ACTIVITY PACK

WITH

10 LESSON PLANS

LEARN
PLAY

FOR USING

unruly Splats™

CODE
JUMP
RUN

IN YOUR CLASSROOM

HAVE FUN

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

unruly STUDIOS

1

Vers. AP9_6_20

DEAR EDUCATOR

Thank you for supporting Unruly Studios and bringing Splats to your learners. This Activity Pack contains ten Unruly activities for you to use with your students. We are very excited to share them with you, but even more excited to learn with you. Whenever possible, please share your creations on social media and tag them with #UnrulyEdu!

If you have any questions, please don't hesitate to email us at education@unruly-studios.com.

Let the Unruliness begin!

Sincerely,
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

VOTING MACHINE

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES K-5
UNRULINESS	WALKING
GROUP SETUP	2-4 STUDENTS SHARE 1 TABLET. GAME PLAY USES 2 SPLATS.

GAME SUMMARY

Program Splats to become a voting machine! Each group chooses their own voting topic and participates in a group vote at the end of this activity. Voting Machine can be used as a low-movement introduction to Splats.

GAME RULES: PRESS THE SPLAT TO ADD A POINT.

NOTES

PLAYING TOGETHER

- ★ Students remain seated in designated spots
- ★ Use hand signals to vote rather than pressing Splats
- ★ Individually play using web-app and ask neighbors/partners to cast votes verbally

REMOTE PLAY

- ★ Splats web-app
- ★ Virtual breakout rooms
- ★ Have a member of each small group share their screen for voting

VOTING MACHINE

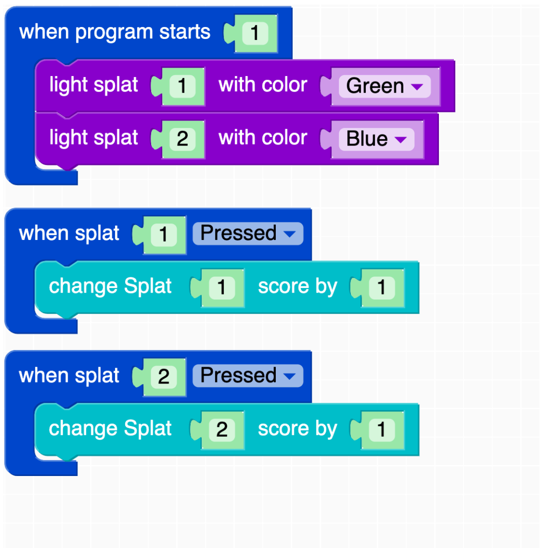
CODE KEY

HOW IT WORKS

This program highlights the difference between the two starting blocks: **when program starts** and **when splat pressed**. It also introduces the concept of scoring.

The different colors for Splats 1 and 2 identify which Splat is which for the vote, so they must light up at the start. The Splats light up with the **when program starts** block. Votes are counted within the **when splat pressed** blocks.

CODE IMAGE



```
when program starts 1
    light splat 1 with color Green
    light splat 2 with color Blue

when splat 1 Pressed
    change Splat 1 score by 1

when splat 2 Pressed
    change Splat 2 score by 1
```

VOTING MACHINE

ACTIVITY OUTLINE

1. INTRODUCE EXERCISE

Introduce the activity and the program students will be building. Play an example voting machine and guide a full-class vote. Model a voting machine using thumbs up or flashcards. Have students discuss the goals of their voting machine program.

2. GUIDED WORK TIME

Have groups select their voting topics. Introduce the essential blocks and give students time to explore how they work. What are the different ways we can program this game?

3. INDEPENDENT WORK TIME

Assist groups in building their programs. Have students customize their program based on their voting topic. Support students in working through bugs. Have groups brainstorm other ways they could build this game. What are other games you could play using this code?

4. EXERCISE RECAP

Review the program as a class and ask groups to share out their programs. Have students describe the choices they made during program development.

5. STUDENT SHOWCASE!

Have the class rotate through every group's voting machine. Have each group announce the winner!

GOING FURTHER

EXTENSIONS

Add sounds, a stopwatch, or a countdown timer. If you have enough splats, have students add additional selections. Additionally, have students label their Splats by recording sounds. This project can also be completed as an independent activity.

SUPPORT

Voting topics can be assigned or restricted to specific categories. Clarify the distinction between **when splat pressed** and **when program starts** – this can be tough!

CSTA STANDARDS: ALGORITHMS AND PROGRAMMING

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

1A-AP-08 Algorithms	Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks. (P4.4)
1A-AP-09 Variables	Model the way programs store and manipulate data by using numbers or other symbols to represent information. (P4.4)
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)
1B-AP-17 Development	Describe choices made during program development using code comments, presentations, and demonstrations. (P.7.2)

FOUR SPLATS

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES K-5
UNRULINESS	WALKING
GROUP SETUP	2-4 STUDENTS SHARE 1 TABLET. GAME PLAY USES 4 SPLATS.

GAME SUMMARY

This game is an Unruly version of the classic game, Four Corners. Students build a program to randomly select which corner is out. Place four Splats in four corners. Press run to start the game. Students have ten seconds to pick a Splat to stand by. If your Splat turns off, you're out! Repeat until there is only one player remaining.

GAME RULES: PROGRAM LIGHTS UP 4 SPLATS & RANDOMLY TURNS OFF 1 AFTER 10 SECONDS.

NOTES

PLAYING TOGETHER

- ★ Smaller groups take turns playing
- ★ Mark the corner/floor with tape to show where students can stand
- ★ Use colored flags, objects, or hand signals to choose corner
- ★ Use the code and play a completely different game! For example, turn the code into a 4-sided die and assign a different exercise for each color (red is push ups, etc.)

REMOTE PLAY

- ★ Splats web-app
- ★ Students place a hand in a corner of their screen
- ★ Similarly, students can use a hand signal or object to indicate their choice, such as putting up 2 fingers for corner #2

FOUR SPLATS

ACTIVITY

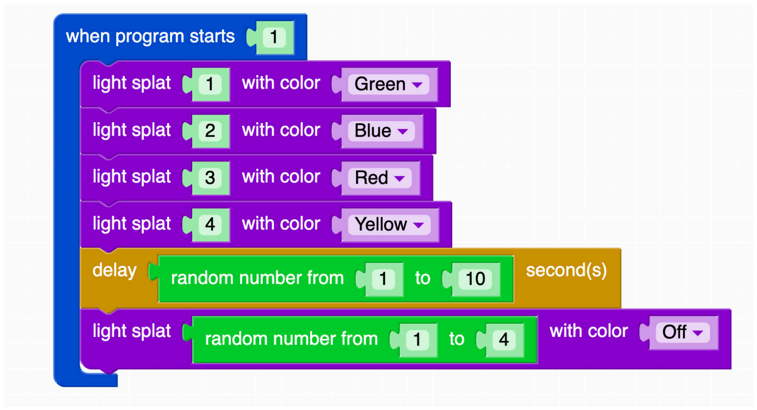
CODE KEY

HOW IT WORKS

This program introduces the use of the **random number** block. In this program, when any Splat is pressed the program begins, all four Splats light up, and a delay begins to count seconds down from 10. After 10 seconds the lights for a random Splat are turned off.

Press Run at the start of each round to restart.

