

Project 2: Simon Says

Purpose

Demonstrate use of conditionals, loops, String methods, and random numbers

Methods are **not** required for this project; however, it will significantly improve your code cleanliness.

Description

Your task is to write a program that plays “Simon Says” with the program user. [Simon Says](#) is a game where a user repeats a presented pattern. Your Simon Says program will support 2 modes. The user will choose which mode to play at the start of the game and beginning of each round.

- “easy” mode will play the game with colors
 - colors: yellow, green, red, and blue
- “hard” mode will play the game with numbers
 - numbers (single digit): 0 1 2 3 4 5 6 7 8 9

At the end of each round (ie. the user has “lost”), the user should be presented with the maximum number of sequences they correctly repeated. Then the user should be prompted to do another round and, if so, they should be prompted to select the mode.

Requirements

- You may only use tools discussed so far in this course: data types, String methods, random numbers, if statements, switch statements, and loops (for, while, and do while). You are not required to use methods, but they will help with code cleanliness. Use of arrays, ArrayLists, HashMaps, etc. is **not** allowed.
- If the user enters an invalid mode, continue prompting until they enter a valid mode.
- After “Simon says: ___”, program should wait 3 seconds and clear the screen, and then get input from the user. See Provided Code Hints section on how to accomplish this.
- In a round (regardless of mode), Simon adds one thing to remember.
- When checking if the user has entered the correct sequence, you should **ignore spaces AND ignore casing** (upper / lower case lettering).
 - ex. Simon Says: red green yellow, user can repeat red green yellow OR redgreenyellow OR RED greenYELLow
- When a round is “lost” (the user did not enter the correct sequence), the user should be offered to play another round - if the user enters “no”, the program should exit. If the user enters “yes”, prompt the user for a mode and start a new round. Repeat question until user enters “yes” or “no”.
- The score should be reset each round.

Provided Code / Hints

Method that clears screen. Place this method in your class above the main method.

```
public static void clearScreen() {  
    System.out.print("\033[H\033[2J");  
    System.out.flush();  
}
```

You can now clear the screen by calling `clearScreen()`; from other methods (like `main`)

Code that sleeps / pauses program for 3 seconds:

```
try {  
    Thread.sleep(3000);  
} catch (InterruptedException e) {  
    Thread.currentThread().interrupt();  
}
```

Source code getting messy? Your Java extension for VSCode includes a formatter. In the Command Palette (accessed with `Ctrl + Shift + P` in Windows), select “Format Document”

Example

```
Let's play Simon Says!  
Select difficulty (easy / hard): nice  
Invalid difficulty  
Select difficulty (easy / hard): easy  
Easy mode - colors  
Simon says: green  
[wait 3 seconds, clear screen]  
Player repeats: green  
Score: 1  
Simon says: green yellow  
[wait 3 seconds, clear screen]  
Player repeats: green yellow  
Score: 2  
Simon says: green yellow green  
[wait 3 seconds, clear screen]  
Player repeats: greenyellowgreen  
Score: 3  
Simon says: green yellow green blue  
[wait 3 seconds, clear screen]  
Player repeats: ahhh  
Round over! Your score was 3  
Would you like to play another round? (yes / no) yes  
Select difficulty (easy / hard): hard  
Hard mode - numbers  
Simon says: 3  
[wait 3 seconds, clear screen]  
Player repeats: 2  
Would you like to play another round? (yes / no) no  
Thanks for playing!
```

Rubric

Notes:

- Projects that don't compile will receive a 0.
 - It is recommended to use block comments if you have non-compiling sections of code.
- If you use a paid solution site (Chegg, CourseHero, etc.), you will receive an F in the course.
- You may only use tools discussed so far in this course: data types, String methods, random numbers, if statements, switch statements, and loops (for, while, and do while). You are not required to use methods, but they will help with code cleanliness. Use of arrays, ArrayLists, HashMaps, etc. is **not** allowed.

Grading:

- [10 pts] The user can choose the difficulty level, and invalid choices are rejected until user enters valid choice (easy / hard).
- [20 pts] Easy mode works correctly.
- [20 pts] Hard mode works correctly.
- [10 pts] Spacing and capitalization are ignored when validating the user's answer.
- [10 pts] The user's score is displayed during and at the end of each round.
- [10 pts] The user can choose if they want to play again, and invalid choices are rejected until user enters a valid choice (yes / no)
- [20 pts] The program is clearly organized, commented, and follows standard coding practices, including variable and class names.