vol. 1

JUMP IN!

BLOCK CODING FOR UNRULY BEGINNERS

unruly splats

Vers. IP_4_26_20_SEL

DEAR EDUCATOR

Thank you for bringing Unruliness to your classroom! We have designed this unit to introduce students of all ages to our block coding language. We hope you enjoy these lessons as much as we enjoyed making them!

On the next page of this document are CSTA standards that this pack aligns to.

Please email us at educators@unruly-studios.com with any questions or feedback. We always love hearing from you!

Best,

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?

STANDARDS

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

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-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-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-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-14 Modularity
Create procedures with parameters to organize code and make it easier to reuse. (P4.1, 4.3)

ACTIVITY #1

LIGHTS AND SOUNDS

FOCUS BLOCKS

when splat pressed

light splat

play sound

SETUP

ONE SPLAT PER STUDENT OR PAIR

LEARNING GOALS

□ Use the **when splat pressed** block

□ Use the **light splat** block

□ Use the **play sound** block

VOCABULARY

Event Block - triggers when something happens

Code Block - gives instruction for a Splat to do something

Script - a series of instructions to be processed

ACTIVITY OUTLINE

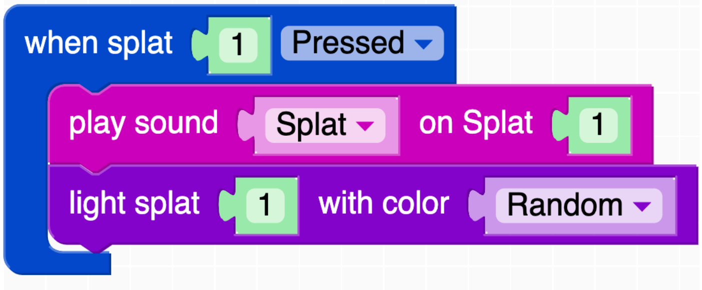
Begin with a **when splat pressed** block, and add the **lights** and **sounds** blocks inside.

When ready, hit your Splat to see it react! Use the drop down menu in the **light splat** block to pick different colors.

Use the drop-down menu in the **play sound** block to choose different noises.

Challenge: There is a choice called ‘random’ for both Lights and Sounds blocks. What does it do?

CODE KEY



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

ACTIVITY #1

LIGHTS AND SOUNDS

NOTES

Students will assemble this simple script and experiment with choosing different sounds and colors.

If students haven't used the Splats App before, demonstrate how to find these blocks in the App how to assemble them into a script.

All the 'Splat Number' fields can be left at the default '1' for this activity.

EXTENSIONS

Challenge: Can you make a Splat that turns green and sounds like a bell?

See game "Crack the Code" at the end of the pack for some ideas!

SUPPORTS

Students can explore the various light and sound options that can happen on a single Splat press. Let students know what their choices are for lights and sounds beforehand, and have them decide before engaging with Splats. For students who might have trouble reading, this means they'll be able to focus on finding 'chicken' out of the list of choices given rather than sounding out each individual word.

ACTIVITY #2

ONE SPLAT, TWO SPLAT

FOCUS BLOCKS

when splat pressed | light splat | all | play sound

SETUP

TWO SPLATS PER GROUP OR PAIR

LEARNING GOALS

□ Use the ‘Splat Number’ variable inside code blocks to control different Splats.

VOCABULARY

Splat Number - the number associated with each connected Splat. This number is also found in blocks, like **sounds** and **lights**.

ACTIVITY OUTLINE

Connect a second Splat to your code from Activity 1.

Depending on what ‘Splat Number’ you enter into a block, your script can control and communicate with each connected Splat.