CODE IMAGE



```
when program starts 1
  light splat 1 with color Green
  light splat 2 with color Blue
  light splat 3 with color Red
  light splat 4 with color Yellow
  delay random number from 1 to 10 second(s)
  light splat random number from 1 to 4 with color Off
```

FOUR SPLATS

ACTIVITY OUTLINE

1. INTRODUCE EXERCISE

Introduce the activity and lead students through an example round of Four Splats. Lead a discussion about the game rules and how they could be broken down into precise instructions. Discuss the needed program and what needs to be included.

2. GUIDED WORK TIME

Give time for groups to brainstorm how to build this program. Have groups share their proposed programs, including timing, sequencing, and overall goals. Walk through the essential blocks with the class.

3. INDEPENDENT WORK TIME

Assist groups in building their own Four Splats programs. Ask each group to write down the rules for their game and be prepared to present them to the class. Support groups in debugging their programs and making key decisions.

4. STUDENT SHOWCASE!

Have all groups lead the class in playing their Four Splats programs. Give groups the opportunity to describe the modifications they made, if any. Restart the Splats after each game to quickly connect to a new device. If time allows, hold a finals round with the winner from each game.

GOING FURTHER

EXTENSIONS

Pair this activity with the exercise Red Splat, Green Splat for a more in-depth exploration of the **random number** block. Additionally, challenge students to come up with alternative ways to code and/or play this game. Compare this program to the Unruly, four Splat example program, Four Corners. What are the benefits of using a variable?

SUPPORT

If students are struggling in the brainstorm, consider providing students with the needed blocks; demonstrating how to use **random number**; or hint that the program can be built inside one starting block. For a quieter game, this can be played seated, students just need color cards to pick their "corner".

CSTA STANDARDS: ALGORITHMS AND PROGRAMMING

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - GRADES K-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)
1A-AP-15 Development	Using correct terminology, describe steps taken and choices made during the iterative process of program development. (P7.2)

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-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)

RED SPLAT GREEN SPLAT

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES K-5
UNRULINESS	RUNNING, FULL CLASSROOM
GROUP SETUP	2-4 STUDENTS SHARE 1 TABLET. GAME PLAY USES 1 SPLAT.

GAME SUMMARY

This game uses Splats to play a version of 'Red Light, Green Light'. Place Splats on the floor or tables across the room. Just like the classic playground game, students run towards their Splat when it is green, but must freeze when it turns red. This game can be played independently by each group or as a full class.

GAME RULES: CHANGE THE SPLAT COLOR FROM GREEN TO RED AT RANDOM INTERVALS.

NOTES

PLAYING TOGETHER

★ Smaller groups take turns playing

★ Create "lanes" using tape or yardsticks

★ To control movement, students roll a die to determine their # of steps

★ Students can also use a die to move figures on a board or grid rather than their own bodies

★ Note: If using the preloaded example, DO NOT stomp

REMOTE PLAY

★ Splats web-app

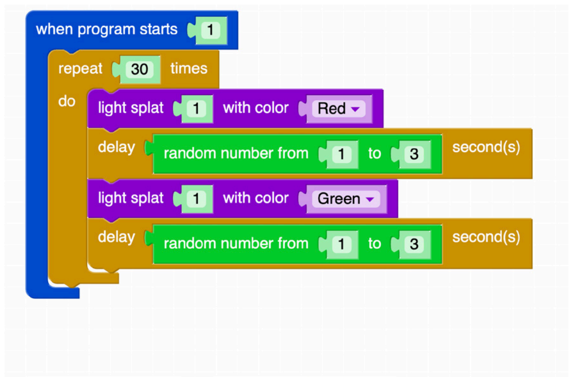
★ Instead of running, students have to sing the Alphabet song (or another familiar song) until they get to the "end"

CODE KEY

HOW IT WORKS

This program introduces nesting and the **repeat** block. The rules for changing the Splat from red to green to red are nested inside the **repeat** block. Within a **repeat** block, blocks are repeated in a loop. This program repeats 30 times, meaning that the color changes from red to green 30 times. The color of the Splat is changed using the **delay** and **random number** blocks. The Splat is lit up red and then turns green for random intervals between 1 and 3 seconds.

CODE IMAGE



```
when program starts 1
  repeat 30 times
  do
    light splat 1 with color Red
    delay random number from 1 to 3 second(s)
    light splat 1 with color Green
    delay random number from 1 to 3 second(s)
```

ACTIVITY OUTLINE

1. INTRODUCE EXERCISE & KEY CONCEPTS

Introduce the rules for the game. Lead a brainstorm of possible ways to build this program and list the rules for the game. Compare algorithms that could be used.

2. WORK TIME

Give time for students to brainstorm how to build their program either independently, in groups, or as a class. Review the concept of nesting. Review how the **random number** block and the **delay** block can be used to control the number of seconds a Splat is lit up green. Support groups in breaking down the game rules and planning their program. Give time for students to compare programs, rework, and debug.

3. GAME PLAY

Give time for students to practice their games in small groups, let groups know they will come together to play their game as a class. Give groups the opportunity to iterate on their program based on the player experience.

4. STUDENT SHOWCASE!

Regroup and review the activity. Play each group's program as a full class, keeping track of each game's winner. Hold a championship round using the example code. Give groups and opportunity to describe their development processes and the choices made.

GOING FURTHER

EXTENSIONS

Add in the Start Stopwatch Block, having it start when the program starts and stop when the Splat is pressed. Groups can add additional rules/features to their game, such as a yellow light for 'walk'. Compare this code to the Unruly example of the same name. What are some key differences in the programming and the game play?

SUPPORT

If students are struggling during work time, provide the essential blocks and have groups write out each step on paper before beginning to code.

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)
1A-AP-15 Development	Using correct terminology, describe steps taken and choices made during the iterative process of program development. (P7.2)

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-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-17 Development	Describe choices made during program development using code comments, presentations, and demonstrations. (P.7.2)

RACE IN PLACE

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES K-5
UNRULINESS	RUNNING IN PLACE
GROUP SETUP	2-4 STUDENTS SHARE 1 TABLET. GAME PLAY USES 2 SPLATS.

GAME SUMMARY

Students create a pedometer and compete to get as many steps as they can. Groups log each player's steps and compete against other groups to get the most total steps.

GAME RULES: GET POINTS BY RUNNING ON TWO SPLATS WHILE A TIMER COUNTS DOWN.

NOTES

PLAYING TOGETHER

★ Smaller groups take turns playing

★ Spread out groups as much as possible

REMOTE PLAY

★ Splats web-app

★ Virtual breakout rooms

★ Instead of running, students must perform a task within the countdown. Ideally, it's a simple task that adds up, like drawing as many stars as possible or doing a "mad minute," e.g., answering basic math facts. Groups can tally up their combined score to compete.

