2)

FITNESS CIRCUIT

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES 3-8
UNRULINESS	RUNNING, JUMPING, PUSH-UPS
GROUP SETUP	2+ STUDENTS PER DEVICE. GAME PLAY USES 2 SPLATS.

GAME SUMMARY

It’s time for a Splat Fitness Circuit! Start with sprints at a running distance, and then move the Splats together for push ups and jogging in place!

GAME RULES: RUN SPRINTS, DO PUSH-UPS, JOG IN PLACE!

NOTES

CODE KEY

HOW IT WORKS

This code uses an **if/do** with an **and/or** connector block to track the exercises.

The **when program starts** block sounds a whistle on start, and later code sounds one again when ten of each activity has been completed.

Each Splat tracks its score differently, to enable tracking of different types of exercises.

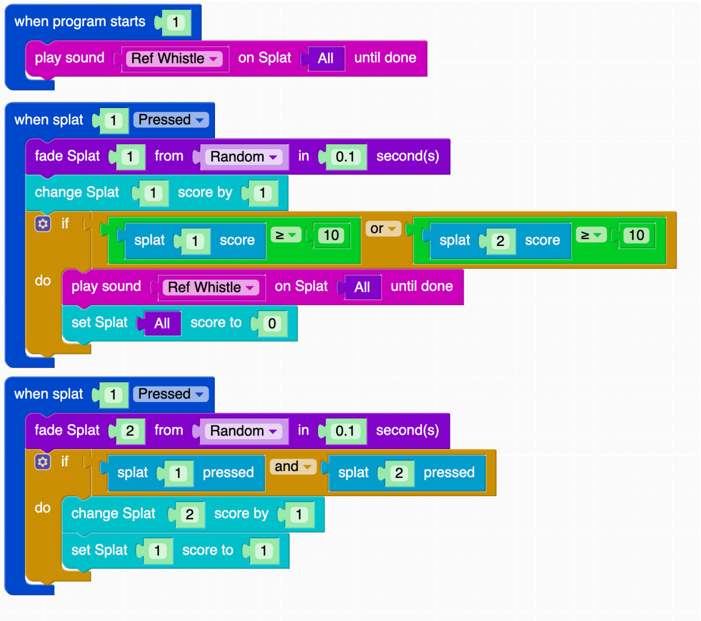
Splat one adds to its score when it is pressed alone, resetting when either Splat's score is ten. This is a great way to track sprints or laps that press one Splat, than another.

Splat two adds one to its score only when BOTH Splats are pressed at the same time. This is how things like push-ups, and jumping jacks could be tracked.

With both scoring mechanisms running at the same time, pressing the Splats one after the other will sound a whistle every ten reps, AND when both Splats are pressed at the same time, ten times in a row.

Simple code with endless uses!

CODE IMAGE



```
when program starts 1
  play sound Ref Whistle on Splat All until done

when splat 1 Pressed
  fade Splat 1 from Random in 0.1 second(s)
  change Splat 1 score by 1
  if  splat 1 score ≥ 10  or  splat 2 score ≥ 10
  do  play sound Ref Whistle on Splat All until done
      set Splat All score to 0

when splat 1 Pressed
  fade Splat 2 from Random in 0.1 second(s)
  if  splat 1 pressed  and  splat 2 pressed
  do  change Splat 2 score by 1
      set Splat 1 score to 1
```

FITNESS CIRCUIT

ACTIVITY OUTLINE

1. INTRODUCE EXERCISE

Introduce the activity and demonstrate each exercise in the circuit. Brainstorm as a class what a program would need in order to track all three exercises. List key requirements and breakdown challenges.

2. GUIDED WORK TIME

Introduce the essential blocks. Talk through the code for each **when splat pressed** block. Lead a class discussion about why the two blocks of code are different. Have groups look at the four Splat example, ‘Race in Place Deluxe.’ What are some key similarities and differences between the games?

3. GROUP WORK TIME

Support students as they build their programs. Encourage students to test their programs throughout the building process, and rotate through a variety of roles. If groups use code from ‘Race in Place’ support them in attributing the borrowed code correctly.

4. STUDENT SHOWCASE!

Give groups time to complete their circuits! Have groups present their code and discuss additional exercises to add in.

GOING FURTHER

EXTENSIONS

Have students add additional exercises and Splats!

SUPPORT

Provide students with the and connector and or comparison blocks. Have students write out how the scoring needs to work for each of the exercises before they start to code.

CSTA STANDARDS: ALGORITHMS AND PROGRAMMING

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - 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	Observe intellectual property rights and give appropriate attribution when creating or remixing programs.(P5.2, 7.3)
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)

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - GRADES 6-8

2-AP-12 Control	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. (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-17 Development	Systematically test and refine programs using a range of test cases. (P6.1)