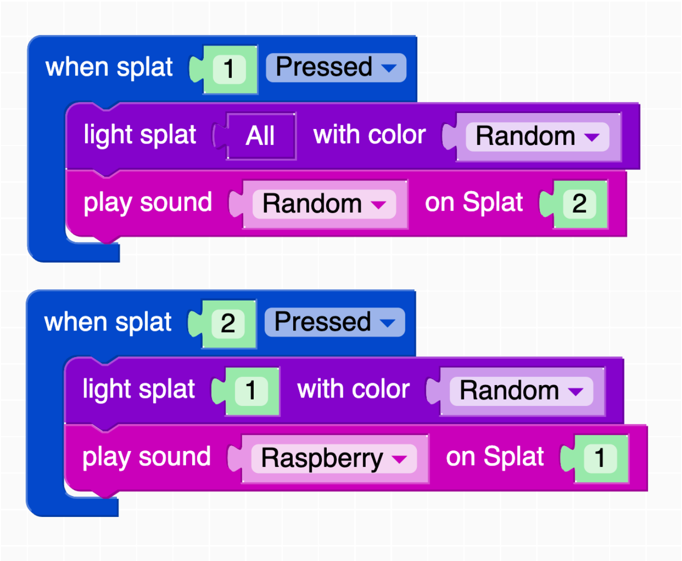
Change the ‘Splat Number’ on the **when splat pressed** block from a 1 to a 2. What happens?

Change the ‘Splat Number’ on the **lights block** from 1 to 2. What happens?

Change the ‘Splat Number’ on the Sound block. What happens?

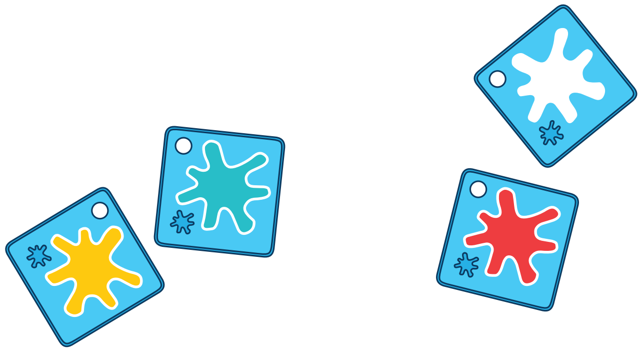
Make two scripts with Lights and Sound blocks -- one that begins **when splat (1) pressed** and another that begins **when splat (2) pressed**.

CODE KEY



```
when splat 1 Pressed
    light splat All with color Random
    play sound Random on Splat 2

when splat 2 Pressed
    light splat 1 with color Random
    play sound Raspberry on Splat 1
```



ACTIVITY #2

ONE SPLAT, TWO SPLAT

NOTES

After setup, point out to students that:
- ‘Splat 2’ is now visible in the editor window
- Each connected Splat has a unique ‘Splat Number’
- ‘Splat Number’ is also referenced in many code blocks by a small green number, which can be edited by clicking and typing.

EXTENSIONS

Challenge: There is a block in the lights block menu called **all** that you can use in place of a ‘Splat Number.’ Can you figure out what it does?

See game “Crack the Code” at the end of the pack for some ideas!

SUPPORTS

Students who are struggling may want to repeat the three ‘What happens?’ questions a few times.

- Change the ‘Splat Number’ on the **when splat pressed** block from a 1 to a 2. What happens?
- Change the ‘Splat Number’ on the **light** block from 1 to a 2. What happens?
- Change the ‘Splat Number’ on the **sound** block. What happens?

Troubleshooting: Some blocks will be ignored if more than one script begins with **when splat pressed** block.

ACTIVITY #3 - PART 1

KEEPING TRACK

FOCUS BLOCKS

when program starts | random number | light splat | all

SETUP

THREE SPLATS PER GROUP OR PAIR

LEARNING GOALS

□ Use the 'Random Number' block to create a surprise value.

VOCABULARY

Random Number - An unpredictable value from a set of possible values.