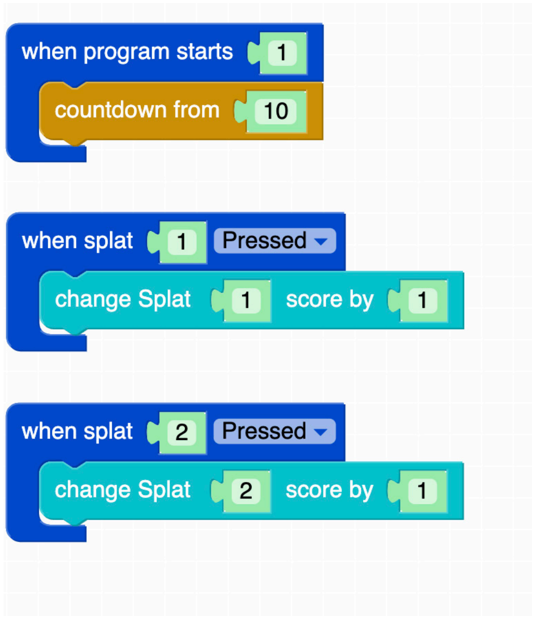
CODE KEY

HOW IT WORKS

This program introduces scoring and demonstrates how to use **delay** and the **stopwatch** blocks. In this program, points are scored each time a Splat is pressed. One press adds one point to the score for that Splat. There are many ways to keep score, this way is the simplest.

The program runs for 10 seconds, and players race to get as many points as they can before the stopwatch ends. It is important to note that there are three **start** blocks in this program.

CODE IMAGE

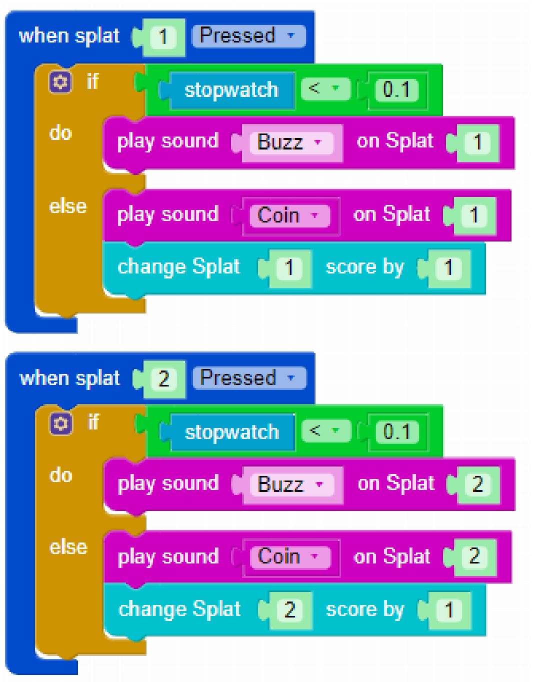


```
when program starts 1
    countdown from 10

when splat 1 Pressed
    change Splat 1 score by 1

when splat 2 Pressed
    change Splat 2 score by 1
```

HOW IT WORKS - PART 2

This program stops allowing points once the timer reaches zero. At the same time, it adds sound feedback; a coin sound for a point gained and a buzzer for the end of the round.

CODE IMAGES - PART 2



```
when splat 1 Pressed
    if stopwatch < 0.1
    do
        play sound Buzz on Splat 1
    else
        play sound Coin on Splat 1
        change Splat 1 score by 1

when splat 2 Pressed
    if stopwatch < 0.1
    do
        play sound Buzz on Splat 2
    else
        play sound Coin on Splat 2
        change Splat 2 score by 1
```

RACE IN PLACE

ACTIVITY OUTLINE

1. CLASS DISCUSSION

What is a pedometer? Support a class discussion about how you would code Splats to be a pedometer. What blocks are needed? What key things does the program need to include?

2. PEDOMETER CREATION -- PART 1

Support groups in building their pedometers. Once complete, have groups share their creations. Ask groups what needs to be added to their code in order to turn their pedometers into a game.

3. GAME CREATION -- PART 2

Give groups time to explore the code for Race in Place, the two Splat example program. Use this code as reference for gamifying the groups' pedometers. What does that program include to add fun? (ex. timing, lights to signal the end of the round, competition!) Support groups in building their pedometer games, giving attribution to the Race in Place code example when needed. Support groups in debugging, testing, and rotating through different group roles.

4. STUDENT SHOWCASE!

Give groups time to present their games, highlighting their game elements, additional blocks, and planning process.

GOING FURTHER

EXTENSIONS

Change the rules for scoring to make new rules for the game. For example, build a program where one Splat adds points and the other subtracts, so two students can play 'tug of war'.

SUPPORT

Note the example program displays the total number of steps for Splat 1 and 2 separately. Students can also count points for Splat 1 and 2 presses under one total for one Splat.

CSTA STANDARDS: ALGORITHMS AND PROGRAMMING

COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - GRADES K-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)
1A-AP-15 Development	Using correct terminology, describe steps taken and choices made during the iterative process of program development. (P7.2)

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-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-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-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)

GAME CREATOR

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES 3-8
UNRULINESS	VARIES
GROUP SETUP	2-4 STUDENTS SHARE 1 TABLET. GAME PLAY USES 4+ SPLATS.

GAME SUMMARY

This is an open-ended activity. Students are presented with Splats code and asked to create their own game. Students will name their game, write the rules, and lead their peers in game play.

GAME RULES: BE UNRULY!

NOTES

PLAYING TOGETHER

- ★ Smaller groups take turns playing
- ★ Spread out as much as possible
- ★ Students can independently create games and have a showcase later on

REMOTE PLAY

- ★ Splats web-app
- ★ Virtual breakout rooms
- ★ Students can independently create games and have a showcase later on

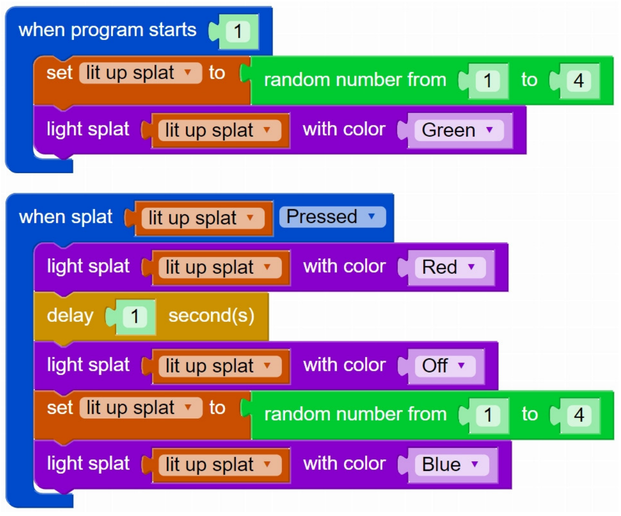
CODE KEY

HOW IT WORKS

This program introduces variables. Players will not be creating a program from scratch, instead they use the provided code to create their own game. Players can change as much or as little as they want, but it must be playable with the number of provided Splats.

The starting program uses a variable to determine which Splat is activated. Variables are a place to store information within a program. The variable changes each loop when the Random Block selects a new number for the variable **lit up splat**.

CODE IMAGE



```
when program starts 1
  set lit up splat to random number from 1 to 4
  light splat lit up splat with color Green

when splat lit up splat Pressed
  light splat lit up splat with color Red
  delay 1 second(s)
  light splat lit up splat with color Off
  set lit up splat to random number from 1 to 4
  light splat lit up splat with color Blue
```

GAME CREATOR

ACTIVITY OUTLINE

1. INTRODUCE EXERCISE & KEY CONCEPTS

Run the code key code for the class. Show the program and have groups identify the key operations.

2. WORK TIME

Project or print out the code for the starting program. Support students as they brainstorm and work together in groups to build their new games. Have groups break down the sequence of events and examine what the program is doing. What could be a goal of this program?

