

# RED SPLAT GREEN SPLAT

## GAME SETUP

2-4      1

PLAYERS



SPLAT



## GAME SUMMARY

**UNRULINESS:** Running, Full classroom

**GAME RULES:** Change the Splat color from green to red at random intervals

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.



GREEN=GO!  
RED=STOP!

## 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
- If using the preloaded example, DO NOT stomp

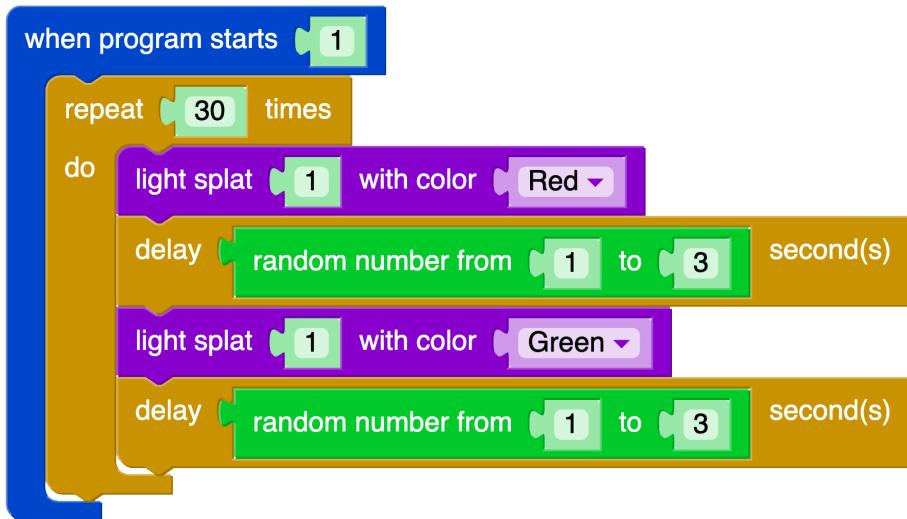
### REMOTE PLAY

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



## 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.



## SUGGESTED OUTLINE



### 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.



### 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.



### 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.



### 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 an opportunity to describe their development processes and the choices made.

## GOING FURTHER

### EXTENSION

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

COMPUTER SCIENCE TEACHERS  
ASSOCIATION STANDARDS

## ALGORITHMS & PROGRAMMING

### GRADES K–2

|                                 |   |
|---------------------------------|---|
| <b>1A-AP-10<br/>CONTROL</b>     | Develop programs with sequences and simple loops, to express ideas or address a problem. ( <b>P5.2</b> )                              |
| <b>1A-AP-11<br/>MODULARITY</b>  | Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. ( <b>P3.2</b> )                   |
| <b>1A-AP-12<br/>DEVELOPMENT</b> | Develop plans that describe a program's sequence of events, goals, and expected outcomes. ( <b>P5.1, 7.2</b> )                        |
| <b>1A-AP-14<br/>DEVELOPMENT</b> | Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops. ( <b>P6.2</b> )                  |
| <b>1A-AP-15<br/>DEVELOPMENT</b> | Using correct terminology, describe steps taken and choices made during the iterative process of program development. ( <b>P7.2</b> ) |

# CSTA STANDARDS

COMPUTER SCIENCE TEACHERS  
ASSOCIATION STANDARDS

## ALGORITHMS & PROGRAMMING

### GRADES 3–5

|                                 |  |
|---------------------------------|--|
| <b>1B-AP-08<br/>ALGORITHMS</b>  | Compare and refine multiple algorithms for the same task and determine which is the most appropriate. <b>(P6.3, 3.3)</b>                             |
| <b>1B-AP-10<br/>CONTROL</b>     | Create programs that include sequences, events, loops, and conditionals. <b>(P5.2)</b>   |
| <b>1B-AP-11<br/>MODULARITY</b>  | Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. <b>(P3.2)</b>                    |
| <b>1B-AP-13<br/>DEVELOPMENT</b> | Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences. <b>(P1.1, 5.1)</b> |
| <b>1B-AP-15<br/>DEVELOPMENT</b> | Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. <b>(P6.1, 6.2)</b>                                    |
| <b>1B-AP-17<br/>DEVELOPMENT</b> | Describe choices made during program development using code comments, presentations, and demonstrations. <b>(P7.2)</b>                               |