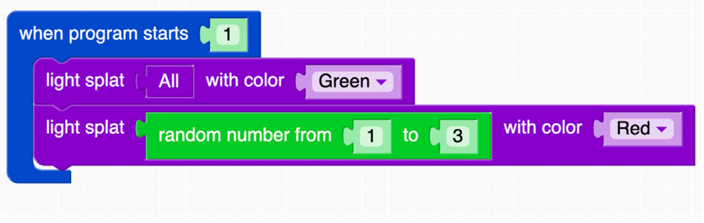
ACTIVITY OUTLINE

Making a random Splat change color

Using the code blocks from Activity 2 and the **when program starts** block, make a script that turns all Splats to the color green. Now, add a new **lights block** with a new color, and replace its 'Splat Number' with a **random number** block.

What happens? Why does it work this way?

CODE KEY



```
when program starts 1
    light splat All with color Green
    light splat random number from 1 to 3 with color Red
```

ACTIVITY #3 - PART 2

KEEPING TRACK

FOCUS BLOCKS

set variable

variable value

play sound

SETUP

THREE SPLATS PER GROUP OR PAIR

LEARNING GOALS

□ Create and set a variable.

□ Use the **variable block** inside other code blocks.

VOCABULARY

Variable - a number that has a specific name and a changeable value.

ACTIVITY OUTLINE

Which number did the **random block** pick?

Imagine this project as a game where a player must press on the Splat that changed color to make a sound. We know how to create a script that makes a sound when a Splat is pressed, but what value can we put in the 'Splat Number' spot when the Splat that changes color is chosen at random?

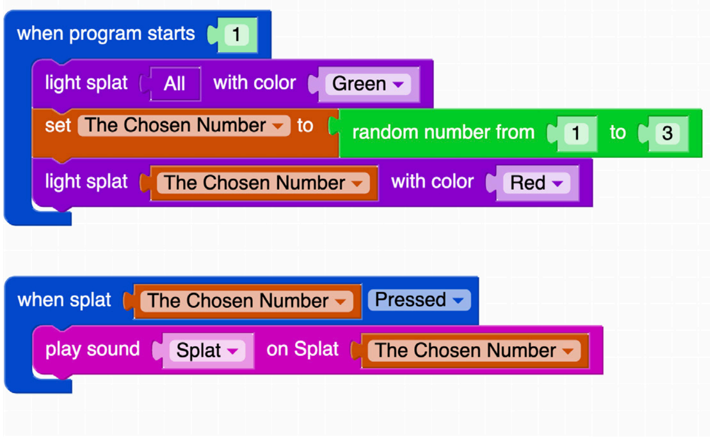
We need to talk about the Splat Number that was picked by the **random number** block, so give it a name, such as 'the chosen Splat number'. If we keep track of this number, we can share that information with other blocks.

Make a new variable and give it the name you chose.

Add a **set variable to** block and attach the **random number** block. This number can be used by both the **light splat** blocks, and the **when splat pressed** blocks.

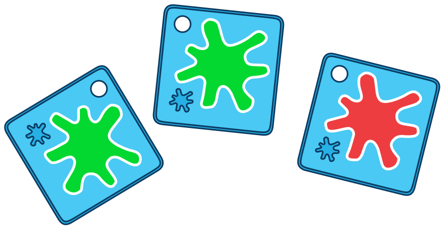
Challenge: Use this new variable in the 'Splat Number' spot in your scripts to make a sound when the randomly lit splat is pressed.

CODE KEY



```
when program starts 1
  light splat All with color Green
  set The Chosen Number to random number from 1 to 3
  light splat The Chosen Number with color Red

when splat The Chosen Number Pressed
  play sound Splat on Splat The Chosen Number
```



ACTIVITY #3

KEEPING TRACK

NOTES

The steps of this activity can be presented as explorations, demonstrations, or a mix of both. For explorations, each step can be presented as a design challenge, with solutions shared after work time.