3. GAME CREATION

Students create their game: writing rules, coming up with a title, and making any code modifications. Write out title and rules clearly. Support groups in modifying the program for their specific game. Ensure groups are rotating through key roles such as testing, debugging, documenting, and game design.

4. STUDENT SHOWCASE!

Have each group present their game, and give time for the groups to play each other's games. Support groups in explaining their modifications and their key decisions.

GOING FURTHER

EXTENSIONS

This activity can be done without any provided code—students need to fully design the code and the rules.

SUPPORT

Provide examples or specific guidelines if students are struggling to come up with game ideas.

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-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-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-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-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)

SPLATS GOALIE

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES 3-8
UNRULINESS	JUMPING
GROUP SETUP	2-4 STUDENTS SHARE 1 TABLET. GAME PLAY USES 4 SPLATS.

GAME SUMMARY

Goalie is a game similar to “whack-a-mole”, but instead of whacking moles you block soccer balls! A ‘ball’, or lit up Splat, will appear randomly on one of four Splats—arranged in a line on the floor. When the green light appears, step on the Splat to block the ball! Students will work together to build and debug the included code.

GAME RULES: STEP ON THE SPLAT WHEN IT TURNS GREEN TO WIN A POINT.

NOTES

PLAYING TOGETHER

- ★ Smaller groups take turns playing
- ★ Have a designated ‘goalie’ press the Splats to avoid crowding
- ★ Individually play using the web-app

REMOTE PLAY

- ★ Splats web-app
- ★ Virtual breakout rooms
- ★ Have a designated ‘goalie’ click on the Splats that light up green

SPLATS GOALIE

CODE KEY

HOW IT WORKS

This challenge is an opportunity to work with a more complex program and learn to debug code. The code for this activity has bugs built in! Debugging this code will be required to come up with a functioning program, an example of which is included below. Important concepts to understand are nesting, **if/do** statements, **repeat** loops and **variables**. The program has two sections—one for making the soccer ball appear and one for scoring.

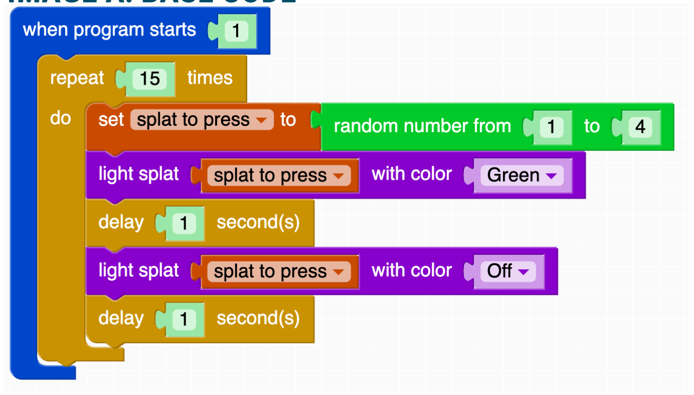
To fix the buggy version of the scoring code, first identify the correct way to use a variable to keep track of when the green Splat is pressed. Debugging is a process of documentation, tracking (working backwards to find the source), and reproduction (re-creating the bug to confirm).

This program lights a random Splat green fifteen times by assigning a random number to the variable **splat to press** in a repeat loop. The corrected code will add a point to the score if the Splat is pressed while green. The **all** block changes the total score on all Splats by one.

Once you have a **working when splat pressed** block for Splat **1**, duplicate it three times by right clicking, or long pressing on the top block. Once copied, change **when splat pressed** to **2**, **3**, and **4** to cover scoring for all four Splats!

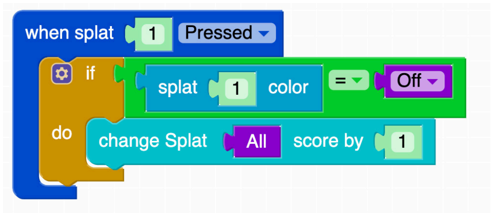
CODE IMAGE

IMAGE A: BASE CODE



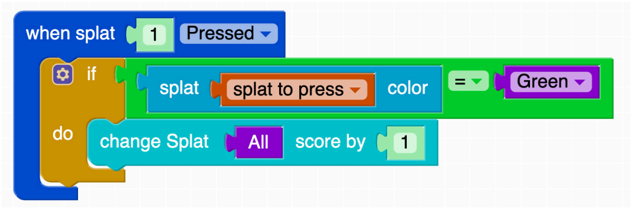
```
when program starts 1
  repeat 15 times
  do  set splat to press to random number from 1 to 4
      light splat splat to press with color Green
      delay 1 second(s)
      light splat splat to press with color Off
      delay 1 second(s)
```

IMAGE B: SCORING WITH BUGS



```
when splat 1 Pressed
  if  splat 1 color = Off
  do  change Splat All score by 1
```

IMAGE C: SCORING WITHOUT BUGS



```
when splat 1 Pressed
  if  splat splat to press color = Green
  do  change Splat All score by 1
```

SPLATS GOALIE

ACTIVITY OUTLINE

1. INTRODUCE CHALLENGE & KEY CONCEPTS

Introduce the activity. Explain that students will work to debug a program. Introduce the essential blocks and key concepts, go over the first section of code as a class! (Code Image A)

2. WORK TIME (PT. 1)

Give time for students to brainstorm how to use the **random number** block in the program to generate their soccer 'ball'. Ask groups to build this first section. Project or print out the code for the starting program. Support students as they work together to finish this section. Have groups identify the key goals of this code and the sequence of events. (Code Image B)

3. WORK TIME (PT. 2)

Provide groups with the buggy scoring code, included in the Code Key. Walk students through the steps of debugging: documentation, tracking (working backwards to find the source), and reproduction (re-creating the bug to confirm). Have groups work together to debug the code. Review the debugged code as a class and have groups test out their debugged code. Support groups in modifying their final game code to add additional elements. (Code Image C)

4. STUDENT SHOWCASE!

Give groups time to present their games. Have groups explain the key choices they needed to make and their debugging process.

GOING FURTHER

EXTENSIONS

Encourage groups to modify their game speed, add a second ball to their game, create a variable for the number of Splats, and explore the **if/do** block.

SUPPORT

Part 1: Provide clues for block placement. Walk groups through the creation and use of variables.
Part 2: Only locate bugs, ask groups to find the bugs, not fix them. Or, only fix bugs and give groups bug location and ask them to come up with solutions.

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-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-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)

RELAY RACES

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES 3-8
UNRULINESS	RUNNING
GROUP SETUP	2-4 STUDENTS SHARE 1 TABLET. GAME PLAY USES 2 SPLATS.

GAME SUMMARY

This activity is an adaptation of the six Splat example game, Relay Race. Place two Splats running distance apart. Students in each group take turns running across the room to stomp on their Splat until it turns green. Once green, they race back to jump on the other Splat at the finish line to get a point and it is the next student's turn. A relay race! A stopwatch keeps track of the time it takes to get 10 points and students compete for the fastest time. Groups first work independently, then come together for a head-to-head competition.

GAME RULES: JUMP ON SPLAT 1 UNTIL IT TURNS GREEN, STEP ON SPLAT 2 TO FINISH.