In part 1, students use familiar blocks to make the beginnings of a simple game. After they assemble the script, pose the challenge question, “What should happen when the randomly lit Splat is pressed? How can we code a script to do it?”

Introducing variables can be confusing, so help your students by reminding them of the variable’s role in this program. For the program to work, the Lights and Sounds scripts each need to know ‘which Splat Number was randomly chosen’ every time the program is run. Creating a variable is just a way to give that number a name. Encourage students to choose a relevant variable name that includes the word ‘number’ to reinforce that the variable is an integer.

Start Part 2 by demonstrating how to create a variable and point out the difference between the **variable block** and the **set variable to** block.

EXTENSIONS

Challenge: What else could happen when the randomly lit Splat is pressed? Can you design a game based on this program?

See game “Crack the Code” at the end of the pack for some ideas!

SUPPORTS

Students may need help finding new blocks in the menu, and understanding the purpose of the variable block.

The **when program starts** block is a new block in this activity and may require brief explanation.

Troubleshooting: Make sure the random block is picking in range of how many Splats are connected.

SCRATCH SPACE

ACTIVITY #4 - PART 1

USEFUL FUNCTIONS

FOCUS BLOCKS

set variable | random number | new function

SETUP

THREE SPLATS PER GROUP OR PAIR

LEARNING GOALS

□ Identify a sequence of blocks that is repeated in part of a script.

VOCABULARY

Variable - a number that has a specific name and a changeable value.

Function - a specific sequence of blocks that has a name, and can be called in a script.

ACTIVITY OUTLINE

In the **when program starts** part of this code taken from Activity 3, notice how we have to press the RUN button in the Splats App if we want to 'reset' the program from the start and get a new random Splat to change color.

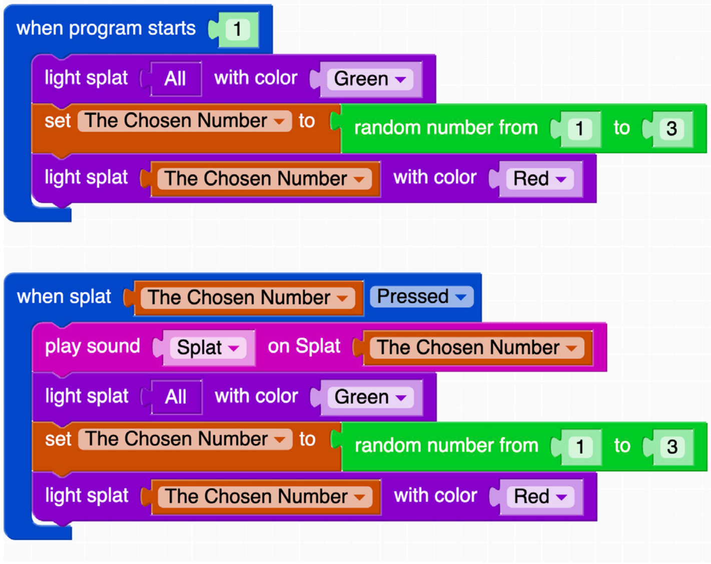
This **when splat pressed** block uses familiar blocks to make the game 'reset' after the player presses the Splat that changed color.

Notice a pattern? The 3 blocks that 'reset' the game are the same 3 that 'start' the game:

- Light all Splats green
- Set new variable to a random number
- Change the color of the Splat matching the new variable

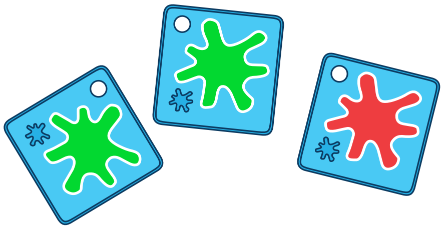
Wherever we see the routine of these 3 steps and their code blocks, we know that the game is being 'reset.' Choose a name for these steps so we can talk about them clearly, such as 'Reset the Splats'.

CODE KEY



```
when program starts 1
    light splat All with color Green
    set The Chosen Number to random number from 1 to 3
    light splat The Chosen Number with color Red

when splat The Chosen Number Pressed
    play sound Splat on Splat The Chosen Number
    light splat All with color Green
    set The Chosen Number to random number from 1 to 3
    light splat The Chosen Number with color Red
```



ACTIVITY #4 - PART 2

USEFUL FUNCTIONS

FOCUS BLOCKS

new function

call function

SETUP

THREE SPLATS PER GROUP OR PAIR

LEARNING GOALS

□ Define a new Function and call it in a script.

VOCABULARY

Function - a specific sequence of blocks that has a name, and can be called in a script.

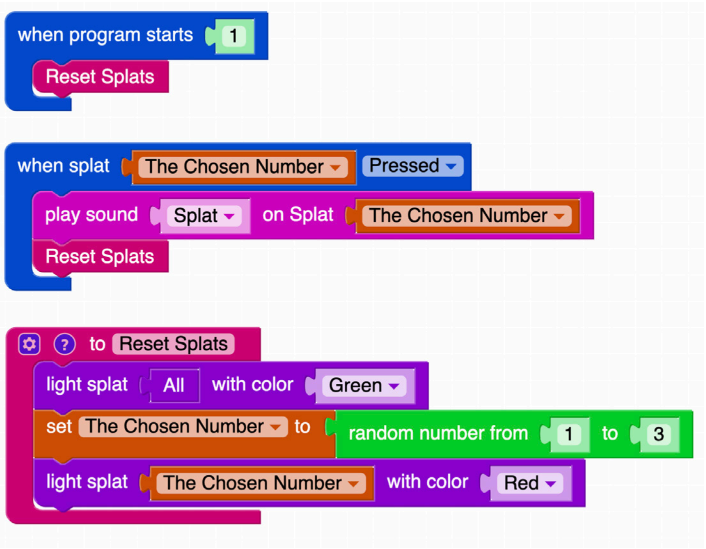
Calling a Function - using a defined Function in a script to perform an action.

ACTIVITY OUTLINE

In Activity 3, we created a variable by naming a randomized number so that different parts of the script could share it. We can also name specific sequences of code blocks so they can be shared by multiple scripts. A named sequence of blocks is called a function.

Create a new function by dragging out a **function block** and give it the name you chose in Part 1. Add the 3 blocks of the 'reset' sequence to the **function block**. With your function defined, a new block with your function's name will appear in the functions menu. Use this new block wherever you want a script to 'reset' the game.

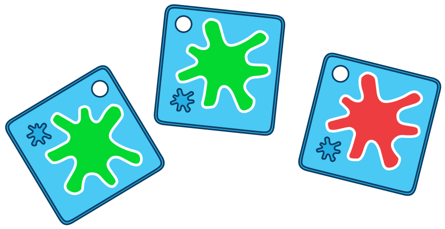
CODE KEY



```
when program starts 1
  Reset Splats

when splat The Chosen Number Pressed
  play sound Splat on Splat The Chosen Number
  Reset Splats

to Reset Splats
  light splat All with color Green
  set The Chosen Number to random number from 1 to 3
  light splat The Chosen Number with color Red
```



ACTIVITY #4

USEFUL FUNCTIONS

NOTES

Similar to variables, introducing functions can be confusing, so help your students by reminding them of the functions role in this program:

- Replacing the repeated sequence of blocks with a function reduces the number of blocks in the script making it easier to read and understand.
- Creating a function for useful procedures like ‘resetting the game’ makes it easier to use in multiple places within a project.
- It is easier to update one function definition than to update each repeated sequence of blocks individually.

EXTENSIONS

Challenge: What will the Splats do to tell a player they have won? Or lost? Create and name a new function for an important event in the game by combining lights and sounds blocks.