NOTES

PLAYING TOGETHER

★ Smaller groups take turns playing

★ Use tape to mark where students can stand while waiting for their turn

★ Create 'lanes' using tape or yardsticks

REMOTE PLAY

★ Splats web-app

★ Virtual breakout rooms

★ Students click on the Splats instead of stomping

RELAY RACES

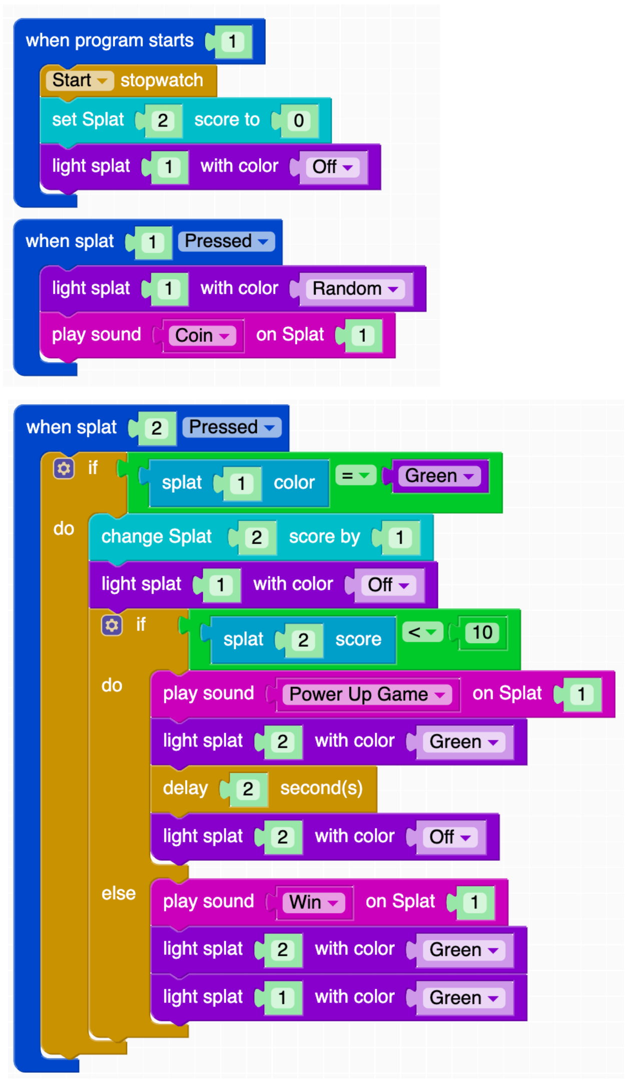
CODE KEY

HOW IT WORKS

This program uses nested conditional statements. In a conditional statement, if the condition is true the program will do one thing, if false it will do another. Nested statements allow for complex logic, because any blocks inside of conditional statements only run if the conditions for the outer statement are met.

When this program begins, a stopwatch starts. Splat 1 lights up a random color when Splat 1 is pressed. Once the random color is green, players run to jump on Splat 2 to get a point. Conditional statements (**if/do** and **if/do/else**) are used to add points and end the game when the group reaches 10 points. An **if/do** block creates a rule for the program that happens if the **if** condition is true. In an **if/do/else** block, an **else** rule is made for the scenario where the condition is not true. The first conditional statement is, **if** Splat 1 is pressed while green, change the Splat score by one. The second conditional statement is, **if** Splat 2 is pressed **and** the score is less than 10, light Splat 1 green and the group keeps playing. **else**, once the score is equal to 10, a sound is played and the game ends.

CODE IMAGE



```
when program starts 1
    Start stopwatch
    set Splat 2 score to 0
    light splat 1 with color Off

when splat 1 Pressed
    light splat 1 with color Random
    play sound Coin on Splat 1

when splat 2 Pressed
    if splat 1 color = Green
    do  change Splat 2 score by 1
        light splat 1 with color Off
        if splat 2 score < 10
        do  play sound Power Up Game on Splat 1
            light splat 2 with color Green
            delay 2 second(s)
            light splat 2 with color Off
        else
            play sound Win on Splat 1
            light splat 2 with color Green
            light splat 1 with color Green
```

RELAY RACES

ACTIVITY OUTLINE

1. INTRODUCE CHALLENGE & KEY CONCEPTS

Introduce the activity, explain how the game will be played, and demonstrate game play. Have groups brainstorm potential algorithms and the key elements for this program. Have groups write down an ordered list of rules for the game or create a flowchart.

2. WORK TIME (PT. 1)

Lead a class brainstorm about how to start the program. Review the code for **when program starts** and **when splat 1 pressed** as a class! Give groups time to build their code for the first two starting blocks. Assist groups in working through the **when splat 2 pressed** code. Have groups break down every element that needs to be included. Talk about the role of nesting in programs.

3. WORK TIME (PT. 2)

Support groups in rotating through different roles: coding, testing, debugging, documentation, etc. Give groups time to play with their code. Review code writing with the class.

4. STUDENT SHOWCASE!

Give groups time to present their code and describe key choices. If possible, have groups compete against each other in a full-class game, or play the biggest Relay Race you can!

GOING FURTHER

EXTENSIONS

This can be an independent activity. Encourage students to build the program in different ways.

SUPPORT

Provide students with the essential blocks and ask them to decide the order, in groups or as a class. Alternatively, provide the whole code to students and ask them to discuss what it does and how it works. If game play is too long, change the number of points needed to win from 10 to 3. You can also modify the example as a full class discussion.

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-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)

SPLATS DJ

ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES 3-8
UNRULINESS	STANDING
GROUP SETUP	2-4 STUDENTS SHARE 1 TABLET. GAME PLAY USES 2+ SPLATS.

GAME SUMMARY

You have a classroom of musicians! Lead students through an exploration of sounds and music using Splats. Get ready for some Unruliness!

GAME RULES: MAKE UNRULY MUSIC WITH SPLATS!

NOTES

PLAYING TOGETHER

★ Smaller groups

★ This activity can also be done independently via the web-app

★ To celebrate, classrooms can host a showcase to share musical creations!

REMOTE PLAY

★ Splats web-app

★ Virtual breakout rooms

★ The whole class can also contribute towards one piece of music! This can be a one-time exercise or ongoing symphony!

SPLATS DJ

CODE KEY

HOW IT WORKS

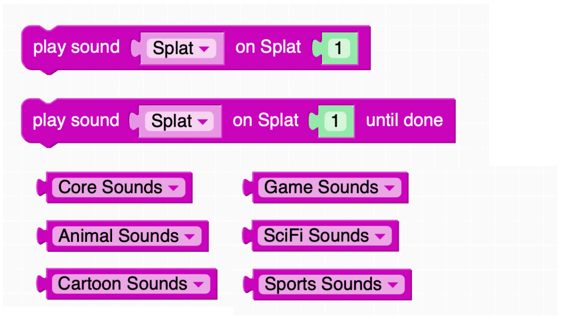
This is an open-ended activity. Players will explore the four different ways to make music on Splats: sounds, recordings, notes, and drums.

Our app uses MIDI numbers to play note pitches. Please refer to the conversion table on the next page for the **play note** and **turn off note** blocks. The blocks needed for each of the four sound types are outlined below.

To use the **recordings block**, click on the microphone on the right side-panel to record. Select the slot you would like to use and press record. When using the **recordings block**, select the slot you used from the drop-down.

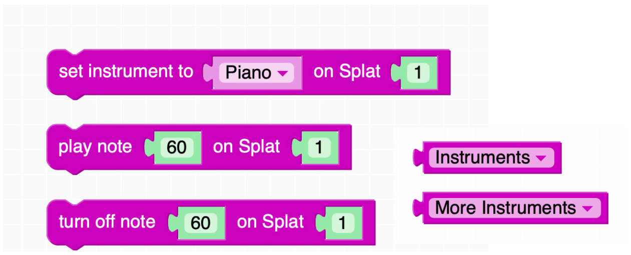
CODE IMAGE

SOUNDS



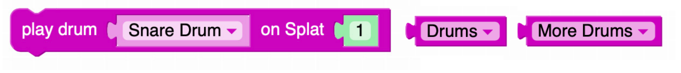
```
play sound [Splat ▾] on Splat [1]
play sound [Splat ▾] on Splat [1] until done
[Core Sounds ▾]      [Game Sounds ▾]
[Animal Sounds ▾]    [SciFi Sounds ▾]
[Cartoon Sounds ▾]   [Sports Sounds ▾]
```

NOTES



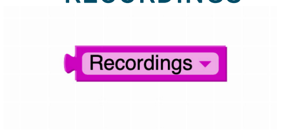
```
set instrument to [Piano ▾] on Splat [1]
play note [60] on Splat [1]          [Instruments ▾]
turn off note [60] on Splat [1]      [More Instruments ▾]
```

DRUMS



```
play drum [Snare Drum ▾] on Splat [1]   [Drums ▾]   [More Drums ▾]
```

RECORDINGS



```
[Recordings ▾]
```

SPLATS DJ

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)'.

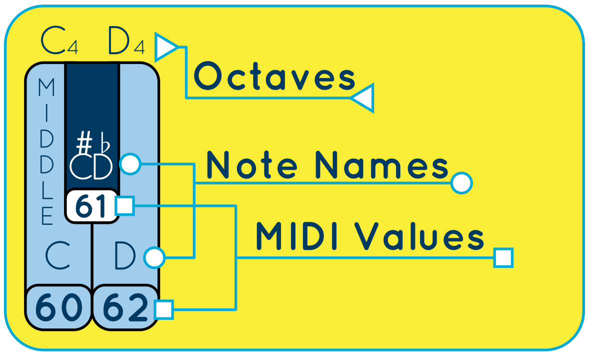
Because there are 12 notes from C to B in each octave, you can go up or down an octave by adding or subtracting 12 from that note's MIDI value.

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.

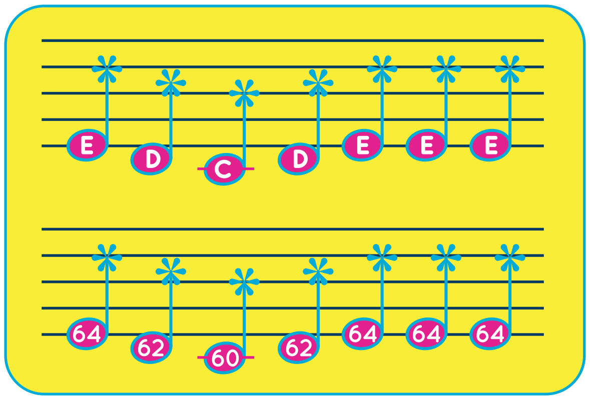
These MIDI conversions don't have to be limited to single notes either! You can build and experiment with basic three or five note chords played at the same time.

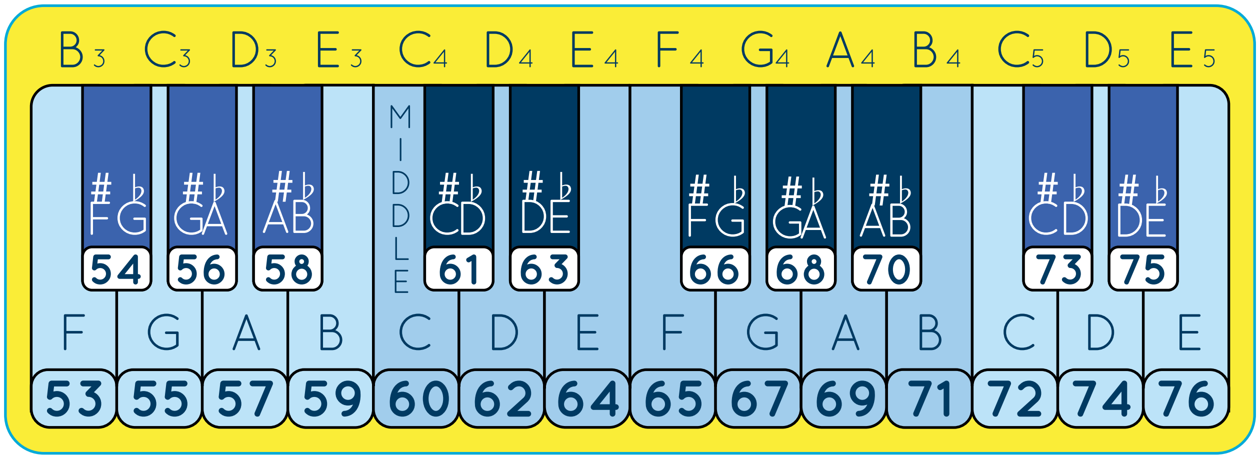
Some examples are C (60, 64, 67), E (64, 68, 71), and G# (68, 72, 75) or even a C major pentatonic (60, 62, 64, 67, 69).

KEY



EXAMPLE: MARY HAD A LITTLE LAMB





ACTIVITY OUTLINE

1. INTRODUCE CHALLENGE & KEY CONCEPTS

Introduce the activity. Tell students they will be creating their own musical creations. Students can use any of the blocks from the sounds tab for their composition. Everyone will perform their creation at the end of class. Show students the MIDI conversion page and explain how it works. Make sure to provide the MIDI note reference for students to use while working. Show students the blocks needed to code sounds, recordings, notes, and drums. Give students 5 minutes to explore the code for each. If your students are unfamiliar with music notes, you can also print example songs and chords as references.

2. WORK TIME

Give students time to build their own musical creations. Support groups in planning their creations and rotating through a variety of roles: coding, testing, debugging, documentation, etc.

3. STUDENT SHOWCASE!

Lead a class concert, having students share their musical creations and explain key decisions they made in their program development.

GOING FURTHER

EXTENSIONS

Have students combine their musical creations. Consider creating a full-class band or orchestra! If students are curious about MIDI files and how the conversion works, the link from the Code Key is a great resource. To extend this activity, challenge students to build specific songs, use more Splats, or mimic the sounds of specific environments (sounds of the bus, the grocery store, a party, etc.)

SUPPORT

If this activity is too open-ended, provide a song for students to re-create on Splats. This activity can be broken up into multiple days, focusing on different sound types each day. If students struggle with the MIDI conversion table, provide MIDI numbers for select notes or example songs.