SUPPORTS

Students can get used to the idea of defining and calling functions for the light or sound block scripts they created in Activity 2.

SCRATCH SPACE

ACTIVITY #5 - PART 1

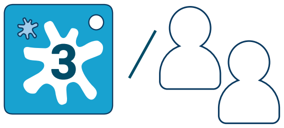
CONDITIONAL STATEMENTS

FOCUS BLOCKS

if do else

math 0 >

SETUP



THREE SPLATS PER GROUP OR PAIR

LEARNING GOALS

□ Use conditional statements to create a goal score.

VOCABULARY

Conditional Statement - A block that performs different actions depending on the result of a true-or-false test.

ACTIVITY OUTLINE

The code from Activity 4, and this part one code of Activity 5 will loop forever, resetting every time a player hits the correct Splat.

To make this a game that the player can win, we have to create:

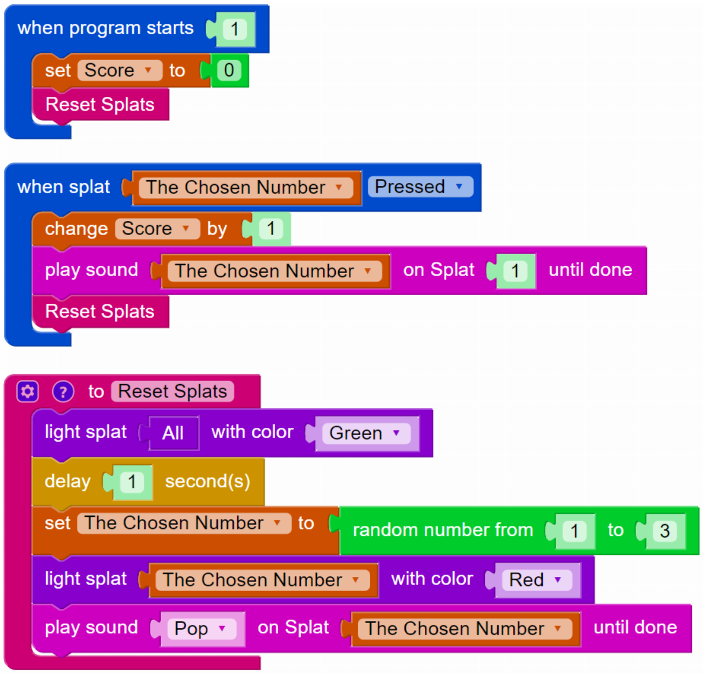
- A goal
- A way to keep track of the player's progress toward this goal

For this activity, we will add a **variable,** and call it Score, to keep track of how many times the player has hit the Splat. This lets us make the game end when the player has scored a number of points

Add a **set Score to**, and a **0 block** under **when program starts** so that the player will start with 0 points.

Add a **change score by**, and a **1 block** to the **when splat pressed** to increase the score by 1 each time the odd-colored Splat is pressed.

CODE KEY



```
when program starts 1
    set Score to 0
    Reset Splats

when splat The Chosen Number Pressed
    change Score by 1
    play sound The Chosen Number on Splat 1 until done
    Reset Splats

to Reset Splats
    light splat All with color Green
    delay 1 second(s)
    set The Chosen Number to random number from 1 to 3
    light splat The Chosen Number with color Red
    play sound Pop on Splat The Chosen Number until done
```

ACTIVITY #5 - PART 2

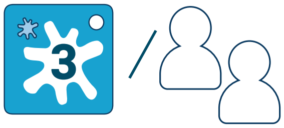
CONDITIONAL STATEMENTS

FOCUS BLOCKS

if do else

math 0 >

SETUP



THREE SPLATS PER GROUP OR PAIR

LEARNING GOALS

☐ Use conditional statements to create a goal score.

VOCABULARY

Function - a specific sequence of blocks that has a name, and can be called in a script.

Variable - a number that has a specific name and a changeable value.