CSTA STANDARDS: ALGORITHMS AND PROGRAMMING

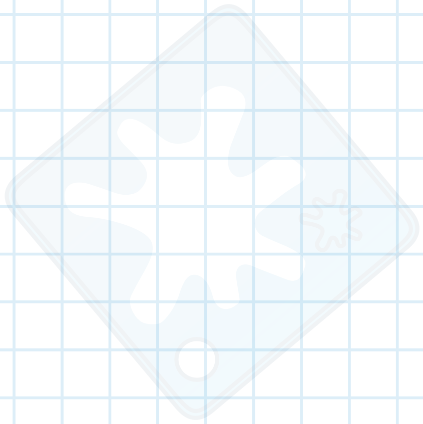
COMPUTER SCIENCE TEACHERS ASSOCIATION STANDARDS (CSTA) - GRADES 3-5

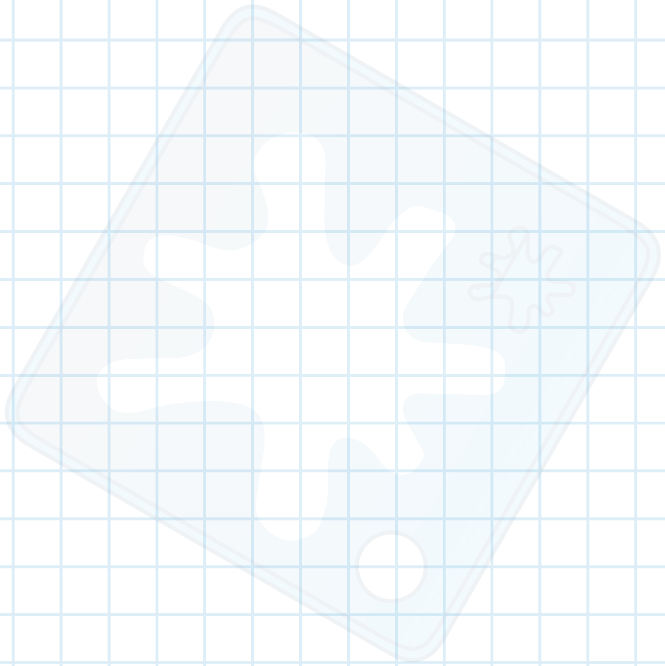
1B-AP-08 Algorithms	Compare and refine multiple algorithms for the same task and determine which is the most appropriate. (P6.3, 3.3)
1B-AP-10 Control	Create programs that include sequences, events, loops, and conditionals. (P5.2)
1B-AP-11 Modularity	Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. (P3.2)
1B-AP-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-12 Control	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. (P5.1, 5.2)
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)

SCRATCH SPACE





CAPTURE THE SPLAT ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES 3-5
UNRULINESS	RUNNING
GROUP SETUP	2-4 STUDENTS SHARE 1 TABLET. GAME PLAY USES 2+ SPLATS.

GAME SUMMARY

This game is an Unruly version of the classic, Capture the Flag. Instead of stealing a 'flag', teams try to run and stomp on the other team's Splat! Students do not play with their own code in this activity. Instead, students work in groups to understand the program then play the game as a class.

GAME RULES: STEP ON THE OPPOSING TEAM'S SPLAT TO GET A POINT, 5 POINTS WINS!

NOTES

PLAYING TOGETHER

- ★ Small groups can play simultaneously
- ★ To control movement, students roll a die to determine their # of steps
- ★ Once they get to the 'flag,' players must successfully win a round of rock paper scissors against the goalie(s)
- ★ If the player beats the goalie, they can attempt to throw a bean bag on top of the Splat to 'capture' it
- ★ If the bean bag does not land on the Splat, the goalie can attempt to make the same throw. If the goalie succeeds, the player is out. If the goalie fails, the player is free and must return to a designated starting point to try again

REMOTE PLAY

- ★ Splats web-app
- ★ Go over the code as a class
- ★ A student on each team is secretly given 'the flag' (this can be done through direct chat)
- ★ As a whole class, teams can ask the instructor questions in a 'guess who' format to figure out who has the flags, e.g., "Does this person wear glasses?"
- ★ First team to reach 5 points wins

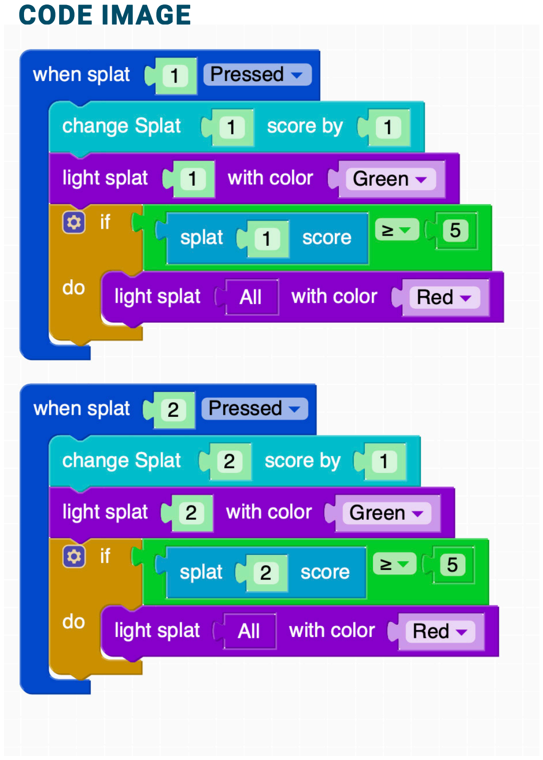
CODE KEY

HOW IT WORKS

This program adds points to the Splat when it is pressed. It uses simple conditional statements, the **if/do** block, to signal when the game is over by lighting the Splat red.

An **if/do** block creates a rule for the program that causes something to happen if the **if** condition is met

CODE IMAGE



```
when splat 1 Pressed
    change Splat 1 score by 1
    light splat 1 with color Green
    if splat 1 score ≥ 5
    do light splat All with color Red

when splat 2 Pressed
    change Splat 2 score by 1
    light splat 2 with color Green
    if splat 2 score ≥ 5
    do light splat All with color Red
```

CAPTURE THE SPLAT ACTIVITY

ACTIVITY OUTLINE

1. INTRODUCE CHALLENGE & KEY CONCEPTS

Introduce the basic game play of Capture the Flag. Explain how students will step on a Splat instead of stealing a flag. Lead groups in identifying key program needs and devolving and comparing algorithms. Have groups write down the game rules and identify key blocks.

2. BRAINSTORM & INSTRUCTION

Support groups in building their programs and rotating through key roles: coding, testing, debugging, documenting, etc. Encourage groups to work together when testing and problem solving.

3. STUDENT SHOWCASE!

Give groups time to present their game, reviewing key decisions and modifications. Make sure to take time to play as many group's games as possible!

GOING FURTHER

EXTENSIONS

Students can add lights and sounds to this activity. Students can also come up with their own rules for different versions of Capture the Splat. For example, students could add a Splat to the jail that, when released, would make a siren sound — alerting the other team of an escape.

SUPPORT

If rounds are too long, change the winning score from 5 to 3. For more practice on conditional statements, check out Relay Races.

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-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)

SPLATS MEMORY CHALLENGE ACTIVITY

GRADE LEVEL	SUGGESTED FOR GRADES 6-8
UNRULINESS	SITTING, WALKING
GROUP SETUP	2-4 STUDENTS SHARE 1 TABLET. GAME PLAY USES 6 SPLATS.