ACTIVITY OUTLINE

The program is now keeping score, but will still continue looping and resetting forever. We want it to increase the score and reset every time the odd-colored Splat is pressed UNTIL the goal of 5 points is reached.

We need a way to see if we reach our goal score!

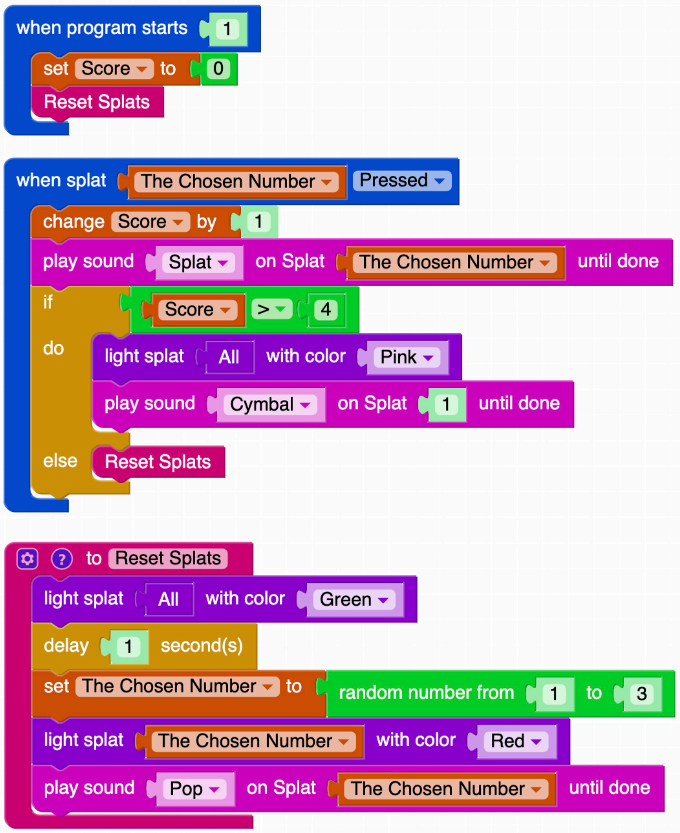
For this we use a new control block called the **if/do/else** block. This block asks a true-or-false question, and then performs different actions based on the answer. In this program, the question will be 'Has the player reached the goal of 5 points?' If that's true, then the program will not 'reset' the round and the game will end. If that's false, then the program will use the 'reset' function and the game will continue.

Add the **if/do/else** block to the **when splat pressed** script.

Use math and variables blocks to create a true-or-false test that checks whether the score variable is greater than 4. Add these blocks next to the **if** part of the control block.

Add some new lights and/or sounds blocks to the **do** part of the Control block that will tell the player they have won. Move the **reset** function block to the **else** part of the control block.

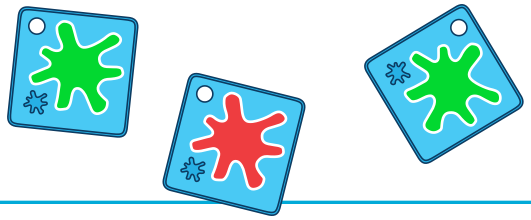
CODE KEY



```
when program starts 1
  set Score to 0
  Reset Splats

when splat The Chosen Number Pressed
  change Score by 1
  play sound Splat on Splat The Chosen Number until done
  if Score > 4
  do
    light splat All with color Pink
    play sound Cymbal on Splat 1 until done
  else
    Reset Splats

to Reset Splats
  light splat All with color Green
  delay 1 second(s)
  set The Chosen Number to random number from 1 to 3
  light splat The Chosen Number with color Red
  play sound Pop on Splat The Chosen Number until done
```



ACTIVITY #5

CONDITIONAL STATEMENTS

NOTES

Begin by explaining the objective of this activity -- add a win condition to the game and stop it from resetting forever. Confirm that students understand why they are creating these new variables and Control blocks.

As this program becomes more complex, the specific order of blocks in the script becomes more important to functioning correctly. Students may need help and extra time for debugging.

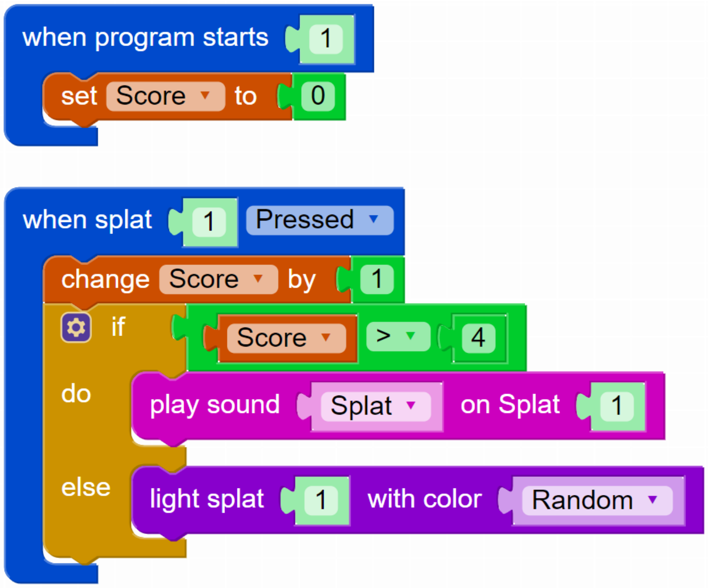
EXTENSIONS

Challenge: See Game Design Extension Activities.

Challenge: Using conditional statement blocks, have the odd-colored Splat make different sounds depending on the player's score.

SUPPORTS

It could be helpful to have students experiment with the conditional statement block in a new project before adding it to the existing program from Activity 4. Here is an image for an example of a simplified conditional statement project!



```
when program starts 1
    set Score to 0

when splat 1 Pressed
    change Score by 1
    if Score > 4
    do
        play sound Splat on Splat 1
    else
        light splat 1 with color Random
```

EXPLORATION GAME

CRACK THE CODE

GRADE LEVEL	SUGGESTED FOR GRADES K-8
UNRULINESS	SITTING
GROUP SETUP	GAME PLAY USES 1-3 SPLATS FOR EACH PAIR OF PLAYERS, WRITING / DRAWING MATERIALS FOR EACH PAIR

GAME SUMMARY

This game has two roles: **CODER** and *INVESTIGATOR*. Choose who will be the **CODER** first.

1. The **CODER** takes the device and has 2 minutes to create a simple program, making the connected Splats behave in a particular way.

2. The *INVESTIGATOR* takes the connected Splats, and has two minutes to experiment with them to see how the Splats react.

3. Without Looking at the device screen, the *INVESTIGATOR* has one minute to draw a picture of what they think the **CODER'S** program looks like.

4. The *INVESTIGATOR* presents their drawing to the **CODER** and explains their thinking, while the **CODER** reveals their program.

5. Players take two minutes to discuss what happened, with the *INVESTIGATOR* sharing what they observed that lead to their guess, and the **CODER** explaining any differences from the script.

6. Switch roles and play again! Reflect on what blocks are the easiest / hardest to guess, and what clues stand out the most to the *INVESTIGATOR*.

Variation Instead of building a new script from scratch each round, the new **CODER** uses their two minutes to modify the code from the previous round and the *INVESTIGATOR* guesses what changed!

NOTES

This game works best when the **CODER** is limited to using blocks the *INVESTIGATOR* is familiar with. When reading the rules, the class should decide on a list of allowed blocks to use!

Challenge - Add a new block type to the list of allowed blocks and play again.

Variation This activity can also be done with the whole class, acting as *INVESTIGATORS* for one **CODER**.