GAME SUMMARY

Students will create a memory game using functions. Six Splats are placed on the ground or tabletop. Two will light up quickly, then disappear. Press on the two Splats you think you saw!

GAME RULES: USE VARIABLES & FUNCTIONS TO KEEP TRACK OF WHICH SPLATS LIT UP GREEN.

NOTES

PLAYING TOGETHER

★ Small groups spaced apart can play simultaneously

★ Students can wear gloves or use objects to press the Splats in order to avoid spreading germs

★ This activity can also be played via the web-app

REMOTE PLAY

★ Splats web-app

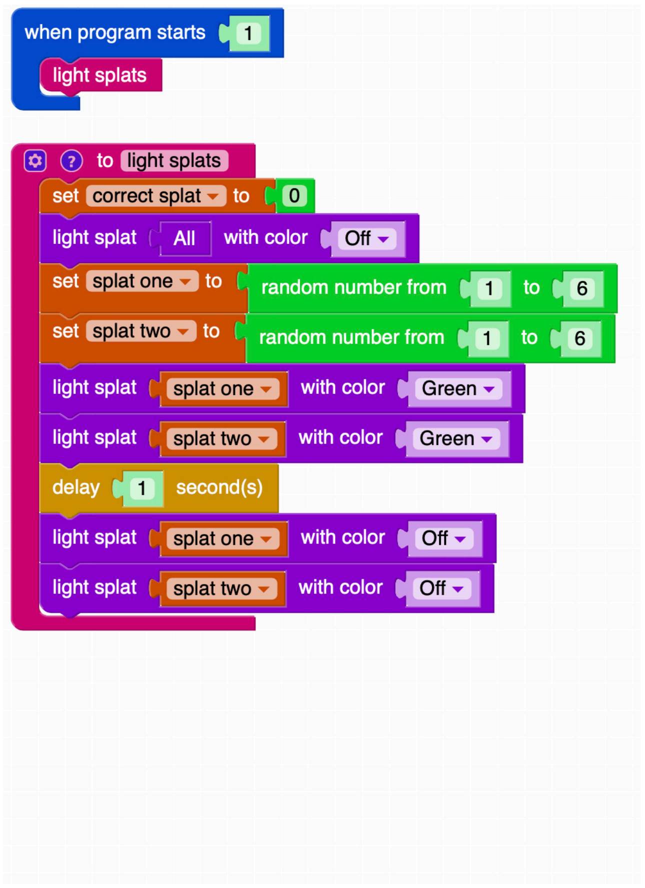
★ Virtual breakout rooms

CODE KEY

HOW IT WORKS(PT. 1)

Functions are a way to package code that performs a specific task. In the first part of this program the function, **light splats**, randomly lights up two out of six Splats green for 1 second and then turns them off. Functions are named by what they do. The purposes of using a function are: to simplify the number of rules a program has to process each time it runs, and to help make the program easier to debug, to read, and understand.

CODE IMAGE



```
when program starts 1
    light splats

to light splats
    set correct splat to 0
    light splat All with color Off
    set splat one to random number from 1 to 6
    set splat two to random number from 1 to 6
    light splat splat one with color Green
    light splat splat two with color Green
    delay 1 second(s)
    light splat splat one with color Off
    light splat splat two with color Off
```

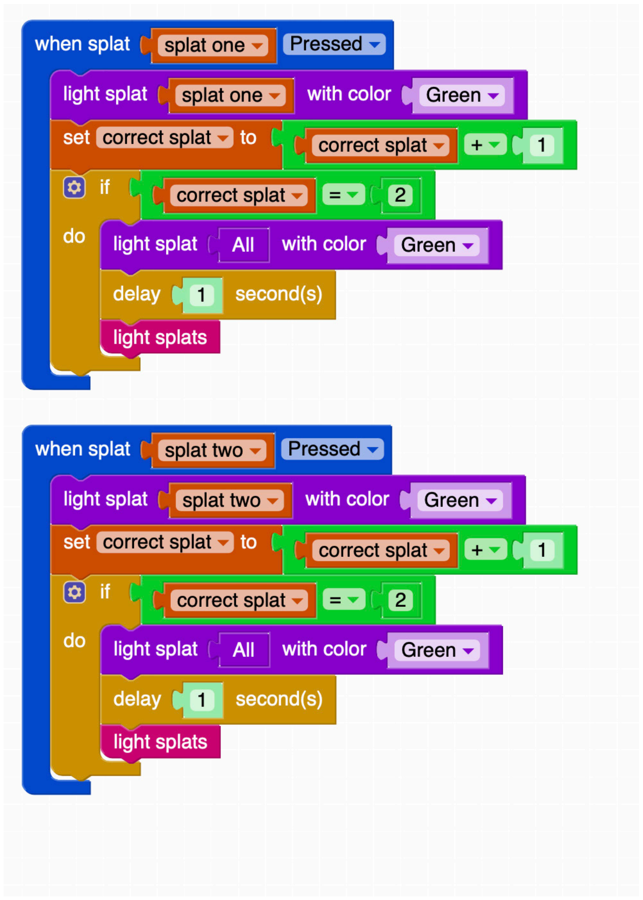
CODE KEY

HOW IT WORKS (PT. 2)

In the second part of this program, by using two conditional statements–**if/do** blocks–we create rules so that if the green Splats were correctly remembered and pressed, all six Splats light up green – and students know they beat the memory challenge.

Variables are used in order to know whether or not the Splats pressed were the correct green Splats. We use the variable **Splat One** and **Splat Two** to keep track of which random 2 out of the 6 Splats turned green. The variable **correct splat** keeps track of whether both **Splat One** and **Splat Two** were pressed. Note that in the **light splats** function, **correct splat** is always reset to 0, to make the next round begin.

CODE IMAGE



```
when splat [splat one] [Pressed]
  light splat [splat one] with color [Green]
  set [correct splat] to [correct splat] [+] [1]
  if [correct splat] [=] [2]
  do
    light splat [All] with color [Green]
    delay [1] second(s)
    light splats

when splat [splat two] [Pressed]
  light splat [splat two] with color [Green]
  set [correct splat] to [correct splat] [+] [1]
  if [correct splat] [=] [2]
  do
    light splat [All] with color [Green]
    delay [1] second(s)
    light splats
```

ACTIVITY OUTLINE

1. INTRODUCE CHALLENGE & KEY CONCEPTS

Demonstrate how to play the game at the front of the room. Have students brainstorm the essential blocks, key program needs, and compare algorithms.

2. WORK TIME (PT. 1)

Review the essential blocks and tie them directly to the game rules. Highlight the tole of functions and variables this program and support groups in breaking down the program development process.

3. WORK TIME (PT. 2)

Support groups in building their programs and rotating through essential roles: coding, testing, debugging, documentation, etc. Encourage groups to test each others programs and iterate based on feedback.

4. STUDENT SHOWCASE!

Give groups time to play and present their games, describing key decisions and modifications.

GOING FURTHER

EXTENSIONS

Ask groups to come up with ways to keep the variables **Splat One** and **Splat Two** from being assigned to the same Splat, ensuring that two different Splats always light up.

SUPPORT

The game can be slowed down by increasing the time on the **delay** blocks. Provide students with the structure of the full program. For additional help, have groups write out all the needed steps on paper first.

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)

SCRATCH SPACE